

# **Popularity metrics and forecasting for Social Networks – Analyze-me**

---

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
Bachelor  
in  
Information and Communication Systems Engineering  
University of Aegean

---

**by**  
**Pericles A. Leros**

SPRING SEMESTER 2012

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ ΕΠΙΚΥΡΩΝΕΙ  
ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΤΟΥ ΠΕΡΙΚΛΗ Α. ΛΕΡΟΥ:

---

Γιάννης Χαραλαμπίδης, Επιβλέπων

Ημερομηνία: 18/9/2012

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

---

Ευριπίδης Λουκής, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

---

Σπύρος Κοκολάκης, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2012



## Abstract

In the framework of this Diploma Thesis a social media analyzing application has been developed based on statistical analysis of user's behavior and actions on certain social networking services such as Twitter and LinkedIn. The application consists of a website ([www.analyze-me.com](http://www.analyze-me.com)) implemented in PHP. The website is based mainly on the KLOUT API and Twitter API. Also, a database has been developed to store user statistics over time which are used as input data to the predictive algorithm for the forecasting capability of the application. Specifically, a Kalman filter prediction system is implemented, resulting from analysis and comparison of several predictive algorithms based on real-life data recorded from random users. Additionally, the LinkedIn API is used to authenticate the user so he can manually input the needed data to proceed with LinkedIn analysis. Finally, a mobile version of the application has been developed with the Android operating system used in most modern mobile devices. The application for mobile devices offers the same services as the website, and both share the same database.

© 2012

PERICLES A. LEROS

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

## Περίληψη

Στο πλαίσιο της παρούσας Διπλωματικής Εργασίας αναπτύχθηκε μια εφαρμογή ανάλυσης κοινωνικών μέσων βασισμένη σε στατιστική επεξεργασία της συμπεριφοράς και ενεργειών των χρηστών στα διάφορα κοινωνικά δίκτυα όπως το Twitter και το LinkedIn. Η εφαρμογή αποτελείται από μια ιστοσελίδα ([www.analyze-me.com](http://www.analyze-me.com)) υλοποιημένη σε γλώσσα PHP. Η ιστοσελίδα βασίζεται κυρίως στη χρήση του KLOUT API και του Twitter API. Επίσης, αναπτύχθηκε μια βάση δεδομένων για την καταγραφή στατιστικών στοιχείων των χρηστών στο χρόνο τα οποία αποτελούν βασικά δεδομένα εισόδου στον προβλεπτικό αλγόριθμο για την προγνωστική δυνατότητα της εφαρμογής. Ειδικότερα, ένα Kalman filter σύστημα πρόβλεψης υλοποιείται μετά από ανάλυση και σύγκριση μερικών προβλεπτικών αλγορίθμων βάσει πραγματικών δεδομένων που καταγράφηκαν από τυχαίους χρήστες. Επιπρόσθετα, χρησιμοποιείται το LinkedIn API για την αυθεντικοποίηση του χρήστη έτσι ώστε να μπορεί να εισάγει τα απαραίτητα στοιχεία για την περαιτέρω LinkedIn ανάλυση. Τέλος, αναπτύχθηκε η αντίστοιχη εφαρμογή στο λειτουργικό σύστημα Android για κινητές συσκευές. Η εφαρμογή για κινητές συσκευές παρέχει τις ίδιες δυνατότητες με την ιστοσελίδα, και οι δυο μοιράζονται την ίδια βάση δεδομένων.

© 2012

του

ΠΕΡΙΚΛΗ Α. ΛΕΡΟΥ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

## THANKS

The current Diploma Thesis was prepared by the pre-graduate student Pericles A. Leros of the “Information and Communication Systems Engineering” department of “University of Aegean”.

Through this Diploma Thesis I was given a chance to expand my knowledge in web programming and especially in android development for mobile devices. Also had the opportunity to review some basic forecasting techniques and actually apply the Kalman filtering algorithm for predictions using time series data.

I would like to thank the Assistant Professor Dr. Yannis Charalabidis for his guidance, support and help during the whole development phase.

Finally, I would like to thank my family and friends for the help and support they gave me all these years of my studies.

## Table of Contents

Abstract.....	iv
Περίληψη.....	v
THANKS .....	vi
Acronym list.....	ix
List of Figures.....	ix
1. Introduction.....	1
2. Review of Time-Series Prediction Algorithms .....	3
2.1 Curve Fitting, Least Squares Method, Regression .....	3
2.1.1 Curve Fitting.....	3
2.1.2 Regression.....	3
2.1.3 Least Squares Method.....	3
2.1.4 The Least-Squares Line .....	4
2.1.5 The Least-Squares Regression Line in Terms of Sample Variances and Covariance.....	5
2.1.6 MATLAB Least-Squares Regression - Polynomial Curve Fitting.....	6
2.2 Basics of Time-Series Data Analysis.....	9
2.2.1 Time-Series - Data Description .....	10
2.2.2 Time-Series - Modeling Stationary stochastic processes .....	10
2.2.3 The correlogram .....	11
2.2.4 Some classes of univariate time-series linear models .....	13
2.2.5 The purely random process or white noise .....	13
2.2.6 The random walk .....	14
2.2.7 Autoregressive processes .....	14
2.2.8 Moving average processes.....	14
2.2.9 The random walk plus noise model .....	15
2.2.10 The ARMA processes .....	15
2.2.11 ARIMA processes .....	16
2.2.12 SARIMA processes.....	17
2.2.13 Model Estimation.....	17
2.2.14 Diagnostic Model Checking.....	18
2.2.15 Model Forecasting.....	18
2.2.16 Box-Jenkins procedure – Summary.....	20
2.2.17 MATLAB Box-Jenkins Methodology .....	20
2.3 Time-Series - State space models .....	24

2.3.1	Forecasting with state-space models – the Kalman filter .....	27
2.3.2	Autoregressive Identification as a Kalman filter problem .....	28
2.3.3	MATLAB – Autoregressive Identification with Kalman filter.....	29
2.4	Time-Series – Neural networks Non-linear model and Forecasting .....	31
3.	Social Networking and Related Work.....	35
3.1	Social networking.....	35
3.1.1	Twitter .....	36
3.1.2	Twitter API.....	36
3.1.3	LinkedIN .....	37
3.1.4	LinkedIn API.....	37
3.2	Related Work - Social Media Analytics: KLOUT .....	38
3.2.1	KLOUT .....	38
3.2.2	EdgeRank Checker.....	39
3.2.3	Twitalyzer.....	39
4.	Architecture of Analyze-me Application.....	40
5.	Development and Implementation of the Analyze-me Application .....	41
5.1	Interface development .....	41
5.2	Database development.....	43
5.3	Prediction Interface development .....	44
5.4	Implementation of the Analyze-me Application.....	45
5.4.1	Twitter Analysis .....	45
5.4.2	LinkedIn Analysis.....	49
6.	Real Analyze-me Data and Selection of Prediction Algorithm .....	54
6.1	Real Analyze-me Data.....	54
6.2	Real Analyze-me Data: Least-Squares Regression-Polynomial Curve Fitting.....	54
6.3	Real Analyze-me Data: Box-Jenkins Methodology.....	55
6.4	Real Analyze-me Data: Autoregressive Identification with Kalman filter.....	61
6.5	Comparative analysis - Prediction Algorithm Selection.....	63
6.6	Description of the program used to develop the Prediction Algorithm.....	64
7.	Use of Analyze-me Application.....	66
7.1	Home .....	66
7.2	Features .....	66
7.3	About us.....	67
7.4	Mobile App.....	68



7.5 Get Started .....	69
7.5.1 Twitter Analysis .....	69
7.5.2 LinkedIn Analysis.....	72
8. Android and Mobile Analyze-me Application.....	73
8.1 What is android.....	73
8.2 Mobile Analyze-me Application.....	74
9. Conclusion and future plans.....	80
REFERENCES.....	81
Source code Appendix .....	82

## Acronym list

SMS	Short Message Service
SQL	Structured Query Language
APP	Application
ADMIN	Administrator
ADT Plugin	Android Development Tools Plugin
OAuth	Open standard for authorization
SSE	Sum-square-error
IDA	Initial Data Analysis
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive moving average
SARIMA	Seasonal autoregressive Integrated Moving Average
ACF	Autocorrelation function
PACF	Partial autocorrelation function
CPI	Consumer Price Index
AR	Autoregressive
NNs	Neural Networks

## List of Figures

Figure 1: Scatter plot data set points $(x_i, y_i)$ .....	7
Figure 2: Plot of straight line $y = a + bx$ on top of data set points $(x_i, y_i)$ .....	8
Figure 3: Stem plot of straight line $y = a + bx$ errors vs. data set $x_i$ values.....	8
Figure 4: Plot of 3rd degree polynomial on top of data set points $(x_i, y_i)$ .....	9
Figure 5: Stem plot of 3rd degree polynomial curve errors vs. data set $x_i$ values.....	9
Figure 6: Box-Jenkins procedure to time-series analysis.....	20
Figure 7: Plot of Quarterly Australian Consumer Price Index (CPI) from 1972 to 1991 .....	21

Figure 8: Plot of sample ACF and PACF of CPI .....	21
Figure 9: Plot of Differenced Quarterly Australian Consumer Price Index (CPI) from 1972 to 1991 .....	22
Figure 10: Plot of sample ACF and PACF of Differenced CPI .....	22
Figure 11: Plot of residuals for goodness of model fit.....	23
Figure 12: ARIMA(2,1,0) Forecasts for the Australian CPI and approximate 95% forecast intervals for the next 4 years (16 quarters) .....	24
Figure 13: MATLAB – Autoregressive Identification with Kalman filter for n=2.....	31
Figure 14: MATLAB – Autoregressive Identification with Kalman filter for n=3.....	31
Figure 15: Architecture for a typical NN for time-series forecasting with three inputs (the lagged values at $(t - 1)$ and $(t - 4)$ , and a constant), one hidden layer of two neurons with logistic functions, and one output linear neuron (the forecast).....	32
Figure 16: Social Network.....	35
Figure 17: Social Network Logos .....	36
Figure 18: Twitter Logo .....	36
Figure 19: LinkedIn Logo .....	37
Figure 20: KLOUT Logo .....	38
Figure 21: KLOUT score graph.....	38
Figure 22: Architecture of Analyze-me application.....	40
Figure 23: Analyze-me dark template.....	41
Figure 24: Joomla! Article manager.....	41
Figure 25: Joomla! Menu Item Manager.....	42
Figure 26: Analyze-me main menu bar .....	42
Figure 27: Photoshop working environment.....	42
Figure 28: phpMyAdmin control panel .....	43
Figure 29: Database test table .....	44
Figure 30: Sample php file .....	45
Figure 31: Twitter analysis getting started.....	45
Figure 32: LinkedIn analysis getting started .....	50
Figure 33: LinkedIn results page explained .....	52
Figure 34: Scatter Plot of RealAnalyze-me data values .....	55
Figure 35: Real Analyze-me data values (SCORES) .....	57
Figure 36: Plot of sample ACF and PACF of real Analyze-me data values (SCORES) .....	57
Figure 37: Plot of Differenced real Analyze-me data values (SCORES).....	58
Figure 38: Plot of sample ACF and PACF of Differenced real Analyze-me data values (SCORES)..	58
Figure 39: Plot of residuals for goodness of ARIMA(2,1,0) model fit .....	59
Figure 40: ARIMA(2,1,0) Forecasts for the next 16 data values (SCORES) .....	60
Figure 41: Error of ARIMA(2,1,0) Predictor .....	60
Figure 42: Real Analyze-me data and Kalman Filter Predictions with window size n=3 .....	62
Figure 43: Kalman Filter Prediction Errors with window size n=3 .....	63
Figure 44: Matlab sample plot.....	64
Figure 45: Matlab workspace.....	65
Figure 46: Analyze-me home page .....	66
Figure 47: Analyze-me Features page.....	67
Figure 48: Analyze-me About us page .....	67

Figure 49: Analyze-me mobile app page.....	68
Figure 50: Analyze-me mobile app page.....	68
Figure 51: Twitter analysis results.....	69
Figure 52: Twitter score graph.....	70
Figure 53: Prediction logo .....	70
Figure 54: Prediction results page.....	71
Figure 55: Prediction algorithm information.....	71
Figure 56: Declining score .....	71
Figure 57: Rising score .....	71
Figure 58: Same score.....	71
Figure 59: Linkedin results page.....	72
Figure 60: Android home screen.....	73
Figure 61: Android customized by HTC lock screen .....	73
Figure 62: Eclipse logo .....	74
Figure 63: Mobile application home screen.....	74
Figure 64: Mobile application get started screen .....	75
Figure 65: Mobile application results page .....	76
Figure 66: Mobile application feedback/rate screen .....	78
Figure 67: Mobile application features and help screen .....	79
Figure 68: Mobile application about screen.....	79

## 1. Introduction

We live in a society where everyone feels the need to be popular within his social circles. As the social media keep getting more and more popular, the need to feel important within a social group is getting stronger. Being popular though requires some kind of recognition, a service that can rank your popularity level and establish a general measurement scale in order to compare with others.

There has been a very good effort by a San Francisco-based company called KLOUT that provides social media analytics and ranking of a user's influence across his or her social network. KLOUT offers a ranking system and has established a global popularity measurement scale allowing users to compare their social influence with others.

The main goal of this Diploma Thesis is the development of an application called Analyze-me. This application uses the KLOUT API and Twitter API to obtain user data from various social networks and present detailed results and a general score, which constitute the user's social ranking. In addition, the Analyze-me application is to provide an estimation of the user's future score. Specifically, the purpose of this Thesis is to research predictive algorithms and compare them on real data to decide which one performs the best. Then develop a web based application that uses the KLOUT and Twitter APIs to gain information and score analysis of a user, and present this data to the user within a graphical interface and at the same time store the results in a database. Furthermore, this data is supplied to the predictive algorithm to provide an estimation of the user's future scores. The LinkedIn API is also used only to authenticate the user before he can input the required data for the LinkedIn analysis.

The application will consist of a website ([www.analyze-me.com](http://www.analyze-me.com)) written in PHP language playing the role of the server, and a database keeping track of every user's scores and statistics over time. The database will be managed from a phpMySQL client. Also, a mobile application will be developed for Android operating system offering exactly what the website offers, actually being a mobile client of the analyze-me service. Both the website and mobile will share the same database and data.

The main feature that differentiates this application from the existing KLOUT service will be the ability to estimate the user's score in the future using time-series predictive algorithms.

The structure of this Diploma Thesis is as follows:

In Chapter 2 a review of some time-series prediction algorithms is given along with typical examples coded in MATLAB programming and solved for the purpose of providing the necessary knowledge and understanding for selecting the most appropriate one to be implemented and provide the Analyze-me application with predicting capabilities.

Chapter 3 presents a short review of the social networking, general information about Twitter, Twitter API, LinkedIn, LinkedIn API, and KLOUT, and gives some related work.

Chapter 4 describes the architecture of the Analyze-me application.

Chapter 5 presents the development and implementation of the application's interface, database, and prediction interface, and describes in details the implementation of Twitter analysis and the LinkedIn analysis.

Chapter 6 presents a real Analyze-me data set and on it performs a least-squares regression-polynomial curve fitting, a Box-Jenkins methodology, and an autoregressive identification with Kalman filter, along with a comparative analysis of these prediction methods for the selection of the most appropriate prediction algorithm.

Chapter 7 addresses the use of the Analyze-me application by giving detailed descriptions of the step-by-step user interface to the Analyze-me.com website, what does every option in the menu do, and all the possible sub-pages the user may visit.

In Chapter 8 the development, implementation, and usage of the mobile version of the Analyze-me application is presented.

The last Chapter 9 contains the final thoughts and conclusions of this Thesis as well as future plans for further development.

Finally, references are presented and at the end of the Thesis the source code files for the Analyze-me application are presented as an appendix. It is noticed that the source code of both the Website application and mobile Android application was developed entirely for the purpose of this Thesis. The only third party code that was used is Oauth library provided by <http://oauth.net/> and was used to complete the authentication process with LinkedIn.

## 2. Review of Time-Series Prediction Algorithms

Prediction is the estimation by a mathematical model of future values given a time-series of sample data of a variable. The terms forecast, prediction, projection, and prognosis typically are used interchangeably. Mathematical models for predictions and time-series analysis of sample data include concepts and techniques of curve fitting, regression, correlation, state-space models and Neural Networks, to name a few.

The objective of this chapter is first to review the basics of the above mentioned predictive and time-series analysis techniques and then, as a proof of concept, to apply appropriate MATLAB commands and/or code to some typical examples. These typical examples will provide the necessary knowledge and understanding in order to apply later some predictive algorithms to real time-series data, compare their performance, choose the best algorithm, and finally implement it in the Analyze-me application.

### 2.1 Curve Fitting, Least Squares Method, Regression

#### 2.1.1 Curve Fitting

Curve fitting [1, 4, 13] is the general problem of finding a mathematical equation  $y = g(x)$  between the independent (or predictor) variable  $x$  and the dependent (or predicted) variable  $y$  that best fits a set of data points  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ . The function  $g(\cdot)$  can take many forms. In practice the use of a scatter diagram  $y$  vs.  $x$  (or transformed variable such as  $\log y$  vs. transformed variable  $\log x$ ) often suggests the type of curve, e.g., a straight line  $y = a + bx$ , a parabolic or quadratic curve  $y = a + bx + cx^2$ , etc..

#### 2.1.2 Regression

The term regression [1, 4, 7, 14] refers to estimating the dependent variable  $y$  with the estimate  $\hat{y}$  from the independent variable  $x$  from a set of data. If this estimate  $\hat{y}$  from  $x$  is by means of some equation, i.e.,  $\hat{y} = \hat{g}(x)$ , this equation is called a regression equation of  $\hat{y}$  on  $x$  and the corresponding curve a regression curve of  $\hat{y}$  on  $x$ . For the linear or straight line relation the relation  $\hat{y} = a + bx$  is called a regression line of  $\hat{y}$  on  $x$ .

#### 2.1.3 Least Squares Method

Since more than one curve of a given type can fit a set of data, for the sake of avoiding individual judgment in constructing these curves, it is necessary to agree on a definition of a best curve. For each given value  $x_1, x_2, \dots, x_n$  of the data set there will be a corresponding error  $e_1, e_2, \dots, e_n$  between the values  $y_1, y_2, \dots, y_n$  from the data set and those values  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$  computed by the equation type  $\hat{y} = \hat{g}(x)$ . A measure of the fit of the curve  $\hat{y} = \hat{g}(x)$  to the data set is provided by the sum-square-error (SSE) quantity  $e_1^2 + e_2^2 + \dots + e_n^2$ . If this SSE is small, the fit is good; if it is large, the fit is bad. Therefore, of all curves in a given family of curves approximating a set of  $n$  data points, a curve having the SSE property that is minimum is called a best-fitting curve in the family. A curve having this property is said to fit the data in the *least-squares sense* and is called a *least-squares regression curve*, or simply a *least-squares curve* [1, 4, 7, 12, 14]. A

line having this property is called a *least-squares line*; a parabola that has this property is called a *least-squares parabola*; etc.

### 2.1.4 The Least-Squares Line

The least-squares line which approximates a set of measurements  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  has the model equation:

$$\hat{y} = a + bx \quad (1)$$

where the constants  $a$  and  $b$  are determined by minimizing the SSE (sum-square-error) between the measurements and the model (or predicted values) [1, 4, 7, 12, 14]:

$$\mathcal{E} = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n [y_i - a - bx_i]^2 \quad (2)$$

Using standard techniques from calculus, the minimization of (2) is achieved by taking the derivatives of  $\mathcal{E}$  with respect to  $a$  and  $b$  and setting them to zero:

$$\frac{\partial \mathcal{E}}{\partial a} = 2na + 2b \sum_{i=1}^n x_i - 2 \sum_{i=1}^n y_i$$

which gives:

$$\boxed{\sum_{i=1}^n y_i = an + b \sum_{i=1}^n x_i} \quad (3)$$

$$\frac{\partial \mathcal{E}}{\partial b} = 2b \sum_{i=1}^n x_i^2 + 2a \sum_{i=1}^n x_i - 2 \sum_{i=1}^n x_i y_i$$

which gives:

$$\boxed{\sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2} \quad (4)$$

Solving simultaneously equations (3) and (4), after some algebra, gives:

$$a = \frac{1}{n} \sum_{i=1}^n y_i - \frac{1}{n} b \sum_{i=1}^n x_i = M_y - bM_x \quad (5)$$

(where  $M_y$  and  $M_x$  denote the means of  $y$  and  $x$ , respectively) and

$$b = \frac{\sum_{i=1}^n (y_i - M_y)(x_i - M_x)}{\sum_{i=1}^n (x_i - M_x)^2} \quad (6)$$

Substituting (5) into (1) we have:

$$\begin{aligned}\hat{y} &= a + bx = M_y - bM_x + bx \Rightarrow \\ \hat{y} - M_y &= b(x - M_x)\end{aligned}\quad (7)$$

And upon using (6) from (7) we get:

$$\hat{y} - M_y = \frac{\sum_{i=1}^n (y_i - M_y)(x_i - M_x)}{\sum_{i=1}^n (x_i - M_x)^2} (x - M_x) \quad (8)$$

The result (8) shows that the constant  $b$ , which is the *slope* of the line (1), is the fundamental constant in determining the line. From (6) it is also seen that the least-squares line passes through the point  $M_x, M_y$ , which is called the *centroid* or *center of gravity* of the data.

### 2.1.5 The Least-Squares Regression Line in Terms of Sample Variances and Covariance

The sample variances and covariance of  $x$  and  $y$  are given by [1, 4, 7, 12, 14]:

$$s_x^2 = \frac{\sum_{i=1}^n (x_i - M_x)^2}{n}, \quad s_y^2 = \frac{\sum_{i=1}^n (y_i - M_y)^2}{n}, \quad s_{xy} = \frac{\sum_{i=1}^n (y_i - M_y)(x_i - M_x)}{n} \quad (9)$$

Using these, the least-squares regression line of  $\hat{y}$  on  $x$  and similarly  $\hat{x}$  on  $y$  can be written as:

$$\hat{y} - M_y = \frac{s_{xy}}{s_x^2} (x - M_x) \quad \text{and} \quad \hat{x} - M_x = \frac{s_{xy}}{s_y^2} (y - M_y) \quad (10)$$

and if we formally define the sample correlation coefficient by:

$$r = \frac{s_{xy}}{s_x s_y} \quad (11)$$

then (10) becomes:

$$\frac{\hat{y} - M_y}{s_y} = r \frac{x - M_x}{s_x} \quad \text{and} \quad \frac{x - M_x}{s_x} = r \frac{\hat{y} - M_y}{s_y} \quad (12)$$

Using the fact that  $(\hat{x} - M_x)/s_x$  and  $(\hat{y} - M_y)/s_y$  are standardized sample values or standard scores, the results in (11) provide a simple way of remembering the regression lines. It is clear that the two lines in (12) are different unless  $r = \pm 1$ , in which case all sample points lie in a line and there is *perfect linear correlation and regression*.

It is also interesting to note that if the two regression lines (12) are written as  $\hat{y} = ax + b$ ,  $\hat{x} = c + dy$ , respectively, then:

$$bd = r^2 \quad (13)$$



### 2.1.6 MATLAB Least-Squares Regression - Polynomial Curve Fitting

The MATLAB command *polyfit* [2, 5] finds the polynomial coefficients that fit a set of data in a least-square sense. Also the MATLAB command *polyval* evaluates the polynomial at different values of the independent variable  $x$ . For a detailed description of the Matlab commands or functions for curve fitting see Matlab help. The syntax of the Matlab function *polyfit* is:

```
p = polyfit(x, y, n)
```

where  $x$  and  $y$  are vectors containing the  $x$  and  $y$  data to be fitted, and  $n$  is the degree of the polynomial to return. The result  $p$  is a row vector of length  $n+1$  containing the polynomial coefficients in descending powers:  $p(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$ .

The syntax of the Matlab function *polyval* is:

```
y = polyval(p, x)
```

where  $p$  is the a row vector of length  $n+1$  containing the polynomial coefficients in descending powers and  $x$  is a row vector containing the data set values of the independent variable  $x$ .

Problem: Consider the following  $x$ - $y$  data set:

x	1	2	3	4	5
y	5.5	43.1	128	290.7	498.4

Find different least-squares regression curves which model the data and evaluate the results.

Solution: The procedure in finding least-squares regression curves for a given data set starts with a scatter plot of the dependent variable  $y$  vs. the independent variable  $x$ . The scatter plot suggests the type of curve to be fitted. The following MATLAB code finds a straight line to fit the data set and also a 3<sup>rd</sup> degree polynomial curve. In addition, the code provides plots from which the results of the straight line and the 3<sup>rd</sup> degree polynomial curves can be evaluated and assessed. Appropriate comments for the MATLAB commands are also provided next to the code lines indicated with the % symbol.

```
clc; clear all; close all; % clear command window, workspace & figures
x = [1 2 3 4 5]; % values of data set independent variable x
y = [5.5 43.1 128 290.7 498.4]; % values of data set dependent variable y
figure(1)
plot(x, y, 'o') % scatter plot of data set variables x vs. y
axis([0 6 min(y)-10 max(y)+10]); % range of plotting axes
title('Scatter plot of data set')
% grid on % insert grids in the plot
% -----
p1 = polyfit(x, y, 1) % p1 coefficients of straight line fit y = a + bx
x1 = 1:1:5; % Define a uniformly spaced vector x1
y1 = polyval(p1, x1); % Evaluate the polynomial p1 at x1 values
figure(2)
plot(x, y, 'o', x1, y1) % Plot the fit y = a + bx on top of data
title('Plot of straight line y = a + bx on top of data set')
```

```

axis([0 6 min(y1)-10 max(y1)+10]);
% grid on
y1 = polyval(p1, x); % Evaluate model at the data x
res1 = y - y1; % residuals by differencing data and model y values
figure (3)
stem(x, res1) % Plot the residuals vs. x values
title('Stem plot of straight line y errors vs. x values')
axis([0 6 min(res1)-5 max(res1)+5]);
% -----
p2 = polyfit(x, y, 3) % p2 coefficients of 3rd degree polynomial fit
x2 = 1: 1:5;
y2 = polyval(p2, x2); % Evaluate the 3rd degree polynomial at x2
figure(4)
plot(x, y, 'o', x2, y2) % Plot 3rd degree polynomial fit on top data
title('Plot of 3rd degree polynomial on top of data set')
axis([0 6 min(y2)-10 max(y2)+10]);
y2 = polyval(p2, x); % Evaluate model at the data x
res2 = y - y2; % residuals by differencing data and model y values
figure (5)
stem(x, res2) % Plot the residuals vs. x values
title('Stem plot of 3rd degree polynomial y errors vs. x values')
axis([0 6 min(res2)-1 max(res2)+1]);

```

Executing the above MATLAB code, we obtain Figure 1 below which provides a scatter plot of the data set points  $(x_i, y_i)$ . From the figure it can be assessed that it is possible to find a straight line curve which can fit the data set points with some degree of accuracy.

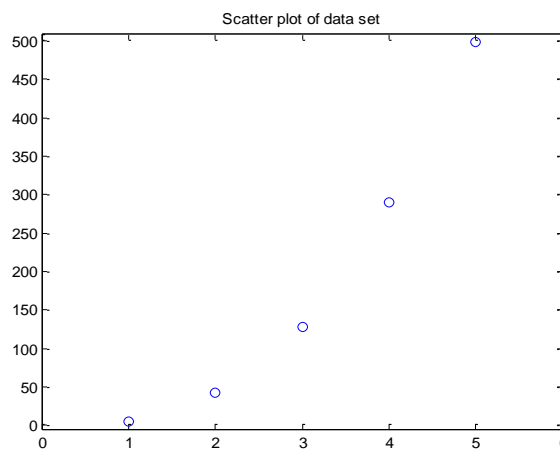


Figure 1: Scatter plot data set points  $(x_i, y_i)$

The 1st degree polynomial (straight line) curve which fits the data set points is:

$$p_1(x) = 123.3400x - 176.8800$$

A plot of the above straight line curve  $p_1(x)$  for different values of  $x$  with the given data set points on the top of the plot is given in Figure 2 below.

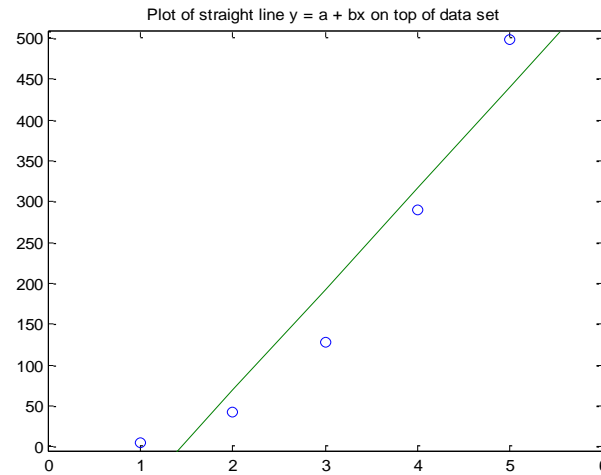


Figure 2: Plot of straight line  $y = a + bx$  on top of data set points  $(x_i, y_i)$

From Figure 2 it can be seen that the straight line curve splits the data set points into two clusters; the first and last data points in a cluster above the straight line and the other three points in a cluster below the straight line. In addition, the distance (error) of the data points from the straight line is not negligible.

Figure 3 below gives a stem plot of the distance (errors) between the data set points and the points from the  $y = a + bx$  straight line model.

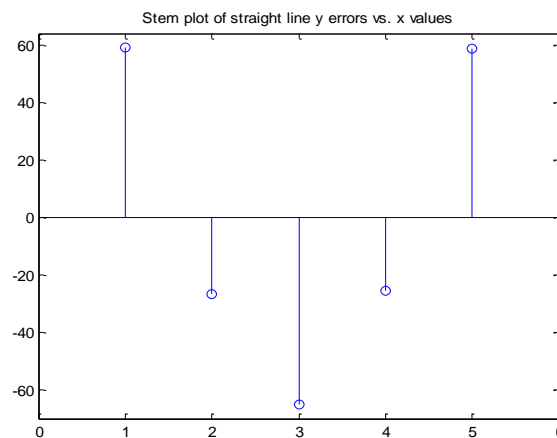


Figure 3: Stem plot of straight line  $y = a + bx$  errors vs. data set  $x_i$  values

From Figure 3 it can be assessed that the errors for the first and last data points which are above the straight line have large values. Also the middle data set point which is below the straight line is large too. These relatively large errors along with their separation into clusters above and below the straight line fit indicate that a better polynomial type model can be found to fit the data. This polynomial type model can be of higher degree such as 3<sup>rd</sup> degree.

From the MATLAB code the 3rd degree polynomial which fits the data points then is:

$$p_2(x) = -0.1917x^3 + 31.5821x^2 - 60.3262x + 35.3400$$

A plot of the above 3<sup>rd</sup> degree polynomial  $p_2(x)$  for different values of  $x$  with the given data set points on the top of the plot is given in Figure 4 below.

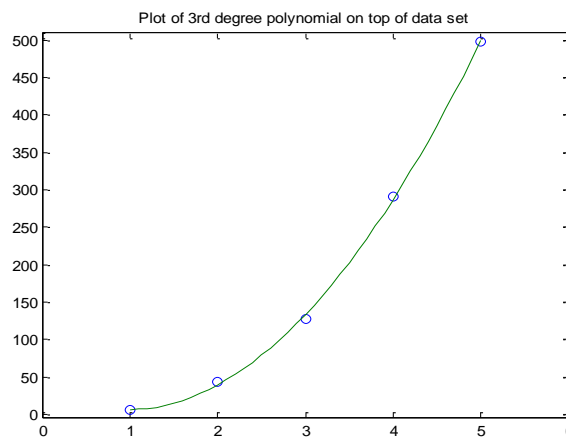


Figure 4: Plot of 3rd degree polynomial on top of data set points  $(x_i, y_i)$

From Figure 4 it can be seen that the 3<sup>rd</sup> degree polynomial curve does a better job of fitting than the straight line. In addition, the distance (error) of the data points from the 3<sup>rd</sup> degree polynomial curve is not that large (in the range of -5.5 to +4).

Figure 5 below gives a stem plot of the distance (errors) between the data set points and the points from the 3<sup>rd</sup> degree polynomial curve model.

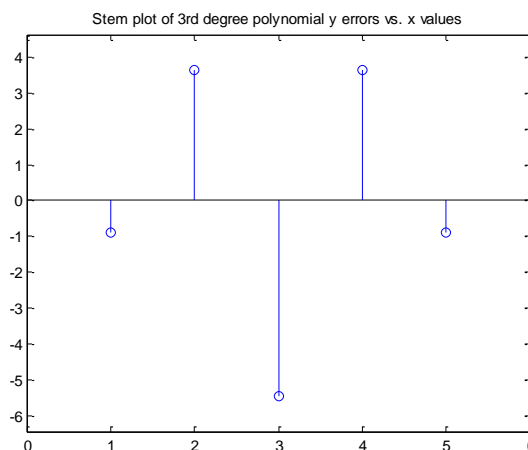


Figure 5: Stem plot of 3rd degree polynomial curve errors vs. data set  $x_i$  values

From Figure 5, since the errors are within the range  $(-5, 4)$ , it can be concluded that the 3<sup>rd</sup> degree polynomial is an appropriate curve to fit the given data set.

The found 3<sup>rd</sup> degree polynomial  $p_2(x) = -0.1917x^3 + 31.5821x^2 - 60.3262x + 35.3400$  can be used as a predictor to forecast future values of  $y$  for corresponding future values of  $x$ . As an example, for  $x=6$ , using the command `polyval(p2, 6)` the answer is  $y = p_2(x=6) = 768.9400$ .

## 2.2 Basics of Time-Series Data Analysis

The aims of time-series analysis [3, 4, 7, 9, 10, 11, 13] are to describe the data using summary statistics and/or graphical methods, fit low-dimensional statistical models or processes generating the data, and make forecasts by estimating future values or predictions of the series using the fitted low-dimensional statistical models.

### 2.2.1 Time-Series - Data Description

The first step in any time-series analysis is to plot the data against time [3, 6]. This *time plot* gives important features of the data such as trend, seasonality, outliers, smooth changes in structure, turning points and/or sudden discontinuities. These features are vital in describing the data, in helping to formulate a sensible model and in choosing an appropriate forecasting method. A second useful graphical tool is the *scatter plot*, which explores the relationship between two variables. Yet a third and very important diagnostic graph is the *correlogram* which plots the *autocorrelation function* vs. different time lags of the stationary data. This whole approach is also called *Initial Data Analysis* (or IDA).

### 2.2.2 Time-Series - Modeling Stationary stochastic processes

A time series  $x_1, x_2, \dots, x_n$  is a set  $x_t$  of a time-ordered sequence of discrete values (or observations) of a variable  $x$  made at equally spaced time intervals  $\Delta t$  for  $t = 1, 2, 3, \dots, n$ . An alternate, more precise notation, which accounts for observations not equally spaced through time, is to use  $x_t, t \in T$ , where  $T$  denotes the index set of times. In the sequel the equally spaced case will be used.

Given an observed time series data  $x_1, x_2, \dots, x_n$ , a *forecasting method* is a procedure or an algorithmic rule for computing future values  $x_{n+h}$  from present and past values for the lead time or forecasting horizon  $h$  based on a model generating the time series data [3]. If future values can be predicted exactly from past values, then a series is said to be *deterministic*. However, most time series are *stochastic*, or *random*, in that the future is only partly determined by past values.

If an appropriate mathematical model for this random behavior can be found, then the model should enable good forecasts to be computed. A model for a stochastic time series is called a *stochastic process*, i.e., a family of discrete random variables indexed by time, denoted by  $X_1, X_2, \dots, X_n$  or more generally by  $X_t$ . The process then can be viewed as potentially generating an infinite number of time series. The observed series  $x_t$  is just one possible realization. The value of the series  $x_t$  at any time  $t = i$  can be considered as a specific realization of a random variable  $X_i$ , with a probability density function  $p(x_i)$ . So any set of  $X_i$  at different times, say  $X_{i1}, X_{i2}, \dots, X_{ir}$  for  $t = i$  with  $i = 1, 2, 3, \dots, r$ , has a joint probability density function. If the joint probability density function is independent of time, the process is said to be *strictly stationary*.

Many statistical procedures assume at least *weak stationarity*, which means that the mean, variance, and autocovariance function are independent of time. The mean of the series is the first moment  $E X_t$ , while the general second moment is the covariance between  $X_t$  and  $X_{t+k}$  for different values of  $t$  and  $k$ . (This type of covariance on the same random variable is called autocovariance). The variance,  $Var X_t$ , is a special case of the autocovariance when the lag  $k$  is zero. Thus a stochastic process (model) is second-order (weakly) stationary if:

$$E X_t = \mu \text{ is a finite constant, for all } t \quad (14)$$

$$\text{Var } X_t = \sigma^2 \text{ is a finite constant, for all } t \quad (15)$$

and more generally, if the autocovariance function depends only on the lag,  $k$ , so that:

$$\text{Cov } X_t, X_{t+k} = E (X_t - \mu)(X_{t+k} - \mu) = \gamma_k, \text{ for all } t \quad (16)$$

The set of autocovariance coefficients  $\gamma_k$ , for  $k = 0, 1, 2, 3, \dots$ , constitute the *autocovariance function* (abbreviated ACVF) of the process. Note that  $\gamma_0$  equals the variance,  $\sigma^2$ . Second-order stationarity is sometimes called *covariance* or *weak* stationarity.

The ACVF is often standardized to give a set of *autocorrelation coefficients*,  $\rho_k$ , given by:

$$\rho_k = \gamma_k / \gamma_0, \text{ for } k = 0, 1, 2, 3, \dots \quad (17)$$

The set of *autocorrelation coefficients*,  $\rho_k$ , constitute the *autocorrelation function* (abbreviated ACF). For stationary processes,  $\rho_k$  measures the correlation at lag  $k$  between  $X_t$  and  $X_{t+k}$ . The ACF is an *even* function of lag, since  $\rho_k = \rho_{-k}$ , and has the usual property of correlation that  $|\rho_k| \leq 1$ . Some additional useful functions, which are complementary to the ACF, include the *partial autocorrelation function* (abbreviated partial PACF) which essentially measures the excess correlation at lag  $k$  which has not already been accounted for by autocorrelations at lower lags.

A stochastic process is said to be a *Gaussian* (or normal) process if the joint distribution of any set of  $X_t$ 's is multivariate normal. Such a process is completely characterized by its first and second moments but it is advisable to remember that this is not so for non-Gaussian processes and that it is possible to find Gaussian and non-Gaussian processes with the same ACF. This creates obvious difficulties in interpreting sample ACFs when trying to identify a suitable underlying model. For non-Gaussian processes it may also be necessary to consider *strict* rather than second-order stationarity, wherein the joint distribution of any set of random variables is not changed by shifting them all by the same time  $\tau$ , for any value of  $\tau$ . It is noted that although strict stationarity sounds like (and is) a strong condition, it does not imply second-order stationarity without the additional assumption that the first and second moments are finite.

### 2.2.3 The correlogram

The correlogram [3] is probably the most useful tool in time-series analysis after the time plot. It can be used at two different levels of sophistication, either as a relatively simple descriptive tool or as part of a more general procedure for identifying an appropriate model for a given time series.

Denote the observed time series data by  $x_1, x_2, \dots, x_N$ . Then the sample autocovariance coefficient at lag  $k$  is usually calculated by:

$$c_k = \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x}), \text{ for } k = 1, 2, \dots \quad (18)$$

where  $\bar{x}$  the sample mean,  $\bar{x} = \sum_{t=1}^N x_t / N$ , and the sample autocorrelation coefficient at lag  $k$  is then calculated by:

$$r_k = c_k / c_0 \quad (19)$$

The graph of  $r_k$  against  $k$  is called the *sample autocorrelation function* (abbreviated ACF) or the *correlogram*. It is an important tool in assessing the behavior and properties of a time series. It is typically plotted for the original series and also after differencing or transforming the data as necessary to make the series look stationary and approximately normally distributed.

For data from a stationary process, it can be shown that the correlogram generally provides an estimate of the theoretical ACF defined in (17). Although intuitively ‘obvious’, this is mathematically hard to prove because it requires that averages over time for an observed time series (like  $\bar{x}$ ) enable us to estimate the ensemble properties of the underlying process (like  $E[X_t]$ ); in other words, that we can estimate the properties of the random variable at time  $t$  with the help of observations made at other times. (*Strictly speaking, the process needs to have appropriate ‘ergodic’ properties so that averages over time from a single realization provide estimates of the properties of the underlying process. For example a covariance-stationary process is said to be ‘ergodic in the mean’ if the sample mean,  $\bar{x} = \sum_{t=1}^N x_t / N$ , converges in probability to  $E[X_t]$  as  $N \rightarrow \infty$  so that  $\bar{x}$  provides a consistent estimate of the ensemble average. A sufficient condition is that  $\rho_k \rightarrow 0$  as  $k \rightarrow \infty$ . We will assume appropriate ergodic properties are satisfied throughout*).

It follows that, for data from a non-stationary process, the correlogram does not provide an estimate of anything! In that case, the values in the correlogram will typically not come down to zero except at high lags, and the only merit of the correlogram is to indicate that the series is not stationary.

Interpreting a correlogram is one of the hardest tasks in time-series analysis, especially when  $N$  is less than about 100 so that the sample autocorrelations have relatively large variance. For a stationary series, the pattern of the correlogram may suggest a stationary model with an ACF of similar shape. The simplest case is that of a purely random process (describe later), where it can be shown that  $r_k$  is asymptotically normally distributed with mean  $-1/N$ , and standard deviation  $1/\sqrt{N}$  for  $k \neq 0$ . As the mean,  $-1/N$ , is small compared with the standard deviation, it is customary to take the mean as being approximately zero and regard values outside the range  $0 \pm 2/\sqrt{N}$  as being significantly different from zero. Several significant coefficients, especially at important low lags, provide strong evidence that the data do *not* come from a purely random process.

One common pattern for stationary time series is to see *short-term correlation* typified by finding perhaps the first three or four values of  $r_k$  to be significantly different from zero. If they seem to decrease in an approximately exponential way then a first order autoregressive [AR(1)] model is indicated. If they behave in a more complicated way, then a higher-order AR model may

be appropriate. If the only significant autocorrelation is at lag one, then a  $MA(1)$  model is indicated.

For seasonal series, there is likely to be a large positive value of  $r_k$  at the seasonal period, and this may still be present to a (much) lesser extent even after the seasonal effect has supposedly been removed. Thus the correlogram is often used to see if seasonality is present.

For series with a trend, the correlogram will not come down to zero until a high lag has been reached, perhaps up towards half the length of the series. The correlogram provides little information in the presence of trend other than as an indicator that some form of trend-removal is necessary to make the series stationary.

#### 2.2.4 Some classes of univariate time-series linear models

Many forecasting procedures are based on a time-series *model* [3]. It is therefore helpful to be familiar with a range of time-series models before starting to look at forecasting methods. This section aims to give an introductory flavor of some simple important univariate models. A *univariate* model describes the distribution of a single random variable at time  $t$ , namely  $X_t$ , in terms of its relationship with past values of  $X_t$  and its relationship with a series of white-noise random shocks as defined next.

#### 2.2.5 The purely random process or white noise

The simplest type of model, used as a ‘building brick’ in many other models, is the *purely random process* [3]. This process may be defined as a sequence of uncorrelated, identically distributed random variables with zero mean and constant variance. This process is clearly stationary and has ACF given by:

$$\rho_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (20)$$

This process is also called (uncorrelated) *white noise*, the *innovations* process or (loosely) the ‘error’ process. The model is rarely used to describe data directly, but is often used to model the random disturbances in a more complicated process. If this is done, the white-noise assumptions need to be checked.

Often the term ‘purely random process’ is used to refer to a sequence of *independent*, rather than uncorrelated, random variables. There is no difference in regard to second-order properties and of course independence implies lack of correlation. Moreover, the converse is true when the process is a *Gaussian* process (as often assumed to be). However, for non-linear models the difference between uncorrelated white noise and independent white noise may be crucial, and the stronger assumption of independence is generally needed when looking at non-linear models and predictors.

By convention, the notation  $Z_t$  is used to denote a purely random process with zero mean and variance  $\sigma_z^2$  when it is a component of a random walk model, of an autoregressive model or of the more general class of (Box-Jenkins) ARIMA (Autoregressive Integrated Moving Average) processes.



### 2.2.6 The random walk

A model of much practical interest is the *random walk* [3] which is given by:

$$X_t = X_{t-1} + Z_t \quad (21)$$

where  $Z_t$  denotes a purely random process. This model may be used, at least as a first approximation, for many time series arising in economics, finance and elsewhere.

The series of random variables generated with random walk does *not* form a stationary process as it is easy to show that the variance increases through time. However, the first differences of the series, namely  $(X_t - X_{t-1}) = Z_t$ , do form a stationary series. Taking differences is a common procedure for transforming a non-stationary series into a stationary one.

### 2.2.7 Autoregressive processes

A process  $X_t$  is said to be an *autoregressive* process [3] of order  $p$  (abbreviated  $AR(p)$ ) if:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t \quad (22)$$

Thus the value at time  $t$  depends linearly on the last  $p$  values and the model looks like a regression model – hence the term *autoregression*. The simplest example of an AR process is the first-order case, denoted  $AR(1)$ , given by:

$$X_t = \phi X_{t-1} + Z_t \quad (23)$$

Clearly, if  $\phi = 1$ , then the model reduces to a random walk (21), and then the model is non-stationary. If  $|\phi| > 1$ , then it is intuitively obvious that the series will be explosive and hence non-stationary. However, if  $|\phi| < 1$ , then it can be shown that the process is stationary, with ACF given by  $\rho_k = \phi^k$  for  $k = 0, 1, 2, 3, \dots$ . Thus the ACF decreases exponentially.

### 2.2.8 Moving average processes

A process  $X_t$  is said to be a moving average process of order  $q$  [3] (abbreviated  $MA(q)$ ) if:

$$X_t = Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q} \quad (24)$$

Thus the value at time  $t$  is a sort of moving average of the (unobservable) random shocks,  $Z_t$ . However, the ‘weights’,  $\theta_j$ , involved in the moving average will generally not add to unity and so the phrase ‘moving average’ is arguably unhelpful.

The simplest example of an MA process is the first-order case, denoted  $MA(1)$ , given by:

$$X_t = Z_t + \theta Z_{t-1} \quad (25)$$

It can be shown that this process is stationary for all values of  $\theta$  with an ACF given by:

$$\rho_k = \begin{cases} 1 & k = 0 \\ \theta / (1 + \theta^2) & k = 1 \\ 0 & k > 1 \end{cases} \quad (26)$$

Thus the ACF ‘cuts off’ at lag 1.

### 2.2.9 The random walk plus noise model

The *random walk plus noise* model [3], sometimes called the *local level*, or *steady* model is a simple example of a class of models called *state-space* models (to be considered in more detail latter). Suppose the observed random variable at time  $t$  may be written in the form:

$$X_t = \mu_t + n_t \quad (27)$$

where the local level,  $\mu_t$ , changes through time like a random walk so that:

$$\mu_t = \mu_{t-1} + w_t \quad (28)$$

The two sources of random variation in the above equations, namely  $n_t$  and  $w_t$ , are assumed to be independent white noise processes with zero means and respective variances  $\sigma_n^2$ ,  $\sigma_w^2$ . The properties of this model depend on the ratio of the two error variances, namely  $\sigma_w^2 / \sigma_n^2$ , which is called the *signal-to-noise ratio*. In the notation of state-space modeling, the unobserved variable,  $\mu_t$ , which denotes the local level at time  $t$ , is called a *state variable*, and the *random walk plus noise* equation  $X_t = \mu_t + n_t$  is called the *observation* or *measurement* equation, while the  $\mu_t = \mu_{t-1} + w_t$  equation is called the *transition* equation.

### 2.2.10 The ARMA processes

The autoregressive moving average process [3],  $ARMA(p, q)$ , is a combination of an  $AR(p)$  and  $MA(q)$  as follows:

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \dots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q} \quad (29)$$

Using the backward shift operator  $B$ , such that  $BX_t = X_{t-1}$ , then by substitution the  $ARMA(p, q)$  model can be written as:

$$\phi(B)X_t = \theta(B)Z_t \quad (30)$$

where  $\phi(B)$ ,  $\theta(B)$  are polynomials in  $B$  of finite order  $p$ ,  $q$ , respectively, i.e.,

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (31)$$

and

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q \quad (32)$$

From the above, as  $B$  is an operator, the properties of the ARMA model (29) depend upon the algebraic properties  $\phi(\cdot)$  and  $\theta(\cdot)$  which are investigated by examining the properties of  $\phi(x)$  and  $\theta(x)$ , say, where  $x$  denotes a complex variable. The ARMA model has a unique causal (not depending upon future values) stationary solution provided that the roots of  $\phi(x)=0$  lie outside the unit circle. The process is invertible (can uniquely determine its input from its output) provided that the roots of  $\theta(x)=0$  lie outside the unit circle.

In addition, a stationary ARMA process may generally be rewritten as an MA process of possibly infinite order (the Wold representation). This can be seen from the expression  $\phi(B)X_t = \theta(B)Z_t$ , which can be rewritten as  $X_t = \theta(B)/\phi(B) Z_t$ . Expanding  $\theta(B)/\phi(B) = \psi(B)$  as a power series in  $B$  we have:

$$\psi(B) = \psi_0 + \psi_1 B + \psi_2 B^2 + \dots \quad (33)$$

Then, it follows:

$$X_t = \theta(B)/\phi(B) Z_t = \psi(B)Z_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j} \quad (34)$$

It can be shown that the process is stationary if  $\psi(x)$  converges on, and within, the unit circle. In the stationary case, the ACF will generally be a mixture of damped exponentials or sinusoids.

The importance of ARMA processes is that many real data sets may be approximated in a more parsimonious way (meaning fewer parameters are needed) by a mixed ARMA model rather than by a pure AR or pure MA process.

### 2.2.11 ARIMA processes

In practice many (or most) time series are non-stationary and so we cannot apply stationary AR, MA or ARMA processes directly. One possible way of handling non-stationary series is to apply *differencing* so as to make them stationary. The first differences, namely  $(X_t - X_{t-1}) = (1-B)X_t$ , may themselves be differenced to give second differences, and so on. The  $d$ th differences may be written as  $(1-B)^d X_t$ . If the original data series is differenced  $d$  times before fitting an  $ARMA(p, q)$  process, then the model for the original undifferenced series is said to be an  $ARIMA(p, d, q)$  process where the letter 'I' in the acronym stands for *integrated* and  $d$  denotes the number of differences taken.

Mathematically, the autoregressive integrated moving average process,  $ARIMA(p, d, q)$ , then can be written as [3]:

$$\phi(B)(1-B)^d X_t = \theta(B)Z_t \quad (35)$$

The combined AR operator is now  $\phi(B)(1-B)^d$ . Replacing the operator  $B$  with a variable  $x$ , it can be seen that the function  $\phi(x)(1-x)^d$  has  $d$  roots on the circle (as  $(1-x)=0$  when  $x=1$ ) indicating that the process is non-stationary – which is why differencing is needed, of course! Note that when  $\phi(B)$  and  $\theta(B)$  are both just equal to unity (so that  $p$  and  $q$  are both 0) and

$d$  equals 1, then the model reduces to an  $ARIMA(0,1,0)$  model, given by  $X_t - X_{t-1} = Z_t$ . This is the same as the random walk model which can be regarded as an  $ARIMA(0,1,0)$  model.

**2.2.12 SARIMA processes**

If the series is *seasonal*, with  $s$  time periods per year, then a seasonal ARIMA (abbreviated SARIMA) model may be obtained as a generalization of the ARIMA model. Let  $B^s$  denote the operator such that  $B^s X_t = X_{t-s}$ . Thus seasonal differencing may be written as  $(X_t - X_{t-s}) = (1 - B^s)X_t$ . A seasonal autoregressive term, for example, is one where  $X_t$  depends linearly on  $X_{t-s}$ .

A SARIMA model with non-seasonal terms of order  $(p, d, q)$  and seasonal terms of order  $(P, D, Q)$  is abbreviated a  $SARIMA(p, d, q) \times (P, D, Q)_s$  model and may be written as [3]:

$$\phi(B)\Phi(B^s)(1 - B)^d(1 - B^s)^D X_t = \theta(B)\Theta(B^s)Z_t \tag{36}$$

where  $\Phi, \Theta$  denote polynomials in  $B^s$  of order  $P, Q$ , respectively.

One model, which is particularly useful for seasonal data, is the SARIMA model of order  $(0,1,1) \times (0,1,1)_s$ . For monthly data, with  $s = 12$ , the latter may be written as:

$$(1 - B)(1 - B^{12})X_t = (1 + \theta B)(1 + \Theta B^{12})Z_t \tag{37}$$

In summary, given an observed time series data  $x_1, x_2, \dots, x_N$ , after removing possible trend and seasonality, the following table provides guidelines for model identification from the behavior of the sample ACF and PACF as a function of lags of the stationary data.

<b>Behavior of ACF and PACF for Causal and Invertible Stationary ARMA Models</b>			
	$AR(p)$	$MA(q)$	$ARMA(p, q)$
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

**2.2.13 Model Estimation**

Once the order of a particular ARMA model for a given time series has been identified, the next step is to estimate the unknown parameters  $(\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q, \sigma_z^2)$  of the model based on the observed data. The main approaches are least squares and maximum (Gaussian) likelihood estimation which is usually performed by high quality software programs. The maximum (Gaussian) likelihood estimation is non-linear in the sense that the function to be maximized is not a quadratic function of the unknown parameters, so the estimators cannot be found by solving a system of linear equations. They are found instead by searching numerically for the maximum of the likelihood surface. The algorithm requires the specification of initial

parameter values with which to begin the search. The closer the preliminary estimates are to the maximum likelihood estimates, the faster the search will generally be. To provide these initial values, a number of preliminary estimation algorithms are available. For pure autoregressive models the choice is between Yule–Walker and Burg estimation, while for models with  $q > 0$  it is between the innovations and Hannan–Rissanen algorithms [7].

#### 2.2.14 Diagnostic Model Checking

Once the ARIMA model is specified and the parameters estimated, the adequacy of the model should be checked [3, 7]. One way is to use the model to forecast one-step-ahead all the known values of the time series. Compute the residuals between the known and forecasted values. A large individual residual may indicate an outlying observation, which may need to be looked at specially and perhaps adjusted in some way. The autocorrelation function of the residual series provides an overall check on whether a good model has been fitted or whether there is still some structure left to explain. The residual autocorrelations may be examined individually, to see if any exceed the value  $2/\sqrt{N}$  in absolute magnitude. If within this value, the residuals are stationary and the model can be judged to be adequate. If the model is not validated in the diagnostic checking stage, a more appropriate model by going back to the identification step and trying a better model.

#### 2.2.15 Model Forecasting

Once a model has been fitted to the data, then the fitted model can be used to predict future values of the time series. Given observations  $x_N, x_{N-1}, x_{N-2}, \dots$  on a single time series up to time  $N$ , we can denote any univariate forecast of  $X_{N+h}$  by  $\hat{x}_N(h)$ . The accuracy of the forecast can be assessed [3] by the widely used mean square error (MSE) measure, namely  $E\left[X_{N+h} - \hat{x}_N(h)\right]^2$ .

For the case of ARMA models we have seen that when conditions for convergence apply the general linear process holds:

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j} \quad (38)$$

The minimum mean square error (MMSE) forecast of  $X_{N+h}$  then for the general linear process is given by:

$$\hat{x}_N(h) = \sum_{j=h}^{\infty} \psi_j z_{N+h-j} \quad (39)$$

where it is assumed, not only that the values of  $\psi_j$  are known, but also that data for the infinite past up to time  $N$ , namely  $x_t, t \leq N$ , are known and hence that the realized values of  $Z_t, t \leq N$ , denoted by  $z_t$ , can be found.

Equation (39) is then intuitively obvious since it follows from (38) by replacing future values of  $Z_t$  by zero, and present and past values of  $Z_t$  by their observed values. This merely requires that the MMSE forecast of all future  $Z$ 's is zero as is obviously the case when the  $Z$ 's are uncorrelated with mean zero.

As an example, let's consider the MA(2) process:

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} \quad (40)$$

This process has MMSE forecast at time  $N$  given by:

$$\hat{x}_N(h) = \begin{cases} \theta_1 z_N + \theta_2 z_{N-1} & h = 1 \\ \theta_2 z_N & h = 2 \\ 0 & h \geq 3 \end{cases} \quad (41)$$

More generally, forecasts from an ARMA model are not obtained via (39), but can be computed directly from the ARMA model equation by replacing:

- (1) future values of  $Z_t$  by zero
- (2) future values of  $X_t$  by their conditional expectation
- (3) present and past values of  $X_t$  and  $Z_t$  by their observed values  $x_t$  and  $z_t$ , respectively.

For example, the ARMA(1, 1) process:

$$X_t = \phi_1 X_{t-1} + Z_t + \theta_1 Z_{t-1} \quad (42)$$

has MMSE forecast at time  $N$  given by:

$$\hat{x}_N(h) = \begin{cases} \phi_1 x_N + \theta_1 z_N & h = 1 \\ \phi_1 \hat{x}_N(h-1) & h \geq 2 \end{cases} \quad (43)$$

Thus the forecasts are generally calculated recursively.

The above rules for computing MMSE forecasts also apply to (nonstationary) ARIMA models and seasonal ARIMA (or SARIMA) models. For example, the ARIMA(1, 1, 1) model:

$$X_t - X_{t-1} = \phi_1 X_{t-1} - \phi_1 X_{t-2} + Z_t + \theta_1 Z_{t-1} \quad (44)$$

may be rewritten as:

$$X_t = 1 + \phi_1 X_{t-1} - \phi_1 X_{t-2} + Z_t + \theta_1 Z_{t-1} \quad (45)$$

from which we see that the MMSE forecasts at time  $N$  may be calculated recursively by:

$$\hat{x}_N(h) = \begin{cases} 1 + \phi_1 x_N - \phi_1 x_{N-1} + \theta_1 z_N & h = 1 \\ 1 + \phi_1 \hat{x}_N(1) - \phi_1 x_N & h = 2 \\ 1 + \phi_1 \hat{x}_N(h-1) - \phi_1 \hat{x}_N(h-2) & h \geq 3 \end{cases} \quad (46)$$

The forecast resulting from the ARIMA(0, 1, 1) model,  $X_t - X_{t-1} = Z_t + \theta Z_{t-1}$ , is of particular interest. Here we find:

$$\begin{aligned}
 \hat{x}_N(1) &= x_N + \theta z_N \\
 &= x_N + \theta (x_N - \hat{x}_{N-1}(1)) \\
 &= (1 + \theta) x_N - \theta \hat{x}_{N-1}(1) \\
 &= \alpha x_N + (1 - \alpha) \hat{x}_{N-1}(1)
 \end{aligned}
 \tag{47}$$

where  $\alpha = 1 + \theta$ . This simple updating formula, which only utilizes the latest observation and the previous forecast, is the basis of many forecasting methods and is called *simple exponential smoothing*.

### 2.2.16 Box-Jenkins procedure – Summary

A summary of the Box-Jenkins [13] procedure to time-series analysis, in a form of a diagram is presented in the following Figure 6:

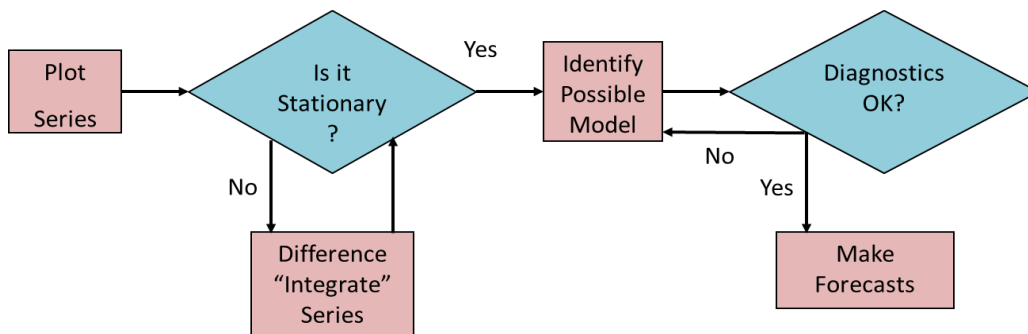


Figure 6: Box-Jenkins procedure to time-series analysis

### 2.2.17 MATLAB Box-Jenkins Methodology

The MATLAB Box-Jenkins methodology [5] is a five-step process for identifying, selecting, and assessing conditional mean models (for discrete, univariate time series data), as follows:

1. Establish the stationarity of your time series. If your series is not stationary, successively difference your series to attain stationarity. The sample autocorrelation function (ACF) and partial autocorrelation function (PACF) of a stationary series decay exponentially (or cut off completely after a few lags).
2. Identify a (stationary) conditional mean model for your data. The sample ACF and PACF functions can help with this selection. For an autoregressive (AR) process, the sample ACF decays gradually, but the sample PACF cuts off after a few lags. Conversely, for a moving average (MA) process, the sample ACF cuts off after a few lags, but the sample PACF decays gradually. If both the ACF and PACF decay gradually, consider an ARMA model.
3. Specify the model, and estimate the model parameters. When fitting nonstationary models in *Econometrics Toolbox*, it is not necessary to manually difference your data and fit a stationary model. Instead, use your data on the original scale, and create an *arima* model object with the desired degree of nonseasonal and seasonal differencing. Fitting an ARIMA model directly is advantageous for forecasting: forecasts are returned on the original scale (not differenced).
4. Conduct goodness-of-fit checks to ensure the model describes your data adequately. Residuals should be uncorrelated, homoscedastic (having constant finite variance), and normally distributed with constant mean and variance.

5. After choosing a model—and checking its fit and forecasting ability—you can use the model to forecast or generate Monte Carlo simulations over a future time horizon.

**Example:** This example shows step-by-step how to use the Box-Jenkins methodology to select an ARIMA model. The time series is the log quarterly Australian Consumer Price Index (CPI) measured from 1972 and 1991.

**Step 1: Plot the sample time series data.** The following MATLAB code in the left hand side, loads the data into the MATLAB Workspace and plots it, as shown in Figure 7 in the right hand side.

```
clc; clear all; close all; % clears command
window, workspace and figures
% Step 1. Load the data and plot it.
load Data_JAustralian % loads data into
workspace
Y = Dataset.PAU;
N = length(Y);
figure(1)
plot(Y)
xlim([0,N])
set(gca,'XTick',1:10:N);
set(gca,'XTickLabel',datestr(dates(1:10:N),17));
title('Log Quarterly Australian CPI')
```

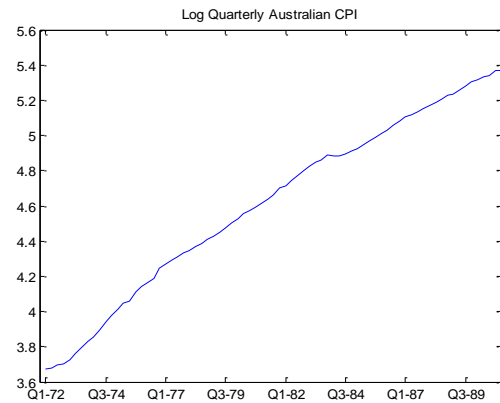


Figure 7: Plot of Quarterly Australian Consumer Price Index (CPI) from 1972 to 1991

As can clearly be seen from the Figure 7 the time series is non-stationary with an upward trend.

**Step 2: Plot the sample ACF and PACF.** The following MATLAB code in the left hand side, calculates the sample autocorrelation function (ACF) and partial autocorrelation function (PACF) for the CPI series and plots them, as shown in Figure 8 in the right hand side.

```
% Step 2. Plot the sample ACF
and PACF.
figure(2)
subplot(2,1,1)
autocorr(Y)
subplot(2,1,2)
parcorr(Y)
```

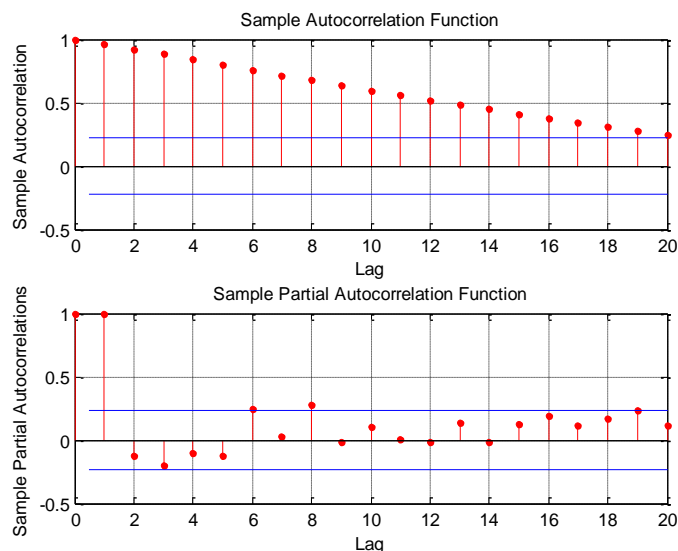


Figure 8: Plot of sample ACF and PACF of CPI

Figure 8 shows significant, linearly decaying of the sample ACF indicating a nonstationary process.



**Step 3: Difference the data.** The following MATLAB code in the left hand side, takes a first difference of the data, and plots the differenced series, as shown in Figure 9 in the right hand side.

```
% Step 3. Difference the data.
```

```
dY = diff(Y);
figure(3)
plot(dY)
xlim([0,N])
set(gca,'XTick',1:10:N);
set(gca,'XTickLabel',datestr(dates(2:10:N),17));
title('Differenced Log Quarterly Australian
CPI')
```

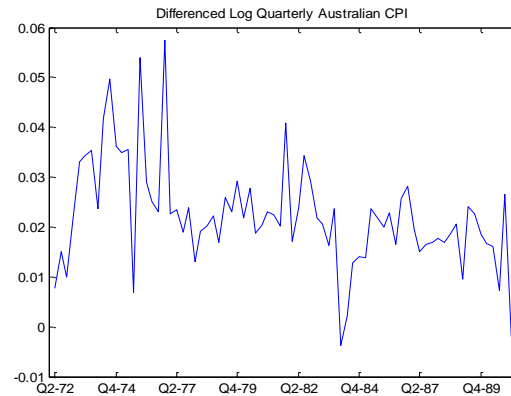


Figure 9: Plot of Differenced Quarterly Australian Consumer Price Index (CPI) from 1972 to 1991

Differencing removes the linear trend. Figure 9 shows that the differenced series appears more stationary.

**Step 4: Plot the sample ACF and PACF of the differenced series.** The following MATLAB code in the left hand side, calculates the sample autocorrelation function (ACF) and partial autocorrelation function (PACF) of the differenced CPI series and plots them, as shown in Figure 10 in the right hand side, to look for behavior more consistent with a stationary process.

```
% Step 4. Plot the sample ACF
and PACF
```

```
% of the differenced series.
```

```
figure(4)
subplot(2,1,1)
autocorr(dY)
subplot(2,1,2)
parcorr(dY)
```

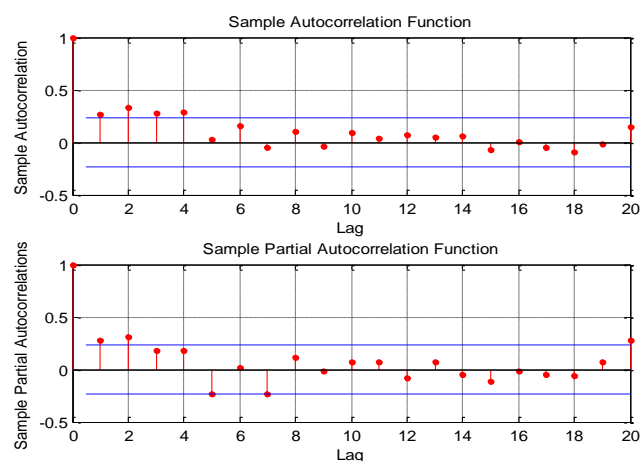


Figure 10: Plot of sample ACF and PACF of Differenced CPI

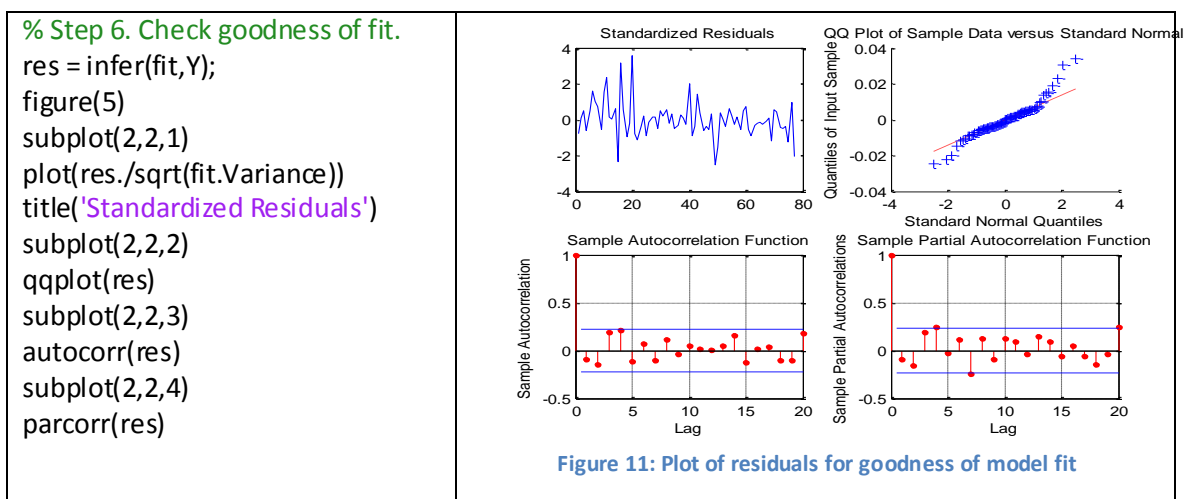
From Figure 10 the sample ACF of the differenced series decays more quickly. The sample PACF cuts off after lag 2. This behavior is consistent with a second-degree autoregressive AR(2) model.

**Step 5: Specify and fit an ARIMA(2,1,0) model.** The following MATLAB code in the left hand side, specifies and then estimates (fits), an ARIMA(2,1,0) model for the log quarterly Australian CPI. This model has one degree of nonseasonal differencing and two AR lags. By default, the innovation distribution is Gaussian with a constant variance. The results of the fitted process, as appear in the command window, are shown in the right hand side.

<pre>% Step 5. Specify and fit % an ARIMA(2,1,0) model. model = arima(2,1,0); fit = estimate(model, Y);</pre>	<p>ARIMA(2,1,0) Model: -----</p> <p>Conditional Probability Distribution: Gaussian</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Standard Value</th> <th>Error</th> <th>t</th> </tr> </thead> <tbody> <tr> <td>Constant</td> <td>0.0100723</td> <td>0.00328015</td> <td>3.07069</td> </tr> <tr> <td>AR{1}</td> <td>0.212059</td> <td>0.0954278</td> <td>2.22219</td> </tr> <tr> <td>AR{2}</td> <td>0.337282</td> <td>0.103781</td> <td>3.24994</td> </tr> <tr> <td>Variance</td> <td>9.23017e-05</td> <td>1.11119e-05</td> <td>8.30659</td> </tr> </tbody> </table> <p>-----</p>	Parameter	Standard Value	Error	t	Constant	0.0100723	0.00328015	3.07069	AR{1}	0.212059	0.0954278	2.22219	AR{2}	0.337282	0.103781	3.24994	Variance	9.23017e-05	1.11119e-05	8.30659
Parameter	Standard Value	Error	t																		
Constant	0.0100723	0.00328015	3.07069																		
AR{1}	0.212059	0.0954278	2.22219																		
AR{2}	0.337282	0.103781	3.24994																		
Variance	9.23017e-05	1.11119e-05	8.30659																		

Both AR coefficients are significant at the 0.05 significance level.

**Step 6: Check goodness of fit.** The following MATLAB code in the left hand side, infers the residuals from the fitted model and checks that these are normally distributed and uncorrelated. Figure 11 in the right hand side shows the results about the residuals.



From Figure 11 it is inferred that the residuals are reasonably normally distributed and uncorrelated.

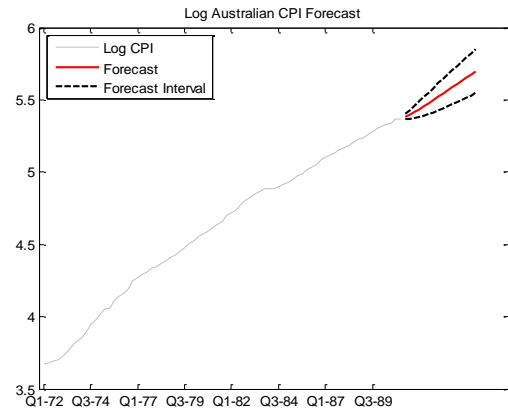
**Step 7: Generate forecasts.** The following MATLAB code in the left hand side generates forecasts and approximate 95% forecast intervals for the next 4 years (16 quarters). Figure 12 in the right hand side shows the forecasts along with the approximate 95% forecast intervals.

**% Step 7. Generate forecasts.**

```

[Yf, YMSE] = forecast(fit,16,'Y0',Y);
UB = Yf + 1.96*sqrt(YMSE);
LB = Yf - 1.96*sqrt(YMSE);
figure(6)
h1 = plot(Y,'Color',[.75,.75,.75]);
hold on
h2 = plot(78:93,Yf,'r','LineWidth',2);
h3 = plot(78:93,UB,'k--','LineWidth',1.5);
plot(78:93,LB,'k--','LineWidth',1.5);
set(gca,'XTick',1:10:N);
set(gca,'XTickLabel',datestr(dates(1:10:N),17));
legend([h1,h2,h3],'Log CPI','Forecast',...
'Forecast Interval','Location','Northwest')
title('Log Australian CPI Forecast')

```



**Figure 12: ARIMA(2,1,0) Forecasts for the Australian CPI and approximate 95% forecast intervals for the next 4 years (16 quarters)**

### 2.3 Time-Series - State space models

The phrase ‘state space’ derives from a class of models developed by control engineers for systems that vary through time [3, 8, 15]. When a scientist or engineer tries to measure a signal, it will typically be contaminated by noise so that:

$$\text{Observation} = \text{Signal} + \text{Noise}$$

In state-space models the signal at time  $t$  is taken to be a linear combination of a set of variables, called *state variables*, which constitute what is called the *state vector* at time  $t$ . Denote the number of state variables by  $m$ , and the  $(m \times 1)$  state vector by  $\theta_t$ . Then the observation may be written as:

$$X_t = h_t^T \theta_t + n_t \quad (48)$$

where  $h_t$  is assumed to be a known  $(m \times 1)$  vector, and  $n_t$  denotes the observation error, assumed to have zero mean.

In control systems engineering, a state variable is the minimum set of information needed to determine the future behavior of a dynamical system. Thus the future is independent of past values. This means that the state vector has a property called the *Markov property*, in that the latest value is all that is needed to make predictions.

It may not be possible to observe all (or even any of) the elements of the state vector,  $\theta_t$ , directly, but it may be reasonable to make assumptions about how the state vector changes through time. A key assumption of linear state-space models is that the state vector evolves according to the equation:

$$\theta_t = G_t \theta_{t-1} + w_t \quad (49)$$

where the  $(m \times m)$  matrix  $G_t$  is assumed known and  $w_t$  denotes an  $m$ -vector of disturbances having zero means.

The above two equations constitute the general form of a univariate state-space model. The first equation (48) modeling the observed variable is called the *observation* (or *measurement*) equation, while the second equation (49) is called the *transition* (or *system*) equation. An unknown constant, say  $\delta$ , can be introduced into a state-space model by defining an artificial state variable, say  $\delta_t$ , which is updated by  $\delta_t = \delta_{t-1}$  subject to  $\delta_0 = \delta$ . The ‘error’ terms in the observation and transition equations are generally assumed to be uncorrelated with each other at all time periods and also to be serially uncorrelated through time.

It may also be assumed that  $n_t$  is  $N(0, \sigma_n^2)$  while  $w_t$  is multivariate normal with zero mean vector and known variance-covariance matrix  $W_t$ . If the latter is the zero matrix, then the model reduces to time-varying regression.

Having expressed a model in state-space form, an updating procedure can readily be invoked every time a new observation becomes available, to compute estimates of the current state vector and produce forecasts. This procedure, called the *Kalman filter*, only requires knowledge of the most recent state vector and the value of the latest observation, and will be described later.

There are many interesting special cases of the state-space model. For example, the *random walk plus noise* model, also called the *local level* or *steady* model, arises when  $\theta_t$  is a scalar,  $\mu_t$ , denoting the current level of the process, while  $h_t$  and  $G_t$  are constant scalars taking the value one. Then the local level,  $\mu_t$ , follows a random walk model. This model depends on two parameters which are the two error variances, namely  $\sigma_n^2$  and  $Var(w_t) = \sigma_w^2$ . The properties of the model depend primarily on the ratio of these variances, namely  $\sigma_w^2 / \sigma_n^2$ , which is called the *signal-to-noise ratio*.

In the *linear growth* model, the state vector has two components,  $\theta_t^T = (\mu_t, \beta_t)$  say, where  $\mu_t$ ,  $\beta_t$  may be interpreted as the local level and the local growth rate, respectively. By taking  $h_t^T = (1, 0)$  and  $G_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ , we have a model specified by the three equations:

$$X_t = \mu_t + n_t \quad (50a)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + w_{1,t} \quad (50b)$$

$$\beta_t = \beta_{t-1} + w_{2,t} \quad (50c)$$

The 1<sup>st</sup> from the above three equations is the observation equation, while the 2<sup>nd</sup> and 3<sup>rd</sup> constitute the two transition equations. Of course, if  $w_{1,t}$  and  $w_{2,t}$  have zero variance, then there is a deterministic linear trend, but there is much more interest in the case where  $w_{1,t}$  and  $w_{2,t}$  do *not* have zero variance giving a local linear trend model.

A seasonal index term,  $i_t$  say, can be included to the right-hand side of 1<sup>st</sup> equation above and add a third transition equation of the form:

$$i_t = -\sum_{j=1}^{s-1} i_{t-j} + w_{3,t} \quad (50d)$$

where there are  $s$  periods in one year. The state vector now has  $(s+2)$  components, as the transition equations involve the current level, the current trend and the  $s$  most recent seasonal indices. None of these state variables can be observed directly, but they can be estimated from the observed values of  $X_t$  assuming the model is appropriate.

The above models have been proposed because they make intuitive sense for describing data showing trend and seasonal variation, and they are in state-space format directly. Many other types of model, including ARIMA models, can be recast into state-space format, and this can have some advantages, especially for estimation. For example, consider the AR(2) model:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t \quad (51a)$$

Given the two-stage lagged dependence of this model, it is not obvious that it can be rewritten in state-space format with (one-stage) Markovian dependency. However, this can indeed be done in several different ways by introducing a two-dimensional state vector which involves the last two observations. One possibility is to take  $\theta_t^T = [X_t, X_{t-1}]$  and rewrite the above equation as follows. The observation equation is just:

$$X_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \theta_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix} = X_t \quad (51b)$$

while the transition equation is:

$$\begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix} = \theta_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 0 & 1 \end{bmatrix} \theta_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} Z_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} X_{t-1} \\ X_{t-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} Z_t \quad (51c)$$

This formulation is rather artificial, and we would normally prefer the usual AR(2) formulation, especially for descriptive purposes. However, the one stage Markovian state-space format does enable the Kalman filter to be applied.

Alternative ways of re-expressing an AR(2) model in state-space form include using the state vector:

$$\theta_t^T = [X_t, \phi_2 X_{t-1}] \quad (52a)$$

or

$$\theta_t^T = [X_t, \phi_2 \hat{X}_{t-1}(1)] \quad (52b)$$

Note that two of the state vectors suggested above are observable directly, unlike the state vectors in earlier trend-and-seasonal models.

The lack of uniqueness in regard to state vectors raises the question as to how the ‘best’ formulation can be found, and what ‘best’ means in this context. We would like to find a state vector which summarizes the information in the data set in the best possible way. This means, first of all, choosing an appropriate dimension so as to include all relevant information into the state vector, but avoid including redundant information.

Except in trivial cases, a state-space model will be non-stationary and hence will not have a time-invariant ACF. Thus state-space models are handled quite differently from ARIMA models in particular. State-space models deal with non-stationary features like trend by including explicit terms for them in the model. In contrast, the use of ARIMA models for non-stationary data involves differencing the non-stationarity away, so as to model the differenced data by a stationary ARMA process, rather than by modeling the trend explicitly.

### 2.3.1 Forecasting with state-space models – the Kalman filter

For the previously presented general state-space model for the observations  $X_t$  and the state vector  $\theta_t$ , the Kalman filter procedure is a two stage procedure. Having the estimate of  $\theta_{t-1}$ , say  $\hat{\theta}_{t-1}$ , based on data up to time (t-1) together with an estimate of its variance-covariance matrix  $P_{t-1}$ , the first stage, called the *prediction stage*, forecasts  $\theta_t$  using the data up to time (t - 1). Denoting the resulting estimate by  $\hat{\theta}_{t/t-1}$  the state-space equation suggests the formula [3, 8, 15]:

$$\hat{\theta}_{t/t-1} = G_t \hat{\theta}_{t-1} \quad (53a)$$

The variance-covariance matrix of this estimate is:

$$P_{t/t-1} = G_t P_{t-1} G_t^T + W_t \quad (53b)$$

When the new measurement at time t,  $x_t$ , becomes available, the second stage of the Kalman filter, called the *updating stage*, is carried out using the formulae:

$$\hat{\theta}_t = \hat{\theta}_{t/t-1} + K_t e_t \quad (54a)$$

$$P_t = P_{t/t-1} - K_t h_t^T P_{t/t-1} \quad (54b)$$

where

$$e_t = x_t - h_t^T \hat{\theta}_{t/t-1} \quad (54c)$$

is the prediction error at time t, and:

$$K_t = P_{t/t-1} h_t / [h_t^T P_{t/t-1} h_t + \sigma_n^2] \quad (54d)$$

is called the Kalman gain matrix.

The point estimate  $\hat{x}_N(h)$  then suggested by the state-space observation equation can readily be:

$$\hat{x}_N(h) = h_{N+h}^T \hat{\theta}_{N+h} \quad (55)$$

where  $h$  is the future time of interest.

In the important special case where  $h_t$  and  $G_t$  are (known) constant functions, say  $h$  and  $G$ , then the forecast formula (55) becomes:

$$\hat{x}_N(h) = h^T G^h \hat{\theta}_N \quad (56)$$

### 2.3.2 Autoregressive Identification as a Kalman filter problem

Kalman filtering can also be applied to provide a technique for the identification of the coefficients in an AR type equation of the form:

$$y_k + a^{(1)}y_{k-1} + \dots + a^{(n)}y_{k-n} = v_k \quad (57)$$

where measurements  $y_k$  become available in discrete time intervals and where  $v_k$  is a zero mean, white Gaussian random process. The aim then is to estimate the values of the coefficients  $a^{(1)}, \dots, a^{(n)}$  using these measurements.

It is realistic to model the  $a^{(i)}$  as being subject to random perturbations and suppose that for each  $i$ :

$$a_{k+1}^{(i)} = a_k^{(i)} + w_k^{(i)} \quad (58)$$

where  $w_k^{(i)}$  is a zero mean, white, Gaussian random process, independent of  $w_k^{(j)}$  for  $i \neq j$  and also independent of  $v_k$ .

We need to assume values for the variances of  $w_k^{(i)}$  and  $v_k$ , and in assigning these values, the fullest possible knowledge must be used of the physical arrangement of which (58) and (57) constitute a representation. In other words, a variance for  $v_k$ , should be assigned on the basis of our knowledge of the noise introduced by measurement sensors, and we should assign a variance to  $w_k^{(i)}$  after an assessment, possibly subjective, of the way the  $a_k^{(i)}$  are likely to vary.

Finally, we need to assume an a priori mean and variance for each  $a^{(i)}$ , reflecting our estimate before measurements are taken of the value of these coefficients and the likely error in the estimate respectively. To apply the Kalman filtering theory, we assume too that  $a_0^{(i)}$  are Gaussian random variables. (We could alternatively drop the Gaussian assumptions and still obtain the Kalman filter as a best linear estimator).

Now we can pose the identification problem in Kalman filter terms. Define an n-dimensional state vector  $x_k$ , by:

$$x_k^{(1)} = a_k^{(1)}, x_k^{(2)} = a_k^{(2)}, \dots, x_k^{(n)} = a_k^{(n)} \quad (59)$$

Define also the n-dimensional, white, zero mean, Gaussian process  $w_k$  as the vector process formed from the  $w_k^i$ . Then (58) and (59) lead to the state equation:

$$x_{k+1} = x_k + w_k \quad (60)$$

Next, define the matrix, actually a row vector:

$$H_k^T = [-y_{k-1} \quad -y_{k-2} \quad \dots \quad -y_{k-n}] \quad (61)$$

and the process  $z_k$  by:

$$z_k = y_k \quad (62)$$

Then (57) and (61) yield:

$$z_k = H_k^T x_k + v_k \quad (63)$$

Notice that at time 0, we cannot say what  $H_k$ , is for  $k > 0$ . However, by the time  $z_k$ , is received, the value of  $H_k$ , is known. This is sufficient for the purposes of defining the Kalman filter. The filter in this case becomes:

$$\hat{x}_{k+1/k} = \hat{x}_{k/k-1} - K_k [z_k - H_k^T \hat{x}_{k/k-1}] \quad (64)$$

with

$$K_k = P_{k/k-1} H_k^T [H_k P_{k/k-1} H_k^T + R_k]^{-1} \quad (65)$$

and

$$P_{k+1/k} = P_{k/k-1} - P_{k/k-1} H_k^T [H_k P_{k/k-1} H_k^T + R_k]^{-1} H_k P_{k/k-1} + Q_k \quad (66)$$

Here,  $R_k = E[y_k^2]$  and  $Q_k = E[w_k w_k^T]$ . Equation (64) is initialized with  $\hat{x}_{0/-1}$  set equal to the vector of a priori estimates of the coefficients, and Eq. (66) is initialized with  $P_{0/-1}$  set equal to the a priori covariance matrix of the coefficients.

The estimate then can be provided by the relation:

$$\hat{y}_{k+1} = H_k^T \hat{x}_{k+1/k} \quad (67)$$

### 2.3.3 MATLAB – Autoregressive Identification with Kalman filter

**Problem:** Consider the following data set:  $y_k \Big|_{k=1}^6 = 0.5 \quad 1.1 \quad 2.2 \quad 3.3 \quad 4.4 \quad 5.5$  .

For different AR type equations of the form  $y_k + a^{(1)}y_{k-1} + \dots + a^{(n)}y_{k-n} = v_k$  and using the Kalman filter identify the coefficients  $a_k^{(1)} = x_k^{(1)}$ ,  $a_k^{(2)} = x_k^{(2)}$ , ...,  $a_k^{(n)} = x_k^{(n)}$  for different  $n = 1, 2, 3, \dots$  which model the data, provide predictions, and evaluate the predicted results.



**Solution:** The following MATLAB code implements the above Kalman filter equations, fits the data for  $n = 2$  and finds estimated forecasts (predictions)  $\hat{y}_{k+1} = H_k^T \hat{x}_{k+1/k}$ .

```
clear all;close all; clc;
z = [0.5 1.1 2.2 3.3 4.4 5.5]; % Measurements
% Kalman Filter estimation of curve fitting coefficients x for the
% estimation of  $y(k) = -[y(k-1)x_1+y(k-2)x_2+\dots+y(k-n)x_n] \implies$ 
%  $y(k) = H(k)x(k) = z(k) \implies$  Measurements  $z(k) = y(k)$ 
%  $x(k+1) = x(k) + w(k)$ ; Linear Model of curve fitting coefficients
%  $z(k) = H(k)x(k) + v(k)$ ; Model of Measurements
% -----
% Initialization of Kalman Filter Parameters
n = 2; % Kalman Filter Window size
Q_d = 0.01*eye(n,n); % Noise Covariance for the model
% G_d = eye(n,n); % Noise Distribution for the model
x_plus = -0.35*z(1:n)'; % Initial conditions for the state estimate
P_plus = 1.0*eye(n,n); % Initial conditions for the state Covariance
x_est = []; % Storage variable vector for fitting coefficients x
y_est = []; % Storage variable vector for estimates
errors = []; % Storage variable vector for estimates  $z(k)-y\_est$ 
% -----
% Beginning of Kalman Filter Loop
for k = n+1:length(z)
    H = -[z(k-n:k-1)]; % Window size data matrix H
    x_minus = x_plus; % Propagation of state estimate x
    P_minus = P_plus+Q_d; % Propagation of covariance P
    y(k) = H*x_minus;
    x_est = [x_est x_minus]; % Storage variable holding estimates x_minus
    y_est = [y_est y(k)]; % Storage variable holding estimates y_est
    % Update equations
    R = std(H); % Covariance of Matrix H
    K = (P_minus*H')*((H*P_minus*H'+R)^(-1)); residual = z(k) - y(k);
    errors = [errors residual]; % Variable holding errors  $z-y\_est$ 
    x_plus = x_minus+K*residual; % Update of state x
    P_plus = P_minus-K*H*P_minus; % Update of covariance P
end % End of Kalman Filter Loop
% -----
% Last Kalman Filter estimation
H_last = -[z(k-n+1:k)];
x_last = x_plus;
P_last = P_plus+Q_d;
y_last = H_last*x_last;
x_est = [x_est x_last] % Storage variable holding estimates x_minus
y_est = [y_est y_last] % Variable holding estimate y_est and y_last
errors
figure(1)
plot((1:length(z)), z, 'b'); hold on; plot([n+1:length(z)+1], y_est, 'm');
plot([n+1:length(z)], [errors], 'r');
legend('Data z', 'Predictions y', 'Errors (z-y)', 'Location', 'NorthWest');
```

Executing the above MATLAB code, for n=2, we obtain the estimates of the coefficients  $x\_est = \hat{x}_{k+1/k}$ , the predictions  $y\_est = \hat{y}_{k+1} = H_k^T \hat{x}_{k+1/k}$  and the residuals or *errors* =  $z_k - \hat{y}_k$ , as follows:

x_est (n=2)	-0,1750	-0,6242	-0,5644	-0,2842	-0,0669
	-0,3850	-1,3732	-1,2830	-1,2448	-1,2694

y_est (n=2)	0,5110	3,7077	5,4756	6,4151	7,2762
-------------	--------	--------	--------	--------	--------

errors (n=2)	1,6890	-0,4076	-1,0756	-0,9151	
--------------	--------	---------	---------	---------	--

For n=3 we obtain the following results.

x_est (n=3)		-0,1750	-0,2515	-0,1687	0,0373
		-0,3850	-0,5534	-0,4176	-0,3027
		-0,7700	-1,1067	-1,0677	-1,1045

y_est (n=3)		2,2050	5,1464	6,4473	7,2835
-------------	--	--------	--------	--------	--------

errors (n=3)		1,0950	-0,7464	-0,9473	
--------------	--	--------	---------	---------	--

A plot of the original data set, the predictions, and the errors which provides an assessment for the evaluation of the results, for n=2 is presented in Figure 13 and for n=3 is presented Figure 14.

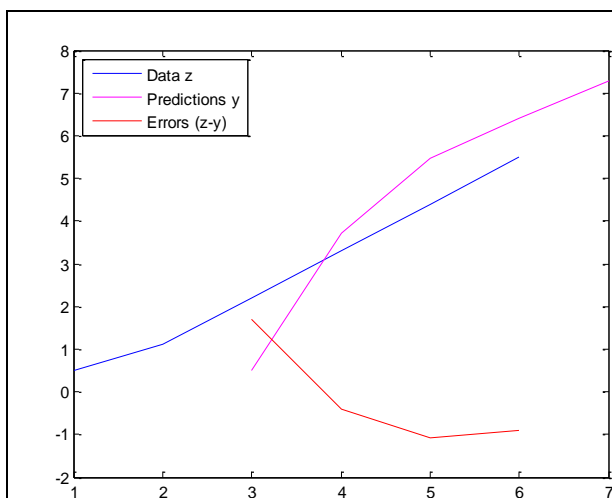


Figure 13: MATLAB – Autoregressive Identification with Kalman filter for n=2

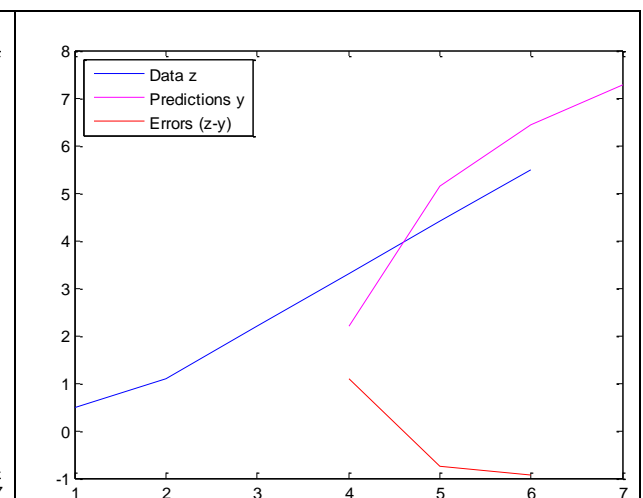


Figure 14: MATLAB – Autoregressive Identification with Kalman filter for n=3

## 2.4 Time-Series – Neural networks Non-linear model and Forecasting

A completely different type of non-linear model is provided by *Neural networks* [3] (abbreviated NNs), whose structure is thought to mimic the design of the human brain in some sense. NNs have been applied successfully to a wide variety of scientific problems, and increasingly to statistical applications, notably pattern recognition.

A neural net can be thought of as a system connecting a set of inputs to a set of outputs in a possibly non-linear way. In a time-series context, the ‘output’ could be the value of a time series

to be forecasted and the ‘inputs’ could be lagged values of the series and of other explanatory variables. The connections between inputs and outputs are typically made via one or more hidden layers of *neurons* or *nodes*. The structure of an NN is usually called the *architecture*. Choosing the architecture includes determining the number of layers, the number of neurons in each layer, and how the inputs, hidden layers and output(s) are connected. Figure 15 shows a typical NN with three inputs, and one hidden layer of two neurons.

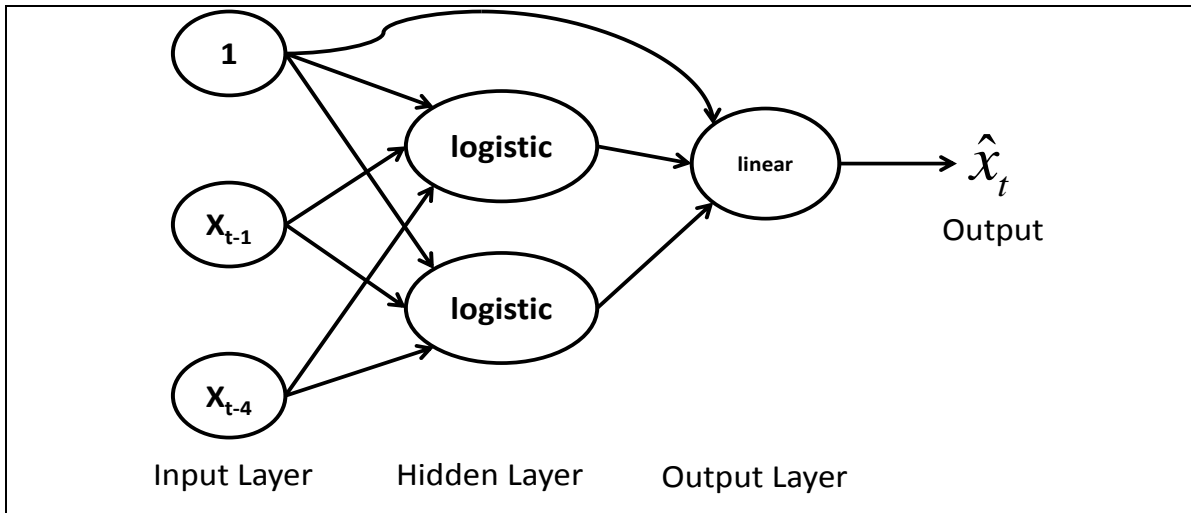


Figure 15: Architecture for a typical NN for time-series forecasting with three inputs (the lagged values at  $(t - 1)$  and  $(t - 4)$ , and a constant), one hidden layer of two neurons with logistic functions, and one output linear neuron (the forecast)

The net in the Figure 15 is of the usual *feed-forward* type as there are no feedback loops. A suitable architecture for a given problem has to be determined from the context, perhaps using external considerations and perhaps using the properties of the data. Sometimes trial-and-error is needed, for example, to choose a sensible number of hidden neurons. Thus if we want to forecast quarterly data, for example, then it is natural to include the values at lags one and four as inputs when determining the one-step-ahead forecast. In addition, it is usually advisable to include a constant input term which for convenience may be taken as unity. One hidden layer of two neurons, as in Figure 15, is usually large enough unless the series is very non-linear. Thus the architecture in Figure 15 seems reasonable for forecasting quarterly data, and can be likened to a sort of non-linear (auto) regression model.

Each input, in Figure 15, is connected to both hidden layer neurons, and both hidden layer neurons are connected to the output neuron. There is also a direct connection from the constant input to the output. The ‘strength’ of each connection is measured by a parameter called a *weight*. There may be a large number of such parameters to estimate.

A numerical value is calculated for each neuron at each time period,  $t$ , as follows. Let  $y_{i,t}$  denote the value of the  $i$ th input at time  $t$ . In our example, the values of the inputs are  $y_{1,t} = 1$ ,  $y_{2,t} = x_{t-1}$  and  $y_{3,t} = x_{t-4}$ . Let  $w_{ij}$  denotes the weight of the connection between input  $y_i$  and the  $j$ th neuron. This is assumed to be constant over time. For each neuron, we now calculate a weighted linear sum of the inputs, say  $\sum w_{ij} y_{i,t} = v_{j,t}$ , for  $j = 1, 2$ . The analyst then has to choose a function, called an *activation function*, for transforming the values of  $v_j$  into a final value for the neuron. This function is typically nonlinear. A commonly used function is the *logistic function*:

$$z = 1 / (1 + e^{-v}) \quad (68)$$

which gives values in the range (0,1). In our example this gives values  $z_{1,t}$  and  $z_{2,t}$  for the two neurons at each time period,  $t$ . A similar operation can then be applied to the values of  $z_{1,t}$ ,  $z_{2,t}$  and the constant input in order to get the predicted output.

However, the logistic function should not be used at the output stage in time-series forecasting unless the data are suitably scaled to lie in the interval (0, 1). Otherwise the forecasts will be of the wrong order of magnitude. Instead, a linear function of the neuron values may be used, which implies the identity activation function at the output stage.

The introduction of a constant input unit, connected to every neuron in the hidden layer and also to the output, avoids the necessity of separately introducing what computer scientists call a *bias*, and what statisticians would call an intercept term, for each relation. Essentially the 'biases' are replaced by weights which measure the strength of each connection from the unit input and so become part of the overall set of weights (the model parameters) which can all be treated in the same way.

For an NN model with one hidden level of  $H$  neurons, the general prediction equation for computing a forecast of  $x_t$  (the output) using selected past observations,  $x_{t-j_1}, \dots, x_{t-j_k}$ , as the inputs, may be written (rather messily) in the form:

$$\hat{x}_t = \phi_o \left( w_{co} + \sum_{h=1}^H w_{ho} \phi_h \left( w_{ch} + \sum_{i=1}^k w_{ih} x_{t-j_i} \right) \right) \quad (69)$$

where  $w_{ch}$  denote the weights for the connections between the constant input and the hidden neurons, for  $h = 1, \dots, H$ , and  $w_{co}$  denotes the weight of the direct connection between the constant input and the output. The weights  $w_{ih}$  and  $w_{ho}$  denote the weights for the other connections between the inputs and the hidden neurons and between the neurons and the output, respectively. The two functions  $\phi_h$  and  $\phi_o$  denote the activation functions used at the hidden layer and at the output, respectively.

We use the notation  $NN(j_1, \dots, j_k; H)$  to denote the NN with inputs at lags  $j_1, \dots, j_k$  and with  $H$  neurons in the one hidden layer. Thus Figure 13 above represents an  $NN(1, 4; 2)$  model.

The weights to be used in the NN model are estimated from the data by minimizing the sum of squares of the within-sample one-step-ahead forecast errors, namely:

$$S = \sum_t (\hat{x}_{t-1}(1) - x_t)^2 \quad (70)$$

over a suitable portion of the data.

This non-linear optimization problem is no easy task. It is sound practice to divide the data into two sections, to fit the NN model to the first section, called the *training set*, but to hold back part of the data, called the *test set*, so as to get an independent check on predictions.

Various fitting algorithms have been proposed for NN models, and many specialist packages are now available to implement them. However, even the better procedures may take several thousand iterations to converge, and yet may still converge to a local minimum. This is partly because there are typically a large number of parameters (the weights) to estimate, and partly because of the non-linear nature of the objective function.

The NN literature tends to describe the iterative estimation procedure as being a ‘training’ algorithm which ‘learns by trial and error’. Much of the available software used a popular algorithm called *back propagation* for computing the first derivatives of the objective function, so that  $S$  may be minimized. The starting values chosen for the weights can be crucial and it is advisable to try several different sets of starting values to see if consistent results are obtained. Other optimization methods are still being investigated and different packages may use different fitting procedures. The last part of the time series, the *test set*, is kept in reserve so that genuine out-of-sample forecasts can be made and compared with the actual observations.

The above equation effectively produces a *one-step-ahead forecast* of  $x_t$ , namely  $\hat{x}_{t-1}(1)$ , as it uses the actual observed values of all lagged variables as inputs, and they could include the value at lag one. If *multi-step-ahead forecasts* are required, then it is possible to proceed in one of two ways. Firstly, one could construct a new architecture with several outputs, giving forecasts at one, two, three . . . steps ahead, where each output (forecast) would have separate weights for each connection to the neurons. Alternatively, the one-step-ahead forecast can be ‘fed back’ to replace the lag-one value as one of the input variables. The same architecture could then be used to construct the two-step-ahead forecast, and so on. The latter option is usually preferred.

Note that some analysts fit NN models so as to get the best forecasts of the test set data, rather than the best fit to the training data. In this case the test set is no longer ‘out-of-sample’ in regard to model fitting and so a third section of data should be kept in reserve so that genuine out-of-sample forecasts can be assessed. This does not always happen!

The number of parameters in an NN model is typically much larger than in traditional time-series models, and for a single-layer NN model is given by:

$$p = (k + 2)H + 1 \quad (71)$$

where  $k$  is the number of input variables (excluding the constant) and  $H$  is the number of hidden neurons. For example, the architecture in the above Figure 15 (where  $k$  and  $H$  are both two) contains 9 connections and hence has 9 parameters (weights). The large number of parameters means there is a real danger that model-fitting will ‘overtrain’ the data and produce a spuriously good fit which does not lead to better forecasts. This motivates the use of model comparison criteria, which penalize the addition of extra parameters. It also motivates the use of an alternative fitting technique called *regularization* wherein the ‘error function’ is modified to include a penalty term which prefers ‘small’ parameter values. This is analogous to the use of a ‘roughness’ penalty term in nonparametric regression with splines. Research is continuing on ways of fitting NN models, both to improve the numerical algorithms used for doing this, and to explore different ways of preventing over-fitting.

### 3. Social Networking and Related Work

This chapter gives a short review about Social Networking and provides related work to the Analyze-me application.

#### 3.1 Social networking

Social networking is the grouping of people into specific groups. While the internet is widely spread, millions of people can join these groups online. Thus, online social networking is very popular.

Once you are granted access to a social networking website you can interact with other people, view their profiles, contact them, share pictures, etc. The following Figure 16 depicts the concept of social networking.



Figure 16: Social Network

Apart from making friends by social networking online, you have the opportunity to discover alternative habits and cultures from all over the world [16]. Social networking allows people to express their individuality and meet people with similar interests. This includes having profiles, friends, and groups as described below.

**Profile.** Profiles contain basic personal information, like age, home address, and other personal information like favorite movies or music [17].

**Friends or connections.** Friends are your trusted members and are allowed to post comments on your profile or send you private messages, view your photos and personal information [17].

**Groups or 'networks'.** Most social networks use groups to help you find people with similar interests, and organize friends. You can create different groups of friends, enabling only a specific group to access certain photos or content of your profile [17].

The following Figure 17 presents some Logos of social networks.



Figure 17: Social Network Logos

The most popular social network websites are the Twitter, Facebook, LinkedIn, and KLOUT. A short description of the ones used in this Thesis is provided below.

### 3.1.1 Twitter

Twitter is an online social networking that enables users to send and read text-based messages, described as the “SMS” of the internet. Its logo is presented in Figure 18 below.



Figure 18: Twitter Logo

Twitter users can send text messages up to 140 characters containing their thoughts, activities, questions, or anything else. Those messages are called tweets. People who follow you will be able to see those tweets. You can choose who to follow, being able to see what he tweets. The major difference compared to instant messenger is that you can send your message to many users simultaneously, spreading it far more effectively [18].

### 3.1.2 Twitter API

The twitter application programming interface offers various features such as “follow buttons” allowing users to follow someone on twitter by simply pressing the button within an application using the API, sharing information by tweeting through an application instead of the twitter web page, and displaying live feed of a user’s tweets in a section of the application using the API. Also, authentication method is provided so users can use an application only after signing in with their twitter account. More information about what twitter API has to offer can be found at: <https://dev.twitter.com/>.

In this application, the twitter API was used to obtain user picture, number of tweets, number of friends, number of followers and name.

### 3.1.3 LinkedIn

LinkedIn is a social networking website for people in professional occupations. Founded in December 2002 and launched on May 5, 2003. It is mainly used for professional networking.

One purpose of the site is to allow registered users to maintain a list of contact details of people with whom they have some level of relationship, called Connections. Users can invite anyone to become a connection. However, if the invitee selects "I don't know" or "Spam", this counts against the inviter. If the inviter gets too many of such responses, the account may be restricted or closed [23]. Its logo is presented in Figure 19 below.



Figure 19: LinkedIn Logo

Users can upload their resume or design their own profile in order to showcase work and community experiences. It can then be used to find jobs, people and business opportunities recommended by someone in one's contact network. Also, employers can list jobs and search for potential candidates.

The "gated-access approach" of LinkedIn is intended to build trust among the service's users. That means contact with any professional requires either an existing relationship, or the intervention of a contact of theirs [23].

### 3.1.4 LinkedIn API

The LinkedIn API offers information about a user such as first name, last name, education, connections, etc. Also, the API provides a form of messaging between users connected to the member sending the message. Furthermore, information about companies such as basic profile data, name, website, RSS streams and twitter feed can be retrieved by using the API. More information on what the API has to offer can be found at: <http://developer.linkedin.com/apis>.

In this application, the LinkedIn API was used in a minimal degree only to force the user to authenticate his account with LinkedIn before proceeding with the analysis. The reason for this is because the LinkedIn API provides only first degree connections of the user, but for the analysis it is mandatory that second and third degree connections are available.



## 3.2 Related Work - Social Media Analytics: KLOUT

### 3.2.1 KLOUT

KLOUT is not a social networking site; it is a company that provides social media analytics based on user's data taken from sites such as Twitter and Facebook. It measures the size of a person's network, his/her actions, and measures how other people interact with those actions. Its logo is presented in Figure 20 below.



Figure 20: KLOUT Logo

Klout scrapes social network data and creates profiles on individuals and assigns them a "Klout score. Each user receives a score between one and 100 that is based on 30 separate variables, but Klout does not provide any further information on the algorithm used for the analysis. Klout scores are supplemented with three nominally more specific measures, which Klout calls "True Reach," "Amplification score," and "Network Impact."

According to Klout.com, True Reach is based on the size of a person's "engaged audience" of followers and friends who actively listen and react to his or her online messages. Amplification Score relates to the likelihood that one's messages will generate actions (retweets, @messages, likes, and comments). Network Score reflects the computed influence value of a person's engaged audience [24]. The following Figure 21 presents a graph of KLOUT analysis.

## Score Analysis

You create content that is spread throughout your network and drives discussions

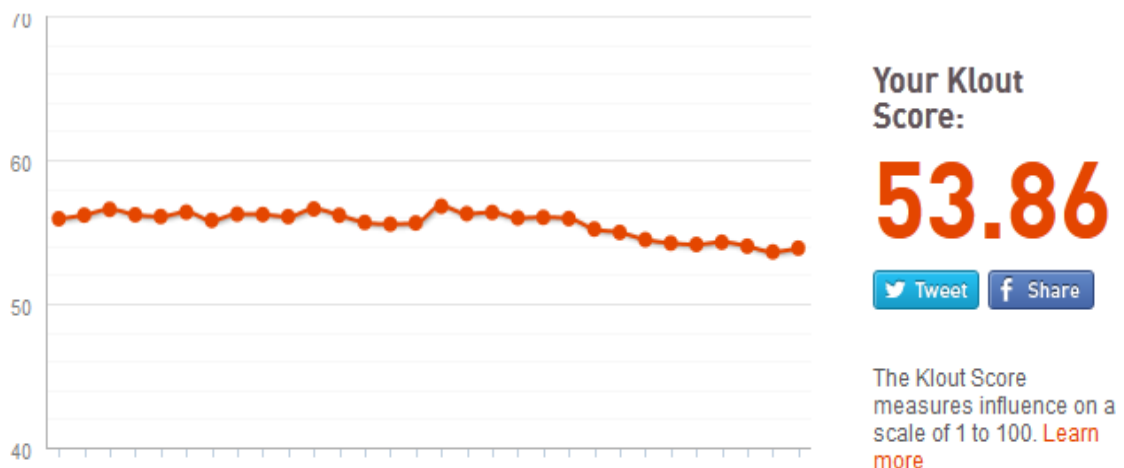


Figure 21: KLOUT score graph

### 3.2.2 EdgeRank Checker

EdgeRank Checker is a Facebook oriented analytic service that lets users measure how well their posts influence their Facebook fans. The user receives rankings for his posts that let him determine which updates have the best results. This simple tool can help users promote their messages to more fans and improve their level of engagement with customers.

More information about the EdgeRank Checker at the website: <https://www.edgerankchecker.com/>. Additional information concerning analytics can be found at the website: <http://www.leapgo.com/blog/bid/148403/Top-5-Social-Media-Analytics-Tools>.

### 3.2.3 Twitalyzer

Twitalyzer is very similar to the KLOUT service with the major difference being that Twitalyzer is not free. Instead, it is a paid service that is geared toward small companies and local businesses with less than 1000 employees. While Klout provides information about a user's influence, Twitalyzer helps users learn more about their followers and the things they care about most.

Twitalyzer analyzes the Twitter streams of followers and presents geographic information and data on trending topics. It allows you to group users by categories and locations in order to get the reports that matter most.

More information about the Twitalyzer at the website: <http://twitalyzer.com/>. Additional information concerning analytics at the website: <http://www.leapgo.com/blog/bid/148403/Top-5-Social-Media-Analytics-Tools>.

#### 4. Architecture of Analyze-me Application

The objective of the Analyze-me application is to provide social analytics and score predictions based on user influence and behavior on social networks. The application relies on information taken from KLOUT and Twitter APIs. Its basic architecture is shown in the following Figure 22 which shows the basic flow of information and processing.

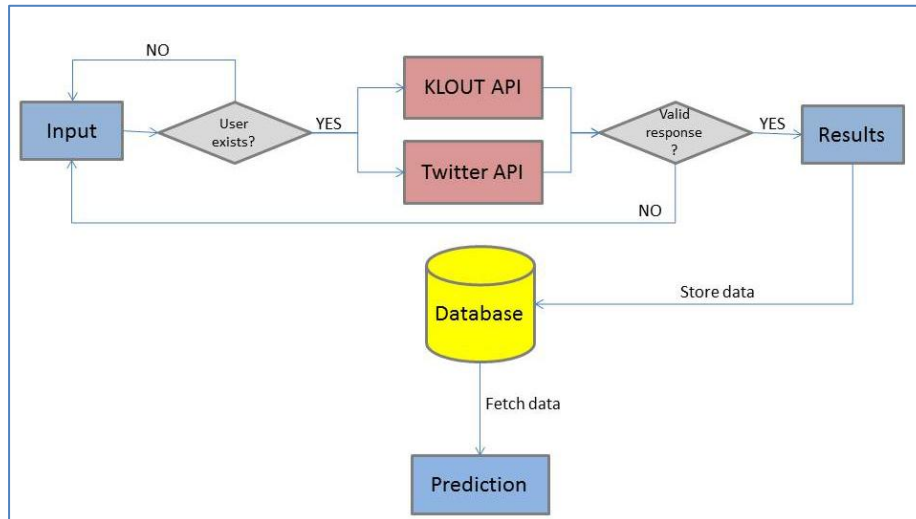


Figure 22: Architecture of Analyze-me application

In more details, the **Input module** obtains the user's name through a graphical interface and tests if the username provided exists. If the username is invalid, the process restarts. Else, if everything is ok the Input module makes an API call to the KLOUT and Twitter API modules using the username.

The **KLOUT API module** makes an http request to the KLOUT server providing the username obtained previously. Then, the API returns a response in xml format containing either an error or the user's analysis data. If the response is valid, the data is passed to the results module. Else, the process is terminated with a response error.

The **Twitter API module** similarly makes an http call to the Twitter server requesting user information matching the previously supplied username. If the response is valid, the data is passed to the results module. Else, the process is terminated with a response error.

The **Results module** is responsible for presenting the analysis results to the user through a graphical interface. Explains the result scores and offers the option to move on to the prediction module. Also, the results module stores the user's score via the database module described next.

The **Database module** stores user score each time a user analysis is being run. This way, the application is able to keep track of each user's score. Also, the database module provides these stored data to the prediction module in order to estimate user's future score.

The **Prediction module** fetches past user scores from the database and uses them to provide an estimation of the future score. Depending on how many past scores are recorded, the predictive algorithm can predict the user's score one day or one week in the future.

## 5. Development and Implementation of the Analyze-me Application

### 5.1 Interface development

The website layout was designed using the Joomla! 1.5 environment (<http://www.joomla.org/>). The selected theme is communa2Plaza dark that looks like in the following Figure 23:



Figure 23: Analyze-me dark template

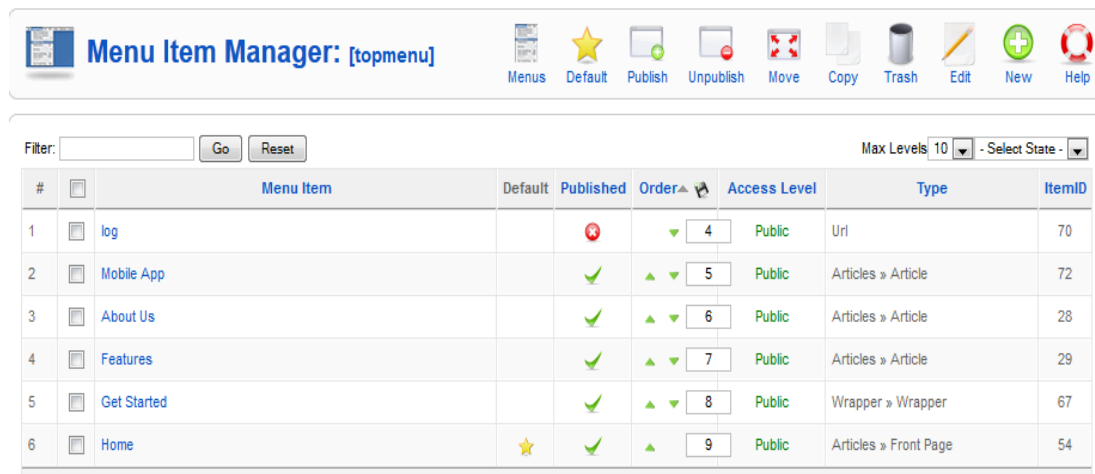
Joomla enables inserting content as articles to the existing sections dynamically. That way it's easy to add or remove content without having to re-upload the website for every change made. The website interface and content is managed directly from the website back-end, accessed from the address: <http://websiteURL/administrator>.

Organizing content (Content->Article manager) looks as in Figure 24 below:

#	Title	Published	Front Page	Order	Access Level	Section	Category	Author	Date	Hits	ID
1	Mobile App			1	Public			Perry Leros	28.04.12	81	59
2	Features			2	Public			Perry Leros	06.05.11	469	56
3	About us			3	Public			Perry Leros	08.05.11	360	57
4	intro			4	Public			Perry Leros	05.05.11	72	51

Figure 24: Joomla! Article manager

Personalizing the menu bar on top of the website to our needs is done in a similar way from Menu->Menu Manager as seen in Figure 25 below:



Menu Item Manager: [topmenu]

Filter:  Go Reset Max Levels: 10 - Select State -

#	Menu Item	Default	Published	Order	Access Level	Type	ItemID
1	log			4	Public	Url	70
2	Mobile App			5	Public	Articles » Article	72
3	About Us			6	Public	Articles » Article	28
4	Features			7	Public	Articles » Article	29
5	Get Started			8	Public	Wrapper » Wrapper	67
6	Home			9	Public	Articles » Front Page	54

Figure 25: Joomla! Menu Item Manager

As seen above, the main menu bar consists of 5 items. Mobile app, about us, features, get started, home (Log is hidden and does not appear on the menu). Looks as in Figure 26 on the actual website:

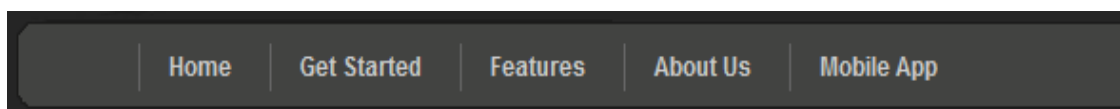


Figure 26: Analyze-me main menu bar

Also, Photoshop CS5 was used to create all images and logos used in the application (<http://www.adobe.com/products/photoshop.html>).

Photoshop is a very popular graphics editing software that was used to create all logos appearing on the Analyze-me.com website (see Figure 27) and mobile application. Also, the background layers of the website were modified using Photoshop to meet the needs of the Joomla dark theme.

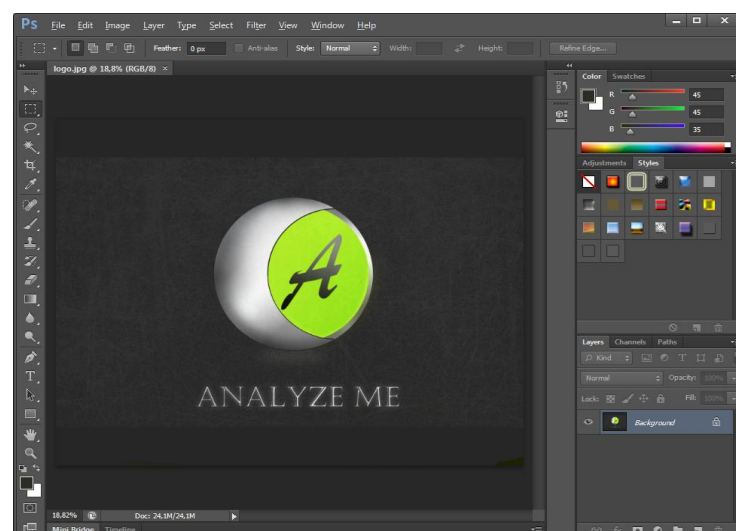


Figure 27: Photoshop working environment

## 5.2 Database development

The website is hosted on [www.freehostia.com](http://www.freehostia.com) which offers phpMyAdmin; a way to handle the main database used for analyze-me.com website and mobile app. (<http://www.phpmyadmin.net>).

So what is phpMyAdmin? It's free software written in PHP language intended to handle the administration of MySQL databases over the web.

It supports most MySQL features such as:

- browse and drop databases, tables, views, fields and indexes
- create, copy, crop, rename, alter databases, tables, fields and indexes
- execute any SQL statement, batch-queries support
- Import data from SQL

The control panel looks as in the following Figure 28:

The screenshot shows the phpMyAdmin interface for the 'perler2\_ei' database. The table structure is as follows:

Field	Type	Collation	Attributes	Null	Default	Extra	Action
score	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
date	varchar(50)	latin1_swedish_ci		Yes	2012-01-01		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
one	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
two	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
three	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
four	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]
five	varchar(50)	latin1_swedish_ci		Yes	NULL		[Edit] [Delete] [Refresh] [Drop] [Add] [Copy]

Below the table structure, the 'Indexes' section shows 'No index defined!'. The 'Space usage' section shows: Data 2,488 Bytes, Index 1,024 Bytes, Total 3,512 Bytes. The 'Row Statistics' section shows: Statements, Value, Format, Collation latin1\_swedish\_ci, Rows 57, Row length 0, Row size 62 Bytes, Creation Apr 28, 2012 at 05:16 PM, Last update Aug 21, 2012 at 09:31 AM.

The 'Run SQL query/queries on database perler2\_ei' section contains the query: `SELECT * FROM `perlyeros` WHERE 1`. The 'Fields' dropdown menu is set to 'score'.

Figure 28: phpMyAdmin control panel

On the left there is a list of all available tables in the database, while on the top are the basic database operations (browse, empty, drop table, insert, etc.).

Having a table selected, previews every field's attributes in the middle while on the bottom of the page there is an empty textbox where you can run SQL commands.

Most of the tables were created within the php code, but phpMyAdmin is a good way to monitor the database and have an overall image of its structure.

For example, if I ask the database to show me the contents of table perryleros using the following command:

```
SELECT * FROM `perryleros`
```

I get the following array (Figure 29):

			score	date	one	two	three	four	five
<input type="checkbox"/>			47.63	2012-04-28	61	119	35	260	36.87
<input type="checkbox"/>			47.63	2012-04-29	62	119	35	260	36.87
<input type="checkbox"/>			47.12	2012-04-30	61	119	32	233	36.27
<input type="checkbox"/>			46.93	2012-05-01	61	119	32	230	35.88
<input type="checkbox"/>			46.93	2012-05-02	61	119	32	230	35.88
<input type="checkbox"/>			46.78	2012-05-03	61	119	31	224	35.67
<input type="checkbox"/>			46.61	2012-05-04	61	119	30	221	35.33
<input type="checkbox"/>			46.55	2012-05-05	62	119	30	228	34.96
<input type="checkbox"/>			46.22	2012-05-06	61	119	29	224	34.23
<input type="checkbox"/>			46.04	2012-05-07	61	119	28	221	33.85
<input type="checkbox"/>			45.83	2012-05-08	61	119	27	218	33.42
<input type="checkbox"/>			45.57	2012-05-09	61	119	26	213	32.91
<input type="checkbox"/>			45.57	2012-05-10	61	119	26	213	32.91
<input type="checkbox"/>			45.37	2012-05-11	62	119	25	210	32.48

Figure 29: Database test table

I can delete an entry by just clicking the 'X' mark on the left of it. It's much easier to handle the database from a graphical interface like this. The items **one, two, three, four, and five** are just sample values used to demonstrate the table layout. There is a table for each user that has used analyze-me at least once. The table name is the same as the user's name, and contains the klout score each day the user used the service, followed by the date the analysis was done.

### 5.3 Prediction Interface development

Microsoft Expression Web 4 is a web page editor that was used to write the PHP code that handles all the operations of the website, such as displaying content, saving and retrieving data from the database, keeping log files of user's actions, etc. ([http://www.microsoft.com/expression/products/Web\\_Overview.aspx](http://www.microsoft.com/expression/products/Web_Overview.aspx)).

As an example, below in Figure 30, we see `get.php` file contents within Expression Web 4 editor:

```

88 $url2=urlencode("http://api.twitter.com/1/users/show.xml?screen_name=".$input1);
89
90
91 $xml = simplexml_load_file($url) or die("User Not Accessible, Profile locked, Inva
92 $xml2 = simplexml_load_file($url2) or die("OOPS!! Temporary API error");
93
94
95 else{
96 echo "<script>alert('Invalid Username!!')</script>";
97 echo ("Invalid Username: Please go Back and retry");
98 exit();
99
100
101 //save to xml to pass to next page
102 $xml->asXML("xml.xml");
103 $xml2->asXML("xml2.xml");
104
105 //save score to file
106 $score = "score.txt";
107 $fh = fopen($score, 'w') or die("can't open data file");
108 fwrite($fh, $xml->user->score->kscore);
109 fclose($fh);
110
111 //save delta1 to file<-----
112 $delta1 = "delta1.txt";
113 $fh = fopen($delta1, 'w') or die("can't open data file");
114 fwrite($fh, $xml->user->score->delta_1day);
115 fclose($fh);
116
117 //save delta5 to file<-----
118 $delta5 = "delta5.txt";
119 $fh = fopen($delta5, 'w') or die("can't open data file");
120 fwrite($fh, $xml->user->score->delta_5day);
121 fclose($fh);
122 $ddd=date("Y-m-d");
123 $nnn=$xml->user->twitter_screen_name;
124 $k1score=$xml->user->score->kscore;
125
126 $username=$xml->user->twitter_screen_name;

```

Figure 30: Sample php file

## 5.4 Implementation of the Analyze-me Application

### 5.4.1 Twitter Analysis

In case the user selects Twitter Analysis from the main menu the following screen appears as in Figure 31 below.

# Twitter Popularity Analysis:

Figure 31: Twitter analysis getting started

The **userF.php** file is run to open the starting page. When the user enters his twitter username, a check is being made to ensure that the input field is not null. This check is done with the following html command:



```
onclick="if(form1.input1.value.length==0) alert('Invalid Username'); else form1.submit();"
```

In this way, when the user presses the submit button, a null check is performed and if everything is ok, then the form gets submitted passing the username to the next page, which is **get.php**.

In the results page **get.php** the following actions occur:

- 1) Obtain the username from previous page
- 2) Call KLOUT API with the stored username and save response to variable
- 3) Call Twitter API with the stored username and save response to variable
- 4) Test if API response is valid or display an error message and save to xml variables
- 5) Save both xml to files for future use
- 6) Save username, score, delta1, delta5 to files for future use
- 7) Connect to database
- 8) Check if user exists in database to add score entry, or else create new user table
- 9) Display analysis results
- 10) Display graph

Let's further describe what happens during these actions.

- 1) Using \$\_POST method, the twitter username from the previous page is caught and saved into the variable \$input1.

```
$input1=$_POST['input1'];
```

- 2) Calling the Klout API is done by url command using klout developer key, for example my klout statistics would require the following command to call:

```
http://api.klout.com/1/users/show.xml?key=w9ug4rrkmcrpmxt3gszscbfq&users=perryleros
```

So in this case to generalize, the API call command is:

```
$url='http://api.klout.com/1/users/show.xml?key=w9ug4rrkmcrpmxt3gszscbfq&users='.$input1;
```

Saving the result in a variable called \$url1.

- 3) The same goes for twitter API call

```
$url2=urlencode("http://api.twitter.com/1/users/show.xml?screen_name=".$input1);
```

Where input1 is the username from the previous page.

- 4) Now we must save the response in xml format variable, after testing for validity

```
$xml = simplexml_load_file($url) or die("User Not Accessible, Profile locked, Invalidusername, or New account. Please go back and retry");
$xml2 = simplexml_load_file($url2) or die("OOPS!! Temporary API error");
}
else{
echo "<script>alert('Invalid Username!!')</script>";
echo ("Invalid Username: Please go Back and retry");
exit();}
```

So if the API response is not valid, an error message appears.

- 5) Next step is to save both xml responses to xml files for future use. This is done simply with the command:

```
$xml->asXML("xml.xml");
$xml2->asXML("xml2.xml");
```

Now we have the responses saved in xml.xml and xml2.xml.

- 6) Accessing parts of the xml structure is done by using “->”, for example I want to access the node score, I will have to use: **\$xml->user->score**.

We want to save the most important nodes of the xml to file for ease of access in the future. Username, score, delta1, delta5 variables are stored in text files like this:

**//save score to file**

```
$score = "score.txt";
$fh = fopen($score, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->kscore);
fclose($fh);
```

**//save delta1 to file** **\$delta1 = "delta1.txt";**

```
$fh = fopen($delta1, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_1day);
fclose($fh);
```

**//save delta5 to file**

```
$delta5 = "delta5.txt";
$fh = fopen($delta5, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_5day);
fclose($fh);
```

**//save username to file**

```
$username=$xml->user->twitter_screen_name;
$usr = "usr.txt";
$fh = fopen($usr, 'w') or die("can't open data file");
fwrite($fh, $username);
fclose($fh);
```

- 7) Connecting to a database through PHP is done simply using the following command:

```
//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());
```

The command uses 3 parameters (in bold): server url, database name, database password. The last like is to catch a connect error.

- 8) Analyze-me keeps record of user's score since the first time he used the service. A table with the user's name is created in the database and is updated every time the user uses the analyze-me service. A query is submitted to the database asking to return all tables with similar name as the username provided.

```
$result=mysql_query("SHOW TABLES FROM perler2_ei LIKE '$input1'")
or die(mysql_error());
```

Then, a check is being made to see if the table exists. If it does, a new entry is created containing current score, date, and all five parameters returned by the API.

If the table doesn't exist, meaning we have new user; a new table is created with the user's name.

```
if(mysql_num_rows($result)==0){
//echo ("User table doesn't exist");

//create new table with user's name
mysql_query("CREATE TABLE $input1 (score varchar(50), date varchar(50) default '2012-01-01', one varchar(50), two varchar(50), three varchar(50), four varchar(50), five varchar(50))");
//insert score and current date to the table
mysql_query("INSERT INTO $input1 VALUES($kscore,CURDATE(),$one,$two,$three,$four,$five);}
```

To avoid multiple entries the same day, another mechanism was placed to check if a user has already made an entry the current day. If the entry exists, no further entries are recorded the same day.

```
else{//if table exists, then if the user hasn't made an entry already the current date make the entry.
$lastentry=mysql_query("SELECT date FROM $input1 ORDER BY date DESC LIMIT 1");
$row1 = mysql_fetch_array($lastentry);
$lastdate=$row1['date'];

$diff=mysql_query("SELECT DATEDIFF(CURDATE(),'$lastdate') AS DiffDate");
$row2 = mysql_fetch_array($diff);
//how many days difference since last entry

if ($row2['DiffDate']!=0){
//insert score to user's table if there is no other entry today

mysql_query("INSERT INTO $input1 VALUES($kscore,CURDATE(),$one,$two,$three,$four,$five)");
}
```

- 9) All results are displayed through HTML textboxes using php code like this:

```
type="text" value="<?php echo "something"; ?>"
```

So replacing “something” with our API response variables below makes the results visible:

```
$xml->user->score->kscore  
$xml->user->score->kclass  
$xml->user->score->kclass_description  
$xml->user->score->amplification_score  
$xml->user->score->network_score  
$xml->user->score->>true_reach  
$xml2->followers_count  
$xml2->friends_count  
$xml->user->twitter_screen_name  
$xml2->profile_image_url
```

- 10) On the bottom of the page there is an iFrame that contains the graph.php file which uses the files that we saved in txt before to create a graph of the user’s progress over time. The graph requires jpgraph.php and jpgraph\_line.php. These files were acquired from the official website <http://jpgraph.net/>. Initially, the 3 files holding scores for the last 5 days are called to import the data into variables:

```
$theData = file_get_contents('score.txt');  
$theData2 = file_get_contents('delta1.txt');  
$theData3 = file_get_contents('delta5.txt');
```

Then, after tweaking parameters such as colors, labels and background picture, the graph is plotted:

```
$p1 = new LinePlot($datay1);  
$graph->Add($p1);
```

Finally, there is a new feature offering prediction of future user score. On the bottom of the page is the prediction button which redirects to the Prediction page. Prediction will be described in later.

#### 5.4.2 LinkedIn Analysis

In case the user selects LinkedIn Analysis from the main menu the following screen appears as in Figure 32 below.

## LinkedIn Popularity Analysis:



Authentication with LinkedIn is required

What type of LinkedIn user are you? Find out now...



Figure 32: LinkedIn analysis getting started

Authentication is required to access personal data from LinkedIn. LinkedIn uses the OAuth 1.0a protocol for authentication. So when user clicks Grant access, a redirection message appears and the **auth.php** file is called in order to start the Oauth process [21].

The OAuth flow is described by LinkedIn official website as follows:

- 1) An application requests a set of temporary credentials, also known as **request token**. At this point this credentials aren't associated with a specific LinkedIn user
- 2) The application redirects the user to a login dialog where he authorizes these temporary credentials to be associated with his LinkedIn account
- 3) The application upgrades the temporary request token for permanent credentials, also known as **access token**. These credentials are necessary to give the application access to the LinkedIn APIs and make calls on behalf of the LinkedIn user [21].

These steps are done through the auth.php file, presented below.

### Initialization

```
$config['base_url']      = 'http://analyze-me.com/auth.php';
$config['callback_url']  = 'http://analyze-me.com/getlinkedin.php';
$config['linkedin_access'] = 'hMWnKT760YTsQmmkNEhYQSPoA6y7UMKz2Y8EumXlgrDoJLCQuz18fWkLjxHNPOq_';
$config['linkedin_secret'] = '6XGQh1M1jUVaCRShx8qYrMMJTXnOga74gp0k2hsSu-DbHQjPm-iCEFAV6xrpD3Hc';
```

```
include_once "linkedin.php"; //the file containing all classes needed for the OAuth process
```

**First step is to initialize with our consumer key and secret.**

```
$linkedin = new LinkedIn($config['linkedin_access'], $config['linkedin_secret'],
$config['callback_url'] );
```

**Now we retrieve a request token. It will be set as \$linkedin->request\_token**

```
$linkedin->getRequestToken();
$_SESSION['requestToken'] = serialize($linkedin->request_token);
```

**With a request token, we can generate an authorization URL, which we'll direct the user to**

```
header("Location: " . $linkedin->generateAuthorizeUrl(),false);
```

The rest of the auth.php code can be found in the appendix section.

When the user completes the authentication, he is redirected back to the results page. Now he needs to manually enter the number of his first, second, and third degree connections in the three input boxes and press save.

After he/she saves the input values, the following actions take place:

- 1) Check if user exists in database and create new entry in LinkedIn table if he doesn't
- 2) Calculate results based on user's connections of all degrees
- 3) Show user description and relating picture based on score

Let's further describe what happens during these actions.

- 1) If the table doesn't exist, makes a new entry in the LinkedIn table containing initial values for all fields.

```
$n = $_POST['id'];
mysql_select_db('perler2_ei') or die (mysql_error());
$res = mysql_query("SELECT * from linkedin WHERE id='$n'");
$row = mysql_fetch_assoc($res);
if ($n != $row['id'])mysql_query("INSERT INTO linkedin VALUES
('$n','$first','$second','$third')");
```

- 2) Below the input fields the results appear in the format as shown in Figure 33 below, where users can see their score analysis:

<b>your friends are</b>	<b>58</b>	people
<b>each of your friends know</b>	<b>72.41</b>	people, in average
<b>each of your friends knows</b>	<b>1.25</b>	times as much people as you know
<b>your friends-of-friends are (FF)</b>	<b>4200</b>	people (that you can access in 2-hops)
<b>each of your friend-of-friend knows</b>	<b>89.71</b>	people, in average
<b>each of your friend-of-friend knows</b>	<b>1.55</b>	times as much people as you know
<b>your fr.-of-fr.-of friends are (FFF)</b>	<b>376800</b>	people (that you can access in 3-hops)
<b>FF strength</b>	<b>114.85</b>	%
<b>FFF Strength</b>	<b>144.68</b>	%
<b>Total Score</b>	<b>1.87</b>	<b>0%</b>

*You are either a new user or a person that only uses his account rarely, only checking for updates upon being notified by email.*

Figure 33: LinkedIn results page explained

Where the values result from the following formulas:

**Ratio1**= 1<sup>st</sup> degree connections

**Ratio2**=2<sup>nd</sup> degree connections/1<sup>st</sup> degree connections

**Ratio3**=3<sup>rd</sup> degree connections/2<sup>nd</sup> degree connections

**Your friends are:** Ratio1

**Each of your friend know:** Ratio2

**Each of your friend knows % people as much as you:** Ratio2/Ratio1

**Your friends of friends are:** 2<sup>nd</sup> degree connections

**Each of your friend-of-friends knows:** Ratio3

**Each of your friend-of-friends knows % people as much as you know:** Ratio3/Ratio1

**Your friends of friends of friends are:** 3<sup>rd</sup> degree connections

**The final score is a result of the following formula:**

**Final score = ((first - \$n2) / 100) + ((\$ratio2 / \$ratio1) - \$n4) + ((\$second-\$n6) / 10000) + ((((\$ratio3 / \$ratio1) - \$n8)) + ((\$third - \$n10) / 1000000)**

Where n2, n4, n6, n8, n10 are base values.

N2=100, n4=0.1, n6=10000, n8=0.1, n10=100000

First, second, third = connections of 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> degree respectively

- 3) Depending on the final score, a short text message is displayed along with a picture categorizing the user based on his score. The following code is responsible for this:

```
$img="starter.jpg";
$desc="temporary text";
$criteria=0.00;
//sets the criteria to the final score value
$criteria=round((((($first-$n2)/100)+(($ratio2/$ratio1)-$n4)+(($second-$n6)/10000)+(($ratio3/$ratio1)-$n8)+(($third-$n10)/1000000),2);

if ($criteria<10)
{
    $img="starter.jpg";
    $desc="You are either a new user or a person that only uses his account rarely, only checking for updates upon being notified by email.";
}
else if ($criteria>=10 && $criteria<20)
{
    $img="casual.jpg";
    $desc="You are a casual linkedIN user, you don't post updates very frequently but you have an increased number of connections, you use your profile mainly for following updates from your connections.";
}
else if ($criteria>=20 && $criteria<40)
{
    $img="advanced.jpg";
    $desc="You are an advanced linkedIN user, you use your account frequently either posting or checking updates. You have a good position within the linkedIN society, with many connections following your updates.";
}
else if ($criteria>=40)
{
    $img="professional.jpg";
    $desc="You are a linkedIN power user, you use your account daily, posting and reading updates very frequently. You use linkedIN as your main social networking platform, and you have made a strong reputation among your connections.";
}
```



## 6. Real Analyze-me Data and Selection of Prediction Algorithm

Having developed the Analyze-me application we were able to collect some real data. In this chapter the real time-series data will be presented. To these data we will apply the prediction algorithms of the MATLAB Least-Squares Regression - Polynomial Curve Fitting (see section 2.1.5), the MATLAB Box-Jenkins Methodology (see section 2.2.17), and the MATLAB – Autoregressive Identification with Kalman filter (see section 2.3.3). The performance results of these three prediction algorithms will be evaluated and assessed in order to select the most appropriate one to be implemented in the Analyze-me application.

The Neural networks Forecasting model (see section 2.4) will not be applied to the data set since this model usually requires several thousands of iterations to be trained on a fairly large amount of historical data of users' behavior.

### 6.1 Real Analyze-me Data

The real Analyze-me data consists of 66 SCORE items (dependent variable y) taken every day from 28-04-2012 till 04-09-2012 (independent variable x). These data items are shown in the following table.

x	1	2	3	4	5	6	7	8	9	10
SCORE	47.63	47.63	47.12	46.93	46.93	46.78	46.61	46.55	46.22	46.04
x	11	12	13	14	15	16	17	18	19	20
SCORE	45.83	45.57	45.57	45.37	44.94	44.28	44	44	43.35	43.16
x	21	22	23	24	25	26	27	28	29	30
SCORE	47.46	47.36	47.12	46.58	46.15	45.66	47.9	46.78	46.56	46.35
x	31	32	33	34	35	36	37	38	39	40
SCORE	46.68	46.68	46.48	46.22	48.78	48.78	48.74	48.75	48.59	48.69
x	41	42	43	44	45	46	47	48	49	50
SCORE	48.56	48.31	48.31	48.24	47.47	47.47	47.14	48.25	49.1	49.78
x	51	52	53	54	55	56	57	58	59	60
SCORE	54.21	54.28	54.28	54.00	53.87	53.87	52.92	52.74	52.93	52.93
x	61	62	63	64	65	66				
SCORE	53.57	53.24	53.24	51.55	51.35	51.10				

### 6.2 Real Analyze-me Data: Least-Squares Regression-Polynomial Curve Fitting

To find the least-squares regression line polynomial algorithm to predict the next score, based on the real Analyze-me data, we use the procedure presented in the example problem in section 2.1.5. Thus the first step is to create a scatter plot of the data. This is done by executing the following MATLAB code.

```
clc; clear all; close all; % clear command window, workspace & figures
x = 1:66; % Real Analyze-me data values of independent variable x
SCORE = [47.63 47.63 47.12 46.93 46.93 46.78 46.61 46.55 46.22 46.04 45.83 45.57 45.57 45.37
44.94 44.28 44.00 44.00 43.35 43.16 47.46 47.36 47.12 46.58 46.15 45.66 47.90 46.78 46.56
```

```

46.35 46.68 46.68 46.48 46.22 48.78 48.78 48.74 48.75 48.59 48.69 48.56 48.31 48.31 48.24
47.47 47.47 47.14 48.25 49.10 49.78 54.21 54.28 54.28 54.00 53.87 53.87 52.92 52.74 52.93
52.93 53.57 53.24 53.24 51.55 51.35 51.10]; % values of data set dependent variable y
figure(1)
plot(x, SCORE, 'o') % scatter plot of data set variables y vs. x
axis([0 67 min(SCORE)-5 max(SCORE)+5]); % range of plotting axes
title('Scatter plot of Real Analyze-me data values')
% grid on      % insert grids in the plot

```

The execution of the above MATLAB code gives the following scatter plot in Figure 34.

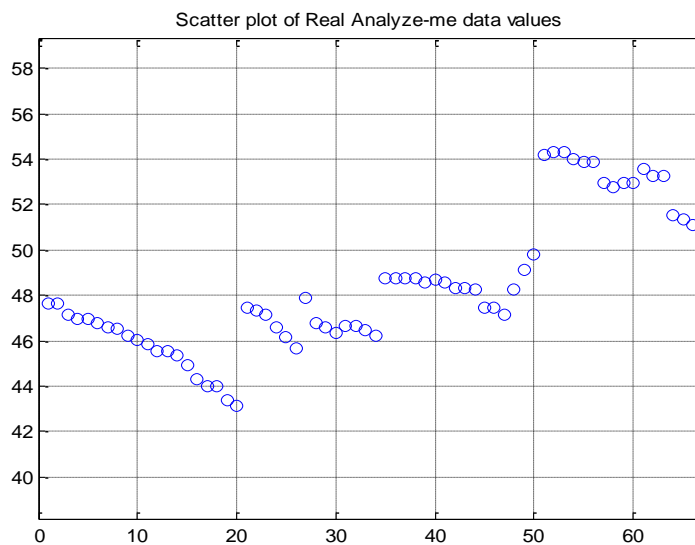


Figure 34: Scatter Plot of RealAnalyze-me data values

Form the scatter plot figure of the real Analyze-me data points  $(x_i, SCORE_i)$  it can deduce that there is no clear pattern which the 66 SCOREs follow from day-to-day. Specifically, for the first 20 SCOREs someone might be tempting to model them by a straight line with negative slope (downward trend). The same can be deduced for the next 6 SCOREs though with different slope due to the SCORE jump at  $x=21$ . The next few SCORE values around  $x=30$  besides the jump from the previous behavior indicate a slower down trend from the previous two. In addition, for the next SCORE values from  $x=47$  to  $x=50$  there is an upward trend followed by a jump at  $x=50$  and down trend thereafter. From all these observations of downward and upward trends and the downward and upward jumps in the scatter plot of the real Analyze-me data points  $(x_i, SCORE_i)$  it is concluded that a least-squares regression - polynomial curve fitting is not a good candidate as a predictor algorithm. Therefore it is not implemented in the Analyze-me application.

### 6.3 Real Analyze-me Data: Box-Jenkins Methodology

To use the Box-Jenkins methodology to find an algorithm to predict the next score, based on the real Analyze-me data, we modify slightly the procedure presented in the example problem in section 2.2.17. The modified MATLAB code to accommodate the real Analyze-me data and find an ARIMA model is the following:

```

clc; clear all; close all; % clears command window, workspace and figures
% Step 1. Provide the Real Analyze-me data values of the independent
% variable x and plot it.
Y = [47.63 47.63 47.12 46.93 46.93 46.78 46.61 46.55 46.22 46.04 45.83 45.57 45.57 45.37 44.94
44.28 44.00 44.00 43.35 43.16 47.46 47.36 47.12 46.58 46.15 45.66 47.90 46.78 46.56 46.35
46.68 46.68 46.48 46.22 48.78 48.78 48.74 48.75 48.59 48.69 48.56 48.31 48.31 48.24 47.47
47.47 47.14 48.25 49.10 49.78 54.21 54.28 54.28 54.00 53.87 53.87 52.92 52.74 52.93 52.93
53.57 53.24 53.24 51.55 51.35 51.10]';
% Use 50 out of 66 data values to model an ARIMA predictor algorithm
SCORE = Y(1:50); N = length(SCORE);
figure(1), plot(SCORE), grid on; xlim([0,N])
% set(gca,'XTick',1:10:N);
% set(gca,'XTickLabel',datestr(dates(1:10:N),17));
title('Real Analyze-me data values')
% Step 2. Plot the sample ACF and PACF.
figure(2), subplot(2,1,1); autocorr(SCORE); subplot(2,1,2); parcorr(SCORE);
% Step 3. Difference the data.
dSCORE = diff(SCORE);
figure(3), plot(dSCORE); grid on; xlim([0,N]);
% set(gca,'XTick',1:10:N);
% set(gca,'XTickLabel',datestr(dates(2:10:N),17));
title('Differenced Real Analyze-me data values')
% Step 4. Plot the sample ACF and PACF
% of the differenced series.
figure(4), subplot(2,1,1); autocorr(dSCORE); subplot(2,1,2); parcorr(dSCORE);
% Step 5. Specify and fit
% an ARIMA(2,1,0) model.
model = arima(2,1,0); fit = estimate(model,SCORE);
% Step 6. Check goodness of fit.
res = infer(fit,SCORE);
figure(5), subplot(2,2,1); plot(res./sqrt(fit.Variance)); grid on; title('Standardized Residuals')
subplot(2,2,2); qqplot(res); grid on; subplot(2,2,3); autocorr(res); grid on;
subplot(2,2,4); parcorr(res); grid on;
% Step 7. Generate forecasts.
[SCOREf,SCOREMSE] = forecast(fit,16,'Y0',SCORE);
UB = SCOREf + 1.96*sqrt(SCOREMSE); LB = SCOREf - 1.96*sqrt(SCOREMSE);
figure(6), h1 = plot(Y,'Color',[.15,.15,.15]); hold on;
h2 = plot(51:66,SCOREf,'r','LineWidth',2); h3 = plot(51:66,UB,'k--','LineWidth',1.5);
plot(51:66,LB,'k--','LineWidth',1.5);
% set(gca,'XTick',1:10:N);
% set(gca,'XTickLabel',datestr(dates(1:10:N),17));
legend([h1,h2,h3],'Real Analyze-me data','Forecast',...
'Forecast Interval','Location','Northwest')
title('Real Analyze-me data Forecast'); grid on;
error = Y(51:66) - SCOREf;
figure(7)
plot(error), grid on; title('Error of ARIMA(2,1,0) Predictor')

```

The above code uses only 50 out of 66 data values to generate the ARIMA predictor. The last 16 values are used to check the performance of this predictor.

The execution of the above MATLAB code gives seven (7) figures. These figures will be presented one-by-one and some observations/comment for each will be provide as follows:

Figure 35 presents a plot of 50 real Analyze-me data values taken on a daily basis.

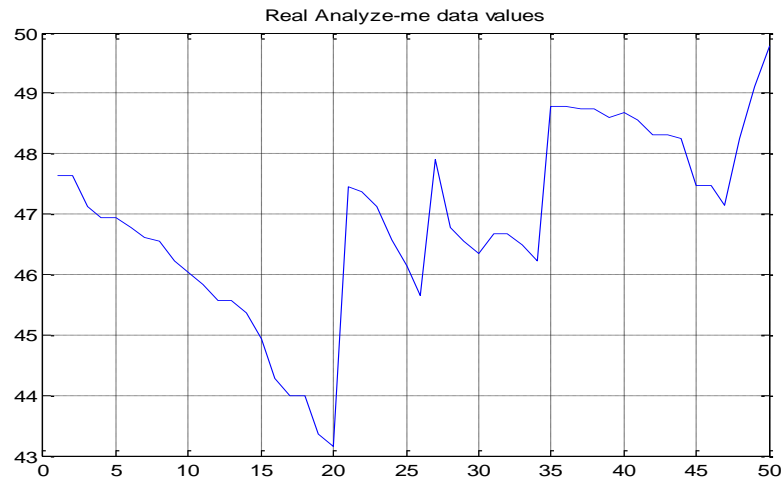


Figure 35: Real Analyze-me data values (SCOREs)

From the plot clearly it can be seen that the time series data set of SCORE values is non-stationary. Initially, the plot indicates a downward trend, then a jump follows at  $x=21$ , and then another downward trend until  $x=27$ . Then a second jump appears at  $x=28$  followed again with a third downward trend until  $x=34$  with different slope than the previous two downward trends. At  $x=35$  another jump appears again followed by a different than the previous three downward trends. At  $x=47$  we see another bigger than the previous jumps followed by an upward trend.

The next Figure 36 shows a plot of the sample autocorrelation function (ACF) and partial autocorrelation function (PACF) for the real Analyze-me data values.

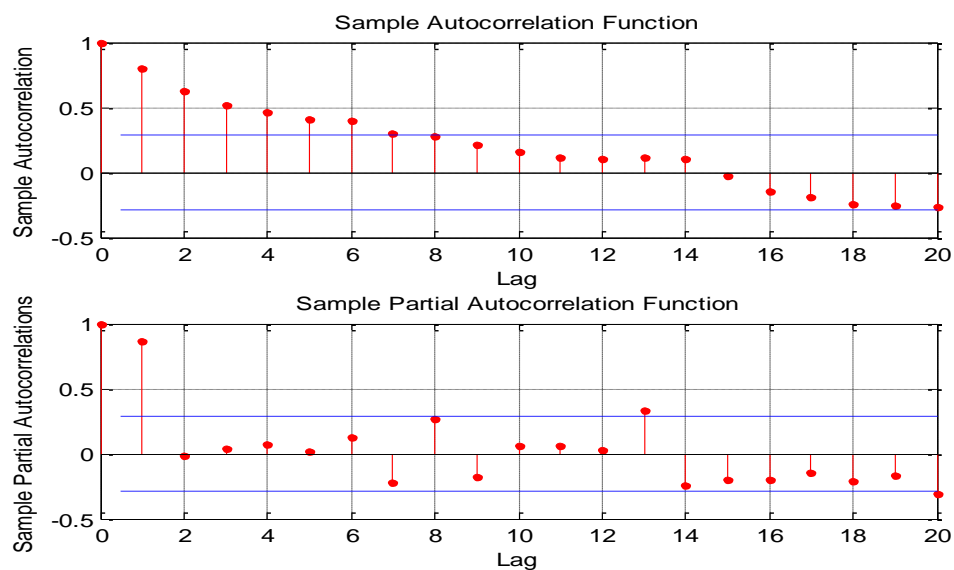


Figure 36: Plot of sample ACF and PACF of real Analyze-me data values (SCOREs)

Figure 36 shows a significant linear decay of the sample ACF indicating a nonstationary process.

The next Figure 37 shows a plot of the differenced real Analyze-me data values to remove the upward and downward trends from the original data set.

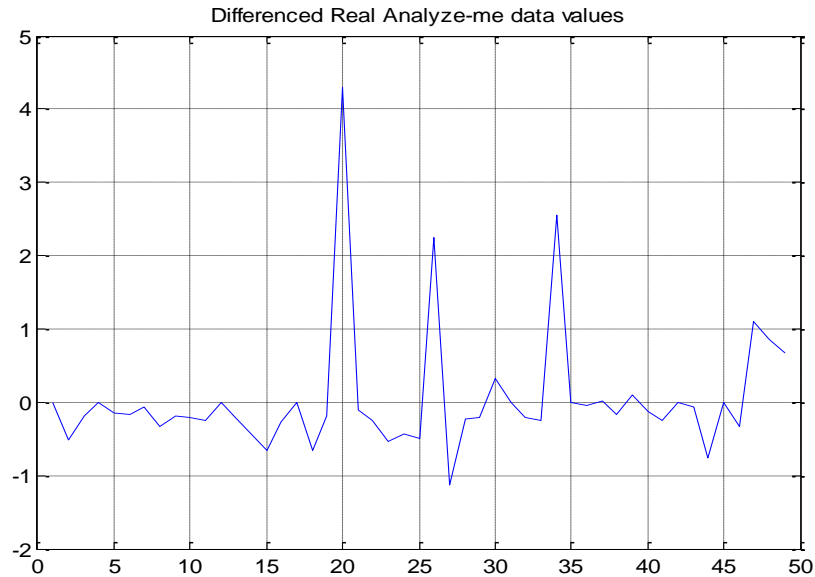


Figure 37: Plot of Differenced real Analyze-me data values (SCORES)

From Figure 37 we can see that the differencing process removed the upward and downward trends from the original data values and also that the differenced series appears more stationary.

The next Figure 38 shows the sample autocorrelation function (ACF) and partial autocorrelation function (PACF) of the differenced real Analyze-me data values (SCORES) and plots them.

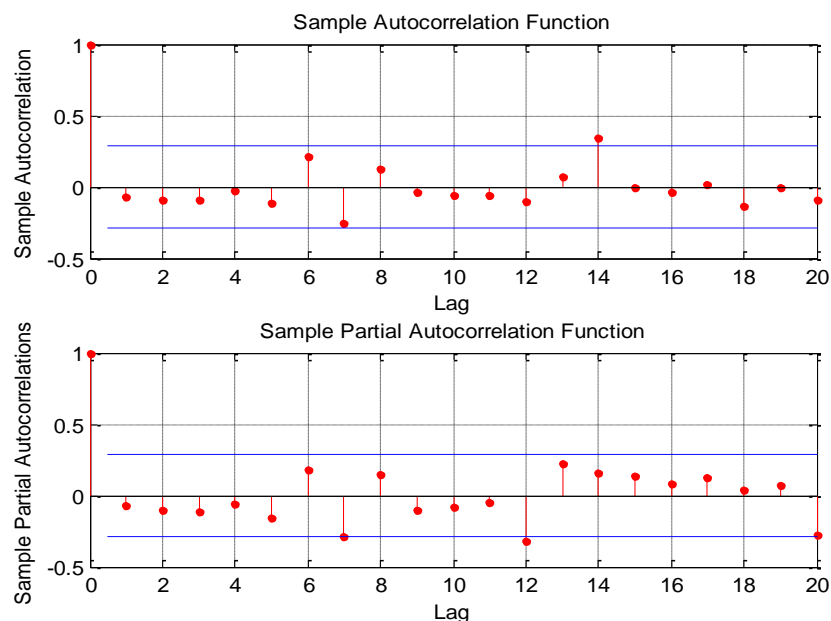


Figure 38: Plot of sample ACF and PACF of Differenced real Analyze-me data values (SCORES)

Now from Figure 38 we can see that the behavior of the differenced real Analyze-me data values (SCOREs) looks more consistent with a stationary process.

The ARIMA(2,1,0) generated model is shown next:

ARIMA(2,1,0) Model:

-----  
 Conditional Probability Distribution: Gaussian

Parameter	Value	Standard Error	t Statistic
Constant	0.0441265	0.240554	0.183437
AR{1}	-0.0763552	0.262095	-0.291327
AR{2}	-0.101104	0.46616	-0.216887
Variance	0.736578	0.162441	4.53445

The following Figure 39 shows a plot of residuals for goodness of ARIMA(2,1,0) model fit.

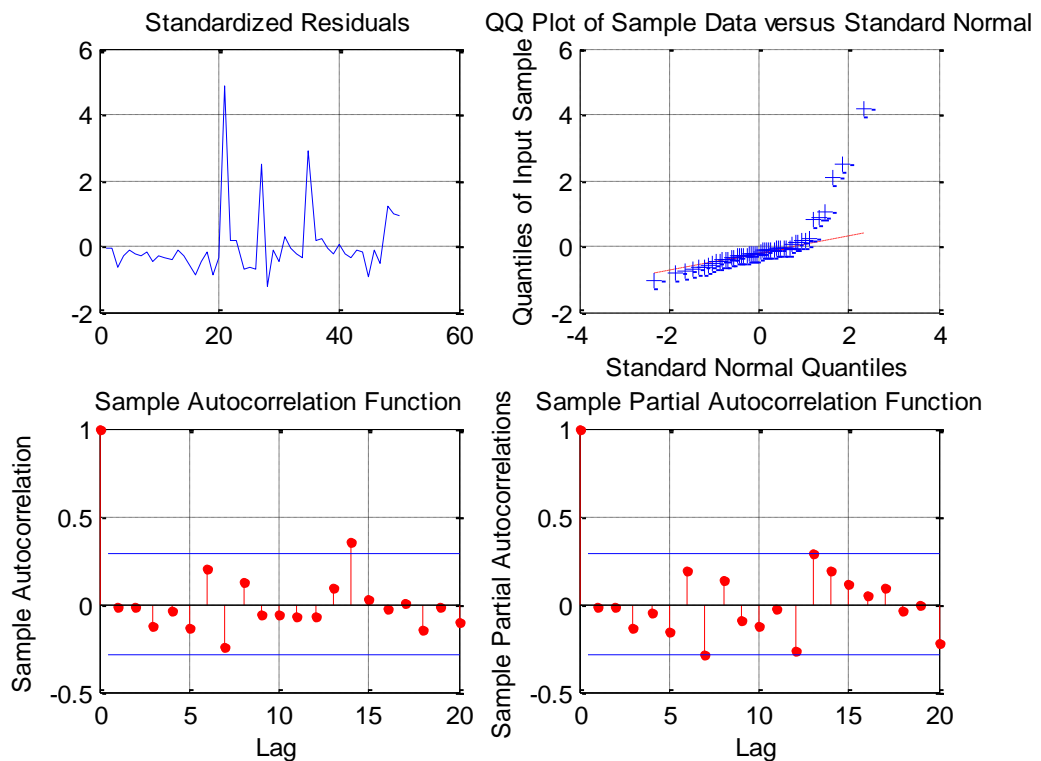


Figure 39: Plot of residuals for goodness of ARIMA(2,1,0) model fit

From Figure 39 it is inferred that the residuals are reasonably normally distributed and uncorrelated.

The following Figure 40 shows the next 16 forecasts along with the approximate 95% forecast intervals (upper and lower black dotted lines).

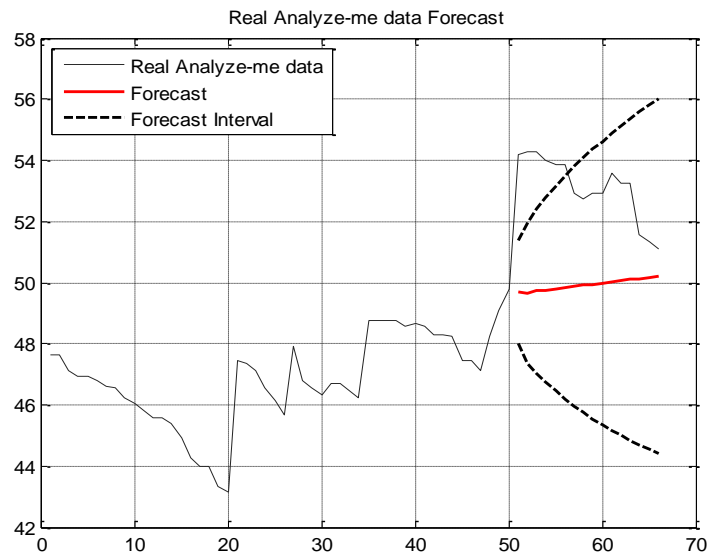


Figure 40: ARIMA(2,1,0) Forecasts for the next 16 data values (SCOREs)

From the above Figure 40 it can be seen that the ARIMA(2,1,0) model predicts (red line) almost a slow upwards trend for the next 16 values. Comparing the predicted values with the actual corresponding (black line above the red line) 16 real Analyze-me data values (SCOREs) it can be seen that the prediction values are not in close agreement with the corresponding measured SCOREs.

This error is fairly large, as shown in the plot of the next Figure 41, ranging almost between 1 and 4.5.

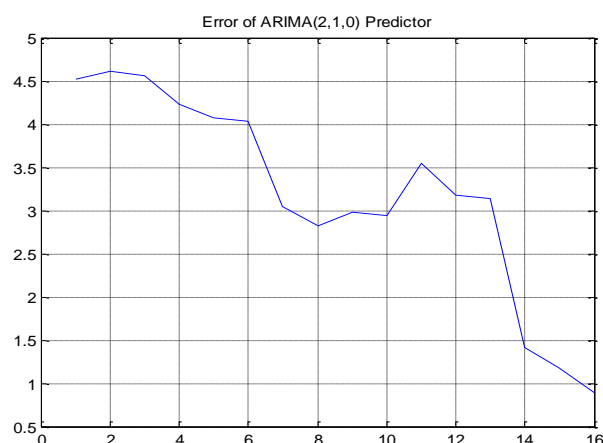


Figure 41: Error of ARIMA(2,1,0) Predictor

Different ARIMA models were created by changing the amount of data values to be used and processed by the Box-Jenkins methodology. The performance results were similar with relatively large errors. Also, from all experiments it was noticed that the predictions do not follow the trend of the real Analyze-me SCOREs. This indicates that the Box-Jenkins methodology is not well suited to be used and find predictors for this type of problems which depend on human behavior. Also, the Box-Jenkins methodology requires the use of a large data set, usually in the order of 200 and more. The large data set though captures the behavior of the users long in the

past and this past behavior not necessarily mean that it will be retained in the near future. Thus these results prompted the need for the investigation of another method to be implemented in the Analyze-me application.

#### 6.4 Real Analyze-me Data: Autoregressive Identification with Kalman filter

To use the autoregressive identification with Kalman filter methodology to find an algorithm to predict the next score, based on the real Analyze-me data, we modify slightly the procedure presented in the example problem in section 2.3.3. The modified MATLAB code to accommodate the real Analyze-me data and find a Kalman filter model is the following:

```
clear all;close all; clc;
z = [47.63 47.63 47.12 46.93 46.93 46.78 46.61 46.55 46.22 ...
     46.04 45.83 45.57 45.57 45.37 44.94 44.28 44.00 44.00 43.35 43.16 47.46 47.36 47.12 46.58
     46.15 45.66 47.90 46.78 46.56 46.35 46.68 46.68 46.48 46.22 48.78 48.78 48.74 48.75 48.59
     48.69 48.56 48.31 48.31 48.24 47.47 47.47 47.14 48.25 49.10 49.78 54.21 54.28 54.28 54.00
     53.87 53.87 52.92 52.74 52.93 52.93 53.57 53.24 53.24 51.55 51.35 51.10]; % Measurements
% Kalman Filter estimation of curve fitting coefficients x for the
% estimation of  $y(k) = -[y(k-1)x_1+y(k-2)x_2+\dots+y(k-n)x_n] \Rightarrow$ 
%  $y(k) = H(k)x(k) = z(k) \Rightarrow$  Measurements  $z(k) = y(k)$ 
%  $x(k+1) = x(k) + w(k)$ ; Linear Model of curve fitting coefficients
%  $z(k) = H(k)x(k) + v(k)$ ; Model of Measurements
% -----
% Initialization of Kalman Filter Parameters
n = 3; % Kalman Filter Window size
Q_d = 0.1*eye(n,n); % Noise Covariance for the model
% G_d = eye(n,n); % Noise Distribution for the model
x_plus = -0.35*z(1:n)'; % Initial conditions for the state estimate
P_plus = 1.0*eye(n,n); % Initial conditions for the state Covariance
x_est = []; % Storage variable vector for fitting coefficients x
y_est = []; % Storage variable vector for estimates
errors = []; % Storage variable vector for estimates  $z(k)-y\_est$ 
% -----
% Beginning of Kalman Filter Loop
for k = n+1:length(z)
    H = -[z(k-n:k-1)]; % Window size data matrix H
    x_minus = x_plus; % Propagation of state estimate x
    P_minus = P_plus+Q_d; % Propagation of covariance P
    y(k) = H*x_minus;
    x_est = [x_est x_minus]; % Storage variable holding estimates x_minus
    y_est = [y_est y(k)]; % Storage variable holding estimates y_est
    % Update equations
    R = std(H); % Covariance of Matrix H
    K = (P_minus*H')*((H*P_minus*H'+R)^(-1));
    residual = z(k) - y(k);
    errors = [errors residual]; % Variable holding errors  $z-y\_est$ 
    x_plus = x_minus+K*residual; % Update of state x
    P_plus = P_minus-K*H*P_minus; % Update of covariance P
end % End of Kalman Filter Loop
% -----
```



```

% Last Kalman Filter estimation
H_last = -[z(k-n+1:k)]; x_last = x_plus; P_last = P_plus+Q_d; y_last = H_last*x_last;
x_est = [x_est x_last]; % Storage variable holding estimates x_minus
y_est = [y_est y_last]; % Variable holding estimate y_est and y_last
errors;
figure(1)
plot((1:length(z)), z, 'b'); grid on; hold on;
plot([n+1:length(z)+1], y_est, 'm');
legend('Real Analyze-me SCOREs', 'Predictions', 'Location', 'NorthWest');
title('Real Analyze-me data and Kalman Filter Predictions'); axis([0 70 min(z)-5 max(z)+5]);
figure(2)
plot([n+1:length(z)], [errors], 'r'); title('Kalman Filter Prediction Errors'); axis([0 70 -5 +5]); grid on;

```

In the above MATLAB code we use for the window size Kalman filter parameter the value  $n=3$ . This indicates that we use 3 past SCOREs to predict the 4<sup>th</sup> one.

The execution of the above MATLAB code gives the next two (2) figures. Figure 42 presents a plot of the real Analyze-me data values (blue line) and the Kalman filter predictions (magenta line).

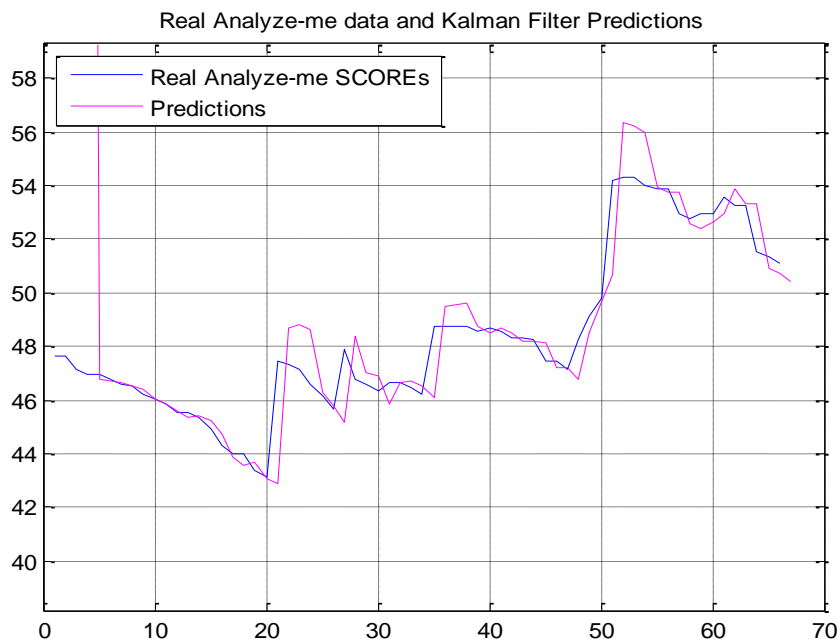


Figure 42: Real Analyze-me data and Kalman Filter Predictions with window size  $n=3$

From the above plots in Figure 42 it can be observed that the Kalman filter predictions follow the real Analyze-me data SCOREs very close. In addition, these predictions follow the trend of the real data values.

The next Figure 43 presents a plot of the Kalman filter predictions error.

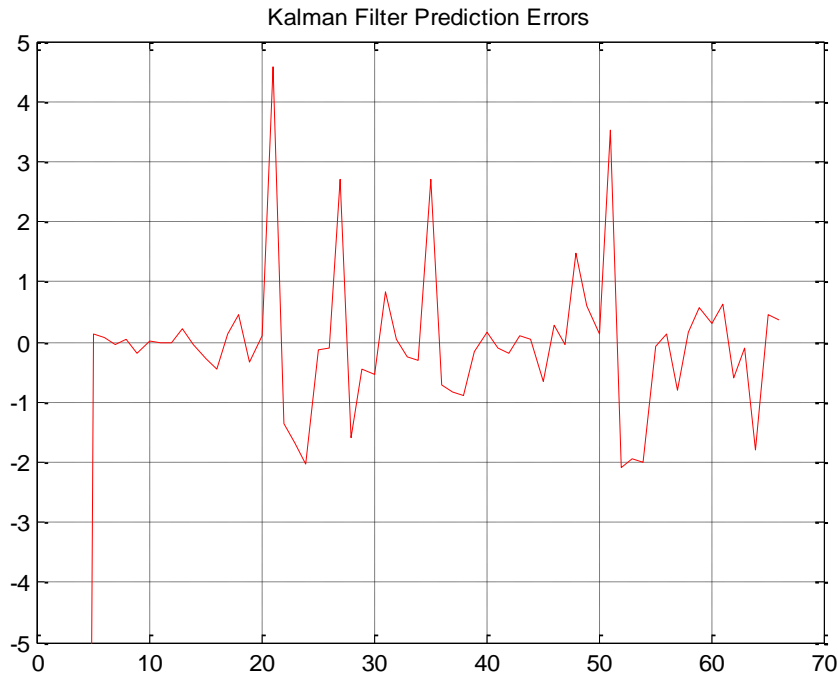


Figure 43: Kalman Filter Prediction Errors with window size  $n=3$

From Figure 43 it can be seen that the Kalman filter prediction errors with window size parameter  $n=3$  are relatively small. Most of them are within the range of  $+2.50$  and  $-2.00$ . There are though a couple of them with values of  $+4.50$  and  $+3.50$  at the jumps  $x=21$  and  $x=51$ , respectively.

Different experiments were run by changing the values of the parameter value  $n$  in the above MATLAB code. The best results were obtained with the window size Kalman filter parameter set to value  $n=3$ , i.e., using 3 SCORE values to make the prediction (the 4<sup>th</sup> one).

## 6.5 Comparative analysis - Prediction Algorithm Selection

In the previous sections using real Analyze-me data SCOREs the least-squares regression - polynomial curve fitting, the Box-Jenkins methodology, and the autoregressive identification with Kalman filter approaches have been considered as possible candidates for prediction algorithms to be implemented in the Analyze-me application. In addition, different comments and issues were presented concerning the performance results for each predictor candidate.

By comparative analysis of these comments, issues, and performance results it is clear that the most appropriate prediction algorithm based on the real Analyze-me data SCOREs is the Autoregressive Identification with Kalman filter. As a result this Autoregressive Identification with Kalman filter has been selected to be implemented in the Analyze-me application.

More specifically, the algorithm uses only a few past scores to predict an estimate of a user's future score unlike the other algorithms. The application Analyze-me saves the user score to the database every time the service is used. The advantage of the Autoregressive Identification with

Kalman filter prediction here is obvious in cases where the user has a small score history recorded.

The Kalman filtering algorithm is applied to provide a technique for the identification of the coefficients in an AutoRegressive type equation containing the score measurements over time. The aim is to estimate the values of the equation's coefficients using a few of the real measurements.

The algorithm, as an example of how it works using the last 6 scores, creates a set of the first 3 values to predict with it the next 4<sup>th</sup> score value and compares the predicted result with the 4<sup>th</sup> real score. Then it drops the 1<sup>st</sup> score value and uses the 2<sup>th</sup>, 3<sup>th</sup>, and 4<sup>th</sup> score values to generate the next set of 3 values to predict the 5<sup>th</sup>, and compares this prediction to the real 5<sup>th</sup> score. Next, uses 3<sup>th</sup>, 4<sup>th</sup>, and 5<sup>th</sup> values to predict the 6<sup>th</sup>, and compare it to the real 6<sup>th</sup> score. In the process these comparisons are used to determine the so called prediction Accuracy. Finally, uses 4<sup>th</sup>, 5<sup>th</sup> and 6<sup>th</sup> to predict the 7<sup>th</sup> future score. The prediction Accuracy does not represent the accuracy of the predicted future score, instead it represents the percentage of the algorithm predicting correctly during the comparison phase.

Thus, in order to estimate a future score, the algorithm needs 3 past scores. So to predict tomorrow's score it needs today's score, yesterday's and the day before yesterday's score. Another example, for weekly predictions, would be next week's score, where the algorithm would need, today's score, previous week's score, and the user's score 2 weeks in the past.

## 6.6 Description of the program used to develop the Prediction Algorithm

MATLAB was used to evaluate the predictive algorithms and help to decide which one to use for the prediction feature of the Analyze-me.com application.

Matlab is a high-performance language for technical computing, offering math and computation, algorithm development, modeling, simulation, data analysis, scientific and engineering graphics all in an easy to use environment (<http://www.mathworks.com>).

Below in Figure 44 is a sample graph using matlab:

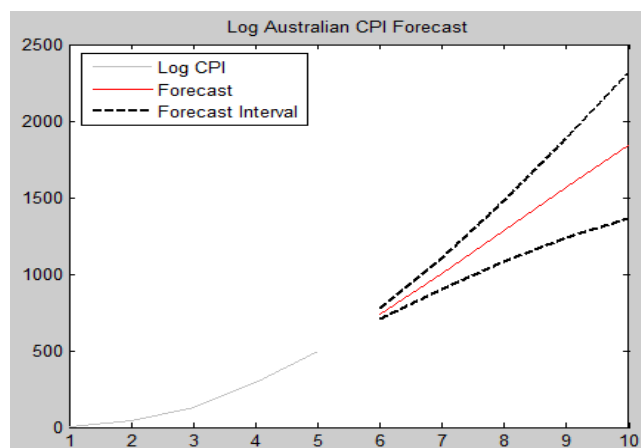


Figure 44: Matlab sample plot

The Matlab working environment looks as in see Figure 45 below:

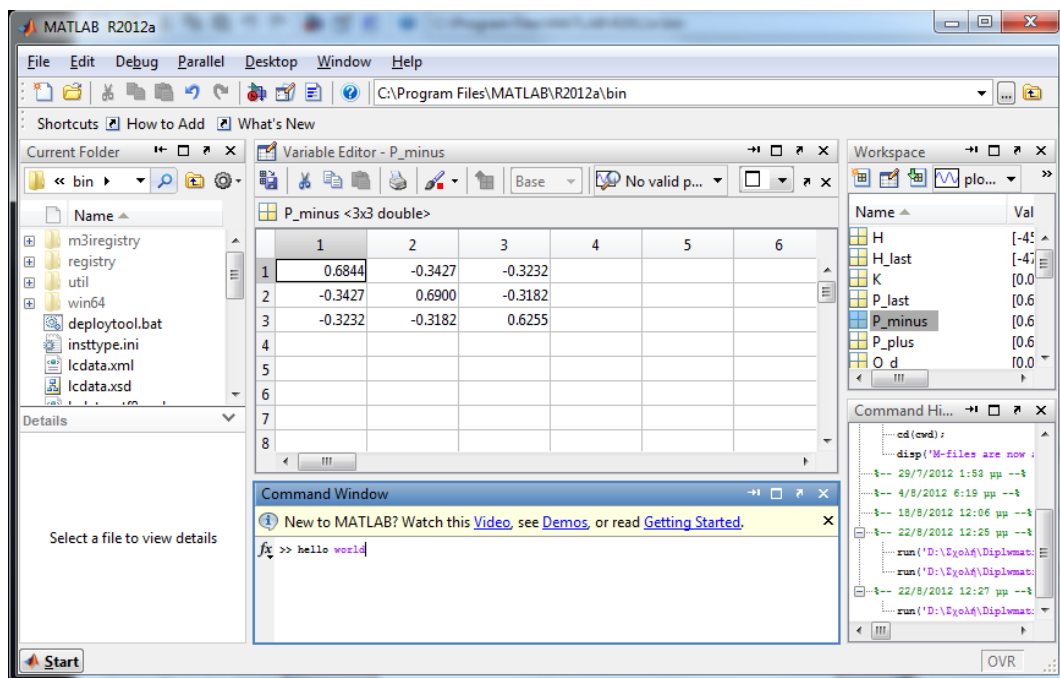


Figure 45: Matlab workspace

## 7. Use of Analyze-me Application

AnalyzeMe is a social network analyzer. It uses KLOUT, LinkedIn, Twitter APIs to obtain information about a user and analyzes them to provide a rating based on how social the user is. The analysis is based on how many followers, friends, tweets, actions the user has.

Opening the website analyze-me.com loads the main page that looks as in Figure 46 below:



Figure 46: Analyze-me home page

The Analyze-me logo, the start here button and the popularity metrics logo were made in Adobe Photoshop CS5. The main menu bar is located on the top of the page so it's easy to navigate through pages no matter which page is currently loaded. Menu options are explained below.

### 7.1 Home

The home button returns to the homepage when clicked. It is used to avoid typing analyze-me.com again to go to the starting page.

### 7.2 Features

The following Figure 47 gives a brief explanation of what the website has to offer:

## What do we offer?

We provide an overall score and type characterisation for the user being analyzed, representing the level of his social activity and ways of influence. Additional score forecasting.



-Using klout, Twitter, LinkedIn APIs to obtain user data.

-Providing detailed graphs

-Providing detailed user status analysis


-Mobile implementation under development 

Figure 47: Analyze-me Features page

## 7.3 About us

Contains information and contact details of the author (Figure 48):

### Authors

Perry Leros contact: [perryleros@gmail.com](mailto:perryleros@gmail.com)

University of Aegean 2011



Figure 48: Analyze-me About us page

## 7.4 Mobile App

This section contains screen shots of the mobile application for android, as well as download links on the bottom (Figure 49):

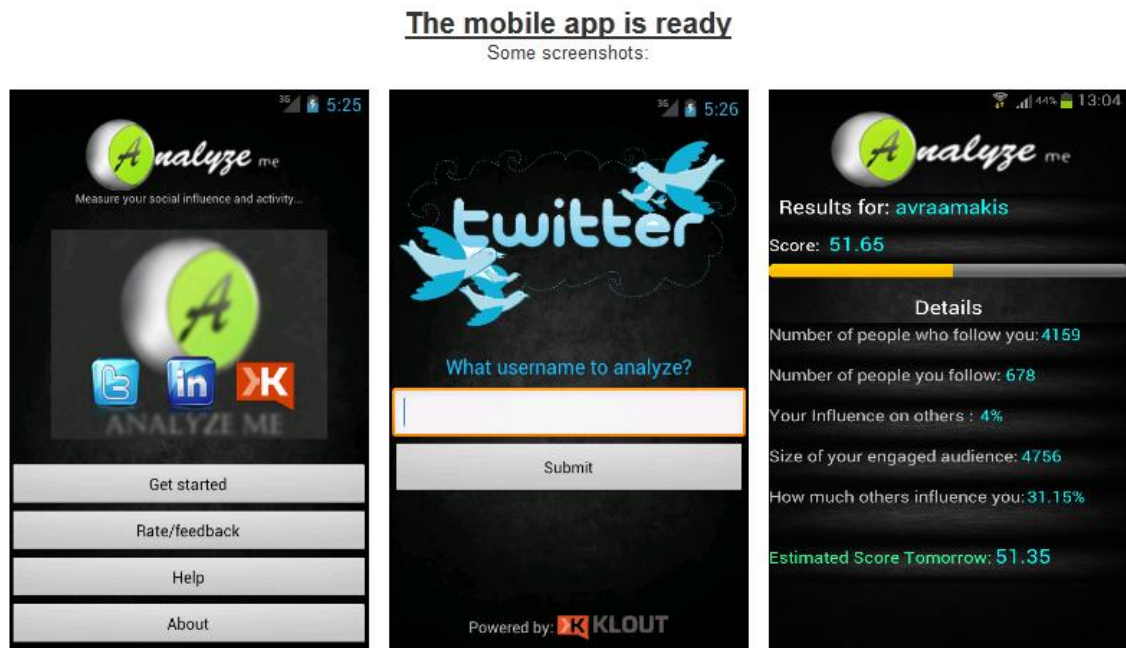
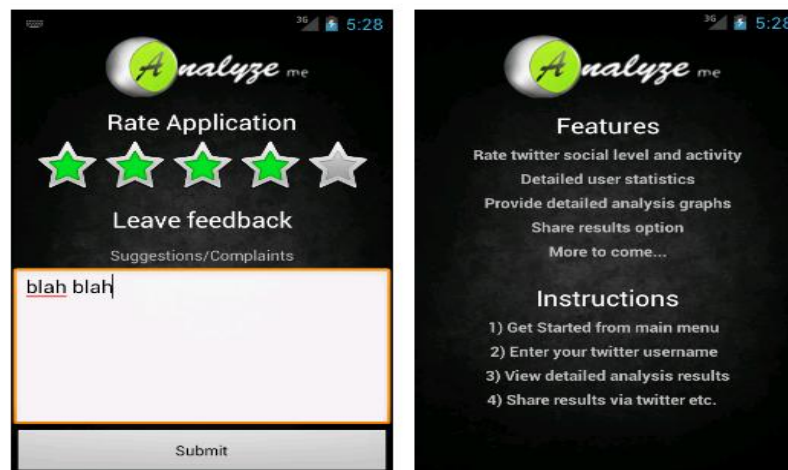


Figure 49: Analyze-me mobile app page

In the screenshots we can see a preview of every screen of the mobile application, as shown in the following Figure 50:



Download Links:

<http://www15.zippyshare.com/v/26964369/file.html>

or

<https://rapidshare.com/files/1405842639/AnalyzeMe.apk>

Figure 50: Analyze-me mobile app page

## 7.5 Get Started

This is the main action page, that can be accessed either by the “Get Started” option from the main menu or by the “Start Here” button on the home page.

The user has two options now; either to proceed with Twitter analysis or LinkedIn, by selecting the correspondent logo.

### 7.5.1 Twitter Analysis

By selecting Twitter, the user is taken to the next page where he is asked for his or any other user’s twitter username. By entering a correct username, he proceeds to the results page.

In case the username does not exist, an error message appears like the one below:

“User Not Accessible, Profile locked, Invalid username, or New account. Please go back and retry”

Moving on to the results page, the user gets to see the following (Figure 51):

#### User Score based on Twitter:



User Name: [perryleros](#)

Twitter Name: [Perry Leros](#)

Followers: 76

Friends: 132

**Score: [53.57](#)**

You are a/an: [Specialist](#)

*You may not be a celebrity, but within your area of expertise your opinion is second to none. Your content is likely focused around a specific topic or industry with a focused, highly-engaged audience.*

Figure 51: Twitter analysis results

On the top of the results page, basic user information is displayed, as well as his twitter profile picture (Twitter name, username, followers and friends).

Below the profile picture the main score appears, following by a description of the user’s classification based on the frequency and effect of his actions.

On the bottom of the page there is a graph showing the user’s score progress over the past 5 days. It’s a simple plot showing the user’s score values 5 days ago, yesterday and today (see Figure 52).



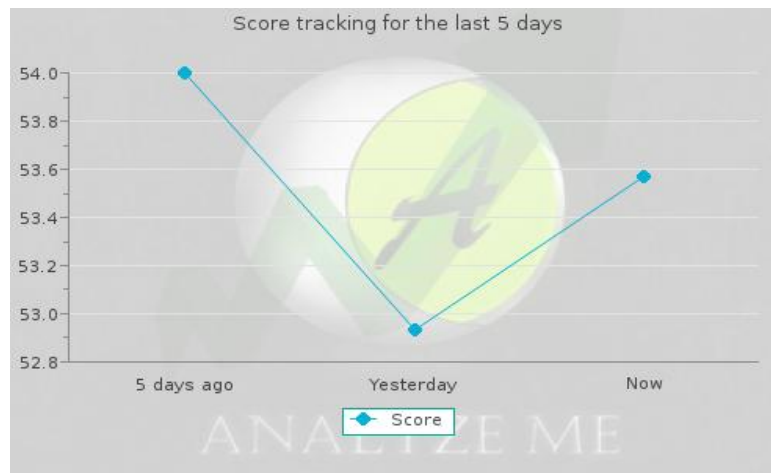


Figure 52: Twitter score graph

On the right hand side there is list containing further analysis (supplied by KLOUT API) as explained below:

**The likelihood that your messages will generate actions such as retweets, @messages, likes and comments:**

*Your Influence on others (%):*  
**33**

**Number of followers and friends who actively listen and react to your messages:**

*Size of your engaged audience:*  
**407**

**How influential is your engaged audience in %:**

*Influenced by your engaged audience (%):*  
**38.58**

Finally on the bottom right of the page, a new feature (see Figure 53) is available, "Prediction".



Figure 53: Prediction logo

The “Prediction” feature offers future score predictions based on past score tendencies. Clicking the prediction icon, the user is transferred to the last page (see Figure 54 below):

### **Future score prediction based on Kalman Filter**

Now: **53.57**

Tomorrow: **53.85**

Next Week: **55.09**

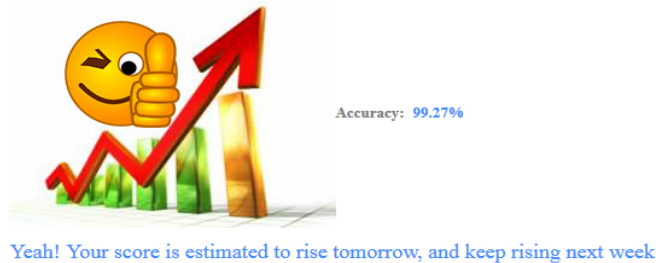


Figure 54: Prediction results page

This page displays the current user score, tomorrow’s estimation and Next week’s estimation. Also, an Accuracy evaluation is provided.

Below the results follows a brief explanation of the algorithm used to provide the estimations followed by some useful links for more details (see Figure 55 below).

#### **Kalman Autoregressive Identification method**

Kalman filtering is applied to provide a technique for the identification of the coefficients in an AutoRegressive type equation containing the score measurements over time. The aim is to estimate the values of the equation's coefficients using these measurements.

**Accuracy:** The algorithm uses last 6 scores, uses the first 3 to predict the next one. Compares the predicted result with the 4th real score. Then uses 2th, 3th, 4th scores to predict the 5th, and compares it to the real 5th score. Next, uses 3th, 4th, 5th to predict the 6th, to compare it to the real 6th score. These comparisons are used to determine the prediction Accuracy. Finally, uses 4th, 5th and 6th to predict the future score. This accuracy does not represent the accuracy of the predicted future score, instead it represents the percentage of the algorithm predicting correctly during the comparison phase.

More info on: [Kalman Filter](#), [Autoregressive Model](#)

Figure 55: Prediction algorithm information

The result picture varies depending on the score’s behavior (rise, fall, no change) as seen in Figures 56, 57, and 58 below:



Figure 56: Declining score



Figure 57: Rising score



Figure 58: Same score

### 7.5.2 LinkedIn Analysis

By selecting LinkedIn at the “Get Started” page, a page that requires granting access to the user’s LinkedIn account appears. After agreeing, the user is transferred to LinkedIn website to complete the authentication process and then is transferred back to Analyze-me results page, as shown in the following Figure 59.

#### User Analysis based on LinkedIN:

**ID:** V4Iat45HVJ

**Name:** Perry Leros



#### Manually input connections:

1st degree	2nd degree	3rd degree	
<input type="text" value="58"/>	<input type="text" value="4200"/>	<input type="text" value="376800"/>	<input type="button" value="save"/>

<b>your friends are</b>	<b>58</b>	people
<b>each of your friends know</b>	<b>72.41</b>	people, in average
<b>each of your friends knows</b>	<b>1.25</b>	times as much people as you know
<b>your friends-of-friends are (FF)</b>	<b>4200</b>	people (that you can access in 2-hops)
<b>each of your friend-of-friend knows</b>	<b>89.71</b>	people, in average
<b>each of your friend-of-friend knows</b>	<b>1.55</b>	times as much people as you know
<b>your fr.-of-fr.-of friends are (FFF)</b>	<b>376800</b>	people (that you can access in 3-hops)
<b>FF strength</b>	<b>114.85</b>	%
<b>FFF Strength</b>	<b>144.68</b>	%
<b>Total Score</b>	<b>1.87</b>	%

*You are either a new user or a person that only uses his account rarely, only checking for updates upon being notified by email.*

Figure 59: LinkedIn results page

Unlike twitter analysis, in this case the user has to manually enter the three required values; first, second, and third degree connections. Then, analysis based on the 3 parameters entered by the user is displayed. At the bottom of the page appears a short description of the user’s behavior on LinkedIn.

## 8. Android and Mobile Analyze-me Application

### 8.1 What is android

Android is a Linux based operating system developed by Google, designed for mobile devices such as smartphones, tablets or even laptops. Unlike Windows mobile and Apple IOS, Android is open source software meaning that every developer is free to alter the operating system in any way that suits his needs. For example, many different devices can have completely different graphical user interface although they are running the same version of android [22].

In Figures 60 and 61 below, we can see a typical android OS home screen, and a customized lock screen by HTC, respectively.



Today, android is the most widely spread development platform for mobile devices.

So the Android development platform was chosen to develop the Analyze-me mobile application.

The mobile application was developed in the Eclipse environment. Eclipse (see Figure 62) is a multi-language development environment mostly written in JAVA and can be used to develop programs in Java, C++, C, Cobol, and pretty much every known programming language. Using the ADT plugin, Eclipse is the ideal platform for developing android applications (<http://www.eclipse.org/>).

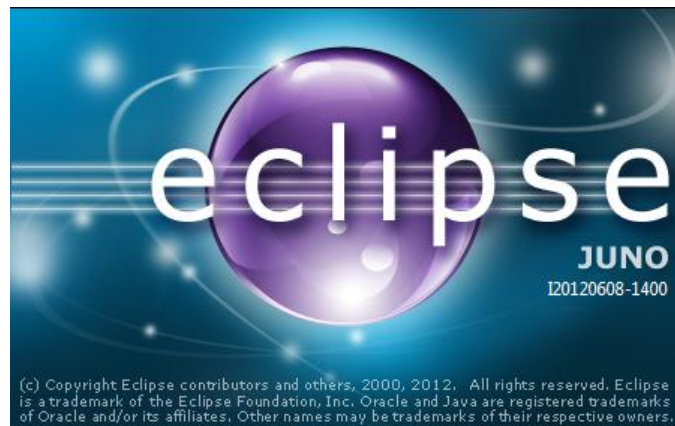


Figure 62: Eclipse logo

## 8.2 Mobile Analyze-me Application

Each page of the application is called a layout in android and it has a java file associated with it that contains actions. Let's start from the first layout, the home screen of the application seen in Figure 63 below:

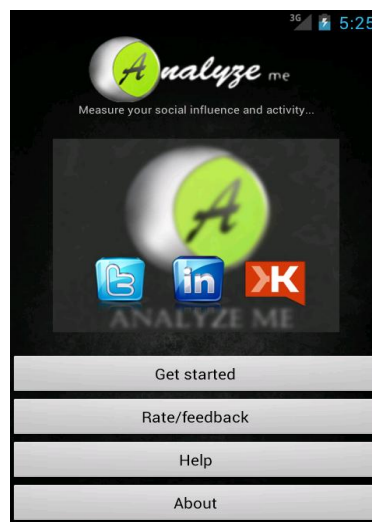


Figure 63: Mobile application home screen

When the application starts up, the screen above appears. The user has four options.

1. Get started
2. Rate/Feedback
3. Help
4. About

These options are described thoroughly below.

- 1) When a user clicks on "Get started" button similar to analyze-me website, the user is taken to the next screen, as seen in Figure 64 below:

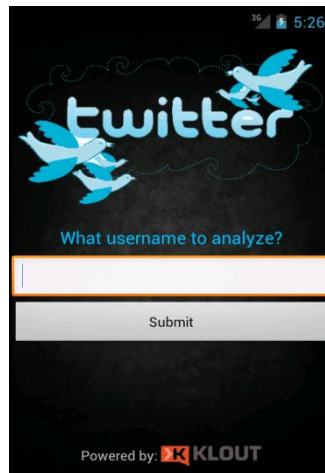


Figure 64: Mobile application get started screen

The code responsible for this layout transition in the first button is the following:

```
//assign new button b1 to the button in the form (button is a view and we reference it by ID)
Button b1 = (Button) findViewById(R.id.button1);

//method of the button.. what it will do
b1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        startActivity(new Intent("com.ptixiaki.twitter.GETSTARTED"));
    }
});
//reference to getstarted layout... from the manifest}});
```

Now, the user is asked for his Twitter username to proceed analyzing.

After entering username, when the user presses the “Submit” button, the following take place:

The mobile application makes a call to [analyze-me.com/mobile.php](http://analyze-me.com/mobile.php) which is the file responsible for handling mobile requests. It needs one parameter, the user’s username .

Then, it receives the response from the server and converts it to string. Lets review the case responsible for this action:

```
b1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {

        String result = null;
        InputStream is = null;
        StringBuilder sb=null;
        String url="http://www.analyze-me.com/mobile.php?username="; //pass username to php via url

        try{
            HttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url+input.getText());
            List nameValuePair = new ArrayList(1);
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
            HttpResponse response = httpClient.execute(httpPost);
            HttpEntity entity = response.getEntity();
            is = entity.getContent();
```

```

}catch(Exception e){
Log.e("log_tag", "Error in http connection"+e.toString());

//convert response to string
try{
BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);
sb = new StringBuilder();
sb.append(reader.readLine() + "\n");
String line="0";
while ((line = reader.readLine()) != null) {
sb.append(line + "\n");
}
is.close();
result=sb.toString();
}catch(Exception e){
Log.e("log_tag", "Error converting result "+e.toString());
}
//Check if the API works or the internet connection is not active!!!
if (result==null){
result="NUL RESPONSE";
Toast.makeText(getBaseContext(), "API response=NULL or No Internet connection",
Toast.LENGTH_LONG).show();
finish();}

```

Finally, checks if the response was valid and transfers the user to the results page

```

try{
JSONArray jArray = new JSONArray(result);
kscore=jArray.getString(0).substring(5).replace('"', '').replace('}', '').replace('{', '').trim();
input.setText("Please wait... working");
startActivity(new Intent("com.ptixiaki.twitter.RESULTS"));
//reference to results layout... from the manifest
finish(); //To close the old window before opening the new one
}catch(JSO NException e1){
Toast.makeText(getBaseContext(), "No such user found...", Toast.LENGTH_LONG).show();
}catch (ParseException e1){
e1.printStackTrace();}

```

Testing with username: **avraamak** gives us the result screen in Figure 65 below:

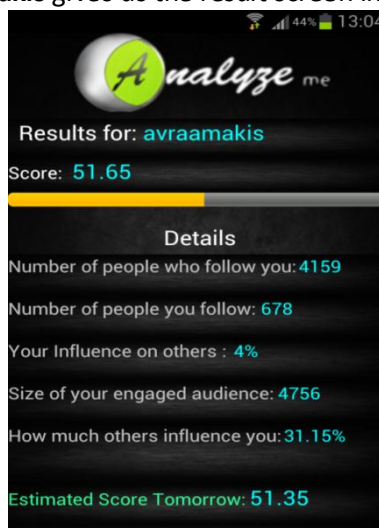


Figure 65: Mobile application results page

The results come from **passdata.php**, a file used on the server to pass the results to the mobile app through an **http** call.

Similarly, the application makes the call to the server, gets the response containing the results and converts the response to string, according to the following code:

```
String result = null;
InputStream is = null;
StringBuilder sb=null;

String url="http://www.analyze-me.com/passdata.php"; //get data from here
try{
HttpClient httpclient = new DefaultHttpClient();
HttpPost httppost = new HttpPost(url);
List nameValuePairs = new ArrayList(1);
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
HttpResponse response = httpclient.execute(httppost);
HttpEntity entity = response.getEntity();
is = entity.getContent();
}catch(Exception e){
Log.e("log_tag", "Error in http connection "+e.toString());

//convert response to string
try{
BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);
sb = new StringBuilder();
sb.append(reader.readLine() + "\n");
String line="0";
while ((line = reader.readLine()) != null) {
sb.append(line + "\n");
}
is.close();
result=sb.toString();
}catch(Exception e){
Log.e("log_tag", "Error converting result "+e.toString());}
```

All the results are stored in a **Json array** and can be accessed like this:

```
kscore=jArray.getString(1).substring(5).replace("'", '').replace('}', '').replace('{', '').trim();
output1.setText(kscore);
```

where the possible values of the array are: (jArray.getString(X)) where x=[0,15]

```
$output[0]=$xml2->name;
$output[1]=$xml->user->twitter_screen_name;
$output[2]=$xml->user->twitter_id;
$output[3]=$xml2->followers_count;
$output[4]=$xml2->friends_count;
$output[5]=$xml->user->score->amplification_score;
$output[6]=$xml->user->score->>true_reach;
$output[7]=$xml->user->score->network_score;
$output[8]=$xml->user->score->delta_1day;
$output[9]=$xml->user->score->delta_5day;
$output[10]=$xml->user->score->slope;
$output[11]=$xml->user->score->kscore;
$output[12]=$xml->user->score->description;
```



```
$output[13]=$xml->user->score->kclass;
$output[14]=$xml->user->score->kclass_id;
$output[15]=$xml->user->score->kclass_description;
```

The same procedure is followed to get the prediction result. The application calls this **url** instead:

```
http://www.analyze-me.com/predictm.php
```

Gets a response containing the predicted user score, converts to string and assigns its value to the text display field.

2) For the Rate/Feedback button, the code responsible for the transition is the following:

```
//assign new button b2 to the button in the form (button is a view and we reference it by ID)
Button b2 = (Button) findViewById(R.id.button2);
//method of the button.. what it will do
b2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        startActivity(new Intent("com.ptixiaki.twitter.FEEDBACK"));
    }
});
//reference to feedback layout... from the manifest}
```

When the user clicks submit (see Figure 66 below), a **url** call is made to **dbfeedback.php**,

```
"http://www.analyze-me.com/dbfeedback.php?rating="+ratebar.getRating()+"&feedback="+htmlEncoded;
```

The **dbfeedback.php** is a file on the remote server responsible for transferring the feedback data to the database.

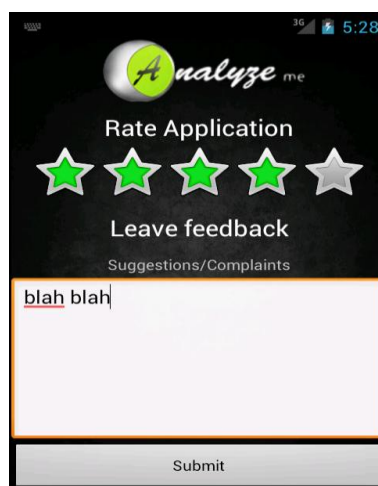


Figure 66: Mobile application feedback/rate screen

The **dbfeedback.php** file connects to the database and inserts the entry:

```
<?php
    $rating = $_GET['rating']; //catch the variable rating coming from android app
    $feedback = $_GET['feedback']; //catch the variable feedback coming from android app
    $feedback = str_replace(" ", "", $feedback);
    //Connect to MySQL
    @ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
    //Select database
```

```
mysql_select_db('perler2_ei') or die (mysql_error());
mysql_query("INSERT INTO dbfeedback VALUES ('$rating','$feedback')");
?>
```

If everything goes fine, a confirmation message appears:

```
//Confirmation message
Context context = getApplicationContext();
Toast toast = Toast.makeText(context, "Feedback sent to database", 3000);
toast.setGravity(Gravity.TOP, -30, 50);
toast.show();

//else, an error message appears:
}catch(Exception e){
Toast.makeText(getApplicationContext(), "Check Internet connection...", Toast.LENGTH_LONG).show();
Log.e("log_tag", "Error in http connection"+e.toString());}
```

3. Help button displays instructions on how to use the application, in simple steps (see Figure 67 below):

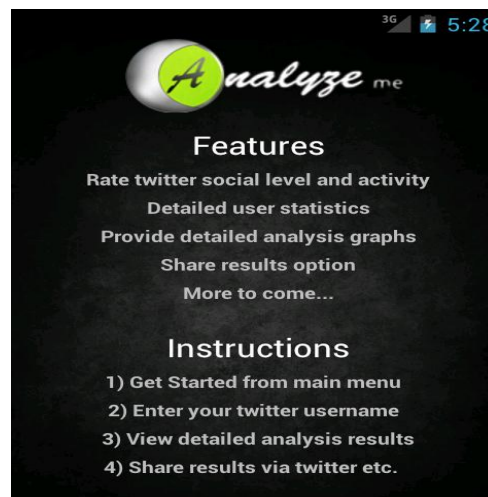


Figure 67: Mobile application features and help screen

4. About button takes the user to the next screen seen in Figure 68 below which contains information about the developer.



Figure 68: Mobile application about screen

## 9. Conclusion and future plans

With the social media being more and more popular every day, a popularity metrics service is getting more and more needed to analyze social influence between members of a social group and even create situations of competition. Also, the Android community is growing rapidly and gets more and more demanding. Therefore, a mobile application of almost anything is useful in today's and future's society.

Understanding how important these services are today, some aspects were researched and studied. Specifically:

- I was given the chance to develop a popularity metrics and forecasting system in PHP language.
- I studied programming in Android development platform in order to be able to complete the mobile application of the system.
- A thorough research was made on predictive algorithms, and turned out that the most useful in this case was Kalman filtering, especially when working with small amount of data. (As in our case, because most users are new to the service and don't have enough past data recorded to run other predictive algorithms with high success rate).

Future plans regarding the development of Analyze-me service include:

- Upgrade system to use the new LinkedIn API when it becomes available, to support more functions and hopefully provide 2<sup>nd</sup> and 3<sup>rd</sup> degree connections through the API.
- Upgrade to the latest KLOUT API when the latest version it available.
- Add encryption to the transferred data between website-database-mobile app.
- Implement more predictive algorithms and let the user choose which one to use.
- Comparative charts between user and his friends who have used the service.
- Option to share score via twitter, Facebook, LinkedIn.
- Possible Google+ support.
- Notification via mail when user score drops below a defined value .
- Update the mobile application to support LinkedIn analysis.

## REFERENCES

- [1] Schaum's Easy Outline of Probability and Statistics Crash Course (2002, 0071383417).pdf
- [2] Stormy Attaway, Matlab: A Practical Introduction to Programming and Problem Solving, Butterworth-Heinemann (an imprint of Elsevier, Inc.), 2009.
- [3] Chatfield Christopher, TIME-SERIES FORECASTING, Chapman & Hall/CRC Press LLC, Boca Raton London New York Washington, D.C., ISBN 1-58488-063-5, 2000.
- [4] Ajoy K. Palit and Dobrivoje Popovic, Computational Intelligence in Time Series Forecasting, Theory and Engineering Applications, Springer-Verlag London Limited, 2005.
- [5] <http://www.mathworks.com/>
- [6] Søren Bisgaard, Murat Kulahci, Time series analysis and forecasting by example, John Wiley & Sons, Inc., 2011.
- [7] Peter J. Brockwell, Richard A. Davis, Introduction to Time Series and Forecasting, 2<sup>nd</sup> Edition, Springer-Verlag New York, Inc., 2002.
- [8] Brian D. O. Anderson, John B. Moore, Optimal Filtering, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979.
- [9] RUEY S. TSAY, Analysis of Financial Time Series, 2<sup>nd</sup> Edition, John Wiley & Sons, Inc., 2005.
- [10] Douglas C. Montgomery, Cheryl L. Jennings, Murat Kulahci, Introduction to time series analysis and forecasting, John Wiley & Sons, Inc., 2008.
- [11] Ross Ihaka, Time Series Analysis, Lecture Notes for 475.726, Statistics Department, University of Auckland, April 14, 2005.
- [12] Kristiaan Pelckmans, Lecture Notes for a course on System Identification, v2012.
- [13] John Frain, Lecture Notes on Univariate Time Series Analysis and Box Jenkins Forecasting, Economic Analysis, Research and Publications, April 1992 (reprinted with revisions).
- [14] NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>, April, 2012.
- [15] Bølviken Erik, Christophersen Nils, Storvik Geir, Linear dynamical models, Kalman filtering and statistics, Lecture notes to IN-ST 259, University of Oslo, October 1998.
- [16] <http://www.whatissocialnetworking.com/> What is social networking, 2012.
- [17] <http://webtrends.about.com/od/socialnetworking/a/social-network.htm>
- [18] <http://www.searchenginemarketing.gr/blog/archives/175>
- [19] <http://en.wikipedia.org/wiki/Facebook>
- [20] <http://www.wisegeek.com/what-is-facebook.htm>
- [21] <https://developer.linkedin.com/documents/authentication>
- [22] [http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)
- [23] <http://en.wikipedia.org/wiki/LinkedIn>
- [24] <http://en.wikipedia.org/wiki/Klout>

## Source code Appendix

The source code of both the Website application and Android application was developed entirely by Perides A. Leros. The only third party code that was used is Oauth library provided by <http://oauth.net/> and was used to complete the authentication process with LinkedIn.

The source code files for both the website and mobile application are provided below:

**Twitter Analysis:** get.php, graph.php, predict.php

**LinkedIn Analysis:** auth.php, getLinkedIn.php, get2.php

**Mobile APP:** dbfeedback.php, mobile.php, passdata.php, predict.php

### Get.php (Responsible for twitter results page)

```
<?php
$input1=$_POST['input1'];
if ($input1!=""){
$url='http://api.klout.com/1/users/show.xml?key=w9ug4rrkmcprmx3gszscbfq&users='.$input1;
$url2=urlencode("http://api.twitter.com/1/users/show.xml?screen_name=".$input1);

$xml = simplexml_load_file($url) or die("User Not Accessible, Profile locked, Invalid username, or New
account. Please go back and retry");
$xml2 = simplexml_load_file($url2) or die("OOPS!! Temporary API error");
}
else{
echo "<script>alert('Invalid Username!!')</script>";
echo ("Invalid Username: Please go Back and retry");
exit();
}

//save to xml to pass to next page
$xml->asXML("xml.xml");
$xml2->asXML("xml2.xml");

//save score to file
$score = "score.txt";
$fh = fopen($score, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->kscore);
fclose($fh);

//save delta1 to file<-----
$delta1 = "delta1.txt";
$fh = fopen($delta1, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_1day);
fclose($fh);
```

```

//save delta5 to file<-----
$delta5 = "delta5.txt";
$fh = fopen($delta5, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_5day);
fclose($fh);
$ddd=date("Y-m-d");
$nnn=$xml->user->twitter_screen_name;
$klscore=$xml->user->score->kscore;

$username=$xml->user->twitter_screen_name;

//save username to file<-----
$usr = "usr.txt";
$fh = fopen($usr, 'w') or die("can't open data file");
fwrite($fh, $username);
fclose($fh);

$one=$xml2->followers_count;
$two=$xml2->friends_count;
$three=$xml->user->score->amplification_score;
$four=$xml->user->score->>true_reach;
$five=$xml->user->score->network_score;

//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());
mysql_query("INSERT INTO log VALUES ('$nnn','Twitter',CURDATE())");

$result=mysql_query("SHOW TABLES FROM perler2_ei LIKE '$input1'")
or die(mysql_error());
//echo(mysql_num_rows($result));

if(mysql_num_rows($result)==0){
//echo ("DEN YPARXEI O PINAKAS XRHSTH");
//ftiaxnoume to table loipon me to onoma tou xristi
mysql_query("CREATE TABLE $input1 (score varchar(50), date varchar(50) default '2012-01-01', one
varchar(50), two varchar(50), three varchar(50), four varchar(50), five varchar(50))");
//eisagoume to score ston pinaka log tou xristi mazi me torini imerominia
mysql_query("INSERT INTO $input1 VALUES($klscore,CURDATE(),$one,$two,$three,$four,$five)");}
else{//an yparxei o pinakas tote an den exei ksanakanei eggrafi tin idia mera kataxoreitai.
$lastentry=mysql_query("SELECT date FROM $input1 ORDER BY date DESC LIMIT 1");
$row1 = mysql_fetch_array($lastentry);
//echo($row1['date']);
$lastdate=$row1['date'];

$diff=mysql_query("SELECT DATEDIFF(CURDATE(),'$lastdate') AS DiffDate");
$row2 = mysql_fetch_array($diff);
//echo($row2['DiffDate']); //poses meres einai i diafora simera me tin teleutea eggrafi sti basi

```

```

if ($row2['DiffDate']!=0){
//eisagoume to score ston pinaka log tou xristi mazi me torini imerominia (an den exei ksanakanei eggrafi
tin idia mera
mysql_query("INSERT INTO $input1 VALUES($sklscore,CURDATE(),$one,$two,$three,$four,$five)");
}}
mysql_close($db);
?>

```

### **Graph.php (produces graphs for twitter analysis results)**

```

<?php // content="text/plain; charset=utf-8"
require_once ('src/jpgraph.php');
require_once ('src/jpgraph_line.php');
$theData = file_get_contents('score.txt');
$theData2 = file_get_contents('delta1.txt');
$theData3 = file_get_contents('delta5.txt');
$forecast=35;

if ($theData3<0 && $theData2>=0) {$vala=$theData-$theData3;$valb=$theData-$theData2;}
else if($theData2<0 && $theData3>=0) {$valb=$theData-$theData2;$vala=$theData-$theData3;}
else if($theData2<0 && $theData3<0) {$valb=$theData-$theData2;$vala=$theData-$theData3;}
else{
$vala=$theData-$theData3;
$valb=$theData-$theData2;
}
$datay1 = array($vala,$valb,$theData);

// Setup the graph
$graph = new Graph(500,325);//525 default , 345
$graph->SetScale("textlin");

$theme_class= new UniversalTheme;
$graph->SetTheme($theme_class);

$graph->title->Set('Score tracking for the last 5 days');
$graph->SetBox(false);

$graph->yaxis->HideZeroLabel();
$graph->yaxis->HideLine(false);
$graph->yaxis->HideTicks(false,false);

$graph->xaxis->SetTickLabels(array('5 days ago','Yesterday','Now',));
$graph->ygrid->SetFill(false);
$graph->SetBackgroundImage("graphlogo.png",BGIMG_FILLFRAME);

$sp1 = new LinePlot($datay1);
$graph->Add($sp1);

```

```

$p1->SetColor("#05B2D6");
$p1->SetLegend('Score');
$p1->mark->SetType(MARK_FILLEDCIRCLE,"1.0");
$p1->mark->SetColor('#05B2D6');
$p1->mark->SetFillColor('#05B2D6');
$p1->SetCenter();

```

```

$graph->legend->SetFrameWeight(1);
$graph->legend->SetColor('#4E4E4E','#00A78A');
$graph->legend->SetMarkAbsSize(8);

```

```

// Output line
$graph->Stroke();
?>

```

### **Predict.php (kalman filter algorithm for prediction)**

```

<?php
//read username from file
$usr = file_get_contents('usr.txt');

//read delta1 from file
$delta1 = file_get_contents('delta1.txt');

//read delta5 from file
$delta5 = file_get_contents('delta5.txt');

//read username from file
$score = file_get_contents('score.txt');

//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());

$query = "SELECT score FROM $usr ORDER BY date DESC LIMIT 6";
$resultaz = mysql_query($query) or die ("no query");

$result_array = array();
$counta=5;
$cnt=0; //poses kataxoriseis

while($row = mysql_fetch_assoc($resultaz))
{
    $result_array[$counta] = $row[score];
    $counta=$counta-1;
    $cnt=$cnt+1;
}

if($cnt==0){

```



```
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
$result_array[2] = $score+$delta5;
$result_array[3] = $score+$delta1;
$result_array[4] = $score+$delta1;
$result_array[5] = $score;
}

if($cnt==1){
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
$result_array[2] = $score+$delta1;
$result_array[3] = $score+$delta1;
$result_array[4] = $score;
}

if($cnt==2){
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
$result_array[2] = $score+$delta1;
$result_array[3] = $score+$delta1;
}

if($cnt==3){
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
$result_array[2] = $score+$delta1;
}

if($cnt==4){
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
}

if($cnt==5){
$result_array[0] = $score+$delta5;
}

mysql_close($db);

//print_r($result_array); //test print

//-----Kalman Filter Algorithm-----//
    $z=array();
    $z=$result_array;
    $n=3;

    //-----eye-----//
```

```
$eye=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        if($i==$j)
        {
            $eye[$i][$j]=1;
        }
        else
        {
            $eye[$i][$j]=0;
        }
    }
}

//-----Q_q-----//
$Q_d=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $Q_d[$i][$j]=0.01*$eye[$i][$j];
    }
}

//-----x_plus-----//
$x_plus=array(0,0,0);
for($i=0;$i<$n;$i++)
{
    $x_plus[$i]=-0.35*$z[$i];
}

//-----P_plus-----//
$P_plus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
```

```

        for($j=0;$j<$n;$j++)
        {
            $P_plus[$i][$j]=$eye[$i][$j];
        }
    }

//-----Storage variable vectors-----//
$x_est=array(array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0));

$y_est=array(0,0,0,0,0,0);
$errors=array(0,0,0,0,0,0);

//-----Beginning of Kalman Filter Loop-----//
$H=array(0,0,0);
$Hlast=array(0,0,0);
$K=array(0,0,0);
$x_minus=array(0,0,0);
$P_minus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

$p=0;
$ptr=0;
$y=array(0,0,0,0,0,0);
$mo=0;
$R=0;
$temp=0;
$temparray=array(0,0,0);
$mul=0;
$result=array(0,0,0);
$resultB;
$residual;
$errorPtr=0;
$k=0;

//Window size data matrix H
for ($k=3;$k<6;$k++)
{
    for ($l=$k-3;$l<=$k-1;$l++)
    {
        $H[$p]--=$z[$l];
        $p++;
    }
}

```

```

    }

    $p=0;

    //Propagation of state estimate x
    for($i=0;$i<$n;$i++)
    {
        $x_minus[$i]=$x_plus[$i];
    }
    //Propagation of covariance P
    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            $P_minus[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
        }
    }

    $y[$k]=$H[0]*$x_minus[0]+$H[1]*$x_minus[1]+$H[2]*$x_minus[2];

    //Storage variable holding estimates x_minus
    for($i=0;$i<$n;$i++)
    {
        $x_est[$ptr][$i]=$x_minus[$i];
    }

    //Storage variable holding estimates y_est
    $y_est[$ptr]=$y[$k];

    //Update equations
    //Covariance of Matrix H
    for($i=0;$i<3;$i++)
    {
        $mo=$mo+$H[$i];
    }

    $mo=$mo/3;

    for($i=0;$i<3;$i++)
    {
        $R=$R+((H[$i]-$mo)*(H[$i]-$mo));
    }
    $R=$R/3;
    $R=sqrt($R);

    //K = (P_minus * H') * ((H * P_minus * H' + R)^(-1))

```

```

$result[0]=$P_minus[0][0]*$H[0]+$P_minus[0][1]*$H[1]+$P_minus[0][2]*$H[2];
$result[1]=$P_minus[1][0]*$H[0]+$P_minus[1][1]*$H[1]+$P_minus[1][2]*$H[2];
$result[2]=$P_minus[2][0]*$H[0]+$P_minus[2][1]*$H[1]+$P_minus[2][2]*$H[2];

$resultB=$result[0]*$H[0]+$result[1]*$H[1]+$result[2]*$H[2];
$resultB=$resultB+$R;
$resultB=1/$resultB;
$result[0]=$result[0]*$resultB;
$result[1]=$result[1]*$resultB;
$result[2]=$result[2]*$resultB;

$K[0]=$result[0];
$K[1]=$result[1];
$K[2]=$result[2];

$residual=$z[$k]-$y[$k];
$errors[$errorPtr]=$residual; //Variable holding errors z-y_est
$errorPtr++;

//Update of state x
for($i=0;$i<$n;$i++)
{
    $x_plus[$i]=$x_minus[$i]+($K[$i]*$residual);
}

$mul=$K[0]*$result[0]+$K[1]*$result[1]+$K[2]*$result[2];

//Update of covariance P
for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        if($i==$j){$P_plus[$i][$j]=$P_minus[$i][$j]-$mul;}
    }
}
//End of Kalman Filter Loop
$ptr++;
}

//-----Last Kalman Filter estimation-----//
$x_last=array(0,0,0);
$P_last=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

$ptr=0;$k=5;
for ($l=$k-3+1;$l<=$k;$l++)

```

```

        {
            $Hlast[$p]=-$z[$i];
            $p++;
        }

for($i=0;$i<$n;$i++)
{
    $x_last[$i]=$x_plus[$i];
}

//Propagation of covariance P
for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $P_last[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
    }
}

$y_last=$Hlast[0]*$x_last[0]+$Hlast[1]*$x_last[1]+$Hlast[2]*$x_last[2];

for($i=0;$i<$n;$i++)
{
    $x_est[$ptr][$i]=$x_last[$i];
}

$y_last=round($y_last,2);

$y_est[$ptr]=$y_last;

$acc1=abs($errors[$errorPtr-2]/$y_est[$ptr-1]);
$acc2=abs($errors[$errorPtr-1]/$y_est[$ptr]);
$finError=100*round(((($acc1+$acc2)/2),4);

$tomorrow=$y_est[$ptr];
//$msr=$errors[$errorPtr-2]*$errors[$errorPtr-2]+$errors[$errorPtr-
1]*$errors[$errorPtr-1];
//$msr=sqrt($msr)/2;

//$finError = round($msr, 4);

/*
                                //TEST PRINTS//
echo ('Prediction: '.$y_est[$ptr]);
echo (" \n");
echo ('Error: '.$finError.'%');
echo (" \n");
echo ('Under Development...');
*/

```

```

//*****
*****
*****//
//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "do tchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());

$query = "SELECT score FROM $usr ORDER BY date DESC LIMIT 30";
$rez = mysql_query($query) or die ("no query");

$res_arr = array();
$counta2=30;
$count2=0; //poses kataxoriseis

while($ro = mysql_fetch_assoc($rez))
{
    $res_arr[$counta2] = $ro[score];
    $counta2=$counta2-1;
    $count2=$count2+1;
}
$week="N/A";
$possible=0;
if($count2>29)$possible=1;

if ($possible==1)
{
    //-----Kalman Filter Algorithm for next week-----//
    $z=array();
    $z=array($res_arr[30],$res_arr[24],$res_arr[18],$res_arr[12],$res_arr[6],$score);
    //$z=array(1,2,3,4,5,6);
    $n=3;

    //-----eye-----//
    $eye=array(array(0,0,0),
    array(0,0,0),
    array(0,0,0));

    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            if($i==$j)
            {
                $eye[$i][$j]=1;
            }
        }
    }
}

```

```
        }
        else
        {
            $eye[$i][$j]=0;
        }
    }
}

//-----Q_q-----//
$Q_d=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $Q_d[$i][$j]=0.01*$eye[$i][$j];
    }
}

//-----x_plus-----//
$x_plus=array(0,0,0);
for($i=0;$i<$n;$i++)
{
    $x_plus[$i]=-0.35*$z[$i];
}

//-----P_plus-----//
$P_plus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $P_plus[$i][$j]=$eye[$i][$j];
    }
}

//-----Storage variable vectors-----//
$x_est=array(array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
```



```

array(0,0,0,0,0,0),
array(0,0,0,0,0,0));

$y_est=array(0,0,0,0,0,0);
$errors=array(0,0,0,0,0,0);

//-----Beginning of Kalman Filter Loop-----//
$H=array(0,0,0);
$Hlast=array(0,0,0);
$K=array(0,0,0);
$x_minus=array(0,0,0);
$P_minus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

$p=0;
$ptr=0;
$y=array(0,0,0,0,0,0);
$mo=0;
$R=0;
$temp=0;
$temparray=array(0,0,0);
$mul=0;
$result=array(0,0,0);
$resultB;
$residual;
$errorPtr=0;
$k=0;

//Window size data matrix H
for ($k=3;$k<6;$k++)
{
    for ($l=$k-3;$l<=$k-1;$l++)
        {
            $H[$p]=-$z[$l];
            $p++;
        }
    $p=0;

    //Propagation of state estimate x
    for($i=0;$i<$n;$i++)
    {
        $x_minus[$i]=$x_plus[$i];
    }
    //Propagation of covariance P
    for ($i=0;$i<$n;$i++)

```

```

{
    for($j=0;$j<$n;$j++)
    {
        $P_minus[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
    }
}

$y[$k]=$H[0]*$x_minus[0]+$H[1]*$x_minus[1]+$H[2]*$x_minus[2];

//Storage variable holding estimates x_minus
for($i=0;$i<$n;$i++)
{
    $x_est[$ptr][$i]=$x_minus[$i];
}

//Storage variable holding estimates y_est
$y_est[$ptr]=$y[$k];

//Update equations
//Covariance of Matrix H
for($i=0;$i<3;$i++)
{
    $mo=$mo+$H[$i];
}

$mo=$mo/3;

for($i=0;$i<3;$i++)
{
    $R=$R+((($H[$i]-$mo)*($H[$i]-$mo)));
}
$R=$R/3;
$R=sqrt($R);

//K = (P_minus * H') * ((H * P_minus * H' + R) ^ (-1))
$result[0]=$P_minus[0][0]*$H[0]+$P_minus[0][1]*$H[1]+$P_minus[0][2]*$H[2];
$result[1]=$P_minus[1][0]*$H[0]+$P_minus[1][1]*$H[1]+$P_minus[1][2]*$H[2];
$result[2]=$P_minus[2][0]*$H[0]+$P_minus[2][1]*$H[1]+$P_minus[2][2]*$H[2];

$resultB=$result[0]*$H[0]+$result[1]*$H[1]+$result[2]*$H[2];
$resultB=$resultB+$R;
$resultB=1/$resultB;
$result[0]=$result[0]*$resultB;
$result[1]=$result[1]*$resultB;
$result[2]=$result[2]*$resultB;

```

```

    $K[0]=$result[0];
    $K[1]=$result[1];
    $K[2]=$result[2];

    $residual=$z[$k]-$y[$k];
    $errors[$errorPtr]=$residual; //Variable holding errors z-y_est
    $errorPtr++;

    //Update of state x
    for($i=0;$i<$n;$i++)
    {
        $x_plus[$i]=$x_minus[$i]+($K[$i]*$residual);
    }

    $mul=$K[0]*$result[0]+$K[1]*$result[1]+$K[2]*$result[2];

    //Update of covariance P
    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            if($i==$j){$P_plus[$i][$j]=$P_minus[$i][$j]-$mul;}
        }
    }
//End of Kalman Filter Loop
    $ptr++;
}

//-----Last Kalman Filter estimation-----//
    $x_last=array(0,0,0);
    $P_last=array(array(0,0,0),
    array(0,0,0),
    array(0,0,0));

    $p=0;$k=5;
    for ($l=$k-3+1;$l<=$k;$l++)
    {
        $Hlast[$p]=$z[$l];
        $p++;
    }

    for($i=0;$i<$n;$i++)
    {
        $x_last[$i]=$x_plus[$i];
    }

    //Propagation of covariance P

```

```

        for ($i=0;$i<$n;$i++)
        {
            for($j=0;$j<$n;$j++)
            {
                $P_last[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
            }
        }

        $y_last=$Hlast[0]*$x_last[0]+$Hlast[1]*$x_last[1]+$Hlast[2]*$x_last[2];

        for($i=0;$i<$n;$i++)
        {
            $x_est[$ptr][$i]=$x_last[$i];
        }

        $y_last=round($y_last,2);

        $y_est[$ptr]=$y_last;

$week=$y_est[$ptr];

}

mysql_close($db);

if (($tomorrow>$score)&&($week>$score))
    {
        $img="rise.jpg";
        $inf="Yeah! Your score is estimated to rise tomorrow, and keep rising next week";
    }
else if(($tomorrow<$score)&&($week>$score))
    {
        $img="rise.jpg";
        $inf="Your score is estimated to decrease tomorrow but rise within next week";
    }
else if(($tomorrow>$score)&&($week<$score))
    {
        $img="crash.jpg";
        $inf="Your score is estimated to rise tomorrow but decrease withing next week";
    }
else if(($tomorrow<$score)&&($week<$score))
    {
        $img="crash.jpg";
        $inf="Ooops! Your score is estimated to decrease during the next days";
    }
else

```

```

    {
        $img="nochange.jpg";
        $inf="No changes estimated";
    }

```

?>

### **Auth.php (For authentication)**

```

<?php
    session_start();

    $config['base_url']      = 'http://analyze-me.com/auth.php';
    $config['callback_url']  = 'http://analyze-me.com/getlinkedin.php';
    $config['linkedin_access'] = 'hMWnKT760YTsQmmkNEhYQSPOA6y7UMKz2Y8Eu mXlgrDoJLCQuz18fWkLjxHNPOq_';
    $config['linkedin_secret'] = '6XGQh1M1jUVaCRSHx8qYrMMJTXnOga74gp0k2hsSu-DbHQjPm-iCEFAV6xrpD3Hc';

    include_once "linkedin.php";

    # First step is to initialize with your consumer key and secret. We'll use an out-of-band oauth_callback
    $linkedin = new LinkedIn($config['linkedin_access'], $config['linkedin_secret'], $config['callback_url'] );
    //$linkedin->debug = true;

    # Now we retrieve a request token. It will be set as $linkedin->request_token
    $linkedin->getRequestToken();
    $_SESSION['requestToken'] = serialize($linkedin->request_token);

    # With a request token in hand, we can generate an authorization URL, which we'll direct the user to
    //echo "Authorization URL: " . $linkedin->generateAuthorizeUrl() . "\n\n";
    header("Location: " . $linkedin->generateAuthorizeUrl(),false);
?>

```

### **GetLinkedIn.php (Responsible for LinkedIn authentication)**

```

<?php
    session_start();

    $config['base_url']      = 'http://analyze-me.com/auth.php';
    $config['callback_url']  = 'http://analyze-me.com/getlinkedin.php';
    $config['linkedin_access'] = 'hMWnKT760YTsQmmkNEhYQSPOA6y7UMKz2Y8Eu mXlgrDoJLCQuz18fWkLjxHNPOq_';
    $config['linkedin_secret'] = '6XGQh1M1jUVaCRSHx8qYrMMJTXnOga74gp0k2hsSu-DbHQjPm-iCEFAV6xrpD3Hc';

    include_once "linkedin.php";

    # First step is to initialize with consumer key and secret. We'll use an out-of-band oauth_callback
    $linkedin = new LinkedIn($config['linkedin_access'], $config['linkedin_secret'], $config['callback_url'] );

```

```

//$linkedin->debug = true;

if (isset($_REQUEST['oauth_verifier'])){
    $_SESSION['oauth_verifier'] = $_REQUEST['oauth_verifier'];

    $linkedin->request_token = unserialize($_SESSION['requestToken']);
    $linkedin->oauth_verifier = $_SESSION['oauth_verifier'];
    $linkedin->getAccessToken($_REQUEST['oauth_verifier']);

    $_SESSION['oauth_access_token'] = serialize($linkedin->access_token);
    header("Location: " . $config['callback_url']);
    exit;
}
else{
    $linkedin->request_token = unserialize($_SESSION['requestToken']);
    $linkedin->oauth_verifier = $_SESSION['oauth_verifier'];
    $linkedin->access_token = unserialize($_SESSION['oauth_access_token']);
}

//You now have a $linkedin->access_token and can make calls on behalf of the current member
$xml_response = $linkedin->getProfile("~:(id,first-name,last-name,num-connections,current-status-
timestamp,current-share)");
$x = str_replace("-", "_", $xml_response);
$x2 = str_replace("UTF_8", "UTF-8", $x);
$x3 = str_replace('s total="1"', 's', $x2);

//write to file
$myFile = "data.txt";
$fh = fopen($myFile, 'w') or die("can't open data file");
fwrite($fh, $x3);
fclose($fh);

header("Location: " . "http://analyze-
me.com/index.php?option=com_wrapper&view=wrapper&Itemid=69");
?>

```

### **Get2.php (responsible for the results page for LinkedIn)**

```

<?php
$xml3 = simplexml_load_file("data.txt");
$ddd=date("Y-m-d");

$nnn=$xml3->id;
//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
$first=1;
$second=1;
$third=1;

```

```

mysql_select_db('perler2_ei') or die (mysql_error());
$rez=mysql_query("SELECT * from linkedin WHERE id='$nnn'");
$row = mysql_fetch_assoc($rez);

if ($nnn!=$row['id'])mysql_query("INSERT INTO linkedin VALUES ('$nnn','$first','$second','$third')");

$first=$row['1st'];
$second=$row['2nd'];
$third=$row['3rd'];

$ratio1=$first;
$ratio2=$second/$first;
$ratio3=$third/$second;

$n2=100;
$n4=0.1;
$n6=10000;
$n8=0.1;
$n10=100000;

$multiplier=0.0;
$img="starter.jpg";
$desc="temporary text";
$criteria=0.00;
$criteria=round((((($first-$n2)/100)+(($ratio2/$ratio1)-$n4)+(($second-$n6)/10000)+(((($ratio3/$ratio1)-$n8))+(($third-$n10)/100000)),2);

if ($criteria<10)
{
    $img="starter.jpg";
    $desc="You are either a new user or a person that only uses his account rarely, only checking for updates upon being notified by email.";
}
else if ($criteria>=10 && $criteria<20)
{
    $img="casual.jpg";
    $desc="You are a casual linkedIN user, you don't post updates very frequently but you have an increased number of connections, you use your profile mainly for following updates from your connections.";
}
else if ($criteria>=20 && $criteria<40)
{
    $img="advanced.jpg";
    $desc="You are an advanced linkedIN user, you use your account frequently either posting or checking updates. You have a good position within the linkedIN society, with many connections following your updates.";
}
else if ($criteria>=40)
{

```

```

    $img="professional.jpg";
    $desc="You are a linkedIN power user, you use your account daily, posting and reading updates
very frequently. You use linkedIN as your main social networking platform, and you have made a strong
reputation among your connections.";
    }
?>

```

#### **Dbfeedback.php (responsible for transmitting feedback message from mobile to database)**

```

<?php
    $rating = $_GET['rating']; //catch the variable rating coming from android app
    $feedback = $_GET['feedback']; //catch the variable feedback coming from android app
    $feedback = str_replace("_", "", $feedback);
//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());
mysql_query("INSERT INTO dbfeedback VALUES ('$rating', '$feedback')");?>

```

#### **mobile.php (Call APIs and return results to mobile)**

```

<?php
    $username = $_GET['username']; //catch the variable username coming from android app
    $url='http://api.klout.com/1/users/show.xml?key=w9ug4rrkmcprpmxt3gszscbfq&users='.$username;
    $url2='http://api.twitter.com/1/users/show.xml?screen_name='.$username;

$xml = simplexml_load_file($url) or die("User Not Accessible, Profile locked, Invalid username, or New
account. Please go back and retry");
$xml2 = simplexml_load_file($url2);

//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());
mysql_query("INSERT INTO log VALUES ('$nnn', 'Mobile', NOW())");

//print to json format
$output[0]=$xml2->name;
$output[1]=$xml->user->twitter_screen_name;
$output[2]=$xml->user->twitter_id;
$output[3]=$xml2->followers_count;
$output[4]=$xml2->friends_count;
$output[5]=$xml->user->score->amplification_score;
$output[6]=$xml->user->score->>true_reach;
$output[7]=$xml->user->score->network_score;
$output[8]=$xml->user->score->delta_1day;
$output[9]=$xml->user->score->delta_5day;
$output[10]=$xml->user->score->slope;
$output[11]=$xml->user->score->kscore;
$output[12]=$xml->user->score->description;
$output[13]=$xml->user->score->kclass;

```



```
$output[14]=$xml->user->score->kclass_id;
$output[15]=$xml->user->score->kclass_description;

print(json_encode($output));

//save username to file
$usr = "usr.txt";
$fh = fopen($usr, 'w') or die("can't open data file");
fwrite($fh, $username);
fclose($fh);

//save username to file
$d1 = "d1.txt";
$fh = fopen($d1, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_1day);
fclose($fh);

//save username to file
$d5 = "d5.txt";
$fh = fopen($d5, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->delta_5day);
fclose($fh);

//save username to file
$d0 = "d0.txt";
$fh = fopen($d0, 'w') or die("can't open data file");
fwrite($fh, $xml->user->score->kscore);
fclose($fh);
?>
```

**Passdata.php ( pass username from one page to another in android app)**

```
<?php
//read username from file
$usr = file_get_contents('usr.txt');
print($usr);
?>
```

**Predict.php (Imports user's past score data, runs kalman filter for mobile and returns prediction)**

```
<?php
//read username from file
$usr = file_get_contents('usr.txt');

//read delta1 from file
$delta1 = file_get_contents('d1.txt');

//read delta5 from file
$delta5 = file_get_contents('d5.txt');
```

```
//read username from file
$score = file_get_contents('d0.txt');

//Connect to MySQL
@ $db=mysql_pconnect("mysql18.freehostia.com", "perler2_ei", "dotchika");
//Select database
mysql_select_db('perler2_ei') or die (mysql_error());

$query = "SELECT score FROM $usr ORDER BY date DESC LIMIT 6";
$resultaz = mysql_query($query) or die ("no query");

$result_array = array();
$counta=5;
$cnt=0; //poses kataxorisais

while($row = mysql_fetch_assoc($resultaz))
{
    $result_array[$counta] = $row[score];
    $counta=$counta-1;
    $cnt=$cnt+1;
}

if($cnt==0){
    $result_array[0] = $score+$delta5;
    $result_array[1] = $score+$delta5;
    $result_array[2] = $score+$delta5;
    $result_array[3] = $score+$delta1;
    $result_array[4] = $score+$delta1;
    $result_array[5] = $score;
}

if($cnt==1){
    $result_array[0] = $score+$delta5;
    $result_array[1] = $score+$delta5;
    $result_array[2] = $score+$delta1;
    $result_array[3] = $score+$delta1;
    $result_array[4] = $score;
}

if($cnt==2){
    $result_array[0] = $score+$delta5;
    $result_array[1] = $score+$delta5;
    $result_array[2] = $score+$delta1;
    $result_array[3] = $score+$delta1;
}

if($cnt==3){
    $result_array[0] = $score+$delta5;
    $result_array[1] = $score+$delta5;
```

```
$result_array[2] = $score+$delta1;
}

if($cnt==4){
$result_array[0] = $score+$delta5;
$result_array[1] = $score+$delta5;
}

if($cnt==5){
$result_array[0] = $score+$delta5;
}

        mysql_close($db);
//print_r($result_array); //test print

//-----Kalman Filter Algorithm-----//
    $z=array();
    $z=$result_array;
    $n=3;

    //-----eye-----//
    $eye=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            if($i==$j)
            {
                $eye[$i][$j]=1;
            }
            else
            {
                $eye[$i][$j]=0;
            }
        }
    }

    //-----Q_q-----//
    $Q_d=array(array(0,0,0),
array(0,0,0),
array(0,0,0));
```

```

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $Q_d[$i][$j]=0.01*$eye[$i][$j];
    }
}

//-----x_plus-----//
$x_plus=array(0,0,0);
for($i=0;$i<$n;$i++)
{
    $x_plus[$i]=-0.35*$z[$i];
}

//-----P_plus-----//
$P_plus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

for ($i=0;$i<$n;$i++)
{
    for($j=0;$j<$n;$j++)
    {
        $P_plus[$i][$j]=$eye[$i][$j];
    }
}

//-----Storage variable vectors-----//
$x_est=array(array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0),
array(0,0,0,0,0,0));

$y_est=array(0,0,0,0,0,0);
$errors=array(0,0,0,0,0,0);

//-----Beginning of Kalman Filter Loop-----//
$H=array(0,0,0);
$Hlast=array(0,0,0);
$K=array(0,0,0);
$x_minus=array(0,0,0);
$P_minus=array(array(0,0,0),
array(0,0,0),
array(0,0,0));

```

```

$p=0;
$ptr=0;
$y=array(0,0,0,0,0,0);
$mo=0;
$R=0;
$temp=0;
$temparray=array(0,0,0);
$mul=0;
$result=array(0,0,0);
$resultB;
$residual;
$errorPtr=0;
$k=0;

//Window size data matrix H
for ($k=3;$k<6;$k++)
{
    for ($l=$k-3;$l<=$k-1;$l++)
    {
        $H[$p]=-$z[$l];
        $p++;
    }
    $p=0;

    //Propagation of state estimate x
    for($i=0;$i<$n;$i++)
    {
        $x_minus[$i]=$x_plus[$i];
    }
    //Propagation of covariance P
    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            $P_minus[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
        }
    }

    $y[$k]=$H[0]*$x_minus[0]+$H[1]*$x_minus[1]+$H[2]*$x_minus[2];

    //Storage variable holding estimates x_minus
    for($i=0;$i<$n;$i++)

```

```

        {
            $x_est[$ptr][$i]=$x_minus[$i];
        }

//Storage variable holding estimates y_est
    $y_est[$ptr]=$y[$k];

//Update equations
//Covariance of Matrix H
    for($i=0;$i<3;$i++)
    {
        $mo=$mo+$H[$i];
    }

    $mo=$mo/3;

    for($i=0;$i<3;$i++)
    {
        $R=$R+((($H[$i]-$mo)*($H[$i]-$mo));
    }
    $R=$R/3;
    $R=sqrt($R);

//K = (P_minus * H') * ((H * P_minus * H' + R)^(-1))
    $result[0]=$P_minus[0][0]*$H[0]+$P_minus[0][1]*$H[1]+$P_minus[0][2]*$H[2];
    $result[1]=$P_minus[1][0]*$H[0]+$P_minus[1][1]*$H[1]+$P_minus[1][2]*$H[2];
    $result[2]=$P_minus[2][0]*$H[0]+$P_minus[2][1]*$H[1]+$P_minus[2][2]*$H[2];

    $resultB=$result[0]*$H[0]+$result[1]*$H[1]+$result[2]*$H[2];
    $resultB=$resultB+$R;
    $resultB=1/$resultB;
    $result[0]=$result[0]*$resultB;
    $result[1]=$result[1]*$resultB;
    $result[2]=$result[2]*$resultB;

    $K[0]=$result[0];
    $K[1]=$result[1];
    $K[2]=$result[2];

    $residual=$z[$k]-$y[$k];
    $errors[$errorPtr]=$residual; //Variable holding errors z-y_est
    $errorPtr++;

//Update of state x
    for($i=0;$i<$n;$i++)
    {
        $x_plus[$i]=$x_minus[$i]+($K[$i]*$residual);
    }

```

```

    }

    $mul=$K[0]*$result[0]+$K[1]*$result[1]+$K[2]*$result[2];

    //Update of covariance P
    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            if($i==$j){$P_plus[$i][$j]=$P_minus[$i][$j]-$mul;}
        }
    }
//End of Kalman Filter Loop
    $ptr++;
}

//-----Last Kalman Filter estimation-----//
    $x_last=array(0,0,0);
    $P_last=array(array(0,0,0),
    array(0,0,0),
    array(0,0,0));

    $p=0;$k=5;
    for ($l=$k-3+1;$l<=$k;$l++)
    {
        $Hlast[$p]=-$z[$l];
        $p++;
    }

    for($i=0;$i<$n;$i++)
    {
        $x_last[$i]=$x_plus[$i];
    }

    //Propagation of covariance P
    for ($i=0;$i<$n;$i++)
    {
        for($j=0;$j<$n;$j++)
        {
            $P_last[$i][$j]=$P_plus[$i][$j]+$Q_d[$i][$j];
        }
    }

    $y_last=$Hlast[0]*$x_last[0]+$Hlast[1]*$x_last[1]+$Hlast[2]*$x_last[2];

    for($i=0;$i<$n;$i++)
    {

```

```
        $x_est[$ptr][$i]=$x_last[$i];
    }

    $y_last=round($y_last,2);

    $y_est[$ptr]=$y_last;

    $finError = round(abs($errors[$errorPtr-1]/($y_est[$ptr-1]+$errors[$errorPtr-1])), 4);

$outputi=$y_est[$ptr];

echo ($outputi);
?>
```