CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

DMForex: A Data Mining Application to Predict Currency Exchange Rates and Trends

A graduate project submitted in partial fulfillment of the requirements

For the degree of Master of Science

in Computer Science

By

Jorge Orrantia

May 2012

The graduate project of Jorge Orrantia is approved:

Gloria Melara, Ph. D.	Date
Rick Covington, Ph. D.	Date
Dr. George (Taehyung) Wang, Chair	Date

California State University, Northridge

Dedication

To Sandy and Ron Kobrine who inspire me every day of my life to be the best that I can be. They are, and will always be, my constant motivation to dream the greatest dreams, to seek the highest goals, and to help others – just the way the Kobrine family helped me.

Acknowledgements

I would like to acknowledge Professor Lingard for his invaluable help and advice. I would also like to extend my eternal gratitude to all the members of my graduate project committee for all their help and the sharing of their knowledge in this work and in past classes. Special thanks to Dr. Melara for showing me not only that the purpose of a user interface is to be useful, but we as people, our most important quality is to be useful. My gratitude is owed to Professor Covington who planted in my mind the desire to try something new with WPF. Last, but not least, my most sincere appreciation to Dr. Wang, my committee chair, who guided me through the whole degree process, and for showing me the techniques of knowledge discovery from huge amounts of data.

Table of Contents

List of Tables

List of Figures

ABSTRACT

DMForex: A Data Mining Application to Predict Currency Exchange Rates and Trends

By

Jorge Orrantia

Master of Science in Computer Science

The foreign exchange (FX) market is one of the most liquid markets in the world today. In this market currency pairs are sold and bought based on the exchange rate. It is important to market participants to accurately predict future values and trends. In this project, DMForex has been designed and implemented to predict currency exchange rates and trends using data mining techniques and algorithms.

DMForex is a composite WPF application designed and built using the PRISM 4 guidance. It uses the Model-View-ViewModel (MVVM) UI (user interface) pattern to separate the presentation and business logic of the application.

To implement the forecasting of the exchange rates and trends, DMForex uses SQL Server 2008 with Analysis Services. Daily historical price data for some of major world currency pairs for the last 12 years is used.

Microsoft SQL Server Analysis Services' Analysis Management Objects (AMO) are used to programmatically implement the predictions of exchange rates and trends by creating

the analysis database, mining structure, and mining models. These mining models can then be queried through ADOMD.Net using Data Mining Extensions (DMX) commands.

To predict the exchange rates, the program creates a mining model that uses the Microsoft's Time Series Algorithm using the standard dataset.

For the prediction of trends, the user is able to change the dataset by adding the Simple Moving Average (SMA) and Relative Strength Index (RSI) columns. These new columns are calculated using the TA-lib library of technical indicators. The user defines a trend. This column is defined in terms of days and PIPs. The program then predicts the next trend using Microsoft's Decision Trees Algorithm.

A testing module allows the user to test the accuracy of the rates and trends predictions.

Chapter 1: Introduction

1.1 Background

The ability to accurately predict the future values of a time series can be beneficial to individuals and organizations. This is especially true when dealing with time series in the Foreign Exchange (FX) market where financial institutions and individual traders engage in the selling and buying of currency pairs in order to make a profit DMForex was conceived with the idea of data mining time series. Time series is a set of discrete numbers associated with time. They are generally spaced at equal intervals. Time series analysis and forecasting has been the subject of numerous studies. Many mathematical models and techniques have been evolved through the years. Data mining techniques have emerged to forecast time series in recent years.

1.2 Objectives

One of the main objectives of DMForex was to implement a program to predict currency exchange rates and trends using data mining techniques and to assess the accuracy of those predictions.

Another objective for developing DMForex was to showcase a series of tools, techniques and technologies involved in the creation of software.

1.3 Approach

I developed a solution to this problem of predicting exchange rates and trends by designing and implementing a solution in Visual Studio 2010. Using Prism 4, a solution was composed of modules in which each task in the data mining process as referred in the industry accepted CRISP-DM guidelines was implemented. One of the modules would be in charge of data loading and preparation. Another module would handle the data mining modeling. There would be a model to visualize the results. The data mining module was implemented using the techniques and technology available in Microsoft SQL 2008 Analysis Services. Predictions were created using some of the algorithms readily available in the Analysis Server. A module to generate and test past predictions was also developed. For the prediction of rates a time series algorithm was used. It used the close price of the currency as input, and it predicted the next close value. For the prediction of trends a classification trees algorithm was used. Two new columns were added to the dataset using the Simple Moving Average (SMA) and Relative Strength Index (RSI) indicators. A new column named trend was calculated by looking ahead the number of days, provided by the user, and looking at whether the price of currency went up, down, both, or remain the same a certain number of pips, also provided by the user.

1.4 Tools and Technologies

There has been a set of technologies and tools that I have used. Some of them are:

- Prism
- WPF
- XAML
- TA-Lib
- SQL
- Team Foundation Server 2010
- SharePoint 2010
- Visual Studio 2010 Ultimate Edition
- SQL Server 2008 with Analysis Services

- Fluent Ribbon
- Windows Server 2008 R2 Datacenter Edition
- Amazon’s Elastic Cloud Computing

1.5 What’s in this Thesis

This thesis is organized as follows. Chapter 2 presents a survey of the literature. Chapter 3 deals with the key decisions I had to make in the implementation of this project. Chapter 4 describes the software implementation of DMForex. Chapter 5 is a summary of the experiments and results obtained with DMForex. Lastly, Chapter 6 offers some conclusions and future work.

Chapter 2: Literature Survey

One of the main goals of this graduate project was to accurately predict future exchange rates in the Foreign Exchange market. In order to carry out this goal my research yielded a collection of books and articles that provided an understanding of the different areas of knowledge involved along with past and present techniques and methodologies used in the forecasting of foreign exchange rates. In this survey, I tried to present the material in the order it helped me understand the subject. First, I cover time series and the different types, including financial time series. I then cover some of the current methodologies and techniques used to predict future values in the series. I then discuss data mining and its use in forecasting time series. The next group of papers covers the data mining of the foreign exchange time series.

2.1 Time Series

A desire to understand currency historical data took me to time series analysis and forecasting. What kind of data is it? What are the methodologies and techniques used to predict future values and trends? How is the accuracy of the models measured? To answer these questions I turned to several books on the topic of time series analysis and forecasting.

Data is everywhere around us. In the last two decades computers and the Internet have made it easier and cheaper to collect and store huge amounts of data. Individuals as well as organizations are constantly in search of ways to analyze their data.

In his book “Data Analysis with Open Source Tools” Jarnet introduces the reader to the concepts and techniques for working with data [22]. He covers a wide range of topics divided in three parts: graphing data, modeling data, and mining data. In particular, a chapter devoted to the time series analysis of time as a variable was useful to this project. He defines time series as the variation of some quantity over time. He gives us some examples of time series such as stock market movements to the CPU utilization of a personal desktop computer. He believes what makes time series data so important is that the variable of time gives this kind of data a "context" for the quantity. He points out that the nature of time-series analysis as a bivariate problem and that a "rather specialized set of methods has been developed to deal with them."

The next three books introduced me to time series analysis in deeper detail. Chatfield discusses some time series and the terminology. He also sets forth the objectives of time-series analysis. Time series is defined as "a collection of observations made sequentially through time." [23]. He gives us some examples that range from economics to engineering. He describes the classic Beveridge wheat price index which is a series of the price of wheat for 50 places recorded each year from 1500 to 1889. He shows us in a chart how the data displays some cyclic behavior over some time period. He gives us many examples of physical time series in the fields of meteorology, marine science and geophysics, such as daily rainfall, hourly, daily, and monthly air temperature. He mentions, among others, additional examples of these time series such as marketing time series and demographic time series such as sales figures and population changes, respectively. He points out the terminology used in time-series analysis. For example he distinguishes between continuous and discrete time series. He defines a continuous time series when observations are made continuously through time and discrete time series when observations are taken only at specific times, usually equally spaced. He stresses the special feature of time series analysis successive observations are independent of past values and points out the importance of taking the time order of the observations into consideration when carrying out the analysis. He defines the two types of predictions in terms of the dependence of observations in the data. If the data is dependent of past values it is a deterministic time series. On the other hand if it is independent of past observations the time series is said to be stochastic, and the future values can partly be predicted by past values with a probability distribution

Chatfield describes the four objectives of time series analysis: description, explanation, prediction, and control. The first objective of time series analyzes tries to get a description of the main properties of the data by plotting the observations against time in a time to get the simple descriptive measures. A time plot can reveal seasonal effects, trends, and outliers, which he defines as “wild observations.” Secondly the author describes the explanation objective of the time-series analysis. In this case an analyst may be interested in explaining how one time series explain the variation in another. Two methods are used here: regression models and linear system. The third objective of time series analysis is prediction. In this case, the goal of the analysis is to predict future values of a given time series. This is one of the objectives in question in this graduate project. I would like to predict future price values of a currency pair given its historical daily data. The last objective in time series analysis is control. When this analysis is conducted, the objective is to improve the control over some physical or economic system. Since this kind of objective is not one of the goals of this project, it has not been researched further.

2.2 Time Series Forecasting

A wide range of data mining methods have been applied to the forecasting of time series. They range from simple statistical methods to more advanced methods such as data mining and chaos theory.

At the present time artificial neural networks are found to be a popular technique as described in [9], [15], [16], and [17]. The use of genetic algorithms has also been proposed. Chiraphadhanakul et al. apply a genetic forecasting algorithm using features of genetic algorithms [13]. It then supplies the results of the algorithm to two time series data: commercial banks deposit and bankruptcy prediction.

Hansen et al. use support vector machines to predict time series [18]. Results from a comparative analysis with classical statistical models such ARIMA were compared.

Modified approaches using a combination of methods such as the work done by Ferreira et al. in which they use a hybrid model composed of an artificial network and a modified genetic algorithm to predict the time series [3].

Another innovative method to forecast time series exchange rates is a study conducted by Peramunteillke and Wong [4]. They conducted a study to forecast exchange rate movements based on news headlines. These headlines became the input to their algorithm which will then predicted whether the currency pair would go up, remain steady or go down. Their approach is an attempt to categorically classify time series based on text instead of quantifiable data. Based on their method they generated a set of rules that would predict the likelihood of the movement of the currency. Their results are said to outperform other approaches and, in particular, much better results than random guessing.

Another novel approach to time series data forecasting was conducted by Hansen and Nelson [9]. Their study had two goals 1) Evaluate the performance of neural networks in the forecasting of time series data, and 2) Use of stacked generalization as a way to refine the process.

Another study by Ferreira, Vasconcelos and Adeadoto [3] used a novel approach to forecasting time series data. They utilized a method using a hybrid model using an artificial neural network (ANN) and a modified genetic algorithm (GA). Their goal was to use the GA to modify the network architecture and find the most relevant parameters that represented the series. The authors used their methods on four sets of data. Their results show that their system can boost the performance of time series prediction. When comparing their results with other methods they found that their method presented a superior performance in all the comparisons made. Note that the description of their TAEF method has been left out since it is beyond the scope of this project, but the point to make here is that there are 'exotic' methods to predict time series.

A study conducted by Nag and Mitra [10] is another hybrid method to forecast currency exchange rates. As before, they use a combination of neural network and genetic algorithm in the prediction. Their results, they conclude, are equally superior to traditional non-linear time series techniques and fixed-geometry neural network models.

A study by Vojinovic, Vecman and Seidel [2] discuss a data mining approach in the modeling and forecasting of financial time series. They used a Radial Basis Function Neural Network (RBF NN) model for forecasting the daily closing exchange rates of the US dollar and the New Zealand dollar. Their research reveals that the foreign exchange market is 'efficient.' and that the traditional statistical methods do not account for the nonlinearity of the volatility inherent the foreign exchange time series. Their research on neural networks, on the other hand, reveals that these models can be used for modeling and forecasting nonlinear time series such as the foreign exchange. In their experiments they chose the most studied and the most used autoregressive integrated moving average (ARIMA) model to compare it with the RBF NN.

There have been advanced methods to forecast time series data. One of these advanced studies was conducted by Hanias and Curties [1]. They used chaos theory to predict the Dollar/Euro exchange rate. In their study they used the method proposed by Grassberger-Procaccia. They conducted a non-linear analysis on the series and carried out the time series prediction experiments. They state that chaotic time series such as the Dollar/Euro exchange rate cannot be predicted in the long time, but can be in the short term. They carried out their experiments for 1, 7, 15, 30, 45, 60 time steps (days) ahead. Their results show that the greater the steps into the future, the greater the prediction error.

2.3 Data Mining

Data mining has been documented extensively in literature in recent years. Han and Kamber provide a textbook [21] in which concepts and techniques are explained extensively. They define data mining as the extraction of knowledge from large amounts of data. They treat knowledge discovery as a process consisting of seven steps: data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge presentation. For each of these steps, they provide techniques to carry out the tasks (input based on type of knowledge to be mined: association, classification and prediction, clustering, and sequential pattern and time-series mining).

For example, for the data mining task they provide algorithms to successfully carry out the data mining tasks. In other words, they treat data mining as one of the steps in the discovery of knowledge.

In his data mining textbook [27], Bramer lays out the principles of data mining in a simple manner. Even though his discussion of the subject is not as extensive as Han's textbook [21], he introduces the concepts of data mining in a plain and simple to understand language. He details methods to use in classification, clustering, and text mining. One particular area of interest is the chapter on estimating the predictive accuracy of any classifier. He describes three strategies that are commonly used: divide the data into a training set and test set, k-fold cross-validation and Nfold cross validation.

A third book worth mentioning in this survey is a book written by Cios, et al. [26]. As in Han's book [21], data mining is defined as a task in the knowledge discovery process. The knowledge discovery process is defined as the "non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." They introduce the CRISP-DM knowledge discovery process consisting of six steps: business understanding, data understanding, data preparation, modeling, evaluation and deployment.

In addition to numerous textbooks on the subject of data mining, several papers have been published.

One of these papers by Hand et al. defines data mining and describes its challenges, and problems [5]. In particular the paper points out one of the differences between statistics and data mining being the algorithm. In data mining the algorithm is central to the analysis of the data at hand. This is in contrast to statistics in which the model is the most important goal of the analysis. Hence, one of the challenges in data mining is to find efficient algorithms. Other challenges include data mining tools, data quality issues and the problem of size. Another problem in data mining process is the need for a standard process.

In [8], Clifton and Thuraisingham present an overview of data mining and discusses standards, existing or proposed. In particular they discuss the emerging CRISP-DM as the standard data mining process to follow in a data mining project. Of particular interest in this project is the data mining for financial applications and, in particular, the data mining of time series data of the foreign exchange market.

In [6], Kovalerchuk and Vityaev mention that data mining can be successful in the forecasting of short-term conditional patterns and trends and that retraining should be performed continuously in the data mining process of financial data.

Zhang and Zhou discussed in [7] that "data mining techniques have been used to uncover hidden patterns and predict future trends and behaviors in the financial markets."

2.4 FX Market

The Bank of International Settlements (BIS) reports in their results of their Triennial Central Bank Survey on global exchange market activity in April 2010 an average daily turnover of $4.0 trillion [20]. This makes the foreign exchange market the largest financial market in the world. One of the reasons for this increase from the 3.3 average turnover reported in 2007 was because of the more diverse group of financial institutions such as "non-reporting banks, hedge funds, pension funds, mutual funds, insurance companies and central banks." Another reason for this increase was that the "Foreign exchange market activity became more global, with cross-border transactions representing 65% of trading activity in April 2010, while local transactions accounted for 35%, the lowest share ever. “In the foreign exchange (FX) market currencies are traded. Businesses and individuals buy or sell other countries’ currencies when doing business in a particular country. Investors also speculate in the market. They bet whether a certain currency pair will increase or decrease in value. They make a profit when the buy a currency pair and the pair increase in price. They also make money when they sell a currency pair and the pair decreases in value. There are many currency pairs that are traded in the market. Some of the most active pairs reported by the BIS in their report is the U.S. dollar (USD) 1) against the euro (EUR) with a daily turnover of over 1.1 Trillion dollars, 2) against the Japanese yen (JPY), with a daily turnover of over 550 billion dollars, and 3) against the British sterling (GBP), with a daily turnover of 368 billion dollars.

Each of these currency pairs generates historical price data. Price data gets recorded at several intervals such as monthly, weekly, daily, hourly, etc. This type of data is one example of a time series. This data is easily available on the Internet. Companies and individuals can get the data for almost every conceivable currency pair traded in the FX market. They can download the data to their computers and analyze it in order to find some interesting patterns that they can use to make better business or trading decisions.

2.5 FX Data Mining

There have been numerous attempts to forecast exchange rates in the FX market using a range of techniques. These techniques range from simple statistical models to sophisticated mathematical models. In [1] the dollar/euro is analyzed using chaos theory by Hanias and Curtis. A creative approach is used in [4] by Peramunetilleke and Wong to investigate how the money market news headlines can be used to forecast intraday currency exchange rates movements. One of the most popular data mining techniques to forecast currency exchange rates in recent years has been the use of artificial neural networks (ANNs). Abu-Mostafa and Atiya in [14] use a neural network to demonstrate the use of hints in the forecasting of the dollar versus four other currencies (dollar/pound, dollar/German mark, dollar/yen, dollar/Swiss franc. In [11] and [12] a group currencies also forecasted using ANNs. These neural networks were trained using different training algorithms. In particular, in [12] the Bayesian learning was used to avoid the over-fitting of the network by Huang et al.

Newer techniques to forecast currency exchange rates involve the use of support vector machines. SVM based Models for predicting foreign exchange rates are used in [19] to investigate the effect of different kernel functions, namely, linear, polynomial, and radial. The study tries to predict six different exchange rates against Australian dollar. Finally, hybrid approaches to forecast the exchange rates are coming of age. One of these methods uses a hybrid artificial intelligence method based on a neural network and a generic algorithm to model daily exchange rates [10].

Chapter 3: Key Project Decisions

Many decisions had to be made to implement this project as shown in Figure 1.

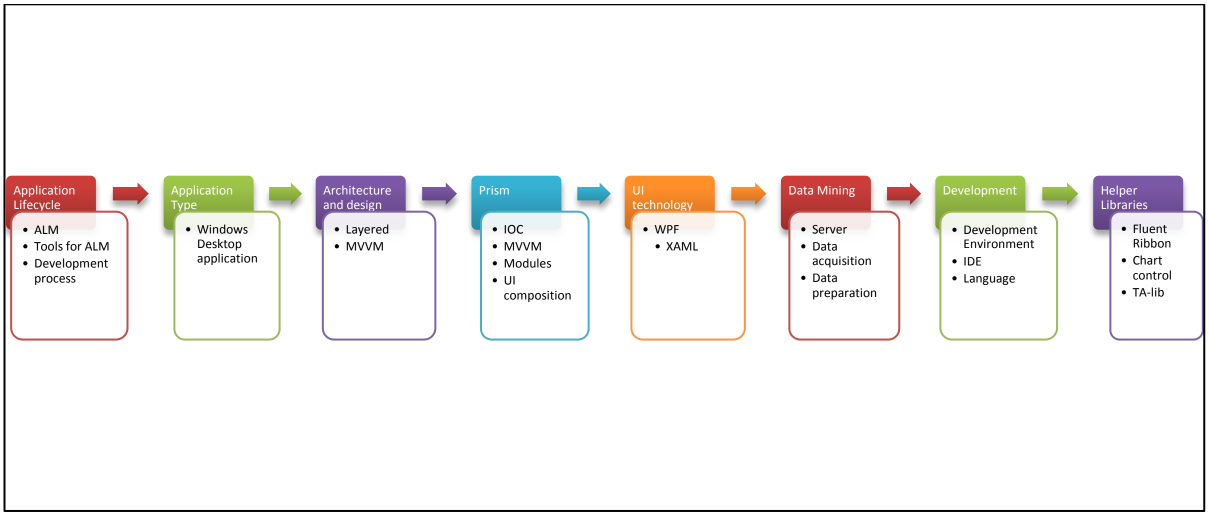


Figure 1: Key Project Decisions

3.1 Application Lifecycle Management

Understanding the software application lifecycle and its management was crucial to the implementation of the solution. Once it was understood, tools to manage it were selected to manage the overall project. One of the personal goals of this project was to understand and implement the whole process of building an application from inception to retirement.

An application is born with an idea. It’s an idea that needs to be implemented or developed in code. After the development is completed the application is deployed. After that, it can go through a series of maintenance or update stages. When the application serves no business objective or use, it comes to the end of its life. This is a description of the application lifecycle. The management and monitoring of which is known as Application Lifecycle Management (ALM).

As Chappels mentions in[31] ALM is not the same as a the software development lifecycle(SDLC). SDLC is only an aspect of ALM. As he describes ALM consists of three aspects:

1) **Governance:** encompasses all of the decision making and project management for this application, extends over this entire time.

2) **Development**: the process of actually creating the application happens first between idea and deployment and then may occur again whenever maintenance or updates needs arise.

3) **Operations**: the work required to run and manage the application, typically begins shortly before deployment, then runs continuously.

An illustration of these three aspects can be seen in Figure 2. As we can see, governance lasts from idea to the application's end of life cycle. The development of an application can consist of multiple SDLC's either because of maintenance or new releases. The operations aspect is important to manage and monitor the deployed application.

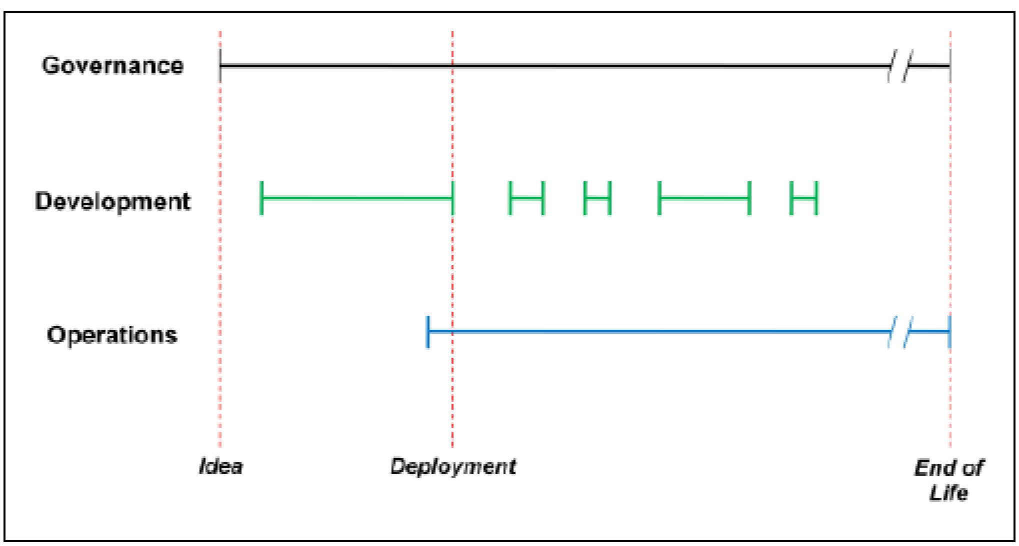


Figure 2: Application Lifecycle Management (ALM) [31]

3.2 Tools Used for ALM

3.2.1 Tools and Technologies for ALM

Visual Studio 2010 Ultimate Edition and Team Foundation Server 2010 were used to manage the application's lifecycle. Using these tools I was able to:

- Plan and track the project
- Design functionality
- Use version control to manage the source code
- Write code
- Test the application

3.3 Application Type

The application that I set out to build is a Windows desktop application. This decision was made in part because of the ubiquity of the desktop computer and the graphical capabilities in Windows 7 available in the Windows .Net 4.0 framework.

3.4 Architecture and Design

I built an application that was based on proven architecture design patterns. DMForex has a Layered Application consisting of the Presentation, Business, and Model layers. Modeling for the architecture was done using the Microsoft Visual Studio architecture feature.

DMForex architecture was chosen to provide two goals: a system that would be easily extensible and easily modifiable. The architecture chosen was an architecture where the application type and the technologies I wanted to learn played an important role. I needed to develop a desktop application to run on the Windows operating system.

To learn the most of this experience, I looked for an architecture style to base the design of the application on. I chose a layered architecture. This meant that the application would be divided in layers. Each layer would encompass a concern of the application. Finally I decided on what relevant technologies would be most useful in implementing the architecture. WPF was the technology of choice to implement the user interface.

At the end of the architecture design phase, I came across Prism. Prism is a framework to build modular applications using well adopted design patterns. For example, to separate the different layers of the application we can use the Model-View-ViewModel pattern to separate the presentation from the business logic.

3.5 Prism

Adopting Microsoft Prism 4.0 [29] to guide in the developing of the application using many design patterns involved many important decisions. The following are just a few of these decisions.

First it was decided to use the Prism Library because of the potential to build applications that were easily extend, modify, and test. Once the decision to use Prism was made, Managed Extensibility Framework (MEF) was chosen as the dependency injector container. This framework allows dependencies between classes be managed in one place, in the container. MEF is newer than Unity, and it’s part of .NET 4. Prism allows you to define application-specific services that can be shared among modules and register them in the container. The built-in service was chosen to keep a log of the activities of the application. Prism allows us to define modules to via explicit code declarations, code attributes on the modules discovered via directory scanning, configuration, or XAML.

Since this application is a desktop application, I decided to declare modules using explicit code.

Another important key decision was how to organize the solution. All prism applications have a Shell module. This module is in charge of bootstrapping or starting up the application. In addition to the Shell module, other modules developed were:

Common: contains project infrastructure used by other modules.

Analysis: contains the analysis models of the application.

Design: contains the design models of the application.

Shell: the main project of the application.

Prediction: this project contains the prediction objects and related objects.

Mining: contains the several type of mining algorithm objects.

Testing: contains classes related to the testing of the results of the predictions

Visualization: contains classes related to the visualization of the predictions.

Implementing MVVM involved making decisions on how to implement it consistently across the entire application. For example, we needed to decide how to connect the view to the view model. Another decision was made the commands from the view models as command objects and not as command methods. And finally, the decision to use the IDataErrorInfo interface to report errors to the view was adopted.

Another set of decisions had to be made when composing the user interface.

3.6 User Interface Technology

3.6.1 WPF

I chose the Windows Presentation Foundation (WPF) to build the graphical user interface (UI). WPF is an Application Programming Interface (API) for desktop applications for the .NET Framework. WPF offered many of the advantages I was looking for such as better graphic capabilities. But the most important reason was the capability to design user interfaces in a declarative manner using XAML.

3.6.2 XAML

The choice to use XAML came from the decision of using WPF. XAML is a language used to build user interfaces in code. It resembles the HTML and XML style of using tags to declare elements of the UI.

3.7 Data Mining

The choice of data mining tools and technologies was based on my desire to experiment with Microsoft SQL Server 2008 Enterprise edition. SQL Server 2008 comes with Analysis Services that include standard data mining algorithms such as the time series and classification trees algorithm chosen in this project to implement the predictions.

3.7.1 SQL Server 2008 Analysis Services

The data mining services provided by SQL Server 2008 are provided in a separate server. These services allow the creation of an analysis database in which data mining models can be created. Once created, these models can be deployed and used in applications or within the Microsoft SQL Server Management studio.

The Analysis server allows the creation of Analysis Management Objects (AMO) to programmatically implement the predictions of exchange rates and trends by creating the analysis database, mining structures, and mining models. These mining models are then queried through ADOMD.Net using Data Mining Extensions (DMX) commands to obtain the prediction of the exchange rate or trend.

3.7.2 Datasets Acquisition

There are many places on the Internet where currency exchange historical data can be downloaded. Some of the providers include it as part of their services to current or prospective customers. The data used for this project was obtained from FXCM. FXCM is a forex trading company that offers demo and real accounts to trade currencies in the FX market. There are many time intervals for each currency pair. The data used in this project is the daily historical price. The data was saved as a comma separated value (CSV).

3.7.3 Dataset Preparation

To prepare the data Microsoft Excel was used. The goal of the preparation was to format the data as shown in Table 1.

date	open	high	low	Close
10/1/1999	1.0679	1.0767	1.0636	1.073
10/4/1999	1.0714	1.0776	1.0689	1.0734
10/5/1999	1.0735	1.076	1.0669	1.0732
10/6/1999	1.0731	1.0779	1.0681	1.0689
10/7/1999	1.0688	1.0748	1.0661	1.071
10/8/1999	1.0711	1.072	1.0602	1.0631
10/11/1999	1.0616	1.0657	1.0608	1.0633

Table 1: Sample Historical Currency CSV File

Microsoft Excel was used to clean the data for anomalies. It was also used to sort the data in ascending order by date. It was important to keep the data in the date column with the date part only without the time. The headers were updated to match the order: date, open, high, low, and close. The file was then saved in a CSV format.

3.8 Development Environment

The software development setup consisted of two laptops and an instance of a virtual server in Amazon's Elastic Cloud Computing (EC2) as shown in Figure 3. Table 2 shows the hardware and software configuration of the machines used in the development of the project.

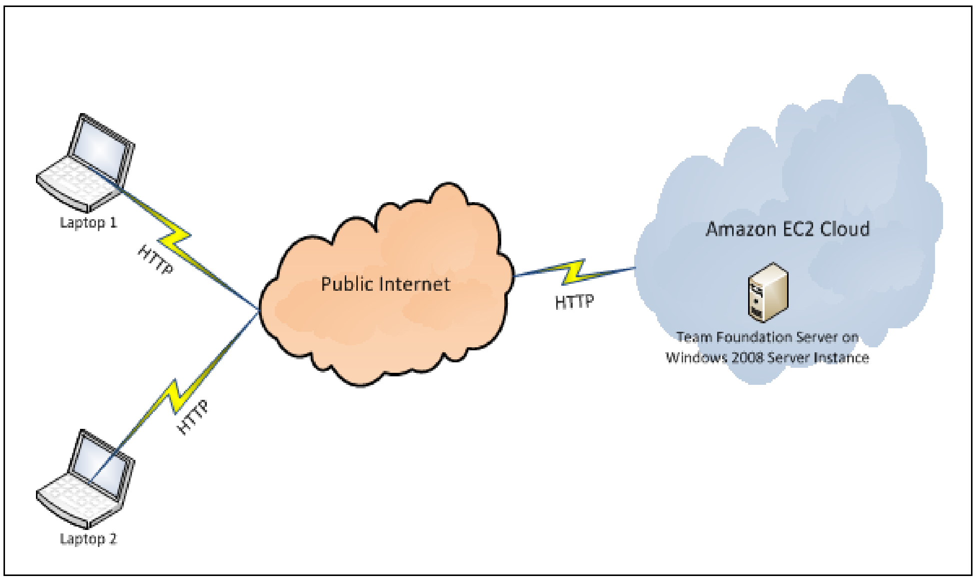


Figure 3: DMForex Software Development Environment

Laptop 1 was used to work on the project on the go over the Internet. It was usually used at school or at Starbucks to access the project locally offline or to access the TFS on the cloud. Laptop 2 was usually used at home. Its primary used was as a backup and test station. Finally, the virtual machine in EC2 was used to overcome the challenges faced when trying to run a full installation of TFS on laptop2. By having a full server operating system, some of the software and hardware requirements problems to install TFS and SharePoint were solved. For example, installing TFS on a client operating system, such as Windows 7, only installs the basic administration options and the ability to configure a SharePoint portal is disabled.

Machine	Hardware	Operating System	Software
Laptop 1	Gateway 3GB of RAM 32 bit	Windows 7 Professional	Visual Studio 2010 Ultimate
Laptop 2	Gateway model 4GB of RAM 64-bit	Windows 7 Professional	Visual Studio 2010 Ultimate
EC2 instance	Virtual	Windows Server R2 Datacenter	Visual Studio 2010 Ultimate Team Foundation Server 2010 SharePoint 2010

Table 2: Software Development Setup

3.8.1 Selection of Integrated Software Environment (IDE)

Microsoft's Visual Studio 2010 Ultimate edition was chosen as the IDE for the implementation of the product due to its many features. For example, the architecture modeling tools and team explorer capability made it an attractive environment to experiment with. In addition to the many development features, Visual Studio is the tool of choice with ALM. It was used to access the team project in the server.

3.9 Helper Libraries and Controls

I used some of the components available to help in some of the areas of the implementation. For example, I used the Fluent Ribbon library to provide the user the familiar Ribbon feature available in the latest Microsoft Office products. I also customized a chart control. To help in the implementation of calculating technical indicators data to modify the original dataset I used the TA-lib library.

3.9.1 Fluent Ribbon

To provide the user with a centralized command center, I used the Fluent Ribbon Control Suite. This library implements an Office-like user interface for the Windows Presentation Foundation (WPF). The RibbonWindow control replaces the typical Windows form. This window could be customized using controls such as RibbonTabControl, Backstage, Gallery, QuickAccessToolbar, and ScreenTip.

3.9.2 Chart Control

To visualize historical data or results of the predictions I implemented a chart control. As it stands the control allows the graphing of only one time series dataset. In order to meet the requirements of the applications, I needed to chart both, the historical data and the actual value of the predictions. For the rates, only the next value was charted. For the trend, there were four possible lines that were drawn one for each of the four possible trend predictions.

3.9.3 TA-Lib

TA-lib is a technical analysis open-source software library that provides approximately 200 technical indicators. The use of the TA-Lib allowed me to focus on the implementation of the building of the new data set used for predicting the trend, instead of building the technical indicators from scratch. Some of the indicators calculated from the library are the Relative Strength Index (RSI) and the Simple Moving Average (SMA).

Chapter 4: DMForex Implementation

4.1 Overview

This chapter presents the implementation of the overall analysis and design of all DMForex components. To accomplish the goals of the application, the actual work was divided in two phases. One phase dealt with managing the application lifecycle (ALM). The second phase dealt with the actual development of the application. I differentiated between the two since managing a software project and implementing it are two separate matters.

The first phase consisted of installing and configuring all the necessary tools and technologies, such as ALM tools, IDE, and SQL server, and Prism.

The second phase dealt with the actual construction of the application. It consisted of three iterations as shown in Figure 4. During Iteration 1, the overall architecture of the application was implemented. This is where the architectural foundations were put in place. In Iteration 2 the predicting of rates was developed. Finally, during Iteration 3 the prediction of trends was implemented. The following sections will describe these two phases in more detail.

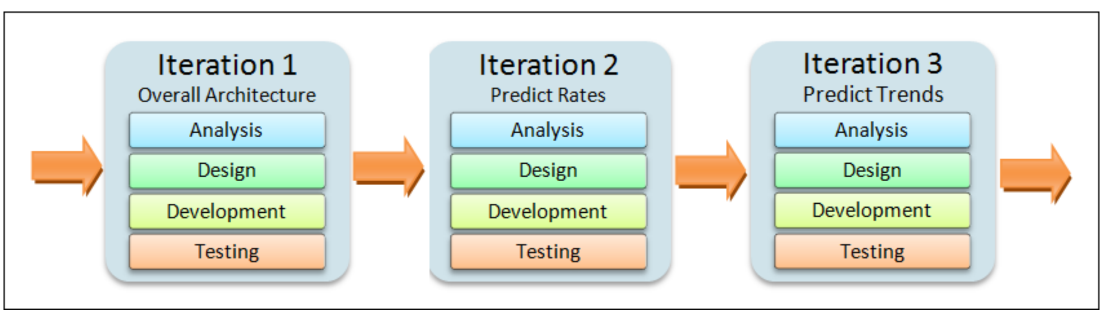


Figure 4: DMForex Iterations

4.2 Phase 1

4.2.1 DMForex Lifecycle Management

The lifecycle of the application was managed with Team Foundation Server. A virtual server was instantiated in Amazon's cloud computing (EC2) service in order to make the source code available when I was away from home or my laptop and needed to work on the project. TFS version control's capabilities allowed me to work virtually from anywhere a connection to the Internet and a browser was available. TFS allowed me to keep track of work through its work item tracking (WIT) features, but the use was minimal, since I would be using my own system of writing notes to keep track of what I needed to do. I found that tracking work through TFS was more than what I needed since I was the only programmer in the project. The WIT feature of TFS can be very useful when a group of programmers collaborate in a team project.

4.2.2 Infrastructure Setup

Before the actual development of the application I needed to do some preliminary work. In particular, I needed to install the tools and technologies that were needed both remotely in Amazon's EC2 and on my laptops.

I setup an instance of a Windows Server 2008 in EC2 in which to run:

- Visual Studio Team Foundation Server 2010
- SharePoint 2010
- SQL Server 2008 with Analysis Services

On my two development laptops and the EC2 environment I installed:

- Visual Studio 2010 Ultimate Edition

- SQL Server 2008 with Analysis Services

In addition to the servers and IDE, following tools and technologies were obtained and installed:

- Prism 4.0
- TA-Lib
- Blend

The development environment was ready and the development of the software began with Iteration 1.

The implementation of DMForex is done entirely in Microsoft Studios Ultimate Edition 2010. It is written in C#. It uses a MVVM pattern to separate the concerns. The graphical classes are represented by a View class, and their corresponding business logic is implemented in a View Model class. The application also uses a shell in which other modules can have their views in regions. For example, a module to view the results is in charge of graphically display the results to the user. The module is just a C# class library which holds the chart classes.

In order to calculate trends, the program uses TA-lib, an open source library to calculate technical analysis indicators. These indicators are used to transform the original time series to calculate more attributes. The main idea is to data mine the data series using the most popular technical indicators just as a trader would when making a decision to enter a trade.

4.3 Phase 2

In this part of the project, I implement all of the tasks involved in developing DMForex. It consists of three iterations. In the first iteration the overall architecture of the application is implemented. The second and third iterations deals with the implementation of the prediction of rates and trends use cases, respectively

4.3.1 Overall Architecture (Iteration 1)

During this iteration the development of the overall architecture was put in place. A Prism 4 application was created in Visual Studio.

4.3.1.1 Analysis

The goals of this iteration were to come up with a software architecture that would serve as the foundations of the solution to our problem and a partial software implementation of it. These would be the decisions that would be hard to change later in the development process.

First, the scope of the project was identified as shown in Figure 5. The system had no external interactions and there would be at most one user interacting with the application. The main constraint was that the prediction of trends would have to be calculated with data mining techniques and algorithms as opposed to classical statistical forecasting techniques.

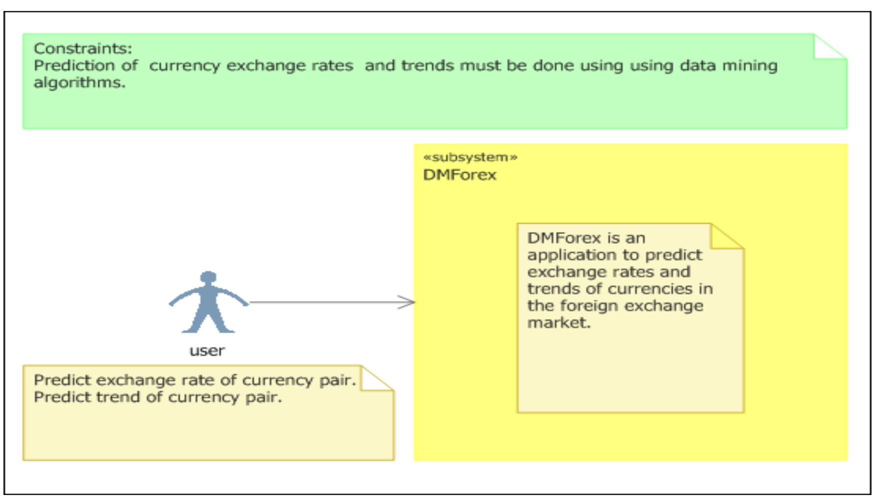


Figure 5: Project Scope Use Case Diagram

4.3.1.2 Design

The application follows a layered-architecture structure as shown in Figure 6. Each layer has a responsibility to deal with a particular concern or aspect of the application. For example the UI Layer is responsible to keep the user interface components separate from the business logic. That business logic is delegated to the Business Layer.

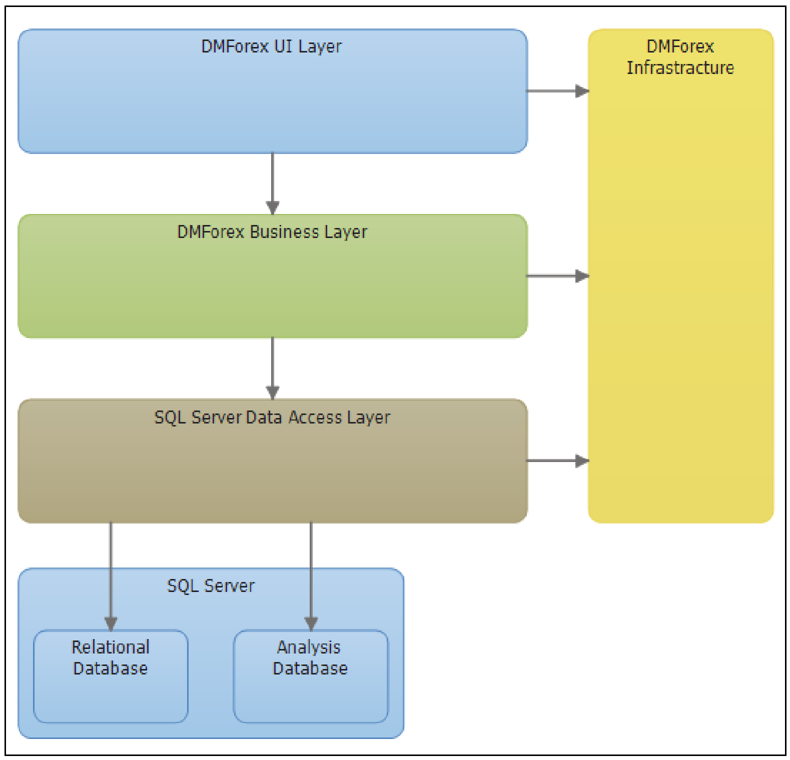


Figure 6: High Level Architectural Design

4.3.1.3 Development

The following tasks were carried out to create a Prism 4 application to implement the architecture and solution.

1. Created a Team Project in Visual Studio
2. Add New Solution To Team Project in TFS
3. Add Solution to Source Control

4. Setup Prism libraries
5. Setup the application Shell
6. Added modules to application

DMForex is an application based on the Prism 4 framework. Prism requires a Bootstrapper class to initialize the different services required to build composite, extensible applications such as DMForex. It is during this bootstrapping process that the application is initialized with all the required services. The framework requires a root object to be the holder the primary user interface content. This is accomplished by creating a shell. This root object is typically a Windows object. However, DMForex uses a RibbonWindow from the Fluent library. The Ribbon is configured in XAML and its main purpose is to be a centralized location for the applications commands and actions replacing the typical menu bar in a Windows application. A Prism application also requires a shell to define regions in which views are allocated a portion of the user interface to display their content. It is the composition of all the views in the shell that make the application's user interface. The DMForex shell consists of only one region as shown in Figure 7.

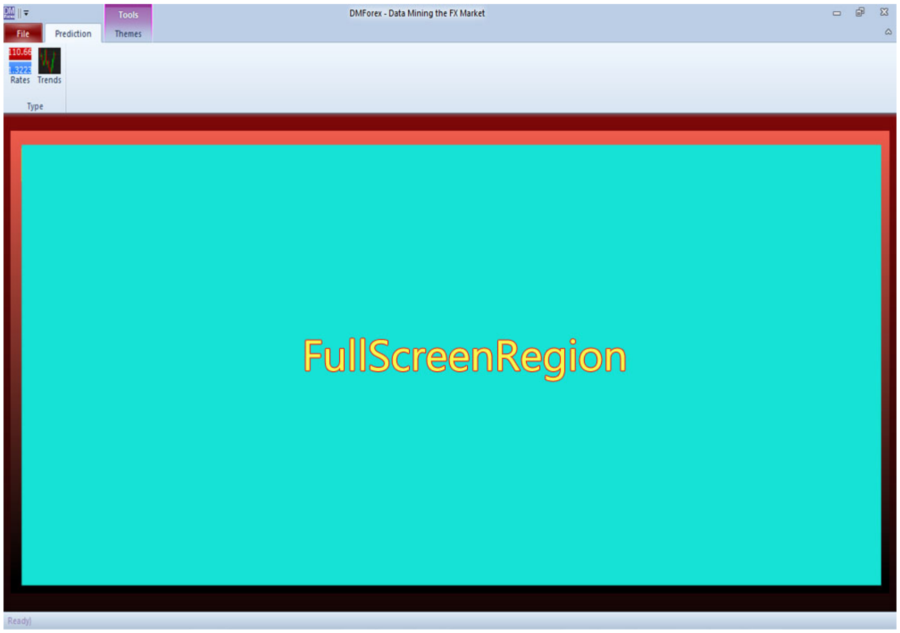


Figure 7: Shell Region

It is in the FullScreenRegion where the other views would be displayed. The main screen of the application is loaded from HomeView. When the Boostrapper completes its job of initializing the application, the HomeView is shown as in Figure 8.

Figure 8: DMForex Main Screen

4.3.1.3.1 Creation of Necessary Modules

After the application was setup and the shell displayed, several projects were added to the solution to implement the functionalities in the following two iterations. The following projects were added to implement the prediction functionality of the application:

- **Common:** contains project infrastructure used by other modules.
- **Shell:** the main project of the application.
- **Prediction:** this project contains the prediction objects and related objects.
- **Mining:** contains the several type of mining algorithm objects.
- **Testing:** contains classes related to the testing of the results of the predictions.
- **Visualization:** contains the classes related to the visualization of the results of the predictions.

Two additional modules were added to design and model the application:

- **Analysis:** contains the analysis models of the application.
- **Design:** contains the design models of the application.

The details of what went into each module will be described later when I describe the implementation of the use cases. Figure 9 shows the created modules in VS 2010 IDE.

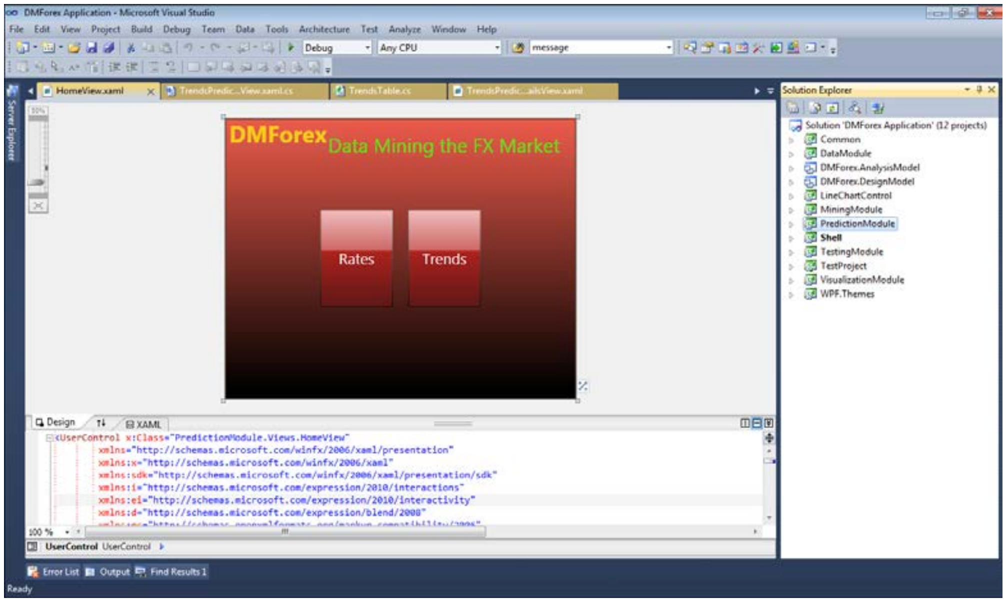


Figure 9: Modules Created in Visual Studio 2010

4.3.1.4 Testing

At this early stage of the development of the application, a test project was added to the solution to unit test the code as it is developed.

4.3.2 Predicting Currency Rates (Iteration 2)

4.3.2.1 Analysis

When predicting rates we are trying to accomplish 4 things in terms of functionality to the user using a historical file as the input:

1. We are trying to predict the next rate of a currency pair given a historical file.
2. We are trying to view the results in textual form.
3. We are trying to view a chart of the results with historical data and predicted values.

4. Finally, we are given the option to test the accuracy of past predictions to give the user an idea how trustworthy the next predicted value is.

Figure 10 shows the use case diagram for this iteration.

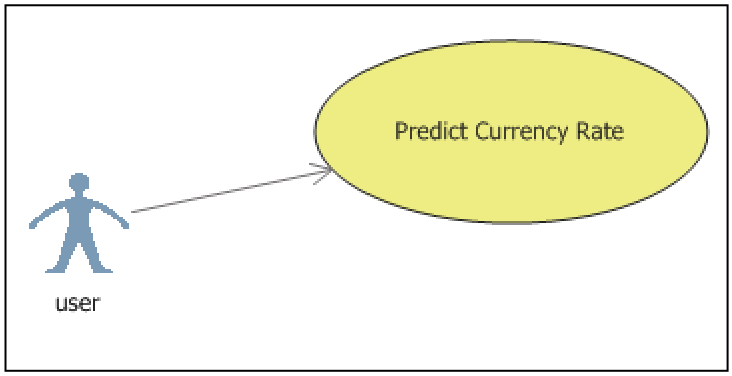


Figure 10: Predict Currency Rate Use Case

4.3.2.2 Design

To achieve this use case, we needed to design the classes that would be used to implement the use case. First we needed a class for the Rate Prediction itself. We then needed a Mining Model appropriate to mine single valued time series. Data mining algorithms provided by SQL Server Analysis Services were used to implement the mining model. The input dataset to the algorithm was captured by loading a CSV file. The presentation of the results to the user was done in two classes RatesPredictionResultsView and RatesPredictionChartView. The third view at the bottom of the screen is the RatesPredictionTestingView . The design of the user interface for this iteration can be seen in Figure 11.

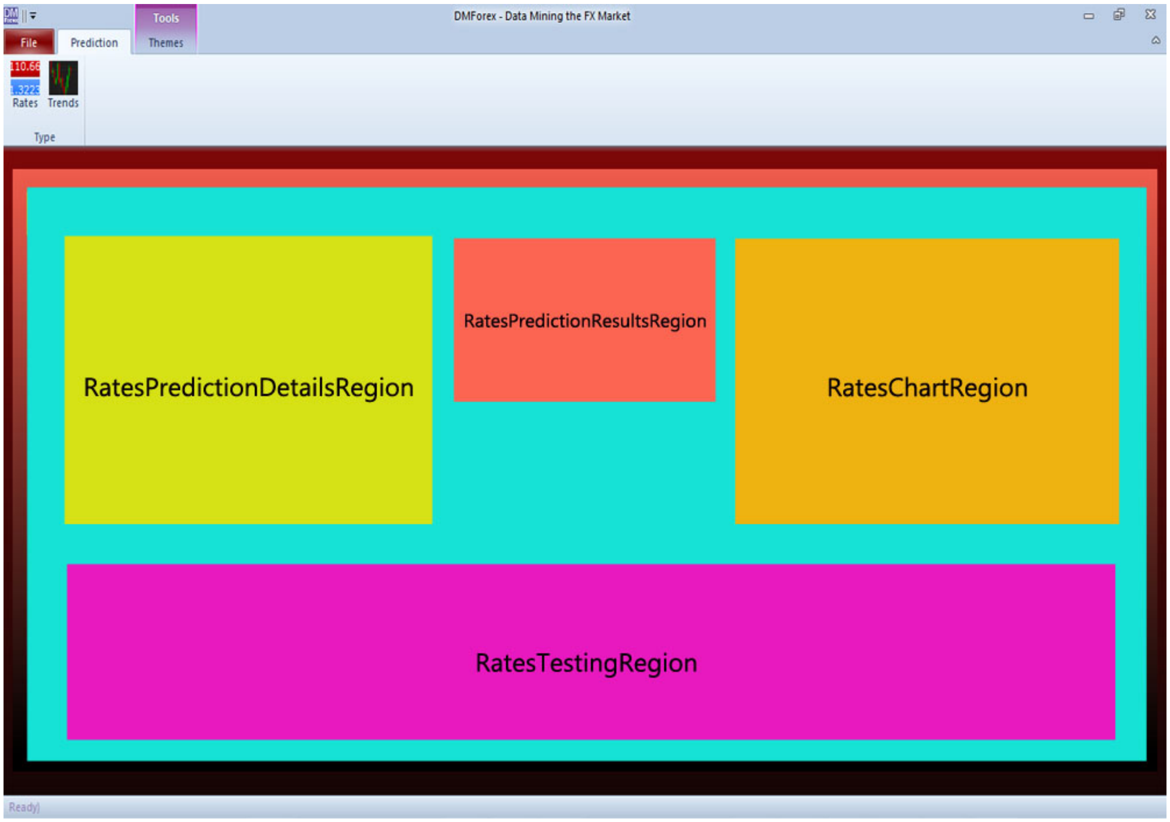


Figure 11: Rates Prediction View with Regions

4.3.2.3 Development

The implementation of this use case was spread out into the several modules that make up the application, mainly the Prediction , the Mining , the Visualization, and the Testing modules. In the Prediction module I created a RatesPredictionDetailsView and its corresponding view model, RatesPredictionDetailsViewViewModel. The purpose of the view was to allow the user to select a CSV file to be used as input. The job of its viewmodel was to create a TimeSeriesMiningModel and display the RatesPredictionResultsView, RatesPredictionChartView, and the RatesPredictionTestingView.

The TimeSeriesMiningModel is a class that is part of the Mining module. It uses Analysis Management Objects (AMO) to build a mining model using the Microsoft Time Series algorithm. The class accomplishes its job by:

- Connecting to an analysis server
- Creating an analysis database
- Creating a mining structure
- Creating a Time Series mining model, and
- Processing the database

The RatesPredictionResultsView is a view from the VisualizationModule that shows the results of the forecasted exchange rate. This is accomplished by having its viewmodel query the mining model created in the previous step. The RatesPredictionResultsViewModel gets a prediction name parameter injected by the view. It then obtains the next exchange rate by using the Data Mining Extensions (DMX) SELECT statement assigned to a CommandText's command object:

```
cmd.CommandText = "SELECT FLATTENED PredictTimeSeries
    ([FX Time Series Model].[Close],1)" + "AS PredictClose From
    [FX Time Series Model]";
```

The result of this query is just a single value and it is assign to the Rate property of the RatesPredictionResultsView.

After the results are shown, the RatesPredictionView injects a chart into the RatesChartRegion. At this time, an instance of the chart control is created and displayed as a line chart for the original dataset and as a point for the predicted value. As soon as the chart view is shown, the user sees the bottom region. This is where the RatesPredictionResultsTestingView is shown by view injection. The objective of this view is to back test the historical data to determine the accuracy and the effectiveness of the mining model. For each day between the From and To input text fields, a mining model is created. It processes the model for as many days between the provided dates. A temporary database table was created in the SQL server to hold a subset of the original dataset. This allowed the next predicted value to be calculated as if the original dataset contained records up to that date. When the processing finishes, the results is shown in the grid. In the test results grid, the prediction is shown against the predicted value. It also calculates the difference between the actual and the predicted value. It then shows the actual and percentage difference between the results. Figure 12 shows a screenshot of the Rates Prediction screen.

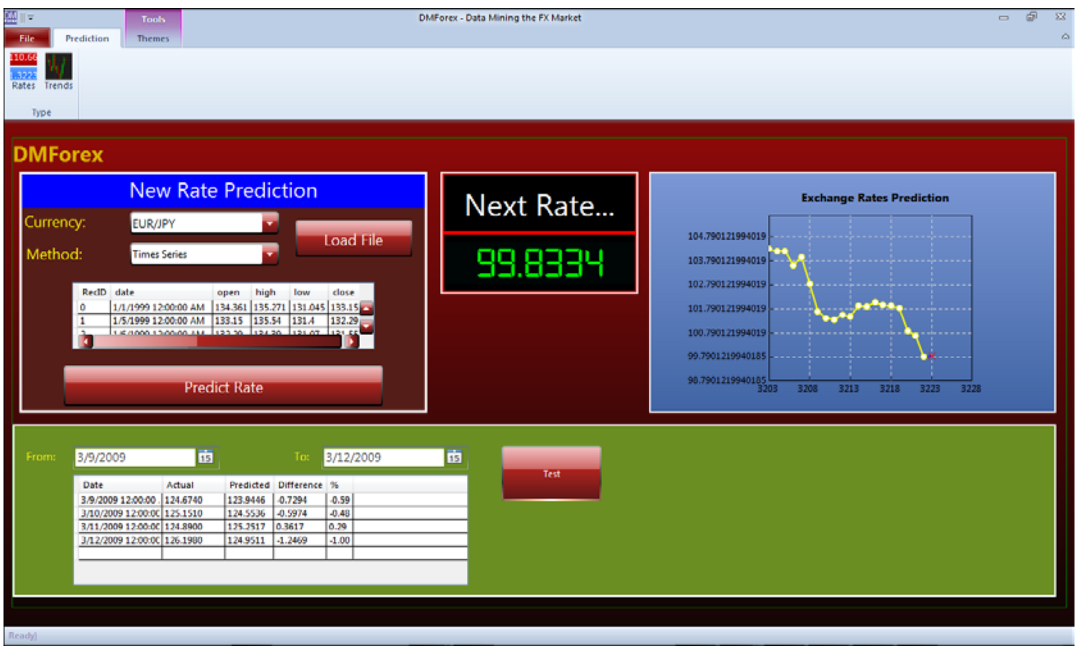


Figure 12: DM Forex Rates Prediction Screen

4.3.2.4 Testing

Unit tests were added to the solution's TestProject to test the methods in the several classes in this iteration. However, the actual testing was performed manually. For example, several currency files were loaded and verified that the data had been loaded correctly. The value in the predicted chart area was also compared to the value in the Next Rate window to be the same. The testing area at the bottom of the screen was also tested. In particular, days in the weekend such as Saturday and Sunday dates were tested to make sure that no predicted value would be generated. For example, a long date range beginning in the middle of the week and ending on a Friday the week such as 6/16/2010 to 6/25/2010 for the EUR/USD skipped Saturday (6/19) and Sunday (6/20).

4.3.3 Predicting Currency Trends (Iteration 3)

4.3.3.1 Analysis

When implementing the predict currency trends use case we are trying to accomplish the same four things as in the prediction of rates use case:

1. We are trying to predict the next trend of a currency pair given a historical file.
2. We are trying to view the results in textual form.
3. We are trying to view a chart of the results with historical data and predicted value.
4. Finally, we are given the option to test the accuracy of past trend predictions to give the user an idea how trustworthy the next predicted trend.

Figure 13 shows the use case diagram for this iteration.

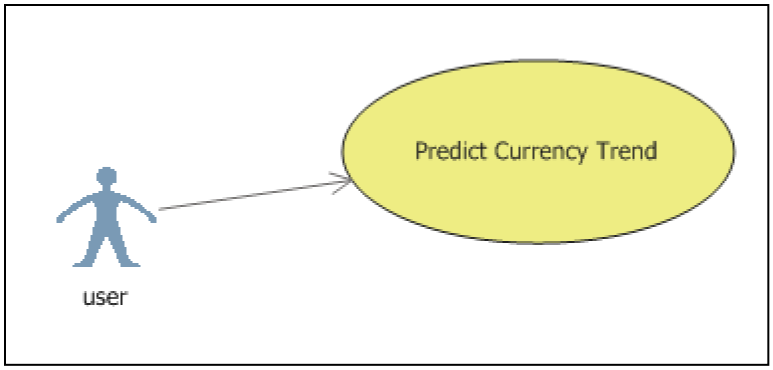


Figure 13: Predict Currency Trend Use Case

4.3.3.2 Design

The design of the trend prediction use case involved the creation of the following views and viewmodels in their respective modules. The design of the user interface for this iteration can be seen in Figure 14.

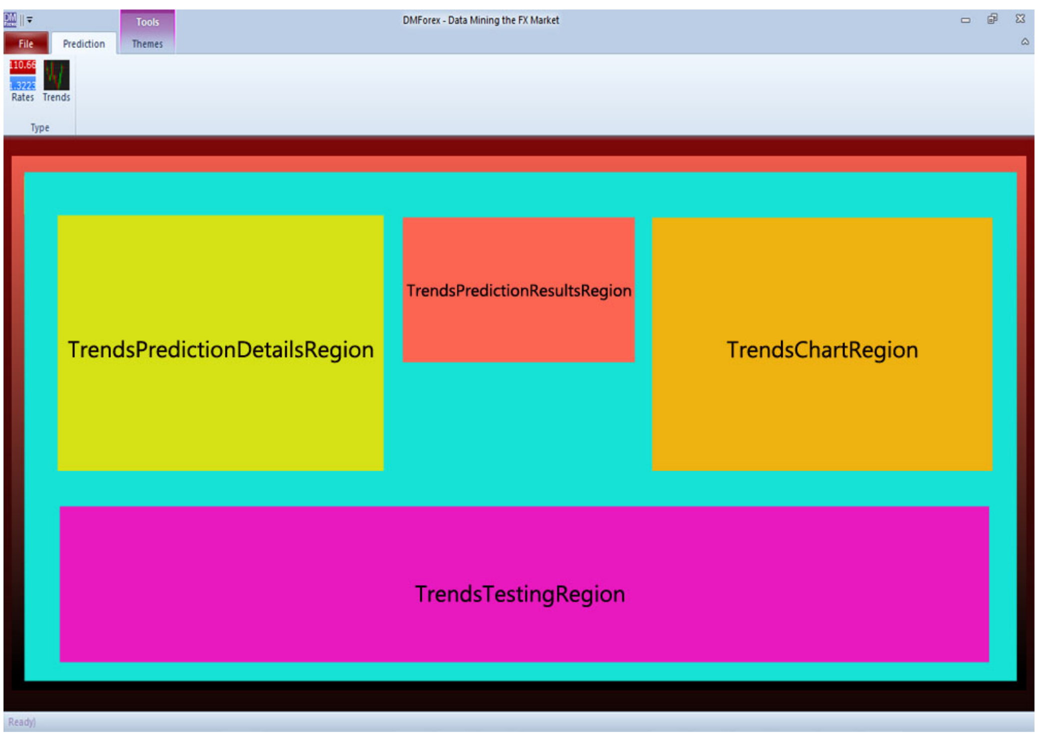


Figure 14: Trends Prediction View with Regions

4.3.3.3 Development

The implementation of predicting trends was very similar to the prediction of rates. The classes, views, and viewmodels required to implement this use case was also spread out into the several modules that make up the application, mainly the PredictionModule, the MiningModule , the VisualizationModule, and the TestingModule. In the PredictionModule I created a TrendsPredictionDetailsView and its corresponding viewmodel, TrendsPredictionDetailsViewModel. As with its rate counterpart, the purpose of the view was to allow the user to select a CSV file to be used as input. However, in this view, the UI had to change to accommodate the modification of the dataset to be used as input. A section of the view contained trend definition parameters. For example, the user is allowed to enter how many days and pips determine a trend. These values were used in the viewmodel to calculate the trend attribute of the dataset. There were four possible values for the attribute field:

- UP
- DOWN
- Both
- Unchanged

The UP trend classification was calculated by looking ahead x number of days for the maximum value of the high column and the specified number of pips.

The DOWN trend is calculated similarly to the UP trend but looking at the minimum value of the next x close values and the specified number of pips.

If both of the UP and DOWN criteria are met, then the trend field for the row is marked as Both

Finally, the Unchanged trend is when the row does not meet the UP, DOWN, or Both criteria.

In addition to the trend information, the UI was modified to capture the technical indicator parameters necessary to create the Relative Strength Index (RSI) and the Simple Moving Average (SMA) indicators using the TA-lib.

Once this information was calculated, the TrendsPredictionDetailsViewModel created a DecisionTreesMiningModel using the newly constructed dataset as input. It then displayed the following views in its corresponding regions:

- TrendsPredictionResultsView
- TrendsPredictionChartView, and
- TrendsPredictionTestingView.

By adding the trend attribute to the dataset, the problem was transformed to a classification problem in which the next trend can be predicted by a decision trees algorithm. The ClassificationTreesMiningModel class was added to the MiningModelModule. It uses AMO to build a mining model using the Microsoft Classification Trees algorithm. The class accomplishes its job by:

- Connecting to an analysis server
- Creating an analysis database
- Creating a mining structure
- Creating a Decision Trees mining model, and
- Processing the database

The TrendsPredictionResultsView is a view from the VisualizationModule that shows the results of the forecasted trend. The TrendsPredictionResultsViewModel gets a prediction name parameter injected by TrendsPredictionDetailsView. It then obtains the next exchange rate by using the model's GetPredictions() method. The GetPredictions() method is implemented in the ClassificationTreesMiningModel. This method uses the Data Mining Extensions (DMX) SELECT statement assigned to a CommandText's command object:

```
        cmd.CommandText = "SELECT Predict([Classification Trees Model].[Trend]
)
FROM [Classification Trees Model]" +" NATURAL PREDICTION JOIN " + "
            (   SELECT " + high + " AS [high]," + low +
 AS [low]," + close + "

As [close], " + rsi + " AS [rsi]," + sma + " AS [sma]" + ") AS T ";
```

The result of this query is just a single value and it is assigned to the Trend property of the TrendsPredictionResultsView.

After the results are shown, the TrendsPredictionView injects a chart into the TrendsChartRegion. At this time, an instance of the chart control is created and displayed as a line chart for the original dataset and as one of four lines (slanted up or down, horizontal or vertical) for the predicted trend. As soon as the chart view is shown, the user sees the bottom region. This is where the TrendsPredictionTestingView is shown by view injection.

The objective of this view is to back test the historical data to determine the accuracy and the effectiveness of the mining model. For each day between the From and To date, a DecisionTreesMiningModel is created. It processes the model for as many days in between the provided dates. A temporary database table was created in the SQL server to hold a subset of the original dataset. This allowed the next predicted value to be calculated as if the original dataset contained records up to that date. When the processing finishes, the results is shown in the grid. In the test results grid, the prediction is shown against the predicted value. It then indicates if these two values match. It then shows the actual and percentage difference between the results. Figure 15 shows a screenshot of the Trends Prediction screen. As can be seen the line in the chart region is pointing down into the future due to the DOWN trend.

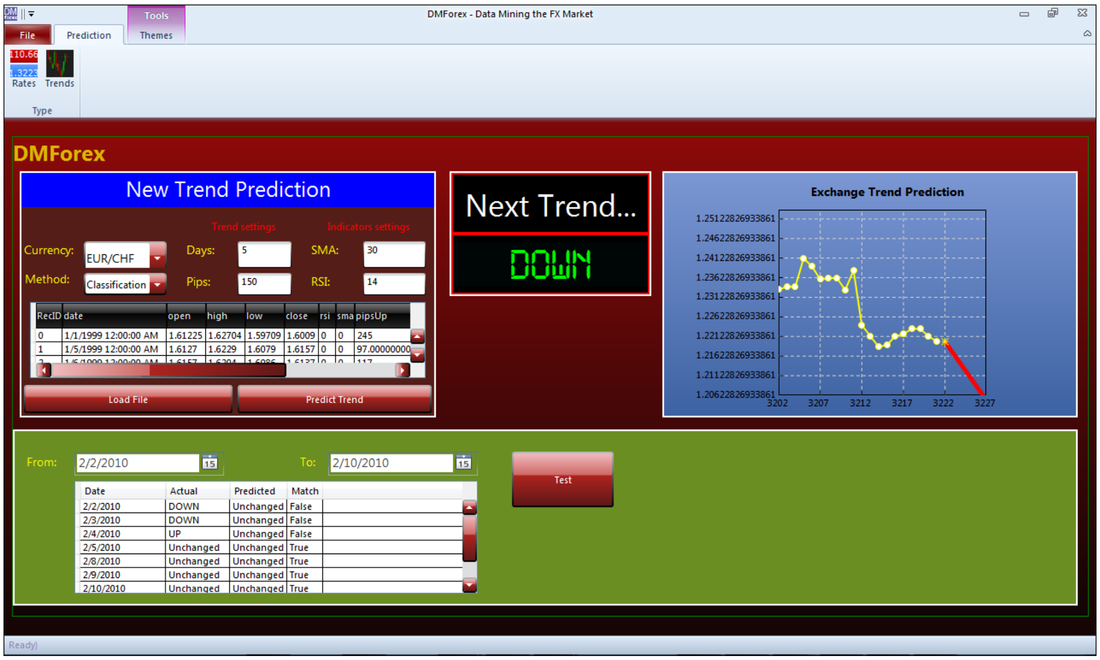


Figure 15: DMForex Trends Prediction Screen

4.3.3.4 Testing

Testing during this iteration was conducted to make sure the software behaved as it should. The input errors were minimized by having the details of the prediction performed data validation using Prism's IDataErrorInfo interface.

Testing was performed on the calculation of the trend attribute to make sure that the maximum and minimum values to predict the trend column was also accurate.

Finally, the chart was verified to make sure it plotted the last points in the data series accurately along with the predicted trend line.

Chapter 5: Experiments and results

Two types of experiments were designed to evaluate the effectiveness of DMForex in predicting rates and trends. The first type was setup to test the accuracy of the rates mining model. The other set tested the accuracy of the trends. The currency pairs shown in Table 3 are three of the seven major currency pairs with the biggest turnover as reported by the Bank of International Settlements. Each daily historical price data from 01/01/1999 to 12/31/2011 (12 years) have been obtained and saved in a comma-delimited file. The data has been formatted as required by the program.

Currency Pair
EUR/USD
EUR/JPY
USD/JPY

Table 3: Currencies Used in Experiments

5.1 Rates Experiments

5.1.1 Experiments Set

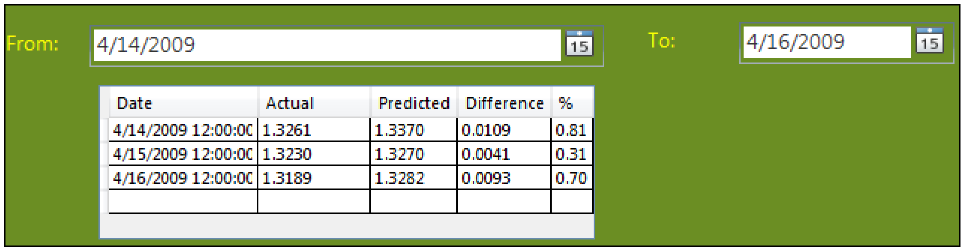
These experiments tested the prediction of rates of three of the major currencies displayed in Table 3 with the parameters shown in Table 4.

Experiment	From Date	To Date
1	4/14/2009	4/16/2009
2	10/19/2010	10/21/2010
3	12/13/2011	12/15/2011

Table 4: Rates Experiments Set Parameters

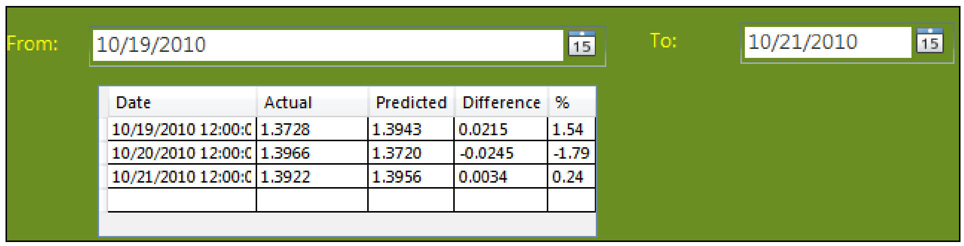
5.1.2 Rates Experiments Results

- **EUR/USD**



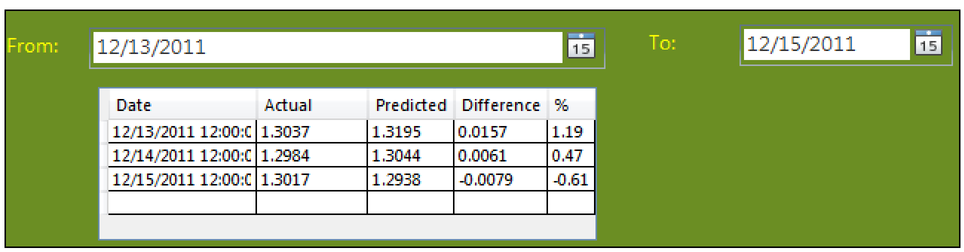
Date	Actual	Predicted	Difference	%
4/14/2009 12:00:0C	1.3261	1.3370	0.0109	0.81
4/15/2009 12:00:0C	1.3230	1.3270	0.0041	0.31
4/16/2009 12:00:0C	1.3189	1.3282	0.0093	0.70

Figure 16: EUR/USD Rates Experiment 1 Results



Date	Actual	Predicted	Difference	%
10/19/2010 12:00:C	1.3728	1.3943	0.0215	1.54
10/20/2010 12:00:C	1.3966	1.3720	-0.0245	-1.79
10/21/2010 12:00:C	1.3922	1.3956	0.0034	0.24

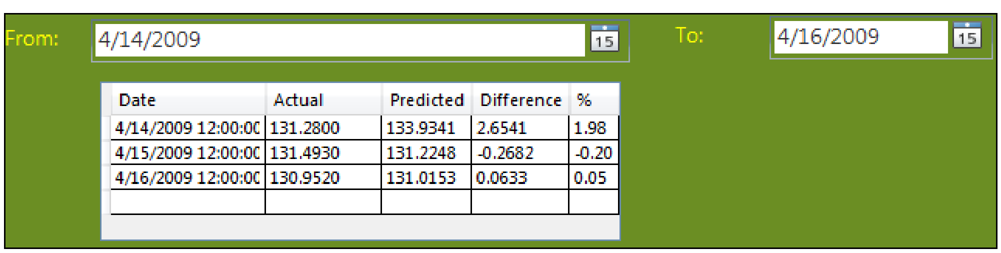
Figure 17: EUR/USD Rates Experiment 2 Results



Date	Actual	Predicted	Difference	%
12/13/2011 12:00:C	1.3037	1.3195	0.0157	1.19
12/14/2011 12:00:C	1.2984	1.3044	0.0061	0.47
12/15/2011 12:00:C	1.3017	1.2938	-0.0079	-0.61

Figure 18: EUR/USD Rates Experiment 3 Results

- **EUR/JPY**



Date	Actual	Predicted	Difference	%
4/14/2009 12:00:0C	131.2800	133.9341	2.6541	1.98
4/15/2009 12:00:0C	131.4930	131.2248	-0.2682	-0.20
4/16/2009 12:00:0C	130.9520	131.0153	0.0633	0.05

Figure 19: EUR/JPY Rates Experiment 1 Results

From: 10/19/2010 To: 10/21/2010

Date	Actual	Predicted	Difference	%
10/19/2010 12:00:C	112.0140	113.5101	1.4961	1.32
10/20/2010 12:00:C	113.2570	112.4012	-0.8558	-0.76
10/21/2010 12:00:C	113.2420	113.6199	0.3779	0.33

Figure 20: EUR/JPY Rates Experiment 2 Results

From: 12/13/2011 To: 12/15/2011

Date	Actual	Predicted	Difference	%
12/13/2011 12:00:C	101.7020	102.4246	0.7226	0.71
12/14/2011 12:00:C	101.4050	100.5743	-0.8307	-0.83
12/15/2011 12:00:C	101.3530	102.6837	1.3307	1.30

Figure 21: EUR/JPY Rates Experiment 3 Results

- **USD/JPY**

From: 4/14/2009 To: 4/16/2009

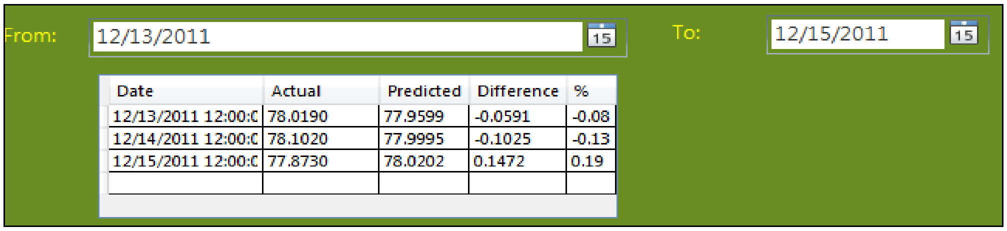
Date	Actual	Predicted	Difference	%
4/14/2009 12:00:0C	99.0080	100.1715	1.1635	1.16
4/15/2009 12:00:0C	99.4090	99.2341	-0.1749	-0.18
4/16/2009 12:00:0C	99.3050	100.1450	0.8400	0.84

Figure 22: USD/JPY Rates Experiment 1 Results

From: 10/19/2010 To: 10/21/2010

Date	Actual	Predicted	Difference	%
10/19/2010 12:00:C	81.6000	81.7402	0.1402	0.17
10/20/2010 12:00:C	81.1070	81.8783	0.7713	0.94
10/21/2010 12:00:C	81.3490	81.6933	0.3443	0.42

Figure 23: USD/JPY Rates Experiment 2 Results



Date	Actual	Predicted	Difference	%
12/13/2011 12:00:C	78.0190	77.9599	-0.0591	-0.08
12/14/2011 12:00:C	78.1020	77.9995	-0.1025	-0.13
12/15/2011 12:00:C	77.8730	78.0202	0.1472	0.19

Figure 24: USD/JPY Rates Experiment 3 Results

5.1.3 Rates Experiments Results Analysis

There were a total of 27 rates predicted. The difference between the actual value and the predicted value in percentage is shown in Table 5.

Result	%
1	-1.79
2	-0.83
3	-0.76
4	-0.61
5	-0.20
6	-0.18
7	-0.13
8	-0.08
9	0.05
10	0.17
11	0.19
12	0.24
13	0.31
14	0.33
15	0.42
16	0.47
17	0.70
18	0.71
19	0.81
20	0.84
21	0.94
22	1.16
23	1.19

Table 5: Percentage Difference between Actual and Predicted Rate

24	1.30
25	1.32
26	1.54
27	1.98

Table 5: Percentage Difference between Actual and Predicted Rate (Continued)

5.2 Trends Experiments

5.2.1 Experiments Set

This experiment set tested the trends of the currencies in Table 3 with the parameters shown In Table 6.

Experiment	From Date	To Date	Days in trend	Pips in trend	SMA periods	RSI Periods
1	4/14/2009	4/16/2009	5	25	30	14
2	10/19/2010	10/21/2010	10	100	30	14
3	12/13/2011	12/15/2011	20	250	30	14

Table 6: Trends Experiments Set Parameters

5.2.2 Trends Experiments Results

- **EUR/USD**

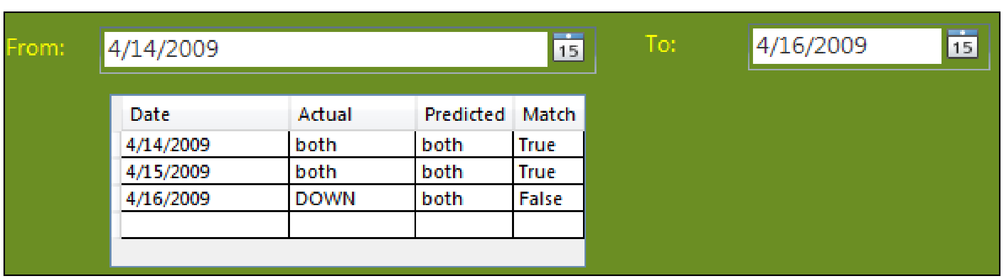
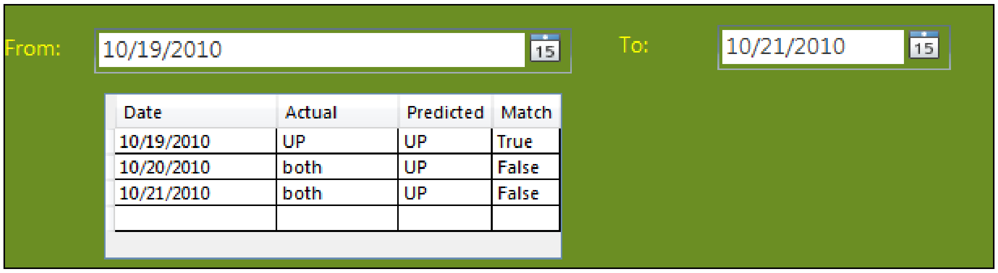
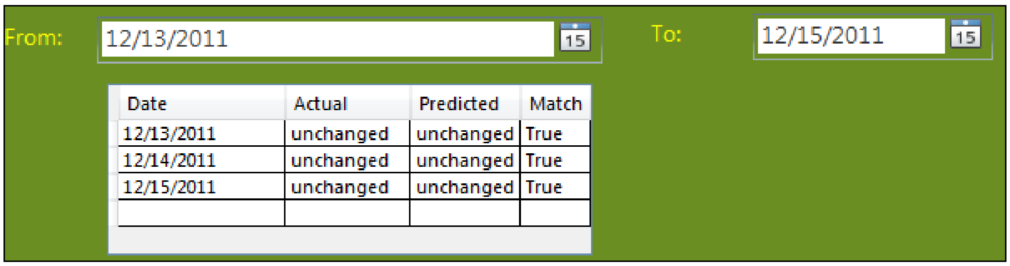


Figure 25: EUR/USD Trends Experiment 1 Results



Date	Actual	Predicted	Match
10/19/2010	UP	UP	True
10/20/2010	both	UP	False
10/21/2010	both	UP	False

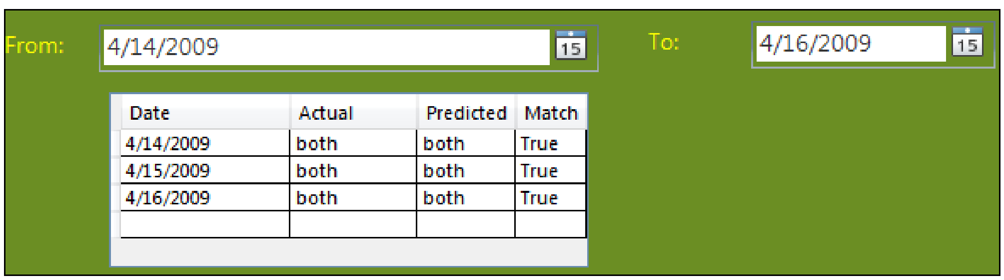
Figure 26: EUR/USD Trends Experiment 2 Results



Date	Actual	Predicted	Match
12/13/2011	unchanged	unchanged	True
12/14/2011	unchanged	unchanged	True
12/15/2011	unchanged	unchanged	True

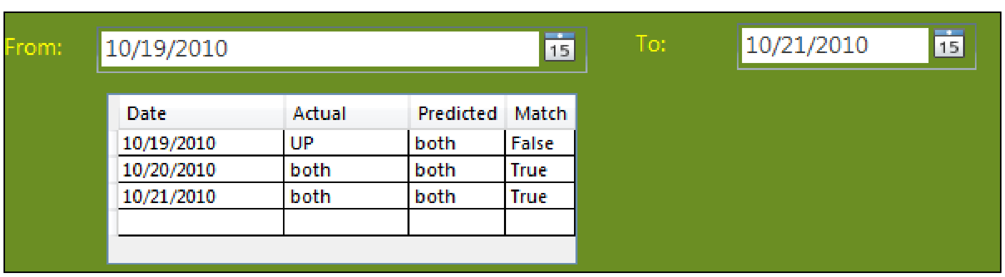
Figure 27: EUR/USD Trends Experiment 3 Results

- **EUR/JPY**



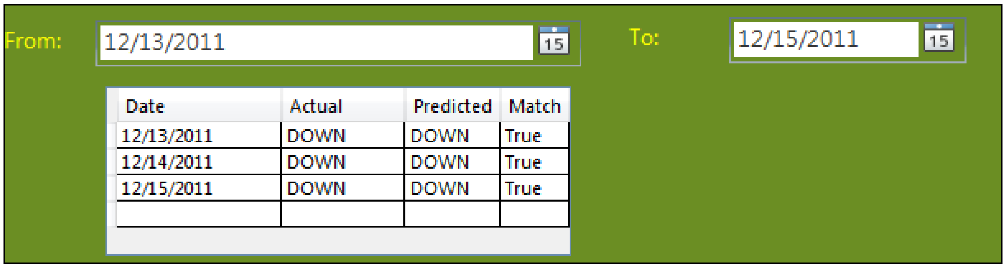
Date	Actual	Predicted	Match
4/14/2009	both	both	True
4/15/2009	both	both	True
4/16/2009	both	both	True

Figure 28: EUR/JPY Trends Experiment 1 Results



Date	Actual	Predicted	Match
10/19/2010	UP	both	False
10/20/2010	both	both	True
10/21/2010	both	both	True

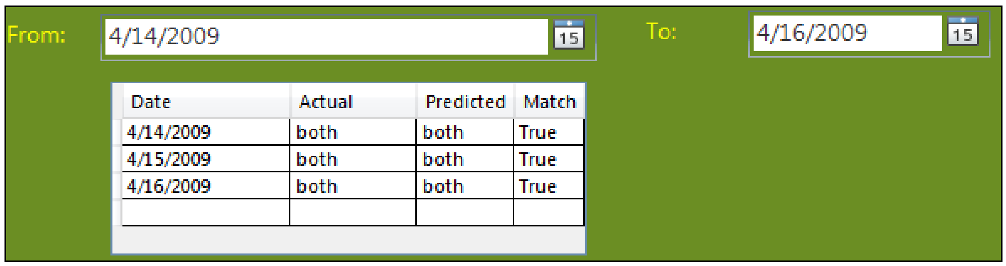
Figure 29: EUR/JPY Trends Experiment 2 Results



Date	Actual	Predicted	Match
12/13/2011	DOWN	DOWN	True
12/14/2011	DOWN	DOWN	True
12/15/2011	DOWN	DOWN	True

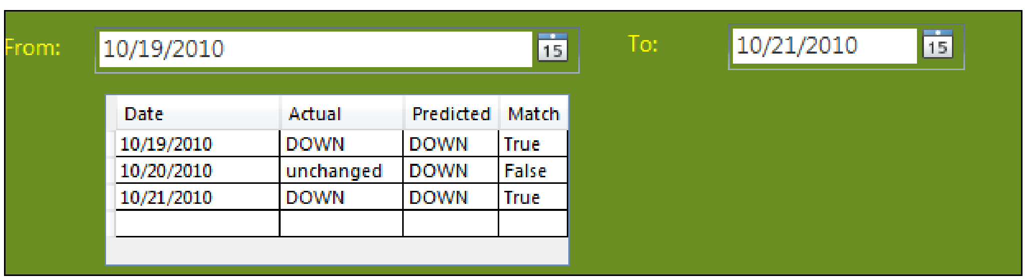
Figure 30: EUR/JPY Trends Experiment 3 Results

- **USD/JPY**



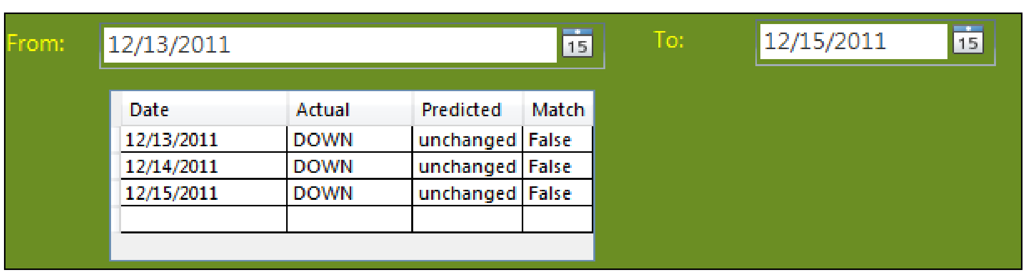
Date	Actual	Predicted	Match
4/14/2009	both	both	True
4/15/2009	both	both	True
4/16/2009	both	both	True

Figure 31: USD/JPY Trends Experiment 1 Results



Date	Actual	Predicted	Match
10/19/2010	DOWN	DOWN	True
10/20/2010	unchanged	DOWN	False
10/21/2010	DOWN	DOWN	True

Figure 32: USD/JPY Trends Experiment 2 Results



Date	Actual	Predicted	Match
12/13/2011	DOWN	unchanged	False
12/14/2011	DOWN	unchanged	False
12/15/2011	DOWN	unchanged	False

Figure 33: USD/JPY Trends Experiment 3 Results

5.2.3 Trends Experiments Results Analysis

The results of the trends experiments yielded 27 results. There were a total 19 predictions that matched to the actual value and 8 that did not match. This result indicates that there is a strong likelihood that the next trend would be predicted with 70% accuracy.

Further testing and experimentation is needed to come up with better results. For example, testing currencies for longer timeframes and experimenting with different parameters such as the number of days and pips to make the trend are possible options to fully understand the value and usefulness of the predictions.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

As almost all the literature suggest, there is no silver bullet to forecast future values and trends in the FX market. Each method has to be constantly refined or new ones discovered. This project implemented an approach to discover future values and trends using data mining techniques.

The prediction of the exchange rates was carried out with a time series algorithm to predict the future numerical value of a currency pair. The results show some promising use.

The prediction of the future trend of a currency pair was implemented by transforming the dataset into a classification problem. The original data was transformed using the SMA and RSI popular technical indicators. A trend was also defined by the user using the days and pips parameters. Finally, the program calculated the class of each row in the dataset and used it a classification tree algorithm.

Learning the tools and technologies to implement DMForex was a time-demanding experience. It took time to learn the tools to manage the application lifecycle. It took time to learn the practices and techniques in Prism 4 to build a modular, MVVM application using WPF. It also took a lot of time to understand and be able to use the building of data mining models in SQL Server 2008 Analysis Services. But at the end, it was all worth it. One of the most satisfying conclusions of this project is that I have learned a whole new way of writing software.

6.2 Future Work

Several improvements can be made to DMForex in the future. One of these is the implementation of some other algorithms available, such as the Support Vector Machines and Neural Networks algorithms. The foundations have been put in place in the program thanks to Prism that allows for this extensibility.

Another area of improvement is in the evaluation of the algorithms. The algorithms implemented already contain statistical information such as probability of the content nodes. They can be incorporated in the program to aid the user make a better trading decision.

Finally, the data mining of the rates and currencies can be improved by adding new views. One of these improvements could be implementing a new view in which the user interacts with the dataset to add, modify or delete input columns. Another view can be added to explore graphically the dataset to spot for data errors or outliers.

References

[1] M. P. Hanias and P. G. Curtis, Time Series Prediction of Dollar\Euro Exchange Rate Index, International Research Journal of Finance and Economics, Issue 15, 2008

[2] Z. Vojinovic, V. Kecman, and R. Seidel, A data mining approach to financial time series modelling and forecasting,Intelligent Systems in Accounting, Finance & Management, Int. J. Intell. Syst. Acc. Fin. Mgmt., John Wiley & Sons, Ltd.P. 225-239, 2001.

[3] T. A. Ferreira, C. Vasconcelos, and P. J. L. Adeodato, A New Evolutionary Approach for Time Series Forecasting, Procedeeings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, 2007.

[4] D. Peramunetilleke and R. K. Wong, Currency exchange rate forecasting from news headlines, presented at the 13th Australasian Database Conf. (ADC 2002), Melbourne, Australia, 2002.

[5] D.J. Hand, G. Blunt, M. G. Kelly and N. M. Adams, Data Mining for Fun and Profit, Statistical Science, Vol. 15, No. 2 (May, 2000), pp. 111-126

[6] B. Kovalerchuk, E. Vityaev, Data Mining for Financial Applications, In: Oded Maimon, Lior Rokach (Eds.): The Data Mining and Knowledge Discovery Handbook. Springer 2005.

[7] Dongsong Zhang, Lina Zhou; Discovering Golden Nuggets: Data Mining in Financial Application; Ieee Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 34, No. 4, November 2004

[8] C. Clifton and B. Thuraisingham, Emerging standards for data mining, Computer Standards & Interfaces 23, P. 187-193, 2001

[9] J.V. Hansen and R.D. Nelson, Forecasting and recombining time-series components by using neural networks, Journal of the Operational Research Society (2003) 54, 307–317, 2003.

[10] A. K. Nag and A. Mitra, Forecasting Daily Foreign Exchange Rates Using Genetically Optimized Neural Networks, Journal of Forecasting, 21, 501-511, 2002.

[11] J. Kamruzzaman and R. Sarker, "Forecasting of Currency Exchange Rates using ANN: A Case Study," Proc. IEEE Intl. Conf. on Neu. Net. & Sign. Process (ICNNSP03), pp. 793-797, China, 2003.

[12] W. Huang, K. K. Lai, J. Zhang, and Y. Bao, Foreign Exchange Rates Forecasting with Multilayer Perceptrons Neural Network by Bayesian Learning, Proceedings of the 2008 Fourth International Conference on Natural Computation - Volume 07, 2008.

[13] S. Chiraphadhanakul, P. Dangprasert, and V. Avatchanakorn, Proceedings of the 1997 IASTED International Conference on Intelligent Information Systems, 1997.

[14] Y. S. Abu-Mostafa and A. F. Atiya, Introduction to Financial Forecasting, Applied Intelligence 6, 205-213, 1996.

[15] M. Suarez-Farinas, C. E. Pedreira, and M. C. Medeiros, Local Global Neural Networks: A New Approach for Nonlinear Time Series Modeling, Journal of the American Statistical Association, 2004.

[16] T. Hill, M. O'Connor, and R. Marcus Neural Network models for time series forecasts, Management Science, 1996.

[17] M. Lam, Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, Decision Support Systems, Volume 37, Issue 4, Pages 567-581, September 2004.

[18] J. V. Hansen, J.B. McDonald, and R. D. Nelson, Some Evidence On Forecasting Time-Series With support Vector Machines, Journal of Operational Research Society, Volume 57, Pages 1053-1063, 2006.

[19] J. Kamruzzaman, R. Sarker and I. Ahmad, "SVM based models for predicting foreign exchange rates," Proc. IEEE Intl. Conf. on Data Mining (ICDM'03), pp. 557–560, Florida, USA, 2003

[20] Bank of International Settlements, Triennial Central Bank Survey: Foreign exchange and derivatives market activity in April 2010, Sept. 2010, <http://www.bis.org/publ/rpfx10.pdf> BIS survey

[21] Jiawei Han, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2005

[22] P. K. Janert, Data Analysis with Open Source, O'Reilly Media, 2010

[23] C. Chatfield, The Analysis of Time Series: An Introduction, 6th Edition, Chapman, & Hall/CRC, 2003.

[24] D. C. Montgomery, C. L. Jennings, and M. Kulahci, Introduction To Time Series Analysis And Forecasting, Wiley, 2008.

[25] P. J. Brockwell, R. A. Davis, Introduction to Time Series and Forecasting, Springer, 2002.

[26] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, Data Mining: A Knowledge Discovery Approach, Springer, 2007.

[27] M. Bramer, Principles of Data Mining, Springer, 2007.

[28] J. MacLennan, Z. Tang and B. Crivat, Data Mining with SQL Server 2008, John Wiley & Sons, Indianapolis, IN, USA, 2009.

[29] Developer's Guide to Microsoft Prism (Last visited April 25, 2012, online: http://msdn.microsoft.com/en-us/library/gg406140.aspx)

[30] TA-Lib: Technical Analysis Library (Last visited April 25, 2012, online: http://www.ta-lib.org/hdr_dw.html)

[31] Chappell, D.: 'What is Application Lifecycle Management?' (Last visited April 25, 2012, online: http://www.davidchappell.com/WhatIsALM--Chappell.pdf)