Stock Price Prediction

Ryan Jude Sukdeo

20 September 2016

Contents

1 Finance Definitions

Financial terms are used extensively in this document. To prevent any confusion the mentioned terms will be defined before proceeding. Definitions for opening price, closing price, high price, low price, volume, ex dividend and adjusted close price were taken from Investopedia[1].

Term	**Definition**
Opening Price	The price at which a security first trades upon the opening of an exchange on a given trading day.
Closing Price	The final price at which a security is traded on a given trading day.
High Price	The highest price at which a stock traded during the course of the day.
Low Price	The lowest price at which a stock trades over the course of a trading day.
Volume	A measure of how much of a given financial asset has been traded in a given period of time.
Adjusted Close Price	The closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open.
Split Ratio	A ratio representing the increase in the number of outstanding stock following a stock split.
Ex-Dividend	Ex-dividend is a classification of trading shares when a declared dividend belongs to the seller rather than the buyer. A stock will be given ex-dividend status if a person has been confirmed by the company to receive the dividend payment.

Adjusted opening price, adjusted high price, adjusted low price and adjusted volume are similar to there non adjusted counterparts just referencing adjusted stock prices instead of ordinary stock prices.

[1] Investopedia website: http://www.investopedia.com/

2 Introduction

Financial markets have been drawing a lot of attention from the machine learning community. In finance there are vast amounts of available data and a lot of technological infrastructure in place making it a natural arena for machine learning to thrive. Even though finance and machine learning seem to be natural compliments successful projection of financial instruments is an extremely difficult exercise.

The purpose of this document is to serve as a report of my findings when building a stock prediction tool. For the sake of brevity from this point onwards the stock prediction tool will be referred to as simply the tool.

3 Overview

3.1 Strategy

Stocks tend to exhibit unique behaviours and nuances making it difficult to apply a single instance of a model successfully across many stocks.

The goal is to build a tool that is flexible and lightweight being able to easily adjust its behaviour to any stocks the user inputs. Instead of building a single complex model which may take a longer time to calibrate and run the tool contains many smaller more basic models. Each model has its advantages and disadvantages but by having many models the dataset is being tested against different assumptions. This is important because the user may request any number of different stocks to be projected. Given the large number of stocks available it is not possible to fully interrogate every data set, hence the tool is relying on a wide coverage of models to reduce the risk of failure.

In order for the tool to be flexible it would need to have access to models that do not share the same weaknesses, if the models have similar weaknesses the tool would be very brittle to circumstances that the models were not designed to handle.

3.2 Models Chosen

The tool contains these four models:

- Linear regression,
- Decision tree regression,
- K nearest neighbours (KNN) model,
- Custom ensemble model.

This document will not go into much detail regarding the theory behind each model. Basically all of the above models work in different ways. Linear regression models are linear parametric models, KNN is a non parametric model that uses the calibration data without fitting any behaviour on the underlying data and decision trees have the ability to model non linear relationships between variables and are also able to handle interactions between variables. The ensemble model is an experimental model which combines the three other models in an attempt to get better results.

3.3 Metrics Chosen

The tool makes decisions based on three different metrics:

- Mean square error (MSE),
- Adjusted r squared,
- Mean absolute percentage difference.

When calibrating models MSE was used as the metric we optimise against. The MSE is a very common metric that penalises very large deviations from the actual values you are trying to predict. The metric is also differentiable which makes optimisation much easier in the case of the linear regression and decision tree regression models. The lower the MSE the closer the predicted and actual values, hence minimizing MSE is the goal during calibration.

The adjusted r square and mean absolute percentage difference are also reported. These metrics are easier for the user to interpret.

A successful model will be defined as having less than a five percent mean absolute percentage difference for a given projection date. This metric will be tested for projection dates going as far forward as three months.

4 Market Data

4.1 Source

When choosing a data provider it was very important that the data provider has an easy to use Python API as well as freely available data. For these reasons I chose to use Quandl as my data provider. Quandl has a freely accessible database that contains US end of day (EOD) stock data, the name of the database is Wiki EOD Stock Prices. The database had all the data required as well as containing adjusted stock price data which is ideal given that the objective is to project the adjusted stock close price.

4.2 Market Data Available

The fields available for each stock are as follows:

- High price,
- Low price,
- Opening price,
- Closing price,
- Volume,
- Adjusted opening price,
- Adjusted closing price,
- Adjusted high price,
- Adjusted low price,
- Adjusted volume,
- Split ratio,
- Ex-dividend.

All the definitions of the mentioned fields can be found in the first chapter of this document. At the time of documenting there were 3181 stock datasets available to the user on the Wiki EOD Stock Prices database.

4.3 Market Data Used

Analysis is performed on the following four stock tickers:

- Apple Inc (AAPL),
- Cincinnati Financial Corp. (CINF),
- LendingTree Inc (New) (TREE),
- Universal American Corp (New) (UAM).

Initially the models were trained on data that is not adjusted for stock splits and dividends. This exercise was performed to see how effective non adjusted data would be when predicting the adjusted closing price. The models were trained on high price, low price, opening price, closing price and volume.

The following results were retrieved when using training data for the period 2015-01-01 to 2016-09-09. Projection date was set to 2016-10-30.

	Linear	**Tree**	**KNN**	**Ensemble**
AAPL	19.83%	37.35%	37.06%	43.35%
CINF	80.32%	80.36%	79.17%	82.14%
TREE	1.33%	54.83%	50.63%	56.47%
UAM	53.02%	55.04%	31.45%	51.31%

Figure 1: R^2 adjusted for models on non adjusted price data

Afterwards the models were trained on data containing information regarding stock splits and dividends such as adjusted opening price, adjusted high price, adjusted low price, adjusted volume, split ratio and ex-dividend.

	Linear	**Tree**	**KNN**	**Ensemble**
AAPL	34.62%	43.09%	38.54%	45.00%
CINF	86.95%	90.22%	86.95%	91.70%
TREE	21.33%	62.31%	55.30%	60.78%
UAM	43.48%	55.14%	57.41%	60.12%

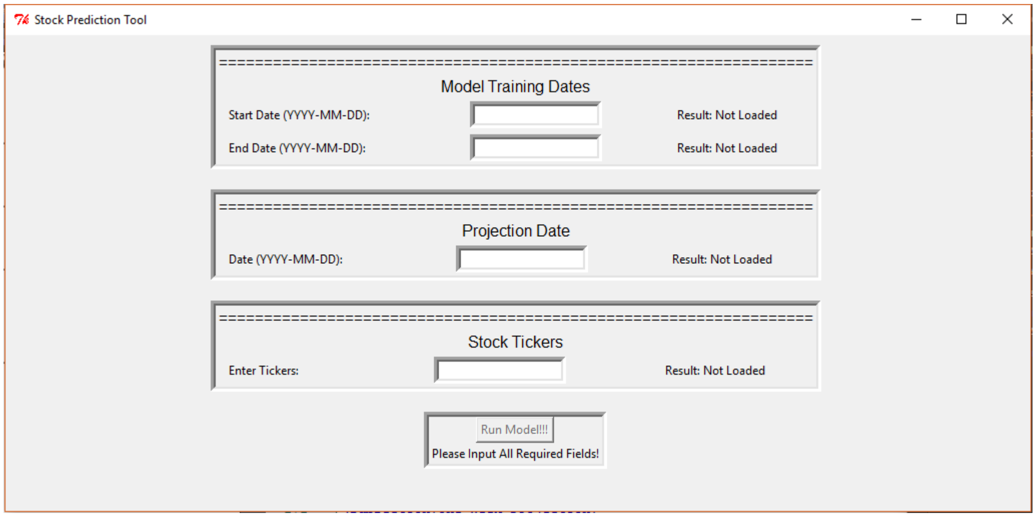
Figure 2: R^2 adjusted for models on adjusted price data

The models that are based on the adjusted stock prices perform better than models based on non adjusted stock prices, this finding strengthens the decision to use the adjusted stock prices data. Hence we train the models on adjusted open, adjusted high, adjusted low, adjusted volume, split ratio and ex dividend.

5 Stock Prediction Tool

5.1 GUI

The user interacts with the stock prediction tool through the GUI.



Model Training Dates

Start Date (YYYY-MM-DD): Result: Not Loaded

End Date (YYYY-MM-DD): Result: Not Loaded

Projection Date

Date (YYYY-MM-DD): Result: Not Loaded

Stock Tickers

Enter Tickers: Result: Not Loaded

Please Input All Required Fields!

Figure 3: Stock Prediction Tool GUI

Under the **Model Training Dates** section there is the start date and end date. Together the dates define the training period of the market data that the models will be trained on.

In the **Projection date** section we have the date. This date represents the date the user wants to project the adjusted closing stock prices to.

Stock Tickers represents the stock tickers of the stocks the user is interested in projecting.

5.2 Data Processing

Before the models can be used for training or projection there are adjustments that must be made. This section will go through the adjustments that the tool makes to the data. There are two instances when data is retrieved from Quandl. Initially market data is retrieved to train and test the models. Afterwards the latest available market data is retrieved which will be used as inputs when projecting the adjusted closing price.

5.2.1 Training and Testing Data

The data responsible for training and testing the models is specified by the start and end date fields in the Model Training Dates section of the GUI. The market data is retrieved on a stock by stock basis, if a stock did not exist during the specified training period it is automatically removed from the model training process. Once the raw data has been retrieved the data is filtered, only the relevant factors will be passed on through the model. The current models are trained on adjusted open, adjusted high, adjusted low, adjusted volume, split ratio and ex dividend.

Next the market data must be normalized before the models can be trained off it. The KNN model uses Euclidean distances hence will be adversely affected if we train the model off non normalized data. The StandardScaler[2] class is used from sklearn to perform the normalization.

Now that the training data is filtered for required factors and normalized, all that is outstanding is to lag the market data.

5.2.2 Projection Data

Note our projection function takes the form:

$$Price_{Adjusted} = F(X_1(Lag), X_2(Lag), ..., X_N(Lag))$$

The tool will attempt to retrieve market data for the current date, if no data exists for the current date then the tool will find the last date that Quandl has market data available for the stock in question. Once the latest data is retrieved the data will be standardized using the same normalization models fitted on the training market data. The resultant data is then used to project the adjusted close price.

[2] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

5.2.3 Train Test Split

Market data will be split into three portions training, testing and validation. It was decided to apportion 70 percent of the data for training, 20 percent for testing and 10 percent for validation.

The training data will be used to calibrate the models. We will not be able to use the training data to test for overfitting, that is were the validation data comes into play. The calibrated models are tested on the validation data set to make sure the models returned are not overfitting, the tool performs this check by returning the models that produce the lowest MSE on the validation data.

Finally the models are tested on the testing data set. This will show the flexibility and predictive power of the model on a data set that the models were not trained on.

5.3 Model Training

Every stock will initially train one linear regression model, 100 decision tree models (due to parameter adjustments) and 10 KNN models. From the 100 decision tree models and 10 KNN models one KNN model and one decision tree model will be chosen, these models will be the models least likely to be affected by overfitting. The linear regression model, decision tree regression model and KNN model will then be used to calibrate the ensemble model.

Once all four models are available the r squared adjusted value is calculated using the validation market data. The model with highest r squared adjusted value would be the model responsible for projecting the adjusted closing stock price.

5.3.1 Linear Regression Model

A linear regression model[3] is included in the tool. No parameter fine tuning was necessary for this model. I chose to include a linear regression model in the stock prediction tool for the following reasons:

- When the relationship between independent and dependent variables are close to being linear this model will produce optimal results,
- Results are easily interpretable.

[3] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

5.3.2 Decision Tree Regression Model

A decision tree regression model[4] is included in the tool. The tool attempts to fine tune two parameters in the decision tree model *max_depth* and *min_sample_split*. According to sklearn documentation *max_depth* represents the maximum depth of the tree and *min_sample_split* represents the minimum number of samples required to split an internal node.

Both *max_depth* and *min_sample_split* are tested for values $\in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. Tree models are calibrated on the training data and is tested on the validation data afterwards.

Models are trained for all combinations of *max_depth* and *min_sample_split* on the training data. Once all models are trained the models are tested on the validation data set. The model with the lowest MSE on the validation data is not likely to be overfitted and is chosen as the best tree model for the stock.

A decision tree regression model was chosen to be included in the stock prediction tool for the following reasons:

- Has the ability to model non linear relationships between variables and are also able to handle interactions between variables,
- Results are easy to interpret,
- Decision trees implicitly perform feature selection.

5.3.3 KNN Model

A KNN model [5] is included in the tool. The tool attempts to fine tune one parameter in the KNN model *n_neighbors*. According to sklearn documentation *n_neighbors* represents the number of neighbors to use.

The parameter *n_neighbors* are tested for values $\in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. KNN models are calibrated on the training data and is tested on the validation data afterwards.

[4] http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[5] http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

Models are trained for each value of *n_neighbors*, this calibration occurs on the training set. Once all models are trained the models are tested on the validation set. The model with the lowest MSE on the validation data is not likely to be overfitted and is chosen as the best KNN model for the stock.

A KNN model was chosen to be included in the tool for the following reasons:

- Does not fit a parametric function to the underlying data instead uses the given data as references (lazy learner) allowing it to pick up on systematic localized features of data without structural breaks or mixtures,
- Works on nonlinear data.

5.3.4 Ensemble Model

The tool contains a custom ensemble model. The ensemble model is derived from the best (calibrated models with least chance of overfitting) linear regression, tree regression model and KNN model available to the stock prediction tool. The ensemble Model takes the form:

$$Y_{Ensemble} = W_1 * Y_{Linear} + W_2 * Y_{Tree} + W_3 * Y_{KNN}$$

$$W_1 + W_2 + W_3 = 1$$

$$W_1, W_2, W_3 \in [0, 0.5]$$

Notice that the weights cannot be larger than 0.5 this forces the ensemble model to use two or more models in its projections. If an individual weight can reach one the ensemble model would assign a weight of one to the best performing in most cases. Note that the ensemble model weights are calibrated using training data.

5.4 Available Results

5.4.1 Metrics Produced

Each stock will have the following metrics captured by the tool:

- R squared adjusted for training set, validation set and test set for all models,
- Parameters *max_depth* and *min_sample_split* that produces the best tree model,

- Parameter *n_neighbours* chosen that produces the best KNN model,
- Ensemble model weights,
- Mean percentage difference on the entire dataset as well as the test data set. This metric will be used to determine the success of the models.

5.4.2 Graphs Produced

For each stock the following graphs are created:

- Linear_Prediction: Shows the actual vs predicted results when applying the linear regression model on the test set(training + testing + validation),
- Linear_Differences: Shows the absolute difference percentage between actual and predicted values when applying the linear regression model on the test set. Also includes the mean absolute difference percentage,
- Tree_Prediction: Shows the actual vs predicted results when applying the decision tree regression model on the test set,
- Tree_Differences: Shows the absolute difference percentage between actual and predicted values when applying the decision tree regression model on the test set. Also includes the mean absolute difference percentage,
- KNN_Prediction: Shows the actual vs predicted results when applying the KNN model on the test set,
- KNN_Differences: Shows the absolute difference percentage between actual and predicted values when applying the KNN model on the test set. Also includes the mean absolute difference percentage,
- Ensemble_Prediction: Shows the actual vs predicted results when applying the ensemble model on the test set,
- Ensemble_Differences: Shows the absolute difference percentage between actual and predicted values when applying the ensemble model on the test set. Also includes the mean absolute difference percentage.

An example of the graphs produced are shown below:

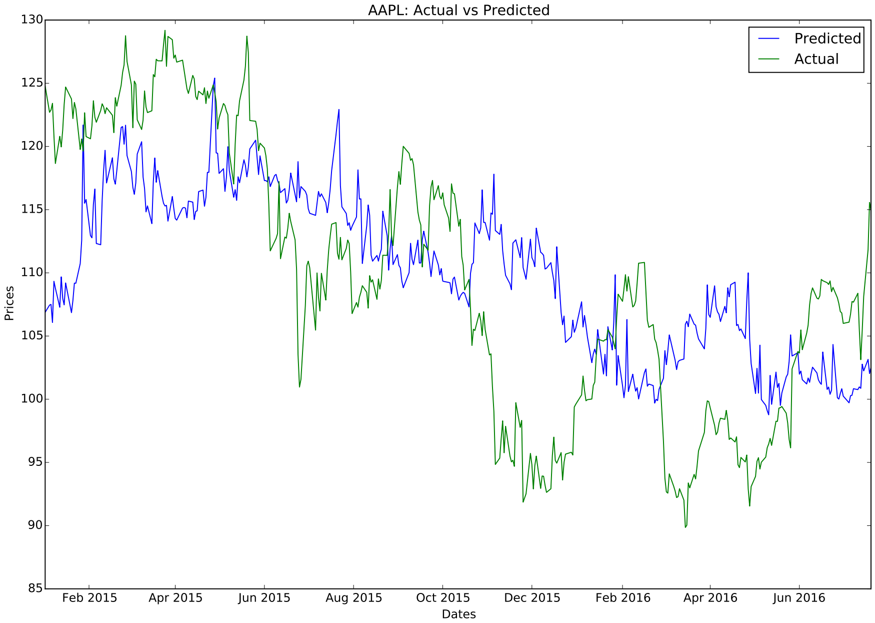


Figure 4: Graph showing actual vs predicted adj closing price

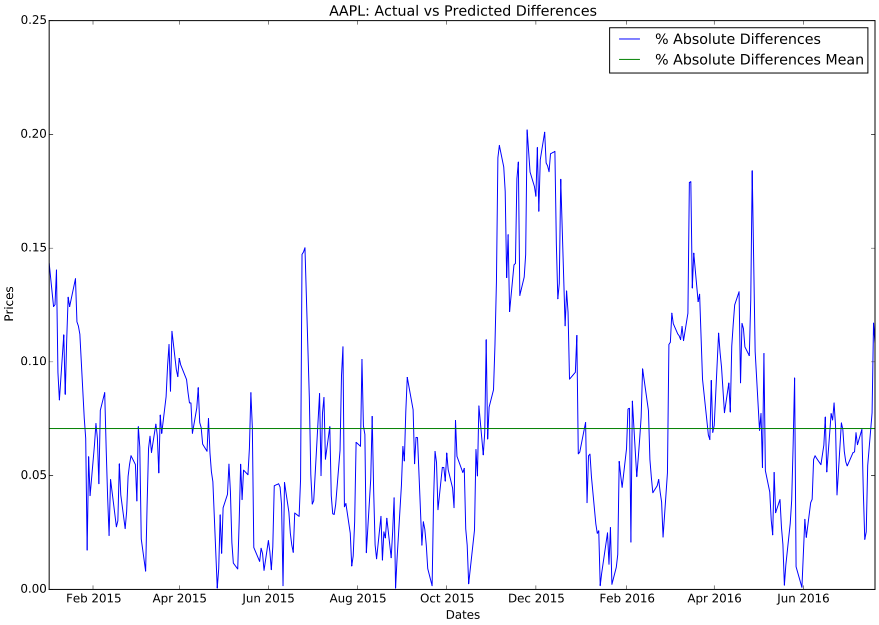


Figure 5: Graph showing absolute percentage difference

5.5 Summary of Overall Process

This section will discuss the steps to project the adjusted closing prices.

1. Retrieve data for each stock in the training period specified by the user,
2. Filter the data for adjusted open, adjusted close, adjusted volume, split ratio, adjusted high, adjusted low and ex dividend,
3. Split data into X and Y arrays,
4. Lag data according to the length between the projection data and the current date,
5. Normalize the X data,
6. Split the data into a test set (70%), validation set (10%) and training set (20%).
7. Train models on training data set,
8. For tree and KNN models test the models against the validation test set, the lowest MSE tree and KNN models will be chosen,
9. Train the ensemble model on the training set, the ensemble model uses the linear model, KNN model and tree model.
10. The model with the highest r square adjusted amongst the linear model, tree model, KNN model and ensemble model will be used in projection of the adjusted stock price,
11. Retrieve latest data, normalize the data then use the model from the last step to do the projection. Note each stock would have a different model for projection.

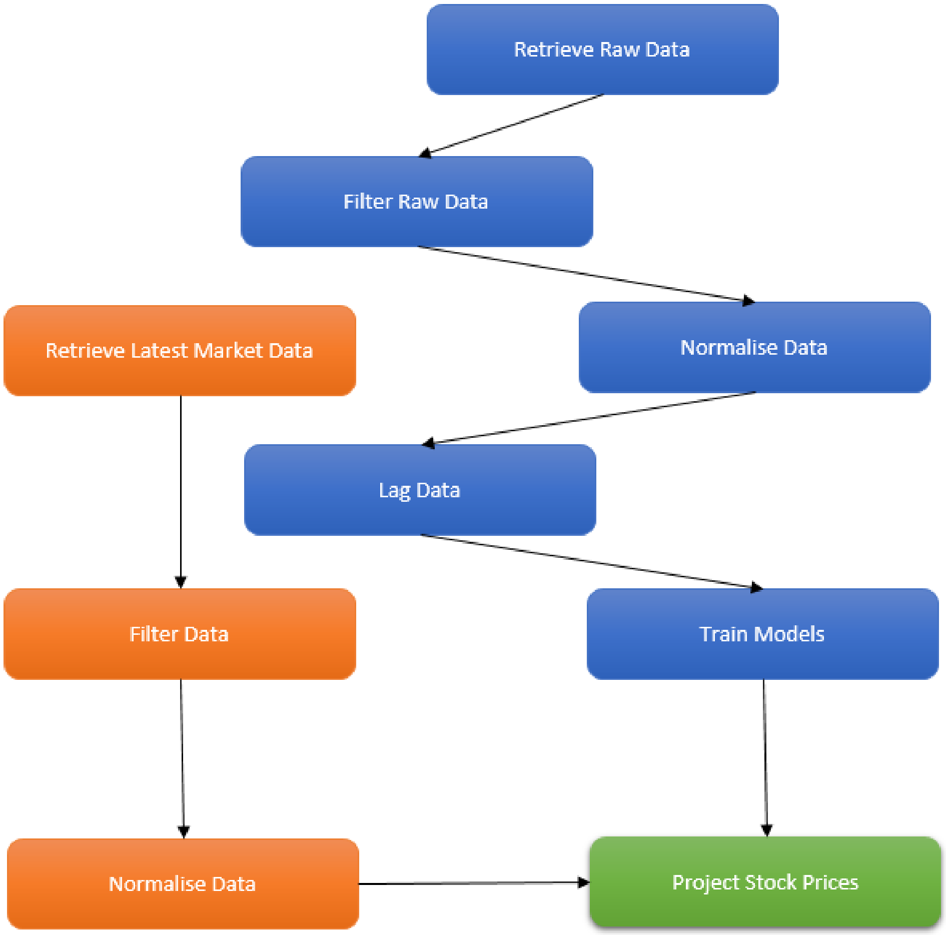


Figure 6: Overview of training and projection process

6 Results

Success will be determined using the mean absolute percentage difference (difference between actual and predicted), if the mean absolute percentage is greater than 5% then the model has failed to project the adjusted closing price. All models included in the results have been trained using data from period 2015-01-01 to 2016-09-16 and the presented results are from the models being tested on the testing sample.

	1D	**7D**	**14D**	**1M**	**2M**	**3M**
AAPL	1.11%	3.00%	4.30%	6.32%	6.41%	6.10%
CINF	1.05%	1.81%	2.48%	3.05%	4.12%	3.77%
TREE	3.38%	6.94%	10.76%	13.57%	9.86%	11.16%
UAM	1.80%	5.0%	5.86%	7.48%	8.32%	7.74%

Figure 7: Mean absolute % difference testing data set

	1D	**7D**	**14D**	**1M**	**2M**	**3M**
AAPL	1.26%	3.36%	3.77%	5.10%	5.85%	5.40%
CINF	0.99%	1.20%	2.33%	2.84%	4.07%	2.92%
TREE	3.1%	7.74%	10.30%	9.67%	11.12%	10.40%
UAM	2.51%	4.82%	5.41%	6.54%	6.16%	6.79%

Figure 8: Mean absolute % difference validation data set

	1D	**7D**	**14D**	**1M**	**2M**	**3M**
AAPL	1.20%	2.40%	3.99%	5.14%	3.70%	4.95%
CINF	0.66%	0.38%	1.8%	2.48%	2.33%	2.51%
TREE	3.09%	6.55%	5.9%	7.16%	11.09%	5.72%
UAM	1.57%	3.76%	5.50%	3.62%	7.78%	5.26%

Figure 9: Mean absolute % difference training data set

7 Conclusion

From the results in the last chapter it is shown that the predictive power of the tool starts to decrease the further into the future we try to project. This is comforting for it is behaviour that is expected, it should become more difficult to project a stock price further into the future. The tool predicts all stocks very well when forecasting one day ahead, after a week it fails to project two stocks (TREE and UAM). Only after trying to project past a month does the tool fail to predict three out of the four stocks. The model can be improved but I am satisfied with the results.

The tool can be improved in the following ways:

- Include feature selection,
- Add more models to the tool.