

FINANCIAL TIME SERIES FORECASTING – A MACHINE LEARNING APPROACH

Alexiei Dingli and Karl Sant Fournier

Department of Artificial Intelligence, University of Malta, Malta

ABSTRACT

The Stock Market is known for its volatile and unstable nature. A particular stock could be thriving in one period and declining in the next. Stock traders make money from buying equity when they are at their lowest and selling when they are at their highest. The logical question would be: "What Causes Stock Prices To Change?". At the most fundamental level, the answer to this would be the demand and supply. In reality, there are many theories as to why stock prices fluctuate, but there is no generic theory that explains all, simply because not all stocks are identical, and one theory that may apply for today, may not necessarily apply for tomorrow. This paper covers various approaches taken to attempt to predict the stock market without extensive prior knowledge or experience in the subject area, highlighting the advantages and limitations of the different techniques such as regression and classification. We formulate both short term and long term predictions. Through experimentation we achieve 81% accuracy for future trend direction using classification, 0.0117 RMSE for next day price and 0.0613 RMSE for next day change in price using regression techniques. The results obtained in this paper are achieved using only historic prices and technical indicators. Various methods, tools and evaluation techniques will be assessed throughout the course of this paper, the result of this contributes as to which techniques will be selected and enhanced in the final artefact of a stock prediction model. Further work will be conducted utilising deep learning techniques to approach the problem. This paper will serve as a preliminary guide to researchers wishing to expose themselves to this area.

KEYWORDS

Stock Prediction, Fin Tech, Machine Learning, Time Series Forecasting, Data Science

1. INTRODUCTION

A stock is a piece of a company, and when an individual buys stock, he/she is essentially becoming a part owner of said company. Throughout this thesis, the term stock may be referred to as 'security', 'asset', 'equity' or 'share', all these terms refer to the same entity. The stock market is a community that brings together traders that want to buy and sell stocks. Among other metrics, the stock price of a company reflects its performance. The law of demand and supply holds that when the demand for a commodity increases or its supply decreases, the price of the commodity increases. On the other hand, if the demand decreases or the supply increases, the commodity price decreases. The same applies for the stock market. Stocks are traded in a stock exchange, this is an organised market through which investors can buy and/or sell securities in a public manner. The ideal scenario is to buy when there is a good chance that the stock price will increase, and to sell when there is a chance that the stock price is about to decrease, hence yielding profit from the difference between the price the stock was bought and the price the same stock was sold. In a nutshell, the role of a stockbroker is to know when to buy and when to sell stocks with the goal of earning a profit.

2. BACKGROUND

2.1. LITERATURE REVIEW

Given the nature of this topic, many researchers and traders have exploited the area of statistical analysis and machine learning in order to generate a profit from their investments. Although the initial bias would be that any successful enterprise would not share its methods in order to keep profits to themselves, there have been a good number of researchers who have published their findings.

2.1. 1 Statistical Methods

Regression analysis is a statistical tool, used to investigate relationships between variables [1].

[2] uses regression to study the relationship between news and stock price changes, in an effort to improve the performance of the conventional stock price forecasting process. The author regresses the weekly stock price changes on the news values from the previous week. Results show that there is a very significant correlation between the changes in stock price and the news values at the start of the week, such that one unit increase in news values, results in approximately 6.33 unit increase in stock price change. The author does note however that the R-squared value is minimal (0.09243). This indicates the proportion of the variability of the weekly price changes predicted by the model. This small value means that the model does not predict the price changes accurately.

Traders and researchers have exploited statistical techniques such as autoregressive integrated moving average (ARIMA) to forecast financial time series. ARIMA models work on stationarized time series. A stationary time series is one whose statistical properties are constant over time [3]. Many statistical forecasting methods are based on the assumption that the time series can be rendered stationary through the use of transformations. Once a prediction is made on the stationary time series, it can be converted back to the original series using the same transformations that made it stationary.

2.2 Machine Learning Methods

Machine learning is the science of getting computers to take decisions without being explicitly programmed to do so. This phenomenon has given positive results in experiments such as speech recognition, self-driving cars, image recognition and a number of other areas. One of the most exciting things about machine learning is that it can be applied to almost anything. In a chaotic life such as that of the 20th century, certain decisions are best delegated to machines. These decisions can range from trivial decisions to critical decisions, according to the nature of the problem at hand. Machine learning techniques can be divided into supervised, unsupervised, association and reinforcement learning. With supervised machine learning, the data being passed to the algorithm is labelled, indicating that a certain data pertains to some class. Some of these techniques include Decision Trees, Regression, Support Vector machines and Naive Bayes. On the other hand, with unsupervised machine learning, the data is not labelled, and the task of the algorithm is to draw inferences from the data without it being labelled from beforehand. Some of these unsupervised techniques include K-Means and Gaussian Mixture models. In the field of finance, various machine learning techniques have been used for predicting stock market prices. Technical indicators and market sentiment are passed as features the various models, which output either a binary or numerical output indicating a direction or change in price respectively.

In the field of machine learning, Support Vector Machines (SVMs) [4] are becoming increasingly popular, a similar enthusiasm was shown when artificial neural networks were discovered. SVMs were first developed by Vladimir N. Vapnik and Alexey Ya Chervonenkis in 1963, these SVMs

were able to classify linear problems in a much more efficient way than ANNs. Later in 1992, along with another two scientists, Vapnik suggested a method called the kernel trick, using the kernel trick, SVMs were able to create non-linear classifiers.

[5] uses SVM to predict direction of future price changes. In the study it showed that the resulting accuracy was highly dependent on the various parameters passed to the SVM model, best results were achieved when using the Gaussian Radial Basis Function kernel as a non-linear classifier. The value of parameter C also had an effect on the results obtained. Features used were selected technical indicators and oscillators. The author compared the results of the SVM to other methods such as Neural Networks using Back-propagation and discovered that better results were achieved using the SVM approach.

Decision trees are a type of classifier which can be represented in a form of a tree. A tree can be divided into a root node, internal nodes and leaf nodes. The feature splitting under the root and internal nodes are calculated using entropy information gain. Decision trees are based on the heuristic of 'Occam's Razor', which favours a short, generalised tree over a complex tree which may overfit the training data.

[6] designs a text-based decision tree for stock market forecasting. Price data is transformed into textual form according to a number of rules, such that each indicator is given a value of either up trend, down trend or no trend. The output is also based on a pre defined rule which outputs whether to buy, hold or sell the stock. Through this text based decision tree approach, the authors achieve an accuracy of 90.82% which they proved to be more effective than the normal, non text-based decision tree. The authors also noted that the model for their text based approach resulted in a simpler, shorter tree than that of the normal decision tree.

[7] use a multi-agent approach for forecasting, namely they use technical and fundamental analysis, neural networks and liquidity methods to forecast the market behaviour. They compare the performance of standard feed-forward networks, Elmand and Jordan RNNs and an architecture known as NEAT (Neuroevolution of augmenting topologies \cite{Stanley}). These architectures are evaluated on their ability to predict price changes on the Bucharest Stock Exchange (BSE). The best results are achieved using the NEAT architecture, which is a genetic algorithm developed by Ken Stanley. This methodology attempts to find a balance between the diversity of evolved solutions and their fitness.

Whilst decision trees have achieved promising results in the field of stock market prediction, some have ventured to the use of ensemble learning algorithms such as random forest. \footnote{\a href="https://arxiv.org/pdf/1605.00003.pdf"} make use of the random forest ensemble to counter act against the overfitting problem of decision trees. The author states that this occurs because decision trees are very sensitive to noise in the data given that they have a very low bias and high variance. Through the Random forest algorithm, the author trains multiple decision trees on a different subspace of the feature space, causing a slight increase in bias. By taking this approach, none of the trees make use of the entire training data, but rather it is shared amongst the various trees in the ensemble. Through this method, the authors achieve an accuracy between the range of 85% and 95% for long term predictions, which is an improvement on what was achieved on the decision tree based approach of [8].

[9] make use of ANNs to forecast stock price direction for the Nikkei 225 index, using a number of popular technical indicators as features. As part of the experiment, the authors opt to divide their technical indicator features into two types. This is done with no mention of the criteria used to divide these features, constantly referring to these as 'type 1' and 'type 2'. which would have been helpful for the reader to better understand the intention for this split. Nonetheless, the authors report an accuracy of 60.87% on the 'type 1' dataset and 81.27% on the

'type 2' dataset for next day price direction forecast, which is an outstanding result given the volatile and unstable nature of day to day prices in the stock market.

3. METHODOLOGY

In this chapter, we discuss the format of the different outputs we would like to achieve, from our experiments. We then identify the various data sources used and highlight any transformations done to these datasets. Given we will be dealing with a large number of features, some of which may be deemed irrelevant, we will be performing feature selection techniques in order to identify the most important features. Once our features are established, we then discuss the various approaches taken to achieve our desired outputs, through machine learning techniques.

3.1. Structure of desired outputs

The below structure is that of which we plan to formulate our outputs, which is mainly threefold:

- Next Period Direction

This output concerns the direction of the price movement with respect to the next period. Natively, this can be considered as a classification problem, whereby we classify each record as 'Up' or 'Down'.

- Next Period Price Change

This output concerns the numerical value of change, for the price movement with respect to the next period. This can be considered to be a regression problem given that we are not dealing with predefined output classes, but with a numerical, continuous value.

- Next Period Actual Price

This output concerns the numerical value for the actual next period price. This can also be considered to be a regression problem given the nature of the output.

The above three outputs will be computed for a number of periods: Daily, Weekly, Monthly, Quarterly and Yearly. By means of this comprehensive approach we will be in a position to identify an optimal combination for forecasting.

3.2. Feature Identification

3.2.1. Historic Stock Prices and Technical Indicators

Historic stock prices are a good place to start, given that the data is publicly available through a number of sources. One of the most widely used mediums is yahoo finance, which provides unlimited historic daily prices in csv format on demand. A python API known as 'yahoo-finance' is used to retrieve these historic prices from 2003 till 2016, which are then stored in our relational database for future use. These historic prices are then used to formulate what are known as technical indicators. These indicators consist of various financial equations using historic prices which are used to paint a clearer picture as to the condition of the company for that point in time. Investopedia describe these indicators as any class of metrics whose value is derived from generic price activity. Technical indicators evaluate price levels, direction and momentum amongst others. Rather than developing custom functions for each indicator, a library called 'TA-Lib' was identified. This library openly provides 200 technical indicators, and is designed in the form of an open source API. Given that

python is the language of choice for experimentation, a Python wrapper for this API is used. The technical indicators used in our approach include a combination

of momentum, volume, volatility and cycle based indicators. As their name suggests, each class of indicator focuses on a different aspect of market activity.

3.2.2. Currency Exchange Rates

When investors purchase foreign equity, they are actually placing a bet on two elements: one on the stock itself and the other on the currency the stock trades in. In order to evaluate this hypothesis, we collect a number of the most popular exchange rates to incorporate within our predictive model. Currencies were collected from yahoo finance and Quandl data repository, and these include: EUR/USD, GBP/USD and BITCOIN/USD. As one may note, USD was taken as the benchmark since it is one of the most prominent currencies in the global stock exchange.

3.2.3. World Indices

Investopedia describes an index as a measure of change in a securities market. Indices consist of a hypothetical portfolio containing a group of securities that generally represent the performance of the overall market. Typical examples of such indices include the Standard & Poor's 500 (S&P500), Dow Jones Industrial Average (DJIA) and the Nasdaq Composite Index (NASDAQ-100). These indices are retrieved from the yahoo finance API.

3.2.4. Commodity Prices

A number of popular commodities were selected as features for our experiments. Prices for Gold and Oil commodities are retrieved from the Quandl data repository, store them in our relational database and incorporate them into our dataset as time series data.

3.3. Feature Scaling

The features identified earlier are of numeric value, however the ranges for each feature vary widely. A number of machine learning classifiers calculate the distance between points by using Euclidean distance. When adopting this methodology for a broad dataset, one of the features may have such a large range, that it automatically takes precedence over the other features, when in fact other features with smaller ranges may possibly be of more importance to the predictor. Apart from the affects on Euclidean distance, gradient descent algorithms are known to converge faster with scaled features due to the nature of the method's architecture [10]. For the purpose of this study, we will be using Z-Score scaling on all our numeric features.

3.4. Feature Selection

Now that various data sources and features have been identified, the next step is to integrate them with our dataset and evaluate which are most important for the problem at hand. In total, we have over 70 features comprising of technical indicators (Overlap, Volume, Momentum, Volatility and cycle), currency exchanges, commodity prices and world indices. Rather than passing all features to our predictive model we use feature selection techniques in order to statistically identify the most relevant features for the data at hand.

In order to gauge the effectivity of the forecast over different periods, a number of different time spans are taken into consideration, namely daily, weekly, monthly, quarterly and yearly averages. When taking this approach, all features in the dataset are averaged according to periodicity and fed into our feature selection mechanism. Given that the initial dataset comprises of one record per trading day, resulting in 252 records per year (one per official trading day as quoted by

NASDAQ), this periodic average transformation has an effect on the final size of the final dataset. For instance, if we were to take weekly averages, the number of records per year would be taken down to 52, for monthly averages this would be 12 and for quarterly we would have only four records. In machine learning, ideally a substantial amount of examples is fed to the classifier in order to be able to classify the outcome of unseen examples. To tackle this, rather than considering a single stock, numerous stocks are concatenated together in order to have a substantial dataset. The criteria for concatenated stocks is the nature of the company behind each equity. Technology companies are considered as one dataset and Finance companies form their own separate dataset. All values are scaled as discussed in the previous section, and the direction of the next day price is calculated accordingly, with each record being independent from the other.

Sklearn library known as ‘feature selection’ is used for identifying the most important and relevant features. We make use of a feature called ‘SelectKBest’ which makes use of one of two functions to evaluate each feature and assign a score accordingly. One function ‘f_classif’ is used when the target output is a predefined class. This function makes use of the ANOVA F-test and outputs a score known as the F-Value. ANOVA test outputs the proportion of variance explained by the features, the higher the ratio, the more proportion. The second function, known as ‘f_regression’ is used for regression type problems when the output is of continuous value. This function computes the cross correlation between each feature and the target output to assign scores accordingly.

Feature selection techniques are performed for each combination of Output type and periodicity. Stock types are divided into Technology and Finance; Output types that will be considered in this study are next period direction, change in price and actual price; periods considered are Daily, Weekly, Monthly, Quarterly and Yearly as mentioned earlier. Through this approach we are making no undue assumptions regarding the importance of features for our output. For instance, one feature that may be important to forecast the next day direction, may be not as important to forecast the next day price itself. A typical example of this scenario could be the importance of the previous day closing price, whereby when predicting the next day price, this feature is vital since the next day price is built as an addition or subtraction to it, it provides no indication whatsoever on the possible direction of the price movement for the following day. For this reason, all features are taken into consideration and will be evaluated accordingly. This approach considers that importance of features may vary according to whether the stock is for instance a technology stock or a financial stock. We are also making the assumption that features that are important for predicting the next day direction may differ from those that point towards the direction of the price for the following quarter. Table 1 illustrates this comprehensive approach which is performed for all features.

Table 1. Comprehensive approach taken for feature selection

Next Period:	Direction	Change in Price	Actual Price
Features	Day	Day	Day
	Week	Week	Week
	Month	Month	Month
	Quarter	Quarter	Quarter
	Year	Year	Year

3.5. Dataset balancing, training and test split

In order to have unbiased models, it is imperative to have a well structured dataset balancing mechanism, and an adequate ratio between training and test datasets. A number of methods exist when dividing data into the various sets required. In other machine learning problems, it may be ideal to balance, shuffle the dataset and randomly split the training and test sets into 70% and 30% respectively. Such an approach ensures that bias is not present in the data. However, the nature of the data at hand is that of a time series, if this approach were to be taken, it is possible to have so called 'future instances' in our training set and 'past instances' in the test set. Although given that we are treating each record separate from the other, the latter method could still be deemed as valid. Nonetheless, we will be taking an approach which is more suited for time series data.

We will be utilising data from 2003 till 2013(11 years) as our training set and 2014 till 2016 (3 years). This ensures a substantial amount of data for training, whereby 11 years of stock movements should cover a wide range of long and short term trends. The remaining three years will be used for testing, this approach will ensure that we are forecasting and calculating our evaluation metrics on unseen, out-of-sample data.

When talking about a balanced dataset, this mainly concerns our classification task whereby our training examples can be either 'Up' or 'Down'. In order to assist the machine learning model, it is in our best interest to provide training samples which have an equal amount of examples for each class, otherwise we run the risk that during training, the classifier becomes biased towards the class with more examples. This is done by randomly shuffling the training set, identifying which subset has the least values n , retaining the first n from the largest subset, and lastly re ordering by date.

When it comes to the test set, we will not be performing any balancing so as to leave the data in its natural distribution. Like this we will be evaluating an unbiased classifier on unseen data with its native distribution.

3.6. Classification Experiments

For classification, our objective is to predict a predefined output label per example based on the variables available to us. We shall approach this task by industry, firstly focusing on the tech stock industry, followed by financial company stocks whereby we compare and contrast between the two.

Our approach consists of forecasting the price movement direction for the next period, which may be daily, weekly, monthly, quarterly or yearly. The output variable is constructed in such a way that it indicates whether the next period average price goes 'up' or 'down'. Through this, we enable the classifier to identify trends from the features identified, at the same time assessing whether trend detection is more suitable for the short term or the long term. As discussed during the literature review, some analysts state that day to day fluctuations are random, through this experiment we would like to test this assumption and compare it to detecting trends in a longer time span.

By taking the approach of evaluating multiple classifiers utilising the same dataset, we minimise the risk of focusing on a single classifier which may not be the best fit for the data at hand. Now that we've prepared our datasets, a number of algorithms can be used for classification, we will be utilising the below:

- K-Nearest Neighbors
- Logistic Regression
- Naive Bayes
- Support Vector Classifier (SVC)
- Decision Trees
- Random Forest
- Multi Layer Perceptron (MLP)
- Ada Boost
- QDA

From the output of each algorithm we can compute metrics such as accuracy, precision and recall. These metrics could give us an idea on the effectivity of each, striving for a high number of true positives and negatives, as against false positives and negatives. Through these results we will be in a position to compare and contrast each algorithm across various dimensions such as number of features, different periodicities and the aforementioned stock categories.

3.7. Regression Experiments

As discussed earlier, we are dividing our predictions into three forms, one of which is a classification problem, and the other two being a regression problem with a continuous output. In this section we will be discussing the approach taking to tackle the two regression tasks. We will be evaluating results by means of the Root Mean Squared Error (RMSE) and below is a list of regressor algorithms that are used for both tasks:

- Linear Regressor
- Support Vector Regressor
- Decision Tree Regressor
- Ada Boost Regressor
- Random Forest Regressor
- K-Nearest Neighbors Regressor
- Bagging Regressor

The first regression task is to forecast the next period price change. As was done in our classification task, we will be sampling the dataset into different periodicities for daily, weekly, monthly, quarterly and yearly. Given these periodicities, we would like to forecast the actual change from one period to another. The end scope here is similar to that of our classification task where we classified the direction of the next period, however here we are forecasting as to what extent the price will move up or down. The data that will be fed into the various algorithms is the output of the feature selection exercise that was done for this form of data, theoretically the features that were relevant for the direction may not be relevant for the change in price.

As mentioned earlier, this task is divided into two sub-tasks. We now attempt to predict the actual next period price itself, which is also a continuous value. At face value, we assume that the features which were important for the change, and direction should be relevant for the actual price.

However, when simply looking at the data, there is a very good chance that those features which were important for our previous two tests will not be as important for the actual price itself.

This is assumed because a volume or momentum indicator which is a calculated metric does not provide the algorithm any knowledge on the price itself. On the other hand, in our full dataset we have features such as moving averages and last period prices which do provide the predictor a base price to start on, whilst other indicators, should in theory provide an indication as to any possible future fluctuations.

4. RESULTS AND EVALUATION

4.1. Feature Selection Results

In this section, we will be discussing the outcome obtained from our feature selection methodology. Discussion and evaluation of feature selection results will be done per stock type. Feature selection scores are tabulated as per Table 2 for each combination of forecast type and periodicity. One can note, that the number of examples indicated in the second row of the table, decreases with each periodicity sampling as discussed earlier. Given that the table is considerably large due to the number of features, in order to easily understand the contents, all feature scores which are above the global average for each forecast type are highlighted in green. These cells are to be considered as highest statistical importance, with reference to the output class. Towards the right of each section, a sum is calculated for each feature, indicating the overall relevance to the industry as a whole rather than to a specific period. The cells highlighted in yellow signify a total of above average for each forecast type.

As discussed earlier, all features are taken into consideration for the tech industry, as we are relying on our feature selection methodology to outline the most important features. Table 2 shows the feature selection results for the Tech industry. As one may note, the number of shaded cells varies from one periodicity to another, signifying that relevance of certain features differs when evaluating the same dataset with different periodicity.

4.2. Classification Results

We attempt to find the optimal combination between the classifier and number of features that are fed to it. We design an experiment whereby we attempt each classifier for 70 times on identical datasets, each iteration increasing the number of features till we reach 70.

Table 4 shows the top 15 results obtained from this test for the monthly dataset, with the best combination being the MLP Classifier with 28 features. It is important to note that the top 15 results, all obtain an accuracy of 69% and over, with a marginal difference. As one may note, these results only feature three classifiers, other classifiers were evaluated in the same test, but failed to reach the top 15 list, hence are omitted from this table.

Table 4. Table showing the top 15 results for optimal classifier and number of features for the Monthly Tech Industry Dataset

Method	NoFeatures	Accuracy
MLPClassifier	28	0.696640
LogisticRegression	21	0.695652
MLPClassifier	20	0.695652
SVC	18	0.693676
SVC	26	0.692688
LogisticRegression	28	0.692688
LogisticRegression	20	0.691700
SVC	16	0.691700
LogisticRegression	24	0.691700
LogisticRegression	22	0.691700
LogisticRegression	23	0.691700
LogisticRegression	30	0.690711
SVC	12	0.690711
SVC	17	0.690711
LogisticRegression	42	0.690711

In order to better visualise this experiment, Figure 1 shows a graph derived from the full results for these three classifiers, illustrating the of Number of Features plotted against the accuracy rate. From this visualisation, we can see that the accuracy peaks when utilising between 20 and 30 features, and starts to overfit beyond 30 features. One can see that accuracy deterioration occurs in different rates for each classifier. In particular, Logistic Regression and SVC maintain a relatively constant accuracy when compared to the Multi Layer Perceptron line. This substantiates the fact that each classifier has its own merits when preventing overfitting of the training data or even dealing with noisy data.

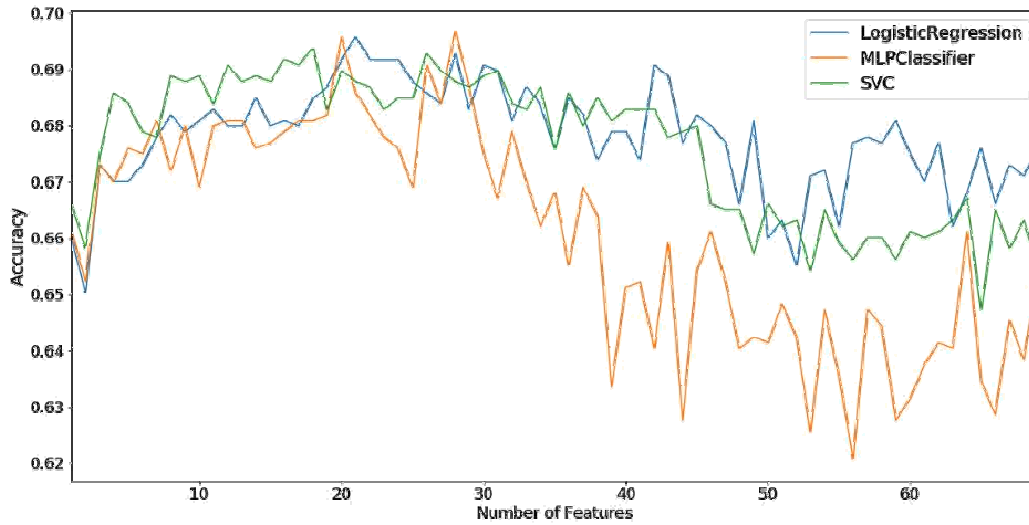


Figure 1. Graph showing accuracy changing according to the number of features passed for each classifier for the Tech industry

Table 5 illustrates the final, optimal combination of classifier and features for each Industry and periodicity, achieving the best results on the yearly datasets for both finance and tech industries.

Table 5. Optimal combination of classifier and features for the Next Period Direction Forecast

Next Period Direction				
Industry	Periodicity	Method	Features	Accuracy
Finance	Daily	<i>AdaBoostClassifier</i>	8	0.52
	Weekly	<i>LogisticRegression</i>	59	0.65
	Monthly	<i>QuadraticDiscriminantAnalysis</i>	2	0.71
	Quarterly	<i>GaussianNB</i>	21	0.64
	Yearly	<i>LogisticRegression</i>	23	0.81
Tech	Daily	<i>LogisticRegression</i>	10	0.52
	Weekly	<i>LogisticRegression</i>	34	0.67
	Monthly	<i>MLPClassifier</i>	28	0.69
	Quarterly	<i>AdaBoostClassifier</i>	3	0.69
	Yearly	<i>KNeighborsClassifier</i>	5	0.77

4.3. Regression Results

We now attempt to achieve the best possible results for each combination by iterating over each regressor, training and testing each one for 70 times, increasing the features that are passed to the model with each iteration. This is similar to what was done in the classification test, with the difference that we are evaluating regressors, and their success is calculated by means of error (RMSE), which is at its optimum when it is lowest.

Table 6 tabulates the optimal combination of algorithm and number of features. One can note that in contrast to our classification results where we had 7 different algorithms that feature in the optimal results, here we have the dominance of Linear Regression and Support Vector Regression, and one instance where KNN proved best.

Table 6. Table showing the optimal regressors for the Next period Actual Price Forecast

Next Period Actual Price				
Industry	Periodicity	Method	Features	RMSE
Finance	Daily	<i>LinearRegression</i>	1	0.011706
	Weekly	<i>LinearRegression</i>	68	0.02047
	Monthly	<i>LinearRegression</i>	8	0.043636
	Quarterly	<i>KNeighborsRegressor</i>	15	0.100451
	Yearly	<i>LinearRegression</i>	26	0.161316
Tech	Daily	<i>LinearRegression</i>	3	0.012995
	Weekly	<i>LinearRegression</i>	67	0.020943
	Monthly	<i>LinearRegression</i>	59	0.04753
	Quarterly	<i>SVR</i>	49	0.086884
	Yearly	<i>SVR</i>	61	0.165858

With regards to the number of features, we can note that the daily forecast required least features to reach optimal accuracy. Moving down the table, we can see that the optimal number of features unlike the RMSE, is not in any way linear to the periodicity.

Table 7 tabulates the optimal combination of algorithm and number of features for our second regression problem, that of forecasting the change in price rather than the price itself. Whilst the error rates are noticeably higher than those for predicting the actual price itself, they follow a similar pattern when it comes to the number of features. It also follows a similar pattern when comparing the RMSE to the periodicity, where a linear relationship seems to exist. Errors from the Finance and Tech industry are relatively similar with the exception of the Monthly and Yearly datasets, where Finance industry stocks achieve considerably better results on the quarterly and yearly datasets.

Table 7. Table showing the optimal regressors for the Next period Change in Price Forecast

Next Period Change in Price				
Industry	Periodicity	Method	Features	RMSE
Finance	Daily	<i>LinearRegression</i>	1	0.061304
	Weekly	<i>LinearRegression</i>	38	0.097223
	Monthly	<i>LinearRegression</i>	20	0.132902
	Quarterly	<i>SVR</i>	29	0.188626
	Yearly	<i>LinearRegression</i>	54	0.157614
Tech	Daily	<i>SVR</i>	1	0.063599
	Weekly	<i>LinearRegression</i>	69	0.104637
	Monthly	<i>LinearRegression</i>	33	0.152169
	Quarterly	<i>SVR</i>	27	0.191362
	Yearly	<i>SVR</i>	66	0.217411

In order to better visualise the forecasts and errors done by the regressor algorithms, Figure 2 shows a plot of the normalised close price and the next day predicted close price against the test period (2014-2016). Figure 3 illustrates the same visualisation but for the weekly tech dataset, here we can see that the plot lines are not as much in line with each other as the daily dataset, this represents the increase in error between one period and another which was discussed earlier. Figure 4 illustrates the same representation, however this time we are dealing with the Monthly dataset, the increase in error here is more prevalent than those discussed up till now.

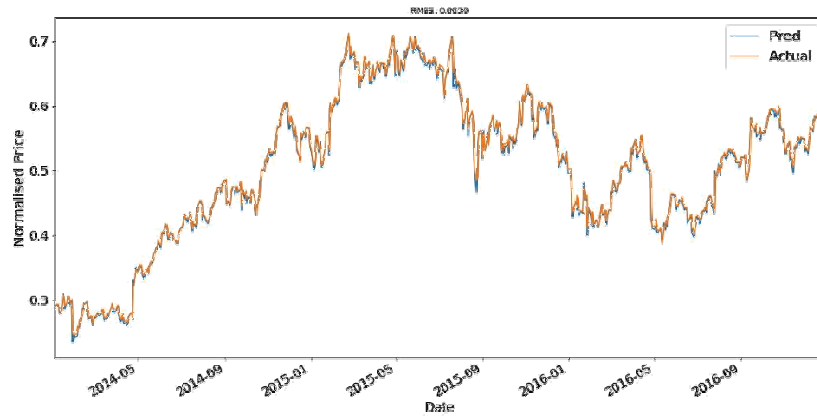


Figure 2. Next Day Actual Price forecast for AAPL Stock using the optimal regressor

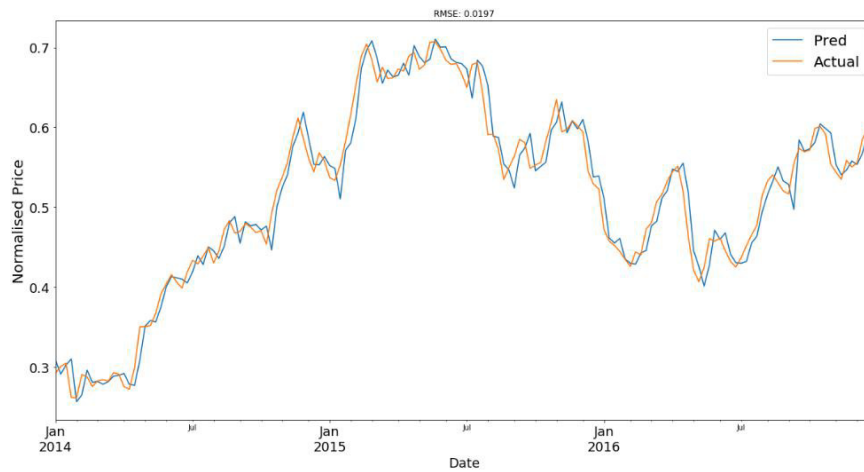


Figure 3. Next Week Actual Price forecast for AAPL Stock using the optimal regressor

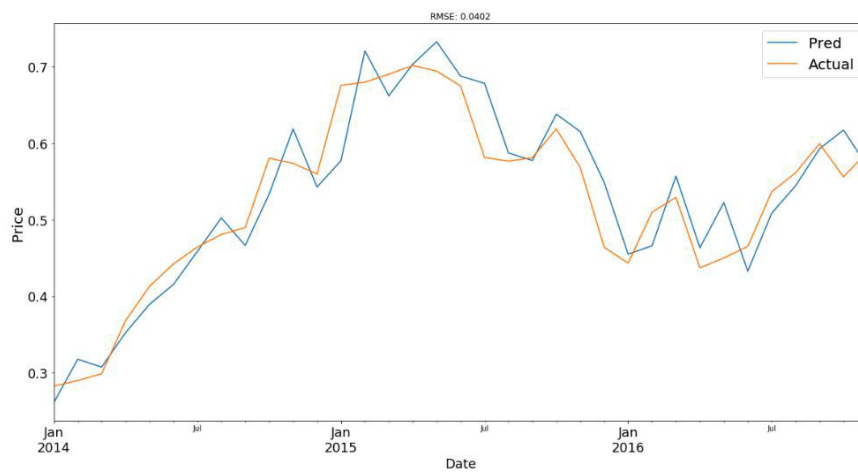


Figure 4. Next Month Actual Price forecast for AAPL Stock using the optimal regressor

5. CONCLUSION AND FUTURE WORKS

This paper gives a basic introduction to the stock market and various approaches taken to predict price movements. Data sources are identified and features are formulated from this data. Through experimentation we achieve 81% accuracy for future trend direction using classification, 0.0117 RMSE for next day price and 0.0613 RMSE for next day change in price using regression techniques. Although many consider movements in the stock market to be somewhat random, our results prove that there are elements that can be learnt, after all the job of a stockbroker or analyst is far from random. By means of identifying relevant features for each stock, a trend can be detected. One of Spencer's Laws of Data holds that anyone can make a decision given enough facts. Whilst extra-ordinary events such as natural disasters are difficult to predict, certain external data such as common market opinions can be gauged through various sentiment analysis techniques. Building on the works discussed in this paper, our approach will be enhanced using deep learning techniques such as Recurrent Neural Networks and Convolutional Neural Networks.

REFERENCES

- [1] Sykes A. O., "An Introduction to Regression Analysis", The Inaugural Coase Lecture, 1993
- [2] Yue Xu S., "Stock Price Forecasting Using Information from Yahoo Finance and Google Trend", UC Berkeley, 2012
- [3] Duke, "Stationarity and differencing", [Online], Available: <http://people.duke.edu/~rnau/411diff.htm> [Accessed January 2017]
- [4] Vapnik V. N., "An Overview of Statistical Learning Theory" *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 1999.
- [5] Kim K. "Financial time series forecasting using support vector machines", Department of Information Systems, Dongguk University 2003
- [6] Panigrahi S. S. and Mantri J. K., "A text based Decision Tree model for stock market forecasting," Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, Noida, 2015
- [7] G. Iuhasz, M. Tirea and V. Negru, "Neural Network Predictions of Stock Price Fluctuations," Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on, Timisoara, 2012
- [8] Siripurapu A., "Convolutional Networks for Stock Trading", Stanford University, Department of Computer Science, 2014
- [9] Qiu M. and Song Y. "Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network", Department of Systems Management, Fukuoka Institute of Technology, Fukuoka, Japan, 2016
- [10] S. and Szegedy C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", 2015

Authors

Prof. Alexiei Dingli is the Head of the Department of Artificial Intelligence within the Faculty of ICT at the University of Malta. He is also one of the founder members of the ACM student chapter in Malta, founder member of the Web Science Research, founder member of the International Game Developers Association (IGDA) Malta and of the Gaming group at the same University. He pursued his Ph.D. on the Semantic Web at the University of Sheffield in the UK under the supervision of Professor Yorick Wilks.



Karl Sant Fournier is a Bachelor's degree graduate in Business and Computing and is a currently a part time student reading for a Masters degree with the department of Artificial Intelligence. He works full time as a Business Intelligence Developer at a local Bank. Given his knowledge of the financial industry he is focusing his thesis on Financial Time Series forecasting, using Artificial Intelligence techniques

