**1. INTRODUCTION**

One of the most prominent use cases of machine learning is “Fintech” (Financial Technology). A large subset of which is in the stock market. Financial theorists, and data scientists for the better part of the last 50 years, have been employed to make sense of the marketplace in order to increase return on investment. However, due to the multidimensional nature of the problem, the scale of the system, and inherent variation with time, it has been an overwhelmingly tough challenge for humans to solve, even with the assistance of conventional data analytics tools. However, with the onset of recent advancements in machine learning applications, the field has been evolving to utilize non-deterministic solutions. What is going on in order to make more accurate predictions. It is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess methods and technologies which allow them to gain future price information. This would imply that all publicly known information about a company, which obviously includes its price history, would already be reflected in the current price of the stock. Accordingly, changes in the stock price reflect release of new information, changes in the market generally, or random movements around the value that reflects the existing information set. While the efficient market hypothesis finds favour among financial academics, its critics point to instances in which actual market experience differs from the prediction of unpredictability the hypothesis implies. A large industry has grown up around the implication proposition that some analysts can predict stocks better than others. But that would be impossible under the Efficient Markets Hypothesis.

**2. ALGORITHM**

# Linear Regression Using Tensorflow

Linear Regression is a very common statistical method that allows us to learn a function or relationship from a given set of continuous data. For example, we are given some data points of x and corresponding y and we need to learn the relationship between them that is called a hypothesis.

In case of Linear regression, the hypothesis is a straight line, i.e,  
  
Where w is a vector called Weights and b is a scalar called Bias. The Weights and Bias are called the parameters of the model.



In statistics, linear regression is a [linear](https://en.wikipedia.org/wiki/Linearity) approach to modelling the relationship between a scalar response (or [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable)) and more [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) (or [independent variables](https://en.wikipedia.org/wiki/Independent_variable)). The case of one explanatory variable is called [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression). For more than one explanatory variable, the process is called multiplelinear regression. This term is distinct from [multivariate linear regression](https://en.wikipedia.org/wiki/Multivariate_linear_regression), where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modelled using [linear predictor functions](https://en.wikipedia.org/wiki/Linear_predictor_function) whose unknown model [parameters](https://en.wikipedia.org/wiki/Parameters) are [estimated](https://en.wikipedia.org/wiki/Estimation_theory) from the [data](https://en.wikipedia.org/wiki/Data). Such models are called [linear models](https://en.wikipedia.org/wiki/Linear_model). Most commonly, the [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of the response given the values of the explanatory variables (or predictors) is assumed to be a [function](https://en.wikipedia.org/wiki/Affine_transformation) of those values less commonly, the conditional [median](https://en.wikipedia.org/wiki/Median) or some other are also used. Like all forms of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](https://en.wikipedia.org/wiki/Conditional_probability_distribution) of the response given the values of the predictors, rather than on the [joint probability distribution](https://en.wikipedia.org/wiki/Joint_probability_distribution) of all of these variables, which is the domain of [multivariate analysis](https://en.wikipedia.org/wiki/Multivariate_analysis).

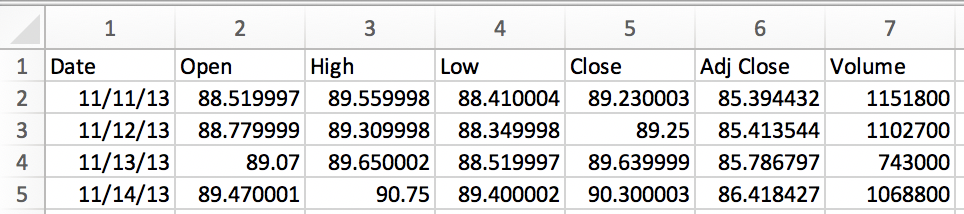
Linear regression was the first type of regression analysis to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine. Linear regression has many practical uses.

**3. CHOOSING THE DATA**

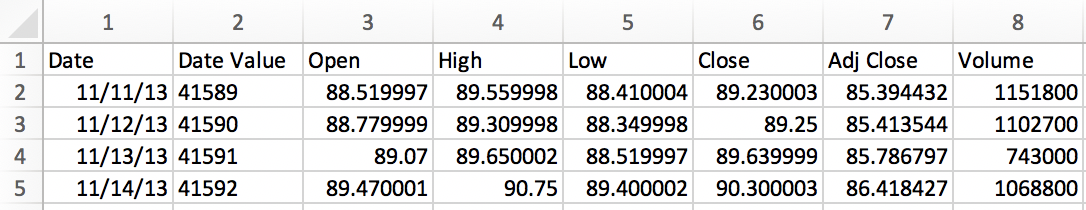
One of the most important steps in machine learning and predictive modelling is gathering good data, performing the appropriate cleaning steps and realizing the limitations.

For this example I will be using stock price data from a single stock, Amazon (ticker: AMZN). Simply go too finance.yahoo.com, search for the desired ticker. Once you are on the home page of the desired stock, simple navigate to the “Historical Data” tab, input the range of dates you would like to include, and select “Download Data.” I chose 5 years, but you can choose as far back as you would like.

Now that we have out data, let’s go ahead and see what we have. Simply open the file in Excel.



We may notice that all of the fields are numerical values, except that date value .We need to fix this. The values that we are going to pass into our model need to be in a format that can be most easily understood. So, we need to perform some “data pre-processing” steps. In our case we are going to insert a new column after 1, name it “Date Value,” and copy all of the dates from column 1 into column 2. Then select all of the data and change the type from “Date” to “Text.” The results should look like the following:



So now the file is saved as “choose\_a\_name.csv” (make sure it is a “.csv “and not one of the excel default formats).Before we start, let’s talk about limitations. You will notice that the only data we feed this model is date and price. There are many external factors that affect the price outside of the historical price. Highly robust models might utilize external data such as news, time of the year, social media sentiment, weather, price of competitors, market volatility, market indices, etc.

**4. CHOOSING THE MODEL**

So now that we have data cleaned up, we need to choose a model. In this case we are going to use a neural network to perform a regression function. A regression will split out a numerical value on a continuous scale. By that our model may be used for classification efforts, which would yield a categorical output. In this situation, we are trying to predict the price of a stock on any given day.

To build our model we are going to use TensorFlow, a simplified module called TFANN which stands for “TensorFlow Artificial Neural Network.” In order to do this, we are going to use Google Colab. Simply navigate to colab.research.google.com, it is free virtual python notebook environment.

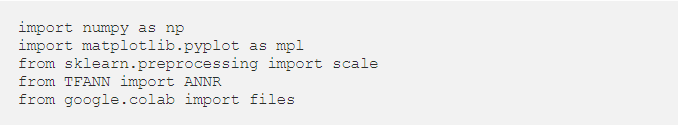
One can see that the networks rapidly adapts to the basic shape of the time series and continues to learn finer patterns of the data. This also corresponds to lower the learning rate during model training in order not to overshoot the optimization minimum. We have a pretty close fit to the test data. The final test MSE equals 0.85 (it is very low, because the target is scaled). The mean absolute percentage error of the forecast on the test set is equal to 5.31%.

**5. BUILDING THE MODEL**

First we need to install TFANN. Open a new Colab notebook (python 3). Colab has numerous libraries which can be accessed without installation; however, TFANN is not one of them so we need to execute the following command:



Now dependencies should be imported:



NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with this arrays. It is the fundamental package for scientific computing with Python.

It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Matplotlib is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK+](https://en.wikipedia.org/wiki/GTK%2B). There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged.[SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](https://en.wikipedia.org/wiki/John_D._Hunter), has an active development community, and is distributed under a [BSD-style license](https://en.wikipedia.org/wiki/BSD_licenses). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012,[[5]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-5) and further joined by Thomas Caswell.

Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Please note that scikit-learn is used to build models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

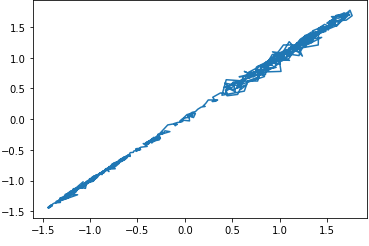
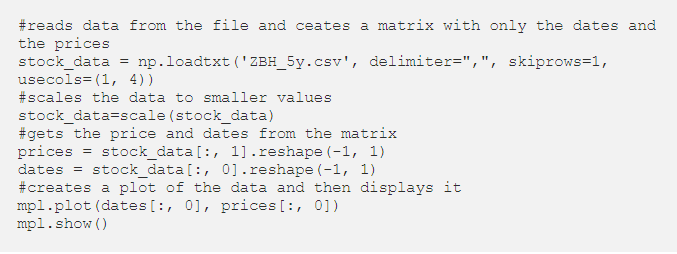
**Supervised learning algorithms:** Think of any supervised learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of algorithms is one of the big reasons for high usage of scikit-learn. I started using scikit to solve supervised learning problems and would recommend that to people new to scikit / machine learning as well.

Tensorflow is an open-source computation library made by Google. It is a popular choice for creating applications that require high-end numerical computations and/or need to utilize Graphics Processing Units for computation purposes. These are the main reasons due to which Tensorflow is one of the most popular choices for Machine Learning applications, especially Deep Learning. It also has APIs like Estimator which provide a high level of abstraction while building Machine Learning Applications. In this article, we will not be using any high-level APIs, rather we will be building the Linear Regression model using low-level Tensorflow in the Lazy Execution Mode during which Tensorflow creates a **Directed Acyclic Graph**or DAG which keeps track of all the computations, and then executes all the computations done inside a **Tensorflow Session**

Now we need to import the data that we have already processed. To do this we will execute the following command, which will provide us with a window to upload the .csv file.



The below code will read the data from the file we have uploaded. Then it creates a matrix with only dates and prices. It uses data from column one to four for this step. In the next step for better understanding the data is reduces to smaller values. From here all the data is displayed on the graph using matplot.

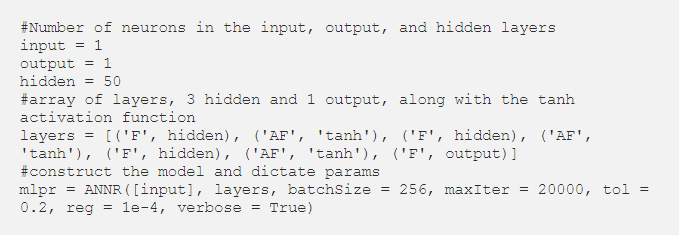


Note, that the scale is no longer in dollars on the y-axis and those arbitrary integer-date values on the x-axis. We have scaled the data down to make the learning process more effective. Try writing some code to return the scale of the y-axis back to dollars and the x-axis to years.

Now, we need to construct the model. In this case we will use *one* input and output neuron (input date, output price) and will have *three* hidden layers of 25 neurons each. Each layer will have a “tanh” activation function.

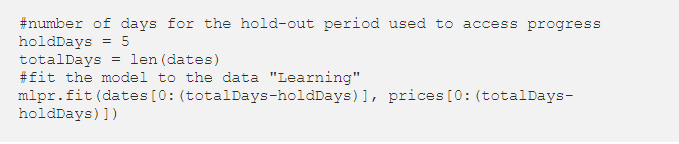
Artificial Neural Network (ANN) is an efficient computing system whose central theme is borrowed from the analogy of biological neural networks. ANNs are also named as “artificial neural systems,” or “parallel distributed processing systems,” or “connectionist systems.” ANN acquires a large collection of units that are interconnected in some pattern to allow communication between the units. These units, also referred to as nodes or neurons, are simple processors which operate in parallel.

Every neuron is connected with other neuron through a connection link. Each connection link is associated with a weight that has information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually excites or inhibits the signal that is being communicated. Each neuron has an internal state, which is called an activation signal. Output signals, which are produced after combining the input signals and activation rule, may be sent to other units.

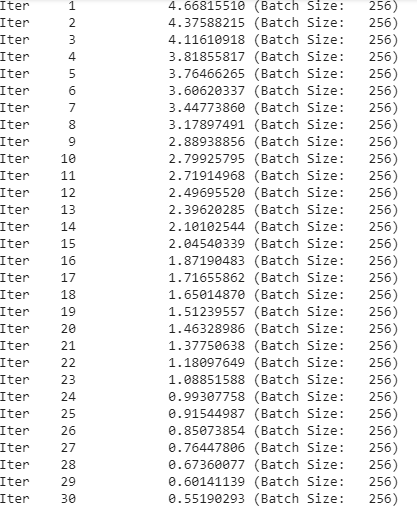


**6. TRIANING THE MODEL**

While creating any machine learning model the data is divided in such a way that 80% of the data is used for training the model and 20% for testing the model. But, in this case the whole model is used for both training and testing so that the accuracy of the model can be tested.

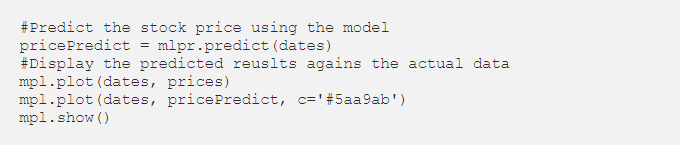


In the above code the hold days is fixed as five. The total number of days is taken as total length of the dates. Then the model is fixed to learning and giving the output using matplot.

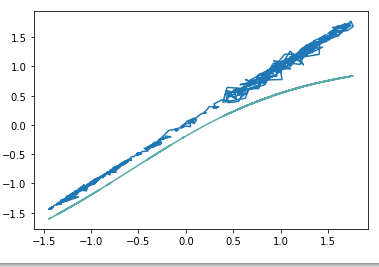


**7. TESTING THE MODEL**

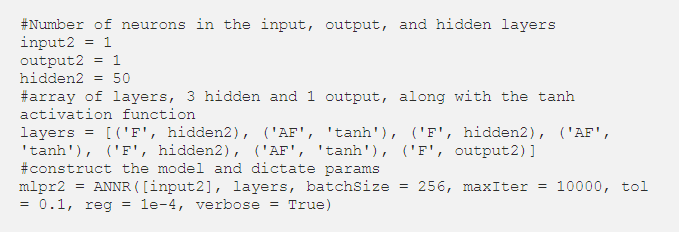
Once the training is complete, we can execute the following commands to see how we did.



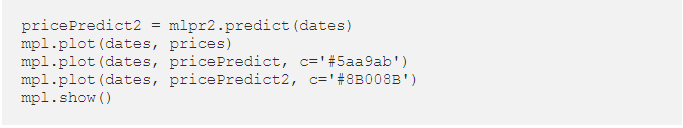
In the above code the stock price is predicted using the model. Later both the predicted plotting and the actual plotting are displayed for comparison.



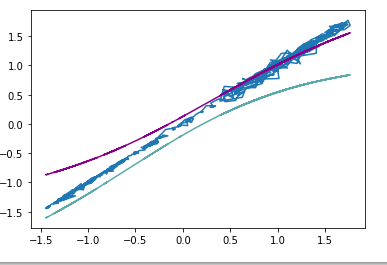
To create more effective prediction now the model is again trained by reducing the error tolerance to .1.



Once it has finished the training with new error tolerance it is again tested. Now the testing is done with the new variables so that the previous variables are also executed for comparison of accuracy.



Now the output is predicted as following:



To create more effective prediction now the model is again trained by reducing the error tolerance to .08 and .09. By this we check the model working how accurate it is predicting.. By this Prediction, we know the price variation of stock price on particular day

**8. CONCLUSION**

By using this model we can know the price variations of a particular company on a particular day starting from opening to closing. We have represented it using graph by taking dates on x-axis and price on y-axis. Stock price prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient market hypothesis suggests that stock price reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information. They evaluate a company's past performance as well as the credibility of its accounts. Many performance ratios are created that aid the fundamental analyst with assessing the validity of a stock, such as the P/E ratio. The principle being that a company is worth all of its future profits added together. These future profits also have to be discounted to their present value. By this prediction, we know the price variation of stock price on particular day and also we can check the model whether it is accurate prediction or not by decrease the error tolerance and see the changes in the graph.

**REFERENCES**

**1.**

**[https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price- machinelearningnd-deep-learning-techniques-python/](https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price- machinelearningnd-deep-learning-techniques-python/ )**

**2.**  [**https://towardsdatascience.com/simple-stock-price-prediction-with-ml-in-python- learners-guide-to-ml-76896910e2ba**](%20https://towardsdatascience.com/simple-stock-price-prediction-with-ml-in-python-%20%20%20%20%20%20learners-guide-to-ml-76896910e2ba)

**3**[**.** **https://www.tutorialspoint.com/numpy/index.htm**](.%20https:/www.tutorialspoint.com/numpy/index.htm)

**4.** [**https://pythonspot.com/matplotlib/**](https://pythonspot.com/matplotlib/)

**5**[**.** **https://en.wikipedia.org/wiki/Scikit-learn**](.%20https:/en.wikipedia.org/wiki/Scikit-learn)