**Machine Learning –CS 545**

**Group Final Programming Project**

**Group Members:**

Nishna Reddy, Sowmya Sri, Sukanya Kothapally

**Abstract:**

The main objective of this project is to find the best model to predict the future stock market value (for next 20 days) from historical data for a particular company (ex: Google). This project will demonstrate how to “tune” the model will affect the results. The experimental results demonstrate that traditional ML algorithms may have a better performance.

**Introduction**:

Predictions on stock markets have been object of studies

for many decades, but given it’s innate complexity, dynamism

and chaoticness, it has proven to be a very difﬁcult task. The

number of variables and sources of information considered

are immense and the signal-to-noise ratio insigniﬁcant. That

makes the task of predicting stock market prices behavior in

the future a very hard one. For many decades, there’s been

discussions in Science regarding the possibility of such a feat

and it’s notable in the related literature that most prediction

models fail to provide precise prediction in a general sense

Predictions on stock markets have been object of studies for many decades, but given its innate complexity, dynamism and chaotic ness, it has proven to be a very difﬁcult task. According to the forecast of stock price trends, investors trade stocks. In recent years, many researchers focus on adopting machine learning (ML) algorithms to predict stock price trends. The number of variables and sources of information considered are immense and that makes the task of predicting stock market prices behavior in the future a very hard one. For many decades, there’s been discussions in Science regarding the possibility of such a feat and it’s notable in the related literature that most prediction models fail to provide precise prediction in a general sense.

The main contributions of this work are the following (1) a prediction model for stock markets using ML algorithms;(2) the validation of the model using real data from google stock exchange; (3) evaluation of the model by comparing and analyzing it against some typical baselines. As every algorithm has its own advantage and disadvantage this research will study 3 different machine learning techniques commonly being in use for stock market prediction. A comparative approach will be followed to find out optimal technique for stock market prediction under different circumstances. Comparison will be made on the basis of their performance on the same input dataset. At the end, performance of these 3 models is evaluated and summarize that which algorithm is better in predicting stock price.

for many decades, but given it’s innate complexity, dynamism

and chaoticness, it has proven to be a very difﬁcult task. The

number of variables and sources of information considered

are immense and the signal-to-noise ratio insigniﬁcant. That

makes the task of predicting stock market prices behavior in

the future a very hard one. For many decades, there’s been

discussions in Science regarding the possibility of such a feat

and it’s notable in the related literature that most prediction

models fail to provide precise prediction in a general sense

Predictions on stock markets have been object of studies

for many decades, but given it’s innate complexity, dynamism

and chaoticness, it has proven to be a very difﬁcult task. The

number of variables and sources of information considered

are immense and the signal-to-noise ratio insigniﬁcant. That

makes the task of predicting stock market prices behavior in

the future a very hard one. For many decades, there’s been

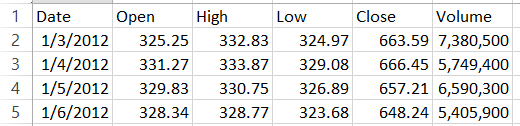
discussions in Science regarding the possibility of such a feat

and it’s notable in the related literature that most prediction

models fail to provide precise prediction in a general sense

**Choosing data:**

One of the most important steps in machine learning and predictive modeling is gathering good data, performing the appropriate cleaning steps and normalize data. The Google stock price data from 2012-2016 is used (source: kaggle). The snapshot of the data is listed below.

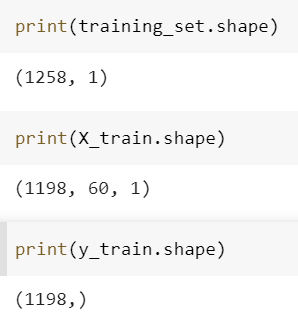


The profit or loss calculation is usually determined by the closing price of a stock for the day, hence we will consider the closing price as the target variable. This will be the training dataset.

**Data Preprocessing:**

The training dataset in divided into two parts x train and y-train. Each row of x-train consists of previous 60 days’ independent time steps. Whereas, y-train is the output, which are called labels consists of 61st closed value, depending on x-train’s previous 60 time steps. For normalizing the data, MinMaxScaler function from sklearn is used. Considering the close value, create a data structure with 60 time steps and 1 output.

After the above data preprocessing the x-train and y-train shapes are shown below.



**ML Algorithms and Their Parameter Settings**

1. **LSTM (Long Short Term Memory)**

he algorithm of choice here is the LSTM (Long-Short term

memory) network. It’s a type of recurrent network that has

proved very successful on a number of problems given its

The algorithm of choice here is the LSTM (Long-Short term memory) network. It’s a type of recurrent network that has proved very successful on a number of problems given its capability to distinguish between recent and early examples by giving different weights for each while forgetting memory, it considers irrelevant to predict the next output. In that way, it is more capable to handle long sequences of input when compared to other recurrent neural networks that are only able to memorize short sequences. You always have to give a three-dimensional array as an input to your LSTM network. Where the first dimension represents the batch size, the second dimension represents the time-steps and the third dimension represents the number of units in one input sequence.

**Modules:**

Sequential for initializing the neural network

Dense for adding a densely connected neural network layer

LSTM for adding the Long Short-Term Memory layer



  
**Parameters:**

Adam optimization is used which replaces the algorithm for gradient descent for training ML models. The default learning rate for this is 0.001 and the activation function is linear.

Hidden layers: 3 hidden layer.

return\_sequences =True; which determines whether to return the last output in the output sequence, or the full sequence

input\_shape: shape of our training set.

Dropout layers: Dropout is used in the hidden layers to avoid overfitting. we specify 0.2, meaning that 20% of the layers will be dropped.

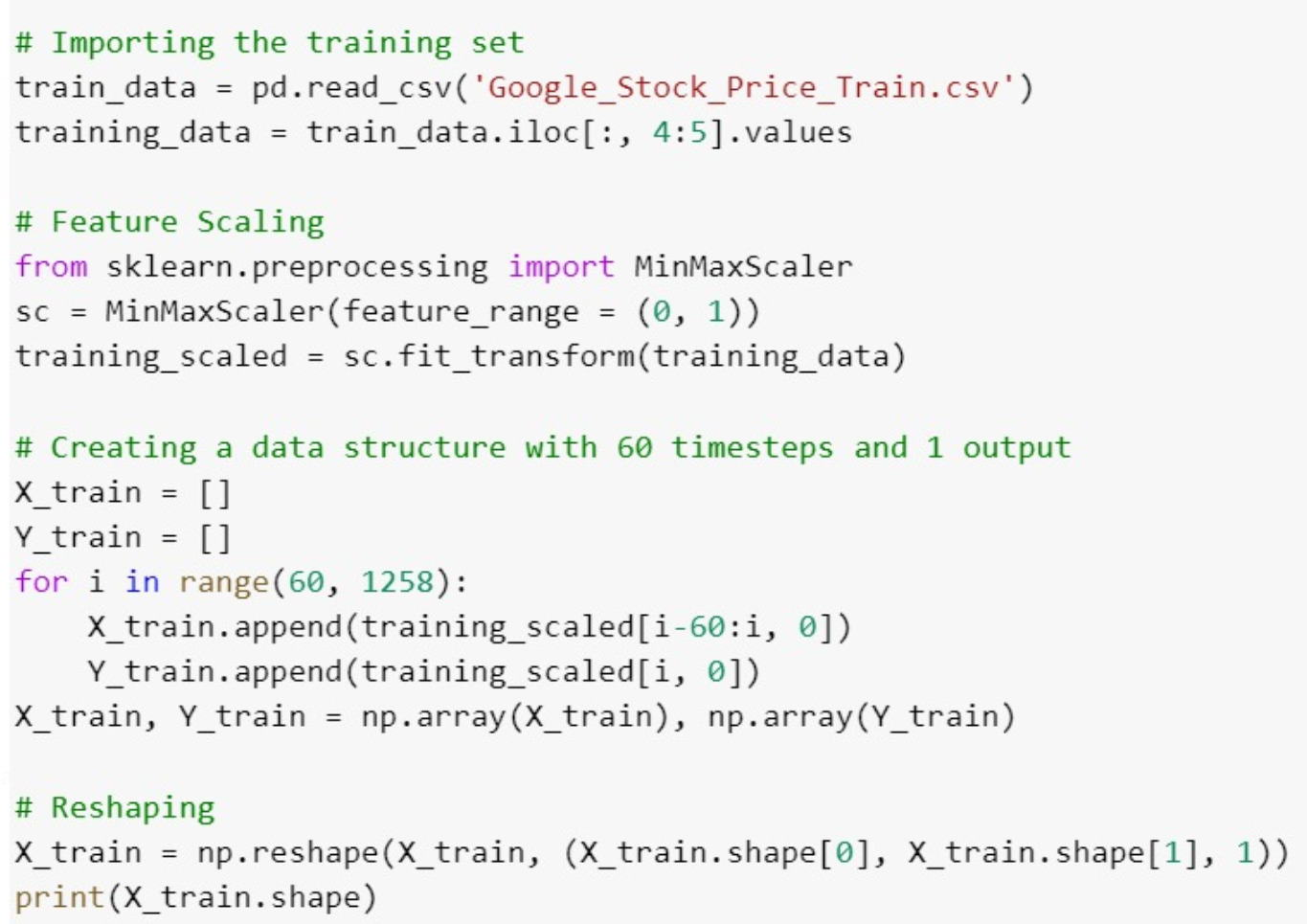
loss function: mean\_squarred\_error

learning\_rate=0.001, beta\_1=0.9, beta\_2=0.999(beta\_1=The exponential decay rate for the first moment estimates)

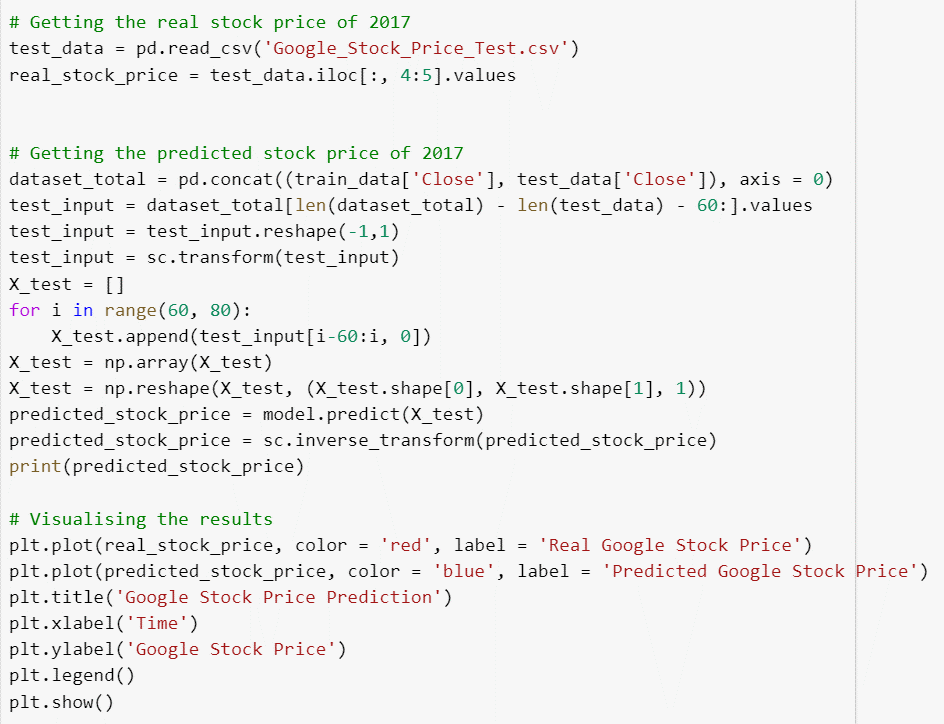
Epochs: 100

**Implementation:**

In order to predict future stock prices after loading in the test set, merge the training set and the test set on the 0 axis and set the time step as 60 as done in training data set. Use MinMaxScaler to transform the new dataset and reshape the dataset. After making the predictions we use inverse\_transform to get back the stock prices in normal readable format.







**2. SVR (Support Vector Regression)**

A Support Vector Regression (SVR) is a type of Support Vector Machine, and is a type of supervised learning algorithm that analyzes data for [regression](https://en.wikipedia.org/wiki/Regression_analysis) analysis unlike svm is used for classification.

**Parameters:**

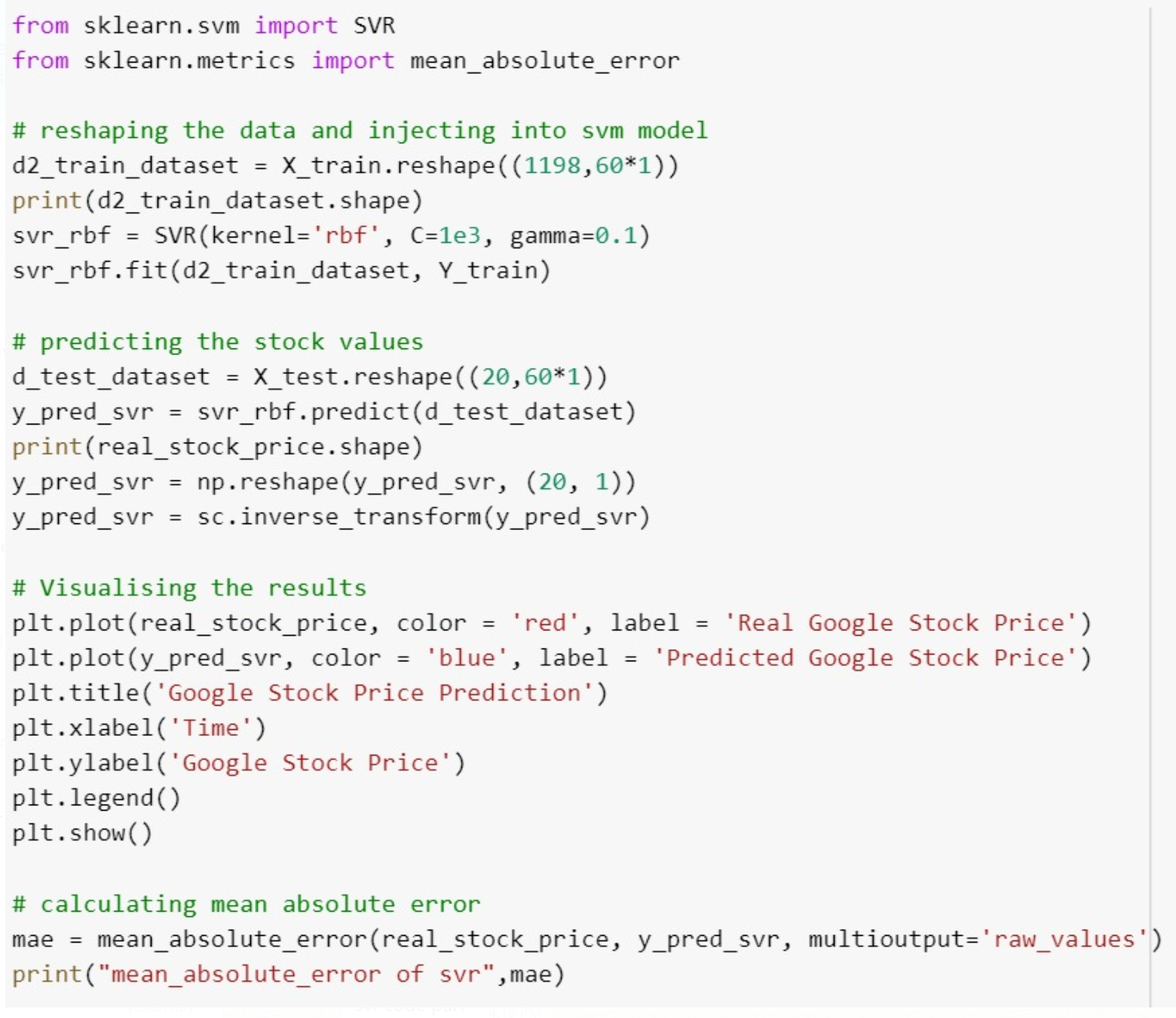
rbf(radial basis function) is used ,which is a gaussian Kernal function.

C is the regularization parameter that controls the trade-off between the slack variable penalty (misclassifications) and width of the margin. Small C makes the constraints easy to ignore which leads to a large margin.

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'.

**Implementation:**

Using rbf Kernel, SVR model is built and fit the training dataset to the model and predict the model. In the context of price prediction, the goal is not necessarily classification into groups but estimation of real values. This project therefore uses SVR, which is employed to obtain a regression model used to predict stock prices.



**3. Linear Regression**

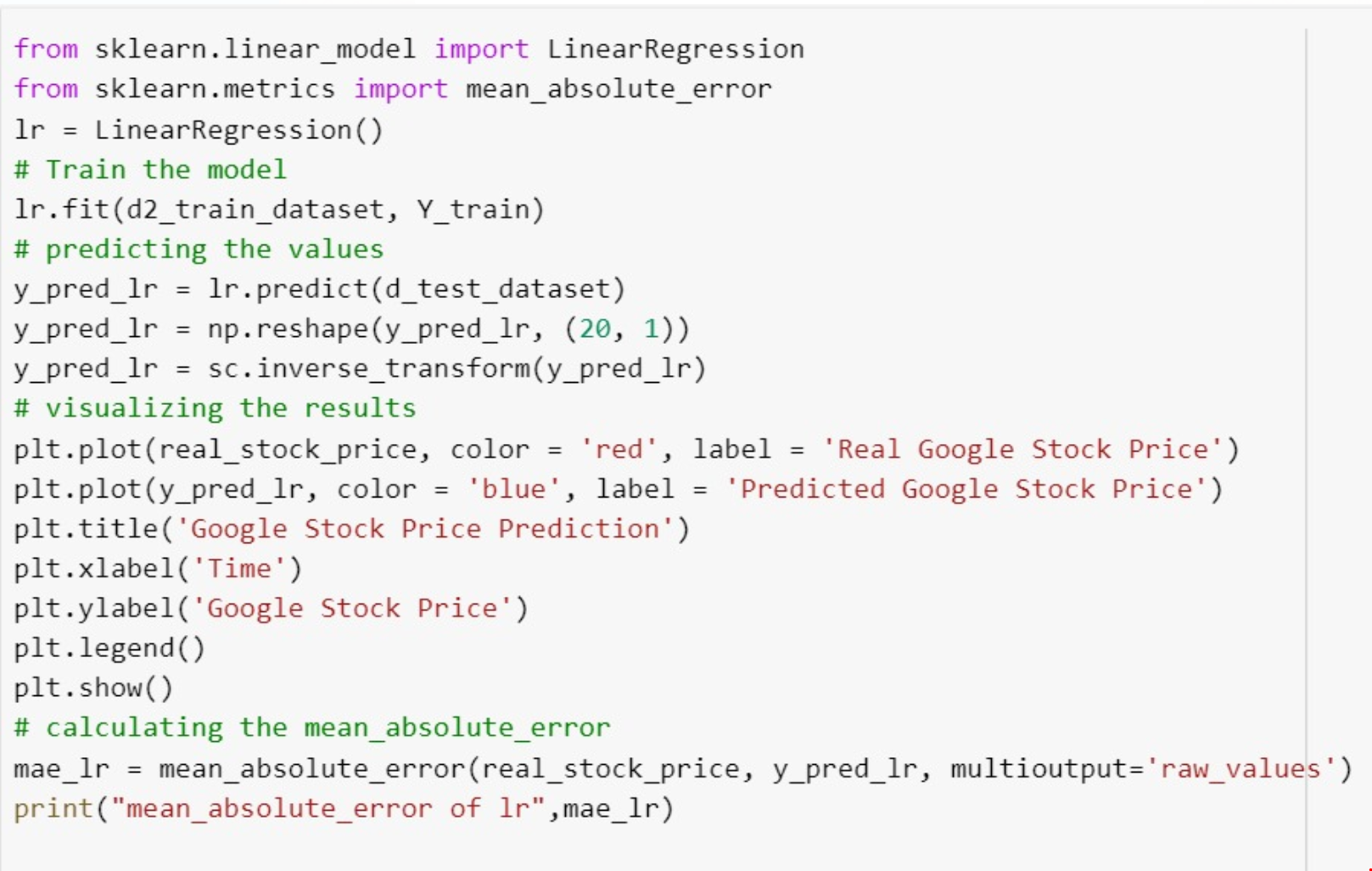
Linear regression is a method used to model a relationship between a dependent variable (y), and an independent variable (x). With simple linear regression, there will only be one independent variable x. There can be many independent variables which would fall under the category of multiple linear regression. The goal of multiple regression is to model the linear relationship between your independent variables and your dependent variable.

**Parameters:**

Linear Regression object uses Ordinary Least Squares solver from scipy, as LR is one of two classifiers which have closed form solution

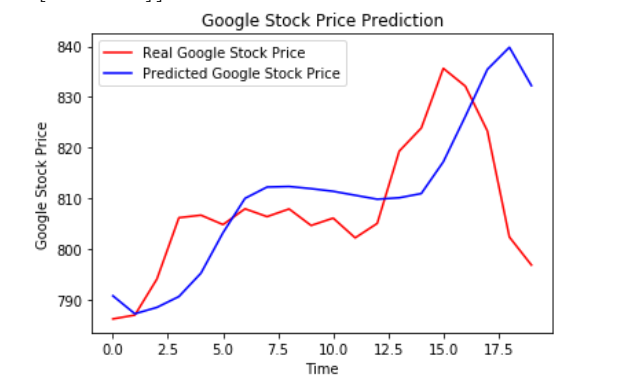
**Implementation:**

In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. So, in the implementation, we have 60 explanatory variables, and one(y) is the dependent variable, as its value is dependent on x. It will be using scikit-learn, csv, numpy and matplotlib packages to implement and visualize linear regression. The fit method fits the X\_train and y\_train (x’s and y’s) to generate coefficient and constant for regression. Finally, the predict method finds the price(y) for the given input and returns the predicted price.

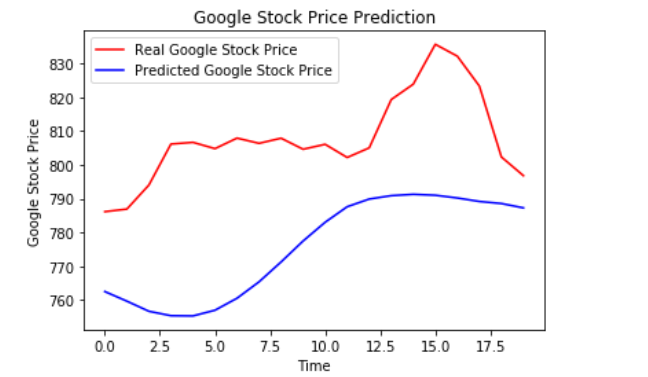


**Evaluation:**

**LSTM Prediction (60 timesteps):**

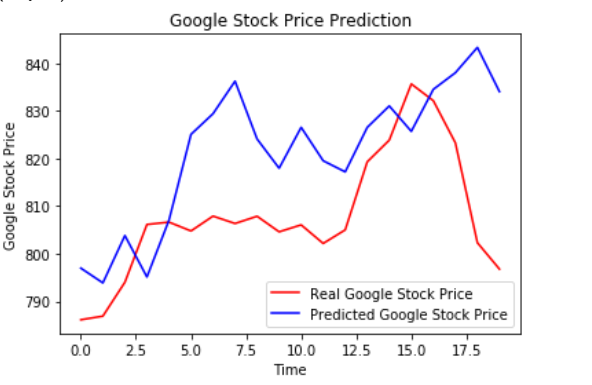


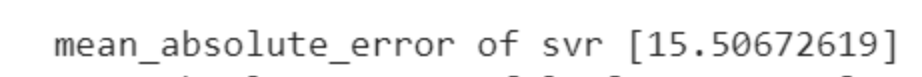
**LSTM Prediction (10 timesteps):**



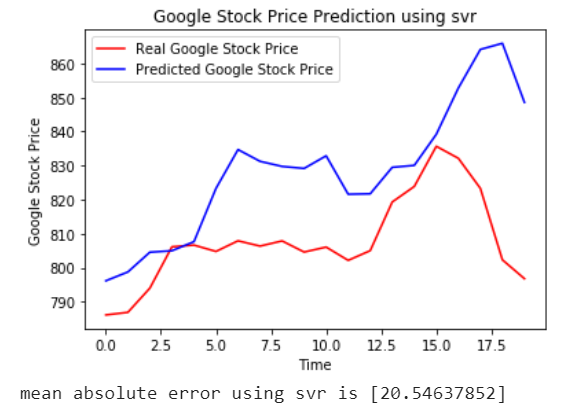


**SVR Prediction (60 timesteps):**

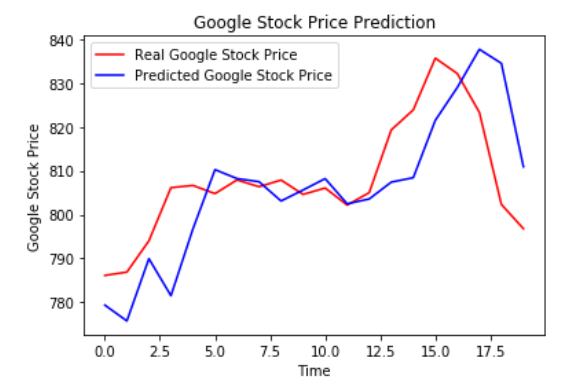


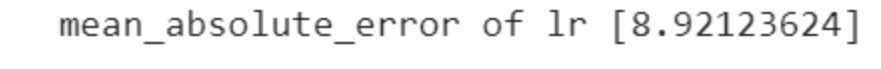


**SVR Prediction (10 timesteps):**

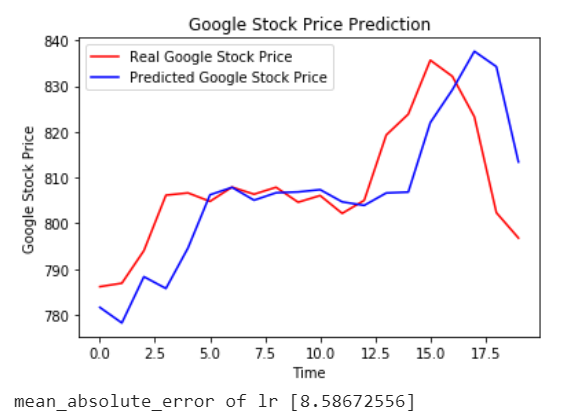


**LR Prediction (60 timesteps):**





**LR Prediction (10 timesteps):**



**Mean Absolute Error:**

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

From graphs LSTM and LR are able to learn the model and predict the stock price where as in SVR there is no exact correlation between the real and predicted stock price. The mean absolute error of LSTM is very less compared to that of LR and SVR. This clearly shows how powerful LSTMs are for analyzing time series and sequential data.

By considering the above factors, it can be said that LSTM performed better than LR and SVR on this problem.

**Conclusion:**

Our conclusions are significant to choose the best algorithm for stock trading in different markets. The problem is that we do not yet know ahead of time which stocks the model will be able to predict accurately and which it will not, so profiting off the model is still difficult without more experimentation.

**References:**

1.*https://www.researchgate.net/publication/318329563\_Stock\_market's\_price\_movement\_prediction\_with\_LSTM\_neural\_networks*

2.[*https://www.cs.princeton.edu/sites/default/files/uploads/saahil\_madge.pdf*](https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf)

3. *https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8212716*