**Final Project**

Hemalatha Subbiah

Bellevue University

DSC 550: Data Minning

Professor (or Dr.) Brett Werner

Date:11/16/2023

Stock prediction using machine learning is a challenging yet intriguing application of artificial intelligence. It involves leveraging historical stock price and trading volume data to develop models that can make predictions about future stock movements. Time series analysis plays a crucial role in this domain as it deals with data points ordered chronologically.

**Problem Statement: Time Series Stock Prediction:**

1. I want to analyze the data, clean and prepare for model training.

2. I want to build model to predict the Stock market movement using deep learning techniques

3. I want to generate realistic buy/sell signals for the next day based on future stock price estimates using time series modeling. A time series is a sequence of data points collected or recorded over successive and equally spaced intervals of time.

4. I want to implement this using Tensor flow open-source framework and want to dig deeper on Recurrent Neural Network.

**why it is important/useful to solve this problem:**

Financial markets are dynamic and influenced by various factors, making stock price prediction a challenging task. Stock prices are influenced by a multitude of factors, including market sentiment, economic indicators, and global events. The complexity of these interactions makes accurate predictions difficult. In this context, the goal is to develop a predictive model capable of forecasting future stock prices using historical data.

**How would you pitch this problem to a group of stakeholders to gain buy-in to proceed?**

The stock market is a key component of the global economy, providing businesses with funding for growth and expansion. It is also a popular way for individuals to invest and grow their wealth over time. Prediction of Stock prices using ML makes informed investment decisions, risk management and competitive advantage.Though the problem solving has own challenges as data limitation, Inherent Uncertainty, and stock market complexity.ML can have potential benefits, challenges, and feasibility of the approach. Many algorithms can try, and accuracy can be relatively compared and strategies can be laid to address overfitting and underfitting issues in the model. Moreover, a lot of approaches are available for validating model robustness.For stocks or share prices, time series forecasting is common to track the price movement of the security over time.

**Explain where you obtained your data**

A Google stock dataset would likely contain historical and possibly real-time data on Google's stock traded on stock exchanges. The dataset consists of historical stock prices, including but not limited to Open, High, Low, Close, and Volume. The stock market pulled from Kaggle.

Variables:

Date: Represents the date of the relevant Transaction Day.

Open: Represents the initial share price of the relevant Trading Day.

High: Represents the highest price of the relevant Trading Day.

Low: It represents the lowest price of the relevant trading day.

Close: It represents the closing price of the stock on the relevant trading day.

Adj Close: Represents the adjusted closing price of the stock of the relevant trading day.

Volume: It represents the trading volume information of the relevant trading day

EDA :

Handling missing values is a crucial step in the data preprocessing pipeline. Before deciding on how to handle missing values, it's helpful to understand the distribution and pattern of missing data. While performing the EDA of the Tesla Stock Price data we will analyze how prices of the stock have moved over the period of time and how the end of the quarter affects the prices of the stock. There were up and down trends observed many times during the last 20 years. The stock price has experienced continuous growth since 2002. There was a sharp fall observed between 2008- 2009, similarly, in Dec-2016, a sharp fall was observed. There are multiple sharp peaks observed for multiple years in Google stock price, which could be a sign of good growth or a stock bubble.

A screenshot of a computer

Description automatically generated

In my project used in this example:

• df.isnull().sum() prints the count of missing values in each column

• sns.heatmap(df.isnull(), cmap='viridis', cbar=False, yticklabels=False) creates a heatmap where missing values are marked in a different color.

This visualization makes it easy to identify columns with a high concentration of missing values.

In Google dataset no null values are found but few outliners are spotted.

A screenshot of a computer

Description automatically generated

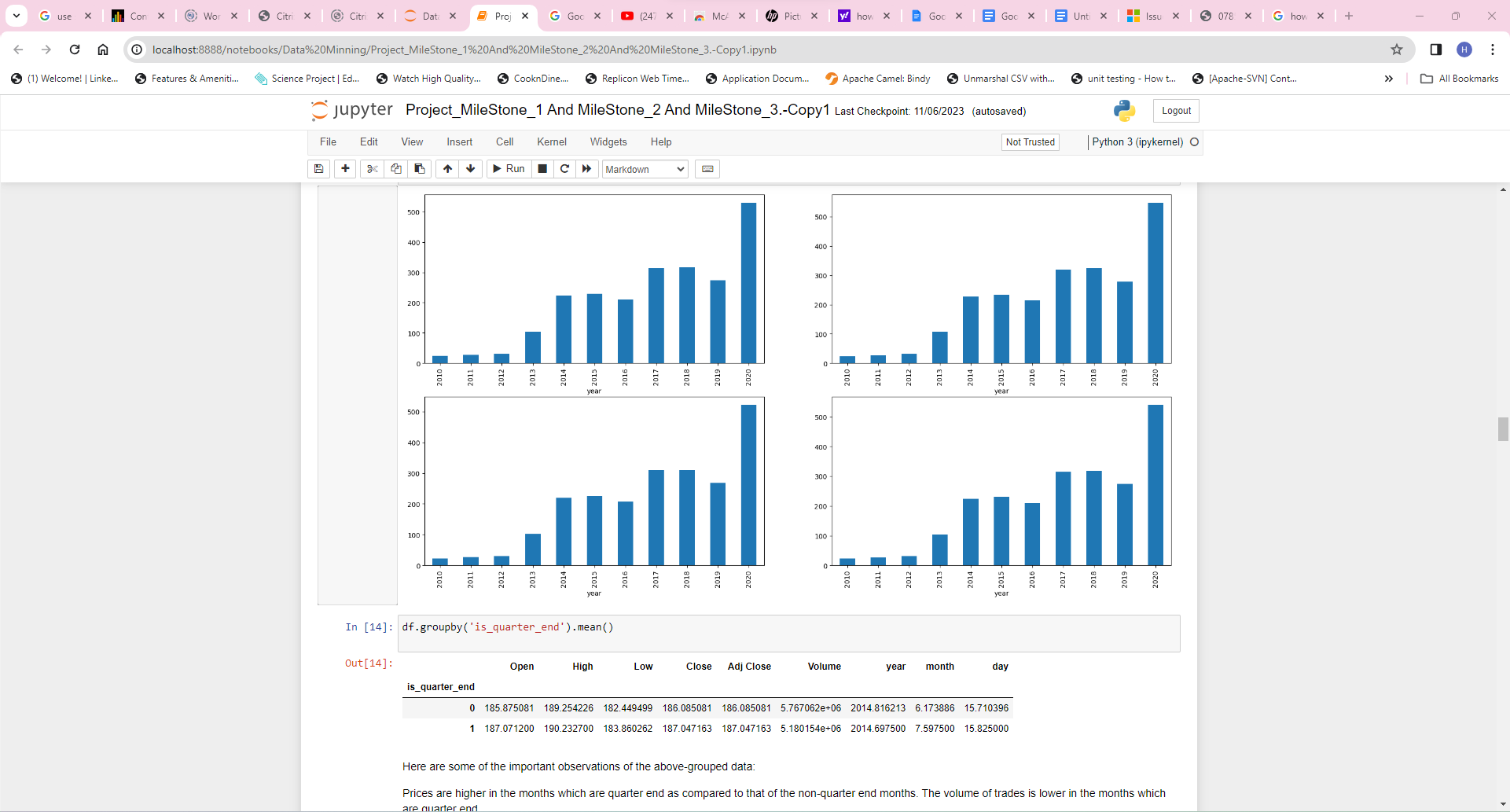
I can conclude that only volume data contains outliers in it but the data in the rest of the columns are free from any outlier.

While performing the EDA of the Google Stock Price data we will analyze how prices of the stock have moved over the period and how the end of the quarter affects the prices of the stock.From the above boxplots I can conclude that only volume data contains outliers in it but the data in the rest of the columns are free from any outlier.

A screenshot of a computer

Description automatically generated

In the distribution plot of OHLC data, we can see two peaks which means the data has varied significantly in two regions. And the Volume data is left-skewed. We can see two peaks which means the data has varied significantly in two regions. And the Volume data is left-skewed.



Prices are higher in the months which are quarter end as compared to that of the non-quarter end months. The volume of trade is lower in the months which are quarter end.

**Data Preparation:**

Correlation Matrix Analysis:

A screenshot of a computer

Description automatically generated

Above visualizes the correlation between different variables in a dataset. A correlation coefficient is a number between -1 and 1 that tells you the strength and direction of a relationship between variables. In other words, it reflects how similar the measurements of two or more variables are across a dataset. We can decide on the feature selection based on the above correlation coefficients.

Created features based on technical indicators like open and close price, low and high prices and figuring about the quarter end.

features = df[['open-close', 'low-high', 'is\_quarter\_end']]

target = df['target']

 Three more columns namely ‘day’, ‘month’ and ‘year’ all these three have been derived from the ‘Date’ column which was initially provided in the data

df['is\_quarter\_end'] **=** np.where(df['month']**%**3**==**0,1,0)

A quarter is defined as a group of three months. Every company prepares its quarterly results and publishes them publicly so that people can analyze the company’s performance. These quarterly results affect the stock prices heavily, which is why we have added this feature because this can be a helpful feature for the learning model.

**Normalization:**

Normalize data to ensure that features are on a similar scale, preventing certain features from dominating others. StandardScaler is a preprocessing technique used in machine learning to standardize the features of a dataset. Standardization is a process that rescales the features so that they have the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. This transformation is also known as z-score normalization.

scaler = StandardScaler()

features = scaler.fit\_transform(features)

**Model Building and Evaluation :**

[Machine learning algorithms](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/) such as [regression](https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/), [classifier](https://www.analyticsvidhya.com/blog/2021/09/a-complete-guide-to-understand-classification-in-machine-learning/), and [support vector machine](https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/) (SVM) help predict the stock market.

Choose appropriate time series forecasting models, such as autoregressive models, moving averages, or machine learning algorithms like Long Short-Term Memory (LSTM) networks or Gradient Boosting Machines. Common algorithms include recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and traditional machine learning models like support vector machines (SVMs) and random forests.

LSTM is used in Recurrent Neural Networks for sequence models and time series data. LSTM is used to avoid the vanishing gradient issue which is widely occurred in training RNN. To stack multiple LSTM in TensorFlow it is mandatory to use return sequences = True. Since our data is time series varying we apply no activation to the output layer and it remains as 1 node.

**Training and Validation:**

* Split the dataset into training and validation sets as 80-20 model.
* Train the selected model on historical data and validate its performance on unseen data.

Derived models from training data in the algorithms below and derived training and validation accuracy as below.

Logistic Regression () :

Training Accuracy: 0.5228802330060918

Validation Accuracy: 0.4923371647509579

SVC (kernel='poly', probability=True) :

Training Accuracy: 0.5294546963173692

Validation Accuracy: 0.46257525998905313

XGBClassifier

Training Accuracy: 0.9382749759754802

Validation Accuracy: 0.4496784345922277

Finally compiled the model with LSTM with three essential parameters:

* optimizer – method that helps to optimize the cost function by using gradient descent.
* loss – loss function by which we monitor whether the model is improving with training or not.
* metrics – evaluate the model by predicting the training and the validation data.

A screenshot of a computer

Description automatically generated

* Epoch size does not always increase accuracy. After one epoch in a neural network, all of the training data had been used to refine the models' parameters. Epoch sizes may boost precision up to a certain limit, beyond which the model begins to overfit the data.

Visualization on Predicted data vs Actual data:

A screenshot of a computer

Description automatically generated

Conclusion:

Stock prediction using machine learning is a dynamic field that requires a combination of domain expertise, data engineering, and advanced modeling techniques. Despite the challenges, successful implementations can provide valuable insights for investors and traders in the financial markets.

Be mindful of ethical considerations, especially if the predictions might influence financial decisions. We need to evaluate, and cost estimate the computational efficiency of each algorithm, especially for large datasets.

Always consider the interpretability of the models, especially if stakeholders require an understandable rationale behind predictions.

And finally remember that there is no one-size-fits-all solution, and the choice of the best algorithm may depend on the specific characteristics of the stock data and the problem at hand. Additionally, it's crucial to consider the potential impact of transaction costs, market frictions, and other real-world factors on the profitability of the predicted trading strategies.