**KBS Deliverable 3**

**By Group 5**

Team Members

| Aarsh Ghotra | 801275191 |
| --- | --- |
| Anupama Ramesh | 801261344 |
| Akanksha Gupta | 801214494 |
| Parth Patel | 801212088 |

**Deliverable 1:**

Dataset :

Apple Stock Data

<https://www.kaggle.com/datasets/00173d0f2fb1537217a1e4422612b2a9bf9d881a527fd948d035a897f7de55eb>

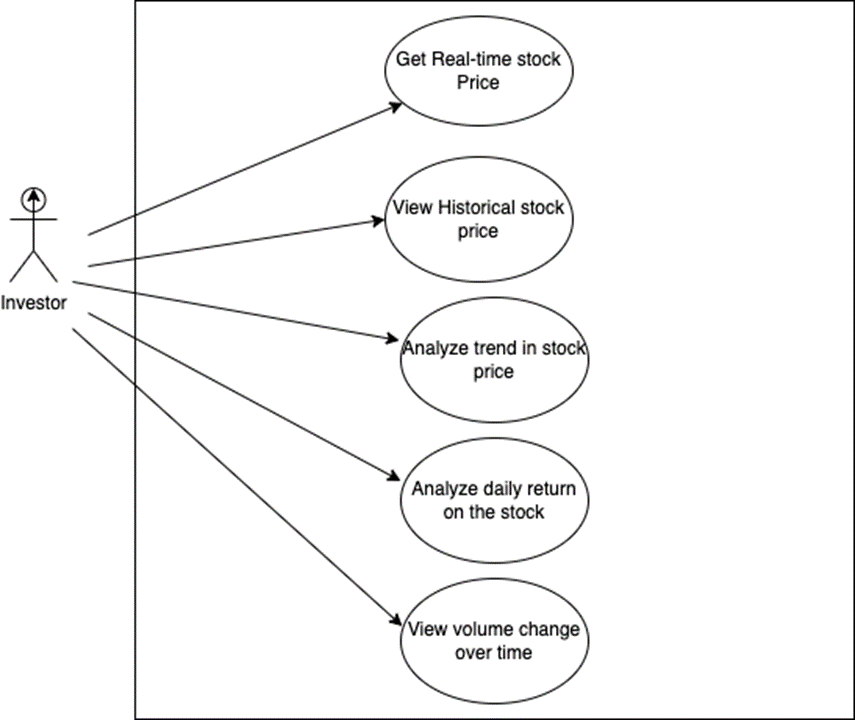
Problem statement:

Stock Market Analysis and Prediction is a project that uses Apple stocks data for technical analysis, visualization, and prediction. By examining data from the stock market, notably from some of the world's largest technological companies and others. We will use pandas to obtain stock data, visualized several elements of it, and then looked at a few different approaches to analyze a stock's risk based on its past performance history. The main goal of this project will be to compare the performance of prediction algorithms on stock market data and gain a broad understanding of this data through visualization in order to forecast future stock behavior and risk value for each stock. We are mainly trying to use NumPy, Pandas, and Data Visualization Libraries in this project.

End user based questions: (5 questions we want to answer about our data)

1. What is the change in stock's price over time?
2. What is the average daily return on the stock?
3. What is the change in the volume traded over time?
4. How much of our capital is at risk when we invest in a certain stock?
5. How can we forecast future stock performance?

Use Case Diagram:



**Deliverable 2:**

**Clean and prepare the data through Jupyter notebooks or Dataprep:**

We used Dataprep to explore, clean and prepare the data for analysis.

We created a storage bucket to store the cleaned Data in, then initialized Dataprep by enabling access to the project. Then we create a pipeline to access and manipulate the dataset. Then we import the dataset and add it to the pipeline. The dataset Apple Stock data that we chose had all string type data for all the attributes. After cleaning and transforming the data using Dataprep the data were converted to float and date data types respectively. The cleaned data was then imported to bigquery via the pipeline from  the cloud storage.

Graphical user interface, text, application, email

Description automatically generated

The data report genereated is as follows:

Graphical user interface, text, application, email

Description automatically generatedTimeline

Description automatically generated**Process the data through Vertex AI, BigQueryML, or AutoML:**

1. Firstly, we import few python libraries that we make use of in our project namely seaborn, pandas, numpy, datetime to name a few.

Graphical user interface, text, application

Description automatically generated

1. Secondly, we execute an sql query where we select all the data from our “apple stocks data” csv file and then we create a dataframe using it.

Graphical user interface, application, Word

Description automatically generated

1. Now, we generate a table of values where we have all the head values from the data set and arrange them accordingly by using the head function.

Graphical user interface, application, table, Excel

Description automatically generated

1. Here we now describe all the values by making use of the attributes mentioned in our stock data file and used describe function for it.

Graphical user interface

Description automatically generated

1. After the above the step, we now make use of info() which prints information about the dataframe. This information contains the number of columns, columns labels, data types to name a few. We also make use of isNull() which **returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False**. Sum()  **returns the number of missing values in the data set**. A simple way to deal with data containing missing values is to skip rows with missing values in the dataset.

Graphical user interface, application

Description automatically generated

1. Now, we generate a new feature named “Diff” which is the difference of highest stock price for the day and lowest stock for the day the from our daily stocks data.

Graphical user interface, application

Description automatically generated

1. Here, we make use of pct\_change , that calculates the percent change in the DataFrame between the current and prior element. So now we generate a new feature named Daily Return which is the result of percentage change of the stocks value for each day.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. From here we do data preparation where we use our stocks value attributes namely Open, close,Low, adjClose, Daily\_Return etc where we predict y = dataset[‘High’] with the use of x = dataset[[‘Open’, ‘Close’,’Low’, ‘AdjClose’, ‘Daily\_Return’]] and then we generate a table by making use of head() method.

Graphical user interface, application

Description automatically generated

1. Later on we now we split our dataset values into two different sets named as training set and testing set to predict the accuracy of our model. But we also need to make sure that we are using the most efficient model for our data set. So we have considered using Linear Regression Model to train our data because This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line.
2. Now with help of LinearRegression(), with simple linear regression when we have a single input, we can use statistics to estimate the coefficients. This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistics.

Graphical user interface, text, application

Description automatically generated

11. We now find out the RMS (root mean square), also known as the quadratic mean, is the square root of the arithmetic mean of the squares of a series of numbers.

RMSE (root mean square) gives us the difference between actual results and our calculated results from the model. It defines the quality of our model (which uses quantitative data), how accurate our model has predicted, or the percentage of error in our model.

A picture containing background pattern

Description automatically generated

1. Now our dataset has been trained efficiently using Linear Regression modeling and the accuracy of our dataset has been predicted.

Table

Description automatically generated

Therefore, by use of Linear Regression Modeling our dataset has been predicted to have a **99% accuracy**. We have come to this conclusion by comparing the **Actual\_High** values with **Predicted\_High** values as you can see that both the values are approximately equivalent.

**Note:** Our group considered 3 classifiers/models to train out data and predict the accuracy. Random Forest Algorithm, K-Nearest Neighbor (KNN) and Linear Regression Modeling. We preferred using Linear Regression Model over the other 2 because Linear Regression is simple to implement and easier to interpret the output coefficients. When you know you have a relationship between independent and dependent variables that have a linear relationship, this algorithm is the best to use because of its complexity when compared with any other algorithms mentioned. Random Forest is one of the simplest algorithms to use as it is more generalized but it is less interpretable because there will be a number of layers added to the model. With KNN Algorithm being an easy and simple algorithm to use, we need to be very wise while we choose the K and also it takes a large computation cost during runtime if sample size is huge.

**Deliverable 3: -**

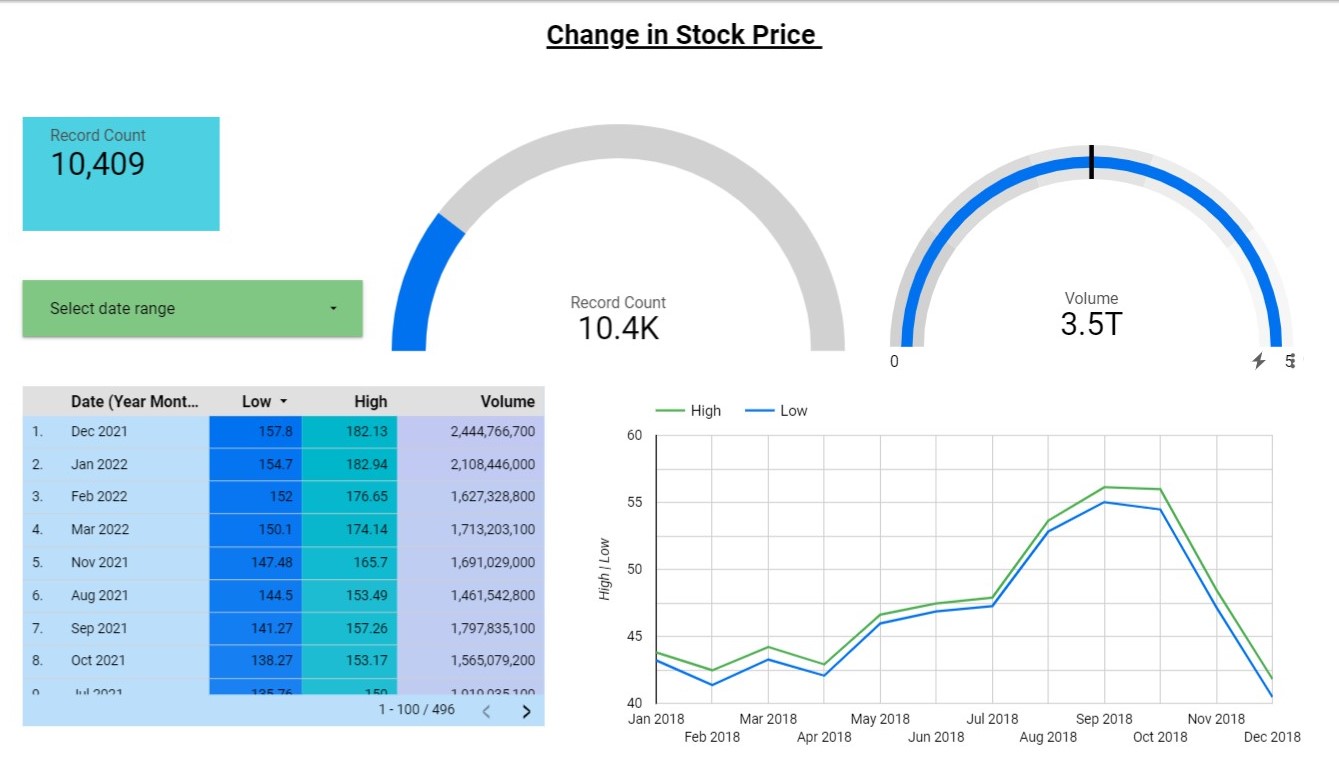
● Why your group chose the visualization tool, dashboard, UI, etc. that you did

Our group has chosen both Google Data Studio and Jupyter Notebook for visualizations. Data studio has been used to show some figures about our dataset and answer some questions for the end users as mentioned in the Deliverable1. Since the modeling is done in Jupyter Notebook,we have chosen to do some visualizations there as well. The files for both visualizations are shared in this folder.

● What visualizations / charts you considered / Summary

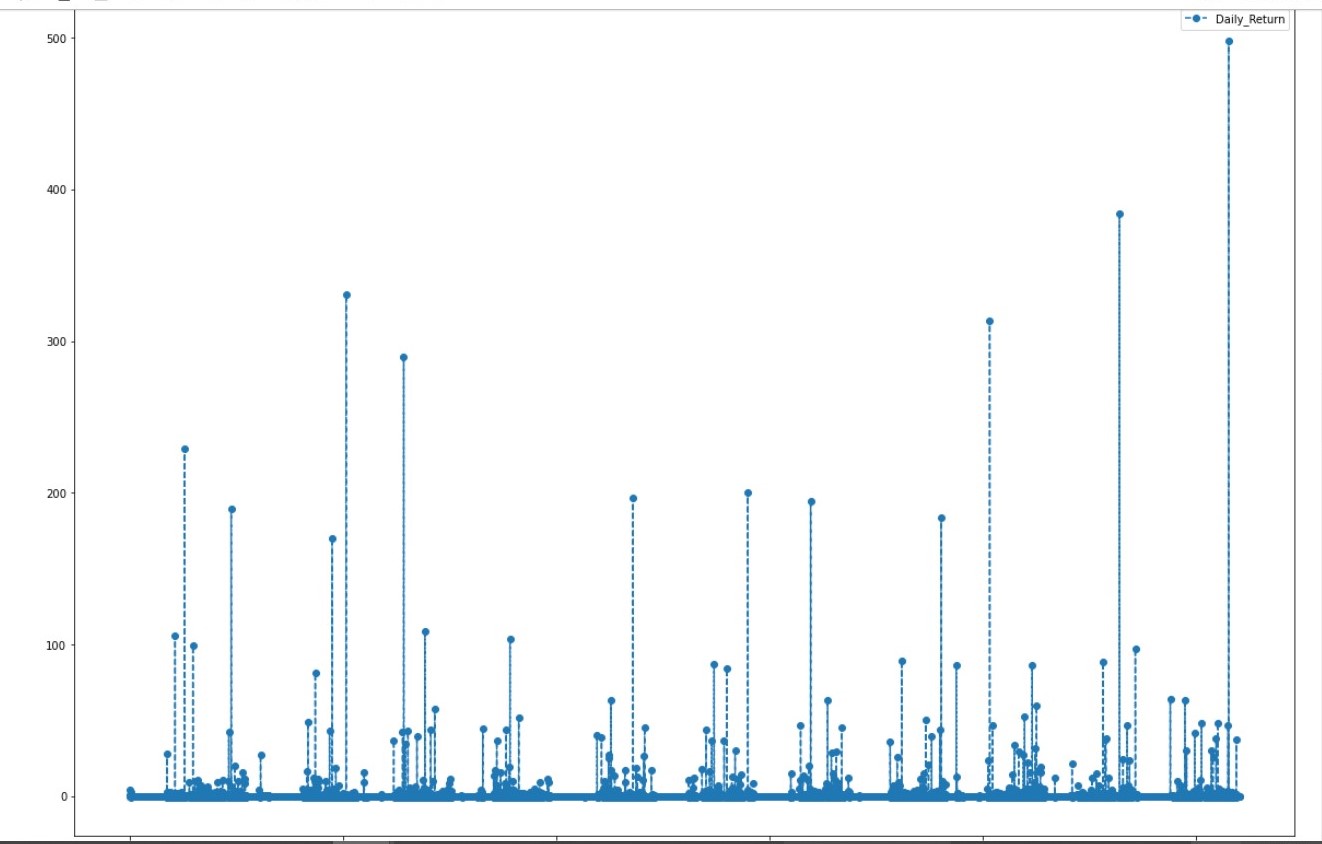
**Change in stock price:**

The change in stock price over time will help the investor to understand the fluctuations in stock price and based on the change over the years the investor might be able to infer whether the stock is a viable option or not. For example, the change in stock price was the most noticeable during Oct 2018, where the difference between the highest and lowest was the largest.



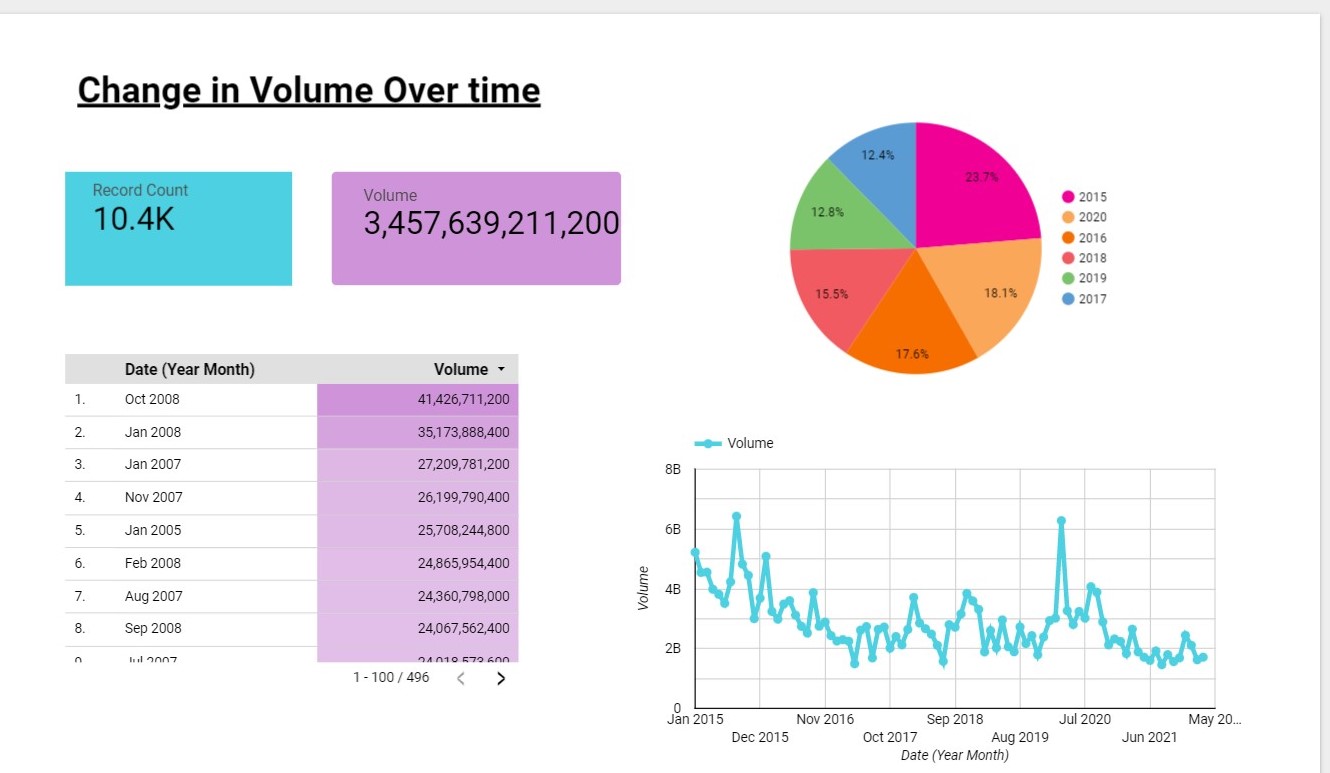
**Average daily return of stock:**

The average daily return will help to understand how the stock performs daily, and what all days of the week the stock closes with a good return and the investor can use this to make decisions about when to buy or sell the stock.



**Change in the volume traded over time**

The volume traded over time will help to understand when there was a lot of buying and selling of stocks (trading) and based on that the investor can decide when to buy and sell stocks. For example, the largest change in volume traded over time was in 2015 when the stock price was the highest, and then in 2020 during the covid time. So this helps us to understand how the change in volume is affected by factors such as price, or occurrence of an event like a pandemic.



**Forecast future stock performance?**

Using modeling and machine learning tools we were able to predict the highest price of the stock on a particular day, so the investor can use this information to decide when is the optimal time and date to buy the stock.

