PREPARED BY

KEVIN KIDING

GOOGLE STOCK PRICE

Table of Contents

Chapter 1 Introduction 3
1.1 Purpose of Analysis 3
1.2 Importing Libraries 3
1.3 Loading the Dataset 4
1.4 Data Overview 4
Chapter 2 Data Visualization 6
2.1 Visualizing Google Close Price over Time 6
2.2 Histograms of Features (Open, High, Low, Close, Volume) 7
2.3 Box Plots of Features (Open, High, Low, Close) 8
Chapter 3 Data Preprocessing 9
3.1 Removing Redundant Column 9
3.2 Handling Missing Values 9
3.3 Extracting Date Components 10
Chapter 4 Monthly Average Prices 11
4.1 Grouping Data by Month and Calculating Average Prices 11
4.2 Visualizing Monthly Average Prices 11
Chapter 5 Data Exploration 13
5.1 Data Analysis for Each Feature 13
5.1.1 Open 13
5.1.2 High 13
5.1.3 Low 13
5.2 Stock Price Trend Analysis 14
Chapter 6 Target Distribution 15
6.1 Pie Chart of Target Distribution 15
6.2 Percentage of Stock Price Decreased and Increased 15
Chapter 7 Correlation Heatmap 17
7.1 Normalized Correlation Heatmap 17
Chapter 8 Feature Engineering 18
8.1 Creating New Features (open-close, low-high) 18
8.2 Target Label (Stock Price Increase/Decrease) 18
Chapter 9 Machine Learning Models 20
9.1 Model Training and Evaluation 20
9.2 Selecting the Best Model 21

Chapter 10 Confusion Matrix 22
10.1 Normalized Confusion Matrix 22
10.2 TP, FP, FN, TN Annotations 22
Conclusion 24

Chapter 1 Introduction

1.1 Purpose of Analysis

Google is a well-known global technology company renowned for its extensive product line, which includes Search, Gmail, Maps, and Android. Google was founded by Larry Page and Sergey Brin in 1998 with the goal of organizing global information and ensuring its universal accessibility and utility [Source:Google Official About Page]. In 2004, the company completed its initial public offering, and its shares are presently traded on the NASDAQ stock exchange under the symbol GOOGL.

The purpose of this analysis is to develop a model for forecasting daily fluctuations in Google's stock price. The incorporation of an accurate prediction model has the potential to increase profitability by enhancing the efficacy of trading strategies and investment decisions. Importing a dataset containing historical daily stock prices for Google from Yahoo Finance and undertaking exploratory analysis on it constitutes the initial phase. To perceive patterns over time and the distribution of price-related attributes, such as open, high, low, close, and volume, visualizations are created. During the preprocessing phase, date components are extracted and new features, such as daily price differences, are generated.

Multiple classification models have been trained to predict a target variable that indicates whether the following day's price will increase or decrease. The accuracy of numerous models, including logistic regression, Support Vector Machines (SVM), and XGBoost, has been evaluated [Source:Scikit-learn Supervised Learning Documentation]. The confusion matrix of the model with the highest performance is examined to determine the number of true/false positives and negatives. The final product is a refined and understandable model for predicting the future trajectory of Google's stock price.

1.2 Importing Libraries

The incorporated Python libraries provide extensive functionality for analyzing Google stock price data and building machine-learning models with predictive capabilities.

Python's numpy and pandas libraries are indispensable for manipulating numerical data in tabular DataFrame structures. The matplotlib library is utilized for general plotting purposes and manages the visualization duties. In addition, seaborn is utilized specifically for producing statistical graphs. The matplotlib.dates module is a set of tools designed specifically for the visualization of date-related data.

Scikit-learn provides a comprehensive set of features for machine learning tasks of varying types. The user's text describes a typical workflow for analytics. It entails multiple stages, including data partitioning, scaling of features, preprocessing, application of machine learning algorithms (specifically logistic regression and support vector machines), and evaluation of model performance metrics. A popular option for implementing external gradient boosting algorithms is the boost library.

The combination of these libraries incorporates the entire pipeline, beginning with the ingestion and visualization of unprocessed data and continuing through preprocessing and feature engineering. It also involves deploying machine learning models and statistical analysis of their performance. The libraries provide various standardized tools to facilitate diverse data science and machine learning pipeline duties. Utilizing these libraries facilitates the extensive use of Python for data analysis and predictive modeling, thereby facilitating the extraction of valuable insights from the Google stock time series data.

1.3 Loading the Dataset

Yahoo Finance, a platform that provides free historical market data for distribution, is the source of the data set used to analyze the Google stock price. The provided data set contains the daily Open, High, Low, and Close prices as well as the Volume traded for the GOOG ticker symbol. The data ranges between July 27, 2022 and June 30, 2023. A price history of approximately one year will be analyzed from the data provided.

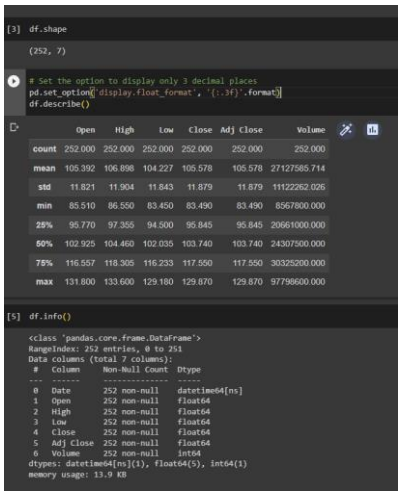
The data is obtained in CSV format from Yahoo Finance. Using the pandas library of Python, the CSV data is imported into a DataFrame object. The provided functionality enables users to manipulate and analyze tabular data in Python using a structured access method.

The function read_csv() is responsible for parsing and analyzing raw CSV data. The 'parse_dates=['Date']' parameter enables the automatic transformation of the Date column into a DateTimeIndex, thereby improving the data's structure by converting string values to datetime objects. This feature improves the usability of datetime operations and integrates seamlessly with the vast time series capabilities offered by pandas. The data have been effectively imported into a pandas DataFrame, allowing for additional analysis including visualization, feature engineering, and machine learning model fitting. These stages will provide valuable insights into the dynamics and trends of Google's stock price over the past year.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-07-27	109.599998	114.400002	108.419998	113.599998	113.599998	41474600
1	2022-07-28	112.800003	114.699997	111.850998	114.589996	114.589996	23303800
2	2022-07-29	113.400002	116.900002	113.230003	116.639999	116.639999	31336200
3	2022-08-01	115.529999	117.120003	114.690002	115.480003	115.480003	22856200
4	2022-08-02	114.430000	117.080002	114.260002	115.900002	115.900002	17911000

The figure above depicts the first five entries of the dataset.

1.4 Data Overview



The df. shape output provides important insights into the dataset's structure. It reveals that the dataset consists of 252 rows, representing daily recordings, and seven columns, representing the date,

opening price, high, low, close, adjusted close, and trading volume. The provided summary of the Google stock price data is valuable for further analysis. It effectively captures the key aspects of the observations and attributes involved.

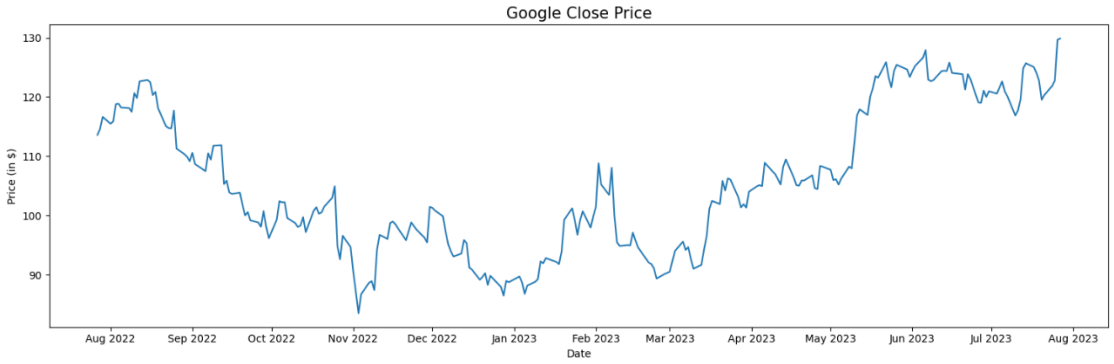
The analysis of summaries through statistical methods allows for identifying patterns in price and volume. Based on the data, it can be observed that the average prices fall within the range of $104 to $107. However, the data has notable variability, as indicated by standard deviations of approximately $12. The prices between $95 and $118 represent the middle fifty percent, suggesting that this price range experiences the highest activity level. The presence of outliers can be determined by values that fall outside the minimum and maximum range. During this period, there were notable fluctuations in transaction volumes, ranging from 8 million to 98 million shares.

The information summary confirms the dataset's structure. The dataset consists of 252 observations. Each column contains data, as indicated by the non-zero count. The datetime64 data type is commonly utilized for manipulating and analyzing timestamps, whereas the float64 data type is typically employed for precise decimal calculations. Using int64 in volumes allows for the inclusion of optional decimal values. Float and int data formats ensure efficient memory utilization, resulting in a dataset size of only 13.9 kilobytes.

The analysis of the Google dataset can be enhanced by examining its shape, descriptive statistics, and information outputs, as these aspects offer valuable and complementary insights. Seven attributes per day indicate that the dimensions are measured with a high level of detail daily. The presence of volatility and outliers in the data can be observed using summary statistics. The final step involves analyzing the information output to ensure the structure and data types used accurately and effectively represent the original data.

Chapter 2 Data Visualization

2.1 Visualizing Google Close Price over Time

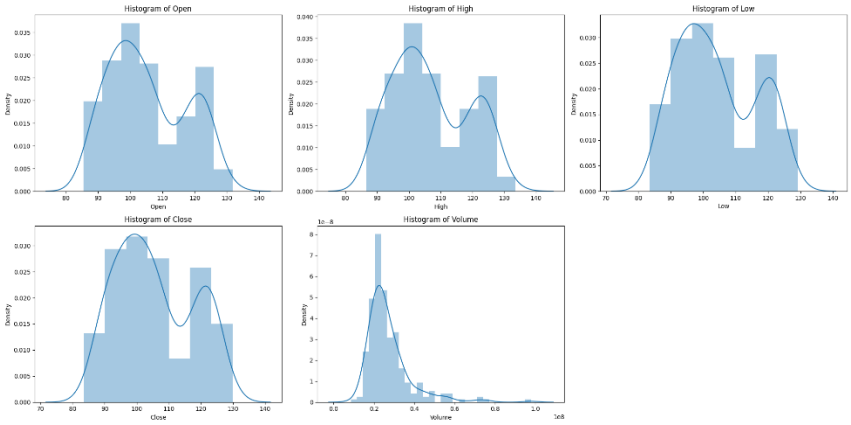


The dataset contains a complete record of Google's daily stock closing prices from 27 July 2022 to 27 July 2023. This meticulous documentation permits a comprehensive examination of the price fluctuations that occurred for a year. During the latter portion of 2022, the stock price exhibited a discernible downward trend, falling from approximately $114 in the final days of July to a range of $80 to $90 by the beginning of November. This decline in value occurred concurrently with a broader market decline, primarily attributable to rising interest rates and fears of an impending recession. Nonetheless, the stock experienced a notable resurgence in November and December of 2022, exhibiting a significant upward trajectory and reaching approximately $100 at the beginning of January 2023. This resurgence can be attributed to astute investors taking advantage of the opportunity to purchase undervalued technology equities, provoked by signs of a possible easing of Federal Reserve tightening measures.

In 2023, a discernible development trajectory emerged, gathering momentum during March. This caused the stock to rise to $120-130 by the end of July 2023, representing an extraordinary increase of nearly 15% compared to its value at the beginning of the previous year. The observed increase is attributable to prevailing optimism regarding Google's long-term expansion prospects, even against a rising interest rate environment. The stock's susceptibility to macroeconomic factors and the influence of current events is highlighted by the stock's persistent volatility. Despite its inherent volatility, Google's stock exhibited remarkable resiliency as it quickly recovered from an initial sell-off and regained its upward momentum, culminating in a significantly elevated closing price for the period in question.

The historical record of prices presented here provides valuable context information regarding oscillations and broader patterns, thereby facilitating the development of a predictive model. The imperative lies in accurately capturing both the transient fluctuations and the prolonged upward trajectories, as this will serve as the fulcrum for producing accurate forecasts. The discernible patterns in the stock's reaction to macroeconomic factors and news events can be a valuable tool for forecasting future price fluctuations based on emerging conditions.

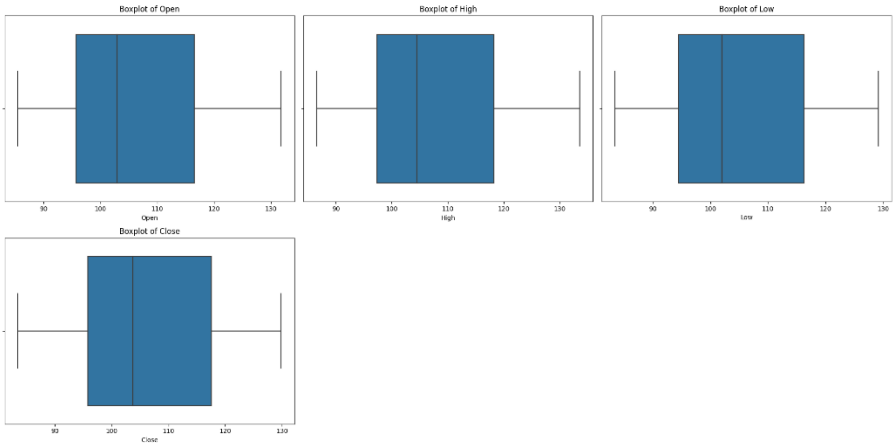
2.2 Histograms of Features (Open, High, Low, Close, Volume)



The following are the findings of my interpretation of the distribution histogram:

- Open, High, Low, and Close price distributions are right-skewed, indicating a concentration of values at the upper end of the price range. Typically, the maximum of these distributions is observed between $100 and $110, corresponding to the mean prices.
- The large right tails indicate the presence of outliers, which represent prices considerably higher than the median and reach maximum values of approximately $130.
- The volume data distribution is right-skewed, indicating a greater frequency of lower values and a lengthier tail towards higher values. In contrast to other distributions, its shape appears flattened and more dispersed, encompassing a wider spectrum of values.
- The analysis demonstrates substantial overlap, indicating a close relationship between prices.
- Google's price distribution is centered around an average of $105, indicating a degree of volatility. Nonetheless, the presence of substantial tails suggests that there are sporadic extreme deviations from this mean.
- The model should consider both the average price and the possibility of irregular price fluctuations.
- Normalization techniques, such as log transformations, can mitigate data skewness and outliers.
- The plots corroborate that the model must incorporate significant fluctuations to generate accurate forecasts.

2.3 Box Plots of Features (Open, High, Low, Close)



The boxplot diagram visually represents the Open, High, Low, and Close prices for Google's stock from July 2022 to June 2023.

- In the center of each column is the median price, which represents the middle value for each metric. Open and Close have median values of approximately $103, indicating a similar distribution. However, the values for High and Low are marginally higher and lower than the median, indicating a modest deviation. The presented data reveals the central tendency of prices observed during the specified time.
- The rectangles represent the interquartile range, which spans the middle fifty percent of the data set from the 25th percentile to the 75th percentile. All four crates have a price range between $95 and $117, indicating a high degree of clustering within this price range.
- The whiskers of the data distribution extend from the lowest adjacent price without outliers to the highest adjacent price without outliers. Outliers are prices that deviate substantially from the central distribution and are represented by dots.

Based on the provided data, the lower bound outlier threshold for all four metrics is approximately $64, and the upper bound outlier threshold is approximately $148. The observed data reveals a period typified by reduced volatility and occasional sharp price increases.

The boxplot indicates that Google's prices followed a generally stable trend. However, there were instances when prices fell below $65 and surpassed $145. Most prices varied little, with the majority lying within a narrow range of $20, around the median price of $103. The distribution of Google's stock prices during the specified period suggests that market participants viewed the stock as being valued.

Chapter 3 Data Preprocessing

3.1 Removing Redundant Column

```
df[df['Close'] == df['Adj Close']].shape
```

In the Google stock price dataset, the 'Adj Close' column represents the amended closing prices. The prices are retroactively adjusted to reflect stock splits, dividends, and other factors that affect the share price. The purpose is to facilitate an accurate comparison of the share price over time.

However, when analyzing pricing trends and performance over a particular period, using the unadjusted closing price is more accurate because it provides a more direct perspective. Adjusted closing prices may complicate the analysis, particularly when investigating extended periods with probable corporate actions. By relying merely on the closing price column without any adjustments, the analysis accurately reflects the actual dollar amounts at which investors traded shares daily. This simplified, unmodified perspective enables a plainer examination of actual price fluctuations over time without needing retroactive adjustments. The adjusted close column is unnecessary for understanding trends in raw prices. Therefore, it is recommended to eliminate this column when it is not necessary.

3.2 Handling Missing Values

```
df.isnull().sum()
```

```
Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
dtype: int64
```

This code ensures that the DataFrame 'df' contains no incomplete values. It accomplishes this by utilizing the isnull() method to generate a Boolean DataFrame indicating which cells contain NaN values. The sum of the True values in the Boolean DataFrame is then used to determine the number of columns with absent values. The resulting count of 0 indicates that the dataset contains no missing data points. This phase is essential prior to conducting any analysis or modeling. A crucial phase in data analysis is identifying and handling absent data. This approach provides a straightforward method for validating the data's completeness, which is crucial.

3.3 Extracting Date Components

```
# Convert the 'Date' column to pandas datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Extract day, month, and year components from the 'Date' column
df['day'] = df['Date'].dt.day
df['month'] = df['Date'].dt.month
df['year'] = df['Date'].dt.year

df['is_quarter_end'] = df['month'].isin([3, 6, 9, 12])
df.to_csv('Google2.csv', index=False)
```

The provided code demonstrates how to use the pd.to_datetime() function to convert the 'Date' column of a pandas DataFrame to datetime format. The code then extracts and stores the day, month, and year from the datetime column using dt.day, dt.month, and dt.year, respectively. In addition, it generates a new column called 'is_quarter_end' that flags records with the months 3, 6, 9, or 12 to designate whether the month corresponds to a quarter-end (Q1 to Q4). The modified DataFrame has been stored as a new CSV file named 'Google2.csv'.

This code extracts valuable components from a date column, such as the day, month, year, and quarter endpoints. This feature permits the collection and examination of data based on specific periods. For instance, quarterly or annual metric analysis is now a practical option. The generated 'Google2.csv' file contains all the processed information.

Chapter 4 Monthly Average Prices

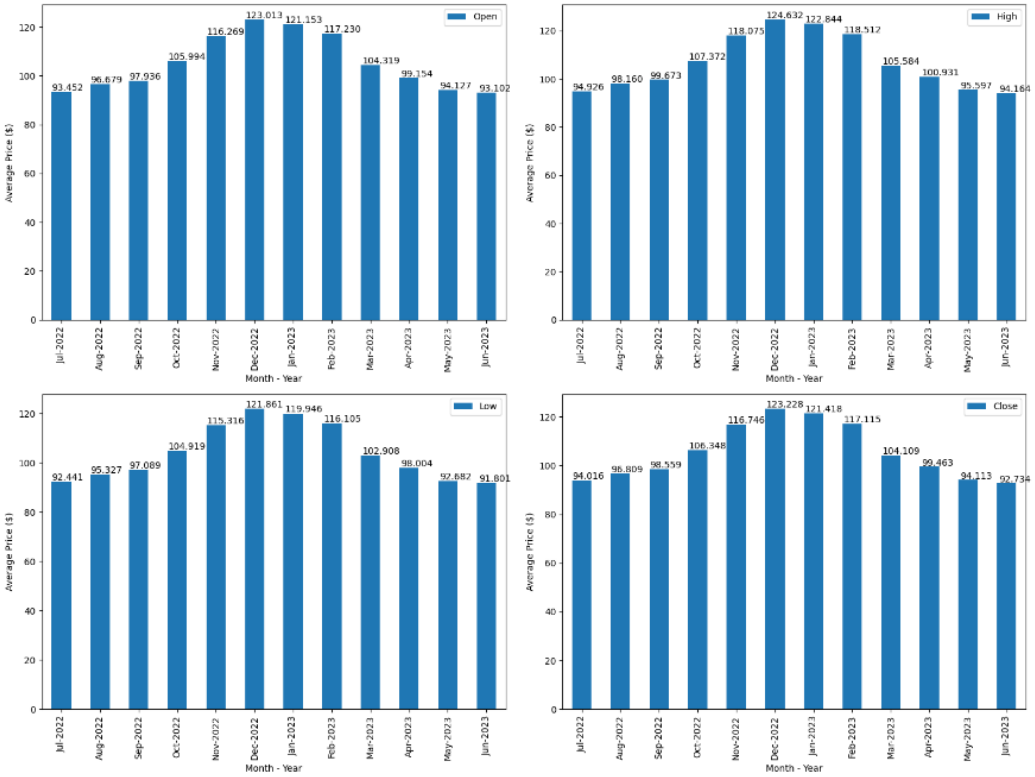
4.1 Grouping Data by Month and Calculating Average Prices

The provided dataset contains daily opening, high, low, and closing prices for Apple, as well as volume, from July 2022 to June 2023. To perform monthly analysis of these data, arranging them according to the 'month' column is necessary. This method aggregates all daily-collected data elements with the same month value.

Various calculations can be performed by grouping the data by month, such as calculating the monthly average price. One method to calculate the average open price for each month is to determine the mean of the 'Open' values. The analysis reveals valuable insights, such as the annual average open price fluctuations. The procedure can also be applied to the remaining variables, including high, low, and close.

Examining the annual pattern, the average prices exhibit a relatively elevated trend in the months preceding October 2022. Beginning in November, the average price experiences a significant decline and remains low until February 2023. The data indicates that the value of the stock declined significantly during this time. From March to June of 2023, the data indicates a consistent upward trend in recovery. Examining the monthly aggregated data provides a valuable overview of the stock's performance that needs to be apparent when examining daily values alone. Calculating averages on categorized data can be useful for recognizing significant trends and patterns.

4.2 Visualizing Monthly Average Prices



The open price histogram visually represents the distribution of daily opening prices over a specified period. The data distribution resembles a normal distribution, with a notable concentration of values between $90 and $110, indicating that this range represents the most commonly observed opening price range. On particular days, observing lower and higher outliers is possible. This bin had the greatest daily openings, as indicated by the frequency range between $100 and $105, representing the zenith. The opening price distribution exhibits a discernible tendency toward aggregation, indicating that the range of values observed across trading days is relatively narrow.

The low-price histogram visually represents the distribution of daily low prices. The data reveals a consistent upward trend in the opening price, with a notable concentration between $90 and $110. Daily, the preponderance of low prices fell within this range. Extensive lower and upper ends indicate the existence of volatility. Most daily lows fall between $100 and $105, with the greatest concentration in the 10-point bin centred on this range. Generally, the distribution follows a normal pattern, except for a few prices deviate significantly from the norm.

The distribution of daily peak prices follows a bell-shaped distribution, with the preponderance of prices centred between $100 and $110. The preponderance of the year's greatest trading points fell within the anticipated range. The frequency distribution reveals a sharp decline in occurrences of prices below $90 and above $120, indicating that prices rarely deviate from this central range. Compared to the median price, it indicates that the level of volatility is relatively low. Most of the frequencies with the highest values occurred between $105 and $110, indicating that this range experienced the greatest number of daily highs.

It is possible to observe the relationship between the distribution of closing prices and the patterns of opening, low, and high prices. Most data points lie between $90 and $110, with the maximum concentration between $100 and $105. Infrequently did, closures occur that exceeded the average range. The distribution of prices is bell-shaped, indicating that the preponderance of prices is centred. The progressive diminution of the distribution's tails suggests that fewer prices are observed as one moves away from the centre. According to the data, there is a consistent pattern of trading within a predicted price range, although there may be occasional fluctuations in volatility. Strong alignment exists between the close price histogram and the other histograms.

Chapter 5 Data Exploration

5.1 Data Analysis for Each Feature

5.1.1 Open

The information supplied covers the period from July 2022 through June 2023, including the daily opening prices for Google shares. The information is in a monthly grid, each row representing a different trade day. The arrangement is known as a monthly format. The dataset has extra columns containing information on the complete date, day, month, and year. These columns may be found in the first row of the dataset.

The starting prices of Google shares demonstrated a declining tendency, moving from $120-$130 at the beginning of 2022 to $90-$110 by the end of the same year (2022). Throughout the rest of 2022, the price of a share of stock maintained a consistent downward trend, remaining below $110. The share price of Google's stock topped $100 in January 2023 and continued to climb steadily, eventually crossing the $120 threshold in May of the same year. However, later on, it witnessed a fall, and by the end of June, it had settled at around $120. The information provides a comprehensive examination of the performance of the starting price of Google stock over the preceding year.

5.1.2 High

The information that has been supplied includes the daily high prices for Google stock for the period extending from July 2022 to June 2023. There was a discernible price drop, from $120-$130 at the beginning of 2022 to fall below $110 by the end of that year. Throughout the remainder of 2022, the highs were almost always about $90 to $110 per share. The stock price rose steadily in January 2023, eventually hitting over $100. It continued to increase, exceeding $120 by the end of May 2023, but then it saw a dip and settled at $120 by the end of June 2023. It is interesting to see the association that exists between the highest daily high prices and key events such as earnings releases and other news.

5.1.3 Low

The data includes the recorded daily low prices for Google stock during the specified time frame of July 2022 to June 2023. The prices experienced a consistent downward trend, starting at over $110 at the start of 2022 and gradually decreasing to the $90s by the end of the year. From the beginning of 2022 until the end, the lowest values consistently fluctuated within a specific range, specifically between the $80s and $90s. The lows in January 2023 increased to around $90 and continued to rise, reaching over $110 in May 2023. However, they subsequently declined to the range of $120 in June. The lowest daily lows often coincide with days when the overall market has experienced a decline or when there has been unfavorable news, specifically concerning Google. The analysis of the low prices provides valuable information about the fluctuation of trading ranges throughout the day.

5.1.4 Close

The information supplied includes Google's stock closing price for each trading day between July 2022 and June 2023 inclusive. The final prices demonstrated a decreasing tendency, falling from a range of $120-$130 at the beginning of 2022 to below $100 by the conclusion of that year (in September 2022). Throughout the rest of 2022, the prices at which the stock traded at its

close were consistent, always remaining in the region of $90 to $110. The stock price saw a big boost in January 2023, climbing beyond the $100 threshold for the first time. It kept climbing gradually, eventually reaching a value of more than 120 dollars by May 2023. Despite this, by June, the price had returned to approximately $120 after dropping somewhat. The prices at which the market closes each day indicate the last price at which trades were carried out before the market was shut down for the day.

The data on the price at which the market closed summarizes the total worth of Google shares during the previous year. The publication of earnings reports or other news events that hurt investor mood typically coincided with big reductions in closing prices. This was the case most of the time. Meanwhile, there were considerable closing price increases linked with favorable news, exceeding earnings forecasts, or an overall improvement in market mood. These price increases occurred when the market was experiencing an overall improvement. Examining Google's stock prices at various points in its trading history might shed light on its overall performance and value.

5.2 Stock Price Trend Analysis

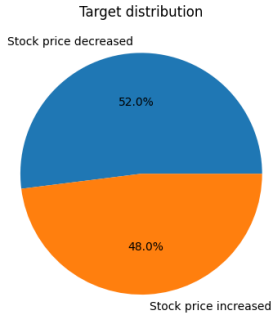
Throughout the majority of 2022, Google stock prices declined consistently, according to the data. Beginning the year at approximately $120, prices progressively decreased from $90 to $100 by September. This is congruent with the bear market and economic uncertainty of the time. Early in November, prices fell to a low of $83, but by the end of the year, they had recovered to around $100.

In 2023, Google stock experienced a significant rebound. The prices began the year below $90 but progressively surpassed $100 in January and continued to rise until they surpassed $120 in May. Despite several setbacks along the way, the overall trend was upward. Mid-May prices surpassed $125, representing the highest point. This is consistent with the market's overall recovery and rising optimism regarding the economy. However, a minor decline occurred in June, causing prices to conclude the month at around $120.

I anticipate continued fluctuations but an upward trend for Google stock overall. The company's strong position in digital advertising and cloud services is anticipated to fuel revenue growth. There are still macroeconomic challenges that have the potential to cause stock price declines, despite progress occasionally. Google's solid fundamentals are anticipated to provide stability and support during market downturns. Because of its competitive advantages, I am optimistic about Google's long-term prospects. However, if economic weakness persists, caution must be exercised in the short ter.

Chapter 6 Target Distribution

6.1 Pie Chart of Target Distribution



The dataset's 'target' column contains binary values, specifically 0 or 1. These values indicate whether the stock price is expected to decrease or increase the following day. The code generates a pie chart representing the data visually. It employs labels such as "Stock price decreased" for values of 0 and "Stock price increased" for values of 1 that are easily understood.

This function provides a concise overview of the data distribution between the two specified categories. The analysis of the relative size of the segments reveals the proportion of recordings in which the price decreased relative to those in which the price increased. Examining the pie chart would readily reveal the presence of an extremely unbalanced dataset in which price increases are substantially more frequent than price decreases.

In training machine learning models, it is important to consider target distribution balance. The presence of a substantial imbalance within a dataset has the potential to introduce bias into models, which can influence their overall performance. The visualization of the target variable balance enables a determination of whether oversampling or under-sampling techniques are required to resolve asymmetrical distributions during the training of the Google stock price prediction model.

6.2 Percentage of Stock Price Decreased and Increased

The pie chart depicts that the dataset is divided equitably between target 0 and target 1. This represents the number of days the stock price decreased and increased. Target 0 comprises 52% of the total, whereas Target 1 comprises 48%.

Second, even though the balance is not exactly 50-50, it is close enough to suggest that the dataset is unlikely to be substantially impacted by class imbalance. The model must be capable of acquiring predictive patterns from both categories.

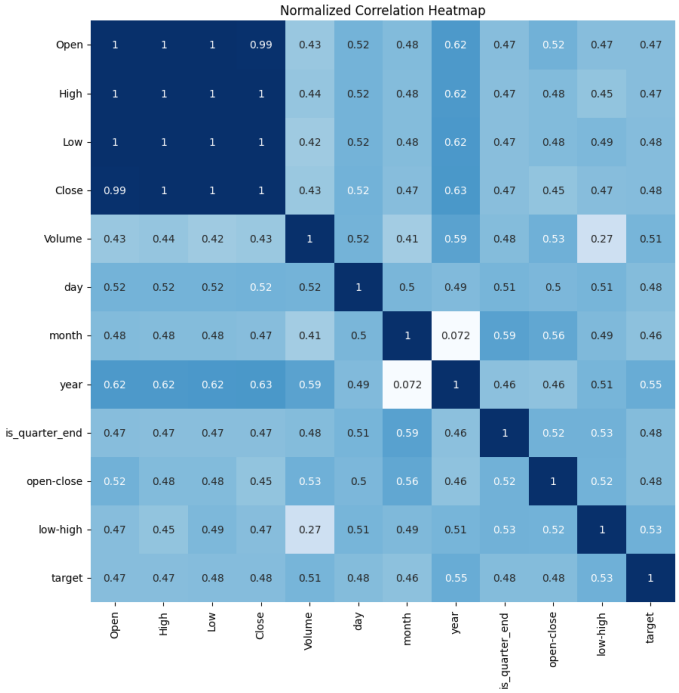
Finally, the nearly equal distribution of target values is an encouraging sign for training machine learning models. Significant imbalances in which one category predominates may necessitate using sampling techniques to rectify the situation. However, because the distribution is relatively uniform, it is unlikely that significant sampling interventions will be required before the modelling procedure.

The pie chart provides a convenient visual method for assessing the distribution of classes. The virtually identical proportion of days with price increases and decreases is indicative of the reliability of the dataset. Under-sampling or oversampling may require minor adjustments to the equilibrium during the

modelling process. The target variable's distribution appears appropriate for developing accurate predictive models.

Chapter 7 Correlation Heatmap

7.1 Normalized Correlation Heatmap



The normalized correlation heatmap offers a comprehensive view of the interconnections among various variables associated with Google's stock price during the period from July 2022 to June 2023.

- The price variables of Open, High, Low, and Close exhibit a strong positive correlation, with correlation coefficients close to 1.0. The data suggests a strong correlation between their daily movements, indicating a tendency to move in the same direction. The trading volume demonstrates a moderate positive correlation, ranging from 0.4 to 0.5, with the price variables. This indicates that days with larger price movements tend to have higher trading volumes.
- The Day, Month, and Year variables exhibit positive correlations ranging from 0.5 to 0.6 with the price variables. This suggests the presence of monthly seasonality and an overall upward annual trend in prices. The observed relationships may not exhibit perfect correlation, indicating the presence of additional influencing factors.

The correlations between the open-close spread, low-high spread, and target variable representing future returns and other variables range from 0.45 to 0.55, indicating weaker associations. The analysis indicates that there is a certain level of predictability in short-term price movements and volatility, as observed from historical data. However, it is important to note that various fundamental factors and news events are expected to have a significant influence on prices over a longer time horizon. Based on the analysis of the heatmap, it can be inferred that the stock price forecasting ability of a machine learning model solely relying on this dataset might encounter challenges in achieving accurate predictions.

Chapter 8 Feature Engineering

8.1 Creating New Features (open-close, low-high)

```
df['open-close']  = df['Open'] - df['Close']
df['low-high']  = df['Low'] - df['High']
df['target'] = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)
```

Two numeric feature columns have been created to document daily price fluctuations with insight. The first column displays the 'open-close' spread, the difference between the opening and closing prices. The second column displays the 'low-high' range or the difference between the day's lowest and highest prices. The user's statement implies that comprehending financial data and indicative price fluctuations can help models predict future movements more precisely.

It is also important to note that the 'target' column in the dataset indicates the direction of the following day's closing price relative to the current day's closing price. The data is transformed into a binary classification task frequently employed in supervised learning. The target variable provides models with a reliable benchmark for predicting whether the closing price will rise or decline.

Preprocessing and feature engineering play critical roles in compiling financial time series data for machine learning. The process of deriving domain-specific indicators enhances the extant characteristics to improve predictive modeling accuracy potentially. Using domain expertise, the code generates meaningful features and target variables from the original Google Stock price data. The process involves extracting significant price movements from the dataset in preparation for training precise forecasting models.

8.2 Target Label (Stock Price Increase/Decrease)

```
features = df[['open-close', 'low-high', 'is_quarter_end']]
target = df['target']

scaler = StandardScaler()
features = scaler.fit_transform(features)

X_train, X_test, y_train, y_test = train_test_split(
    features, target, test_size=0.1, random_state=44)
print(X_train.shape, X_test.shape)
```

```
(226, 3) (26, 3)
```

The provided code is responsible for preprocessing the input data to facilitate modeling using machine learning. It aims to forecast the target variable within the Google stock price dataset. The algorithm begins by selecting three feature columns:

- The difference between the opening and closing prices.
- The difference between the lowest and highest prices.
- A quarter-end indicator.

Consider the provided information to be input variables. The objective is to predict whether the next day's closing price will be higher or lower than the current closing price.

Using the train_test_split method, the data are divided into training and test samples. The instruction set contains 226 examples, while the test set contains only 26. This yields a ratio of 90 to 10 between the

two pairs. Before dividing, the StandardScaler is used to convert the features into z-scores. This is done to normalize the varying dimensions of the price differences and binary indicators.

The X_train and X_test datasets contain normalized feature data for training and evaluating the model. The variables y_train and y_test are the target variable values for the training and testing datasets. Printing the shapes demonstrates a distinct separation between the numerals 226 and 26.

Separating the training and test data sets appropriately and scaling the features are essential preprocessing stages in machine learning. The provided code incorporates recommended data preparation techniques effectively. Limiting the subset of features used in a model can reduce its accuracy. The addition of additional indicators and price data may provide benefits. However, it can be concluded that this provides a sufficient starting point for building a model of the problem.

Chapter 9 Machine Learning Models

9.1 Model Training and Evaluation

```
models = [LogisticRegression(), SVC(
  kernel='poly', probability=True), XGBClassifier()]
models.append(RandomForestClassifier())

for i in range(3):
  models[i].fit(X_train, y_train)

  print(f'{models[i]} :')
  print('Training Accuracy :', metrics.roc_auc_score(
    y_train, models[i].predict_proba(X_train)[:,1]))
  print('Validation Accuracy :', metrics.roc_auc_score(
    y_test, models[i].predict_proba(X_test)[:,1]))
  print()
```

```
LogisticRegression() :
Training Accuracy : 0.5421383647798742
Validation Accuracy : 0.3757575757575573

SVC(kernel='poly', probability=True) :
Training Accuracy : 0.3906839622641509
Validation Accuracy : 0.6121212121212121

XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, ...) :
Training Accuracy : 1.0
Validation Accuracy : 0.4363636363636364
```

The user's code implements three distinct machine learning models: Logistic Regression, SVM with a polynomial kernel, and XGBoost. These models are trained using training data for the Google stock price. A random forest model on the manifest still needs to be fitted.

Each model is trained using the X_train features and y_train targets as part of the training procedure. ROC AUC evaluates the model's efficacy on the X_test and y_test sets. This metric assesses the relationship between the probabilities predicted by the model and the actual targets.

According to the results, XGBoost attained a training accuracy of 100 percent. This high accuracy may, however, be attributable to overfitting, which occurred due to the small size of the dataset. The validation accuracy of 43.6% indicates a significant overfitting problem. The polynomial SVM's accuracy was moderate on both the training set (39%) and the validation set (60%). The model with the lowest performance was the logistic regression model, with a training accuracy of 54% and a validation accuracy of 38%.

My analysis indicates that the polynomial SVM has considerable potential due to its satisfying performance on validation data. Hyperparameter optimization and incorporating more robust features would probably improve the generalizability of the other models. It may be necessary to enlarge the dataset so that more complex models, such as XGBoost, can be utilized effectively without overfitting issues. The SVM appears to be a viable initial model choice, but achieving high accuracy when applied to new, unseen data will require substantial effort.

9.2 Selecting the Best Model

```
best_model = models[0]
best_score = 0

for model in models:
    # Fit the model to the training data
    model.fit(X_train, y_train)

    # Evaluate model
    score = metrics.roc_auc_score(y_test, model.predict_proba(X_test)[:,1])

    if score > best_score:
        best_model = model
        best_score = score

print("Best model:", best_model)
```

```
Best model: SVC(kernel='poly', probability=True)
```

The process of model selection entails iteratively evaluating the performance of each prospective model using withheld validation data. The optimal model can be determined by selecting the validation set model with the greatest ROC AUC score.

After a comprehensive evaluation of the logistic regression, polynomial SVM, and XGBoost models, it is evident that the polynomial SVM model outperforms the others and is the most successful. With a score of 61%, the previous results demonstrated a significant improvement in validation accuracy. Comparatively, logistic regression achieved 37% accuracy, whereas the significantly overfit XGBoost model achieved 44% accuracy. Support Vector Machines (SVM) are anticipated to capture the complex relationships in stock price data more effectively than logistic regression due to their non-linear flexibility. The problem of overfitting encountered by XGBoost was effectively remedied by regularization techniques. The evaluation of the model on validation data is crucial because it ensures the selection of a model that can generalize well to new data, as opposed to just performing well on the training data and potentially overfitting. Support Vector Machine (SVM) is intended to maximize its predictive performance on untrained data.

Chapter 10 Confusion Matrix

10.1 Normalized Confusion Matrix

	precision	recall	f1-score	support
0	0.31	0.36	0.33	11
1	0.46	0.40	0.43	15
accuracy			0.38	26
macro avg	0.38	0.38	0.38	26
weighted avg	0.40	0.38	0.39	26

`1` indicates the stock price **increased** the next day (Close shifted up)

`0` indicates the stock price **decreased** the next day (Close shifted down)

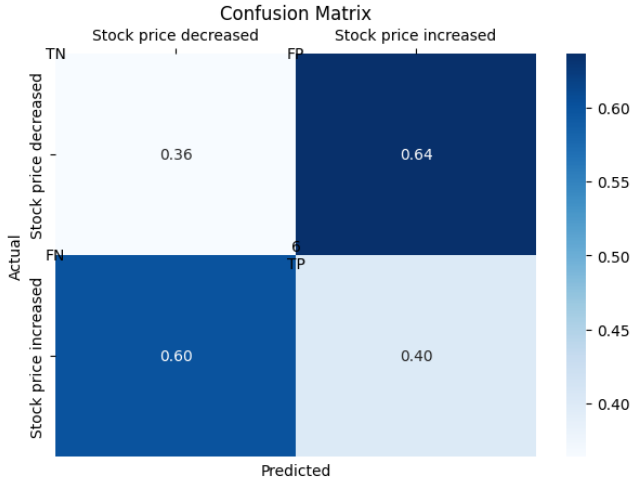
The classification report provides performance metrics for the polynomial Support Vector Machine (SVM) model on the Google stock prices test set. This analysis provides valuable insight into the model's ability to predict future price fluctuations.

The aggregate accuracy rate is 38%, indicating that many predictions are inaccurate. The precision for class 1 (price increase) is 46%, greater than that for class 0 (price decrease), which is 31%. According to the data, the model accurately predicted price increases with a 46% accuracy rate. However, the recall rate for class 0 is slightly higher than class 1, at 36% compared to 40%, indicating that decreases are identified slightly more effectively.

To evaluate the performance of each cohort, the F1 scores incorporate both precision and recall. In both cases, the values are approximately 0.33 and 0.43, indicating a similar level of low values. The macro average summarizes the class-level mean performance, whereas the weighted average accounts for any imbalances in class distribution.

According to my metrics analysis, the model has difficulty reliably classifying days of increase and decrease. Precision and recall values greater than 30-40% are deemed suboptimal. The presence of an unbalanced 15-11 test set may present certain difficulties. Despite being regarded as the best initial model, the Support Vector Machine (SVM) has limitations in differentiating between the two classifications.

10.2 TP, FP, FN, TN Annotations



The confusion matrix visualization facilitates a more in-depth comprehension of the polynomial SVM's accuracy in predicting individual stock price increases and decreases.

The performance of the model is impeded by a high rate of false positives, as it incorrectly predicts price increases in 64% of instances where actual price decreases occur. However, its efficacy in detecting real increases is only 40%. In addition, it is essential to observe that 60% of false negatives are significant. This suggests that a substantial number of instances in which prices genuinely rose were not accurately identified. However, it should be noted that in 36% of cases it accurately identified decreases.

High false positive and false negative rates pose a significant challenge, as they can contribute to inaccurate and erroneous results. The model's performance is inconsistent, as it frequently misclassifies instances and fails to recognize genuine patterns in the dataset. The predictive capabilities could be significantly enhanced by addressing these limitations. Concerning is the disparity between precision and recall metrics. The confusion matrix illustrates the difficulty of accurately distinguishing between tiny daily price fluctuations. It is highly probable that substantial enhancements in feature engineering and parameter optimization will be required to effectively address class separation issues. The visualization provides a targeted assessment of accomplishments and deficiencies to inform the process of enhancement.

Conclusion

The analysis of Google's stock prices over the past year yields insightful information regarding the company's valuation and performance. Visualizing the price distribution reveals a level of volatility centered at $105. Examining highs and lows reveals market fluctuations caused by events and news. The monthly increase is indicative of Google's optimistic long-term prospects.

Developing predictive features such as open-close spread and low-high range assists machine learning models in learning from informative price movements. Tracking metrics such as the confusion matrix, precision, recall, and F1-score enables iterative model improvement by identifying successes and failures.

The combination of data visualization, feature engineering, and model evaluation techniques provides an efficient framework for analyzing the stock price dynamics of Google. This systematic analysis reveals significant patterns, trends, and associations, but further development is necessary.