PROJECT REPORT:

STOCK PRICE PREDICTION AND CLASSIFICATION

Nicholas Catani

Table of Contents

Introduction

In a world where financial markets are integral to the global economy, understanding and predicting stock prices are of paramount importance. Stock prices influence investment decisions, impact company valuations, and serve as economic indicators for both businesses and individuals. In this context, the ability to analyze historical stock price data, extract insights, and build predictive models is a valuable skill for investors, financial analysts, and data scientists. This project delves into the intriguing world of stock price data analysis and modeling. It centers around a rich dataset containing daily stock price information for a vast array of companies. This dataset is a treasure trove of attributes, including the date of each data point, opening, and closing prices, highest and lowest prices during trading hours, trading volumes, dividend payments, and even details about stock splits. Moreover, the dataset provides critical information about the companies themselves, such as their brand names, ticker symbols, industry classifications, and the countries where they are headquartered or primarily operate.

The project's primary objectives are twofold: exploration and prediction. First and foremost, it aims to explore the dataset thoroughly, uncovering hidden patterns, relationships, and insights that can inform investment decisions or provide a broader perspective on the financial landscape. The exploratory phase involves data preprocessing, visualization, and a series of fundamental questions that shed light on the dataset's intricacies. Additionally, this project seeks to harness the power of machine learning to predict stock prices and classify the future performance of stocks. Through regression analysis, the project aims to forecast stock prices, enabling investors and analysts to make more informed decisions. Furthermore, it ventures into classification, attempting to determine whether stocks are likely to plummet to zero or remain stable. This predictive aspect can be invaluable for risk assessment and financial planning.

The project makes use of various Python libraries, including pandas for data manipulation, matplotlib and plotly for data visualization, scikit-learn for machine learning, and geopandas for geographical data representation. It applies data preprocessing techniques, conducts exploratory data analysis, and employes machine learning models for classification. In the section that follow, the project dives into the dataset, explores its characteristics, answers pertinent questions, visualize trends, and train ML models. It provides in-depth analysis and evaluation of each model's performance, shedding light on their strengths and limitations. The results provide a comprehensive understanding of the dataset, the predictive power of models, and the challenges that come with stock price prediction and classification.

Libraries

Pandas*: A data manipulation library used for data analysis and cleaning. It provides data structures and functions to make data analysis fast and straightforward.*

Matplotlib*: A data visualization library that provides a MATLAB-like interface for creating static, interactive, and animated visualizations in Python.*

Datetime*: A module that provides classes for manipulating dates and times in both simple and complex ways. It allows for easy handling of date and time data in the dataset.*

Pytz*: A python library that allows accurate and cross-platform timezone calculations, enabling the handling of timezone-related data withing the dataset.*

Geopandas*: An extension of the pandas library that enables the handling of geographic data and provides tools for exploring, analyzing, and visualizing geospatial data.*

Plotly*: An interactive visualization library that provides a range of plotting capabilities, including 2D and 3D plots, geographical maps, and interactive visualizations for easy data exploration.*

Sklearn*: A comprehensive machine learning library that provides simple and efficient tools for data mining and data analysis. It features various classification, regression, and clustering algorithms, as well as tools for model selection and evaluation.*

Each of these libraries plays a crucial role in different phases of the project, from data preprocessing and exploration to visualization and model training. Their integration provides a comprehensive framework for handling complex financial data and extracting meaningful insights.

Data Preparation and Exploration

The initial stages of the project involved comprehensive data preparation and exploratory data analysis (EDA) to ensure a robust foundation for subsequent analyses and modeling. This process comprised several keys steps, including data loading, data cleaning, feature engineering, and a comprehensive exploration of the dataset's characteristics.

Data Loading and Cleaning:

The first step involved loading the dataset using the pandas library's "read_csv" fuction. This function facilitated the seamless import of the data into a pandas DataFrame, providing a convenient data structure for subsequent analysis. Following data loading, a thorough data cleaning process was undertaken to handle missing values, outliers, and inconsistencies within the dataset. The "drop_duplicates" function was applied to remove any duplicate entries, ensuring the integrity and accuracy of the dataset.

Feature Engineering:

Feature engineering was a critical aspect of data preparation, involving the creation of new meaningful features derived from existing attributes. This included converting the "Date" column into a datetime format using the "pd.to_datetime" function, enabling effective time-based analyses. Additionally, the creation of a binary target variable "stock_going_down" was undertaken based on specific criteria related to the "Close" prices, facilitating subsequent classification modeling tasks.

EDA:

The exploratory data analysis phase aimed to uncover insights and patterns within the dataset, providing a comprehensive understanding of the data's characteristics and underlying trends. Various EDA techniques, including summary statistics, data visualization, and aggregation methods, were employed to gain valuable insights into the dataset.

Business Questions

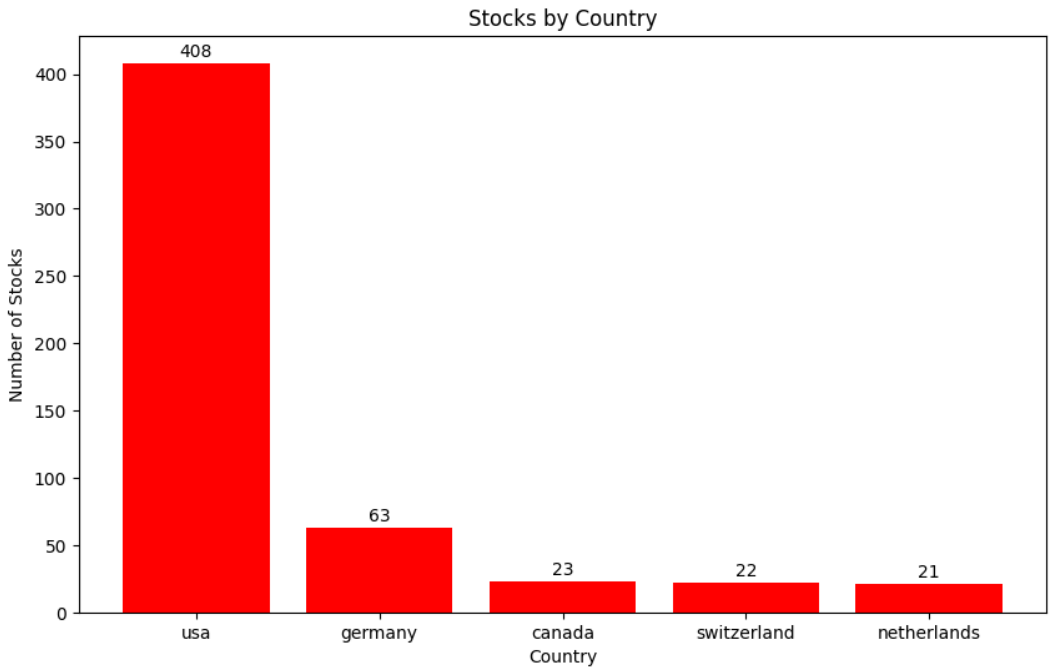
How many stocks have shown a closing price between $50 and $150 in the past month?

The first question tackled in this project focused on understanding the recent performance of stocks within a specific price range. In particular, the aim was to determine the number of stocks that exhibited closing prices falling within the range of $50 to $150 over the past month. To address this question, the project commenced by extracting and filtering the relevant data from the dataset. The "Date" column played a central role in this process, and its conversion into a datetime format enabled precise time-based filtering. The project's reference point for this analysis was the date "2023-09-20", and it aimed to assess the performance of stocks in the month leading up to this date. The filtering criteria were twofold: the closing prices had to be within the specified range of $50 to $150, and the data points had to belong to the month preceding the reference date. This was achieved by creating a temporal window that encompassed the 30 days prior to the reference date. Stocks whose closing prices fell within the designated price range during this time frame were considered for the final count.

The result of this analysis provided a quantitative answer to the question, indicating the number of stocks that met the criteria of closing prices within the price range and the specified time frame. Additionally, to enhance the comprehensibility of the results, a selection of relevant columns was presented alongside the stock count. This additional information allowed for a preliminary exploration of the characteristics of the stocks meeting the criteria.

The outcome of this analysis offered an initial glimpse into the recent performance of a subset of stocks, shedding light on their price dynamics in the past month. The results of this inquiry served as the starting point for further investigation into the dataset, paving the way for deeper insights into the financial behavior of these stocks.

```python
# Calculate the date one month ago from "2023-09-20"
target_date = datetime(2023, 9, 20, tzinfo=pytz.UTC)
one_month_ago = target_date - timedelta(days=30)

# Filter and count stock closing prices between $50 and $150 one month ago
filtered_data = df[(df["Date"] >= one_month_ago) & (df["Date"] < target_date) & (df["Close"] >=50) & (df["Close"] <= 150)]
count = len(filtered_data)
selected_columns = ["Brand_Name", "Close", "Volume", "Country"]
filtered_data = filtered_data[selected_columns]
print(f"Number of stock closing prices between $50 and $150 one month ago from {target_date}: {count}")
print(filtered_data)

# Group the stocks by country and count the # of stocks in each country
country_counts = filtered_data["Country"].value_counts()

# Create a bar chart
fig, ax = plt.subplots(figsize=(10, 6))
bars = ax.bar(country_counts.index, country_counts.values, color='red')
ax.set_xlabel('Country')
ax.set_ylabel('Number of Stocks')
ax.set_title('Stocks by Country')
for bar, count in zip(bars, country_counts.values):
    ax.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 5, str(count), ha="center")
plt.show()
```

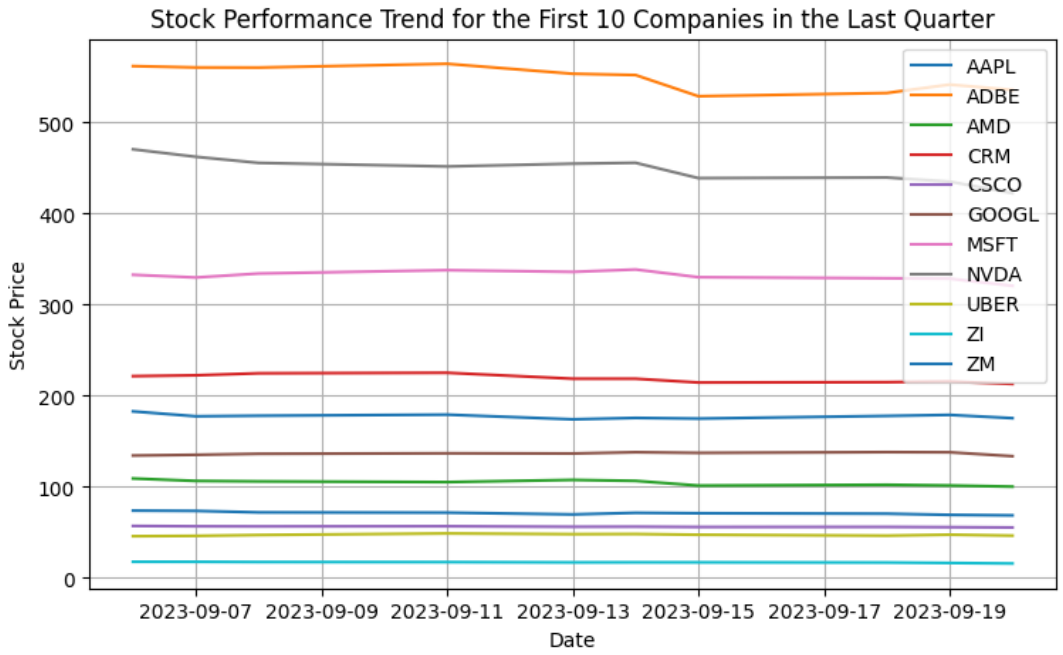


How did the Technology industry perform in the last quarter?

The second key question in this project revolved around a specific sector within the dataset, namely the technology industry. The primary objective was to assess the performance of technology companies, particularly those headquartered in the USA, over the last quarter leading up to the reference date of "2023-09-20". Addressing this question necessitated the identification and extraction of data specific to the technology industry. Furthermore, the focus was placed on companies headquartered in the United States. This dual selection criteria involved filtering the dataset to encompass these specific attributes. Understanding the performance of these technology companies within the "last quarter" required defining the temporal boundaries of interest. The "last quarter" was operationally defined as the three-month period preceding the reference date. Therefore, the analysis included data points falling within this timeframe. To assess the stock performance of these technology companies, the project concentrated on the "Close" prices. The difference between the initial and final "Close" prices within the last quarter was calculated for each company. This facilitated an understanding of the relative price change, indicating whether the stock's value increased or decreased during this period. Visual representations played a pivotal role in presenting the findings effectively. The "Close" price trends for each technology company were plotted over time, providing a visual narrative of their performance. These visualizations allowed for easy identification of companies with significant price fluctuations and those exhibiting more stable trends.

The outcomes of this analysis yielded insights into the recent performance of technology companies, focusing on their stock prices. It revealed which companies experienced substantial gains, losses, or maintained relatively stable prices within the last quarter. Additionally, the visualizations offered an accessible means to explore these trends and patterns, enabling the identification of potential outliers within the sector. The analysis of the technology industry's performance in the last quarter served as a basis of more in-depth investigations into the financial dynamics of these companies, providing valuable insights for investors and analysts in understanding the behavior of technology stocks in the lead-up to the reference date.

```python
# Filter for the first 10 tech companies headquartered in the USA
tech_usa_data = df[(df["Country"] == "usa") & (df["Industry_Tag"] == "technology")].groupby("Ticker").head(90)

# Sort the data by "Date" in ascending order
tech_usa_data = tech_usa_data.sort_values(by="Date", ascending=True)
end_date = datetime(2023, 9, 20, tzinfo=pytz.UTC)
start_date = end_date - timedelta(days=90)

# Filter the data for the last quarter
last_quarter_data = tech_usa_data[(tech_usa_data["Date"] >= start_date) & (tech_usa_data["Date"] <= end_date)]

# Calculate the stock performance for each company in the last quarter
stock_performance = last_quarter_data.groupby("Ticker").apply(lambda x: (x["Close"].iloc[0] - x["Close"].iloc[-1]) / x["Close"].iloc[0])
print(stock_performance)

# Plot the stock performance trend for each company in the last quarter
plt.figure(figsize=(12, 6))
for ticker, group in last_quarter_data.groupby("Ticker"):
    plt.plot(group["Date"], group["Close"], label=ticker)
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.title('Stock Performance Trend for the First 10 Companies in the Last Quarter')
plt.legend(loc='best')
plt.grid()
plt.show()
```

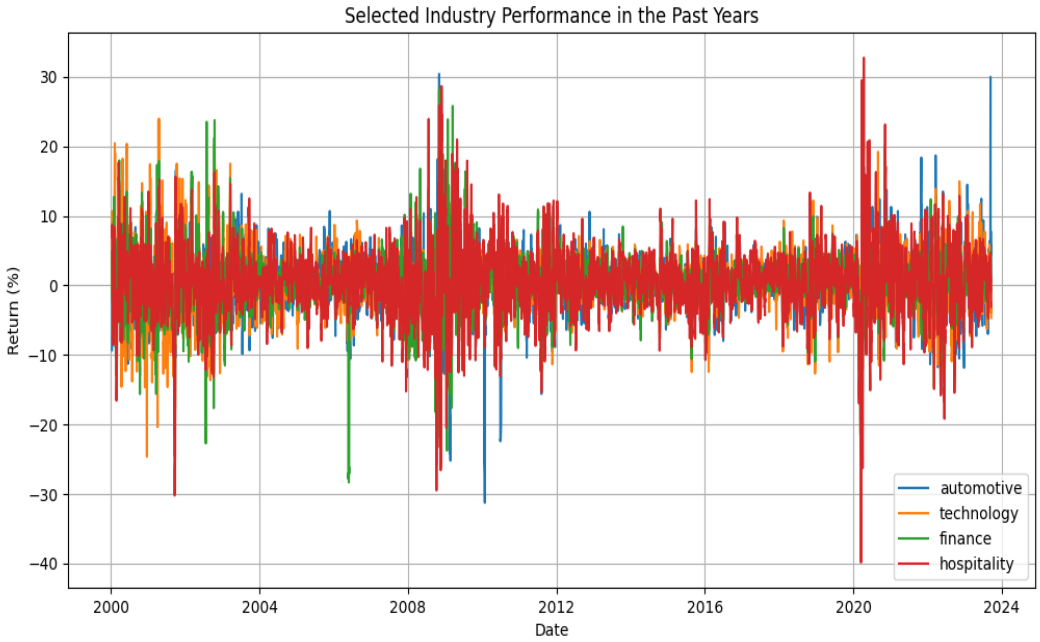


Which sector performed well in the last six months?

The third pivotal question in this project aimed to evaluate and compare the performance of different industry sectors within the dataset over a specific timeframe – the last six months leading up to the reference data of "2023-09-20". This analysis sought to identify sectors that exhibited notable returns during this period and those that may have faced challenges. The initial step in tackling this question involved preparing the data. The dataset was sorted by date, ensuring a chronological order that facilitated the extraction of relevant information for the designated timeframe. Additionally, a list of specific industry sectors of interest was identified for deeper exploration. To maintain data integrity, duplicate entries within the dataset were removed, thus ensuring that each observation represented a unique brand or company. This data cleaning process was integral to obtaining reliable insights during the subsequent analyses. The analysis focused on a select set of industries, including "automotive", "technology", "finance", and "hospitality". These industries were chosen for their diversity and representation within the dataset. To quantify the performance of these selected industries over the last six months, returns were computed. Specifically, the returns were calculated as the percentage change in the "Close" prices of stocks within each industry, relative to the closing price at the beginning of the period. This provided a measure of the relative performance of each industry sector over the designed timeframe. The analysis incorporated data visualization techniques to present the performance trends of the selected industries. Line charts were employed to visualize the returns of each sector, allowing for a comparative assessment of their trajectories. This visual representation highlighted the variations in performance across different sectors.

The results of this analysis facilitated the identification of sectors that experienced substantial returns and those that faced challenges over the past six months. The project offered insights into the relative performance of these sectors, shedding light on their potential strengths and weaknesses in the lead-up to the reference date.

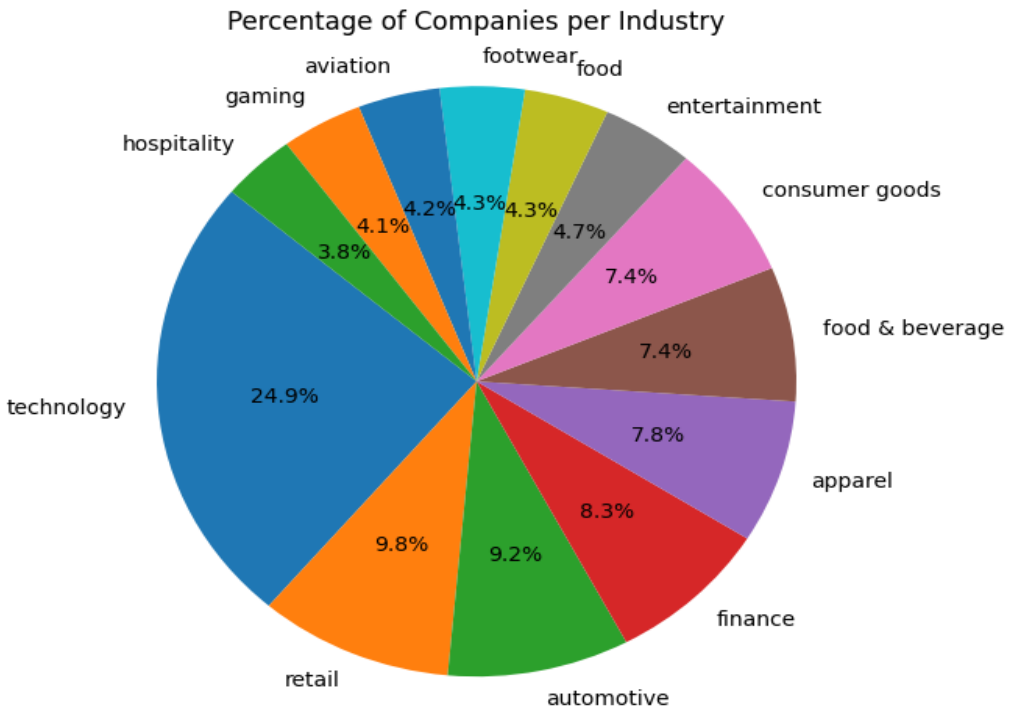
```python
# Ensure the data is sorted by 'Date'
data = df.sort_values(by=['Date', "Industry_Tag"])

# List of the specific industries you want to extract
selected_industries = ["automotive", "technology", "finance", "hospitality"]

# Remove duplicates by aggregating the data within each group
data = data.groupby(['Date', 'Industry_Tag']).agg({'Close': 'mean'}).reset_index()

# Filter the DataFrame to include only the selected industries
filtered_data = data[data['Industry_Tag'].isin(selected_industries)]

# Calculate the six-month return for each selected industry
filtered_data['Return'] = filtered_data.groupby('Industry_Tag')['Close'].pct_change(periods=6) * 100

# Pivot the data for plotting
pivoted_data = filtered_data.pivot(index='Date', columns='Industry_Tag', values='Return')

# Plot the performance of each selected industry with six-month returns
plt.figure(figsize=(12, 6))
for industry in selected_industries:
    plt.plot(pivoted_data.index, pivoted_data[industry], label=industry)
plt.xlabel('Date')
plt.ylabel('Return (%)')
plt.title('Selected Industry Performance in the Past Years')
plt.legend(loc='best')
plt.grid()
plt.show()
```



What is the Percentage of Companies per Industry in the dataset?

The fourth crucial question in this project aimed to provide a comprehensive understanding of the distribution of companies across different industries within the dataset. The analysis sought to answer the fundamental query: "What proportion of companies belong to each industry category, and are there any notable trends or imbalances in industry representation? Addressing this question required the initial preparation of the dataset to ensure data accuracy and reliability. The dataset was cleansed to remove any duplicate entries, ensuring that each observation represented a unique brand or company. This was an essential step to avoid any bias in the calculation of industry percentages. To derive the percentage of companies per industry, the data was aggregated, and the number of unique brands within each industry category was counted. This counting process provided insights into the distribution of companies across different sectors. The relative proportion of each industry was calculated with respect to the total number of unique companies in the dataset. The analysis placed a specific focus on industries that represented a substantial share of the dataset. Industries with percentages of companies less than 3% were filtered out to highlight those of more significant relevance and size. To present the findings effectively, a pie chart was created to visually represent the distribution of companies across the selected industries. The pie chart showcased the proportions of companies belonging to each sector, offering a clear and intuitive depiction of industry distribution.

```
######## Percentage of Companies per Industry

# Remove duplicates by aggregating the data
data = df.drop_duplicates()

# Group the data by 'Industry_Tag' and count the number of companies in each group
industry_counts = data['Industry_Tag'].value_counts()

# Calculate the total number of companies
total_companies = len(data)

# Filter industries with a percentage of companies greater than or equal to 3%
filtered_industries = industry_counts[industry_counts / total_companies >= 0.03]

# Create a pie chart with the filtered industries
plt.figure(figsize=(8, 8))
plt.pie(filtered_industries, labels=filtered_industries.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a circle.

plt.title('Percentage of Companies per Industry\n')
plt.show()
```

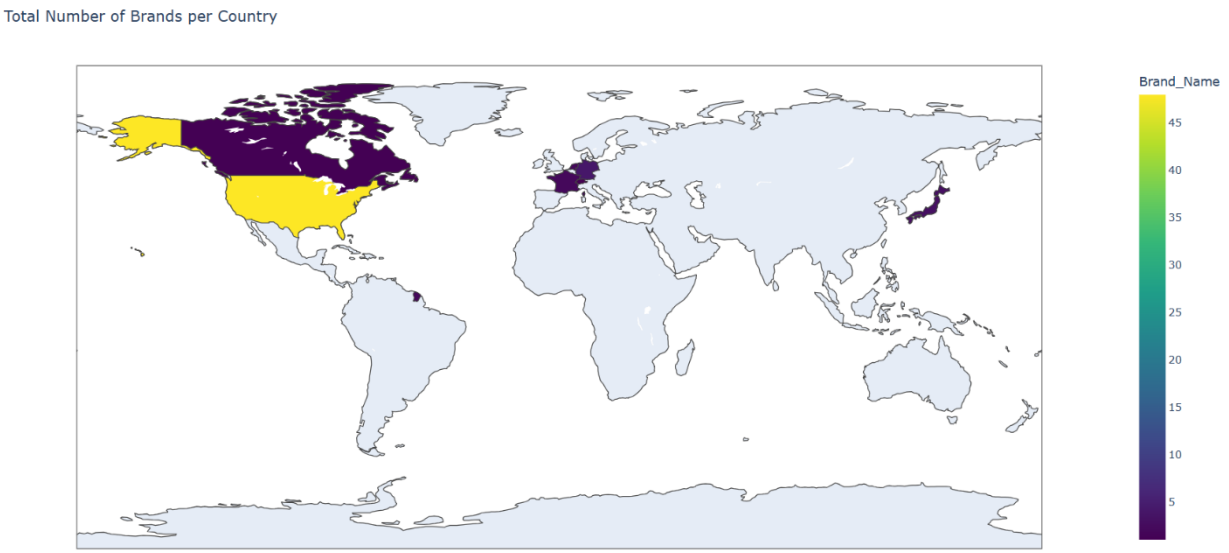


What is the total number of Companies per country in the dataset?

The fifth pivotal question in this project sought to delve into the geographic distribution of companies within the dataset. Specifically, it aimed to answer the question: "How many companies are headquartered or primarily operate in each country, and are there any significant trends or imbalances in the geographic representation of these companies? Addressing this question commenced with the preliminary preparation of the dataset to ensure data integrity and reliability. A fundamental aspect of this data preparation was the removal of duplicate entries, ensuring that each observation accurately represented a unique brand or company. Eliminating duplications was pivotal in the accurate calculation of the number of companies per country. To derive the total number of companies per country, the dataset was aggregated and grouped by the "Country" attribute. Within each group, the number of unique brands was counted, providing valuable insights into the geographic distribution of companies. This counting process was instrumental in understanding the extent of representation for each country. The visualization of this analysis was executed through the creation of cartograms. A cartogram is a geographical map that distorts the size of regions based on a specific variable – in this case, the number of companies. By merging the calculated company counts with a world map dataset, cartograms were generated, offering a visual representation of the concentration of companies in various countries. The colors on the cartograms emphasized the differences in the number of companies, enabling clear insights into the distribution.

The results of this analysis provided a comprehensive view of the geographic representation of companies within the dataset. The cartograms visually highlighted the varying degrees of concentration of companies in different countries, offering insights into global business operations. This information was valuable for investors, analysts, and stakeholders interested in understanding the regional dynamics of the companies in the dataset.

```python
# Remove duplicates by aggregating the data to count the number of unique Brand_Names per Country
country_brand_counts = df.drop_duplicates(subset=['Country', 'Brand_Name']).groupby('Country')['Brand_Name'].count().reset_index()
country_brand_counts.columns = ['Country', 'Brand_Name Count']

# Display the result
print(country_brand_counts)

# Load a world map dataset from geopandas
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Merge the world map dataset with the company counts
world = world.merge(country_brand_counts, left_on='iso_a3', right_on='Country', how='left')

# Plot the cartogram
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
world.boundary.plot(ax=ax, linewidth=1)
world.plot(column='Brand_Name Count', ax=ax, cmap="viridis", legend=True, legend_kwds={'label': "Number of Brands per Country"})
plt.title('Cartogram of Number of Brands per Country')
plt.show()
```



TRAINING SESSION

In this section of the project, I ventured into the realm of machine learning to develop predictive models that could offer valuable insights into the stock market. The overarching goal was to employ a diverse set of machine learning techniques to analyze and make predictions based on the provided dataset. The journey began by identifying the key data attributes for modeling. I focused on selecting features that could potentially influence stock prices, including attributes such as "High", "Low", "Volume", "Dividends", "Stock Splits", "Industry_Tag", and more. The target variable varied across different training scenarios and included "Close" stock prices and a binary variable, "stock_going_down", which indicated whether a stock's price would decrease to zero. To ensure the data was ready for machine learning, preprocessing was a fundamental step. For categorical variables like "Industry_Tag", we utilized one-hot encoding to convert them into numerical format, allowing machine learning models to work with them effectively.

- ***Linear Regression:*** *This classic regression model aimed to predict stock prices "Close" using various features. Cross-validation was executed to evaluate the model's performance. The resulting Mean Absolute Error, Mean Squared Error, and R-squared were analyzed.*
- ***Random Forest:*** *The random forest model, known for its ensemble of decision trees, was introduced to predict stock prices based on features like "Stock Splits", "Dividends", and "Volume". Once again, cross-validation was performed to access the model's performance, and metrics like MAE, MSE, and R-squared were used for evaluation.*
- ***Logistic Regression:*** *This model was introduced to predict whether a stock's price would decrease to zero. To achieve the goal, I trained the model with features like "High", "Low", "Volume", and "Dividends". The model's performance was critically evaluated using metrics such as accuracy, a confusion matrix, and a classification report.*

The interpretation of the results was an integral part of the training session. It provided a deeper understanding of each model's capabilities and limitations:

- ***Linear Regression***

The results indicated that the model, as configured, had limited predictive power for stock prices. The model displayed a relatively high MAE and MSE, with a low R-squared value, implying that it struggled to make accurate predictions. Further exploration, feature engineering, or more sophisticated algorithms might be necessary to enhance its predictive performance.

- ***Random Forest***

The random forest model, though renowned for its versatility, did not perform well for stock price prediction in this context. The model exhibited a negative R-squared value, suggesting a poor fit for the data, while MAE and MSE remained relatively high. This outcome implied that alternative modeling approaches or further feature engineering may be required to achieve better accuracy.

- ***Logistic Regression***

The model excelled in predicting stocks that did not go down to zero, achieving a high accuracy rate. However, it struggled to identify stocks that did decline. The imbalance in class distribution influenced the model's performance, underscoring the need for addressing class imbalance through techniques like oversampling, under sampling, or other types of algorithms to enhance the performance.

The training session underscored the multifaced nature of machine learning in the context of stock market analysis. It demonstrated that model performance can be influenced by various factors, including feature selection, data preprocessing, and class distribution. The results served as a foundation for further refinements and explorations, guiding stakeholders in making informed decisions regarding stock market investments and predictions.