team-9-flight-traffic-report

July 30, 2023

1 Non-Technical Executive Summary

1.1 Introduction

In the modern day, the United States boasts a host of commercial airline companies, making it possible to travel conveniently and efficiently throughout the country, and by extension, the world. With the rise of flying as a means of transportation, Americans can easily attend major events across the continent. Such surges in activity could affect the traffic at the location of the event in question, potentially causing delays for flights.

On a slightly related note, when considering the financial market, one notes that companies' performances are often tied to its stock price, and commercial airlines are no different. In the case of airlines, flight delays directly affect consumers, leading one to wonder how they might disturb these stock prices.

In our project, we sought to model the effect major public events have on flight delays and in turn, on airline stock prices. To do this, we identified major public events and analyzed their impact on flight delay patterns, then studied any subsequent changes in the respective airlines' stock prices.

The datasets we used included the ones on airports, airlines, major events, flight traffic, stock prices, and weather.

1.2 Key findings

1.2.1 Correlations between delays

The heatmap comparing the correlations of different types of delays shows that each of the 5 types of delays had a negative correlation with the other 4 types of delays, meaning higher values of one delay are associated with lower values of other delays. The strongest correlations can be analyzed by comparing the absolute values of the correlation coefficients, providing the following correlations between delays in order of decreasing absolute strength:

Aircraft and air system delay: r = -0.14

Airline and air system delay: r = -0.097

Aircraft and airline delay: r = -0.076

Aircraft and weather delay: r = -0.024

Airline and weather delay: -0.032

Air system and weather delay: 0.013

Air system and security delay: 0.0089

Airline and security delay: -0.0086

Aircraft and security delay: -0.0081

Weather and security delay: -0.0032

The main takeaway from this correlation analysis is that there are negative relationships between all types of delays. It suggests that when one type of delay increases, other types of delays tend to decrease. However, the correlations are generally weak, indicating that the relationships between these delay types are not very strong.

Please keep in mind that correlation does not imply causation, and further analysis and domain knowledge are required to understand the underlying reasons behind these correlations.

1.2.2 Correlation between the number of events occurring in the US on a particular day and the stock prices of different US airlines on that same day

American Airlines (AA): Correlation coefficient = -0.09796177540098208. This value is close to zero, indicating a weak or negligible negative correlation between the number of events and AA's stock price.

United Airlines (UA): Correlation coefficient = 0.10116632654785204. Again, this value is close to zero, indicating a weak or negligible positive correlation between the number of events and UA's stock price.

JetBlue Airways (B6): Correlation coefficient = 0.07165759733887636. Similarly, this value is close to zero, indicating a weak or negligible positive correlation between the number of events and B6's stock price.

SkyWest Airlines (OO): Correlation coefficient = -0.20610406382362. Here, we see a slightly stronger negative correlation, suggesting that a higher event count is weakly associated with a decrease in OO's stock price.

Alaska Airlines (AS): Correlation coefficient = 0.2692075272286969. AS shows a moderate positive correlation, meaning that a higher event count is somewhat associated with an increase in AS's stock price.

Spirit Airlines (NK): Correlation coefficient = 0.0190081591665456. NK's correlation is extremely close to zero, indicating a very weak or negligible positive correlation.

Southwest Airlines (WN): Correlation coefficient = 0.11406459573328544. WN also exhibits a weak positive correlation, suggesting that a higher event count is weakly associated with a slight increase in WN's stock price.

Delta Air Lines (DL): Correlation coefficient = -0.08704720859110898. DL has a correlation close to zero, indicating a weak or negligible negative correlation between the number of events and DL's stock price.

Hawaiian Airlines (HA): Correlation coefficient = 0.08467351442461774. HA shows a weak positive correlation, meaning that a higher event count is weakly associated with a slight increase in HA's stock price.

The data suggests that there are weak to negligible correlations between the number of events occurring in the US and the stock prices of most airlines on a given day. Only Alaska Airlines shows a moderate positive correlation, and SkyWest Airlines shows a slightly stronger negative correlation. However, it's important to note that correlation does not necessarily imply causation, and other factors could be influencing the stock prices of these airlines.

1.2.3 Trend of stock prices over time for each airline

Here are some basic descriptive statistics for each airline's stock prices:

Airline	Mean	Standard Deviation
American Airlines (AA)	50.33	2.76
United Airlines (UA)	65.56	4.14
JetBlue Airways (B6)	20.29	1.20
SkyWest Airlines (OO)	46.82	3.48
Alaska Airlines (AS)	72.64	7.03
Spirit Airlines (NK)	34.72	2.28
Southwest Airlines (WN)	54.94	2.49
Delta Airlines (DL)	50.99	2.01
Hawaiian Airlines (HA)	38.85	2.35

Here is a correlation analysis, where we calculate the correlation matrix to see how the stock prices of different airlines are related to each other:

Airline	AA	UA	B6	OO	AS	NK	WN	DL	HA
AA	1.00	0.91	0.78	0.95	0.50	0.92	0.96	0.89	0.64
UA	0.91	1.00	0.84	0.97	0.50	0.86	0.92	0.95	0.66
B6	0.78	0.84	1.00	0.90	0.46	0.80	0.80	0.86	0.61
OO	0.95	0.97	0.90	1.00	0.44	0.87	0.95	0.94	0.67
AS	0.50	0.50	0.46	0.44	1.00	0.33	0.47	0.48	0.32
NK	0.92	0.86	0.80	0.87	0.33	1.00	0.91	0.86	0.61
WN	0.96	0.92	0.80	0.95	0.47	0.91	1.00	0.92	0.63
DL	0.89	0.95	0.86	0.94	0.48	0.86	0.92	1.00	0.67
HA	0.64	0.66	0.61	0.67	0.32	0.61	0.63	0.67	1.00

1.3 Significance

1.3.1 Correlations between delays

Interpretation of correlation values: The negative correlations between most types of delays might indicate that airlines or airport authorities are managing these different types of delays to some extent, attempting to offset the impact of one delay by minimizing others.

Impact on airports and airlines: Considering there is a strong negative correlation between air system delays and aircraft delays, airlines might focus on improving air system infrastructure to indirectly reduce aircraft-related delays.

Weather delays: The weak negative correlations between weather-related delays and other types of delays suggest that weather disruptions might not significantly influence other delay factors. However, severe weather events could still pose substantial risks, and airlines and airports may want to develop contingency plans to deal with weather-related disruptions.

Operational efficiency: By analyzing these correlations over time, airlines and airports can track improvements in their operational efficiency: reductions in correlations between certain delay types may indicate successful efforts to manage and balance operations more effectively.

1.3.2 Correlation between the number of events occurring in the US on a particular day and the stock prices of different US airlines on that same day

American Airlines (AA): Weak Negative Correlation (-0.0979). American Airlines is one of the largest legacy carriers with significant international exposure. Factors such as geopolitical events, fuel prices, and economic conditions could impact both the number of events in the US and AA's stock price. Additionally, intense competition in the airline industry may also play a role in the weak negative correlation.

United Airlines (UA): Weak Positive Correlation (0.1012). United Airlines, being another major legacy carrier, shares similar factors with AA. Economic trends, fuel costs, and international operations may contribute to the weak positive correlation. Additionally, UA's hub locations and route network could also play a role in influencing the stock price and event count.

JetBlue Airways (B6): Weak Positive Correlation (0.0717). JetBlue is a low-cost carrier with a strong focus on leisure travel. As such, factors such as passenger demand, seasonal variations, and competition with other low-cost carriers might impact both the number of events and B6's stock price, contributing to the weak positive correlation.

SkyWest Airlines (OO): Strong Negative Correlation (-0.2061). SkyWest operates as a regional airline, and its performance might be more closely tied to domestic regional travel patterns and contracts with major airlines. Economic downturns or reductions in regional flight demand could lead to an increase in events and a decrease in OO's stock price, explaining the strong negative correlation.

Alaska Airlines (AS): Moderate Positive Correlation (0.2692). Alaska Airlines has a strong presence on the West Coast, which includes hub cities vulnerable to certain events. Positive economic conditions, regional travel demand, and its network presence might contribute to the moderate positive correlation between events and AS's stock price.

Spirit Airlines (NK): Weak Positive Correlation (0.0190). Spirit Airlines' focus on budget travel and a la carte pricing could make its stock price less sensitive to event counts. However, competition in the low-cost carrier market and passenger demand might still have a minor impact on both the number of events and NK's stock price.

Southwest Airlines (WN): Weak Positive Correlation (0.1141). Southwest's unique point-to-point network and focus on domestic travel might result in a weaker correlation between events and stock price. Nonetheless, economic conditions and domestic travel demand could still influence both factors, contributing to the weak positive correlation.

Delta Airlines (DL): Weak Negative Correlation (-0.0870). Delta's significant international exposure, hub locations, and operational scale make it sensitive to geopolitical events, fuel prices, and economic trends. These factors could lead to a weak negative correlation between events and DL's stock price.

Hawaiian Airlines (HA): Weak Positive Correlation (0.0847). Hawaiian Airlines' focus on leisure and vacation travel to Hawaii and other Pacific destinations might lead to a weaker correlation between events and its stock price. Seasonal variations, economic conditions, and demand for leisure travel could play a role in the correlation.

1.3.3 Trend of stock prices over time for each airline

Descriptive statistics findings **Mean Performance:** Alaska Airlines has the highest mean performance (mean = 72.64). This suggests that, on average, Alaska Airlines performs the best among all the airlines in the measured metrics. JetBlue has the lowest mean performance (mean = 20.29). This indicates that, on average, JetBlue performs the poorest among all the airlines in the measured metrics; at the time, it might have been facing stiff competition from both legacy carriers and low-cost carriers.

Performance Variability: Alaska Airlines also exhibits the highest standard deviation (StDev = 7.03). This means that its performance metrics have relatively high variability or fluctuation, indicating possible sensitivity to market changes or other factors. Hawaiian Airlines has the lowest standard deviation (StDev = 2.35), indicating more consistent and stable performance in comparison to other airlines.

Correlation matrix findings **Strong Positive Correlations:** American Airlines and United Airlines have a strong positive correlation of approximately 0.91. This suggests that these two airlines tend to have similar performance patterns in the measured metrics. They might face similar market conditions or be affected by similar external factors.

Strong Negative Correlation: Alaska Airlines and Spirit Airlines have a strong negative correlation of approximately -0.33. This indicates that these two airlines have different performance patterns. When one of them performs well, the other may not perform as well, and vice versa. They might have different market strategies or cater to different customer segments.

Weak to Moderate Positive Correlations: There are moderate positive correlations (around 0.5 to 0.8) between various airlines such as Alaska Airlines, JetBlue, Skywest, and Southwest. This suggests some degree of similarity in their performance patterns. United Airlines, Skywest Airlines, and Deta Airlines have moderate positive correlations with Southwest, indicating some similarities in their performance trends.

Weak Correlations: Alaska Airlines and JetBlue have a very weak positive correlation (approximately 0.46). This implies that they are relatively independent of each other in terms of performance metrics. Similar weak correlations exist between Alaska Airlines and Southwest, JetBlue and Southwest, Alaska Airlines and SkyWest, and Spirit and Alaska Airlines, indicating little to no linear relationship between their performance metrics.

Combined findings Alaska Airlines stands out with the highest mean performance, but it also has the highest variability in performance among all the airlines. This suggests that while it performs well on average, its performance can be volatile and subject to significant fluctuations.

American Airlines and United Airlines are strongly positively correlated and have relatively high mean performances. This indicates that these two airlines share similar market conditions or are affected by similar factors, and they both perform well compared to other airlines. Their correlation might indicate a certain level of code-sharing or partnerships to strengthen their market presence;

they could explore potential additional collaboration or partnership opportunities based on their similarities to leverage their strengths.

Spirit Airlines shows the least variability with the lowest standard deviation, indicating that its performance remains more consistent over time compared to other airlines.

The negative correlation between Alaska Airlines and Spirit Airlines indicates that when one of them performs well, the other tends to perform poorly. This suggests that they might be in direct competition or have contrasting market strategies. Understanding the reasons behind this negative correlation could help both airlines identify opportunities for improvement or differentiation in their respective markets.

Overall, the combination of correlation and descriptive statistics helps identify relationships and patterns among the airlines' performance metrics. It provides insights into which airlines are similar or dissimilar in their market performance, which airlines are more stable or volatile, and which airlines tend to outperform others on average. These findings can assist in strategic decision-making, partnership evaluations, and market positioning for the airlines in the industry. However, further analysis and domain knowledge are required to interpret these findings fully and make informed business decisions.

2 Technical Exposition

The approach taken in this code is to analyze multiple datasets related to airline operations, flight traffic, stock prices, and events in the USA. The goal is to gain insights from the data and build predictive models for flight traffic delays and airline stock prices. The code involves data cleaning, feature engineering, exploratory data analysis (EDA), and modeling using various machine learning algorithms. The models' performances are evaluated using appropriate metrics, and an ensemble model is developed to predict airline stock prices.

An ensemble model is developed for predicting airline stock prices. For each airline, three regressors (RandomForestRegressor, LinearRegression, GradientBoostingRegressor) are trained separately. Then, a VotingRegressor is created, combining the predictions of these regressors. The performance of the ensemble model is evaluated using mean absolute error, mean squared error, and R-squared.

Data manipulation and exploration

The data manipulation process involves reading multiple CSV files and performing data cleaning to handle missing and problematic values. The clean_data() function is used to replace empty strings with NaN and remove specific problematic characters from the datasets. The cleaned data is then saved back to the files.

After cleaning, the datasets are read into pandas DataFrames. The flight_traffic.csv file is modified to create a new binary column 'delay_occurred,' indicating whether any delay occurred on a particular flight. The 'year', 'month', and 'day' columns are combined into a single datetime column, 'date'.

The stock_prices.csv file is merged with the events_US.csv file based on the 'timestamp' and 'date' columns, respectively. The number of events for each date is counted, and the datasets are combined, with NaN values filled as 0.

2.1 Reading, cleaning, and wrangling the data

Quality control

The ‘clean_data()’ function addresses common data quality issues like empty strings and specific problematic characters in the datasets. These data issues are replaced with NaN or removed to ensure better data quality.

Transforming datasets

The flight_traffic.csv file is transformed to include a new column, ‘delay_occurred,’ which indicates whether any delay occurred on a specific flight. The ‘year’, ‘month’, and ‘day’ columns are merged into a single ‘date’ column, facilitating time-based analysis.

Feature engineering

In the new_stock.csv file, a new column ‘event_count’ is created, representing the count of events on each day. Similarly, in the flight_traffic_processed.csv file, a new column ‘event_day’ is created, indicating whether any event occurred on a particular day. These new features might help in understanding the relationship between flight delays and events.

2.2 Exploratory data analysis

EDA

The EDA process involves descriptive statistics and data visualization to gain insights into the datasets. Descriptive statistics are used to summarize key statistics of the flight_traffic.csv and new_stock.csv files. Data visualization techniques, such as histograms and heatmaps, are employed to understand the distributions of delay occurrences, stock prices, and correlations between delay columns and stock prices.

Hypothesis test and ad-hoc studies

The code includes various ad-hoc studies and correlation analyses. For instance, the code calculates the correlation between ‘event_count’ and each airline’s stock price. It also explores the correlation between flight delay occurrences and the number of events.

How we interpreted the results of the tests and analyses

The results of the EDA provide insights into the data distributions, patterns, and correlations between different variables. These insights are used to inform subsequent decisions, such as model selection and feature engineering.

Patterns noticed

During the exploratory data analysis (EDA), several patterns and insights were observed from the visualizations and correlation analyses. Some notable patterns include:

Correlation between Flight Delay Occurrences and Number of Events: The code investigated the relationship between flight delay occurrences and the number of events on a particular day. This analysis may help in understanding whether certain events contribute to an increased likelihood of flight delays.

Correlation between Event Count and Airline Stock Prices: The code calculated the correlation between the count of events on each day and the stock prices of different airlines. This correlation analysis can provide insights into how events impact the stock prices of specific airlines.

Stock Price Trends Over Time: The code visualized the trends of stock prices for different airlines over time. These visualizations can help identify overall market trends and seasonality patterns that may influence stock prices.

Model Performance Metrics: The code evaluated the performance of classifiers and regressors using various metrics, such as accuracy, precision, recall, ROC-AUC score, mean absolute error, mean squared error, and R-squared. By analyzing these performance metrics, decisions can be made regarding the suitability of different models for the prediction tasks. The observed patterns and insights from the EDA process can guide subsequent decisions, such as feature selection, model fine-tuning, and the development of ensemble models. Additionally, they can help in interpreting the results of the predictive models and drawing meaningful conclusions from the analyses.

2.3 Analytics and modeling

Choices made

The code uses machine learning models like RandomForestClassifier, LogisticRegression, and SVC for flight traffic prediction. For stock price prediction, RandomForestRegressor, LinearRegression, and GradientBoostingRegressor models are employed. An ensemble model, VotingRegressor, is also created to combine the predictions of multiple regressors.

Feature selection process

In the flight traffic prediction, the code uses 'airline_id', 'origin_airport', and 'destination_airport' as categorical columns and performs one-hot encoding on them. For stock price prediction, one airline's stock price is predicted at a time, and the relevant columns are used as features.

Models

The code evaluates the performance of each classifier and regressor using various metrics like accuracy, precision, recall, ROC-AUC score, mean absolute error, mean squared error, and R-squared. These metrics help in understanding the models' capabilities and identifying any shortcomings.

Visualizations and statistical tests

The code includes visualizations like histograms, heatmaps, and line plots to showcase data distributions, correlations, and trends over time. Statistical tests are not explicitly performed in this code, but correlation analysis is used to explore relationships between variables.

2.4 Conclusion

In conclusion, this code provides a comprehensive analysis of multiple datasets related to airline operations, flight traffic, stock prices, and events. It employs data cleaning, feature engineering, exploratory data analysis, and machine learning models to gain insights and make predictions. The code's strengths include handling data quality issues, creating new features, and using a variety of models for predictions. However, potential areas for improvement could be further fine-tuning of model hyperparameters and conducting cross-validation for more robust evaluation. Overall, the code presents a structured approach to extract valuable information from diverse datasets and build predictive models for flight traffic delays and airline stock prices.

3 Appendix

3.1 Future analysis pathways

Trying to predict stock price movement is the key. With that in mind, here are a few future research areas:

Is there a correlation between airlines that have the most flights to destinations with the big events in a quarter and their stock price performance? Does this vary by customer sentiment in that region or the US (there are plenty of data sets that measure customer sentiment for the US as a whole and regionally)?

When there are weather events, how do we pick the winners and losers? For example, if there is a weather event close to an important regional hub of one airline, that would lead to delays and cancellations and weaker stock performance for that airline BUT also, is there a winner who gets all the overflow traffic having a regional hub not too far away from weather events.

On the weather event, getting into depth of what characteristic of weather event (time of day, month, duration, was it during a regional event, was it in a location where the airline had other hubs close by), competitor airline proximity that could explain the impact on flight traffic for that airline. And then answer what level of flight traffic dislocation is needed to show up in earnings for the quarter and therefore stock prices. In the end, what one should know is if they see a weather event with certain characteristics happening in a certain region, with what level of certainty can they buy or sell a stock?

Remember, when trying to find stock price moves, we must look for relative stock price moves of one airline vs another. All airlines may go up and down together for various macro reasons, so we want to find what causes that specific airline to underperform or outperform the others based on the hypothesis.

Another different angle could be around airline fares. First, we establish if a change in airline fares has any predictive power on stock prices and do the bigger airlines (more routes) have a bigger impact on their stock price in that scenario. If there is a decent correlation, then the goal is to predict airline fare increases based on demand (full airplanes today = higher fare tomorrow?), supply characteristics (some regions have less competition), consumer sentiment, and cost dynamics (inflation, oil prices, etc.).

To identify the characteristics of a weather event, other ideas could be around the frequency of events (does 2 in the same month make a difference?), are there any ways to characterize the severity of the event (rain, wind speed, etc?). Also, test the importance of that hub with weather events to airlines to what % impact on airline traffic.

4 Code and graphics

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪roc_auc_score

from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

from sklearn.ensemble import VotingRegressor

os.chdir('/content/drive/MyDrive/Citadel Datathon/
 ↪Summer_Invitational_2023_Datathon_Datasets/')

def clean_data(file_name):
    # Read the file with 'ISO-8859-1' encoding
    data = pd.read_csv(file_name, encoding='ISO-8859-1')

    # Replace problematic characters with NaN or remove them
    data = data.replace(r'^\s*s', np.nan, regex=True)  # Replace empty strings␣
 ↪with NaN
    data = data.replace({"\x92": "", "\xe4": "", "\x80": ""}, regex=True)  #␣
 ↪Remove problematic characters

    # Save the cleaned data back to the file
    data.to_csv(file_name, index=False)

# List of your file names
files = ["airlines.csv", "airports.csv", "events_US.csv", "fares.csv",␣
 ↪"flight_traffic.csv", "stock_prices.csv", "weather.csv"]

# Clean all files
for file in files:
    clean_data(file)

# Read the file, treating empty fields as NaN
airports = pd.read_csv('airports.csv')
events_US = pd.read_csv('events_US.csv')
airlines = pd.read_csv('airlines.csv')
fares = pd.read_csv('fares.csv')
flight_traffic = pd.read_csv('flight_traffic.csv')
stock_prices = pd.read_csv('stock_prices.csv')
weather = pd.read_csv('weather.csv')
```

[ ]:

```
# Modifying the flight_traffic.csv file

# Define the columns to check
delay_cols = ['airline_delay', 'weather_delay', 'air_system_delay',␣
 ↪'security_delay', 'aircraft_delay']

# Create a new binary column 'delay_occurred'
# The new column is 1 if any delay is greater than 0 or NaN, else 0
flight_traffic['delay_occurred'] = flight_traffic[delay_cols].apply(lambda row:␣
 ↪int(any(row > 0)), axis=1)

# Combine 'year', 'month', and 'day' in flight_traffic to a single datetime␣
 ↪column
flight_traffic['date'] = pd.to_datetime(flight_traffic[['year', 'month',␣
 ↪'day']])
flight_traffic.drop(columns=['year', 'month', 'day'], inplace=True)

flight_traffic.to_csv('flight_traffic.csv', index=False)
```

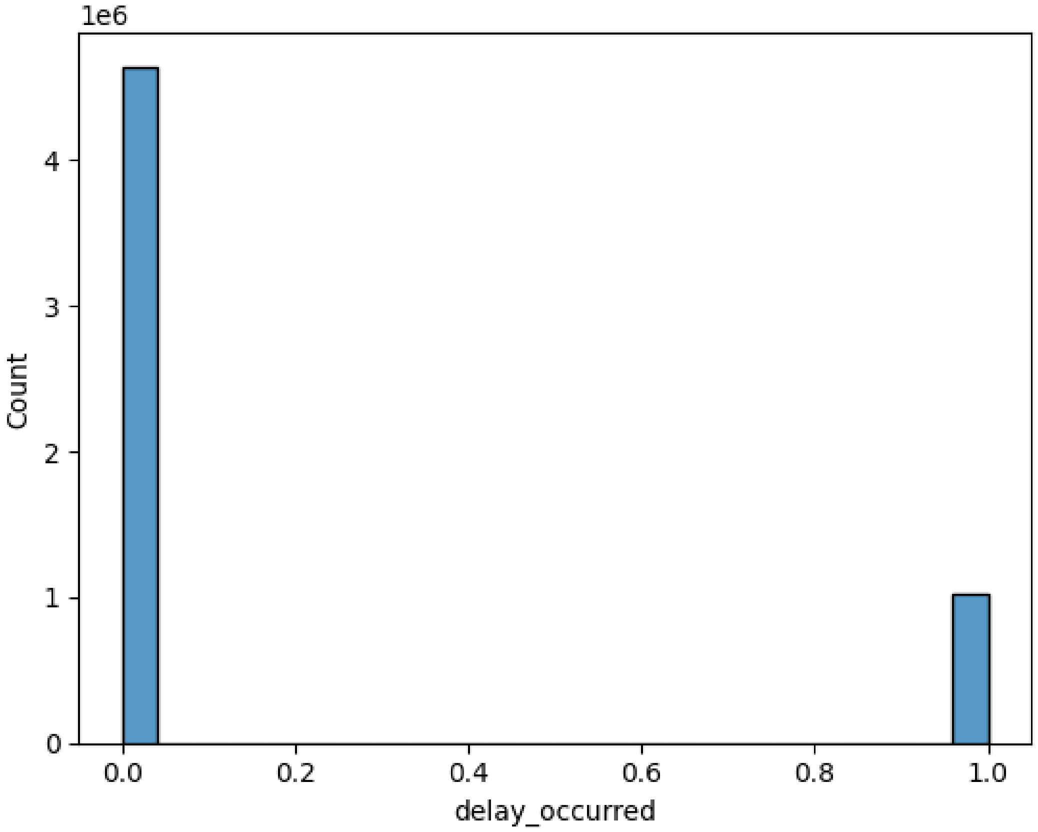
[ ]:

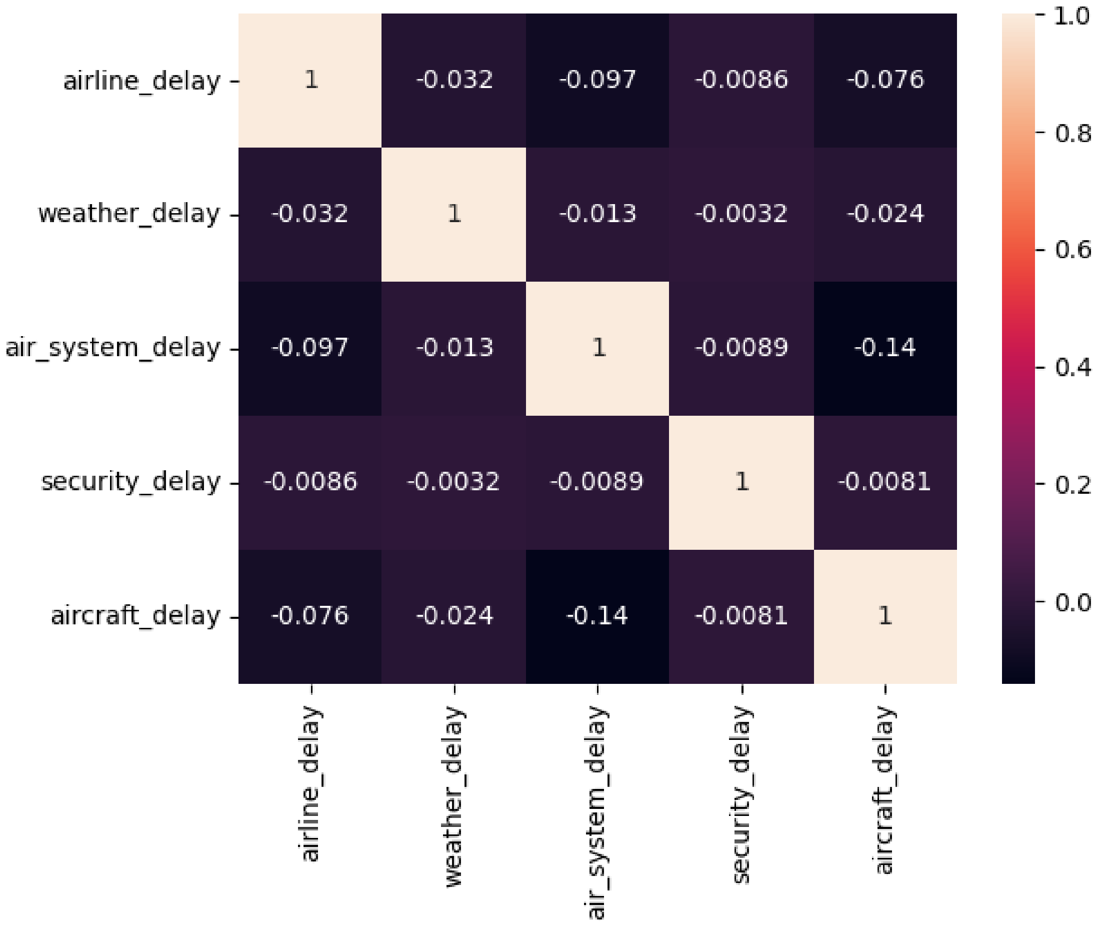
```
# Merging the stock_prices.csv file and the events_US.csv file

# Make sure that the timestamp and date columns are in the correct datetime␣
 ↪format
stock_prices['timestamp'] = pd.to_datetime(stock_prices['timestamp'],␣
 ↪format='%m/%d/%y', errors='coerce')
events_US['date'] = pd.to_datetime(events_US['date'], format='%d/%m/%Y',␣
 ↪errors='coerce')

# Remove the rows where timestamp or date couldn't be converted
stock_prices.dropna(subset=['timestamp'], inplace=True)
events_US.dropna(subset=['date'], inplace=True)

# Count the number of events for each date
event_counts = events_US.groupby('date').size().reset_index(name='event_count')

# Merge the datasets on the 'timestamp' and 'date' columns
new_stock = pd.merge(stock_prices, event_counts, left_on='timestamp',␣
 ↪right_on='date', how='left')

# Fill NaN values with 0 (assuming that NaN means there were no events at that␣
 ↪time)
new_stock['event_count'] = new_stock['event_count'].fillna(0)

# Convert 'event_count' column to integers
new_stock['event_count'] = new_stock['event_count'].astype(int)
```

```
# Drop 'date' column
new_stock.drop(columns=['date'], inplace=True)

# Save the new dataset
new_stock.to_csv('new_stock.csv', index=False)
```

[ ]:

```
# Flight Traffic Dataset

# Descriptive Statistics
flight_traffic.describe()

# Visualizing the distributions
sns.histplot(data=flight_traffic, x="delay_occurred")
plt.show()

# Check correlation of delay columns
sns.heatmap(flight_traffic[delay_cols].corr(), annot=True)
plt.show()
```

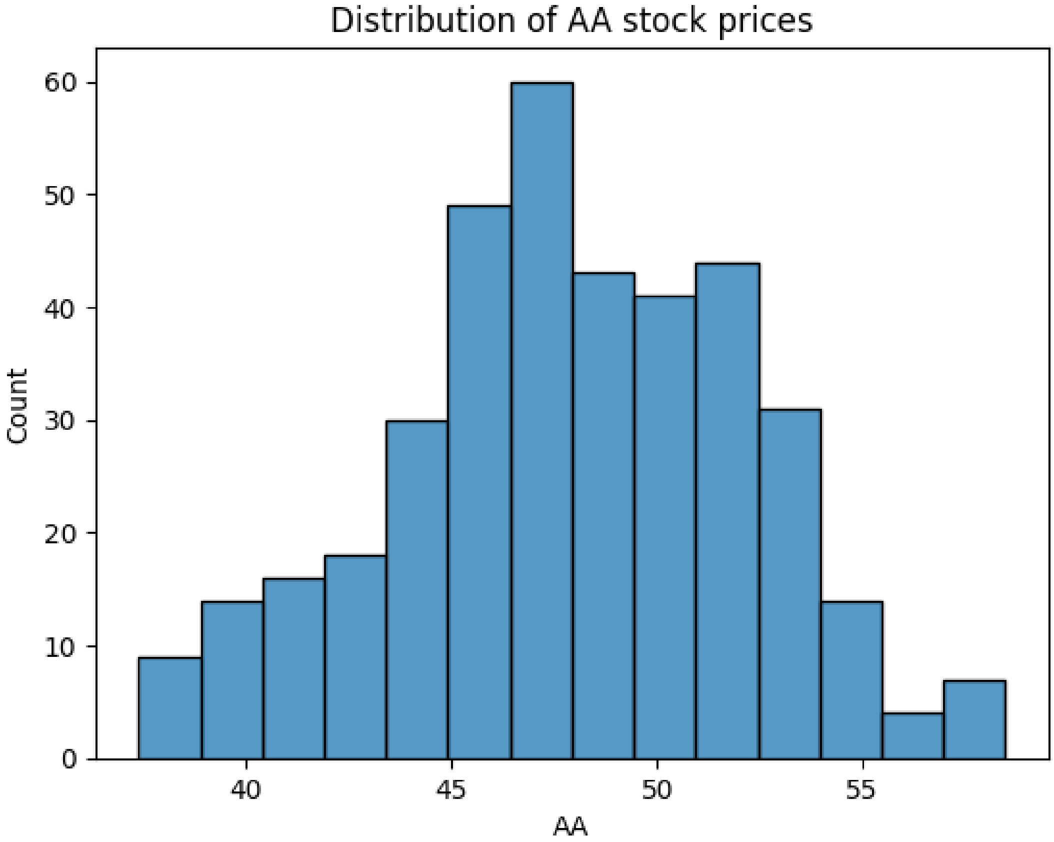


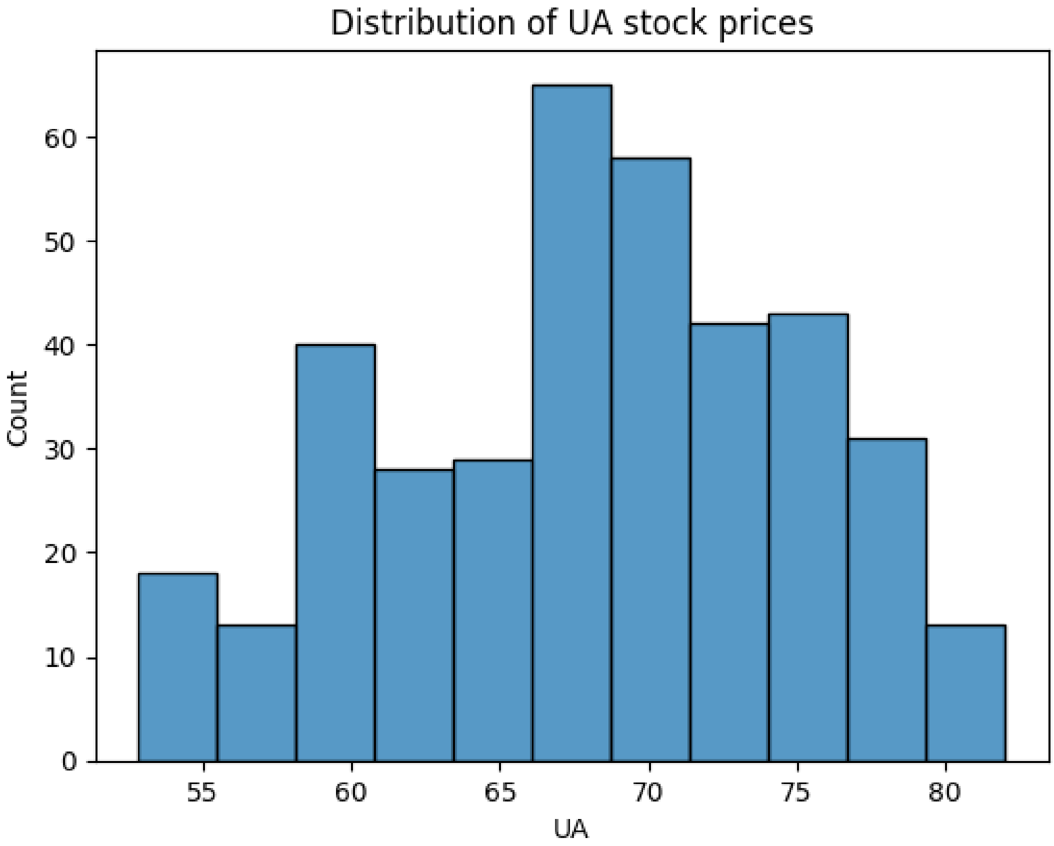


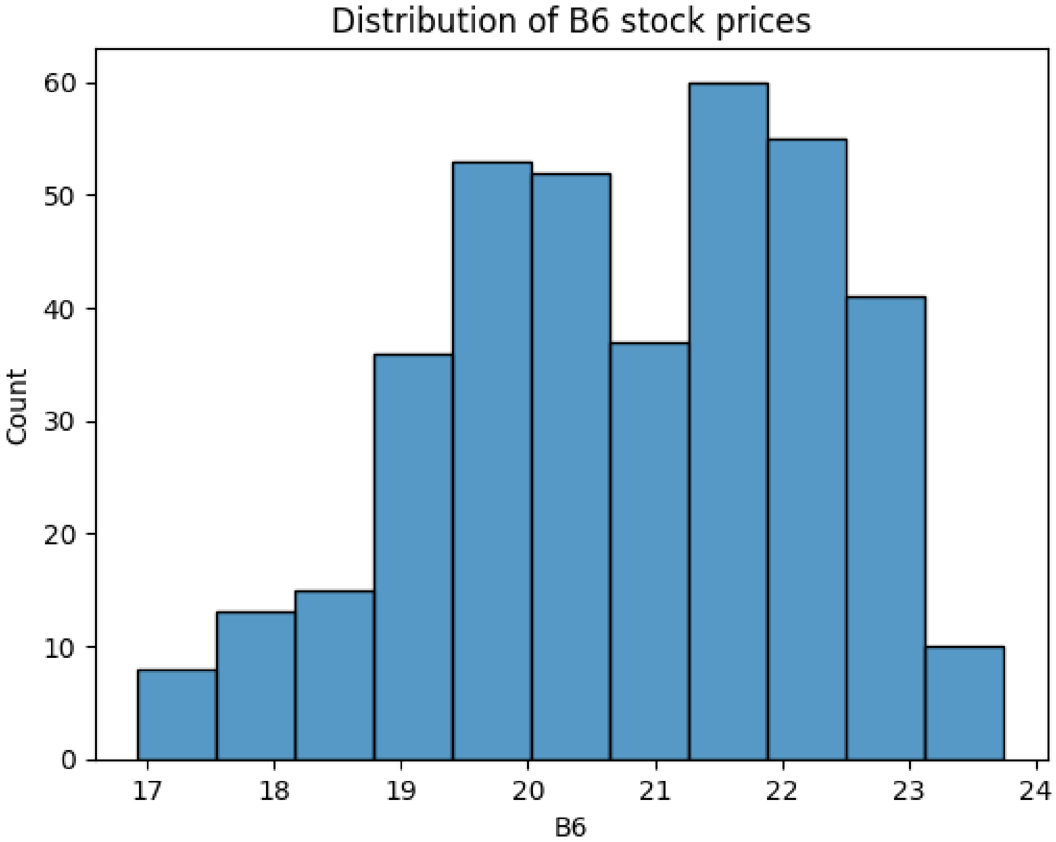
[ ]:

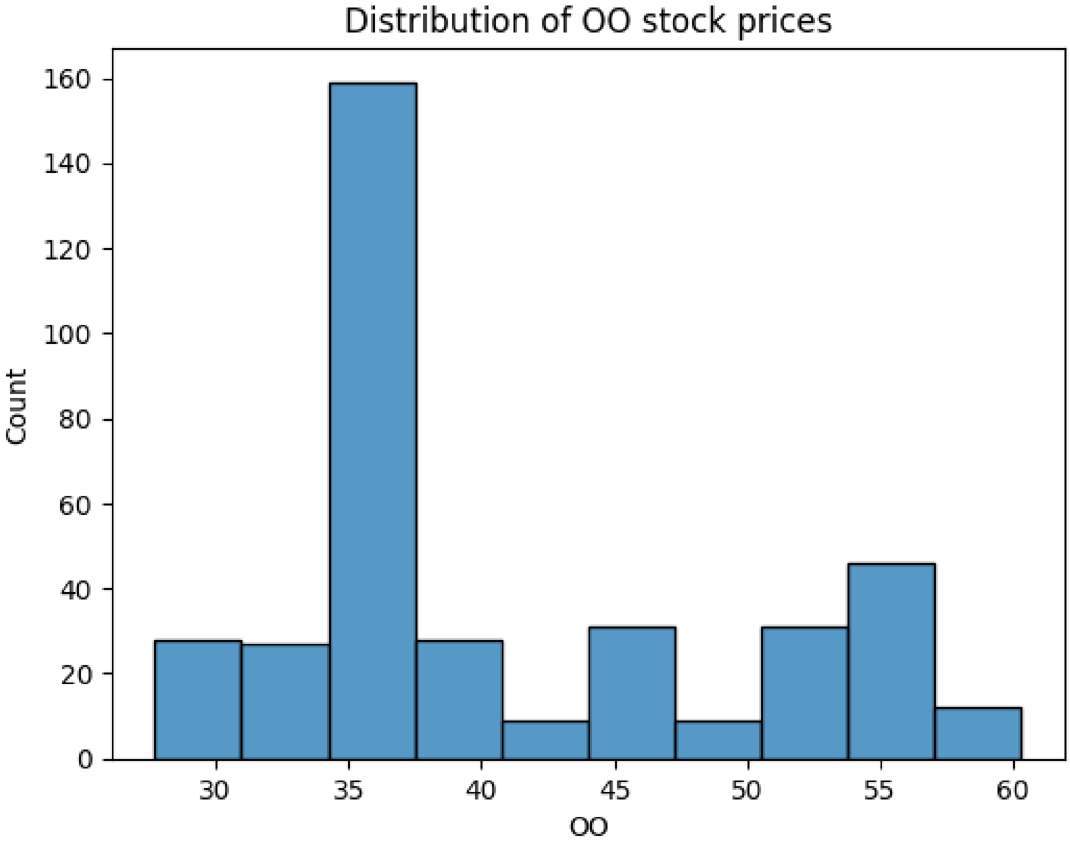
```
# Stock Dataset

# Descriptive Statistics
new_stock.describe()

# Visualizing the distributions of each airline's stock price
for airline in ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']:
    sns.histplot(data=new_stock, x=airline)
    plt.title(f'Distribution of {airline} stock prices')
    plt.show()

# Check the trend of stock prices over time for each airline
for airline in ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']:
    new_stock.plot(x='timestamp', y=airline)
    plt.title(f'Trend of {airline} stock prices over time')
    plt.show()

# Visualizing the distributions
```

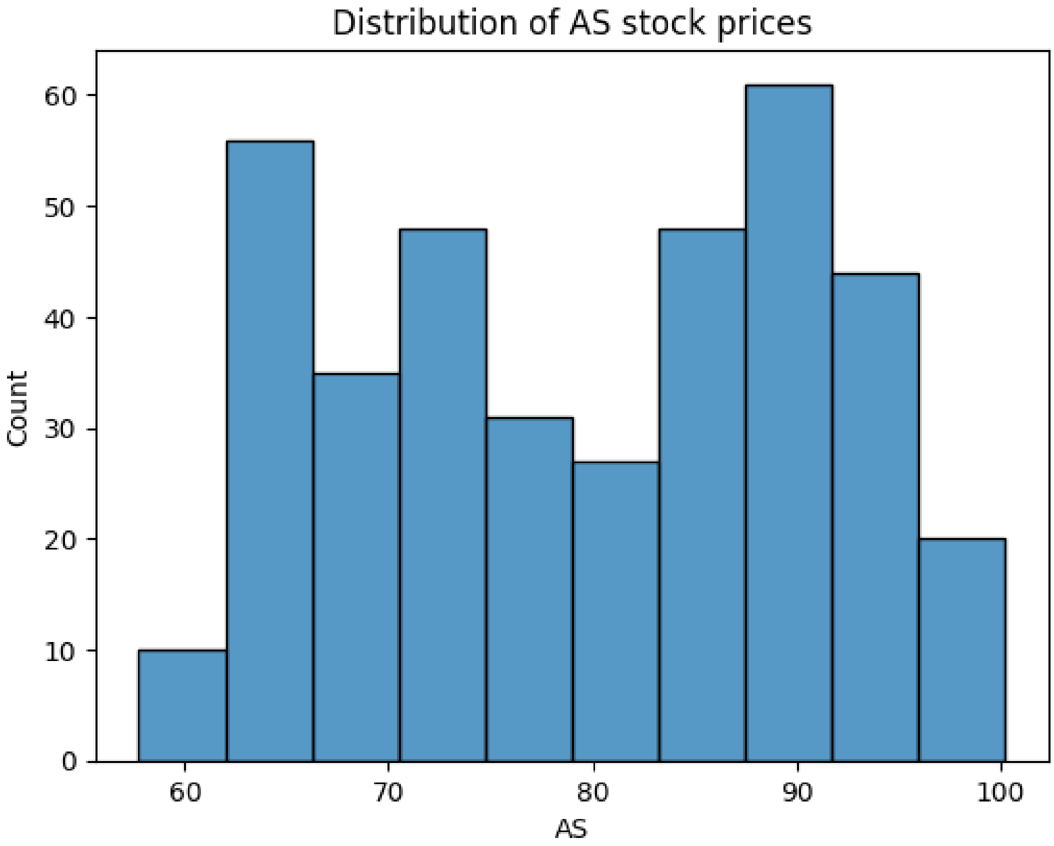
```
sns.histplot(data=new_stock, x="event_count")
plt.show()
```

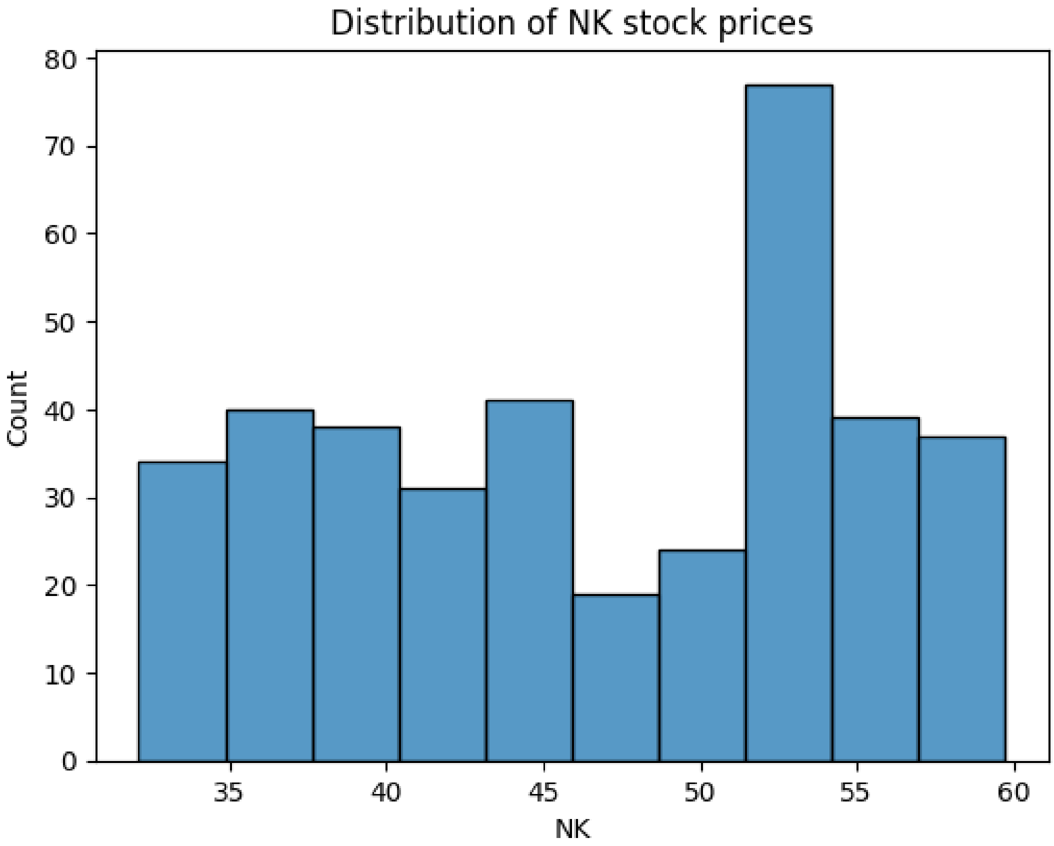


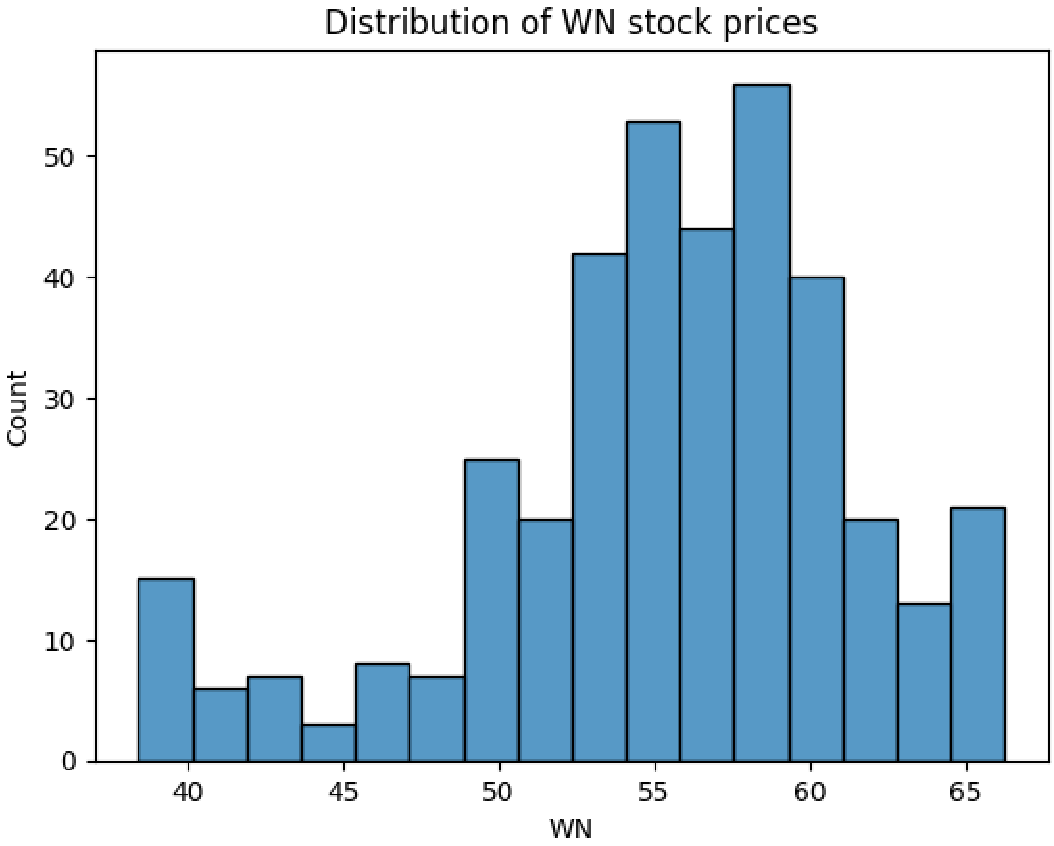

Distribution of UA stock prices
Count
0
10
20
30
40
50
60
55
60
65
70
75
80
UA

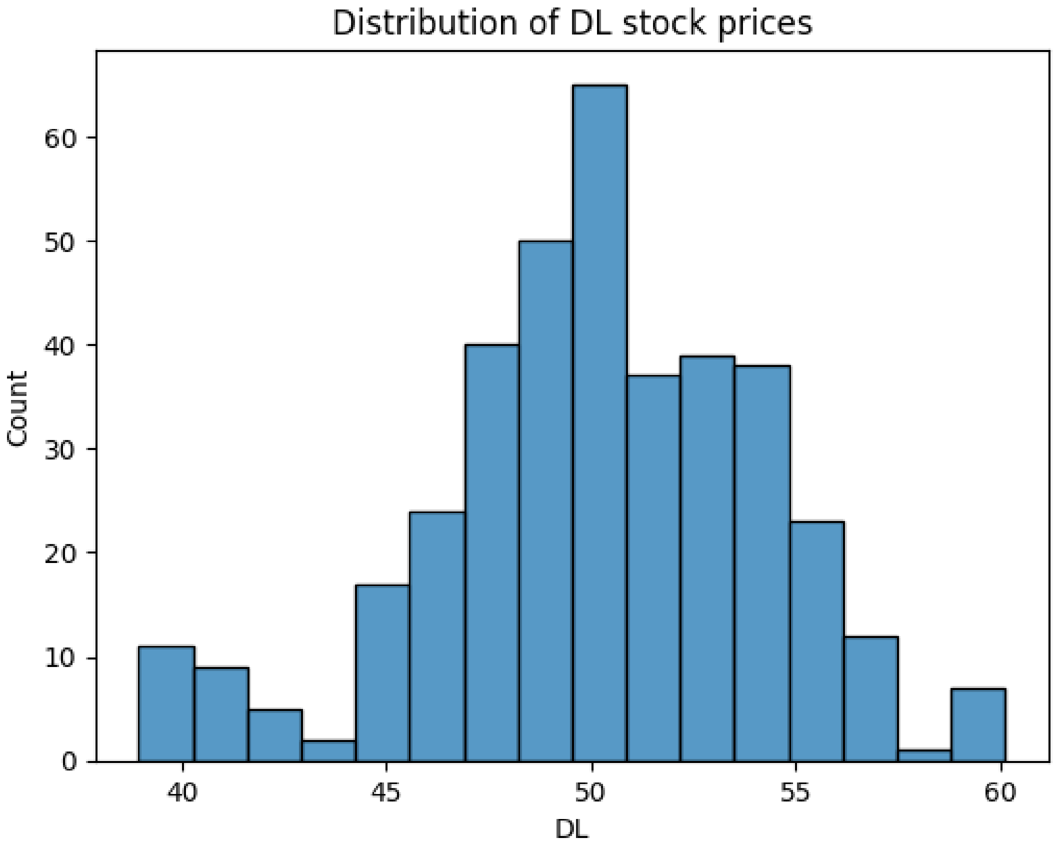

Distribution of B6 stock prices
Count
0
10
20
30
40
50
60
17
18
19
20
21
22
23
24
B6

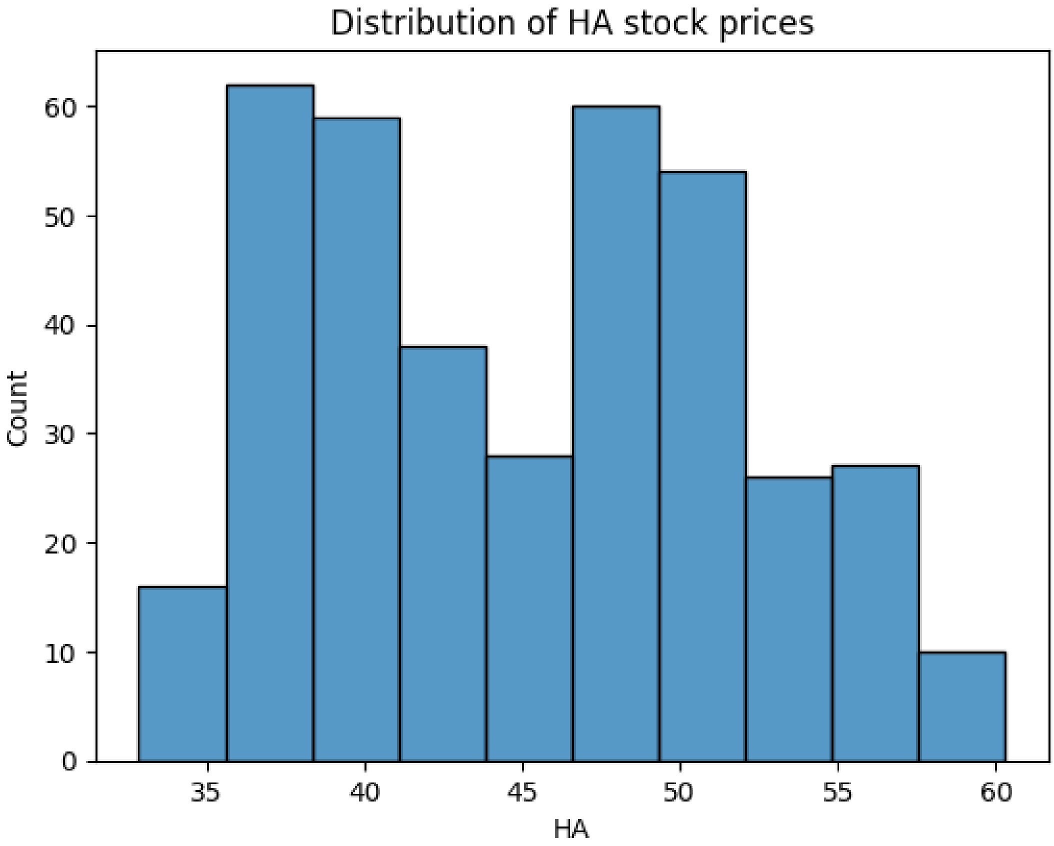

Distribution of OO stock prices
Count
0
20
40
60
80
100
120
140
160
30
35
40
45
50
55
60
OO

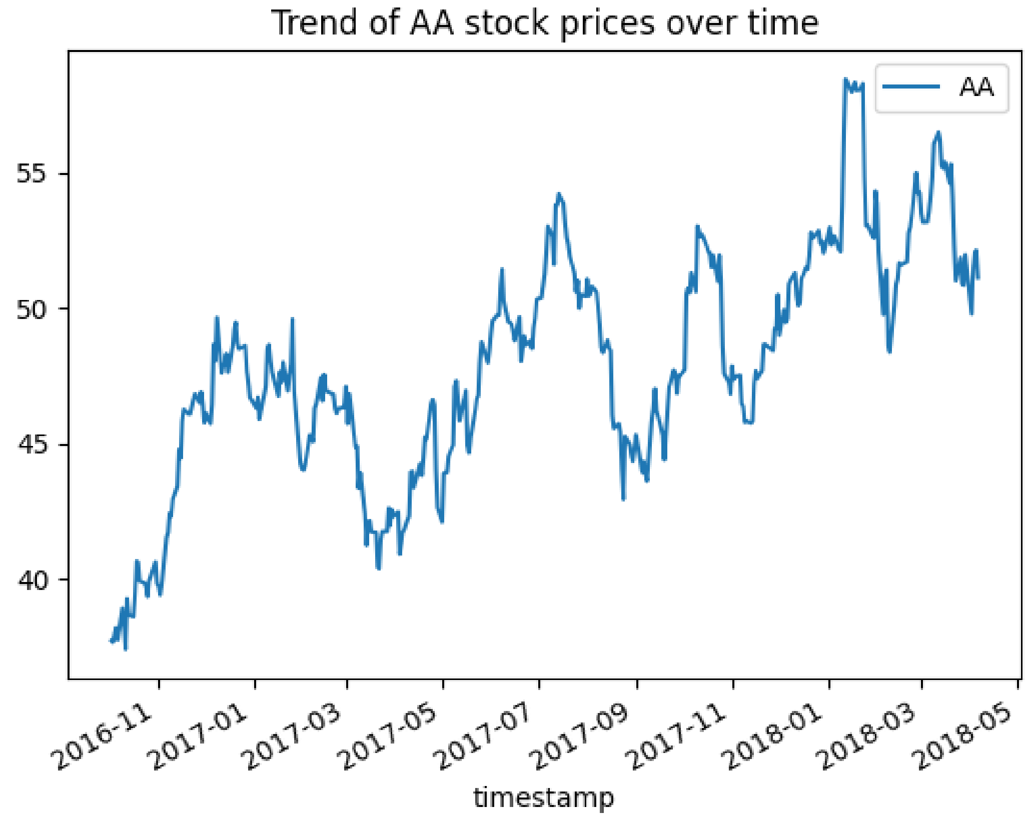

Distribution of AS stock prices
Count
0
10
20
30
40
50
60
AS
60
70
80
90
100

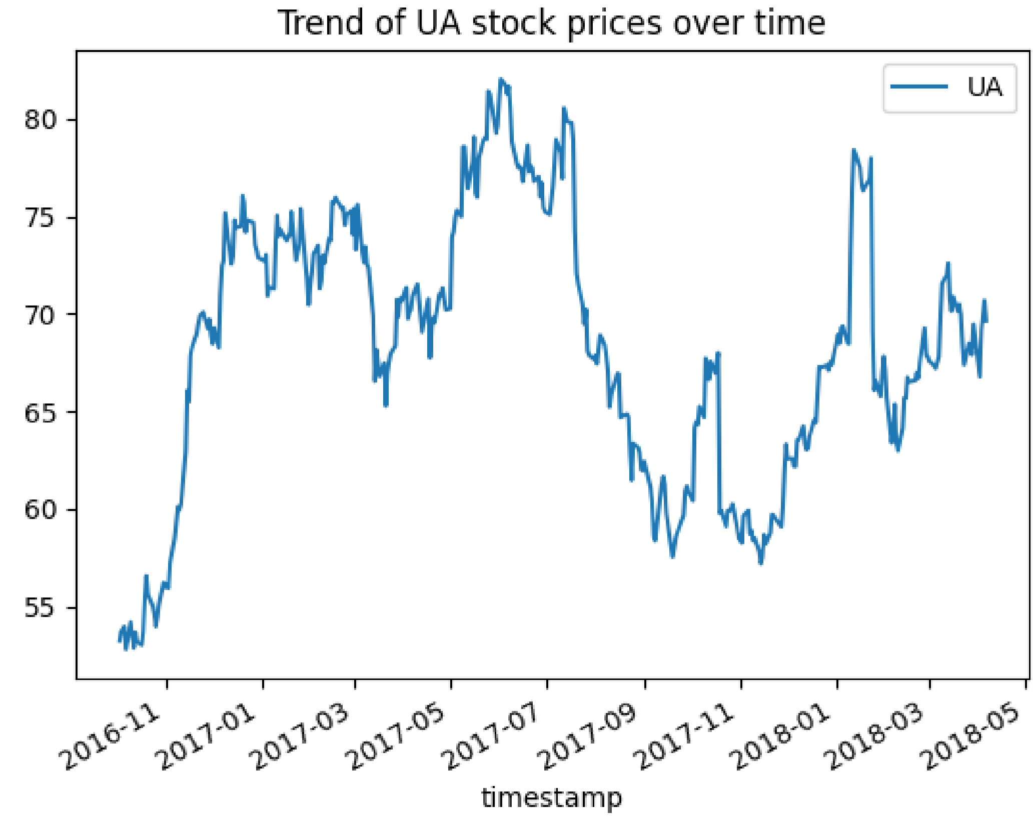

Distribution of NK stock prices
Count
0
10
20
30
40
50
60
70
80
35
40
45
50
55
60
NK

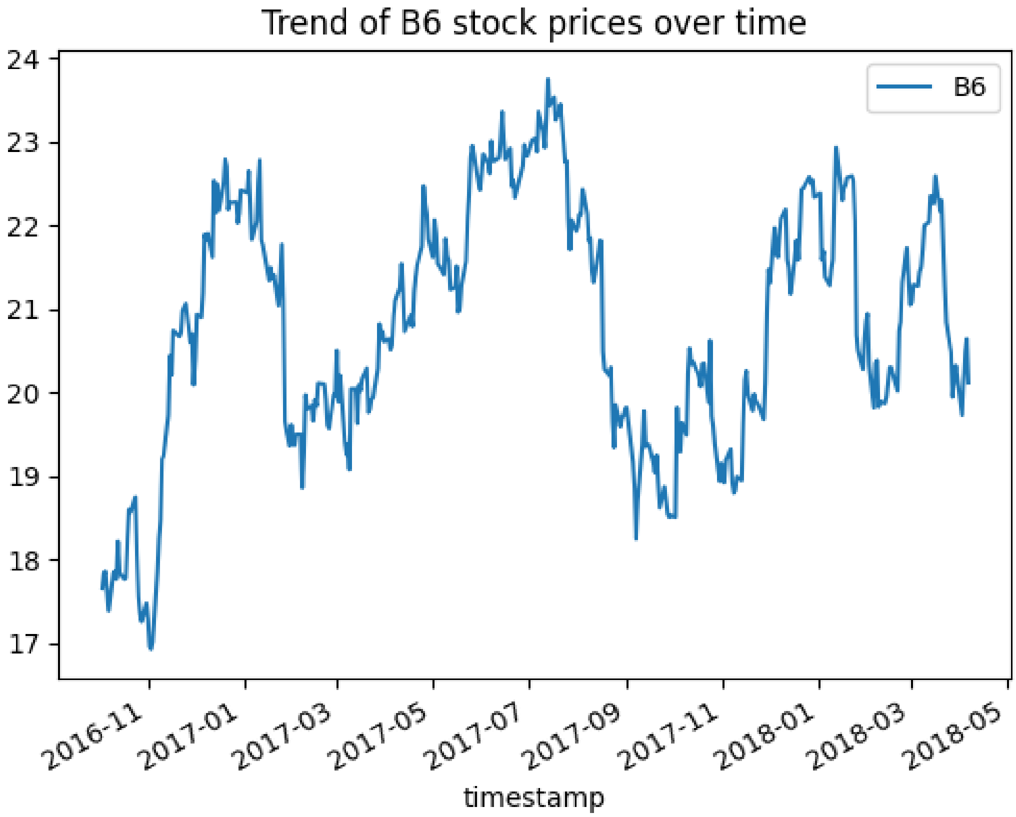

Distribution of WN stock prices
Count
0
10
20
30
40
50
40
45
50
55
60
65
WN

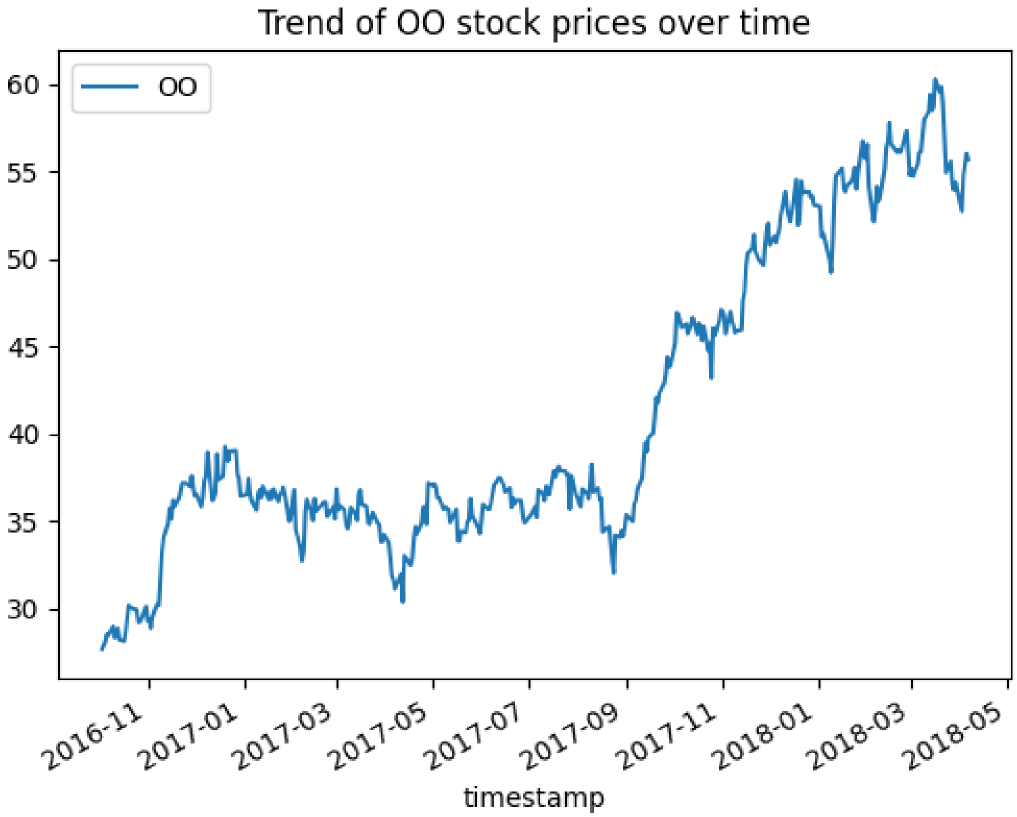

Distribution of DL stock prices
Count
0
10
20
30
40
50
60
DL
40
45
50
55
60

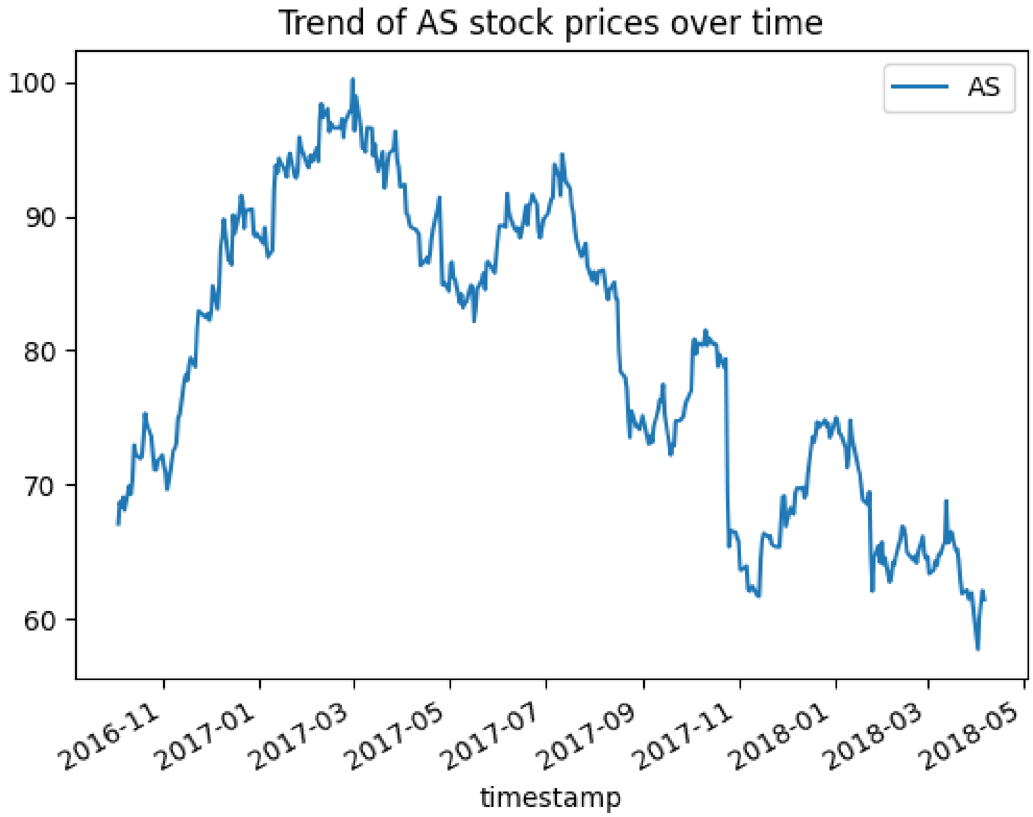

Distribution of HA stock prices
Count
0
10
20
30
40
50
60
35
40
45
50
55
60
HA

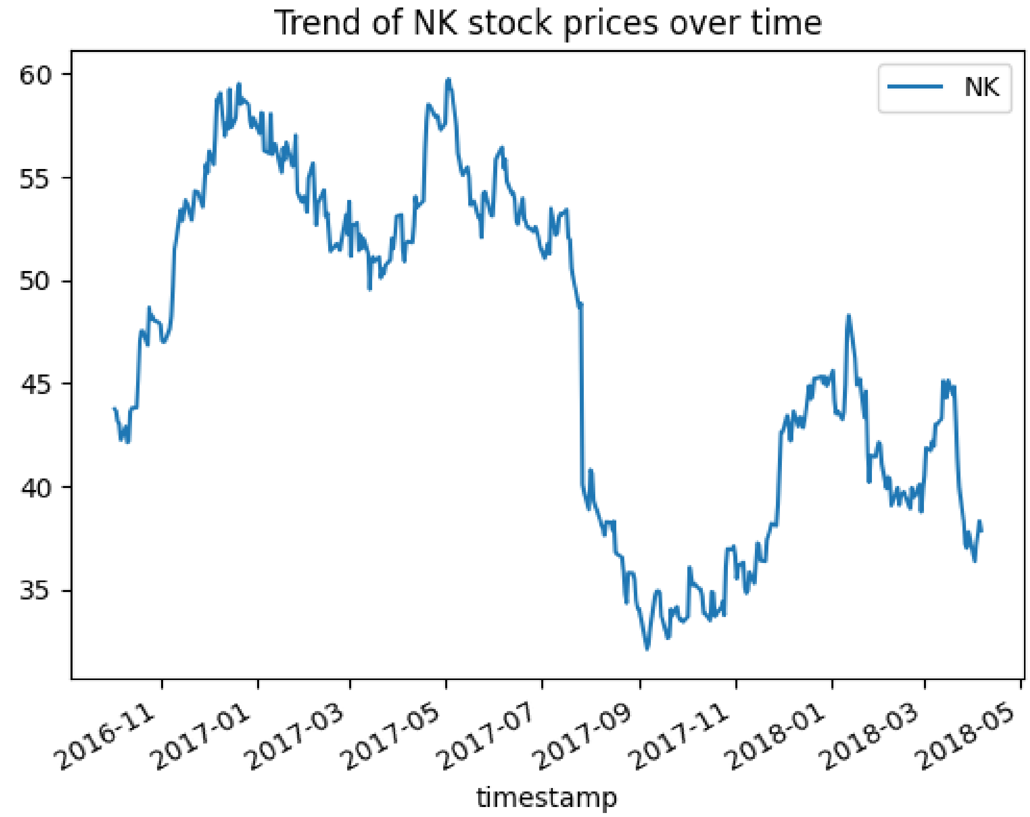

Trend of AA stock prices over time
AA
55
50
45
40
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp

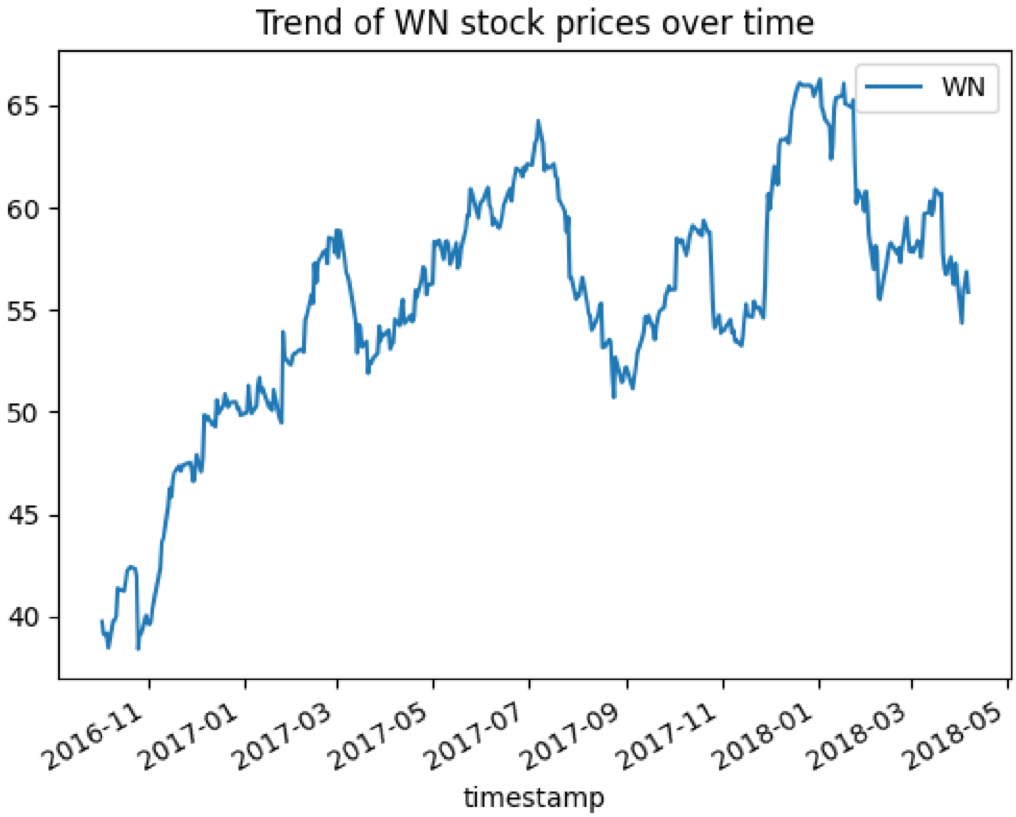

Trend of UA stock prices over time
UA
80
75
70
65
60
55
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp

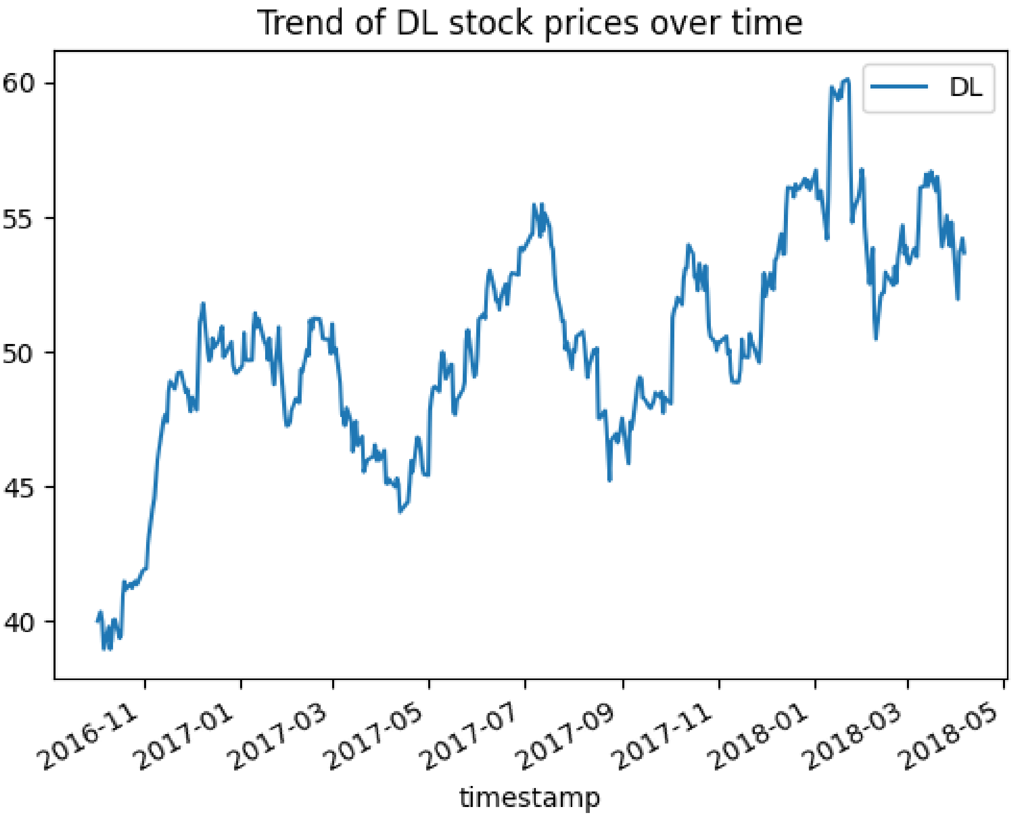

Trend of B6 stock prices over time
24
23
22
21
20
19
18
17
B6
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp

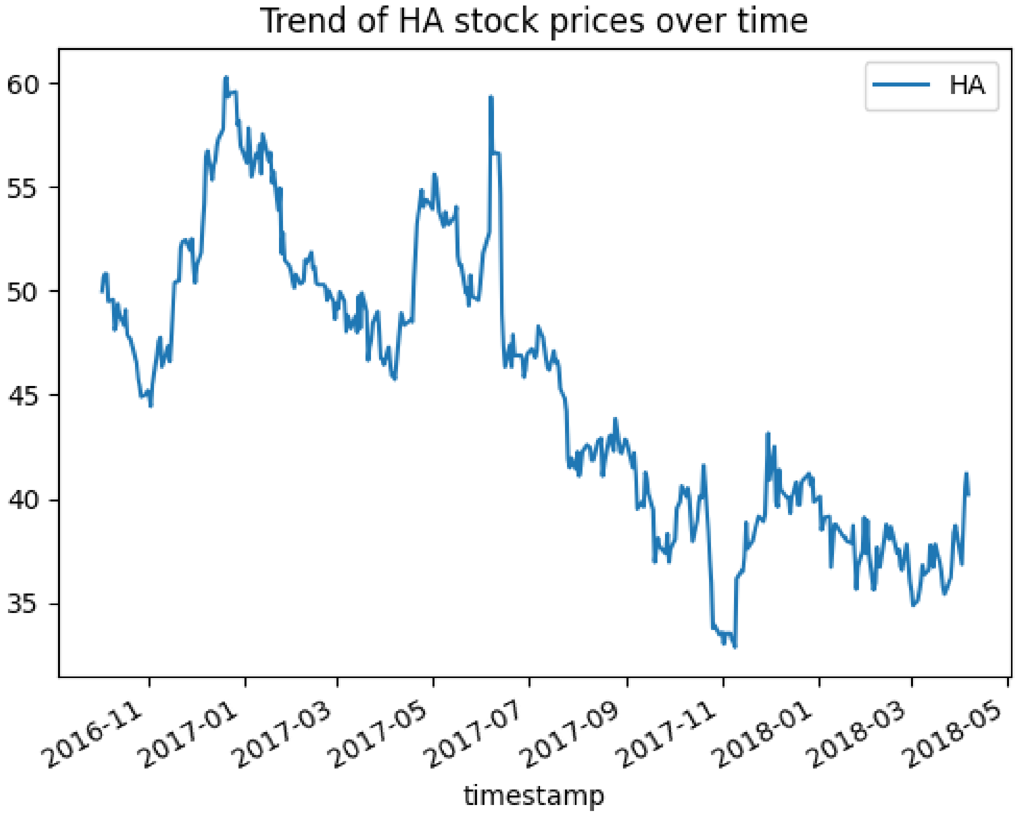

Trend of OO stock prices over time
OO
60
55
50
45
40
35
30
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp

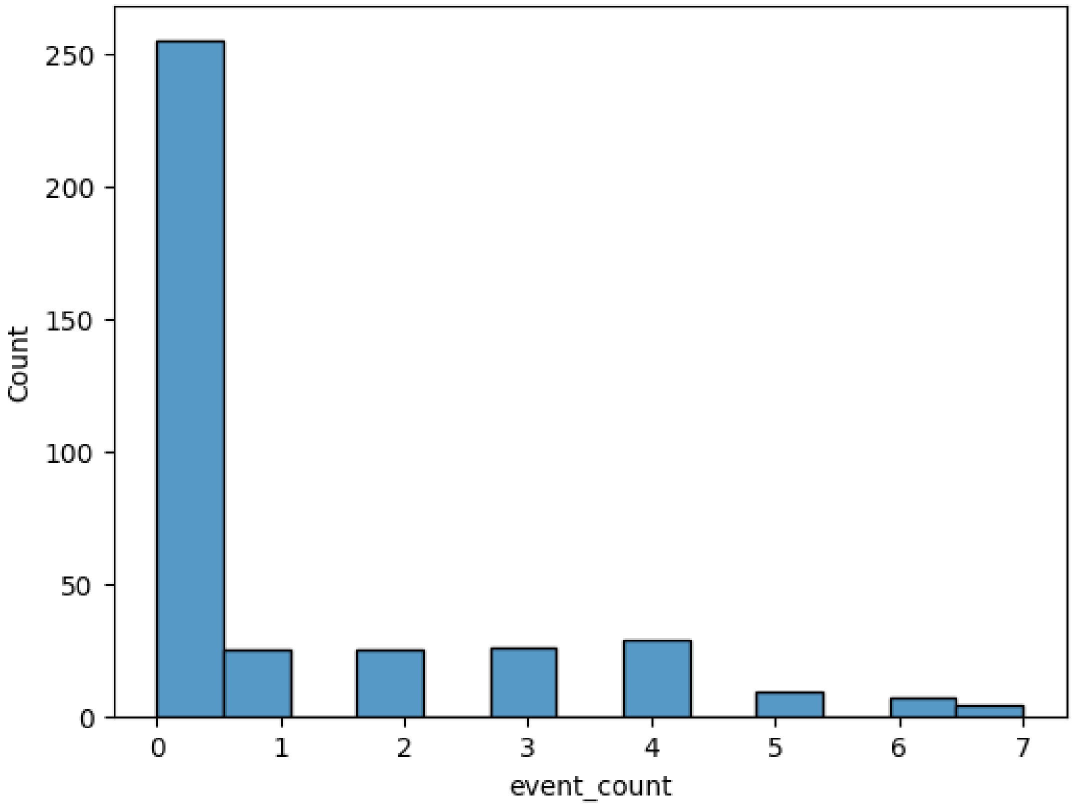

Trend of AS stock prices over time
AS
100
90
80
70
60
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp


Trend of NK stock prices over time
NK
60
55
50
45
40
35
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp


Trend of WN stock prices over time
WN
65
60
55
50
45
40
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp


Trend of DL stock prices over time
DL
60
55
50
45
40
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp

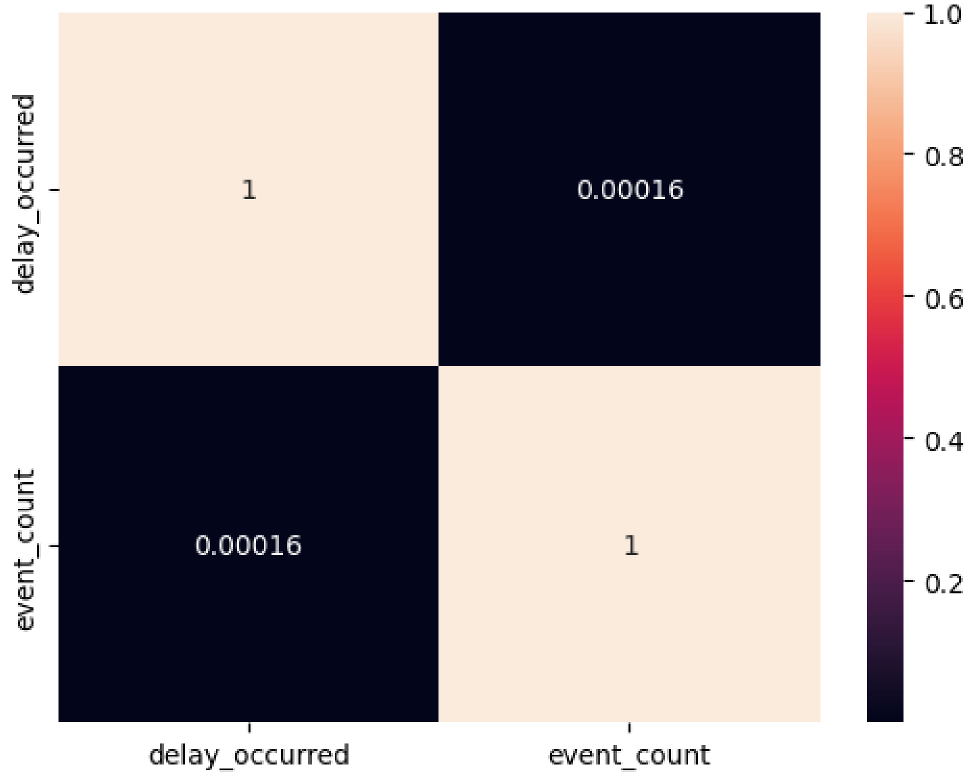

Trend of HA stock prices over time
HA
60
55
50
45
40
35
2016-11
2017-01
2017-03
2017-05
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
timestamp



[ ]:

```
# Check correlation between event_count and each airline's stock price
for airline in ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']:
    print(f"Correlation between event_count and {airline}'s stock price:",␣
  ↪new_stock[['event_count', airline]].corr().iloc[0,1])

# Correlation between flight delay occurrences and number of events
merged_flight_stock = pd.merge(flight_traffic, new_stock, left_on='date',␣
  ↪right_on='timestamp', how='inner')
sns.heatmap(merged_flight_stock[['delay_occurred', 'event_count']].corr(),␣
  ↪annot=True)
plt.show()
```

```
Correlation between event_count and AA's stock price: -0.09796177540098208
Correlation between event_count and UA's stock price: 0.10116632654785204
Correlation between event_count and B6's stock price: 0.07165759733887636
Correlation between event_count and OO's stock price: -0.20610406382362
Correlation between event_count and AS's stock price: 0.2692075272286969
Correlation between event_count and NK's stock price: 0.0190081591665456
Correlation between event_count and WN's stock price: 0.11406459573328544
Correlation between event_count and DL's stock price: -0.08704720859110898
Correlation between event_count and HA's stock price: 0.08467351442461774
```



```
[ ]: # Feature Engineering

     # Load the processed flight data and stock data
     flight_traffic = pd.read_csv('flight_traffic.csv')
     new_stock = pd.read_csv('new_stock.csv')

     # Convert 'date' and 'timestamp' columns to datetime
     flight_traffic['date'] = pd.to_datetime(flight_traffic['date'])
     new_stock['timestamp'] = pd.to_datetime(new_stock['timestamp'])

     # Create 'event_day' feature in flight_traffic dataset
     # It's 1 if there's any event on the day, else 0
     flight_traffic = pd.merge(flight_traffic, new_stock[['timestamp',␣
      ↪'event_count']], left_on='date', right_on='timestamp', how='left')
     flight_traffic['event_day'] = flight_traffic['event_count'].apply(lambda x: 1␣
      ↪if x > 0 else 0)

     # Create 'delay_day' feature in new_stock dataset
     # It's 1 if there's any delay on the day, else 0
     new_stock = pd.merge(new_stock, flight_traffic[['date', 'delay_occurred']],␣
      ↪left_on='timestamp', right_on='date', how='left')
```

```
new_stock['delay_day'] = new_stock['delay_occurred'].apply(lambda x: 1 if x > 0␣
 ↪else 0)

# Drop unnecessary columns
flight_traffic.drop(columns=['timestamp', 'event_count'], inplace=True)
new_stock.drop(columns=['date', 'delay_occurred'], inplace=True)

# Save the processed datasets
flight_traffic.to_csv('flight_traffic_processed.csv', index=False)
new_stock.to_csv('new_stock_processed.csv', index=False)
```

[ ]:

```
# Model development for flight traffic prediction

# Define the feature matrix X and the target vector y
X = flight_traffic.drop(columns=['delay_occurred'])
y = flight_traffic['delay_occurred']

# Identify categorical columns
categorical_columns = ['airline_id', 'origin_airport', 'destination_airport'] ␣
 ↪# add or remove columns as needed

# Perform one-hot encoding for categorical columns
X_encoded = pd.get_dummies(X, columns=categorical_columns)

# Split the dataset into a training set and a test set
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.
 ↪2, random_state=42)

# List of classifiers
classifiers = [
    RandomForestClassifier(random_state=42),
    LogisticRegression(random_state=42),
    SVC(probability=True, random_state=42),
]

# Dictionary of classifier performance
classifier_performance = {}

# Train and evaluate each classifier
for clf in classifiers:
    clf_name = clf.__class__.__name__

    # Train the classifier
    clf.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = clf.predict(X_test)
    y_pred_proba = clf.predict_proba(X_test)[:, 1]

    # Calculate performance metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    auc_roc = roc_auc_score(y_test, y_pred_proba)

    # Store the performance metrics
    classifier_performance[clf_name] = [accuracy, precision, recall, auc_roc]

# Convert the performance metrics into a DataFrame
performance_df = pd.DataFrame(classifier_performance, index=['Accuracy',␣
 ↪'Precision', 'Recall', 'AUC-ROC']).T

print(performance_df)
```

[ ]:

```
# Model development for stock prices prediction

# Define the airline stock names
airline_stocks = ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']

for airline_stock in airline_stocks:
    # Define the feature matrix X and the target vector y
    X = new_stock.drop(columns=[airline_stock])
    y = merged_flight_stock[airline_stock]

    # Split the dataset into a training set and a test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    # List of regressors
    regressors = [
        RandomForestRegressor(random_state=42),
        LinearRegression(),
        GradientBoostingRegressor(random_state=42),
    ]

    # Dictionary of regressor performance
    regressor_performance = {}

    # Train and evaluate each regressor
    for reg in regressors:
        reg_name = reg.__class__.__name__

        # Train the regressor
        reg.fit(X_train, y_train)
        # Make predictions on the test set
        y_pred = reg.predict(X_test)

        # Calculate performance metrics
        mae = mean_absolute_error(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)

        # Store the performance metrics
        regressor_performance[reg_name] = [mae, mse, r2]

    # Convert the performance metrics into a DataFrame
    performance_df = pd.DataFrame(regressor_performance, index=['Mean Absolute␣
↪Error', 'Mean Squared Error', 'R-squared']).T

    print(f"Performance metrics for {airline_stock} stock:")
    print(performance_df)
    print("\n")
```

[ ]:

```
# Ensemble model

# List of airline stocks
airline_stocks = ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']

# Dictionary to store trained regressors for each stock
stock_regressors = {}

for airline_stock in airline_stocks:
    # Define the feature matrix X and the target vector y
    X = new_stock.drop(columns=[airline_stock])
    y = merged_flight_stock[airline_stock]

    # Split the dataset into a training set and a test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

    # Train each regressor and store it
    regressors = [
        ('random_forest', RandomForestRegressor(random_state=42)),
        ('linear_regression', LinearRegression()),
        ('gradient_boosting', GradientBoostingRegressor(random_state=42)),
    ]

    # Train ensemble regressor
    ensemble = VotingRegressor(regressors)
    ensemble.fit(X_train, y_train)
    # Store the trained ensemble regressor
    stock_regressors[airline_stock] = ensemble

    # Evaluate the ensemble regressor
    y_pred = ensemble.predict(X_test)
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"Performance metrics for ensemble regressor on {airline_stock} stock:
 ↪")
    print(f"Mean Absolute Error: {mae}")
    print(f"Mean Squared Error: {mse}")
    print(f"R-squared: {r2}\n")
```

[ ]:

```
# Model Evaluation and Selection

# List of airline stocks
airline_stocks = ['AA', 'UA', 'B6', 'OO', 'AS', 'NK', 'WN', 'DL', 'HA']

# Dictionary to store trained regressors for each stock
stock_regressors = {}

# Dictionary to store performance metrics
performance_metrics = {}

for airline_stock in airline_stocks:
    # Define the feature matrix X and the target vector y
    X = new_stock.drop(columns=[airline_stock])
    y = new_stock[airline_stock]

    # Split the dataset into a training set and a test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    # List of regressors
    regressors = [
        ('random_forest', RandomForestRegressor(random_state=42)),
        ('linear_regression', LinearRegression()),
        ('gradient_boosting', GradientBoostingRegressor(random_state=42)),
    ]

    for reg_name, reg in regressors:
        # Train the regressor
        reg.fit(X_train, y_train)
        # Make predictions on the test set
        y_pred = reg.predict(X_test)

        # Calculate performance metrics
        mae = mean_absolute_error(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)

        # Store the performance metrics
        performance_metrics[(airline_stock, reg_name)] = [mae, mse, r2]

    # Train ensemble regressor
    ensemble = VotingRegressor(regressors)
    ensemble.fit(X_train, y_train)

    # Store the trained ensemble regressor
    stock_regressors[airline_stock] = ensemble

    # Evaluate the ensemble regressor
    y_pred = ensemble.predict(X_test)
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    # Store the performance metrics
    performance_metrics[(airline_stock, 'ensemble')] = [mae, mse, r2]

# Convert the performance metrics into a DataFrame
performance_df = pd.DataFrame(performance_metrics, index=['Mean Absolute␣
 ↪Error', 'Mean Squared Error', 'R-squared']).T
performance_df.reset_index(inplace=True)
performance_df.columns = ['Airline Stock', 'Regressor', 'Mean Absolute Error',␣
 ↪'Mean Squared Error', 'R-squared']

print(performance_df)
```

```
[ ]:
```