IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ziad El-Qurneh
15 March 2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- **Summary of methodologies:**
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Visual Dashboard with ploty
  - Machine Learning Prediction

- **Summary of all results:**
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- **Project background and context:**
  Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- **Problems you want to find answers:**
- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful
- landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

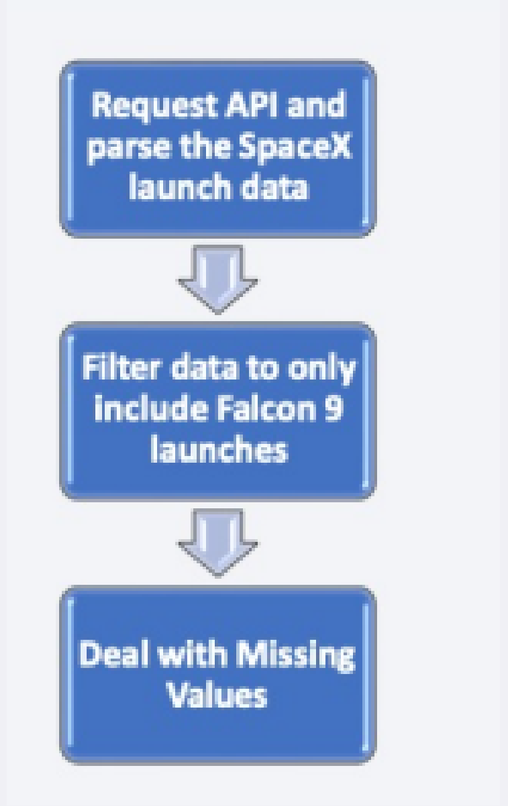
- Data collection methodology:
  - Data from SpaceX website
  - WebScraping from Wikipedia
- Perform data wrangling
  - Collected data got improved with a landing outcome label based on feature summarization and analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data gathered up to this point was normalized, separated into training and test data sets, and evaluated by four different classification models, with the accuracy of each model assessed using different parameter combinations.

Data Collection

- Describe how data sets were collected.
  The dataset was from SpaceX API :
  https://api.spacexdata.com/v4/rockets
  WebScraping from Wikipedia:
  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

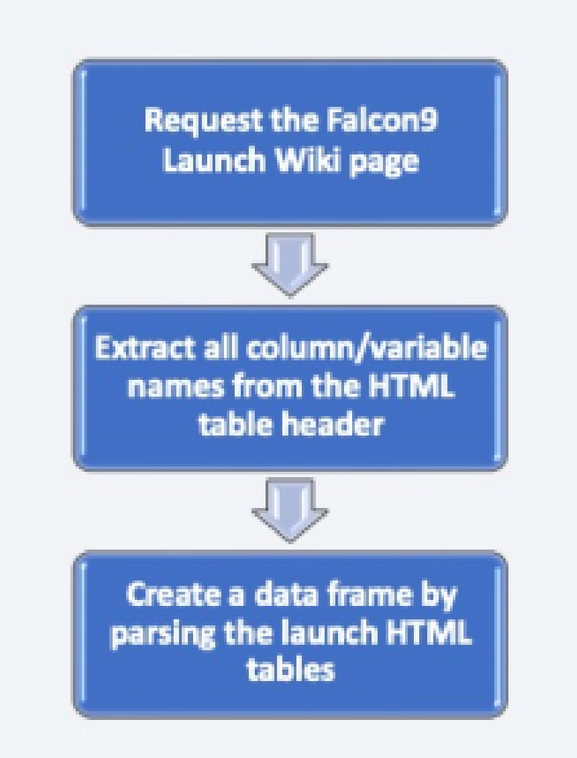
Data Collection – SpaceX API

- SpaceX provides a public API from which data can be retrieved and used. This API was used in accordance with the flowchart below, and data was then persisted.
- https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



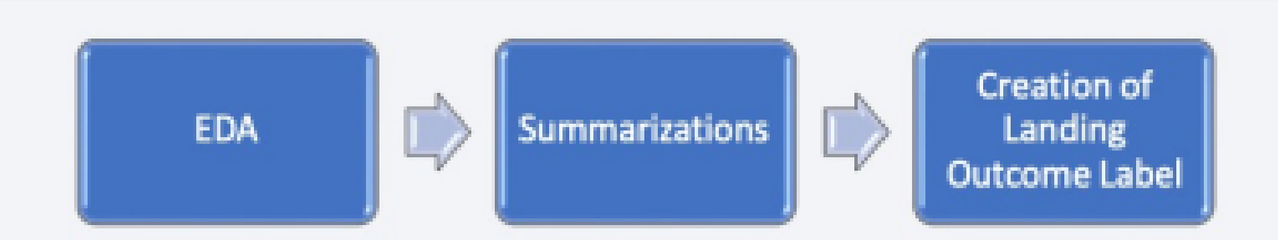
Data Collection - Scraping

- Data from SpaceX launches can also be found on Wikipedia. Data is obtained from Wikipedia and saved following the flowchart.
- https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

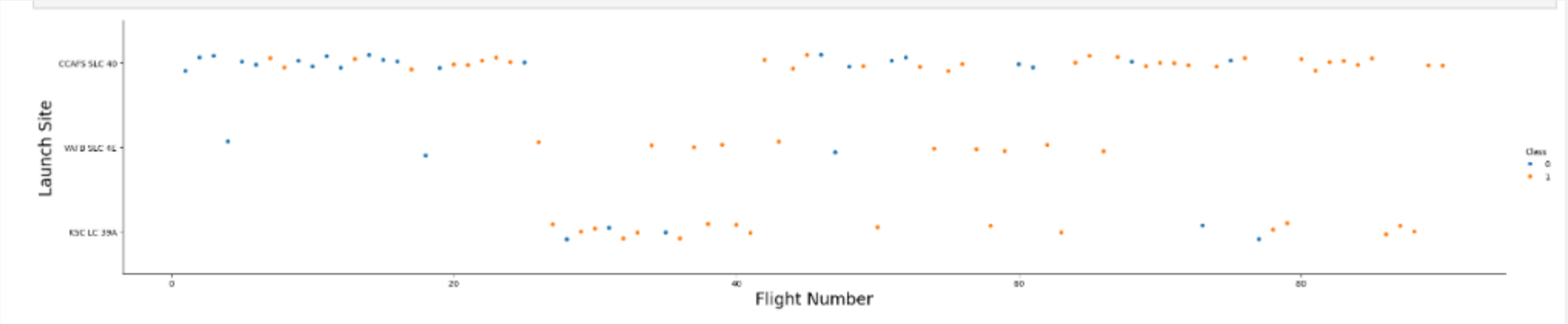
- At first, some exploratory data analysis (EDA) was performed on the dataset. The summary of launches per site, orbits, and mission outcomes per orbit type were calculated. In the end, the landing outcome label was produced using the Outcome column.



- https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- To analyze the data, scatterplots and bar charts were utilized to visualize the relationships between different feature pairs. The explored relationships included Payload Mass versus Flight Number, Launch Site versus Flight Number, Launch Site versus Payload Mass, Orbit versus Flight Number, and Payload versus Orbit.



https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

- The SQL queries executed included extracting the distinct names of launch sites involved in space missions and identifying the top five launch sites that start with 'CCA'. The total payload mass transported by NASA (CRS) boosters was determined, along with the average payload mass for the booster version F9 v1.1. Additionally, the query identified the date of the first successful landing on a ground pad. It also retrieved the names of boosters that successfully landed on a drone ship while carrying payloads between 4000 and 6000 kg. The total count of both successful and failed mission outcomes was calculated. Moreover, the booster versions that transported the highest payload mass were identified. The analysis further included retrieving details of failed landings on drone ships in 2015, specifying their booster versions and launch site names. Lastly, the ranking of landing outcomes, such as failures on drone ships and successes on ground pads, was determined for the period between June 4, 2010, and March 20, 2017.

  https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium Maps were enhanced with markers, circles, lines, and marker clusters to visualize key locations and data points. Markers represented specific points such as launch sites, while circles highlighted areas around particular coordinates, such as the NASA Johnson Space Center. Marker clusters grouped events occurring at the same location, like multiple launches at a single site. Additionally, lines were used to illustrate distances between two coordinates.

https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Predictive Analysis (Classification)

- The data was loaded using NumPy and Pandas, then transformed and split into training and testing sets. Various machine learning models were built, and hyperparameters were optimized using GridSearchCV. Model performance was evaluated based on accuracy, and enhancements were made through feature engineering and algorithm tuning. Ultimately, the best-performing classification model was identified.

https://github.com/ziadq7/IBM-Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

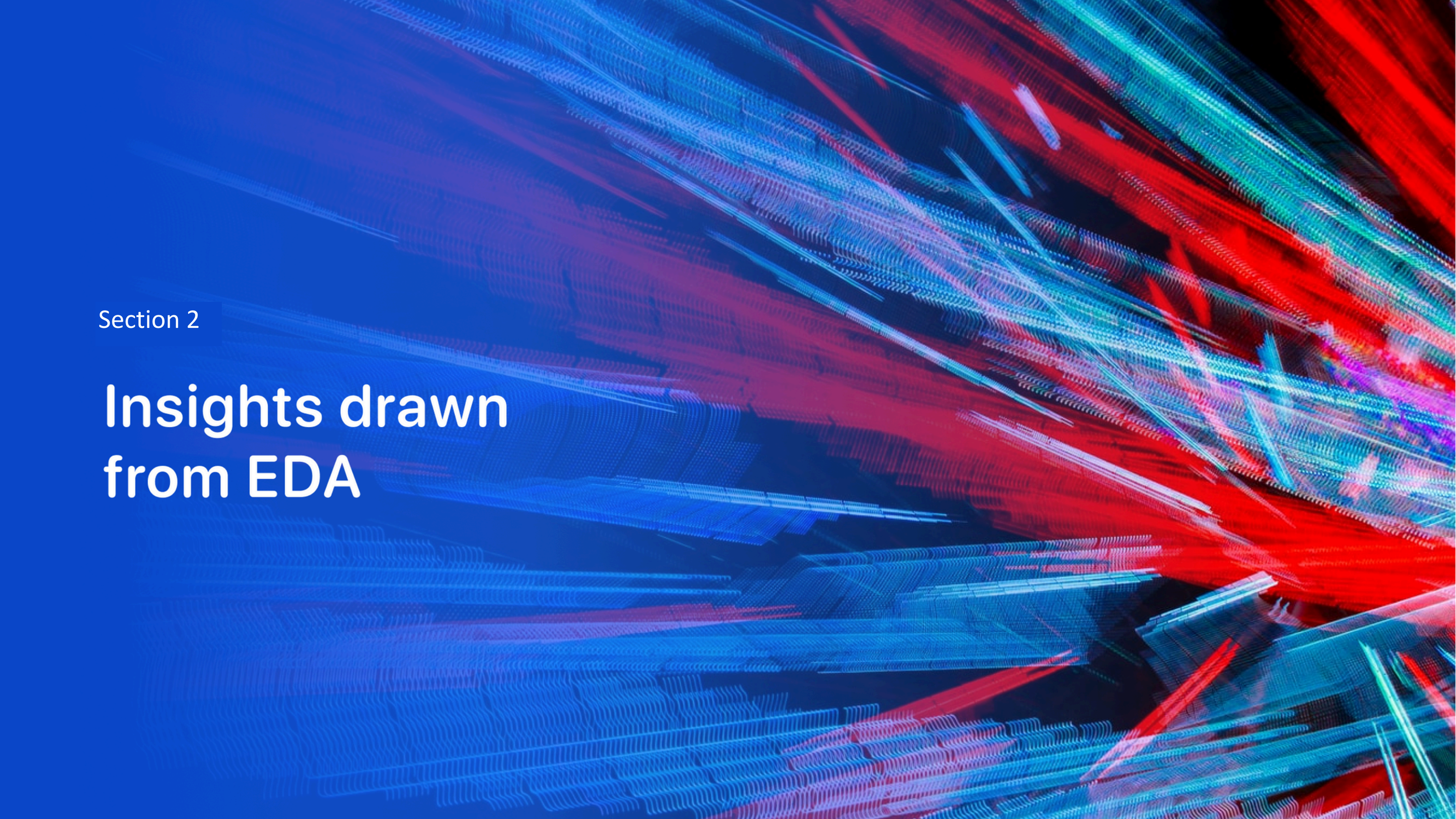
- Exploratory data analysis results:
  The exploratory data analysis revealed that SpaceX operates four distinct launch sites. Initially, the launches were conducted for SpaceX and NASA. The average payload capacity of the F9 v1.1 booster was found to be 2,928 kg. The first successful landing occurred in 2015, five years after the inaugural launch. Several Falcon 9 booster versions successfully landed on drone ships while carrying payloads exceeding the average. Nearly all mission outcomes were successful. However, two booster versions, F9 v1.1 B1012 and F9 v1.1 B1015, experienced failed landings on drone ships in 2015. Over time, the frequency of successful landings significantly improved.
- Interactive analytics demo in screenshots
- Predictive analysis results

Results

Interactive analytics revealed that launch sites are typically located in safe areas, often near the sea, and are supported by well-developed logistical infrastructure. Additionally, most launches take place at East Coast launch sites.

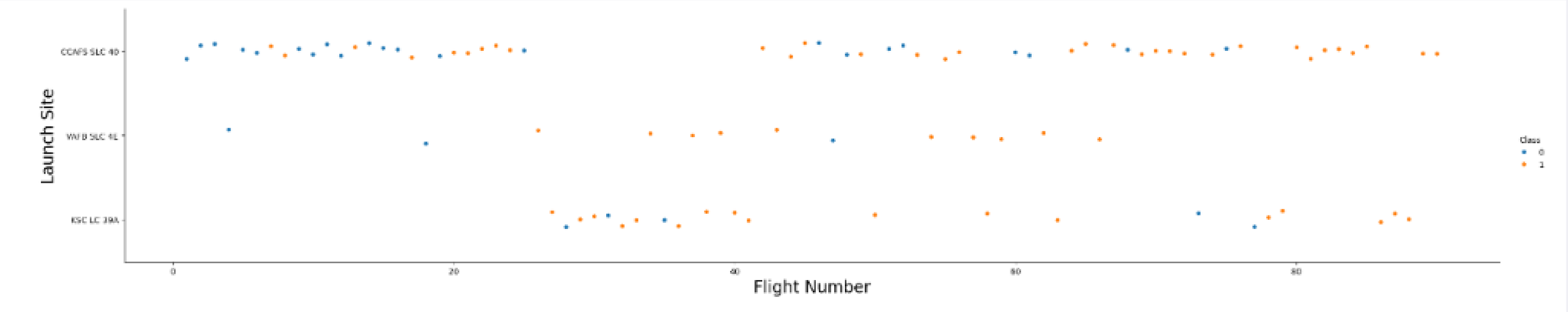
Section 2

Insights drawn from EDA



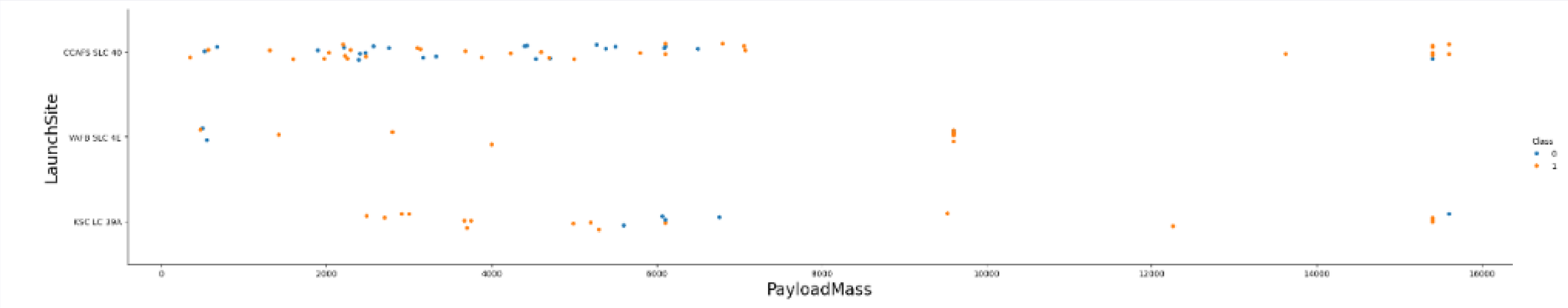
Flight Number vs. Launch Site

- The analysis showed that launch sites with a higher number of flights tend to have a greater success rate.



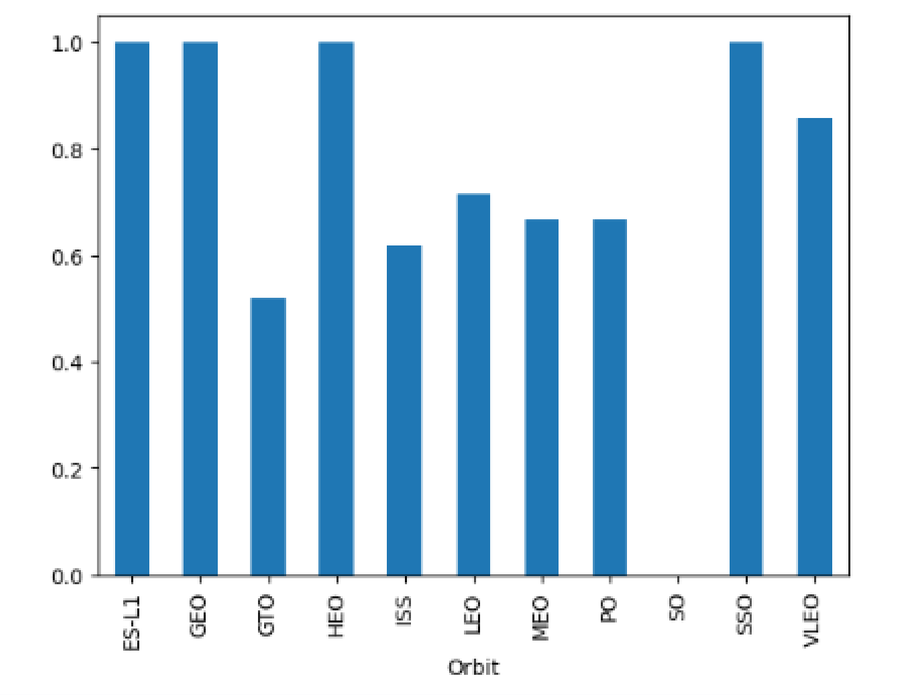
Payload vs. Launch Site

- The scatter plot shows the relationship between payload mass and launch sites, with mission outcomes indicated by color. It highlights variations in success rates across different payload ranges and launch locations.



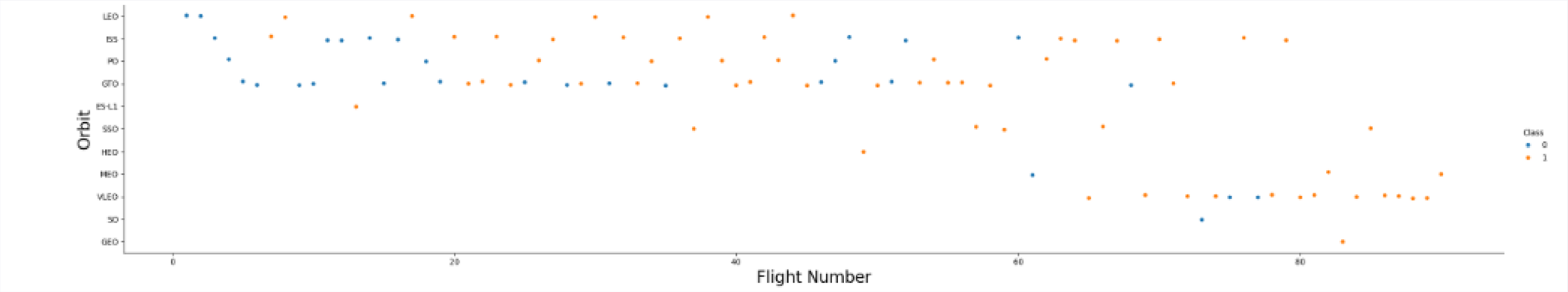
Success Rate vs. Orbit Type

- The bar chart visualizes the success rate for different orbit types by calculating the mean success rate for each. It helps identify which orbits have the highest and lowest success rates.



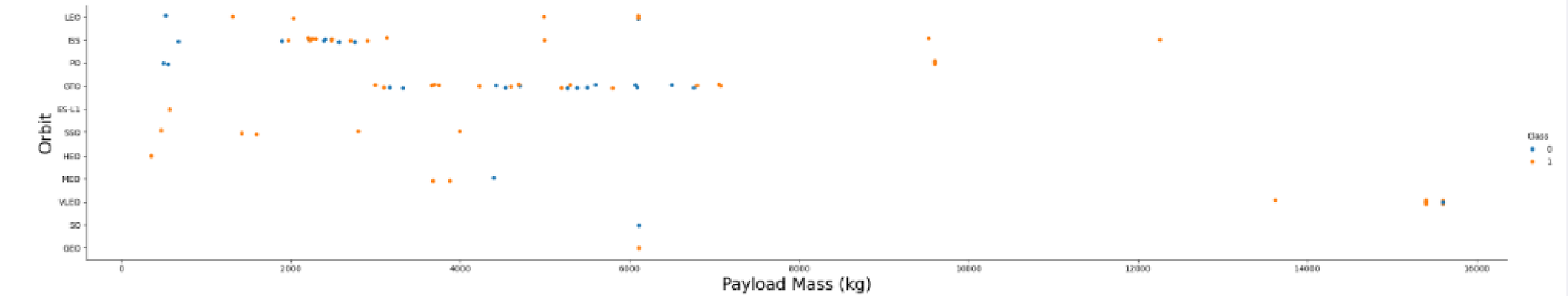
Flight Number vs. Orbit Type

- The scatter plot illustrates the relationship between flight number and orbit type, with mission success and failure represented by different colors. It helps analyze how success rates vary across different orbits and flight numbers.



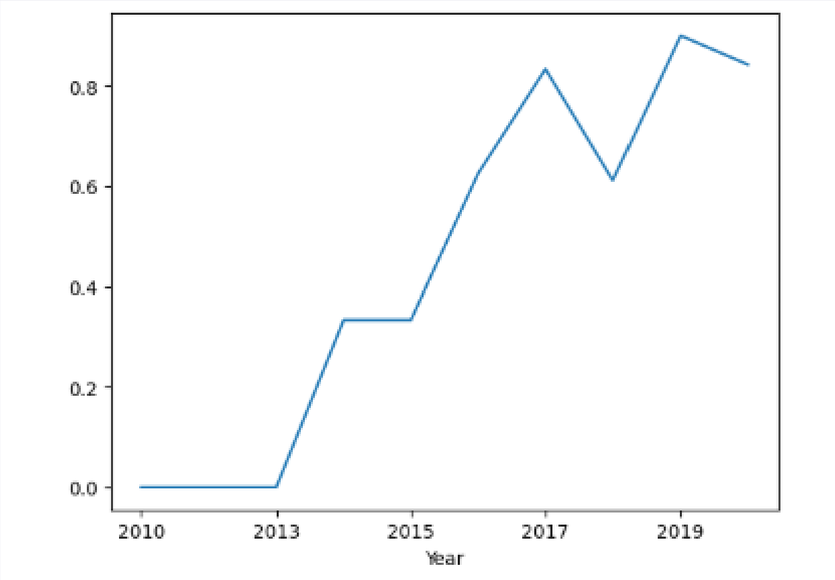
Payload vs. Orbit Type

- The scatter plot displays the relationship between payload mass and orbit type, with mission outcomes represented by different colors. It helps identify how payload mass influences success rates across various orbits.



Launch Success Yearly Trend

- The line chart illustrates the yearly trend of launch success rates, with the x-axis representing the year and the y-axis showing the average success rate. It highlights an overall improvement in launch success over time.



All Launch Site Names

- %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- %sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5;

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)'

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
[18]: %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

[18]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%';

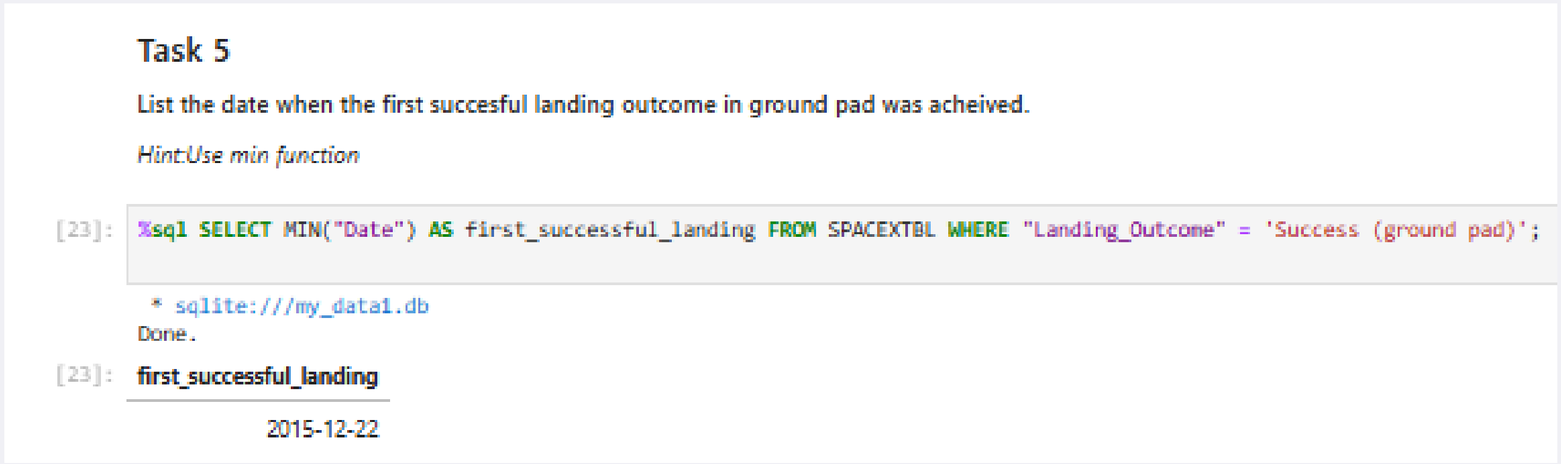
Task 4

Display average payload mass carried by booster version F9 v1.1

```
[19]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%';
 * sqlite:///my_data1.db
Done.
[19]: average_payload_mass
      2534.6666666666665
```

First Successful Ground Landing Date

- %sql SELECT MIN("Date") AS first_successful_landing FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)';



Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
[23]: %sql SELECT MIN("Date") AS first_successful_landing FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)';
```

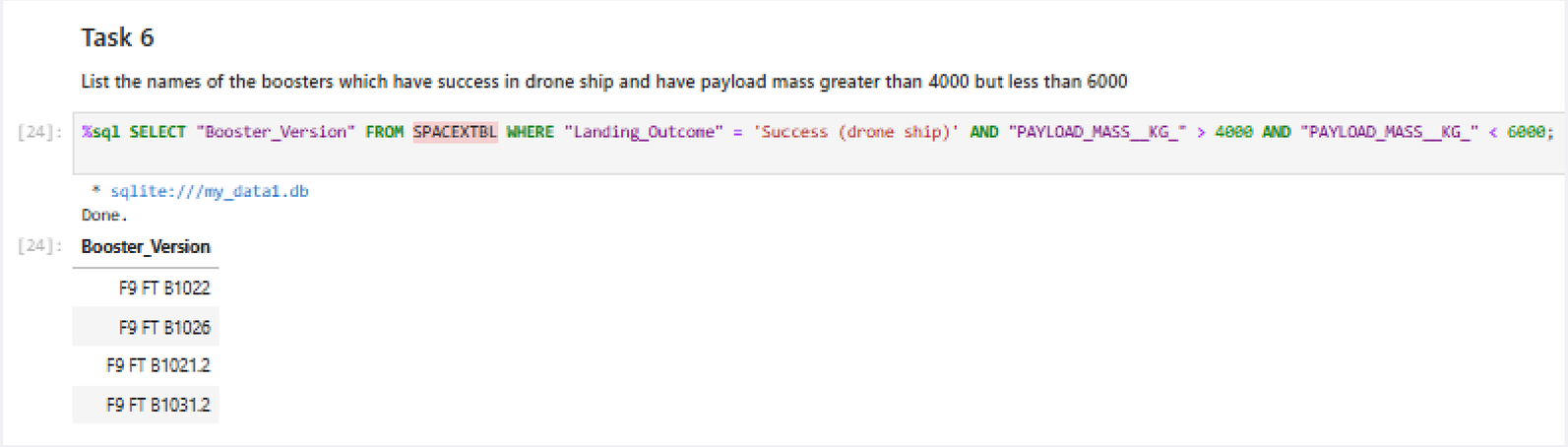
```
 * sqlite:///my_data1.db
Done.
```

[23]:

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;



Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

[24]:
```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

[24]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- %sql SELECT "Mission_Outcome", COUNT(*) AS total_count FROM SPACEXTBL GROUP BY "Mission_Outcome";

Task 7

List the total number of successful and failure mission outcomes

```
[25]: %sql SELECT "Mission_Outcome", COUNT(*) AS total_count FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

```
 * sqlite:///my_data1.db
Done.
```

[25]:

Mission_Outcome	total_count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[26]: %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

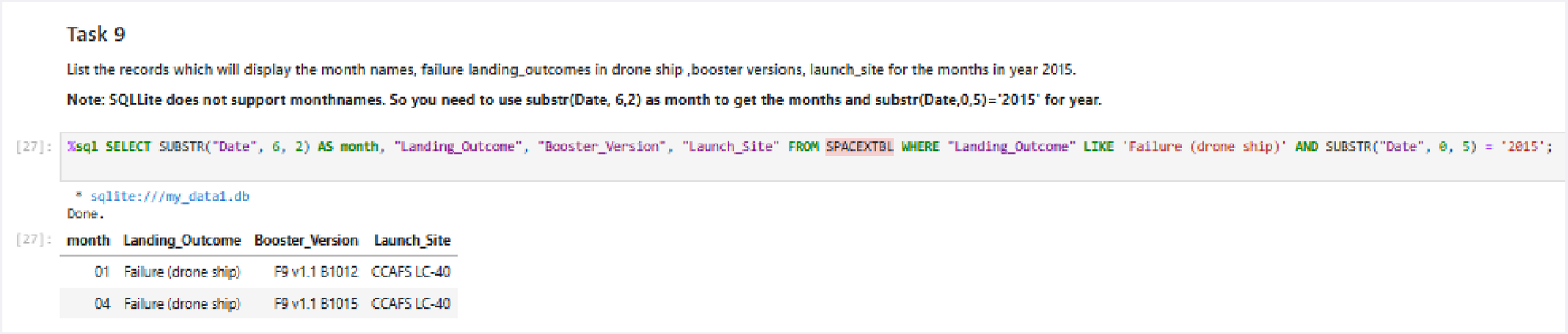
```
 * sqlite:///my_data1.db
Done.
```

[26]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- %sql SELECT SUBSTR("Date", 6, 2) AS month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Failure (drone ship)' AND SUBSTR("Date", 0, 5) = '2015';



Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[27]:
```
%sql SELECT SUBSTR("Date", 6, 2) AS month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Failure (drone ship)' AND SUBSTR("Date", 0, 5) = '2015';
```

```
 * sqlite:///my_data1.db
Done.
```

[27]:

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- %sql SELECT "Landing_Outcome", COUNT(*) AS outcome_count FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY outcome_count DESC;

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
[28]: %sql SELECT "Landing_Outcome", COUNT(*) AS outcome_count FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY outcome_count DESC;

 * sqlite:///my_data1.db
Done.
```

[28]:

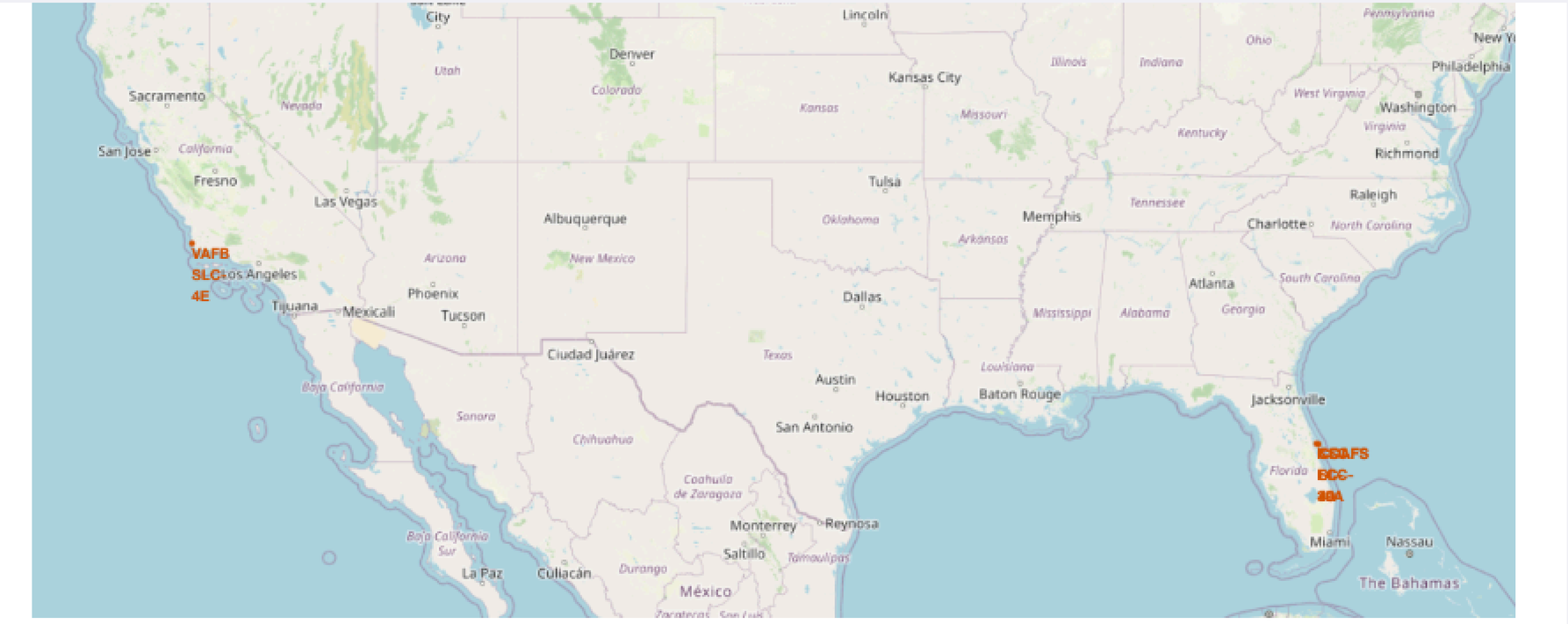
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All Launch Sites Global Map Markers

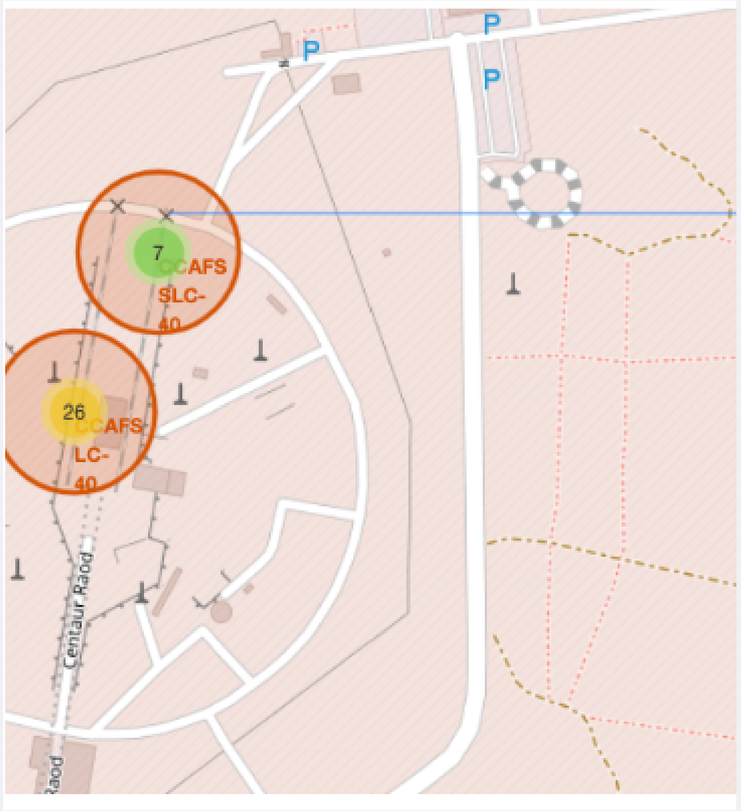
- SpaceX launch sites are located along the coasts of the United States, specifically in Florida and California.



Markers Showing Launch Sites With Color Labels

Launch Site Distance to Landmarks

- Distance to Coast



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree model has the highest classification accuracy.

TASK 12

Find the method performs best:

In [35]:

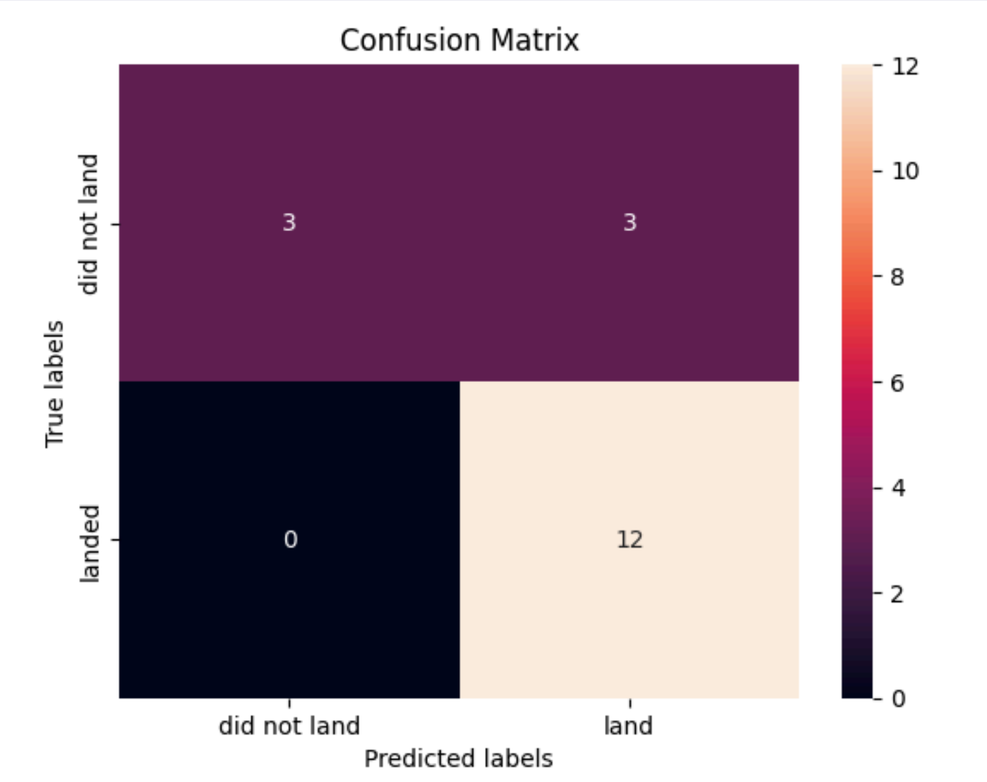
```
algo_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tree': [tree_cv.best
df = pd.DataFrame.from_dict(algo_score, orient='index', columns=['Best scores'])
df
```

Out[35]:

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.891071
KNN	0.848214

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates that it can differentiate between various classes. However, a significant issue is the presence of false positives, where the model incorrectly predicts unsuccessful landings as successful ones.



Conclusions

- A higher number of flights at a launch site is associated with a greater success rate.
- The launch success rate showed an upward trend from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO achieved the highest success rates.
- KSC LC-39A recorded the highest number of successful launches among all sites.
- The decision tree classifier proved to be the most effective machine learning algorithm for this task.

Thank you!