IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bing Han
07/21/2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

In the burgeoning commercial space era, SpaceX stands out for its cost-effective, reusable Falcon 9 rocket launches. This project involves analyzing SpaceX's launch data to predict whether the first stage of the Falcon 9 will land successfully. By leveraging machine learning models and public information, we aim to determine launch outcomes and potentially reduce costs for our hypothetical competitor, Space Y. Our goal is to create dashboards for launch cost estimation and first stage reuse prediction, providing insights into SpaceX's operational efficiency and enhancing our understanding of rocket launch dynamics.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - Extract a Falcon 9 launch records HTML table from Wikipedia
- Perform data wrangling
  - Using Pandas to perform initial exploration and cleaning of the data, and to adjust column attributes to a usable state.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data was collected from two sources:

- Request from the SpaceX API

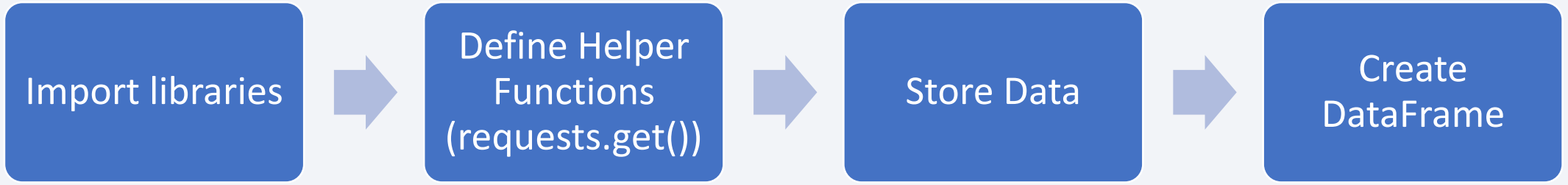
Import libraries-> Define Helper Functions (Call API for Detailed Information with requests.get() function) -> Store Data -> Create DataFrame

- Extract from a Falcon 9 launch records HTML table from Wikipedia

Import libraries-> Fetch HTML -> Parse HTML (BeautifulSoup) -> Define Helper Functions -> Extract and Clean Data -> Populate Dictionary -> Create DataFrame

Data Collection – SpaceX API

- Key phrases and flowcharts:



- GitHub URL: IBM-Applied-Data-Science-Capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

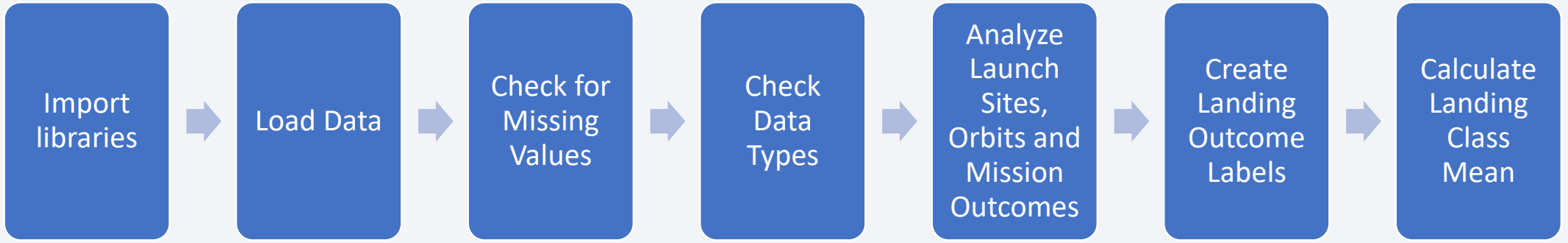
Data Collection - Scraping

- Key phrases and flowcharts:

- GitHub URL: IBM-Applied-Data-Science-Capstone/jupyter-labs-webscraping.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

Data Wrangling

- Key phrases and flowcharts:



- GitHub URL: IBM-Applied-Data-Science-Capstone/labs-jupyter-spacex-Data wrangling.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

EDA with Data Visualization

- Plots Summarize:

  •**Payload Mass vs. Flight Number (Catplot)** To examine the relationship between payload mass and flight number, differentiating by mission success (Class).
  •**Launch Site vs. Flight Number (Catplot)** To analyze the relationship between launch site and flight number, differentiating by mission success (Class).
  •**Launch Site vs. Payload Mass (Catplot)** To explore the relationship between payload mass and launch site, differentiating by mission success (Class).
  •**Mean Success Rate by Orbit (Barplot)** To display the average success rate for different orbits.
  •**Flight Number vs. Orbit (Catplot)** To analyze the relationship between flight number and orbit, differentiating by mission success (Class).
  •**Payload Mass vs. Orbit (Catplot)** To examine the relationship between payload mass and orbit, differentiating by mission success (Class).
  •**Yearly Success Rate (Lineplot)** To show the trend in mission success rates over different years.

- GitHub URL: IBM-Applied-Data-Science-Capstone/edadataviz.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

EDA with SQL

- SQL queries:

  Display the names of the unique launch sites in the space mission、

  Display 5 records where launch sites begin with the string 'CCA'

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

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

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

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

  List the total number of successful and failure mission outcomes

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

  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.

  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.

- GitHub URL: IBM-Applied-Data-Science-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

Build an Interactive Map with Folium

Map Objects Created and Their Purposes:

1.Circles:

1. **Blue circles for NASA Johnson Space Center and launch sites**
2. **Purpose: To visually highlight key locations on the map**

2.Markers:

1. **Text markers for NASA JSC and launch sites**
2. **Colored markers (green/red) for successful/failed launches**
3. **Purpose: To label important locations and show launch outcomes**

3.MarkerCluster:

1. **Grouping of launch site markers**
2. **Purpose: To improve map readability when zoomed out**

4.MousePosition:

1. **Shows latitude and longitude on mouse hover**
2. **Purpose: To allow users to easily identify coordinates on the map**

5.PolyLines:

1. **Connecting launch sites to nearby points of interest**
2. **Purpose: To visualize distances between launch sites and key locations**

6.Distance Markers:

1. **Text markers showing distances to nearby features**
2. **Purpose: To display calculated distances on the map**

- GitHub URL: IBM-Applied-Data-Science-Capstone/lab_jupyter_launch_site_location.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

Build a Dashboard with Plotly Dash

Plots/Graphs and Interactions in the SpaceX Launch Dashboard:

1.Dropdown Menu:
    1. Allows selection of specific launch sites or all sites
    2. Purpose: Enables users to filter data by launch site

2.Pie Chart:
    1. Shows success rate of launches
    2. Dynamically updates based on selected site
    3. Purpose: Visualizes success/failure ratio for each site or overall

3.Range Slider:
    1. Controls payload mass range
    2. Purpose: Allows users to filter launches by payload capacity

4.Scatter Plot:
    1. Displays correlation between payload mass and launch success
    2. Color-coded by booster version
    3. Updates based on site selection and payload range
    4. Purpose: Illustrates relationship between payload and success rate

- GitHub URL: IBM-Applied-Data-Science-Capstone/spacex_dash_app.py at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

Predictive Analysis (Classification)

Model Development Process:

1.Data Preparation:

1. Loaded and preprocessed data
2. Standardized features
3. Split data into training and test sets (80/20)

2.Model Building and Evaluation:

1. Implemented four classification algorithms: a) Logistic Regression b) Support Vector Machine (SVM) c) Decision Tree d) K-Nearest Neighbors (KNN)
2. Used GridSearchCV for hyperparameter tuning (10-fold cross-validation)
3. Evaluated models using accuracy scores and confusion matrices

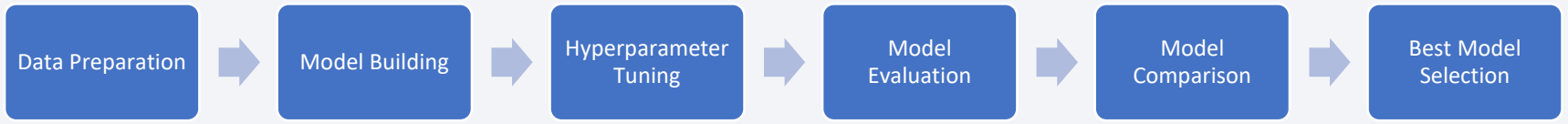
3.Model Improvement:

1. Fine-tuned hyperparameters for each algorithm
2. Compared performance across models

4.Best Performing Model:

1. Selected based on highest accuracy on test data

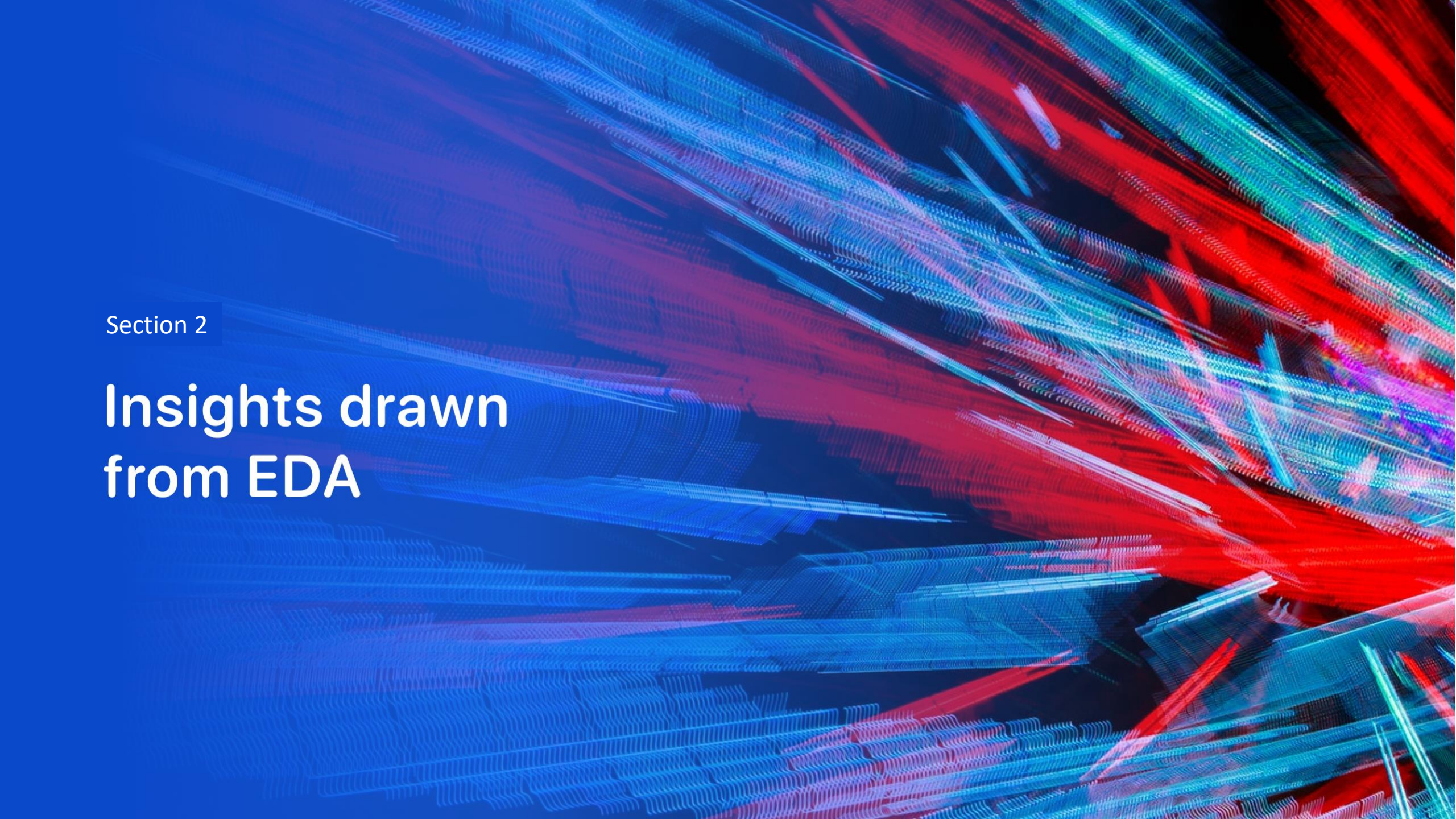
Flowchart:



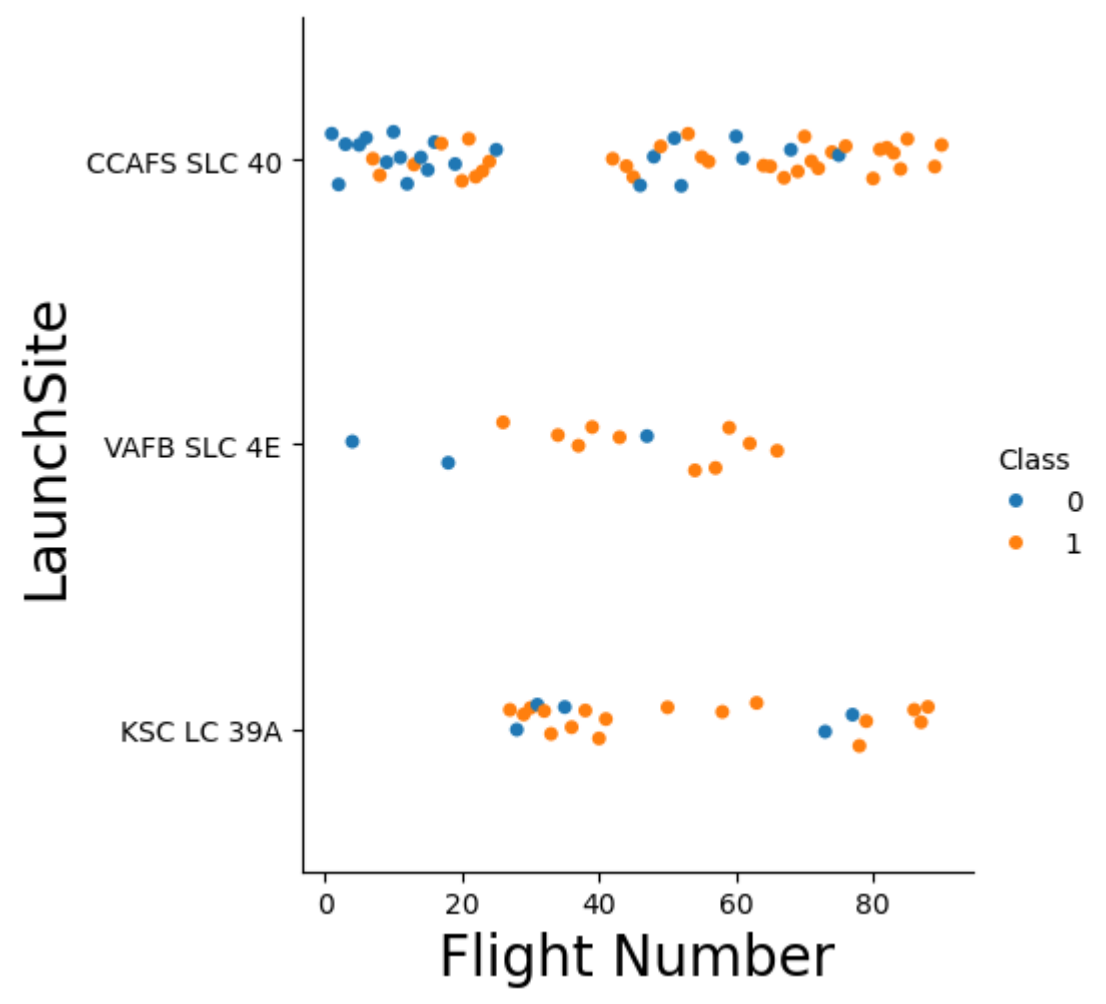
- GitHub URL: IBM-Applied-Data-Science-Capstone/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · teaninja/IBM-Applied-Data-Science-Capstone (github.com)

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

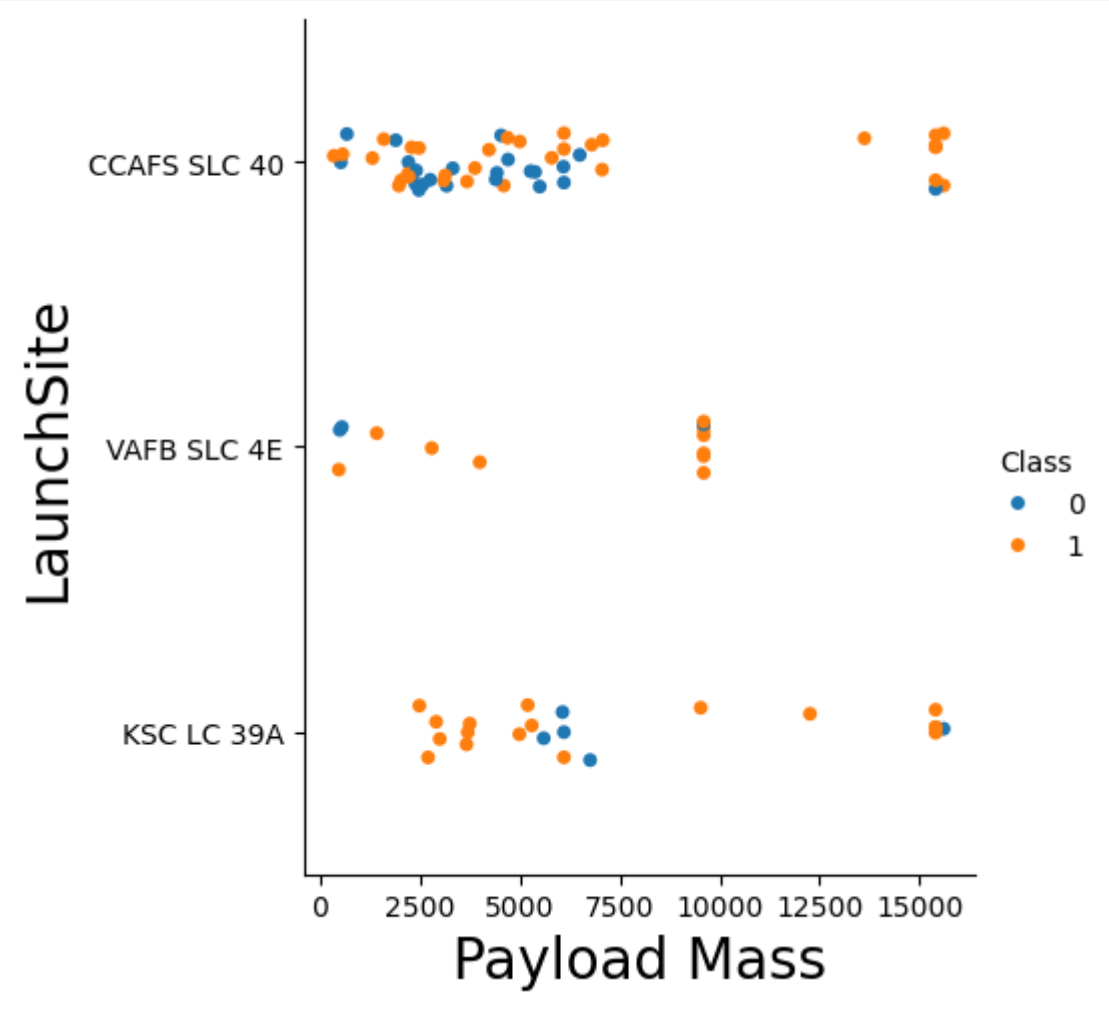

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



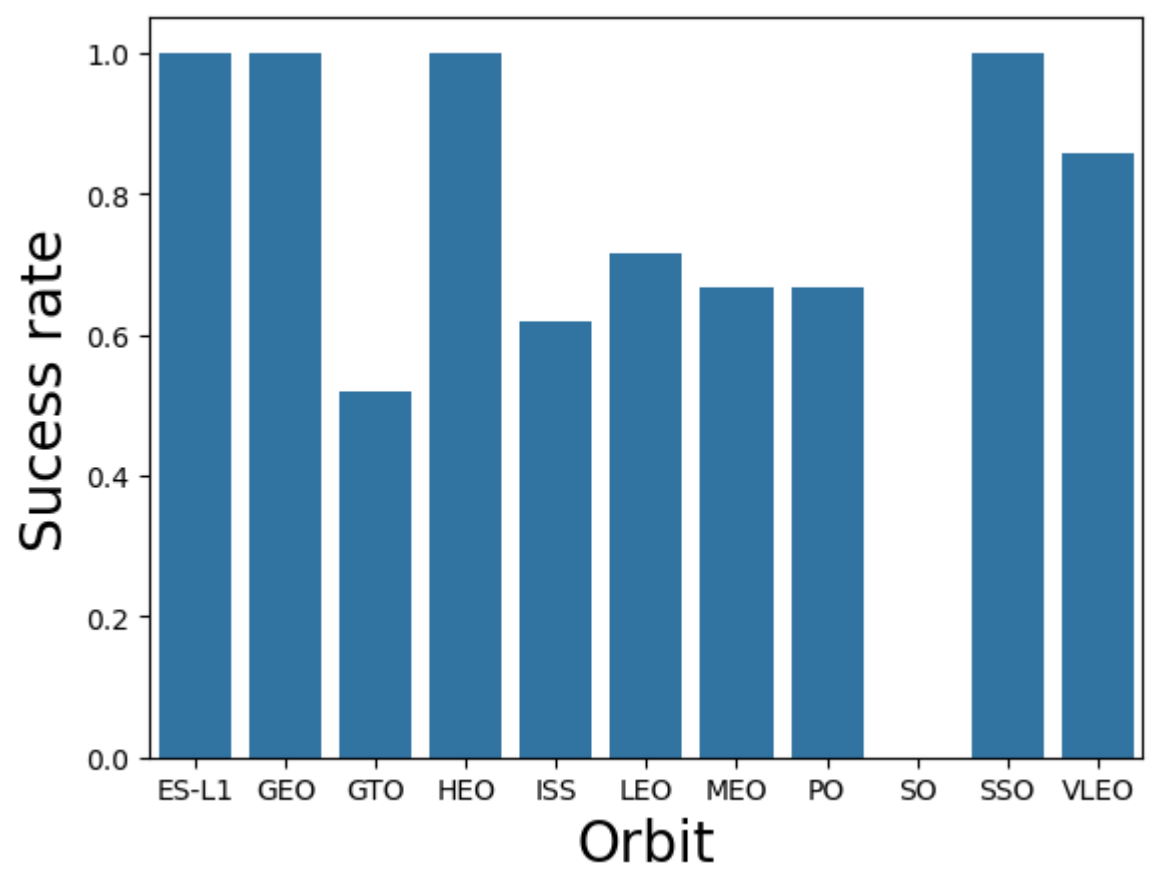
- As the flight number increases, the first stage is more likely to land successfully.
- And it seems true for all launch sites

Payload vs. Launch Site



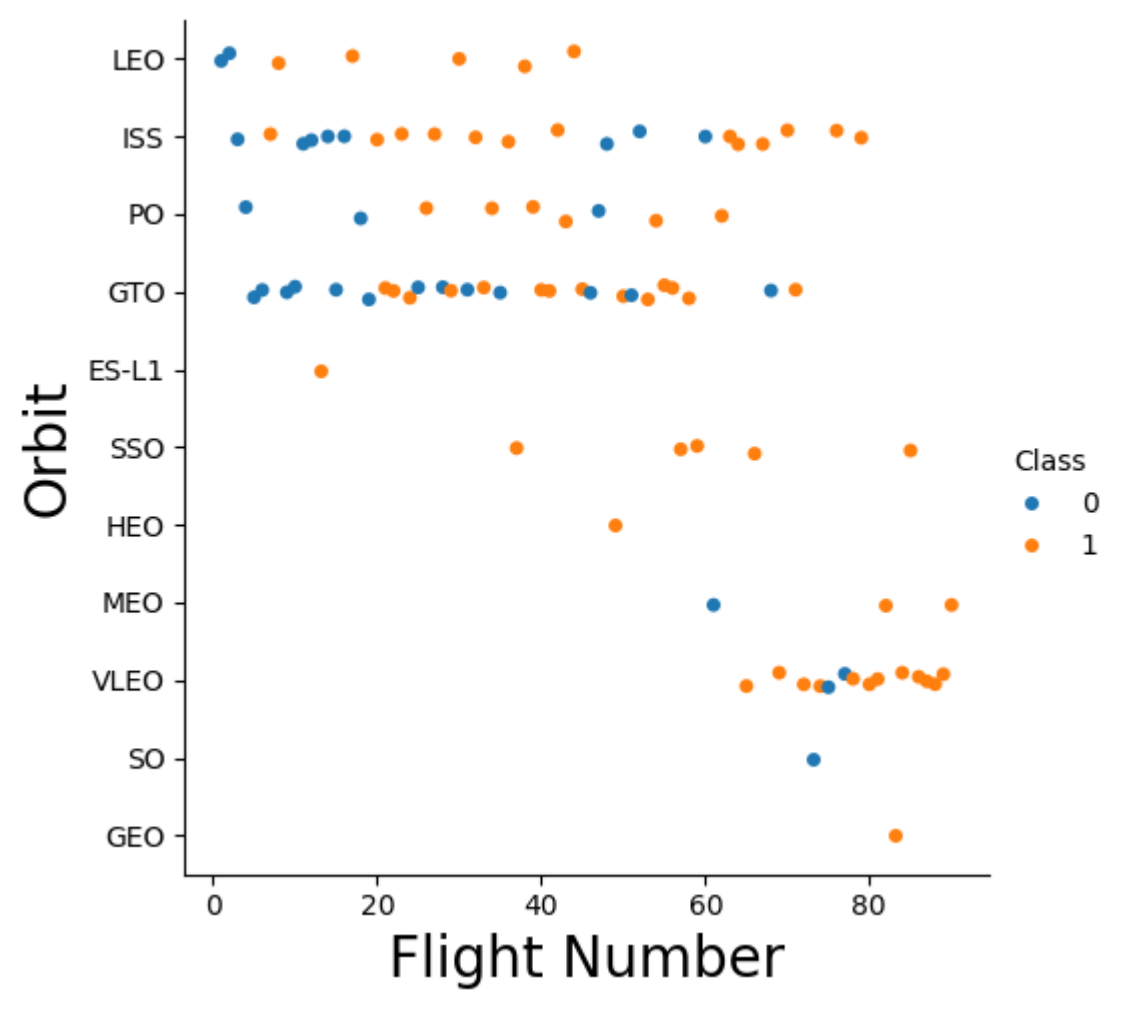
- As the Payload Mass increases, the first stage is more likely to land successfully.
- And it seems true for all launch sites

Success Rate vs. Orbit Type



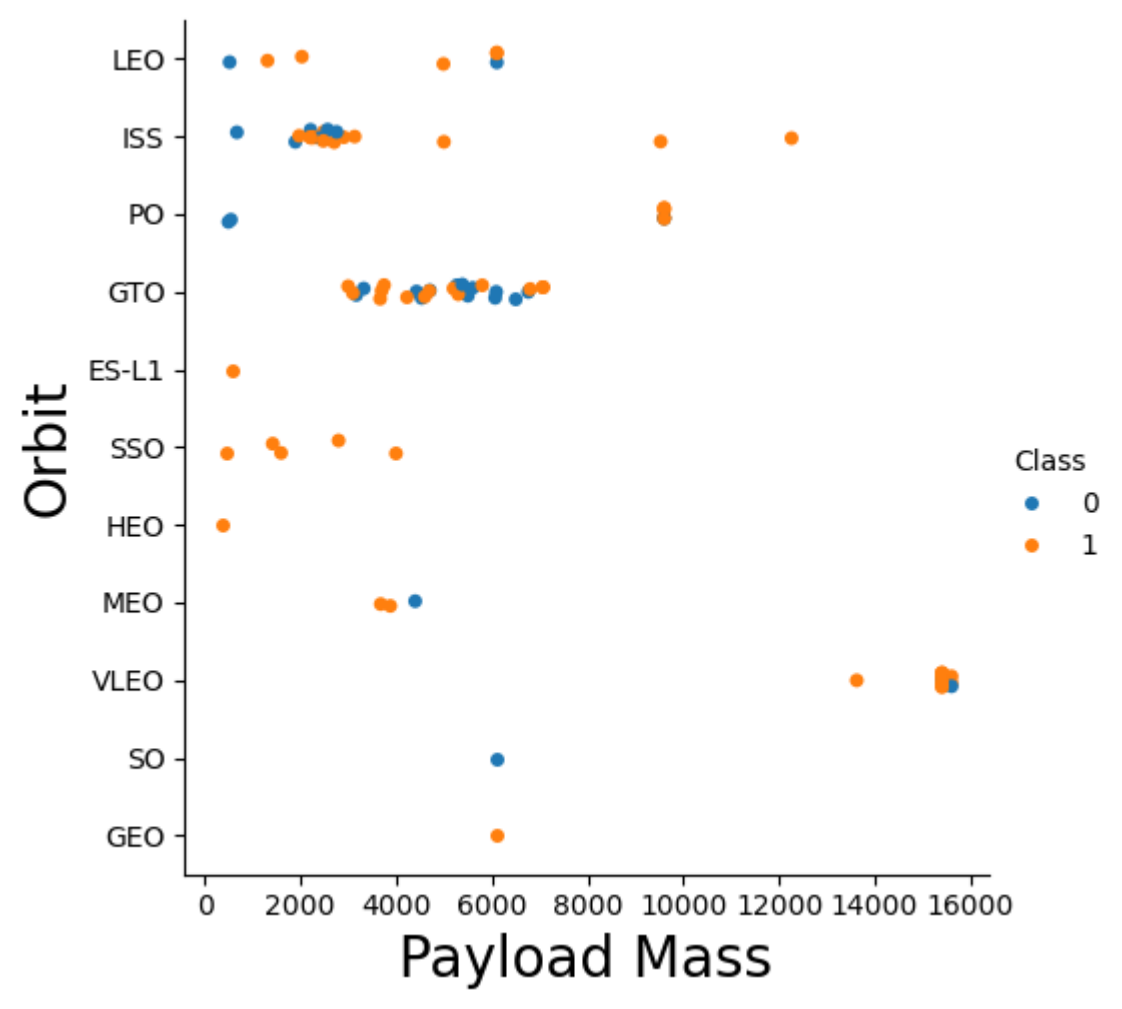
- There seems a correlation between orbit type and success rate.
- ES-L1, GEO, HEO and SSO have the highest success rate, while GTO has the lowest success rate.

Flight Number vs. Orbit Type



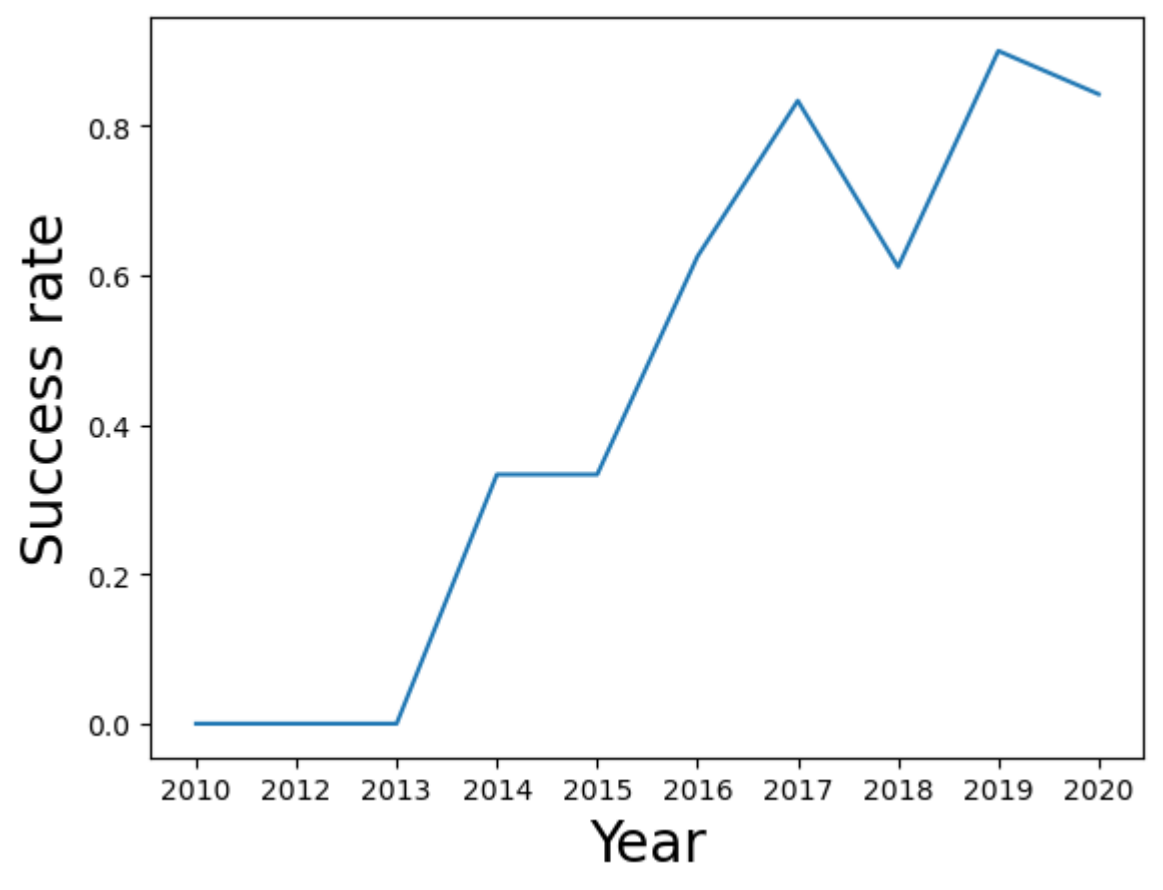
- The distribution of flight numbers for LEO, ISS, PO and GTO is pretty average.
- The overall flight numbers of VLEO is high.
- All other orbit types don't have much data yet.

Payload vs. Orbit Type



- VLEO has the highest average payload mass.
- As the Payload Mass increases, the first stage is more likely to land successfully.

Launch Success Yearly Trend



- The launch success rate has significantly improved over time.
- The launch success rate slightly decreased in 2018.

""

"The launch success rate slightly decreased in 2018."

All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
%sql select distinct Launch_Site from SPACEXTBL
```

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

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

- There are four unique launch sites in this dataset.
- “distinct” was the critical keywords to display the unique launch sites.

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

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

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

- "%" was used as wildcard after "CCA" in "like" clause in this query.
- "limit 5" clause defined how many records were queried.

Total Payload Mass

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer=='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

- The function of sum() was used to calculate the total payload mass.
- “where” clause was used to restrict the object being calculated.

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version=='F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

- The function of avg() was used to calculate the average of payload mass.
- “where” clause was used to restrict the object being calculated.

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome=='Success (ground pad)'
```

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

min(Date)
2015-12-22

- The function of min() was used to find the first date of successful landing.
- “where” clause was used to restrict the outcome as ‘Success’ (ground pad).

Successful Drone Ship Landing with Payload between 4000 and 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

```
%%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.
```

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

- “and” logic was used when multiple filtering conditions were set up.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select Landing_Outcome, count(Landing_Outcome) as count from SPACEXTBL group by Landing_Outcome
```

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

Landing_Outcome	count
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

- "count()" function and "group by" clause in was critical in this query.

Boosters Carried Maximum Payload

Task 8

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

```
%%sql
    select Booster_Version
    from SPACEXTBL
    where PAYLOAD_MASS__KG_ in (select PAYLOAD_MASS__KG_ from SPACEXTBL order by PAYLOAD_MASS__KG_ desc limit 1)
```

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

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

- A nested subquery was carried out first to get the number of the maximum payload mass.
- Then “in” logic was used to filter out the booster versions which have the maximum payload mass number.

2015 Launch Records

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.

```
%%sql
    SELECT substr(Date, 6, 2) as Month, Landing_Outcome, Booster_Version, Launch_Site
    FROM SPACEXTBL
    WHERE substr(Date, 1, 4)='2015' AND Landing_Outcome='Failure (drone ship)'
```

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

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

- “substr()” function was used to handle the date information in this query.

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

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.

```
%%sql
SELECT Landing_Outcome, count(Landing_Outcome) AS count
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY count DESC
```

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

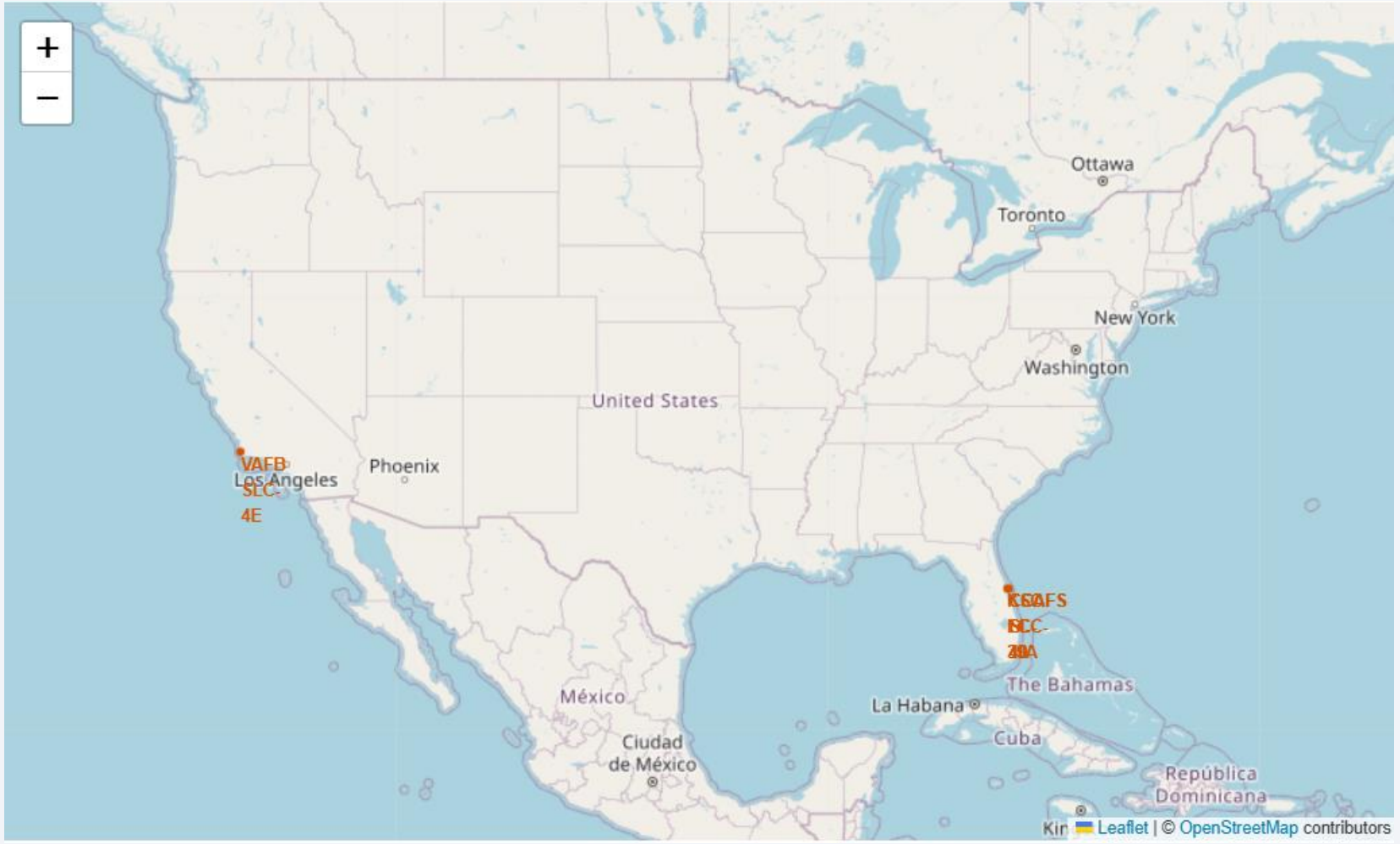
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- “between, and” clause was used to filter the time range.
- “order by, desc” clause was used to define the order of the displays of the outcomes of the query.

Section 3

Launch Sites Proximities Analysis

<Distribution of four spaceX launch site in the USA>

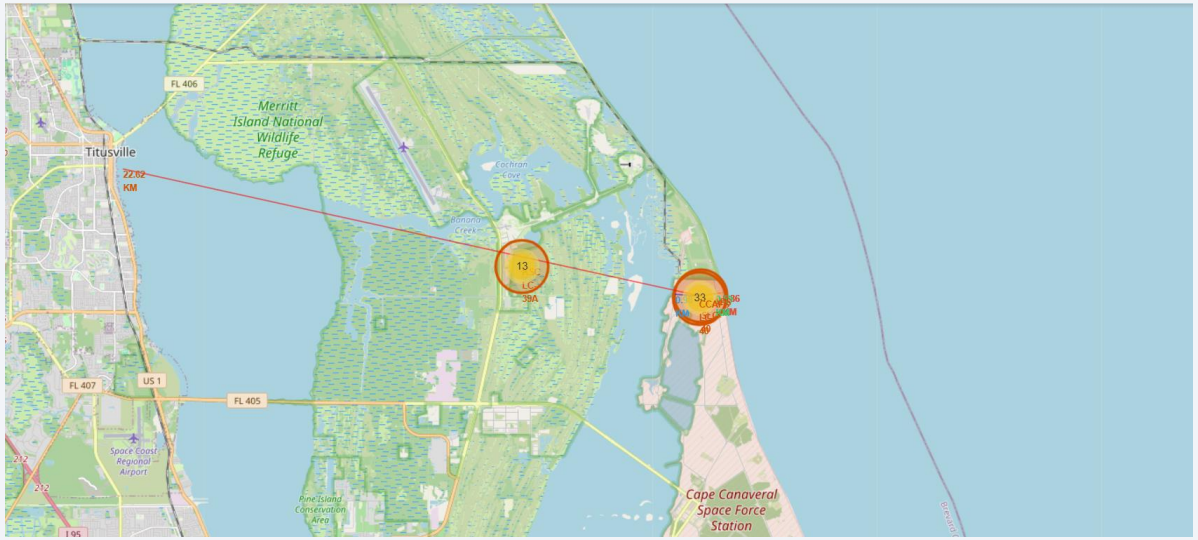


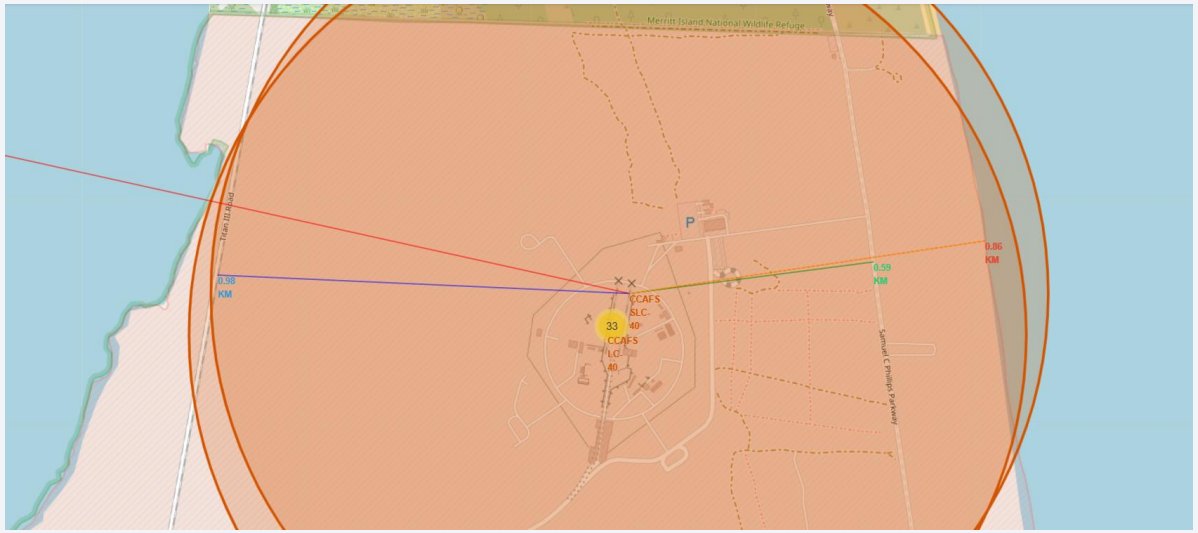
The name of four launch sites (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E) are labeled in red color.

< The display of successful and failed launches at CCAFS LC-40 launch site on the map. >

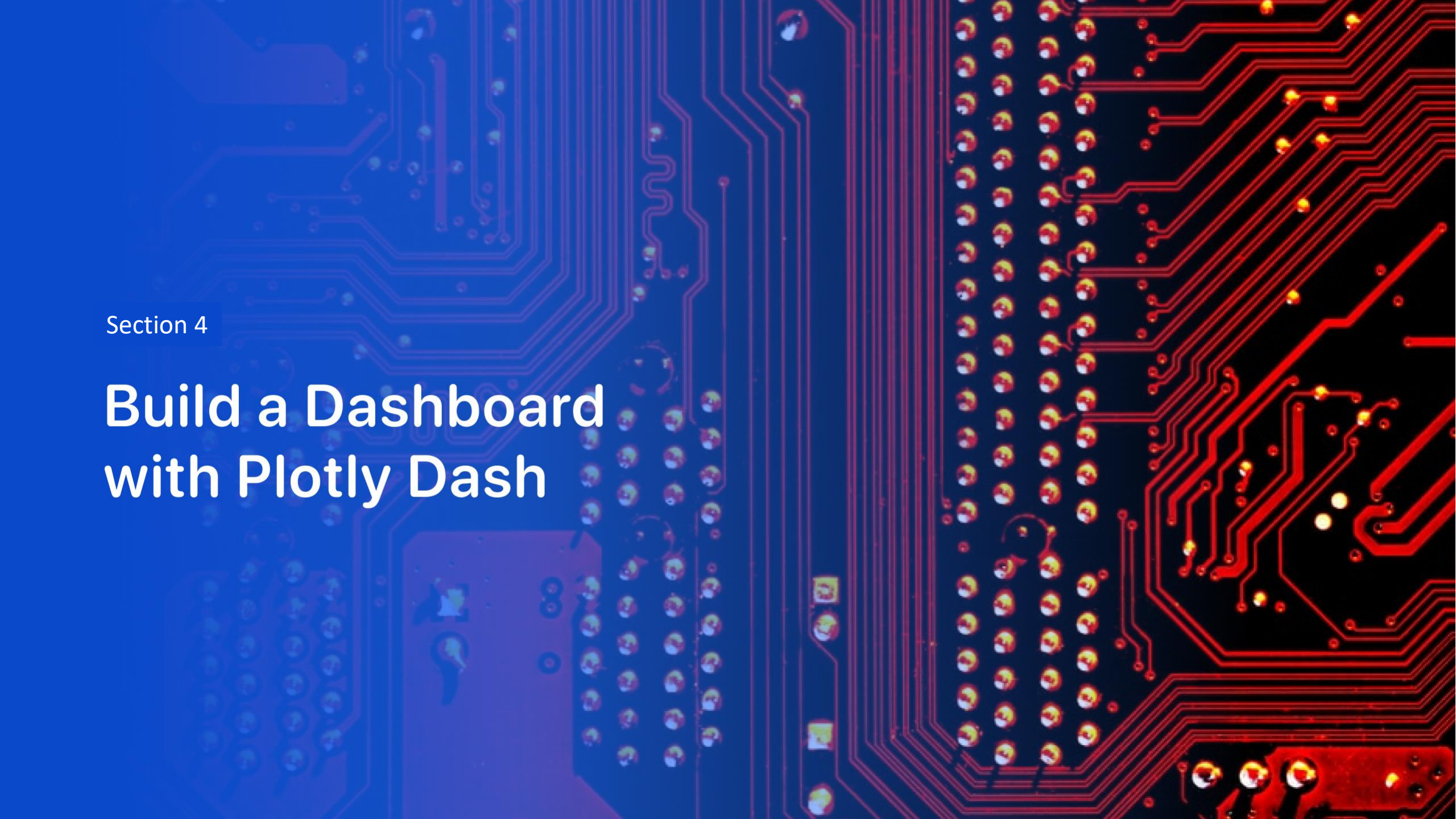
The MarkerCluster() function is used to cluster the markers of successful and failed launch events, making the map display spatially compact and visually appealing.

<Distance measurement and display from the CCAFS SLC-40 rocket launch site to the nearest city, coastline, railway, and highway.>

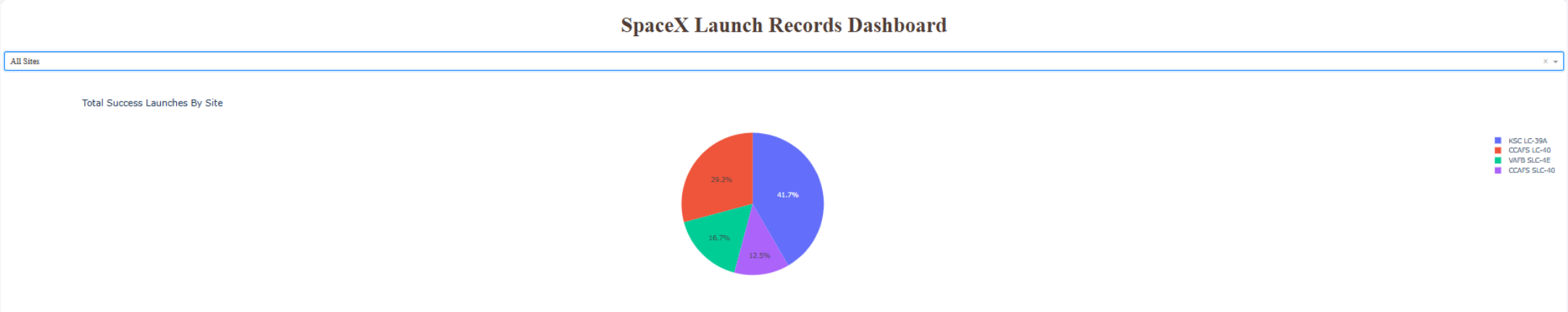




This map shows the distances and positional relationships from the CCAFS SLC-40 rocket launch site to the nearest city, coastline, railway, and highway. This allows stakeholders to visually assess whether there is sufficient safety distance from the city and if the launch site is adequately close to transportation facilities.

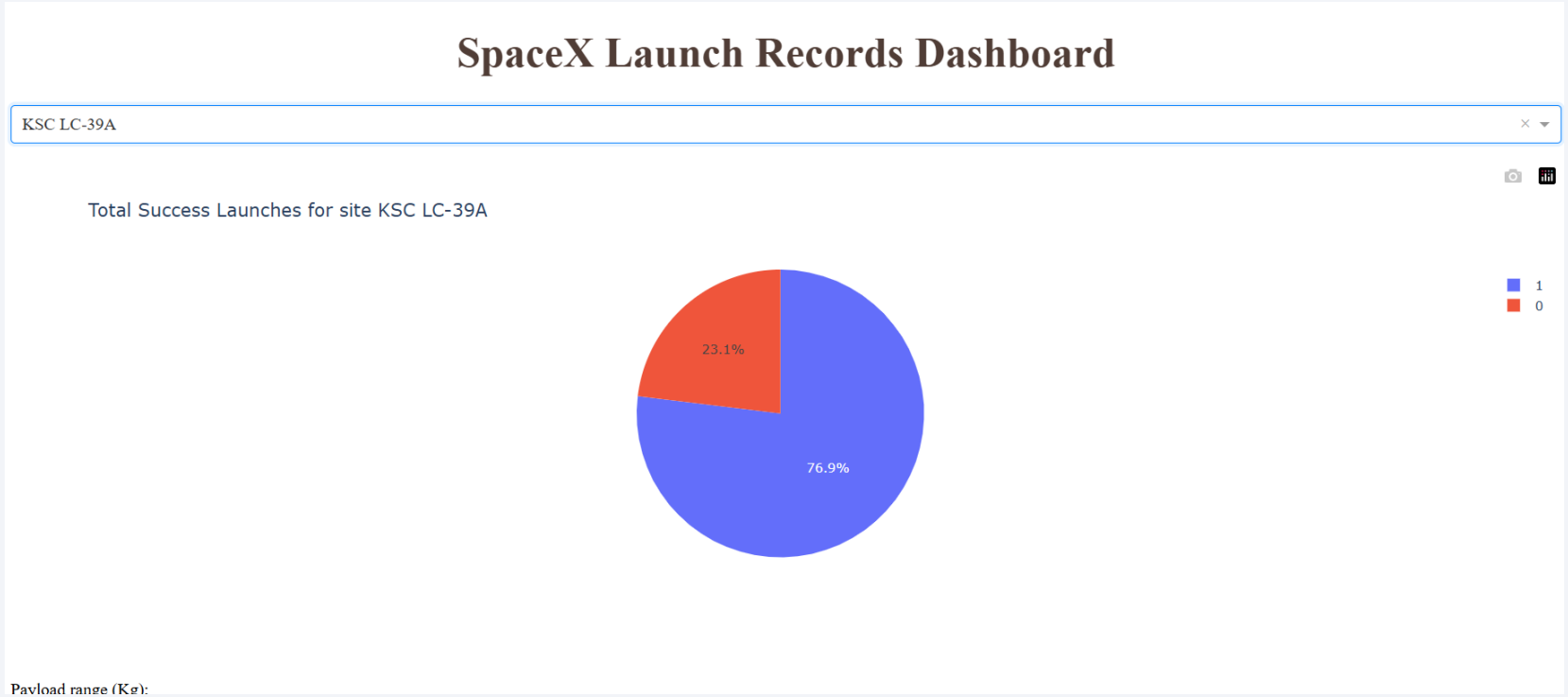

Section 4
Build a Dashboard with Plotly Dash

<The distribution of successful launches at the four launch sites.>



In this dashboard, we have set up a dropdown menu. By adjusting this dropdown menu, we can choose to view the distribution of successful launches across all launch sites or the ratio of successful and failed launches at a specific launch site. Specifically, what is displayed here is the distribution of all successful launches across all launch sites. The results shown by the pie chart indicate that the most successful launches were achieved from the KSC LC-39A launch site, while the fewest were from CCAFS SLC-40.

<Success rate display of the KSC LC-39A launch site>



As mentioned above, KSC LC-39A is the launch site with the most successful rocket launches. The results displayed on this page are obtained by selecting the dropdown menu to view the success rate of the KSC LC-39A site. As shown in the pie chart, the success rate of this launch site is 76.9%.

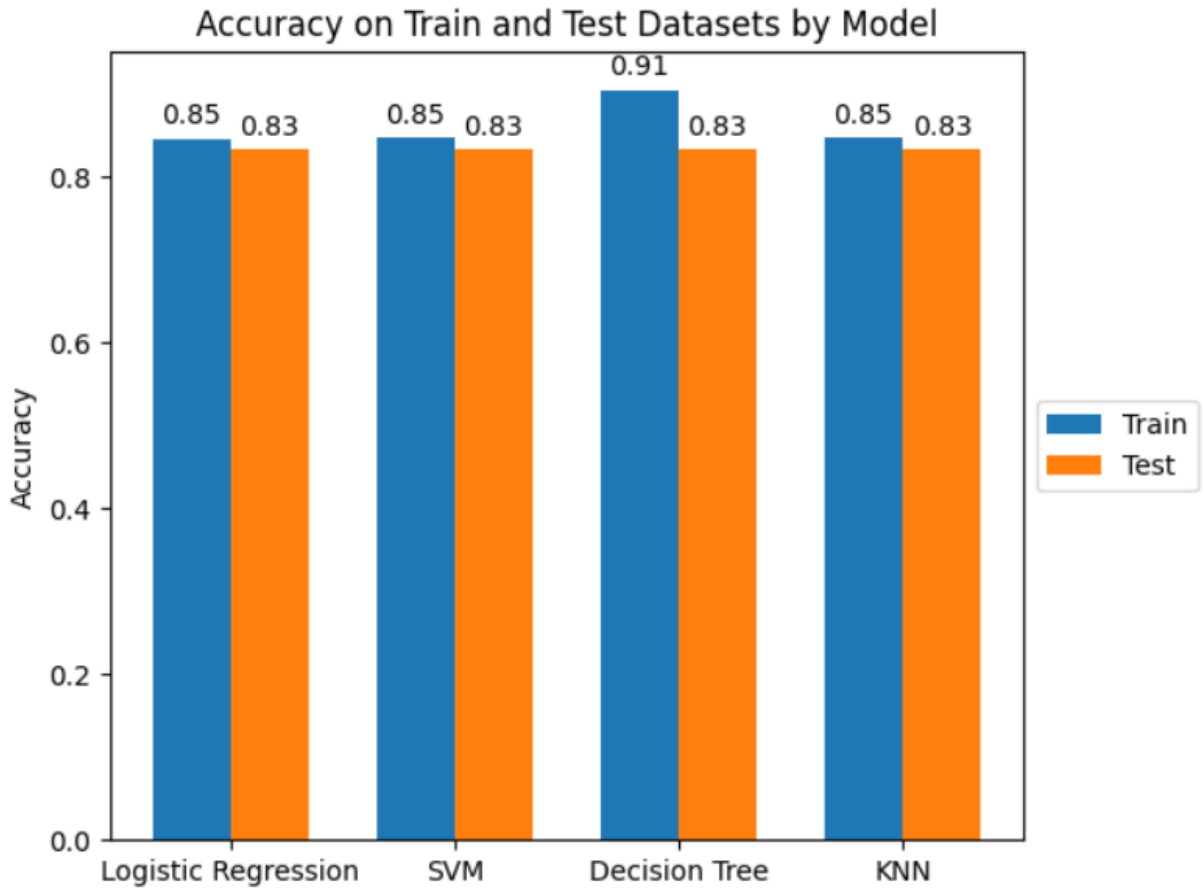
<Correlation between Payload and Success for all Sites>

By observing the scatter plot above, we find that payloads in the range of 6000 to 7000 and booster versions FT and B4 have the highest launch success rates.

Section 5

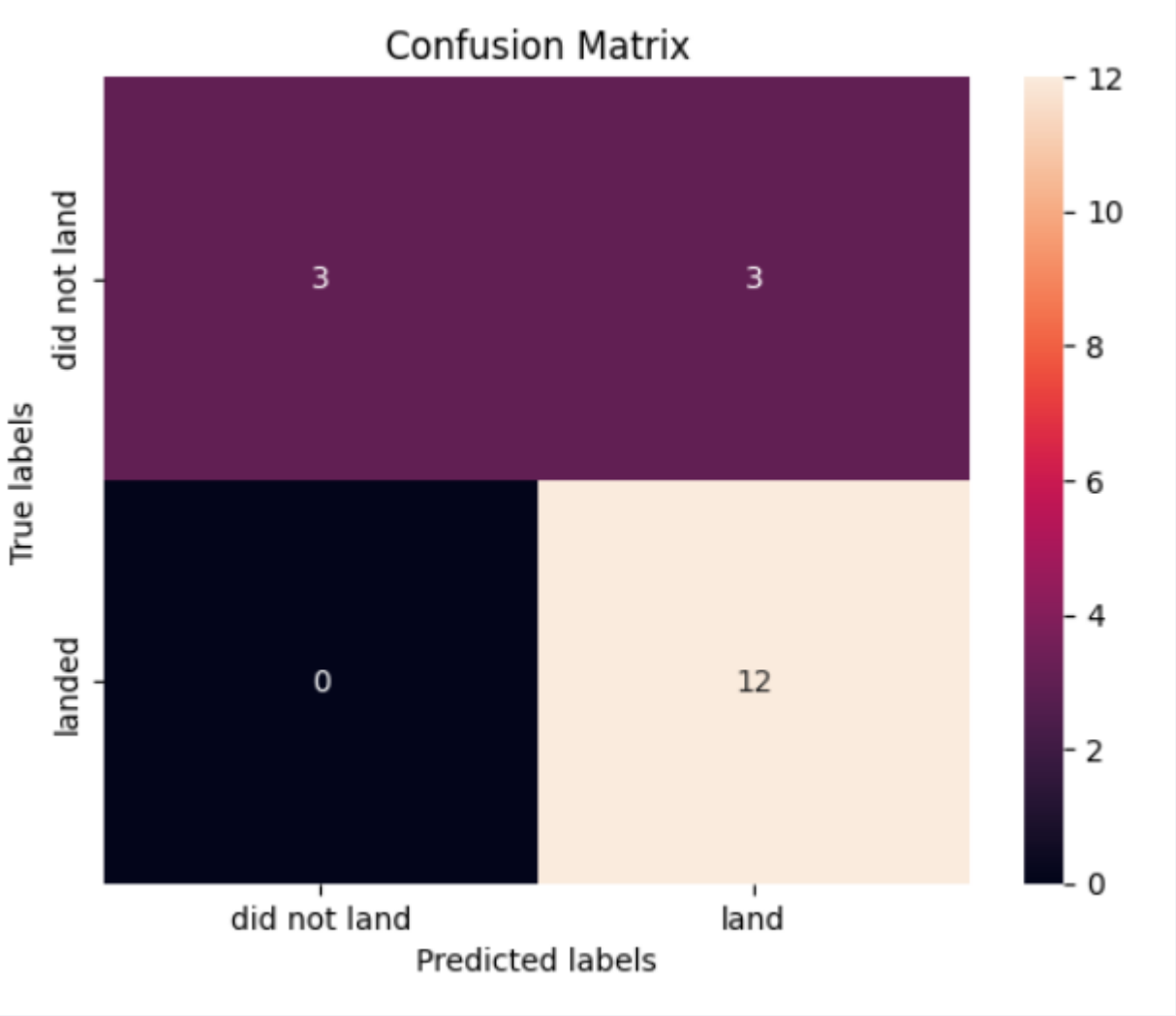
Predictive Analysis (Classification)

Classification Accuracy



Despite the fact that the decision tree outperformed the other three models based on training dataset accuracy, the performance of the four models on the test dataset was very consistent, with an accuracy of around 83%.

Confusion Matrix



Evaluation Summary

•**Accuracy (83.33%)**: The model correctly classified 83.33% of the test data.

•**Precision (100%)**: When the model predicts a positive class, it is always correct.

•**Recall (50%)**: The model correctly identifies 50% of the actual positive cases.

•**F1 Score (66.67%)**: The F1 score indicates a moderate balance between precision and recall, though it suggests that the model struggles with recall.

Conclusion

While the decision tree model shows perfect precision, indicating no false positives, its recall is quite low at 50%, meaning it misses half of the actual positive cases. This imbalance suggests that while the model is reliable in predicting positives, it fails to identify a significant portion of them. Therefore, improvements are needed to increase recall without sacrificing precision to enhance the overall performance of the model.

Conclusions

1. **Overall Accuracy**:
    1. The decision tree model achieved an accuracy of **83.33%** on the test dataset, indicating generally good performance.
2. **Precision**:
    1. The model has a perfect precision of **100%**, meaning all positive predictions are correct with no false positives.
3. **Recall**:
    1. The recall is **50%**, suggesting the model misses half of the actual positive cases.
4. **Performance Imbalance**:
    1. The high precision but low recall indicates the model is better at predicting negatives correctly but struggles with identifying all positive cases.
    2. The F1 score is **66.67%**, reflecting this imbalance between precision and recall.

Appendix

Recommendations for Model Improvement

1.Feature Engineering:

1. Enhance feature selection by investigating the importance of current features and engineering additional features.
2. Consider adding external data (e.g., weather, payload characteristics) to provide more context.

2.Handling Class Imbalance:

1. Apply resampling techniques like SMOTE or ADASYN to balance the training data.
2. Adjust class weights in the model to give more importance to the minority class.

3.Model Complexity:

1. Conduct a more extensive hyperparameter tuning process with a wider range of parameters.
2. Explore ensemble methods (Random Forest, Gradient Boosting, XGBoost) for improved performance.

4.Model Evaluation Metrics:

1. Utilize ROC-AUC and precision-recall curves to evaluate model performance more comprehensively.
2. Implement stratified k-fold and nested cross-validation for more reliable performance estimates.

Thank you!