IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dr Ernest Bolek (CFE)
06.03.2024

Outline

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

Introduction

- Project background and context

In this project, we will predict if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX 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 SpaceX for a rocket launch.

- Problems you want to find answers

Determine the price of each Space X rocket launch

Determine what attributes are correlated with successful landings

Explore the maps of the launch site locations and their close proximities to discover any patterns and explain how to choose an optimal launch site

Train a machine learning model based on the public information to predict if the first stage will land successfully and will SpaceX reuse the first stage

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX rocket launch data was collected based on two public sources SpaceX API and Falcon 9 launch records from Wikipedia.
- Performing data wrangling
  - Firstly Payload Mass mean values were calculated and used to replace the missing or NaN values. Secondly using the .value_counts() method on columns to calculate respectively (1) the number and occurrence of each orbit in the column Orbit, (2) the number and occurrence of mission outcome of the orbits on the column Outcome.
- Performing exploratory data analysis (EDA) using visualization and SQL

Methodology

Executive Summary

- Performing interactive visual analytics using Folium and Plotly Dash
- Performing predictive analysis using classification models
    - How to build, tune, evaluate classification models

    Preparing the data for the Predictive Analysis (Classification) process. Applying various Machine Learning methods including Logistic Regression, Support Vector Classification (SVC), Decision Tree Classifier, and K Nearest Neighbors (KNN). Evaluating the results based on the variance and accuracy score, and confusion matrix to find the best-performing classification model.

Data Collection

- SpaceX rocket launch data was collected based on two public sources
  - SpaceX API
  - Falcon 9 launch records from Wikipedia

- Data collection process

1) Requesting the data

Requesting rocket launch data using GET request from SpaceX API. After reviewing the data more detailed information including the rocket, payloads, launchpad, and cores using the rocket launch IDs was requested.

2) Parsing data

Defining a series of auxiliary functions used to extract information. Extracting a subset of the data frame with the relevant data and features e.g. booster name, the mass of the payload, orbit, launch site name and geographical location (longitude, latitude), landing outcome, type of landing, number of flights with that core, grid fins used, core reused, legs used, landing pad used, number of the core block version, number of times a core was reused, core serial number.

Data Collection

3) Preparing the data frame

Creating a data frame with global variables using predefined get functions. Constructing a dataset using the data gathered and combining the columns into a dictionary. Creating final Pandas data frame from the dictionaries.

4) Data filtering

Filtering relevant data from the data frame. Falcon 1 rocket launches were removed as irrelevant and only Falcon 9 rocket launch data was saved to the final data frame data_falcon9.

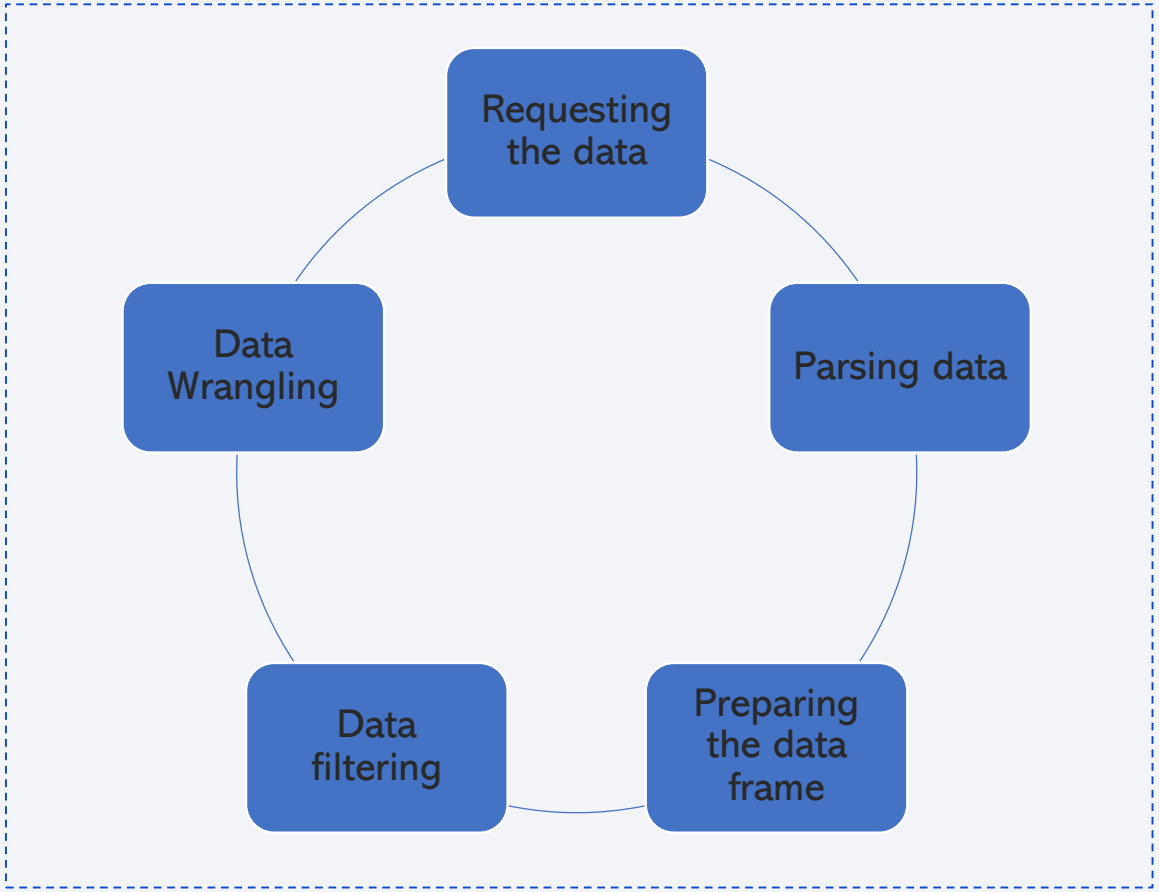
5) Data Wrangling

After additional parsing of the dataset, rows with missing values were identified in the Payload Mass column. Payload Mass mean values were calculated and used to replace the missing or NaN values.

Data Collection – SpaceX API

- Data collection with SpaceX REST calls flowchart

- GitHub URL of the completed SpaceX API calls notebook https://github.com/ErnestBol/Capstone-SpaceX/blob/main/1%20jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data collection web scraping process

1) Requesting the data

Requesting the Falcon 9 and Falcon Heavy Launches Records HTML table from Wikipedia. Performing an HTTP GET method to request the data HTML page, as an HTTP response.

2) Extracting data

Creating a BeautifulSoup object from the HTML response. Extracting all Falcon 9 launch records, columns, and variable names from the HTML table header.

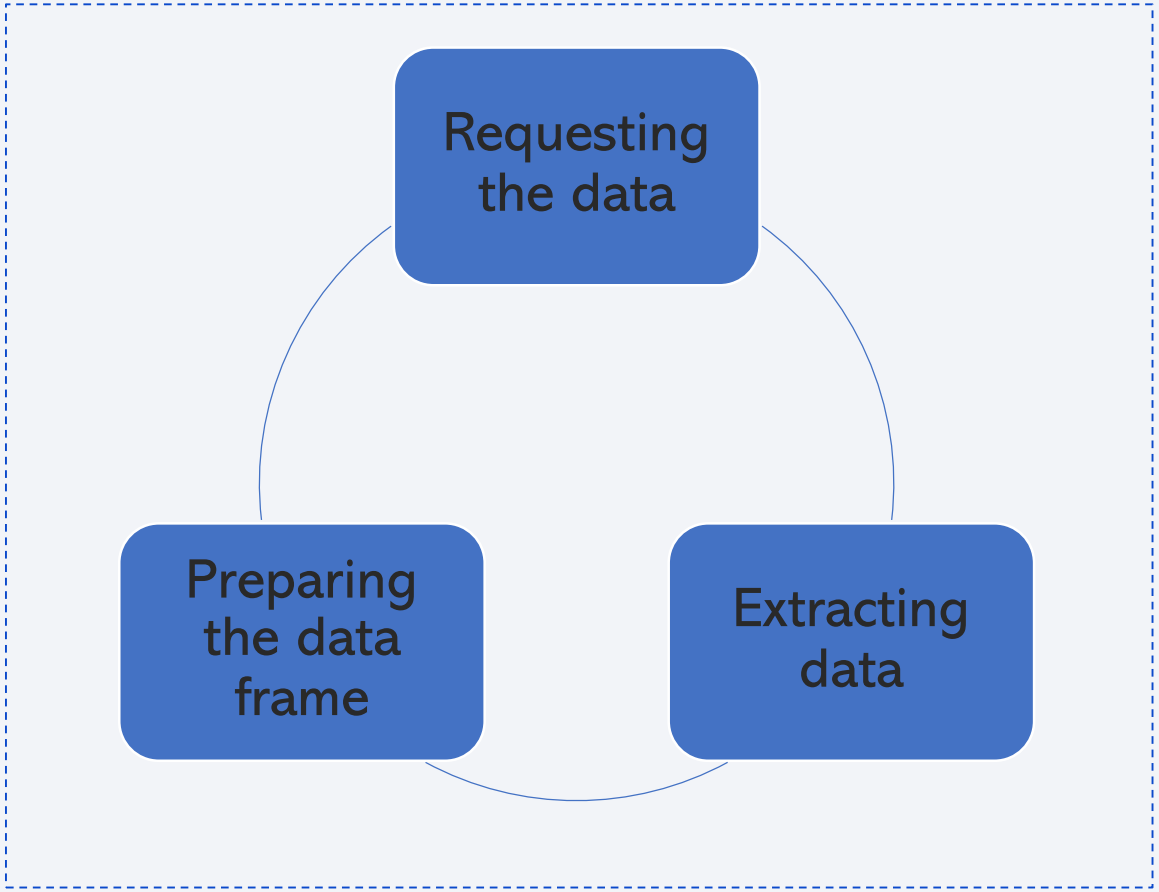
3) Preparing the data frame

Creating an empty dictionary with keys from the extracted column names. Preparing several suport functions to process web scraped HTML table. Parsing the launch HTML tables and filling up the launch_dict with launch records extracted from table rows. Converting this dictionary into a Pandas data frame.

Data Collection - Scraping

- Web scraping process flowchart

- GitHub URL of the completed web scraping notebook https://github.com/ErnestBol/Capstone-SpaceX/blob/main/2%20jupyter-labs-webscraping.ipynb



Data Wrangling

- Data Wrangling process

1) Initial data analysis

Loading Space X dataset prepared in the previous phase. Identifying and calculating the percentage of the missing values in each attribute (for example for Landing Pad equals 28.88%). Identifying numerical and categorical columns.

2) Rocket launches data analysis

Using the .value_counts() method on columns to calculate respectively (1) the number and occurrence of each orbit in the column Orbit, (2) the number and occurrence of mission outcome of the orbits on the column Outcome. The results were assigned to a landing_outcomes variable.

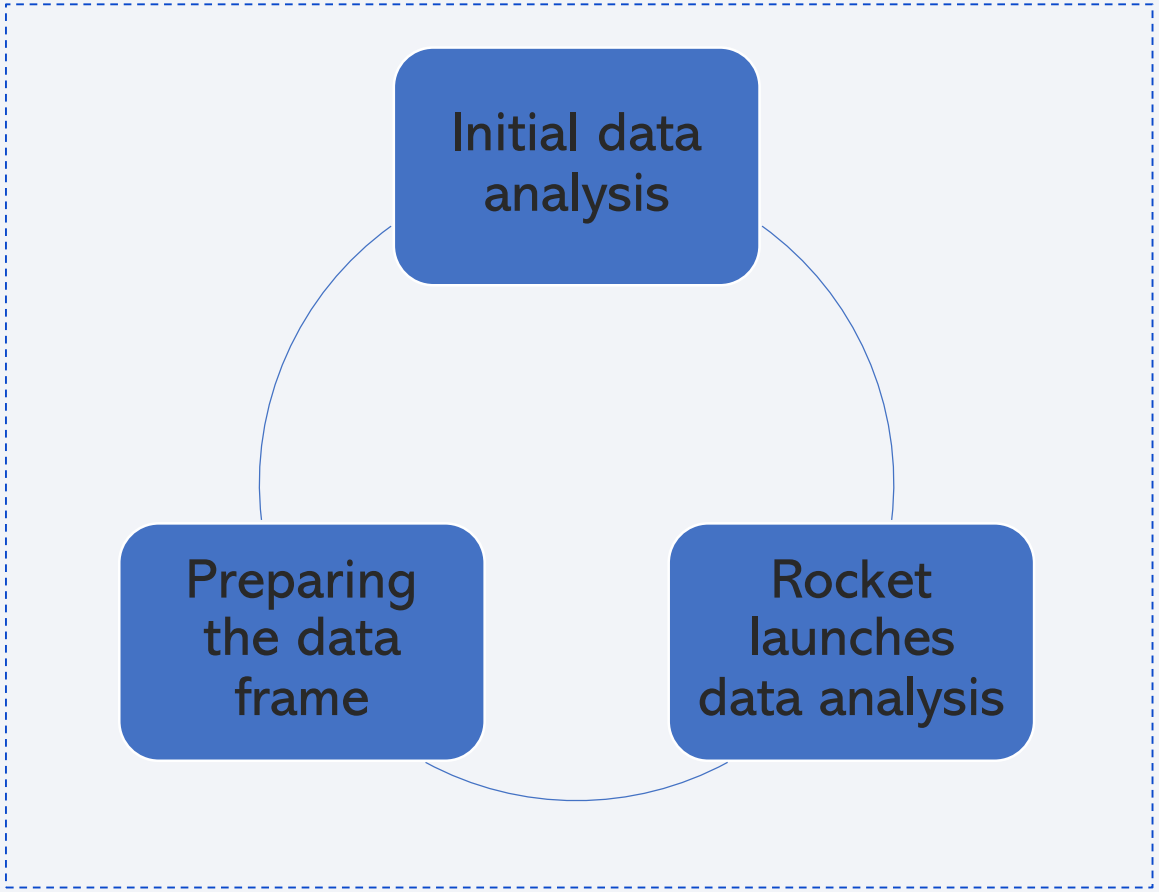
3) Preparing the data frame

Creating a set of bad_outcomes where the second stage did not land successfully based on the landing_outcomes variable. Creating a landing_class list to determine first stage landing result equals one for successful and zero for unsuccessful landing (bad_outcome). Assigning the results to the Class data frame. The success rate of the first stage equals 0.66.

Data Wrangling

- Data Wrangling process flowchart

- GitHub URL of the completed data wrangling notebook https://github.com/ErnestBol/Capstone-SpaceX/blob/main/3%20labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Exploratory Data Analysis with Data Visualization Process:
  - **Flight Number vs. Payload Mass** scatter plot illustrates the relationship between the Flight Number (indicating the continuous launch attempts) and the Payload Mass (kg) variables
  - **Flight Number vs. Launch Site** scatter plot illustrates the relationship between the Flight Number (indicating the continuous launch attempts) and the Launch Site variables
  - **Payload Mass vs. Launch Site** scatter plot illustrates the relationship between the Launch Site and the Payload Mass (kg) variables.
  - **Success Rate (Class) vs. Orbit Type** bar chart illustrates the success rate for each orbit type
  - **Flight Number vs. Orbit Type** scatter plot illustrates the relationship between the Flight Number (indicating the continuous launch attempts) and the Orbit Type variables

EDA with Data Visualization

- Exploratory Data Analysis with Data Visualization Process:
  - **Payload Mass vs. Orbit Type** scatter plot illustrates the relationship between the Payload Mass (kg) and Orbit Type variables.
  - **Average Rocket Launch Success Trend** line chart illustrates the increase of the yearly average success rate with time.

- GitHub URL of completed EDA with data visualization notebook

https://github.com/ErnestBol/Capstone-SpaceX/blob/main/5%20jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Summary of the SQL queries performer on the DataSet
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome on the ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes

EDA with SQL

- Summary of the SQL queries performer on the DataSet
  - 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 the year 2015.
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order

- GitHub URL of your EDA with SQL notebook
  https://github.com/ErnestBol/Capstone-SpaceX/blob/main/4%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Summary of interactive map building with Folium process

1) Marking all launch sites on an interactive map

Creating a folium Map object and adding folium.Circle and folium.Marker to a highlighted circle area with a text label on a specific coordinate for each launch site on the site map.

2) Marking the success/failed launches on the map

Enhance the map by adding the launch outcomes for each site to illustrate which sites have high success or failure rates. Obtaining detailed launch records from the spacex_df data frame and mission result (success/failed) from the class column. Adding a folium.Marker to marker_cluster for each launch result.

Build an Interactive Map with Folium

- Summary of interactive map building with Folium process

3) Calculate the distances between a launch site and proximities

Marking down all relevant points using MousePosition and calculating the distance between those points and the launch site for example coastline, closest city etc.

Closest coastline point and closest city marked on the Folium map are especially relevant due to the safety reasons.

- GitHub URL of interactive map with Folium map

https://github.com/ErnestBol/Capstone-SpaceX/blob/main/6%20lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Building an Interactive Dashboard Process (relevant plots, graphs and interactions described below together with the justification):

1) Preparation phase

Reading the airline data into a pandas data frame. Creating a dash application and an app layout.

2) Dashboard Dropdown List

Adding a dropdown list to the dashboard to enable Launch Site selection by the User.

3) Dashboard Pie chart

Adding a pie chart to the dashboard to show the total successful launch count for all sites. The User will be able to select a specific launch site and review Success vs. Failed counts for that site.

Build a Dashboard with Plotly Dash

- Building an Interactive Dashboard Process:

4) Dashboard Callback Function

Adding a callback function to the dashboard for (1) `site-dropdown` as input, and `success-pie-chart` as output, (2) `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output. Enabling User interaction with the dashboard that is selecting a specific launch site or values range.

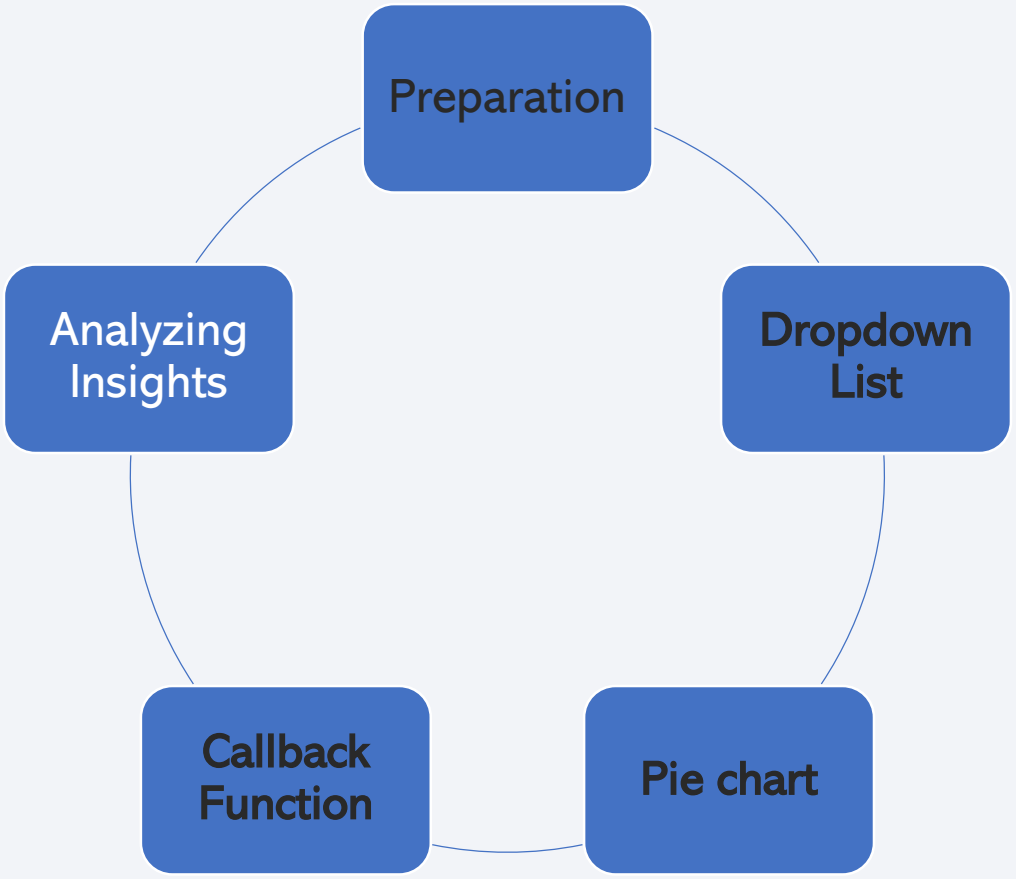
5) Analyzing Insights Visually Phase

Analyzing the SpaceX launch data and evaluating the insights visually to present the findings.

- GitHub URL of completed Plotly Dash lab

https://github.com/ErnestBol/Capstone-SpaceX/blob/main/7%20spacex_dash_app.py

Build a Dashboard with Plotly Dash



Predictive Analysis (Classification)

- Machine Learning Prediction Process

1) Data preparation

Defining auxiliary functions to plot the confusion matrix. Loading the data frame. Creating a NumPy array applying the method to_numpy()and saving the output as Pandas series. Standardize the data and split it into training and testing data using the function train_test_split. Dividing the training data into two sets (1) validation data, and (2) training data. Train the model and select the hyperparameters using the GridSearchCV function.

2) Logistic Regression

Creating a logistic regression object and a GridSearchCV object logreg_cv with cv = 10. Fitting the object to find the best parameters from the dictionary parameters. GridSearchCV object output was used for logistic regression. Displaying the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_. Calculating the accuracy of the test data using the method score and plotting the confusion matrix.

Predictive Analysis (Classification)

3) Support Vector Classification (SVC)

Creating a support vector machine object and a GridSearchCV object svm_cv with cv - 10. Fitting the object to find the best parameters from the dictionary parameters. Calculating the accuracy of the test data using the method score and plotting the confusion matrix.

4) Decision Tree Classifier

Creating a decision tree classifier object and a GridSearchCV object tree_cv with cv = 10. Fitting the object to find the best parameters from the dictionary parameters. Calculating the accuracy of tree_cv on the test data using the method score and plotting the confusion matrix.

5) K Nearest Neighbors (KNN)

Creating a K Nearest Neighbors object and a GridSearchCV object knn_cv with cv = 10. Fitting the object to find the best parameters from the dictionary parameters. Calculating the accuracy of knn_cv on the test data using the method score and plotting the confusion matrix.

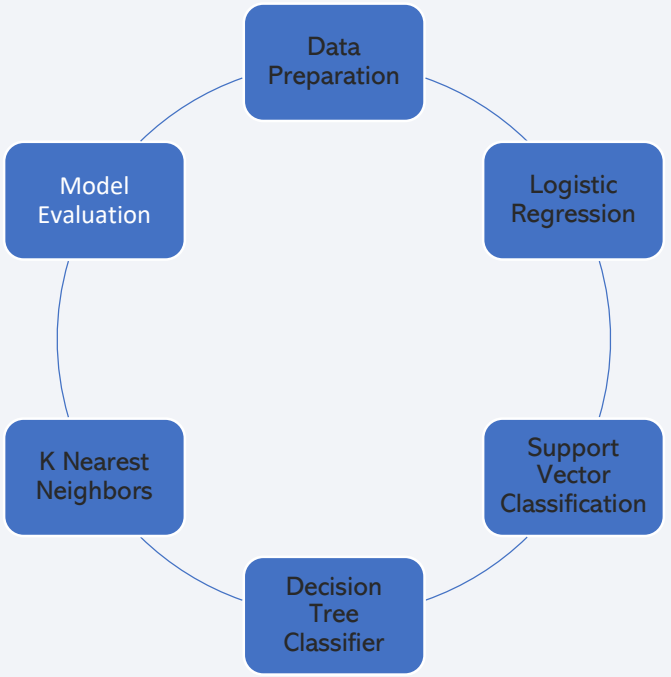
Predictive Analysis (Classification) Flowchart

6) Model evaluation

Evaluating the results of the Machine Learning methods to find the best-performing classification model. The best algorithm for the training data set is: Decision Tree with a score of 0.8875

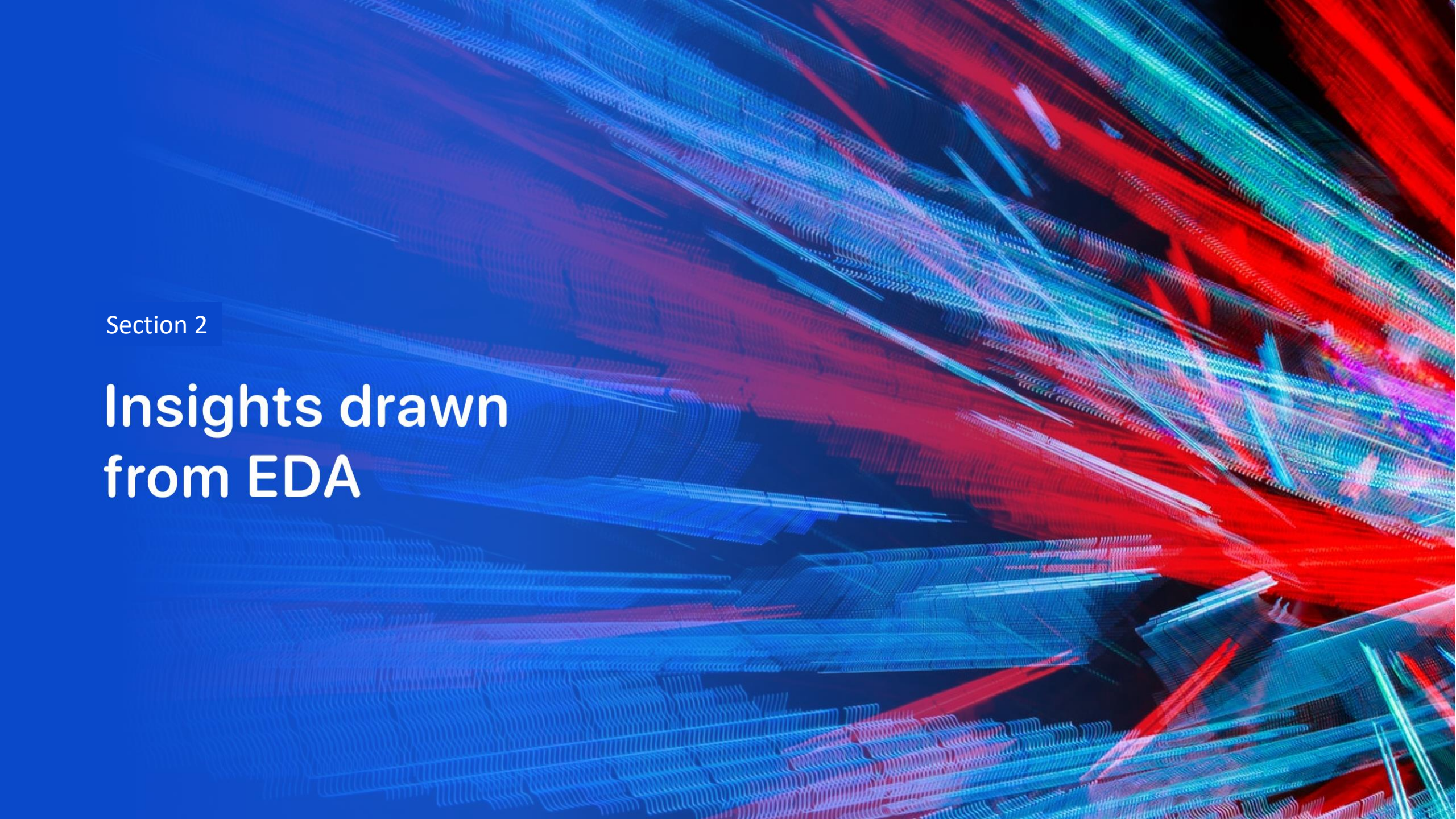
- GitHub URL of completed predictive analysis lab

https://github.com/ErnestBol/Capstone-SpaceX/blob/main/8%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(1).ipynb

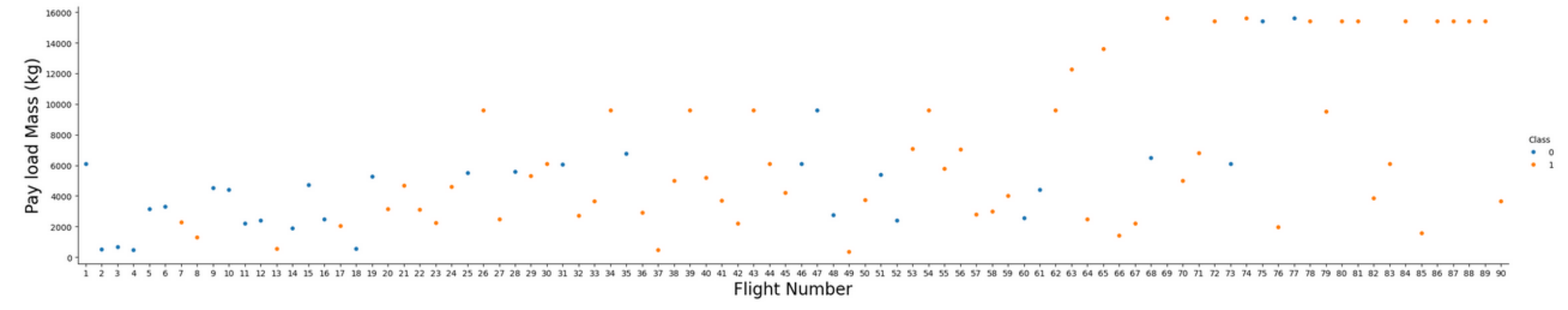


Results

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

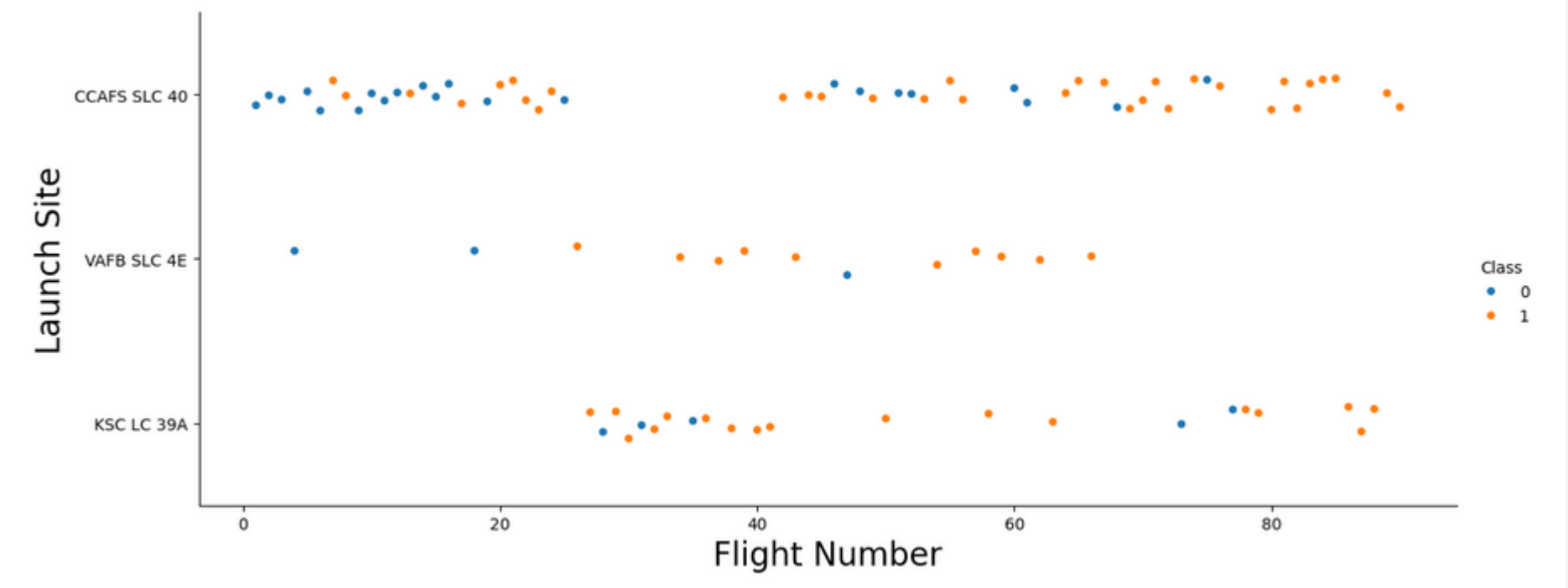

Section 2
Insights drawn from EDA

Flight Number vs. Payload Mass



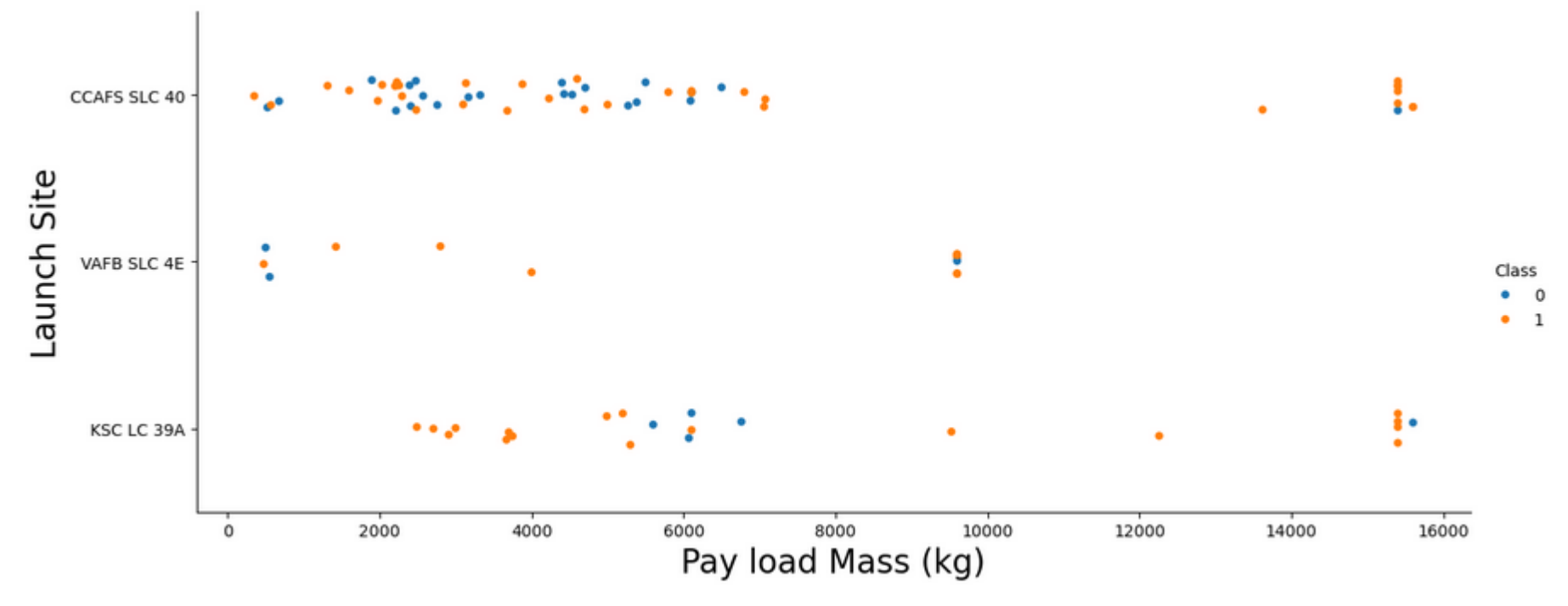
- The relationship between Flight Number vs. Payload Mass
- As the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Flight Number vs. Launch Site



- The relationship between Flight Number vs. Launch Site
- As the flight number increases, the first stage is more likely to land successfully. The launch site is also important; CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%. Also, it seems that various locations had higher success rates after different number of rocket launches.

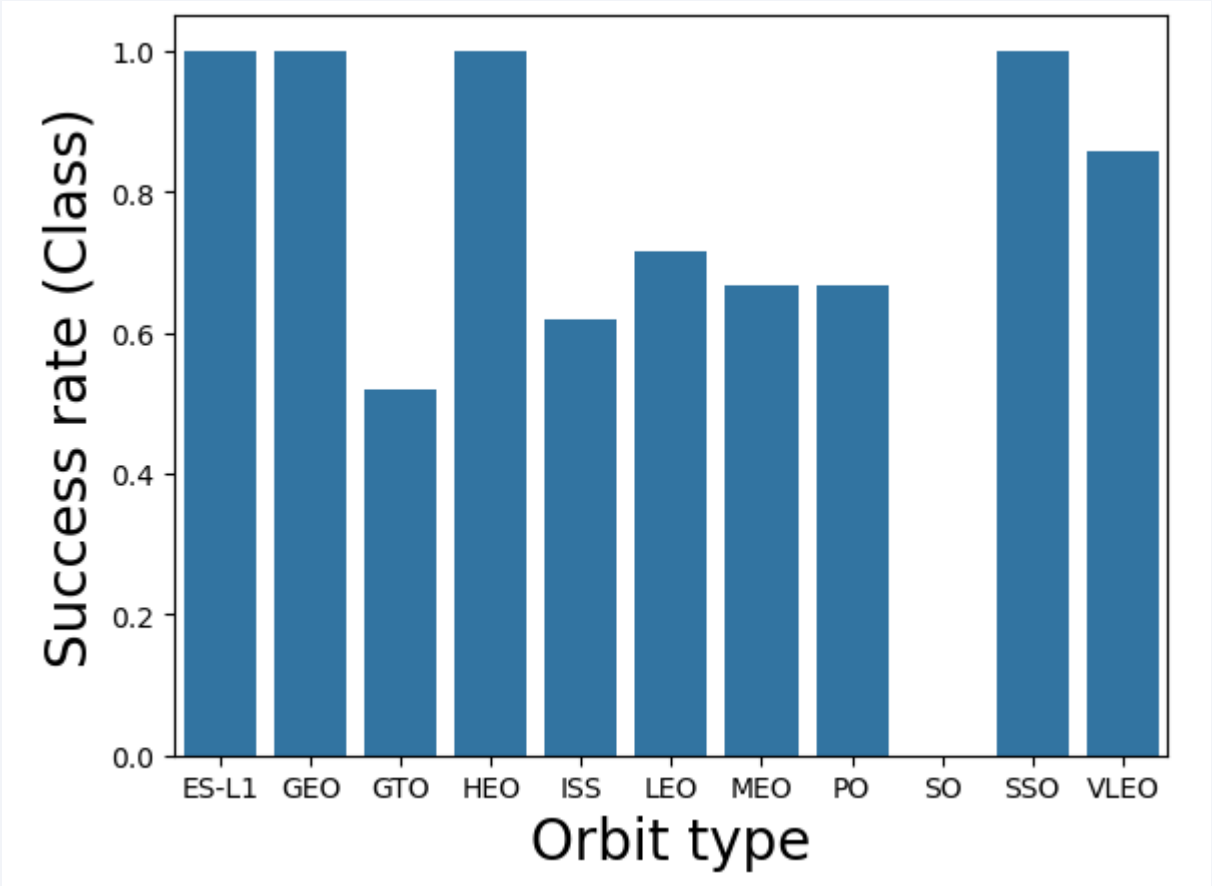
Payload vs. Launch Site



- The relationship between Payload Mass vs. Launch Site
- As the payload mass (kg) increases above 7000, the first stage is more likely to land successfully. Also both KSC LC-39A and VAFB SLC 4E have higher success rate than CCAFS LC-40. For the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

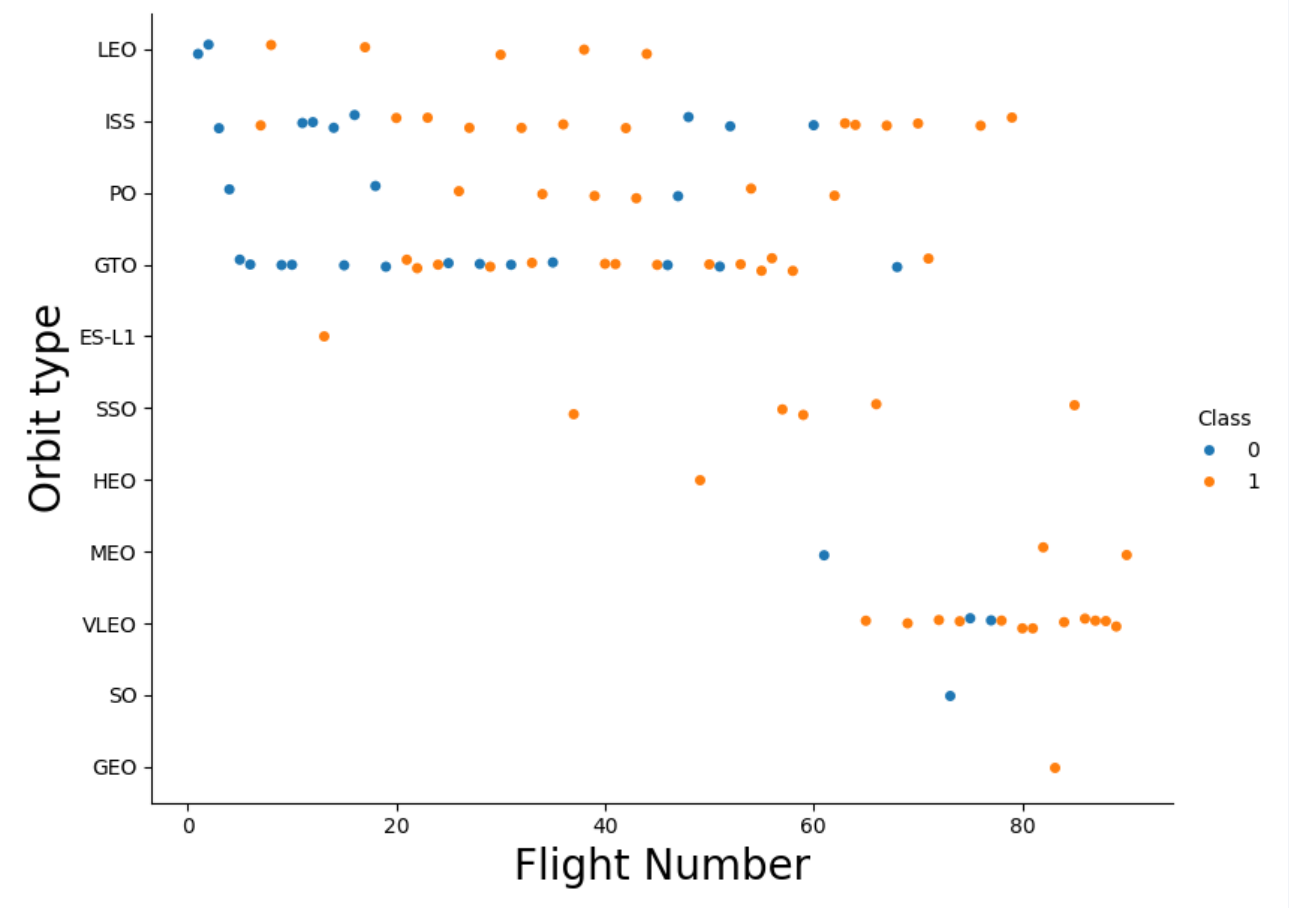
Success Rate vs. Orbit Type

- The relationship between Success Rate (Class) vs. Orbit Type
- For ES-L1, GEO HEO, and SSO Orbit Type, the first stage is more likely to land successfully (Success rate equals 1.0. GTO Orbit Type has the lowest success rate equal to 0.5.

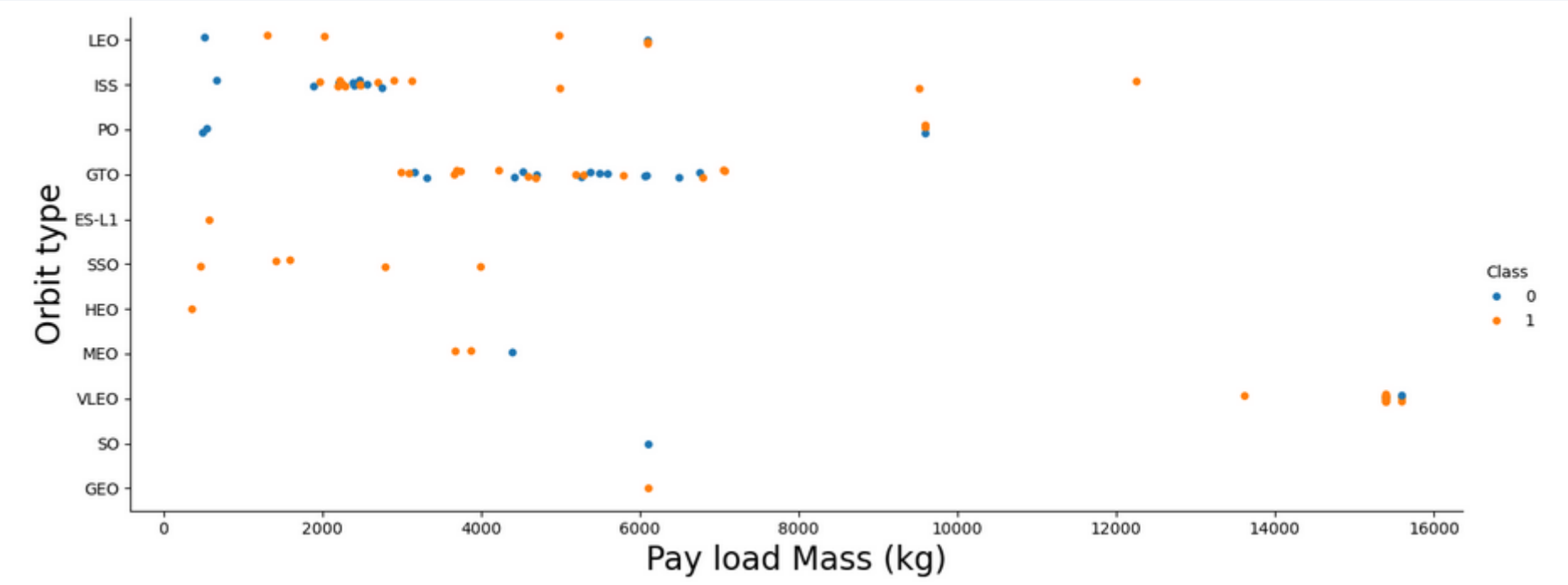


Flight Number vs. Orbit Type

- The relationship between Flight Number vs. Orbit Type
- In the LEO orbit the Success appears to be related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



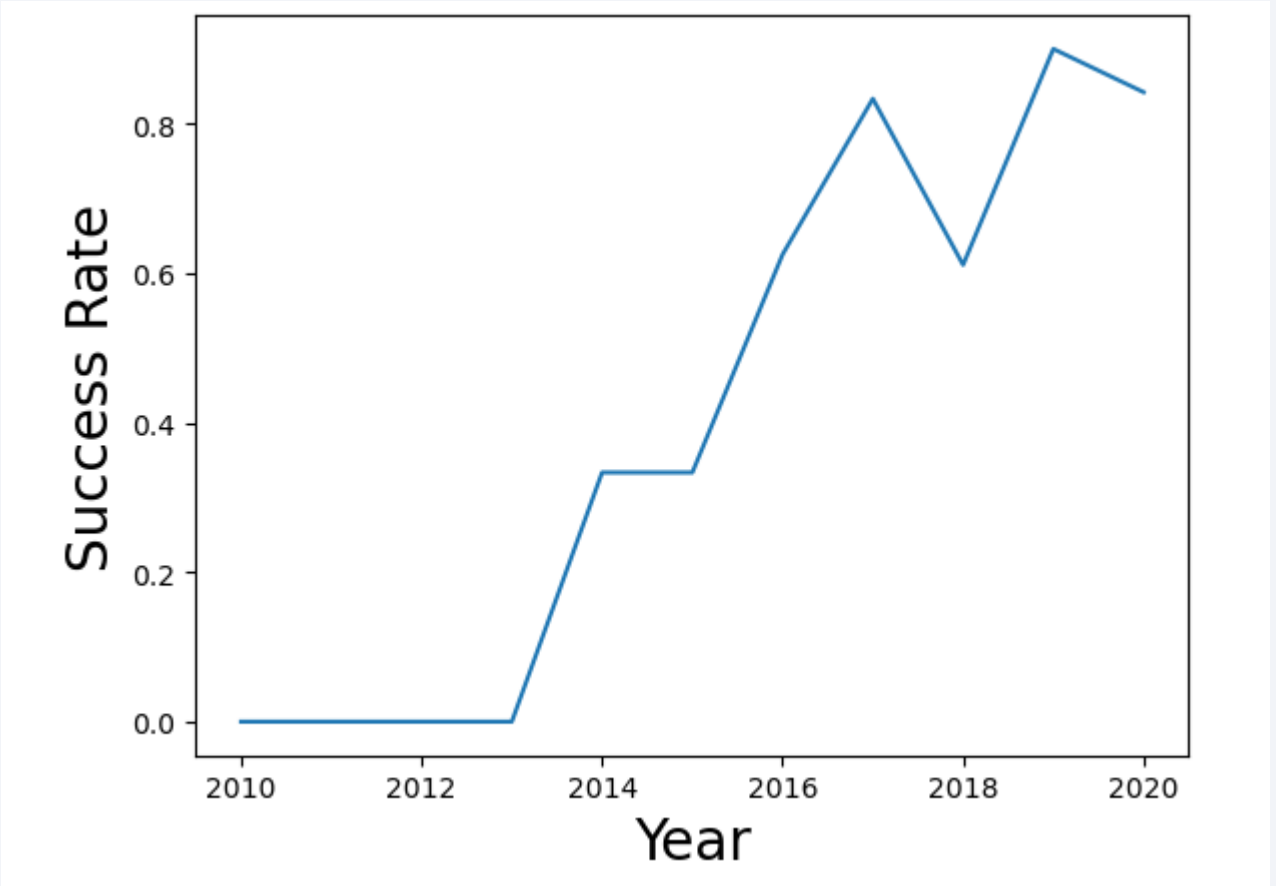
Payload vs. Orbit Type



- The relationship between Payload Mass vs. Orbit Type
- With heavy payloads, the successful landing or positive landing rate is higher for Polar, LEO, and ISS. However, for GTO we cannot distinguish this relationship well as both positive landing rate and negative landing (unsuccessful mission) appear there.

Launch Success Yearly Trend

- Average Rocket Launch Success Trend
- The success rate kept increasing steadily from 2013 till 2020 with a small decrease in 2018.



All Launch Site Names

- Displaying the names of the unique launch sites in the space mission using the following SQL query

%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'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parac
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 (parac
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No att
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No att
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No att

- Displaying five records where launch sites begin with the string 'CCA' using the following SQL query

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

Total Payload Mass

- Calculating the total payload carried by boosters from NASA using the following SQL query

SUM(PAYLOAD_MASS__KG_)
45596

```
%sql select SUM(PAYLOAD_MASS__KG_) from
SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

Average Payload Mass by F9 v1.1

- Calculating the average payload mass carried by booster version F9 v1.1 using the following SQL query

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Finding the date of the first successful landing outcome on ground pad using the following SQL query

%sql select MIN(Date) from SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listing the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 using the following SQL query

```
%sql select DISTINCT Booster_Version from SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)' and
PAYLOAD_MASS__KG_  between 4000 and 6000
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculating the total number of successful and failure mission outcomes using the following SQL query

- Per specific mission outcomes

%sql select Mission_Outcome, Count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome order by Count(Mission_Outcome)

- In total success\failure

%sql select substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from SPACEXTBL group by 1

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

Mission_Outcome	count(*)
Failure	1
Success	100

Boosters Carried Maximum Payload

- Listing the names of the booster which have carried the maximum payload mass using the following SQL query

```
%%sql select DISTINCT Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=(select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 using the following SQL query

%%sql select substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome ='Failure (drone ship)' and substr(Date,0,5)='2015';

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

- Ranking 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(*) from SPACEXTBLwhere Date between '2010-06-04' and '2017-03-20'group by Landing_Outcome order by 2 desc;

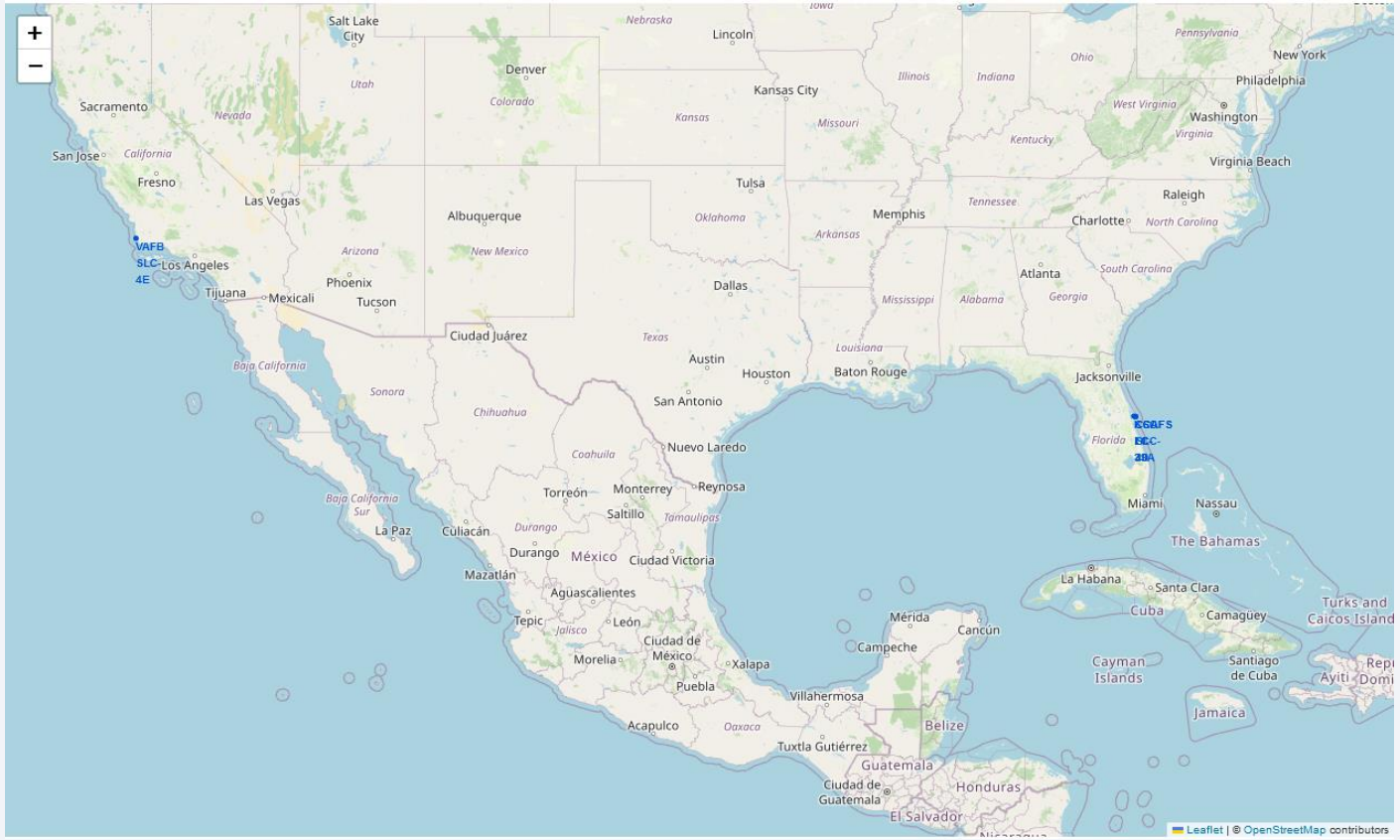
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

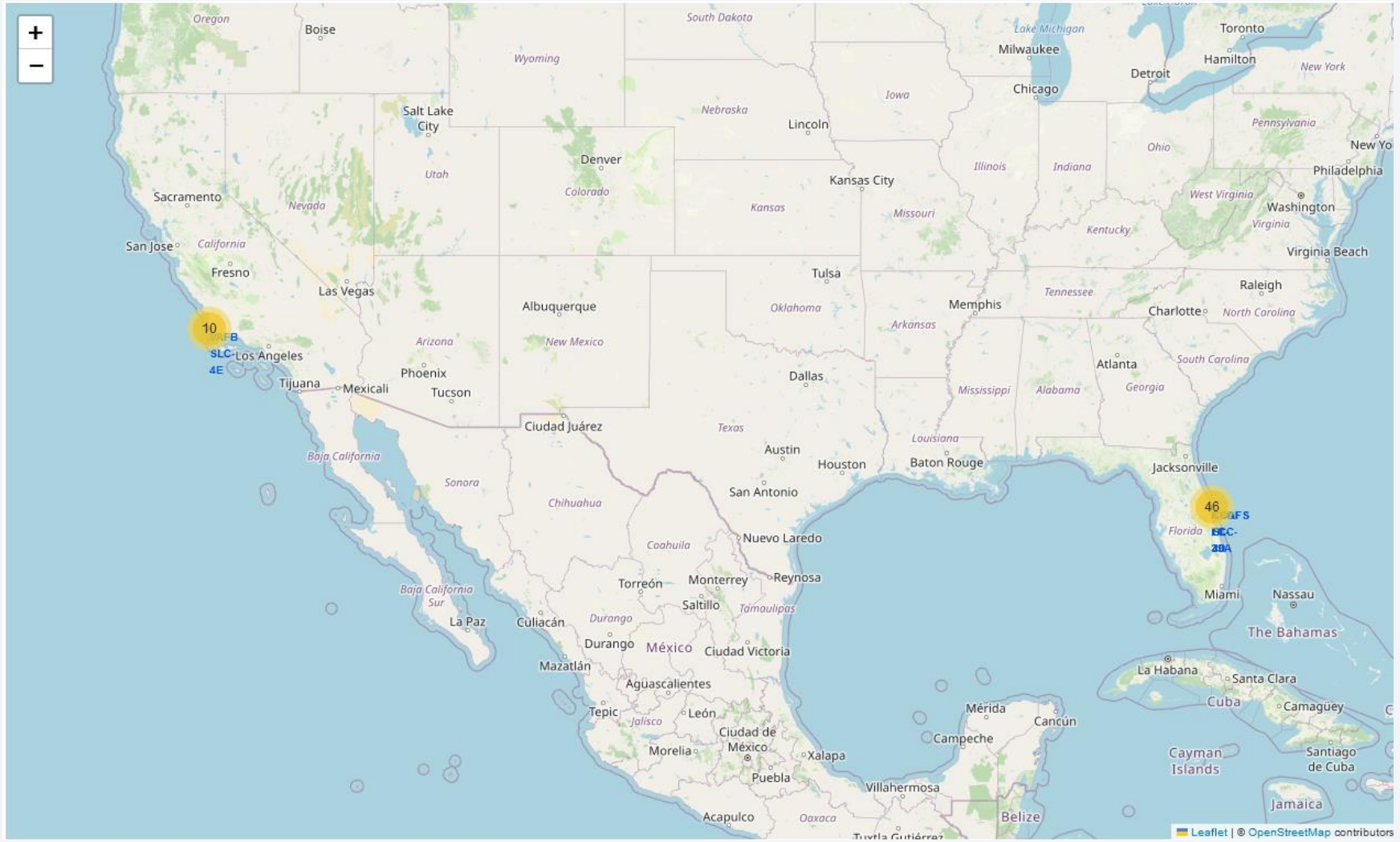
Launch Sites Proximities Analysis

Folium Map With Marked Launch Sites

All launch sites are in proximity to the Equator line and very close to the coast. There are two basic reasons behind this (1) reducing the fuel consumption required due to Earth's rotation, (2) safety reasons.

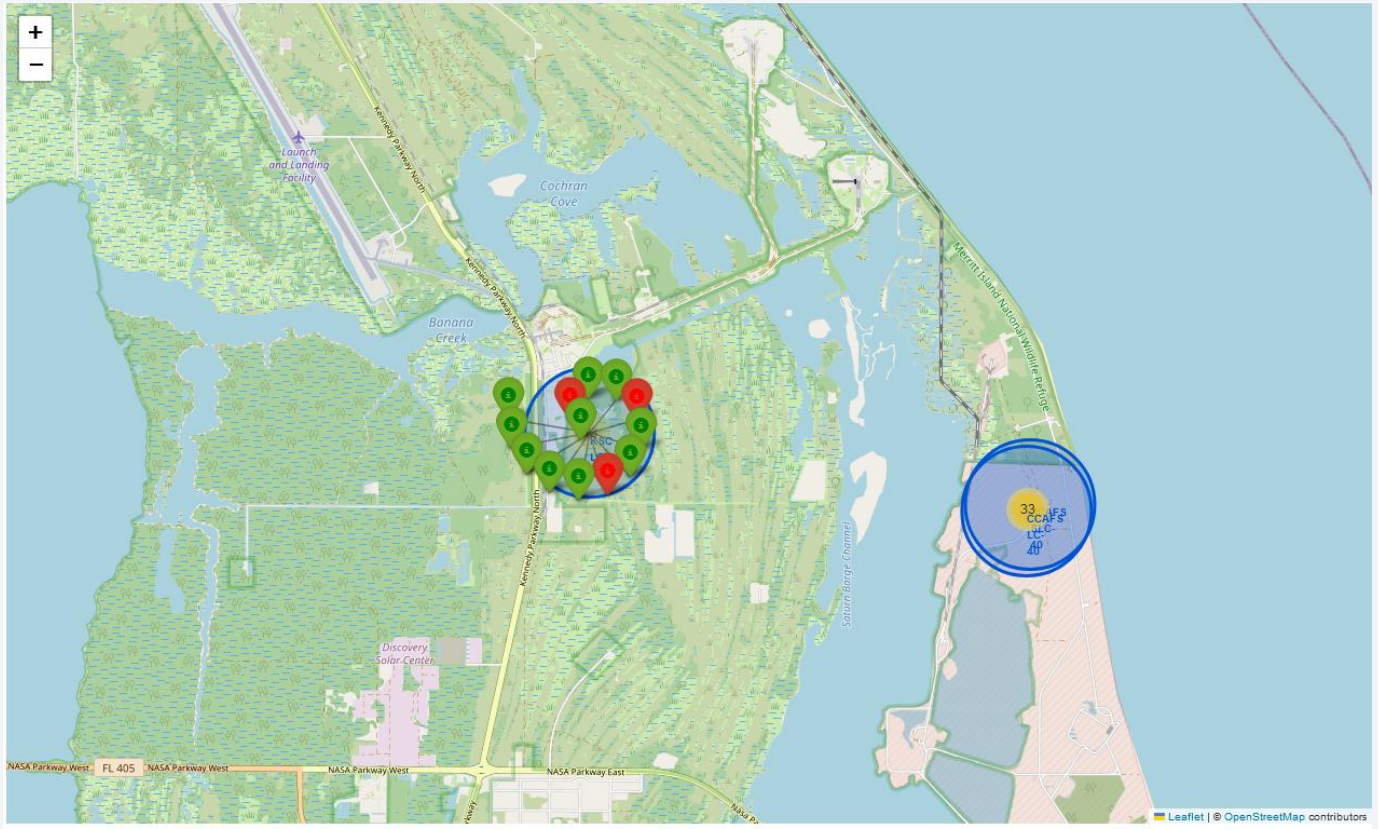


Map With Success/Failed Launches Marked



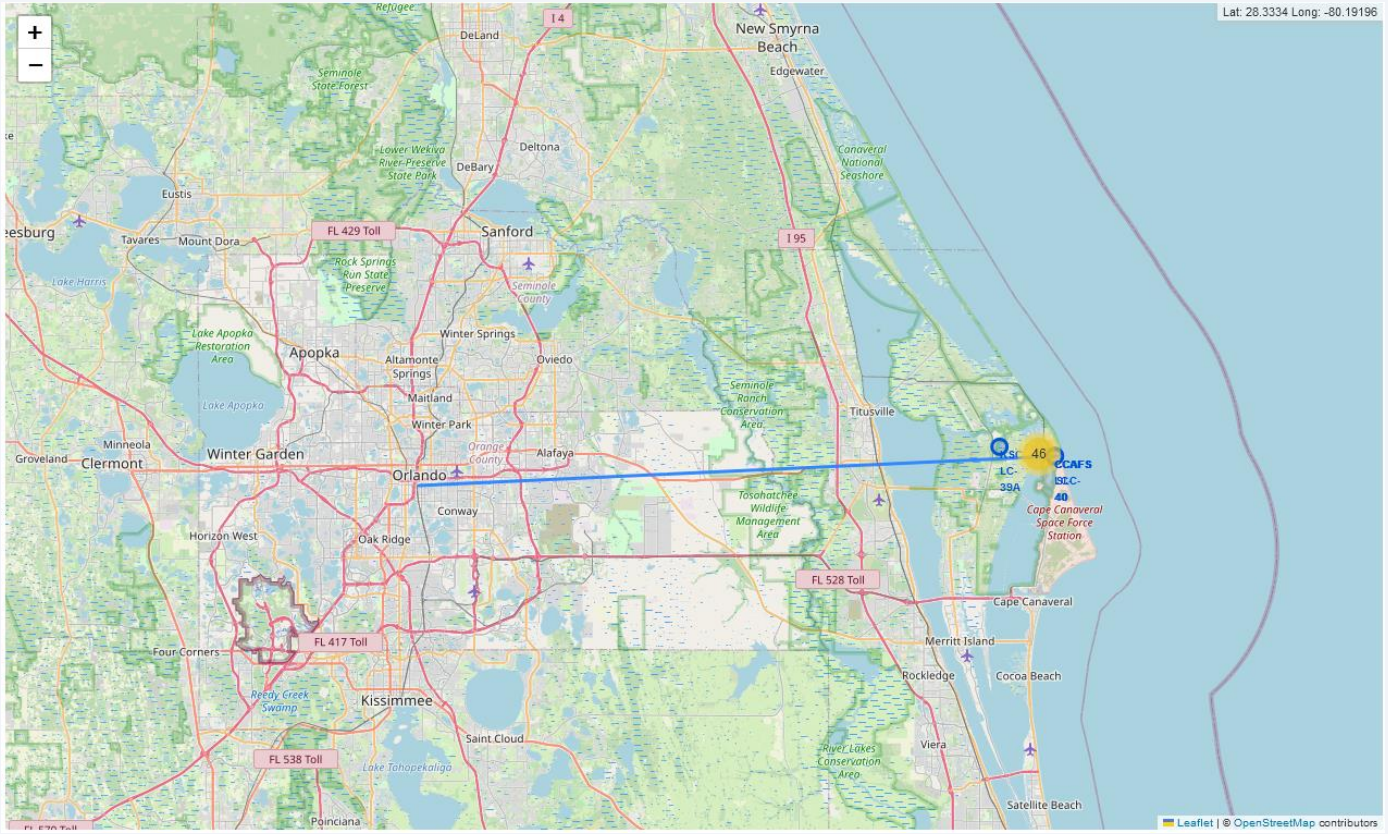
Map With Success/Failed Launches Marked

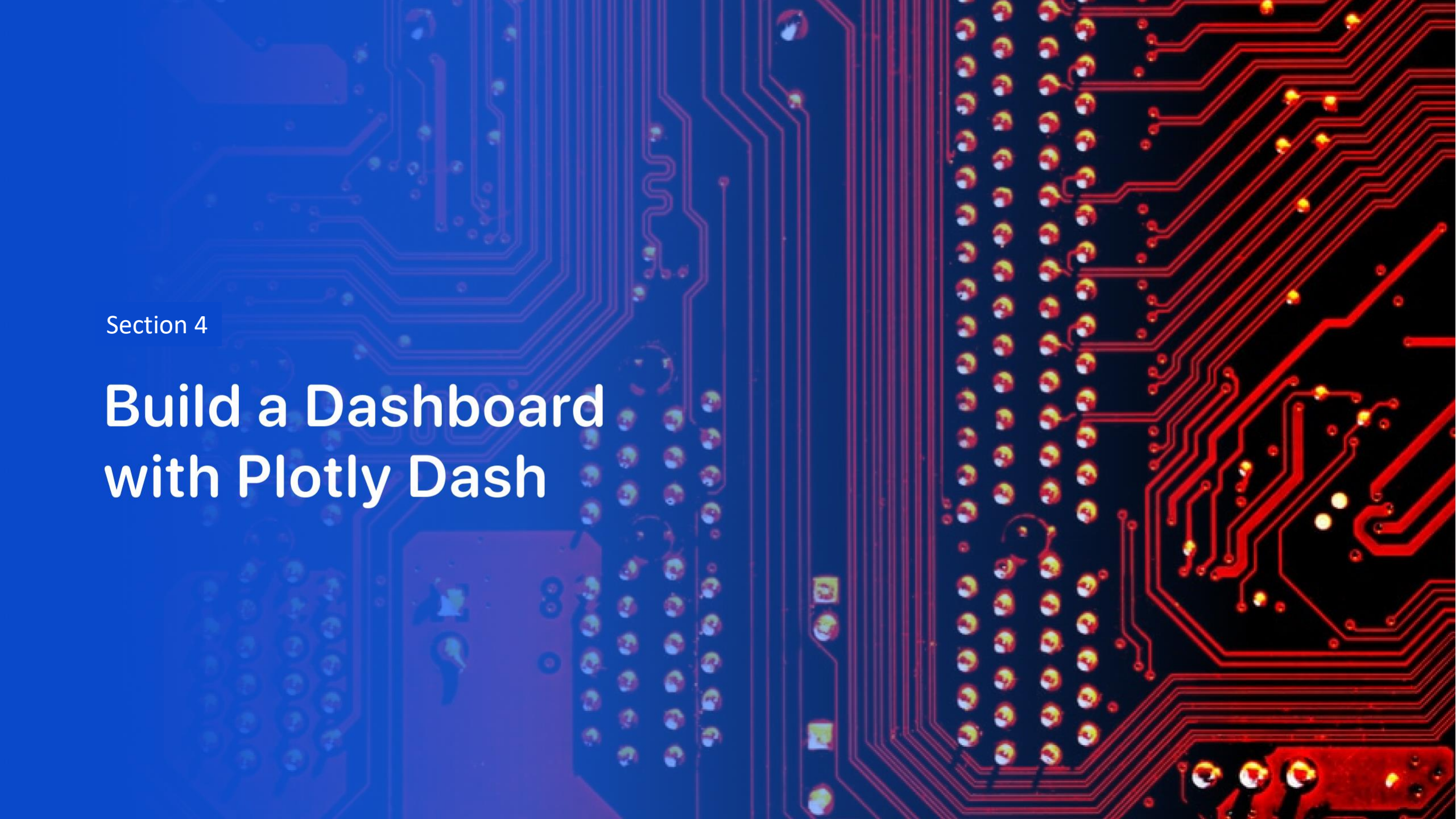
Using color-labeled markers in marker clusters enables quick and easy identification which launch sites have relatively high success rates. Accordingly, the KSC LC-39A launch site presented in the screenshot has a high success rate.



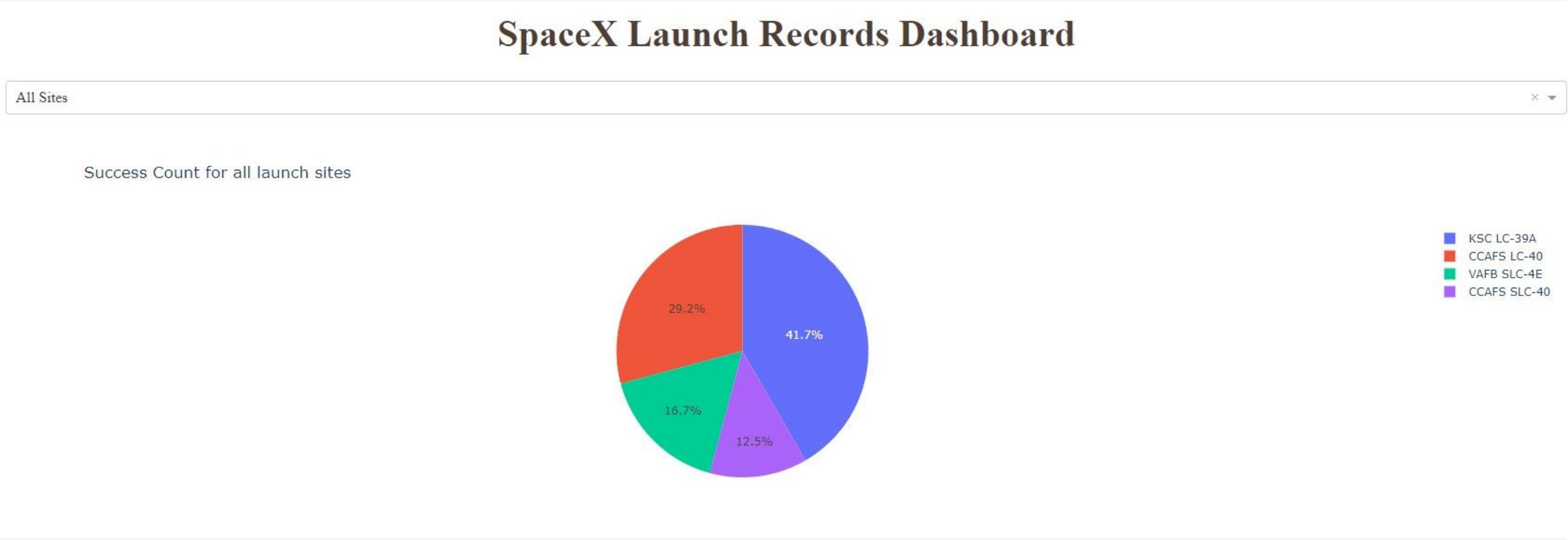
Map With Line To Relevant Points

Relevant points such as the closest coastline point and closest city were marked on the Folium map as critical due to safety reasons.



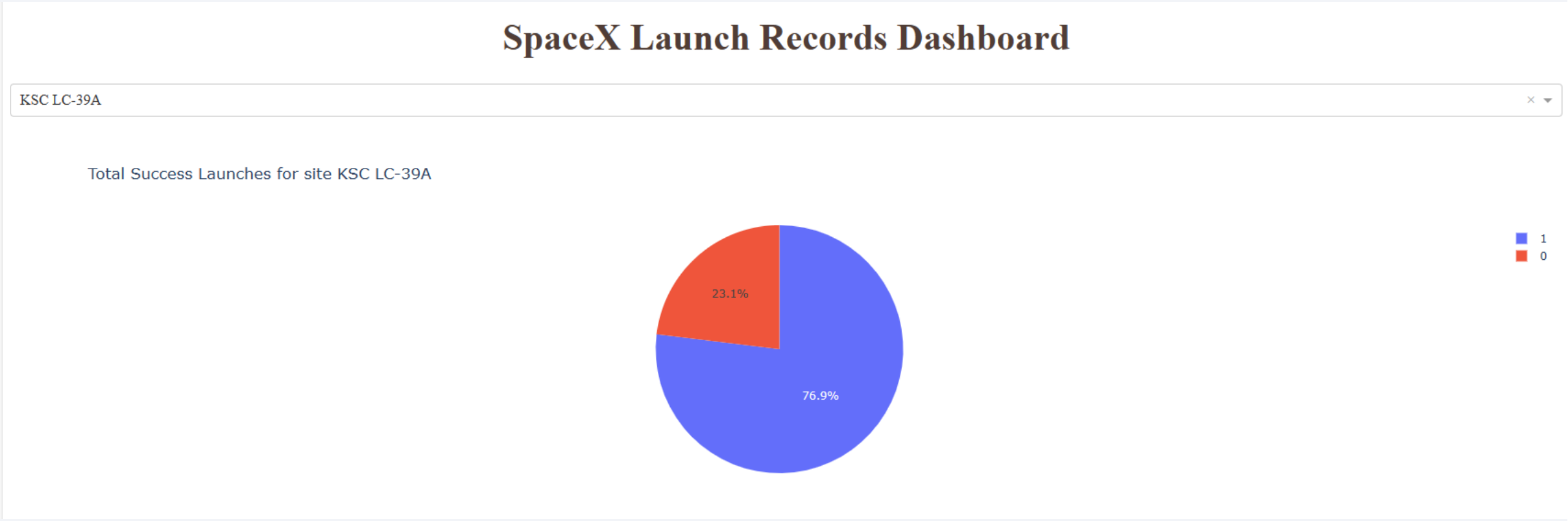

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



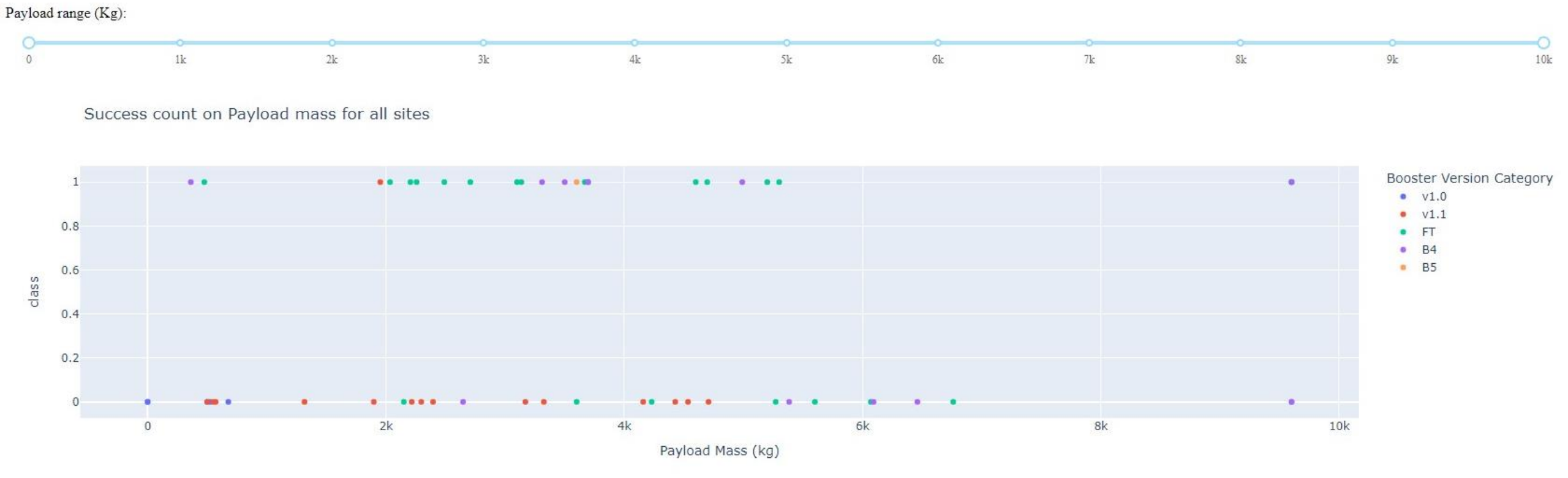
- Comparing successful counts for all launch sites
- Launch site with the largest successful launches: KSC LC-39A

SpaceX Launch Records Dashboard



- Launch site with the largest successful launches: KSC LC-39A
- Highest launch success rate: KSC LC-39A 76.9%

Payload Mass Success Count



- Highest launch success rate: 2000-4000
- Lowest launch success rate: 6000-8000
- F9 Booster version with the highest launch success rate: B5 and FT

Findings Summary

- Analyzing the SpaceX launch data and evaluating the insights visually to present the following findings

Insight	Result
Launch site with the largest successful launches	KSC LC-39A
Launch site with the highest launch success rate	KSC LC-39A success rate 76.9%
Payload range(s) with the highest launch success rate	2000-4000
Payload range(s) with the lowest launch success rate	6000-8000
F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) with the highest launch success rate	B5 with only one successful start FT with 15 successes, 8 failures

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

[42]:
```
algorithms = {'KNN':knn_cv.best_score_,'DecisionTree':tree_cv.best_score_,'SVM':svm_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm for training data set is:',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Decision Tree':
    print('Best Params is :',tree_cv.score)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.score)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.score)
if bestalgorithm == 'SVM':
    print('Best Params is :',svm_cv.score)
```

```
Best Algorithm for training data set is: DecisionTree with a score of 0.8875
```

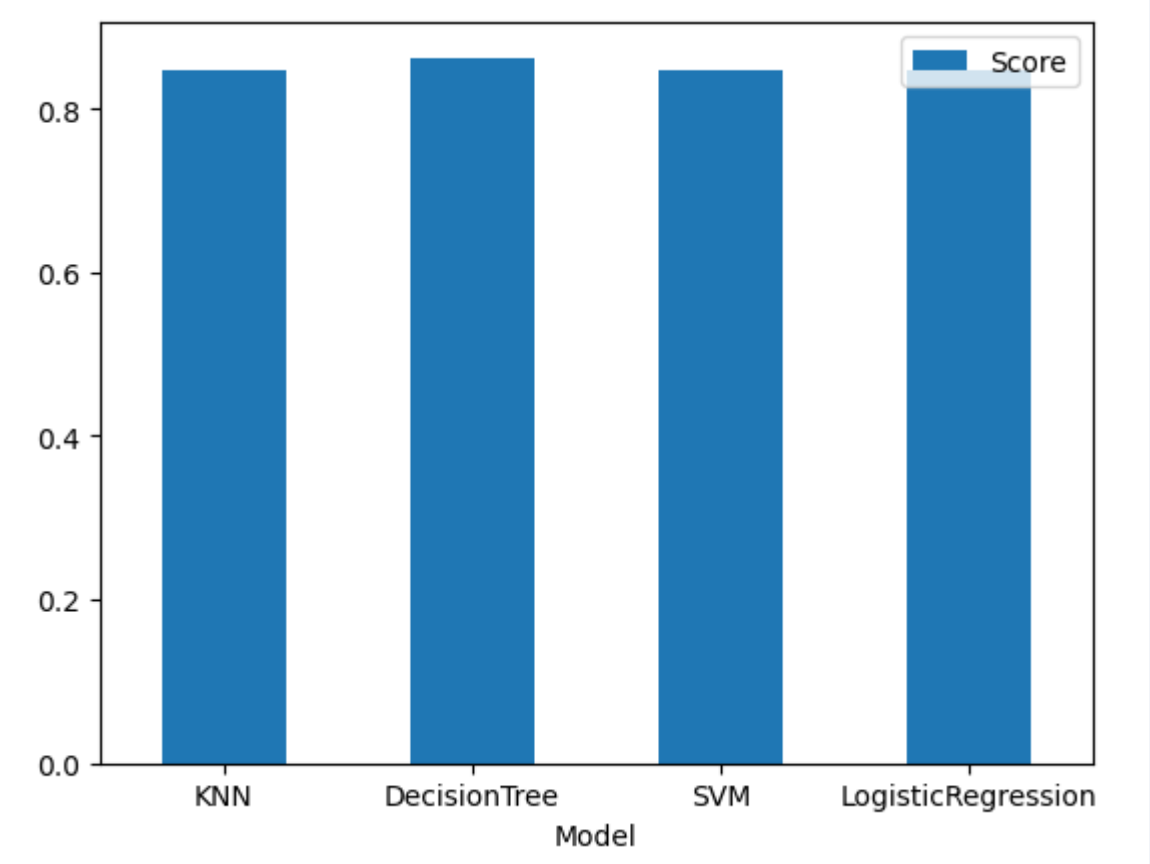
[48]:
```
models = pd.DataFrame({
    'Model': ['KNN','DecisionTree', 'SVM','LogisticRegression'],
    'R-squared Score': [K_acc*100,T_acc*100, S_acc*100, L_acc*100]})
models.sort_values(by='R-squared Score', ascending=False)
```

[48]:

	Model	R-squared Score
1	DecisionTree	88.888889
0	KNN	83.333333
2	SVM	83.333333
3	LogisticRegression	83.333333

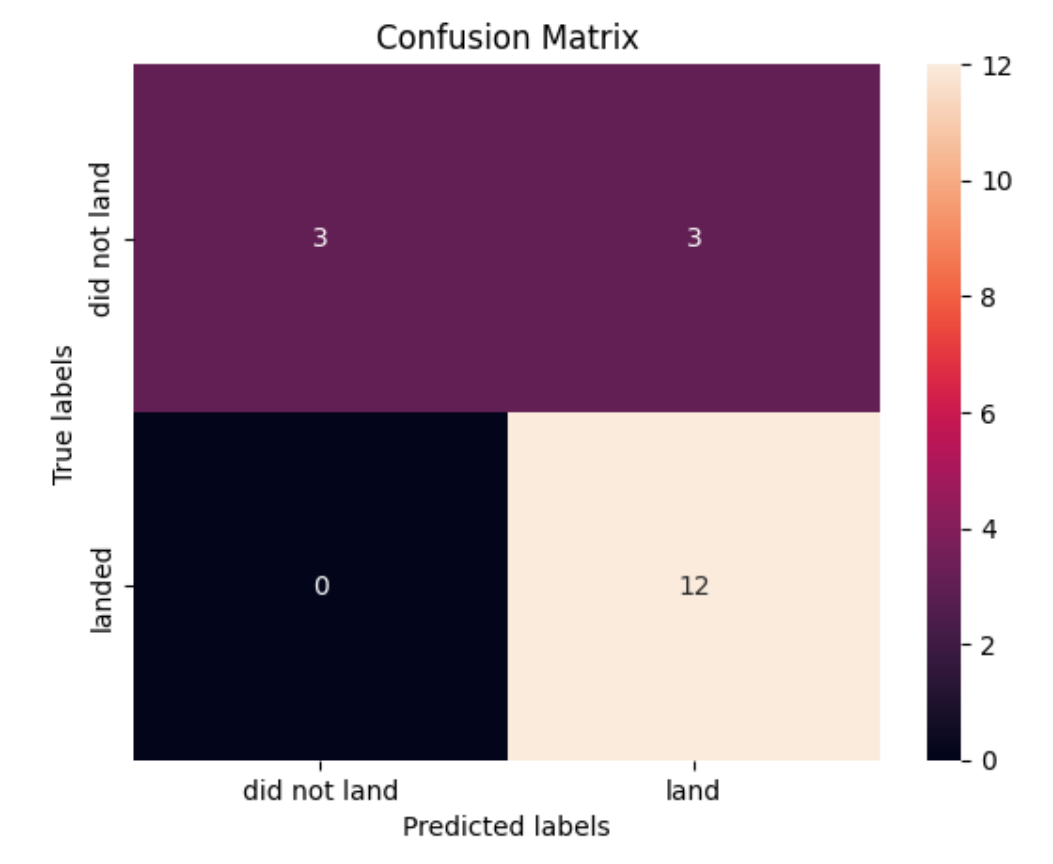
- Comparing the built model accuracy for all built classification models
- Decision Tree model has the highest classification accuracy score of 0.8875

Classification Accuracy



- Bar chart comparing the built model accuracy for all built classification models
- Decision Tree model has the highest classification accuracy score of 0.8875

Confusion Matrix



- Confusion matrix of the best-performing model: Decision Tree Classifier

Confusion Matrix

- **True Positive**

Correctly identified positive cases12. These are the cases where the actual label is positive (landed) and the model correctly predicted it as positive (land).

- **True Negative**

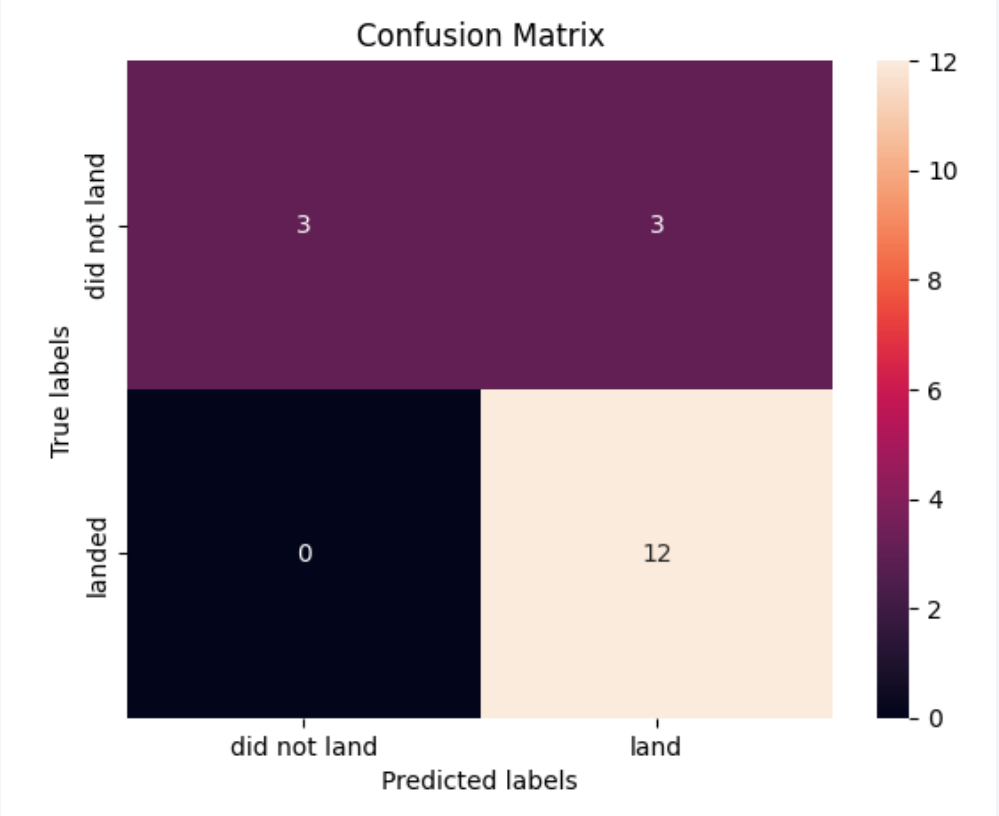
Correctly identified negative cases 3. These are the cases where the actual label is negative (did not land), and the model correctly predicted it as negative (did not land).

- **False Positive**

Incorrectly predicted positive cases equals 3. These are the cases where the actual label is negative (did not land), but the model predicted it as positive (land).

- **False Negative**

Incorrectly predicted negative cases equals 0. In other words, these are the cases where the actual label is positive (landed), but the model predicted it as negative (did not land).



Conclusions

- **Insights drawn from the Exploratory Data Analysis (EDA)**
- As the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
- As the flight number increases, the first stage is more likely to land successfully. The launch site is also important; CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%.
- As the payload mass (kg) increases above 7000, the first stage is more likely to land successfully. Also both KSC LC-39A and VAFB SLC 4E have higher success rate than CCAFS LC-40.
- ES-L1, GEO HEO, and SSO Orbit Type, the first stage is more likely to land successfully (Success rate equals 1.0. GTO Orbit Type has the lowest success rate equal to 0.5.
- With heavy payloads, the successful landing or positive landing rate is higher for Polar, LEO, and ISS. However, for GTO we cannot distinguish this relationship

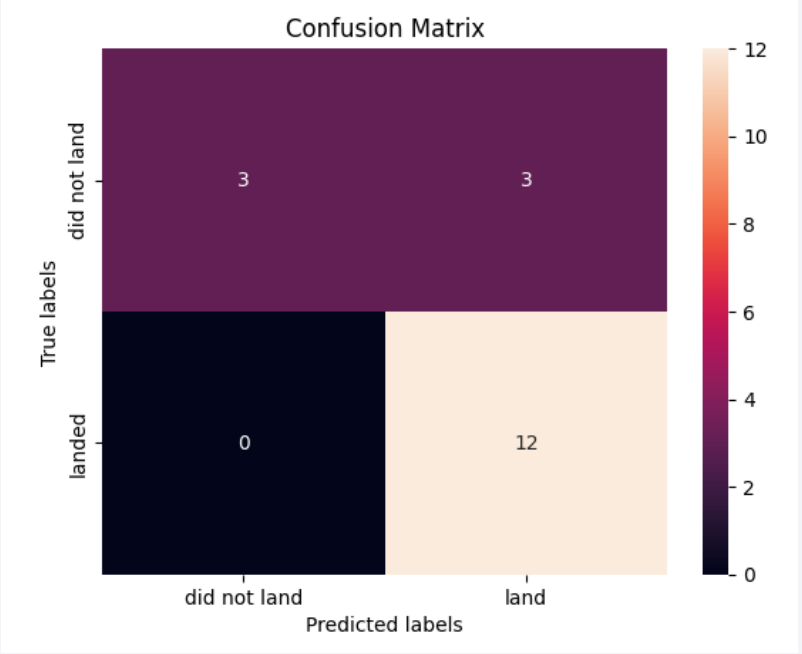
Conclusions

- **Insights drawn from the Proximities Analysis**
- All launch sites are in proximity to the Equator line and very close to the coast. There are two basic reasons behind this (1) reducing the fuel consumption required due to Earth's rotation, (2) safety reasons.

- **Insights drawn from the Interactive Dashboard Process**
- Launch site with the largest successful launches: KSC LC-39A
- Highest launch success rate: KSC LC-39A 76.9%
- Highest launch success rate: 2000-4000
- Lowest launch success rate: 6000-8000
- F9 Booster version with the highest launch success rate: B5 and FT

Conclusions

- **Insights drawn from the Predictive Analysis (Classification)**
- Decision Tree model has the highest classification accuracy score of 0.8875
- Confusion matrix of the best-performing model: Decision Tree Classifier
- True Positive Correctly identified positive cases12.
- True Negative Correctly identified negative cases 3.
- False Positive Incorrectly predicted positive cases equals 3.
- False Negative Incorrectly predicted negative cases equals 0.



Appendix

- Capstone-SpaceX include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets created during this project:
- https://github.com/ErnestBol/Capstone-SpaceX

Thank you!