IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)

- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- In this capstone project, our goal is to predict the successful landing of the Falcon 9 first stage. SpaceX promotes Falcon 9 rocket launches on its website at a cost of $62 million, significantly cheaper than other providers whose costs exceed $165 million each. This cost discrepancy is largely due to SpaceX's ability to reuse the first stage of the rocket. Therefore, accurately determining the success of the first stage landing directly impacts the overall cost of a launch. This information becomes crucial in scenarios where alternate companies aim to bid against SpaceX for rocket launches. The ability to predict the outcome of the first stage landing not only ensures cost efficiency but also enhances competitiveness within the space launch market.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Use SpaceX Rest API
  - Use Web Scrapping from Wikipedia
- Perform data wrangling
  - Deal with missing values
  - Use One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune and evaluate four classification models

Data Collection

To collect Falcon 9 historical launch records from the Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches," we utilized web scraping techniques with BeautifulSoup. First, we requested the Falcon 9 Launch Wiki page from its URL. Then, we extracted all column/variable names from the HTML table header to understand the structure of the data. With this information, we created a data frame by parsing the launch HTML tables, ensuring accurate representation of the launch records. This method allowed for systematic retrieval and organization of Falcon 9 launch data from the specified Wikipedia page snapshot dated June 9, 2021.

Data Collection – SpaceX API

1) Request rocket launch data from SpaceX API using a specific URL.
2) Decode the response content as JSON using .json().
3) Convert it into a Pandas dataframe using .json_normalize().
4) Utilize the API again to retrieve additional information about the launches using the IDs provided for each launch.
5) Store the extracted data in lists to construct a new dataframe.
6) Create a dataset by combining the columns into a dictionary.
7) Generate a Pandas dataframe from the dictionary.
8) Filter the dataframe to only include Falcon 9 launches.
9) Address missing values by replacing np.nan values with the calculated mean values.

GitHub URL : 1_SpaceX_Data_Collection_API.ipynb

Data Collection - Scraping

1) Request the Falcon9 Launch Wiki page from its URL by performing an HTTP GET method using requests.get() with the provided static URL.
2) Assign the response to an object.
3) Extract all column/variable names from the HTML table header.
4) Create a dataframe by parsing the launch HTML tables. Begin by creating an empty dictionary with keys extracted from the column names obtained.
5) Populate the launch dictionary with launch records extracted from table rows, filling it with the necessary data.
6) Convert the populated dictionary into a Pandas dataframe for further analysis and manipulation.

GitHub URL : 2_SpaceX_Webscraping.ipynb

Data Wrangling

We conduct Exploratory Data Analysis (EDA) to discern patterns within the dataset and establish training labels for supervised models. The dataset encompasses various scenarios where the Falcon 9 booster did not achieve successful landings, ranging from successful oceanic landings to failed attempts. Through meticulous examination, we aim to categorize these outcomes into binary training labels, assigning a value of 1 for successful landings and 0 for unsuccessful ones.

1) Calculate the number of launches on each launch site.
2) Determine the number and occurrence of each orbit.
3) Calculate the number and occurrence of mission outcomes associated with each orbit.
4) Create a landing outcome label from the "Outcome" column, where a value of zero indicates an unsuccessful landing of the first stage, while a value of one represents a successful landing.
5) Export the dataset, including the newly created landing outcome label, to a CSV file for further analysis or sharing.

GitHub URL : 3_SpaceX_Data_Wrangling.ipynb

EDA with Data Visualization

1. We want to visually check if there are any relationship between
   - Flight Number and Launch Site
   - Payload and Launch Site
   - success rate of each orbit type
   - Flight Number and Orbit type
   - Payload and Orbit type
2. Visualize the launch success yearly trend to to get the average launch success trend

GitHub URL : 4_SpaceX_EDA_Data_Visualization.ipynb

EDA with SQL

- Load the SpaceX dataset, load the SQL extension and establish a connection with the database.
- Create a table named SPACEXTABLE by selecting all records from SPACEXTBL with non null values.
- Execute the following queries:

1. Display the names of the unique launch sites in the space mission.
2. Display 5 records where launch sites begin with the string 'CCA'.
3. Display the total payload mass carried by boosters launched by NASA (CRS).
4. Display the average payload mass carried by booster version F9 v1.1.
5. List the date when the first successful landing outcome on a ground pad was achieved.
6. List the names of the boosters which have successfully landed on a drone ship and have a payload mass between 4000 and 6000.
7. List the total number of successful and failure mission outcomes.
8. List the names of the booster versions which have carried the maximum payload mass using a subquery.
9. List the records displaying the month names, failure landing outcomes in a drone ship, booster versions, and launch sites for the months in the year 2015.
10. Rank the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order.

GitHub URL : 5_SpaceX_EDA_SQL.ipynb

Build an Interactive Map with Folium

- Add each launch site's location on a map using its latitude and longitude coordinates. Create and add folium.Circle and folium.Marker for each launch site on the site map.
- Mark the success or failure of launches for each site on the map. Create a MarkerCluster object and generate markers for all launch records. Use green markers for successful launches (class=1) and red markers for failed launches (class=0). Add a folium.Marker to MarkerCluster for each launch result in the spacexx dataframe.
- Calculate the distances between a launch site and its proximities. Add a MousePosition on the map to obtain coordinates when hovering over a point. This enables easy identification of points of interest, such as a railway. Calculate the distance between a selected location and a given launch site. Draw a PolyLine between a launch site and the selected point to analyze geographical patterns about launch sites.

GitHub URL : 6_SpaceX_Launch_Site_Location_Folium.ipynb

Build a Dashboard with Plotly Dash

- Implement a dropdown list to facilitate Launch Site selection, allowing users to choose from available options.
- Integrate a pie chart to visualize the total count of successful launches across all sites. Additionally, for a selected Launch Site, display a breakdown of Success vs. Failed counts.
- Include a slider component enabling users to select a Payload Mass range, providing flexibility in filtering data based on Payload criteria.
- Develop a scatter chart illustrating the relationship between Payload Mass and Launch Success for various Booster Versions. This visualization aids in understanding any potential correlation between Payload Mass and Launch Success rates across different versions of the booster.

GitHub URL : 7_SpaceX_Dash_App.py , 7_SpaceX_Dashboard.png

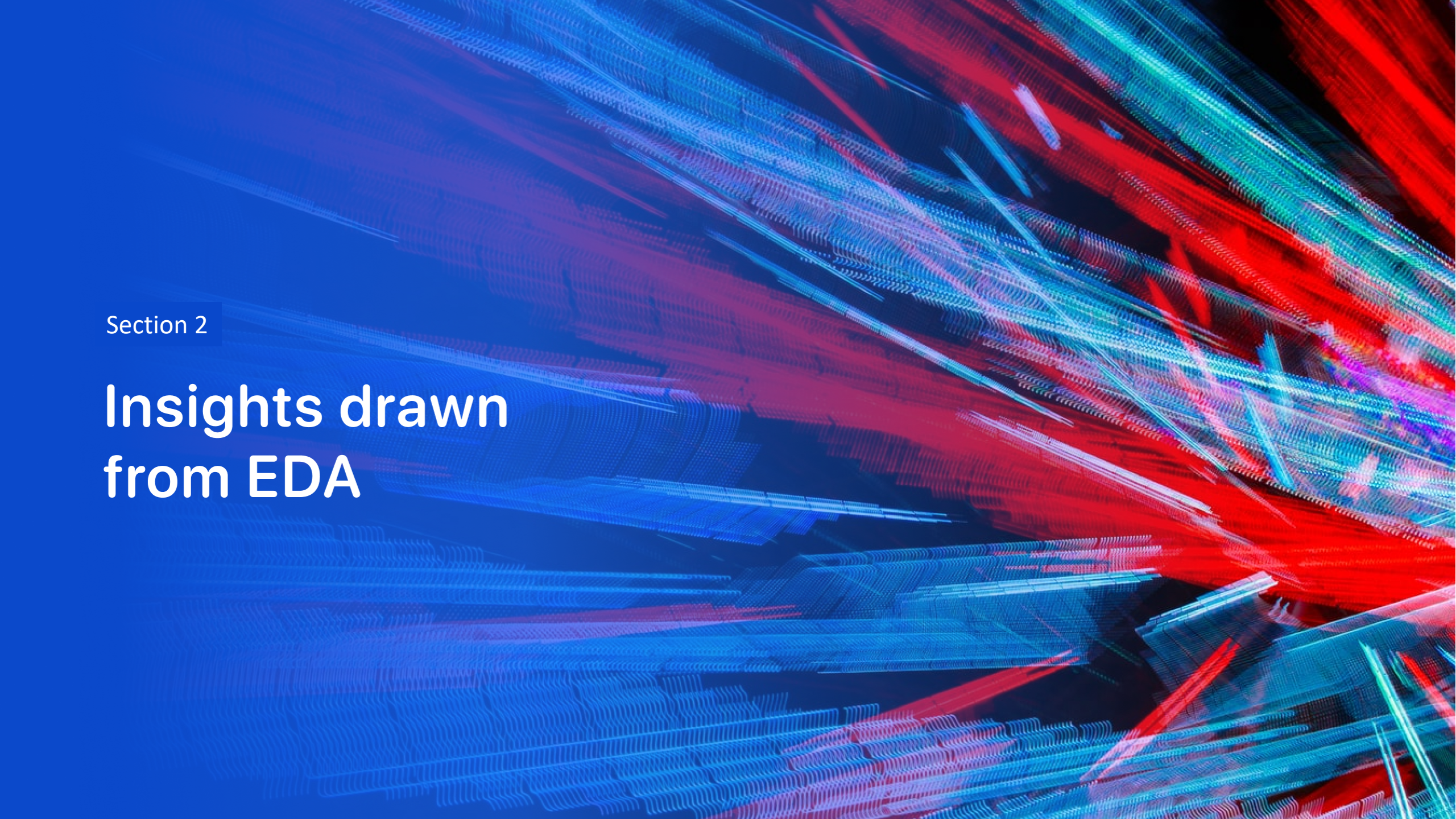
Predictive Analysis (Classification)

1) Create a NumPy array from the column Class in the data dataframe by applying the method to_numpy() and assign it to the variable Y. Ensure the output is a Pandas series.
2) Standardize the data in X and reassign it to the variable X_scaled.
3) Split the data into training and testing sets using the train_test_split function.
4) Use ML models to train and select hyperparameters using the GridSearchCV function. Create a logistic regression object and a GridSearchCV object; Fit the object to find the best parameters for logistic regression model. Repeat this process for support vector machine, decision tree classifier, and k-nearest neighbors.
5) Check the confusion matrix and accuracy score for each of these ML methods and compare the results.

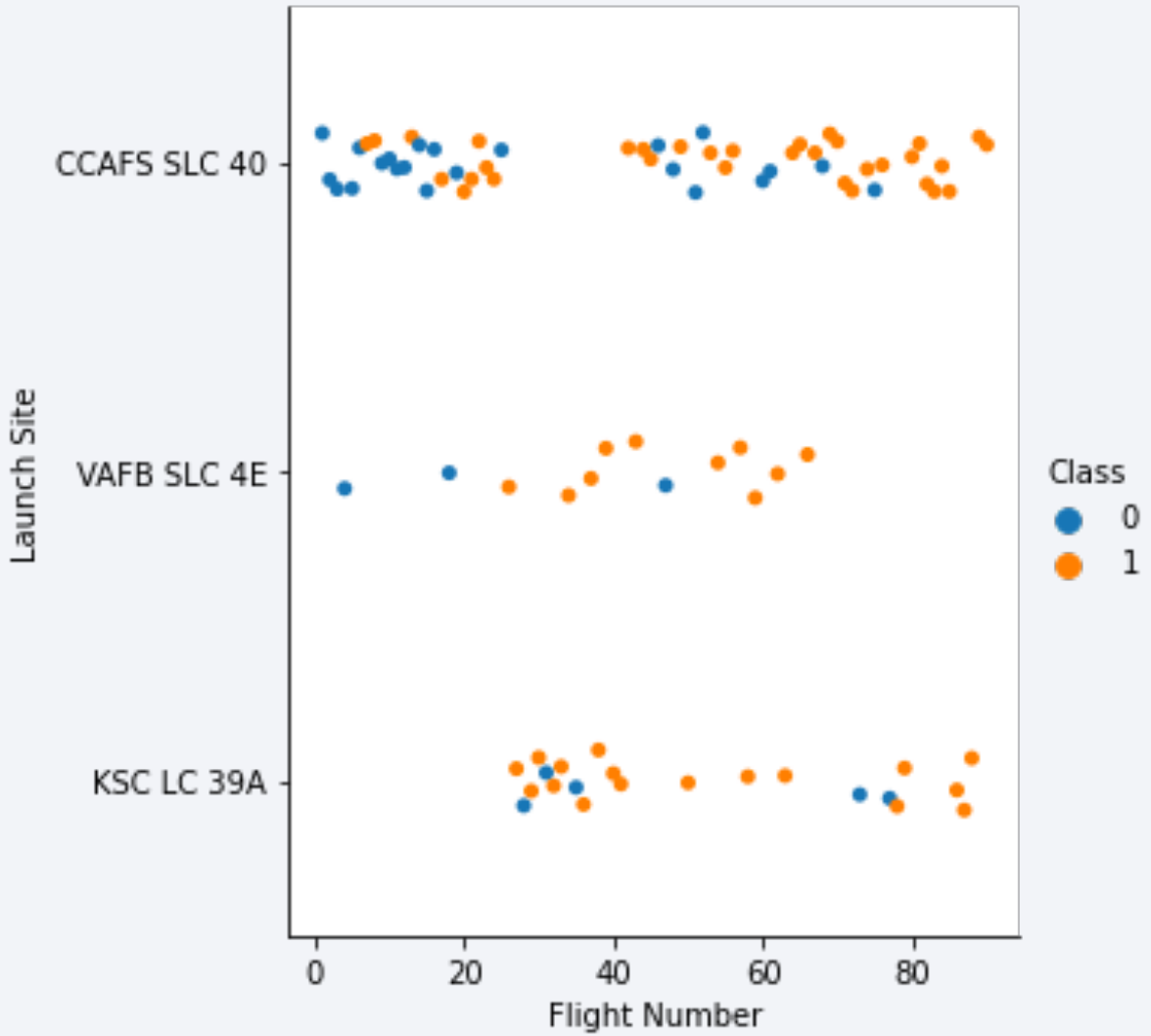
GitHub URL : 8_SpaceX_ML_Prediction.ipynb

Results

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

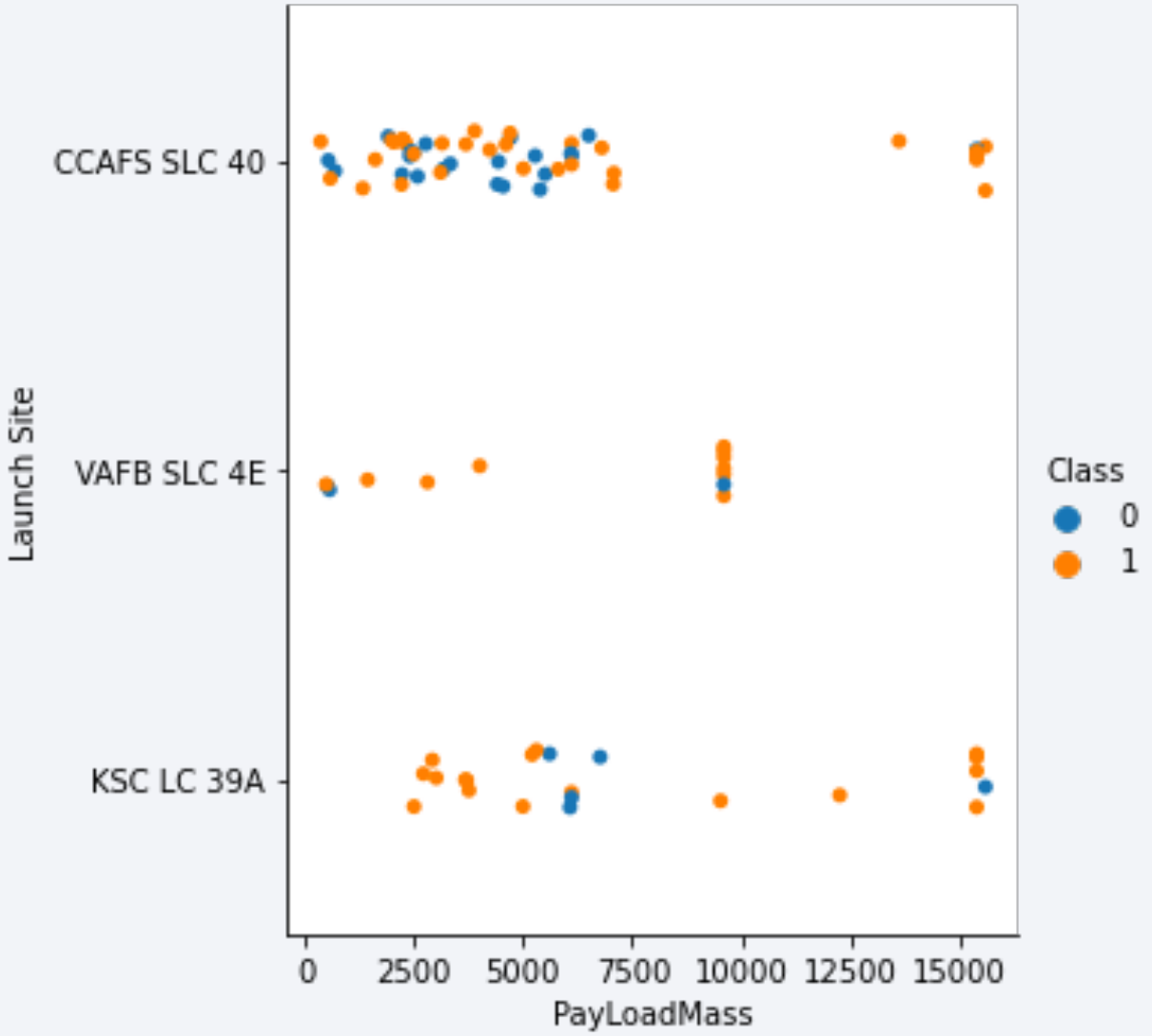

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



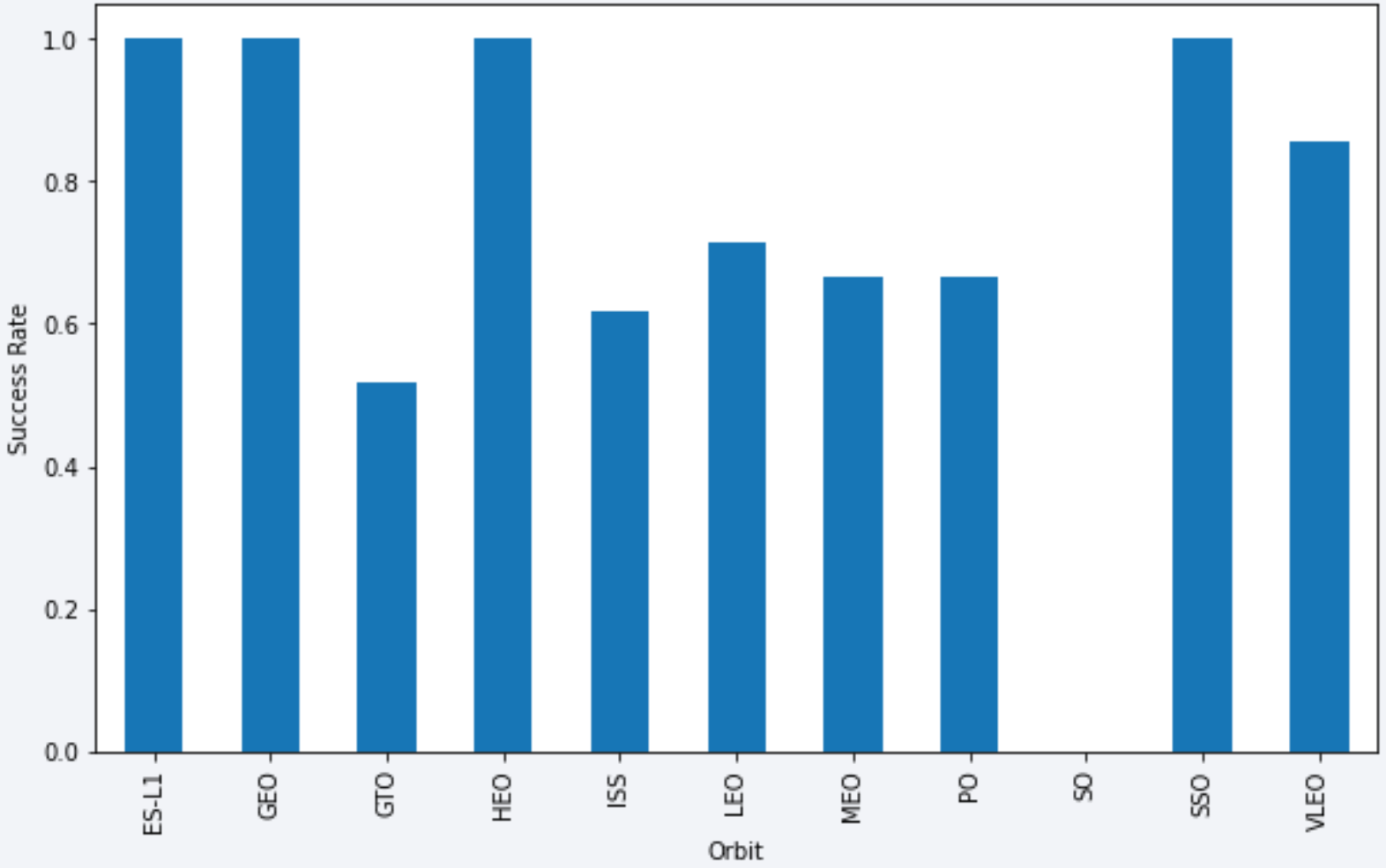
- For CCAFS-SLC, there seems to be no relationship between the success and the number of flights.
- For CCAFS-SLC, the success appears related to the number of flights. Specifically, flight numbers greater than 20 mostly succeed.
- At KSC-LC, flight numbers smaller than 35 do not appear to have a relationship with success.
- All rockets launched at KSC-LC with flight numbers between 35 and 65, as well as those greater than 80, succeed.

Payload vs. Launch Site



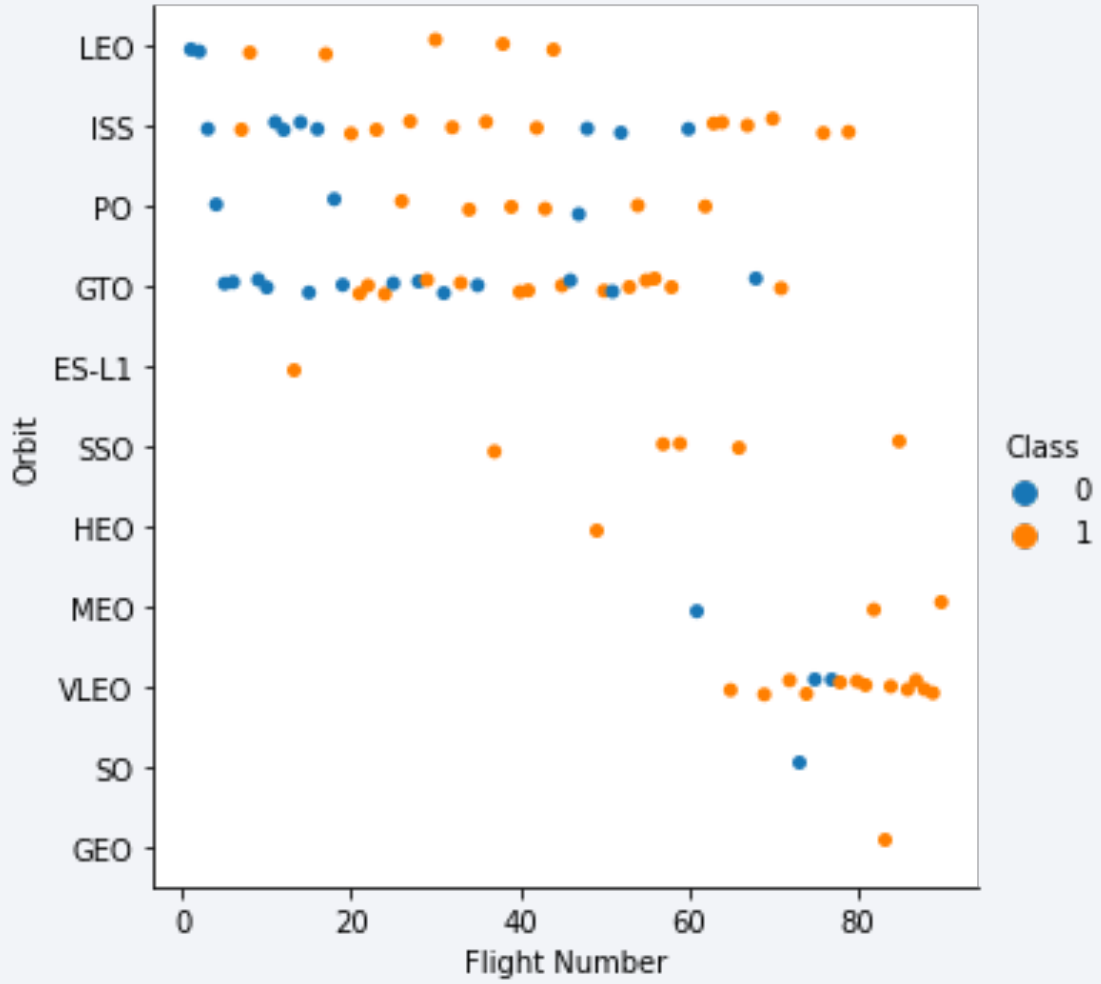
- Most of the launches with pay load mass over 7500 are successful.
- At CCAFS-SLC, most rockets launched for payload masses less than 7500 have approximately a 50% success rate. However, there are a few rockets launched for payload masses greater than 12500, with a noticeable higher success rate.
- Rockets launched at VAFB-SLC with payload masses ranging from 1000 to 4000 all succeed. There is also a focus on payload masses around 9000 for VAFB-SLC, with a high success rate.
- There are no rockets launched for heavy payload masses greater than 10000 at VAFB-SLC.
- Most failed rockets launched at KSC-LC have payload masses around 6000-7000. Otherwise, they mostly succeed.

Success Rate vs. Orbit Type



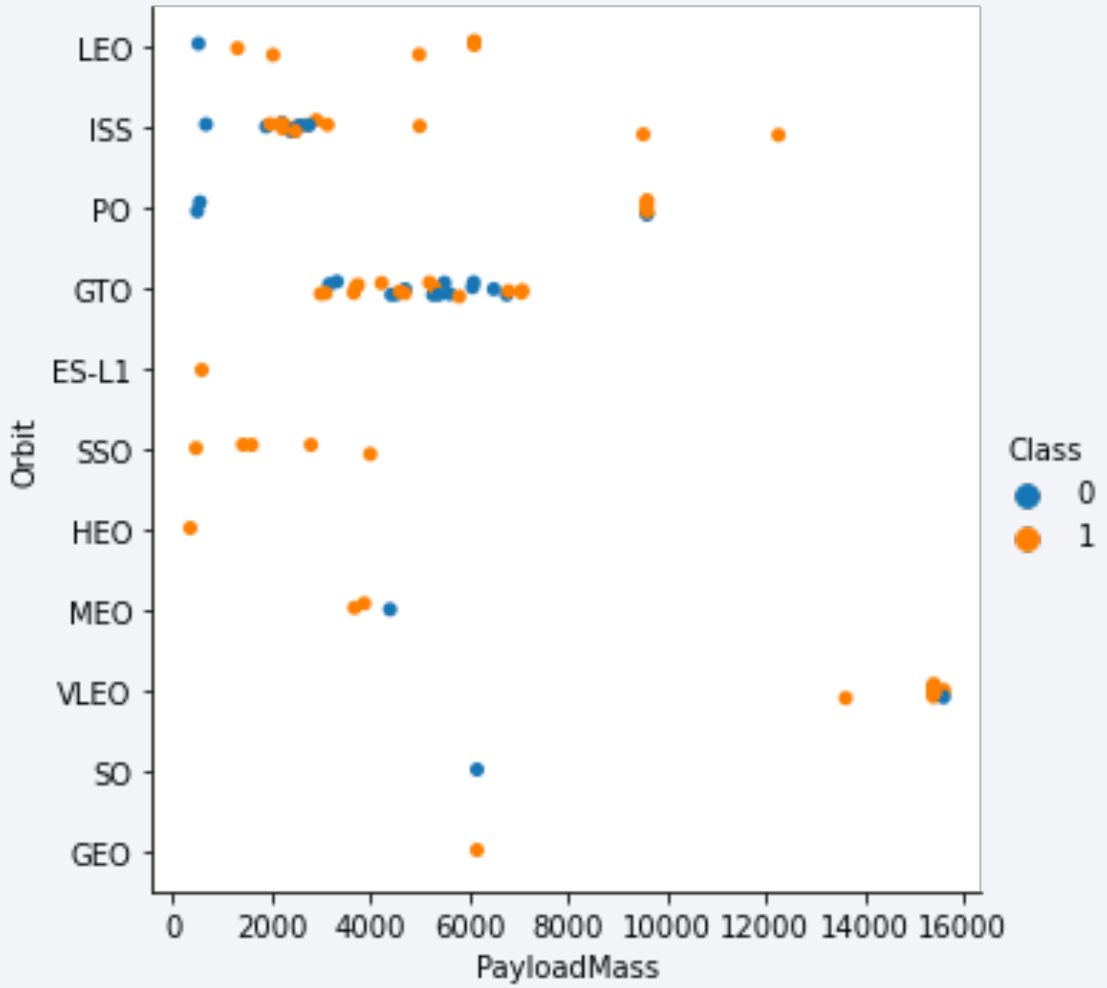
- ES-L1, GEO, HEO and SSO orbits have 100% success rates, while SO has 0% success rates.
- VLEO has the second highest success rates greater than 85%.
- ISS, LEO , MEO and PO have a medium success rates between 60% to 70%.
- GTO orbit has the second lowest success rate.

Flight Number vs. Orbit Type



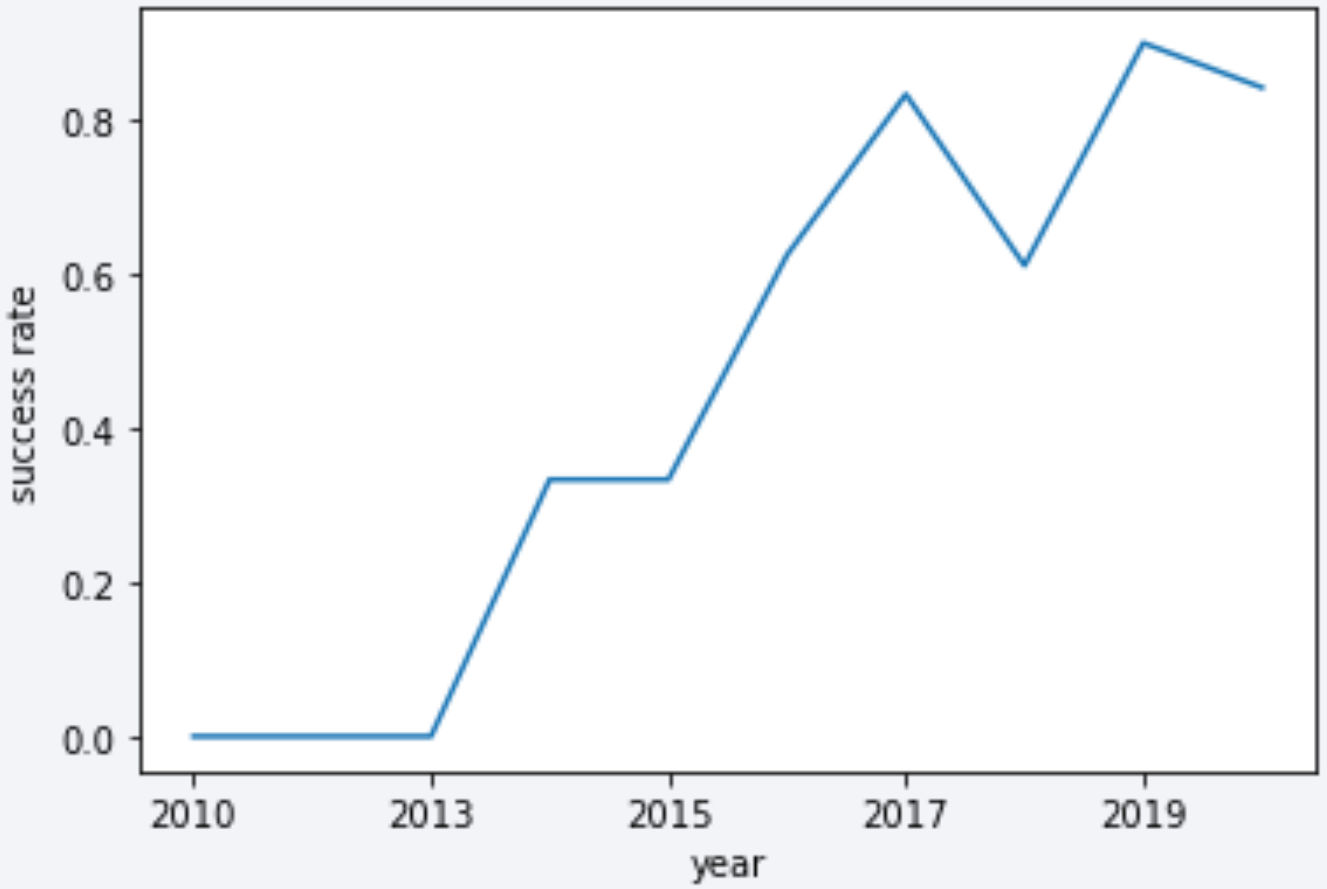
- In the LEO orbit, success appears related to the number of flights, with flight numbers greater than 5 resulting in success.
- In the VLEO orbit, success is observed for flight numbers smaller than 75 and larger than 80.
- There seems to be no relationship between success and flight number in the GTO orbit.
- SSO orbit results in success independent of the flight number.
- For ISS and PO orbits, successful launches are observed for flight numbers between 20 and 45, as well as those larger than 60.

Payload vs. Orbit Type



- With heavy payloads, the successful landing is higher for PO, LEO, and ISS orbits.
- However, for GTO, distinguishing between successful landing rate and unsuccessful landing is challenging as both outcomes occur frequently.
- SSO orbit results in success independent of the payloads.
- For MEO and VLEO orbits, most successful launches are concentrated on payload masses of 4000 and 15000, respectively.

Launch Success Yearly Trend



- The success rate keeps constant from 2010 to 2013.
- The success rate since 2013 keeps global increasing till 2020.
- The success rate keeps constant from 2014 to 2015.
- A sudden decrease occurs from 2017 to 2018.

All Launch Site Names

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

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
✓ 0.0s

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

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

Launch Site Names Begin with 'CCA'

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

```
%%sql
SELECT * FROM SPACEXTABLE
WHERE Launch_Site like 'CCA%'
LIMIT 5
```

✓ 0.0s

```
 * 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

Total Payload Mass

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)'
```

✓ 0.0s

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Booster_Version like 'F9 v1.1%'
```

✓ 0.0s

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

List the date when the first successful landing outcome on a ground pad was achieved.

```
%%sql
SELECT MIN(date) FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
```

✓ 0.0s

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

MIN(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have successfully landed on a drone ship and have a payload mass between 4000 and 6000.

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)' AND
      PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

✓ 0.0s

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

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes.

```
%%sql
SELECT COUNT(*) FROM SPACEXTABLE
WHERE Mission_Outcome like 'Success%'
```

✓ 0.0s

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

COUNT(*)
100

```
%%sql
SELECT COUNT(*) FROM SPACEXTABLE
WHERE Mission_Outcome like 'Failure%'
```

✓ 0.0s

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

COUNT(*)
1

Boosters Carried Maximum Payload

List the names of the booster versions which have carried the maximum payload mass using a subquery.

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

✓ 0.0s

```
 * 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

2015 Launch Records

List the records displaying the month names, failure landing outcomes in a drone ship, booster versions, and launch sites for the months in the year 2015.

```
%%sql
SELECT substr(date,0,5) AS Year, substr(date,6,2) AS Month , Booster_Version , Launch_Site, Landing_Outcome
From SPACEXTABLE
WHERE Landing_Outcome  = 'Failure (drone ship)' and YEAR = '2015'
```

✓ 0.0s

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

Year	Month	Booster_Version	Launch_Site	Landing_Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Rank the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order.

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

✓ 0.0s

```
 * 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


Section 3
Launch Sites
Proximities Analysis

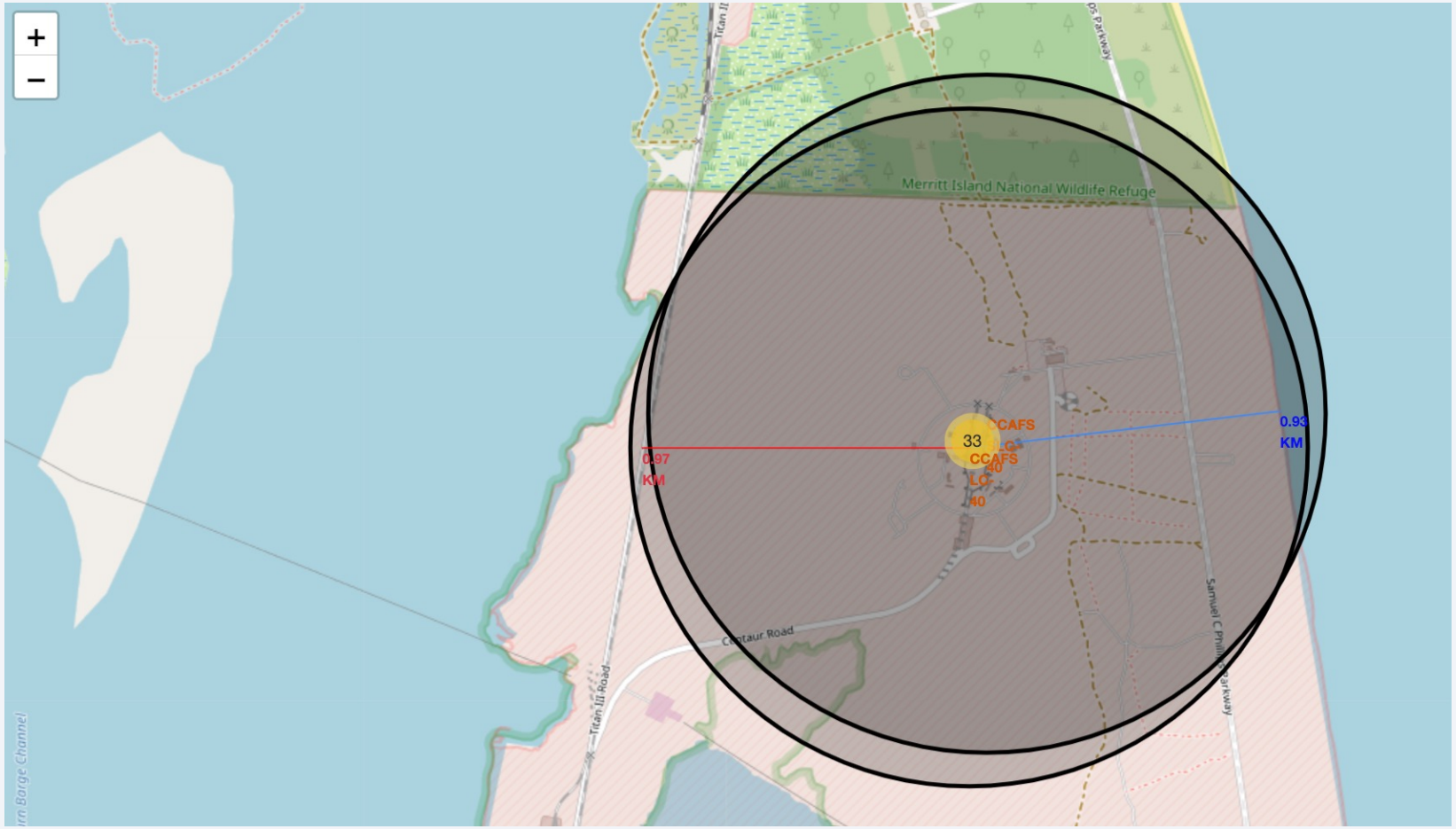
Marking All Launch Sites on a Map

Upon observation, it is evident that all launch sites are located near the Equator line and close to coastlines. This strategic positioning is deliberate, as the Equator provides a natural advantage for rocket launches by harnessing the Earth's rotational speed, thus enabling more efficient launches. Additionally, coastal locations offer safety benefits by providing a controlled environment for launches over water, reducing the risk to populated areas in the event of malfunctions.

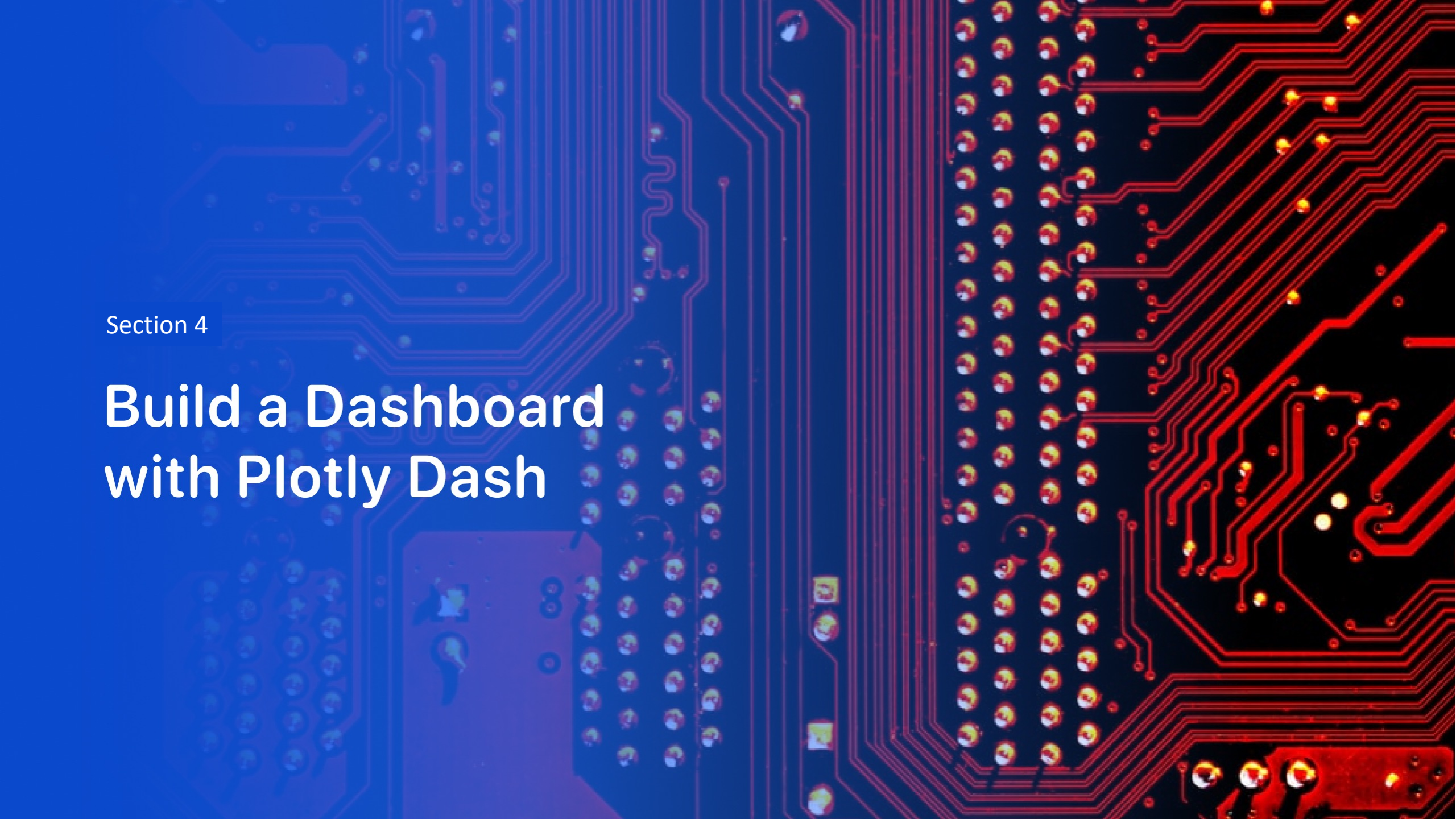
Displaying Success/Failure of Launches for Each Site on the Map

The color-coded markers offer a straightforward way to visually represent the success rates across various launch sites. Green markers indicate successful launches, while red markers represent failed ones. Users can simply click on each launch site to visualize its success rate. For instance, by clicking on Launch Site KSC LC-39A, users can readily identify sites with particularly high success rates, facilitating quick analysis.

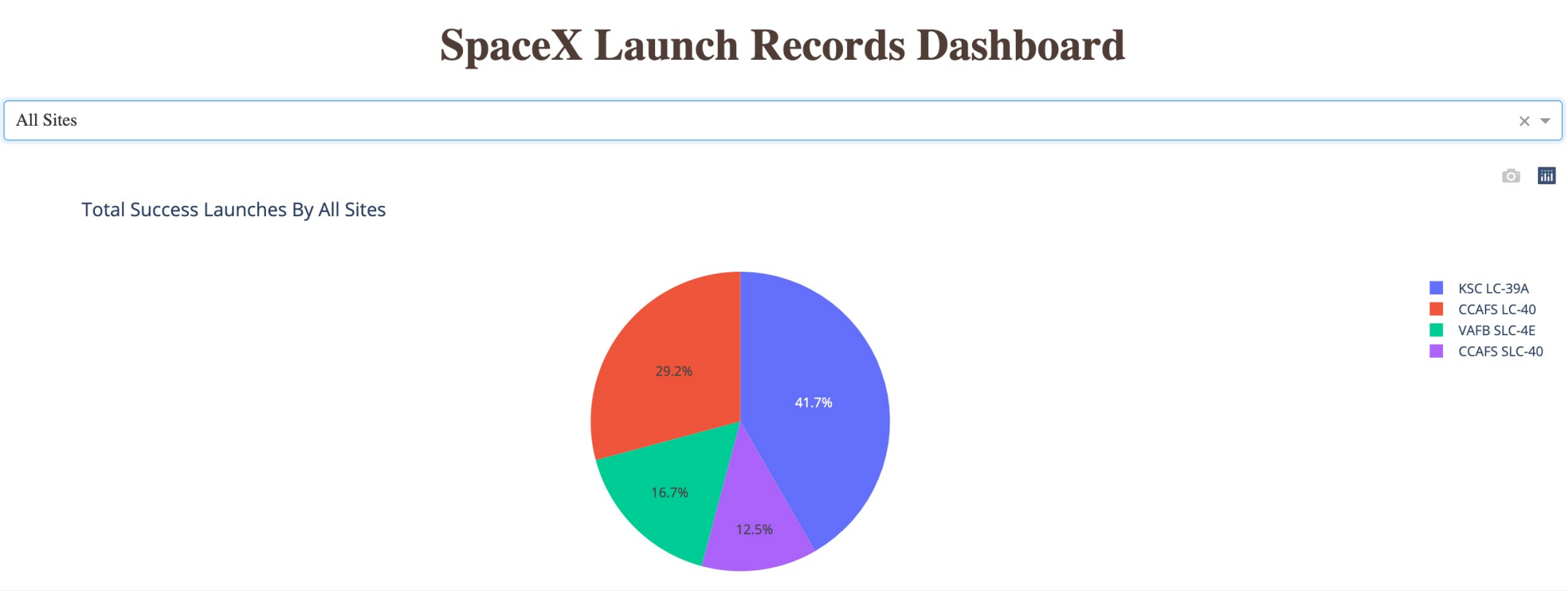
Calculating Distances and Drawing Polylines between Launch Sites and Proximities



Based on visual analysis, the launch site CCAFS-LC-40 is situated in close proximity to a coastline, approximately 0.93 kilometers away, as well as a railway, approximately 0.97 kilometers away.

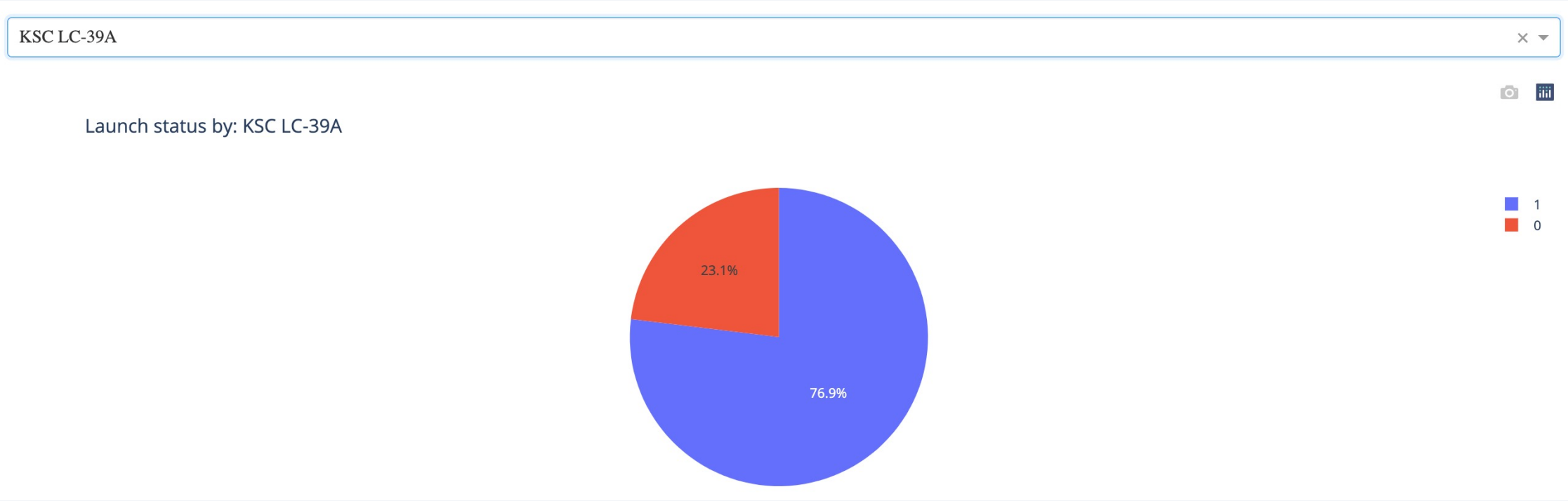

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



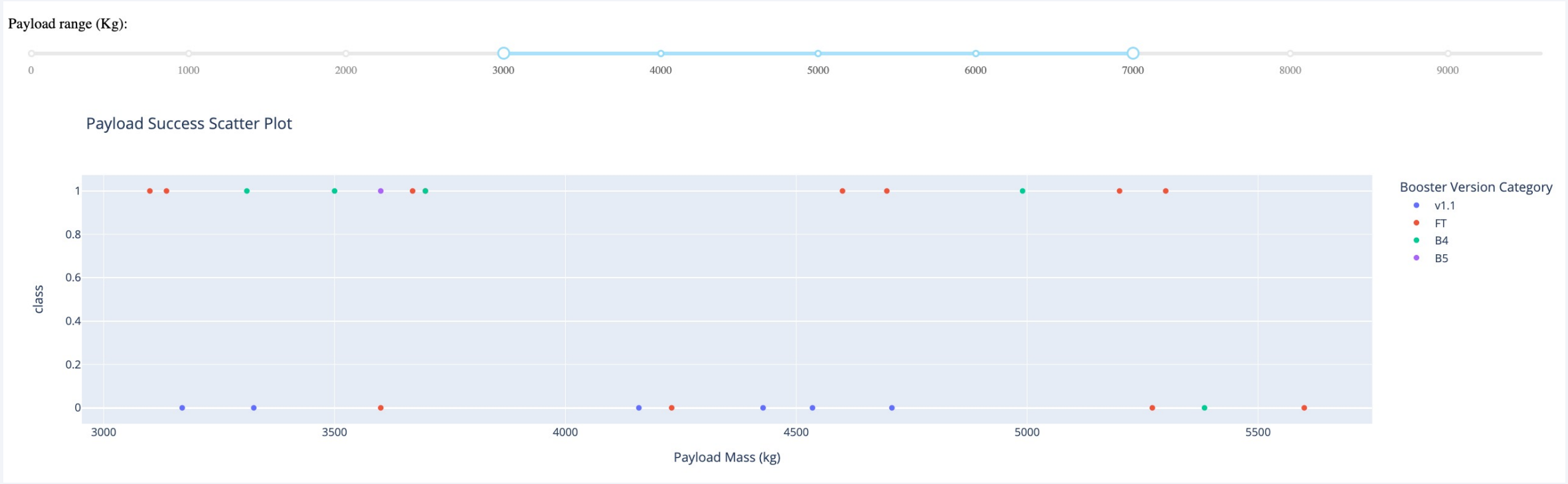
KSC LC-39A has the highest number of successful launches, while CCAFS SLC-40 has the lowest.

Launch Site with Highest Launch Success Ratio



KSC LC-39A has the highest successful launches ratio 76.9%.

Payload vs. Launch Outcome Scatter Plot for All Sites

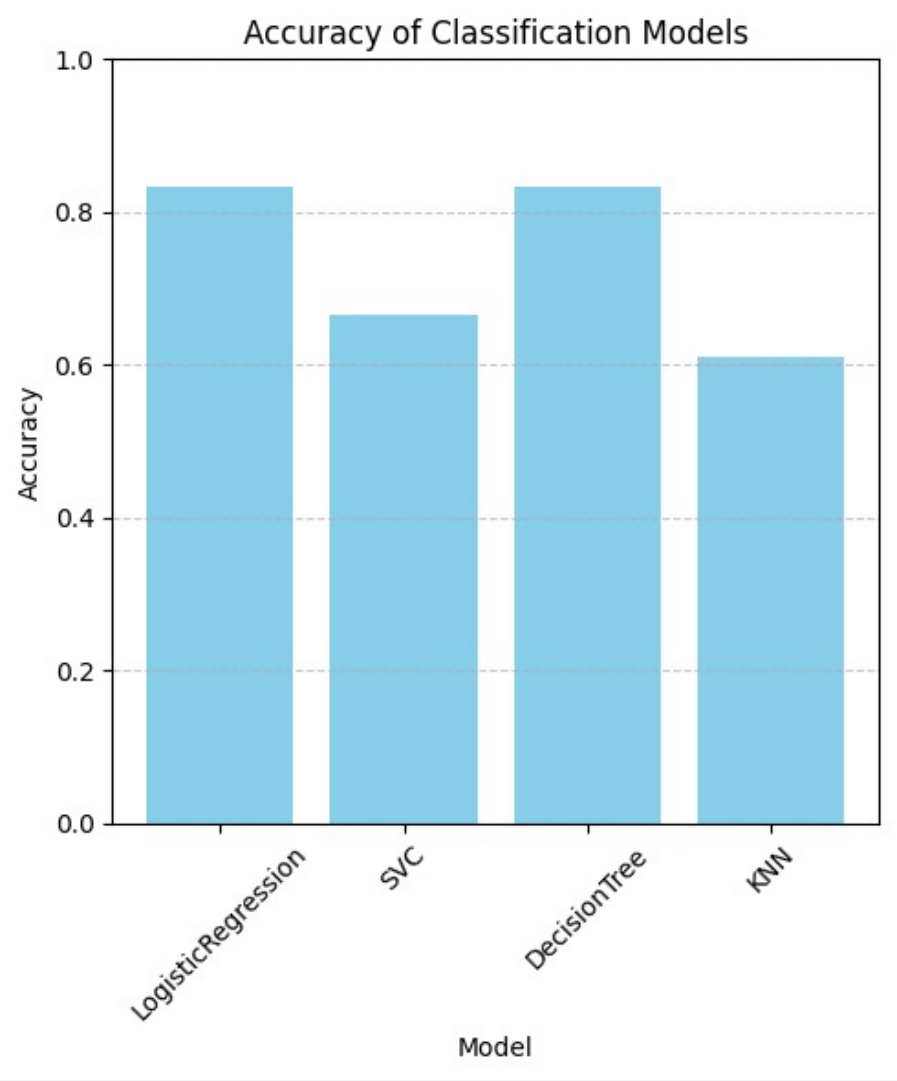


Scatter plot of the class versus payload mass in the range of 3000 kg to 7000 kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy




```
MLscore = pd.DataFrame({
    'Model': ['LogisticRegression', 'SVC', 'DecisionTree', 'KNN'],
    'Accuracy': [lr_score, svm_score, tree_score, knn_score]
})
MLscore
```

✓ 0.0s

	Model	Accuracy
0	LogisticRegression	0.833333
1	SVC	0.666667
2	DecisionTree	0.833333
3	KNN	0.611111

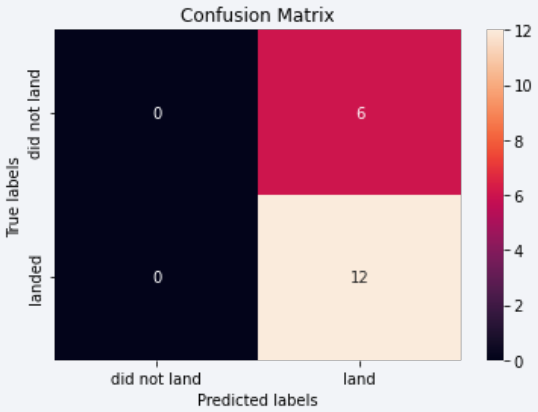
Logistic Regression and Decision Tree model has the highest classification accuracy.

Confusion Matrix

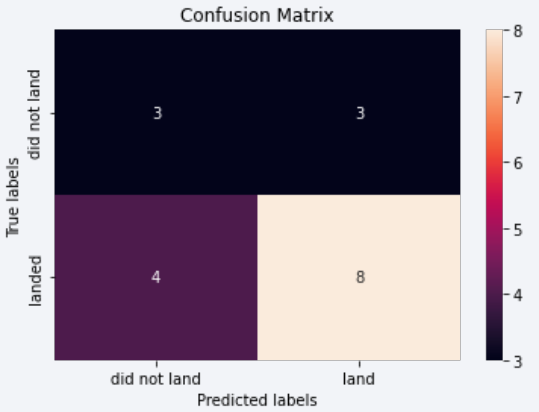
Logistic Regression

SVC

Decision Tree



KNN



	True Negatives	False Positives	False Negatives	True Positives
LogisticRegression	3	3	0	12
SVC	0	6	0	12
DecisionTree	0	6	0	12
KNN	3	3	4	8

Method	Precision	Recall	F1-Score
LogisticRegression	0.800000	1.000000	0.888889
SVC	0.666667	1.000000	0.800000
DecisionTree	0.666667	1.000000	0.800000
KNN	0.727273	0.666667	0.695652

Conclusions

- Logistic Regression and Decision Tree had the highest precision, both around 80.00%, indicating a high proportion of correct positive predictions. SVC and KNN had slightly lower precision at around 66.67%.
- All methods except KNN achieved perfect recall (100%), indicating they correctly identified all positive instances. KNN had a recall of approximately 66.67%, suggesting it missed around 33.33% of positive instances.
- Logistic Regression achieved the highest F1-score at approximately 88.89%, indicating a good balance between precision and recall. SVC and Decision Tree followed with F1-scores around 80.00%. KNN had the lowest F1-score among the methods, approximately 69.57%.
- Logistic Regression appears to perform the best based on these metrics, with high accuracy, precision, recall, and F1-score. SVC and Decision Tree also perform reasonably well, while KNN lags behind slightly in terms of overall performance.

Thank you!