IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexis
Porzolis
13.10.2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collected via API & Web Scraping
  - Data cleaning, one-hot encoding, normalization
  - Exploratory data analysis performed
  - Machine Learning models were built
  - Hyperparameter tuning for improved accuracy
- Summary of all results
  - The optimal model was acquired
  - Visualizations were great for decision making

Introduction

- Project background and context

  The goal of this project is to predict the landings of the first stage of SpaceX's Falcon 9 rocket. The Falcon 9 is a reusable rocket designed to reduce space transportation costs by recovering and reusing the rocket's first stage. Since its first successful launch in 2010, SpaceX has continually worked to perfect these landings, both on sea-based platforms and land-based pads.

  Predicting whether a landing will be successful is critical from both an economic and operational standpoint. Accurate predictions can help allocate resources more efficiently, reduce risks, and increase the likelihood of successfully recovering the rocket stage. This project analyzes historical launch and landing data to identify patterns and develop machine learning models capable of predicting future landing outcomes.

  The project leverages publicly available SpaceX data along with weather information to uncover the factors that most influence
  landing success.

- Problems you want to find answers

  1. What factors influence the success or failure of a specific space mission project?
  2. How can the model be best optimized for classifying space mission outcomes?
  3. What patterns can be identified from SpaceX's launch and mission data?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Open Source Rest API
  - Web Scrapping from Wikipedia page ‘List of Falcon 9 and Falcon Heavy Launches’
- Perform data wrangling
  - Data were transformed and one-hot encoded to be apply later on Machine learning models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification machine learning models were built to achieve this goal

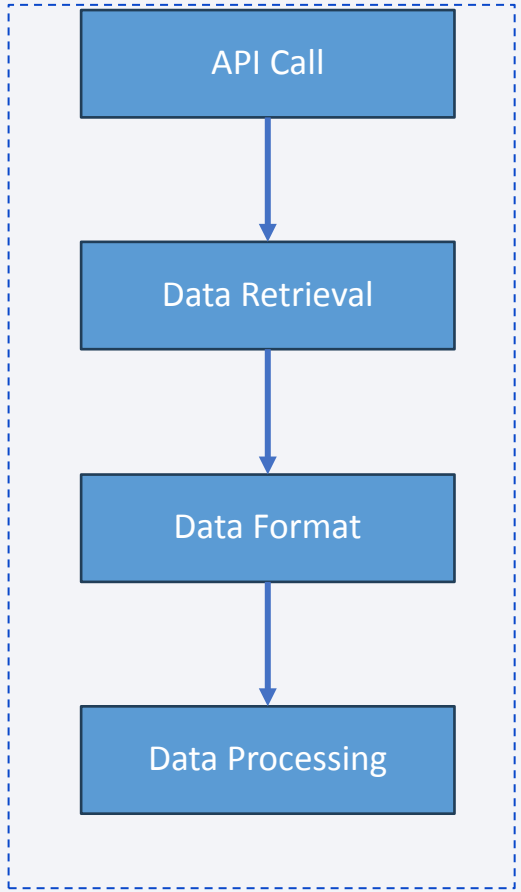
Data Collection

The data was collected using various methods

- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

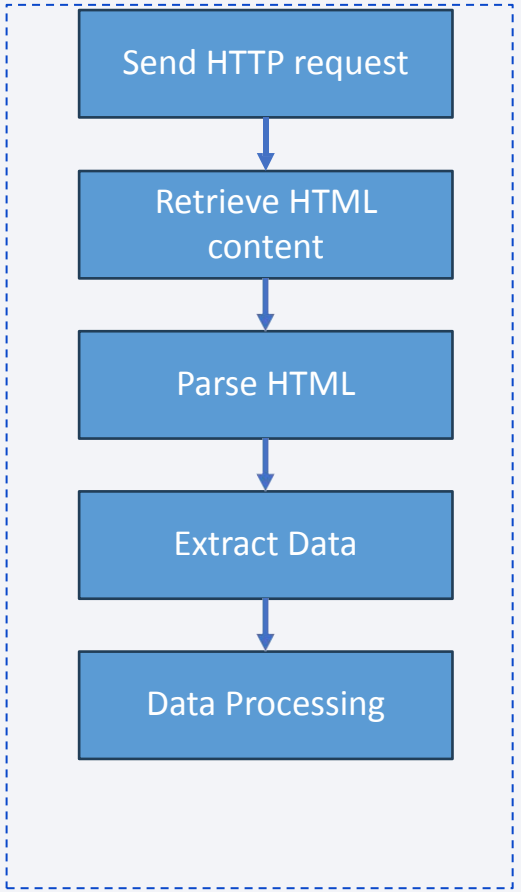
- API Endpoint: Used SpaceX API endpoint /v4/launches to retrieve rocket launch data.
- Request: Fetched data via an HTTP GET request using Python's requests library.
- Data Format: The API returns data in JSON format, including information such as launch date, rocket name, payload, and success status.
- Data Processing: The JSON data is converted into a DataFrame for further analysis (e.g., cleaning and filtering).



Github Link: SpaceX Data Collection with API

Data Collection – Scraping

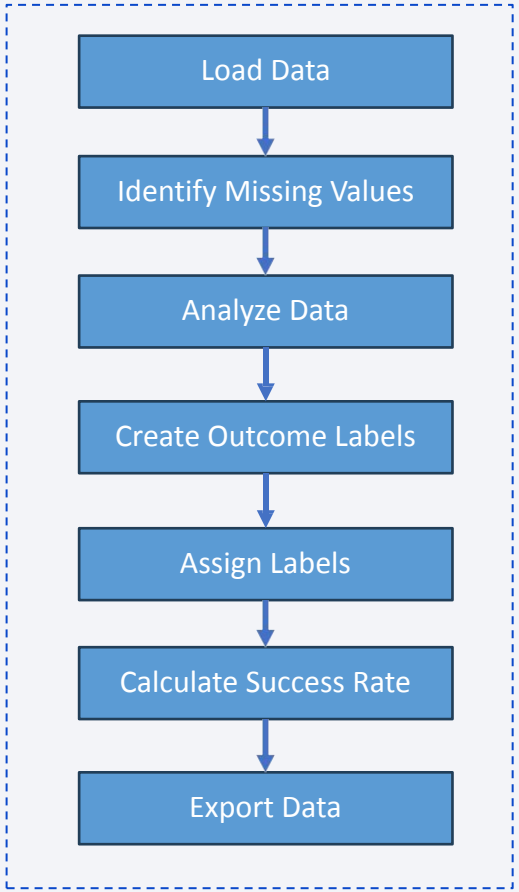
- Send HTTP request: Access the desired website (e.g., Wikipedia page).
- Retrieve HTML content: Load the entire HTML structure of the webpage.
- Parse HTML: Use libraries like BeautifulSoup to process the HTML data.
- Extract data: Extract specific data (e.g., tables or text) from the HTML content.
- Data processing: Convert the extracted data into a DataFrame, clean, and filter it



Github Link: [Web scraping from Wikipedia]

Data Wrangling

- Objective: Convert outcomes into training labels (1 = success, 0 = failure).
- Install Libraries: Use pandas and numpy for data manipulation.
- Load Data: Read SpaceX dataset with pd.read_csv().
- Identify Missing Values: Calculate percentage of missing values for each attribute.
- Data Analysis:
  - Determine number of launches at each site using value_counts().
  - Calculate occurrence of each orbit.
  - Analyze outcomes in the "Outcome" column.
- Categorize Outcomes:
  - Create a set of unsuccessful outcomes.
  - Assign labels (0 for failure, 1 for success) to a new "Class" column.
- Calculate Success Rate: Compute the average of the "Class" column to determine success rate.
- Export Data: Save the processed dataset as a CSV file for further analysis.



Github Link: SpaceX Data Wrangling

EDA with Data Visualization

- Objective: Explore the SpaceX dataset to uncover patterns and insights related to launch success.
- Install Libraries: Utilize pandas, numpy, seaborn, and matplotlib for data manipulation and visualization.
- Load Data: Read the SpaceX dataset using pd.read_csv().
- Data Overview: Display the first few rows of the dataset to understand its structure and contents.
- Visualize Relationships:
  - Create scatter plots to show the relationship between FlightNumber and PayloadMass against launch outcomes.
  - Use bar charts to visualize the success rate for different orbits.
  - Analyze the success rate across different launch sites using categorical plots.
- Identify Patterns:
  - Assess how the number of flights impacts the success rate.
  - Observe trends in payload mass concerning successful landings.
- Export Insights: Save visualizations and analyses for presentation and reporting.

Github Link: EDA Data Visualization

EDA with SQL

- Objective: Perform SQL queries on the SpaceX dataset to analyze various aspects of launch data.
- Connect to Database: Load the SpaceX dataset into a Db2 database for querying.
  - Task 1: Display unique launch sites from the dataset.
  - Task 2: Retrieve 5 records where launch sites begin with 'CCA’.
  - Task 3: Calculate total payload mass carried by boosters launched by NASA (CRS).
  - Task 4: Display average payload mass for booster version F9 v1.1.
  - Task 5: List the date of the first successful landing on a ground pad.
  - Task 6: List boosters with successful drone ship landings and payload mass between 4000 and 6000 kg.
  - Task 7: Count successful and failed mission outcomes.
  - Task 8: Find booster versions that carried the maximum payload mass using a subquery.
  - Task 9: List records with landing failures on a drone ship for 2015, including month names.
  - Task 10: Rank landing outcomes between specific dates in descending order.

Github Link: EDA with SQL

Build an Interactive Map with Folium

Markers:

- Added to indicate launch sites for Falcon 9 missions.
- Provide interactive labels with details about each launch site.

Circles:

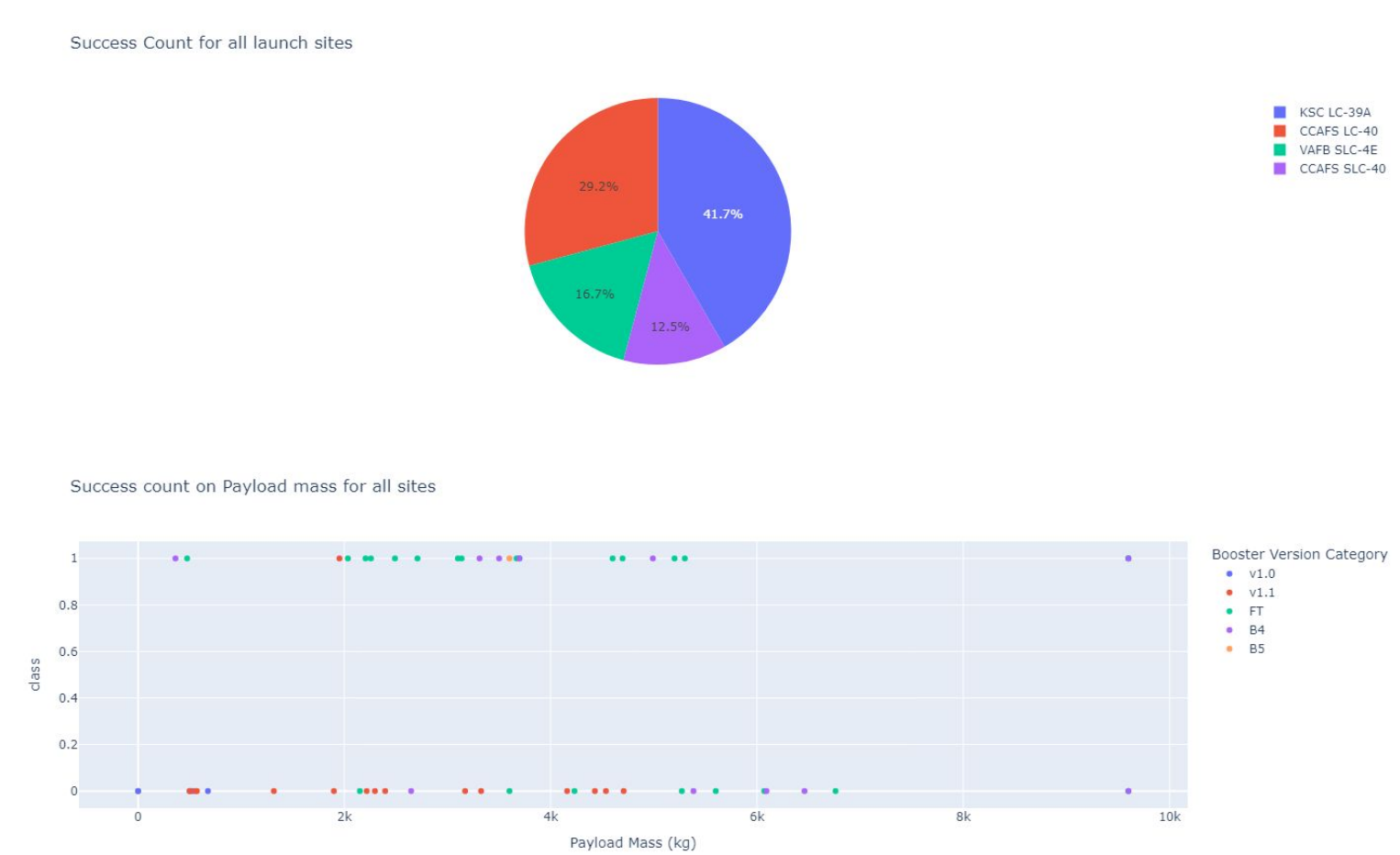
- Represented the area of influence around each launch site.
- Help visualize the geographical context of launch operations.

Popups:

- Included with markers to display additional information about each launch, such as launch dates and success rates.
- Enhance user engagement and understanding of the data.

Github Link: Map with Folium

Build a Dashboard with Plotly Dash



pie chart to show the success launches of all / each site

Scatter plot to show the success launches of all / each site by payload mass

Github Link: Dashboard

Predictive Analysis (Classification)

- LR, SVM, Decision Tree and KNN objects are created and fit with GridSearchCV object to find the best parameters,
  then the models are trained on the training set.
- The accuracy of test data are calculated for each machine learning model. It is found that the methods performed best are LR, SVM, KNN where all 3 achieved the highest accuracy of 83,33 %

TASK 12

Find the method performs best:

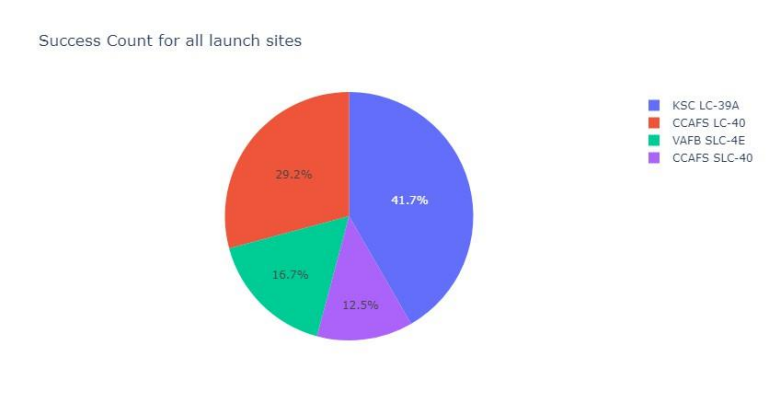
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7777777777777778
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

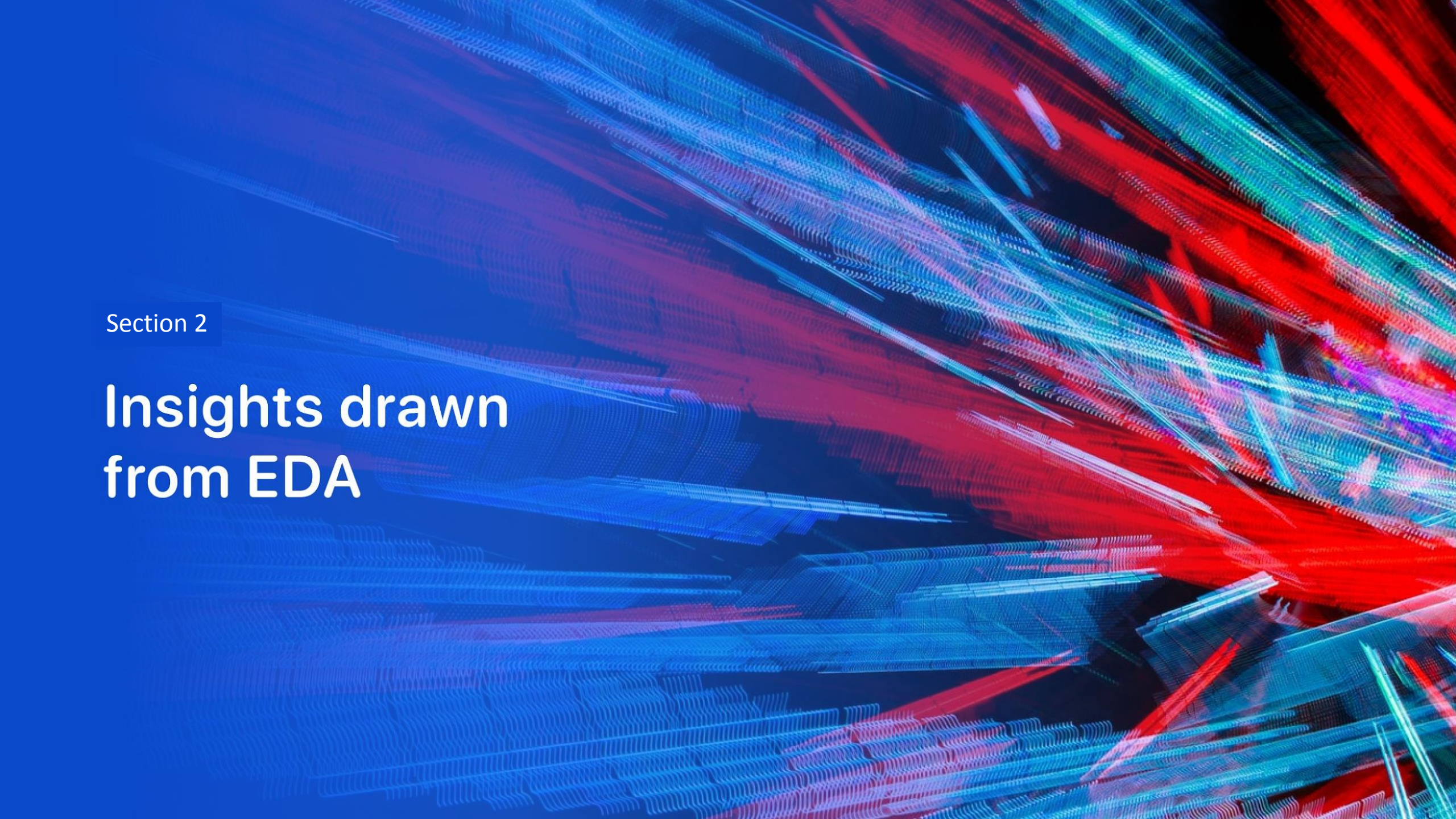
Github Link: [Machine Learning Prediction]

Results

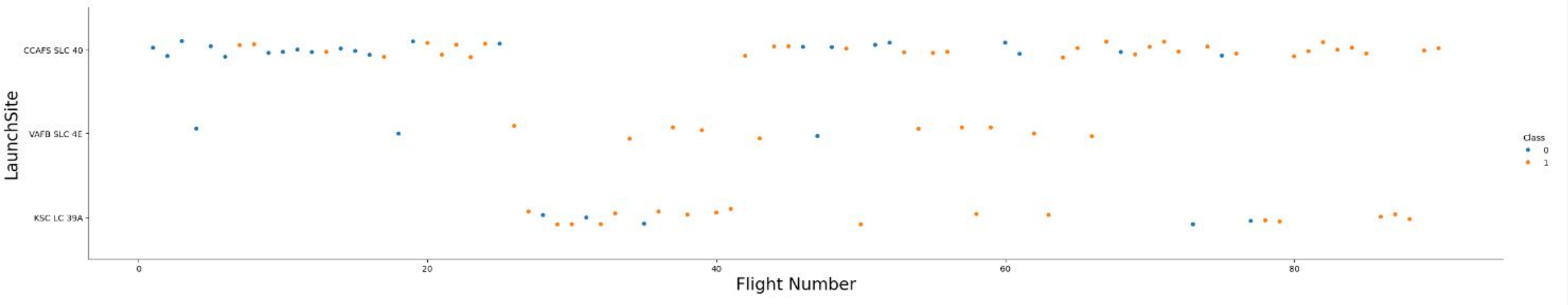
- Exploratory data analysis results:
  - Space X uses 4 different launch sites;
  - Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;
  - The number of landing outcomes became as better as years passed.
  - LR, SVM, KNN are the top-performing models for forecasting outcomes in this data
  - GEO, HEO, SSO, ES L1 orbit types exhibit the highest rates of successful launches



KSC-LC-39A has the most successful launches overall

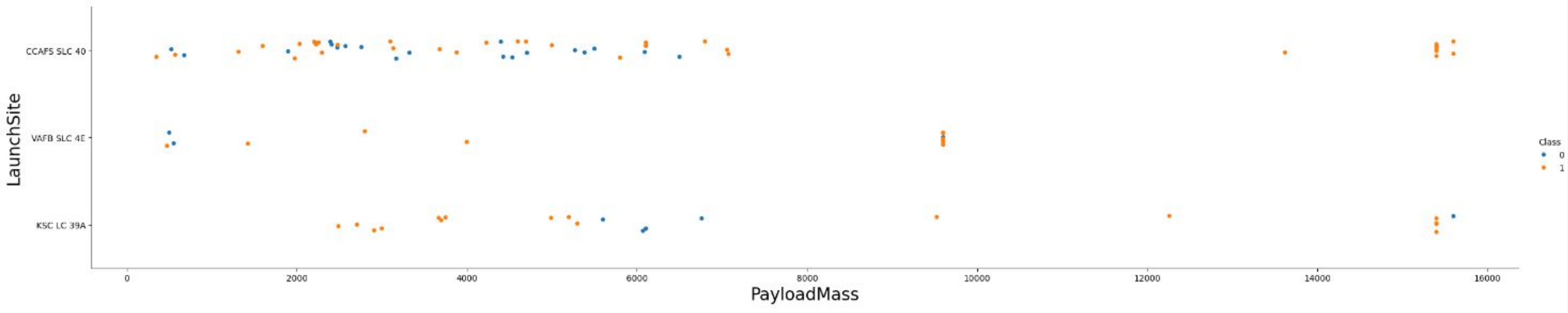

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



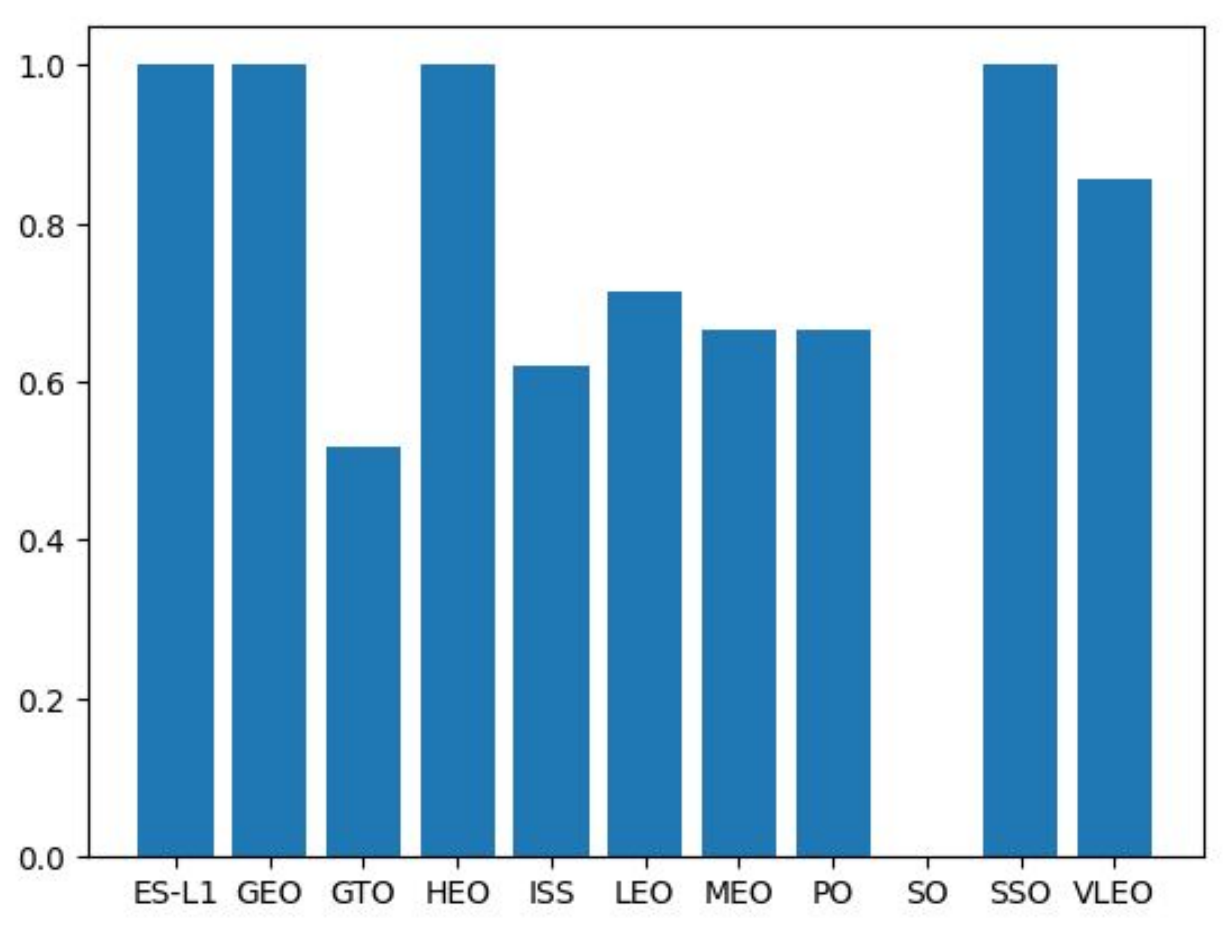
Total numbers of launches from launch site CCAFS SLC 40 are significantly higher than the other launch sites

Payload vs. Launch Site



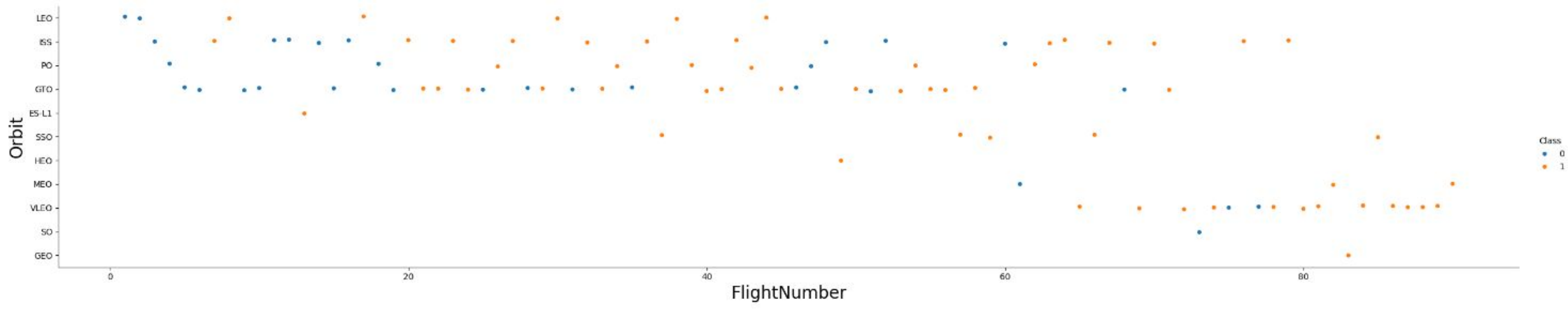
Payloads with lower mass are have more launches compared to those with higher mass across all three launch sites

Success Rate vs. Orbit Type



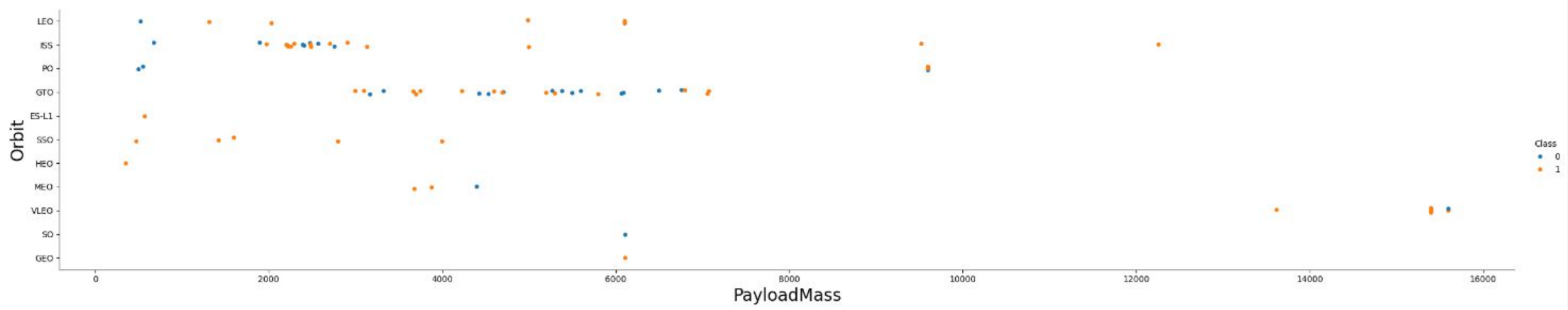
Orbit types ES-L1, GEO, HEO, SSO have
the highest success rate among all.

Flight Number vs. Orbit Type



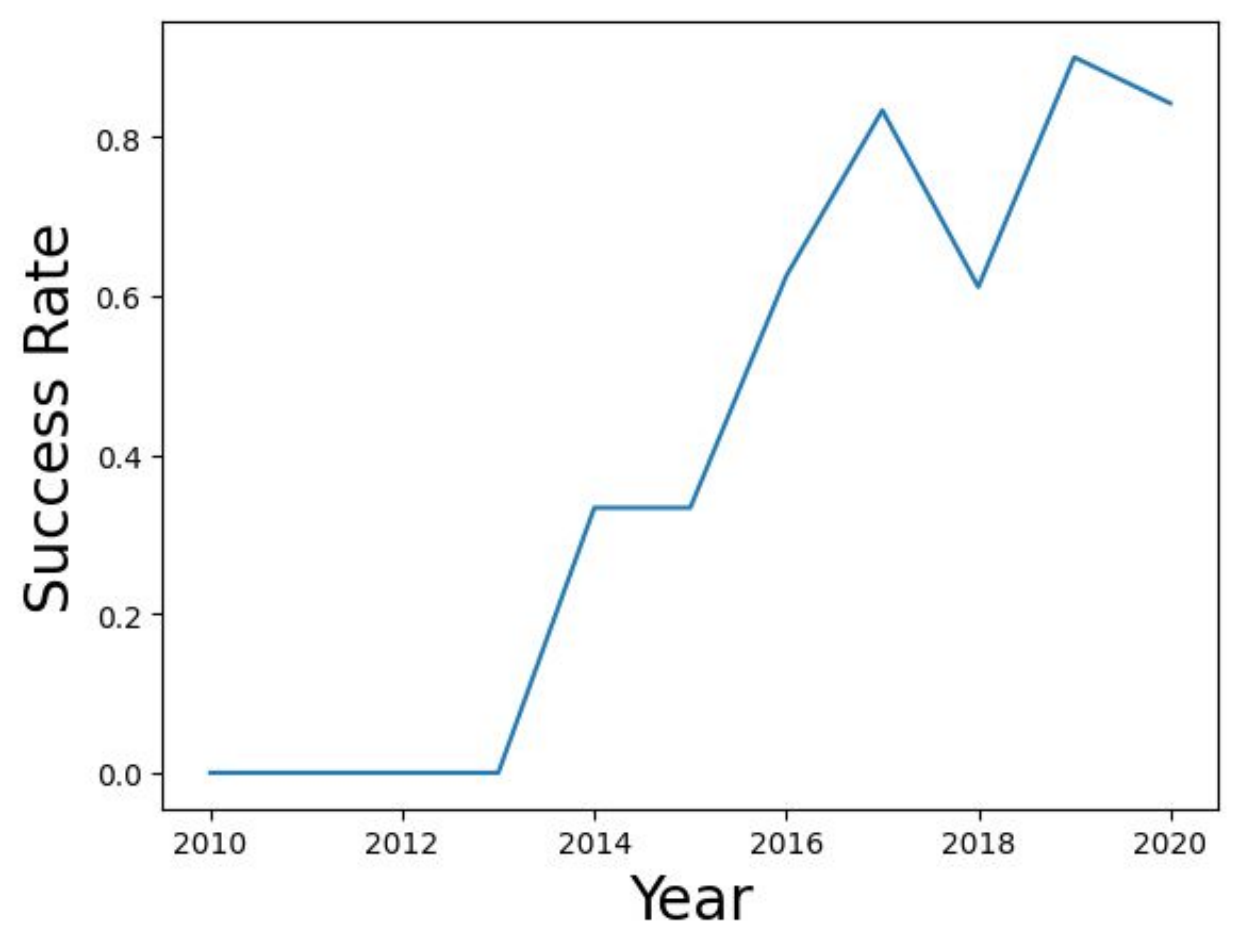
LEO, ISS, PO, GTO orbits have the most launches in the earlier years, but it slowly shifted to VLEO orbit in the later years

Payload vs. Orbit Type



Heavy payloads tend to have higher successful landing rates for PO, LEO and ISS orbit, gut GTO orbit success is less predictable with an almost equal mix of success and failures

Launch Success Yearly Trend



The success rate of launches have been increasing since 2013 till 2020, possibly due to technology advancement and experience

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

Performed an SQL query to obtain all
launch site names

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

Performed an SQL query to obtain 5 launch site names that begin with
‘CCA’

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

Performed an SQL query to obtain the total payload mass carried by boosters launched by NASA (CRS)

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 LIKE "F9 V1.1%";
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

Performed an SQL query to calculate the average payload mass carried by booster version F9 v1.1

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";
```

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

MIN(DATE)
2018-07-22

Performed an SQL query to find the date of the first successful landing outcome on 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 (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000) AND (Landing_Outcome="Success (drone ship)");
```

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

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

Performed an SQL query to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

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

Performed an SQL query to calculate the total number of successful and failure mission outcomes

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_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * 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

Performed an SQL query to list the names of the boosters which have carried the maximum payload mass

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, DATE,BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome='Failure (drone ship)' AND substr(Date,0,5)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Performed an SQL query to list failed landing outcomes in drone ship, their booster versions, and launch sit names for the year 2015

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(*) FROM SPACEXTBL WHERE DATE BETWEEN "2010-06-04" AND "2017-03-20" GROUP BY Landing_Outcome HAVING Landing_Outcome="Success (ground pad)" OR Landing_Outcome="Failure (dro
```

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

Landing_Outcome	COUNT(*)
Success (ground pad)	3
Failure (drone ship)	5

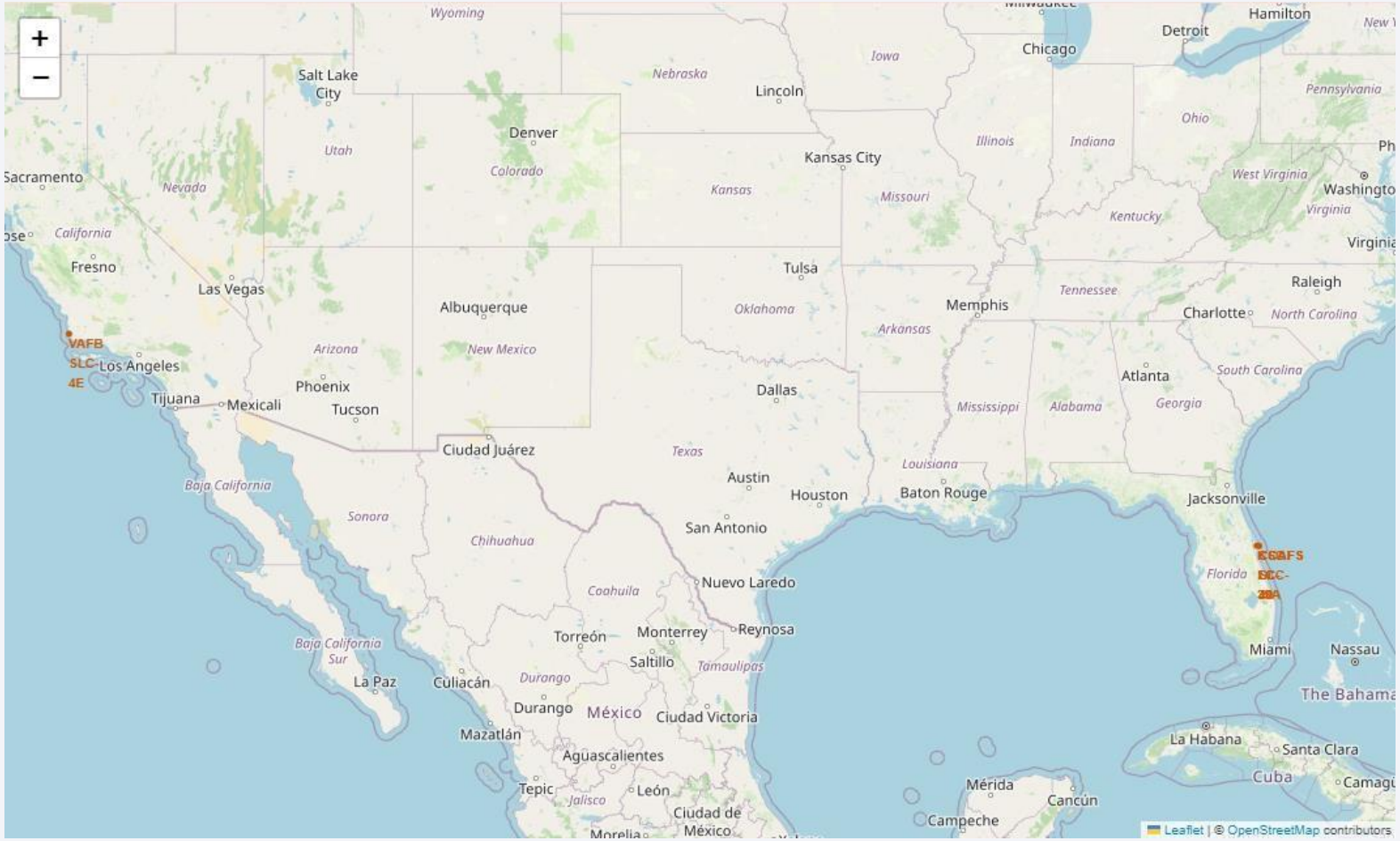
We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

Section 3

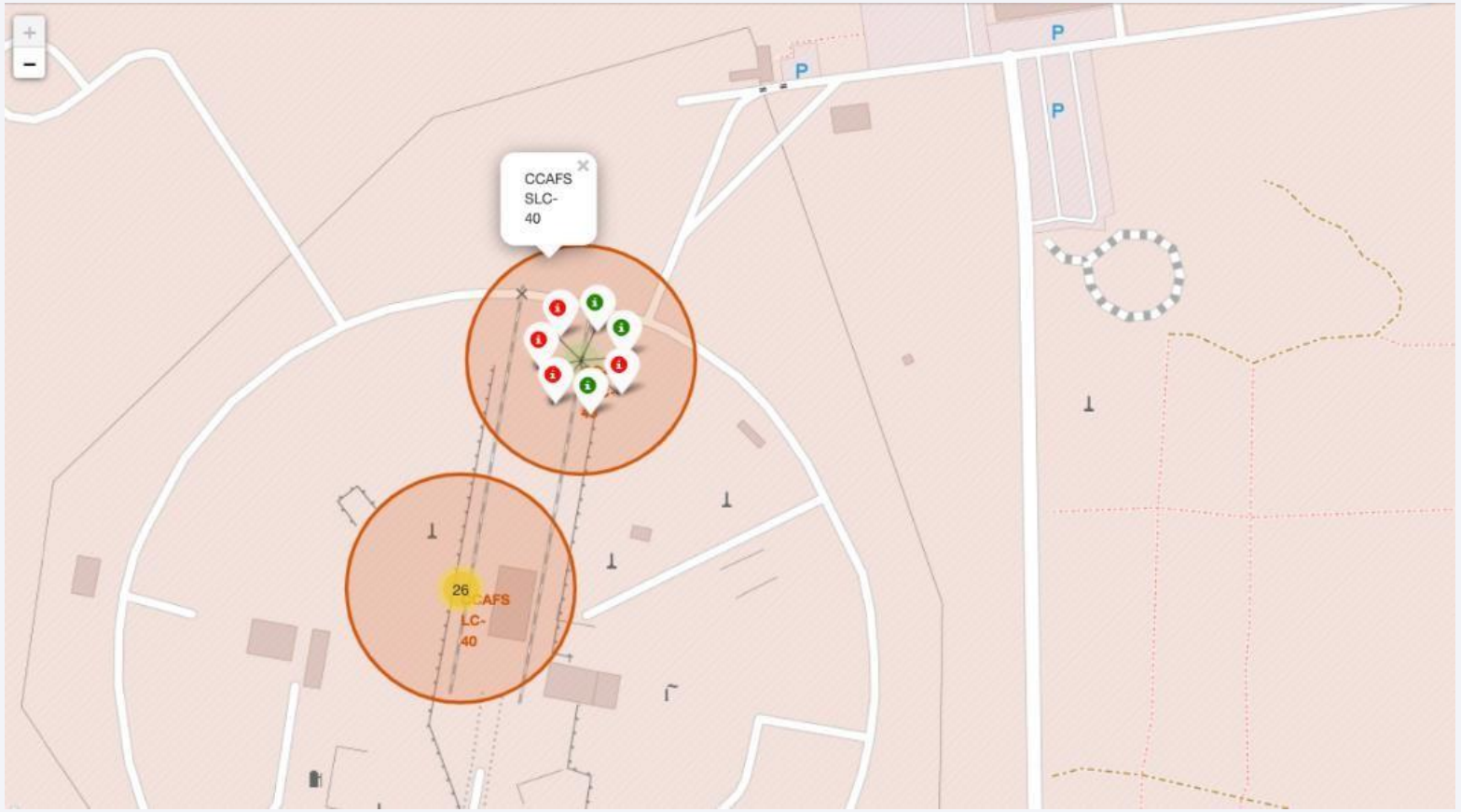
Launch Sites Proximities Analysis

All launch sites on a map



The launch sites are labelled by a marker with their names on the map

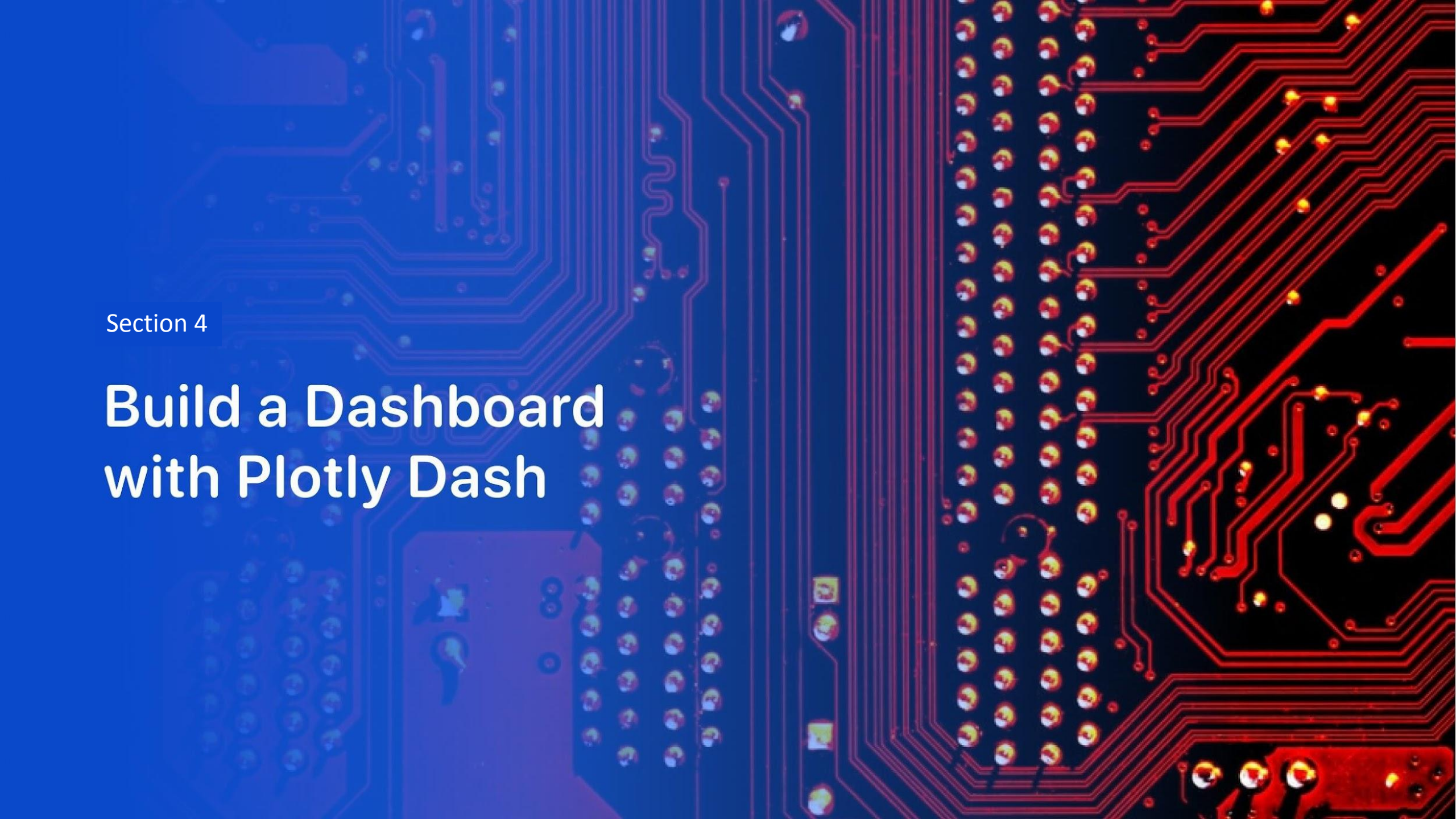
<Folium Map Screenshot 2>



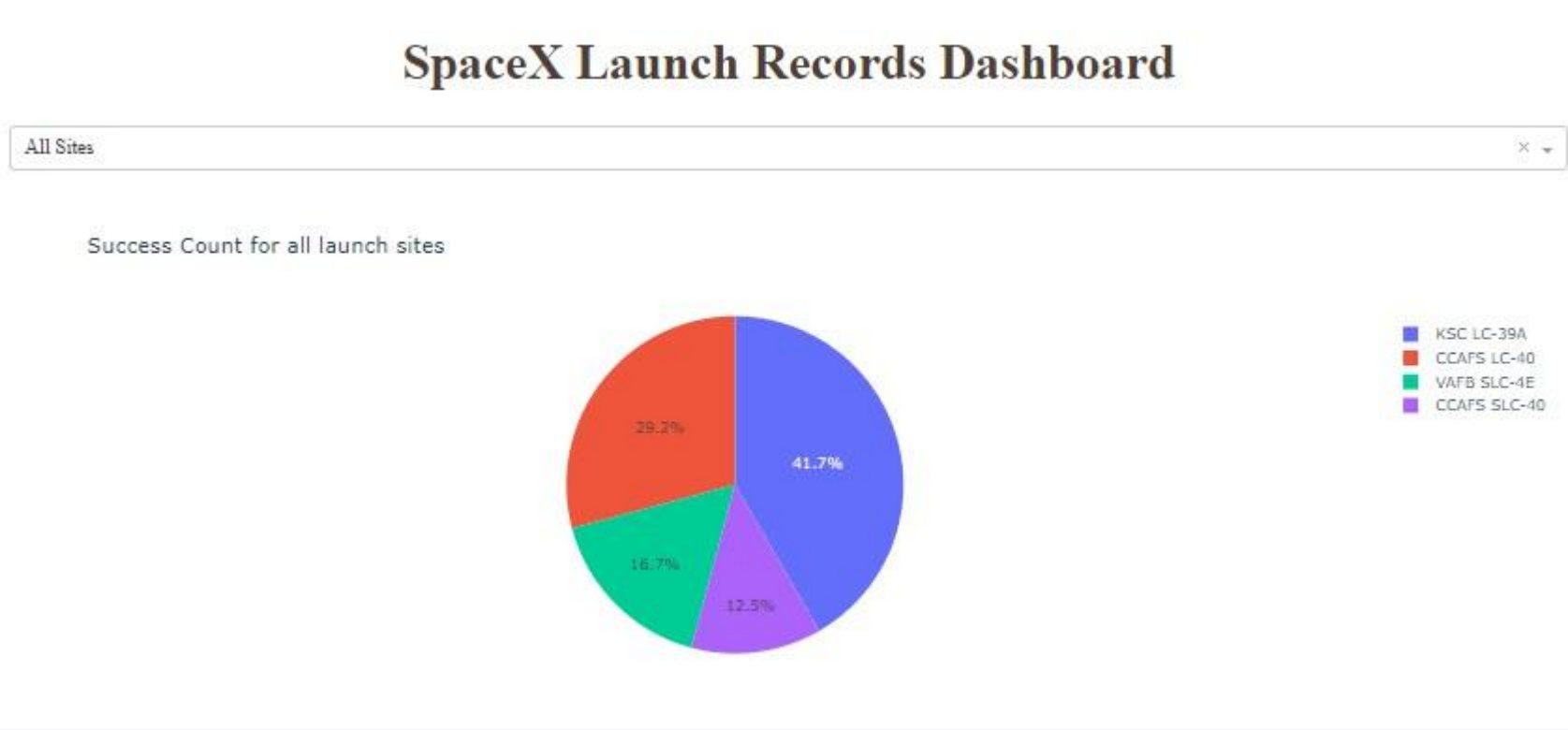
The launch records are grouped in clusters on the map, then labelled by green markers for successful launches, and red markers for failure

Distance between a launch site to its proximites

Draw a PolyLine between a launch site to the selected coastline point

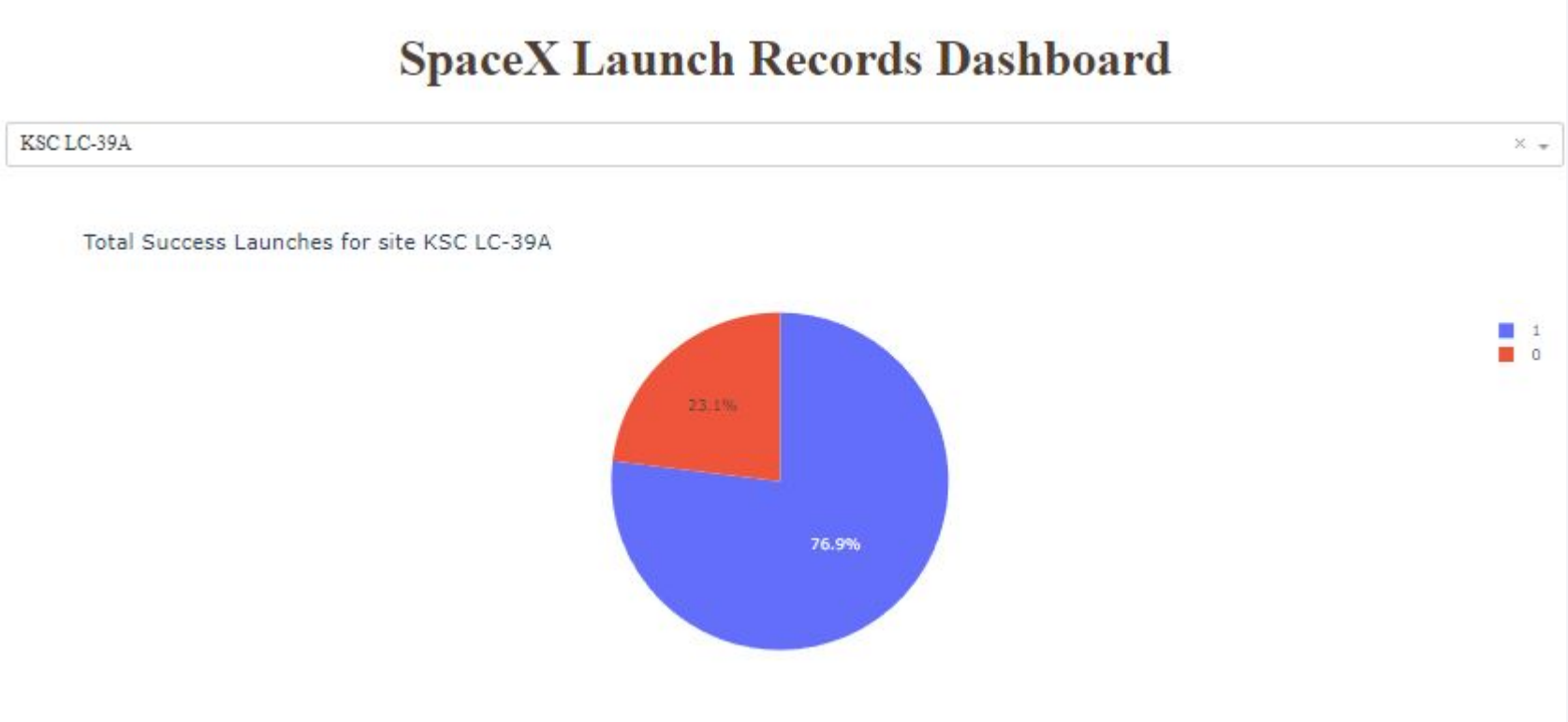

Section 4
Build a Dashboard with Plotly Dash

Total success launches for all sites



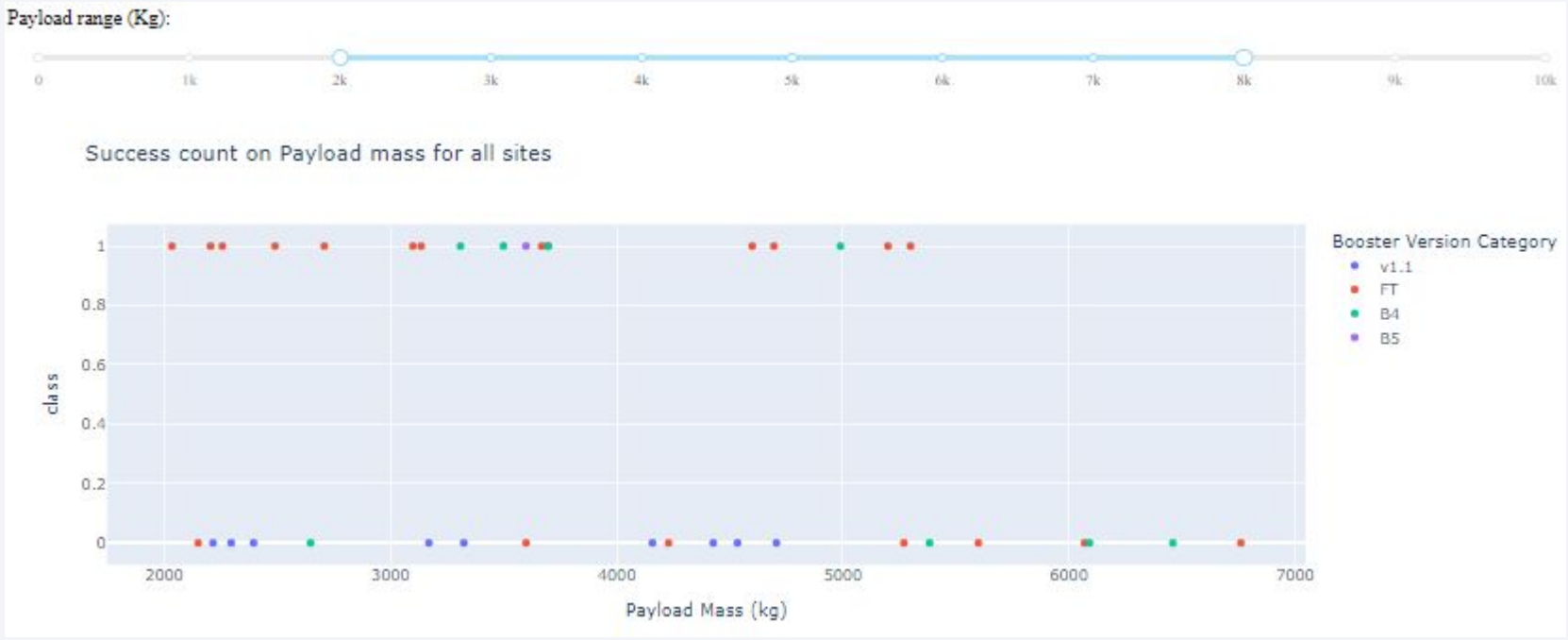
KSC LC-39A has the highest amount of success launches with 41.7% from the entire record, CCAFS SLC-40 has the lowest amount of success launches with only 12,5 %

<Dashboard Screenshot 2>



KSC LC-39A which is the launch site with highest amount of success, has a 76,9 success rate for the launches from its site, and 23,1 failure rate

Payload range with highest success launches

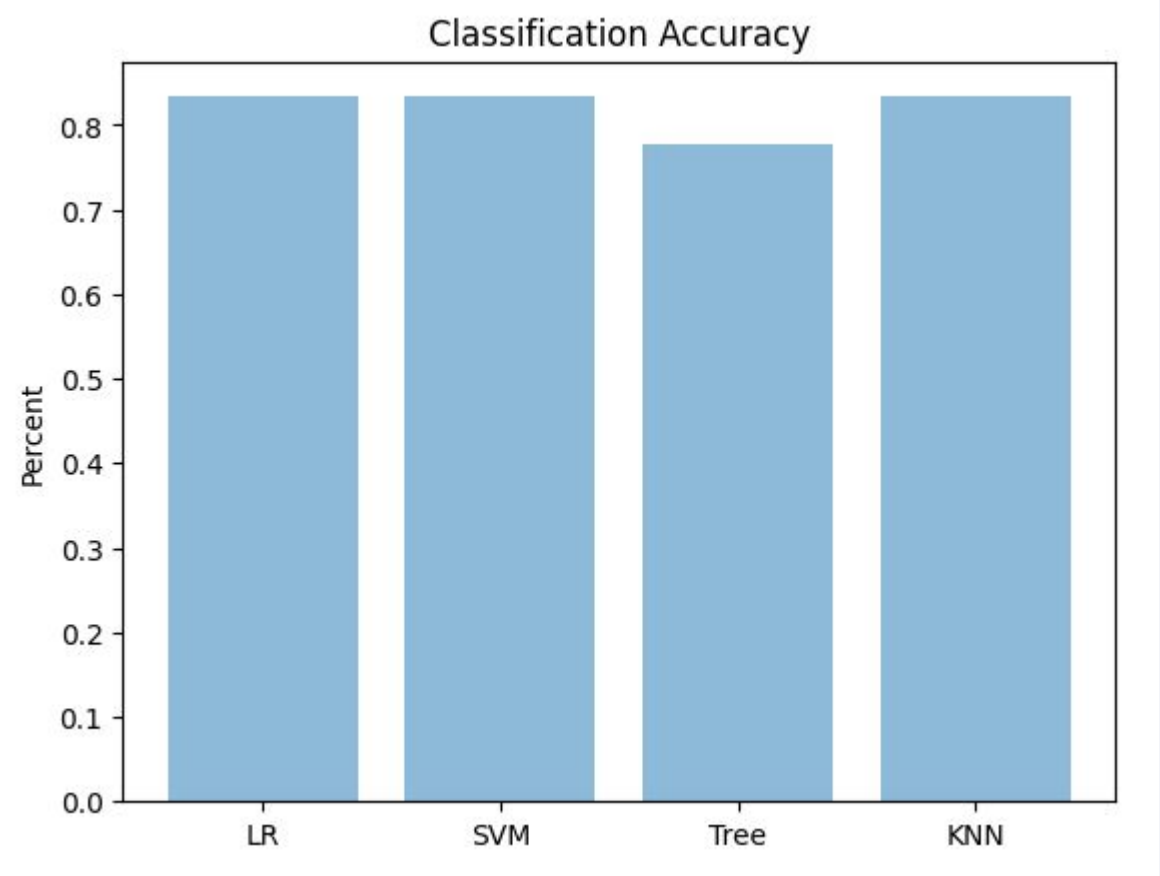


The payload range that has the highest success launches is between 2.000 and
4.000 kg, which can be seen the most number of plots in that range

Section 5

Predictive Analysis (Classification)

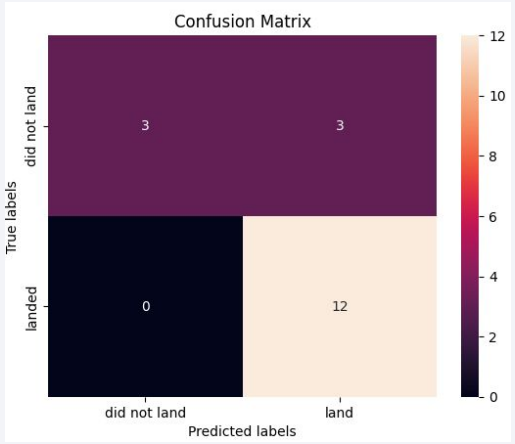
Classification Accuracy



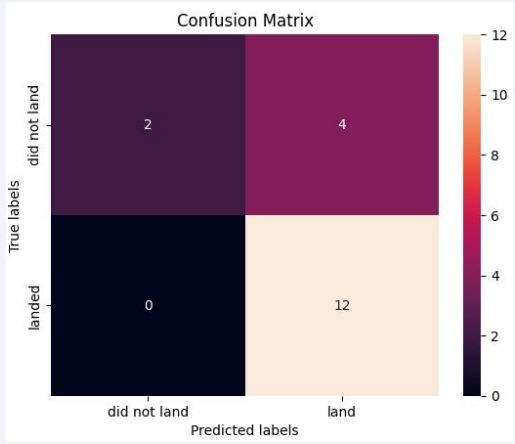
The best performed methods are: LR, SVM, KNN where all 3 achieved the highest accuracy of 83,33%

Confusion Matrix

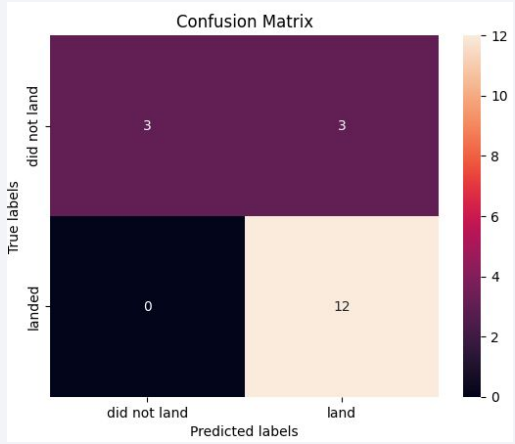
Logistic Regression Confusion Matrix



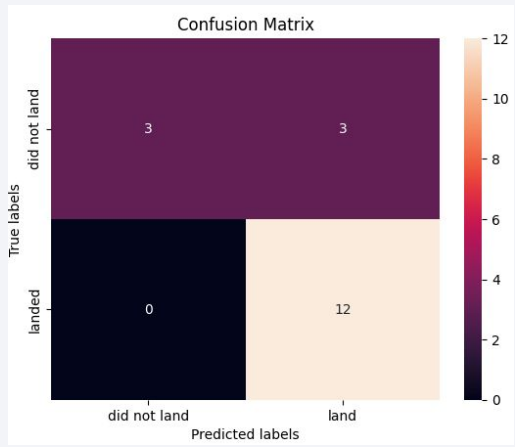
Tree Confusion Matrix



Support Vector machine Confusion Matrix



KNN Confusion Matrix



LR, SVM, KNN models have the same accuracy of 83,33% as displayed earlier, hence the same
confusion matrix

Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!