IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ilaria Battaglia
24 November 2024

Outline

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

Executive Summary

Summary of methodologies:

- Data Collection (API and Web Scraping)
- Data Wrangling
- EDA with SQL
- EDA with Visualization (Pandas and Matplotlib)
- Interactive Visual Analytics with Folium
- Interactive Dashboard (Plotly Dash)
- Predictive Analysis

Summary of all results:

- Exploratory Data Analysis Results
- Interactive Analytics, Dashboard
- Predictive Analysis Results

Introduction

Project background and context:

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward 0f 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against Space X for a rocket launch.

Problems to find answers to:

- What are the variables that most influence the outcome of a landing?
- How do these variables influence the outcome of a landing?
- Lastly we want to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

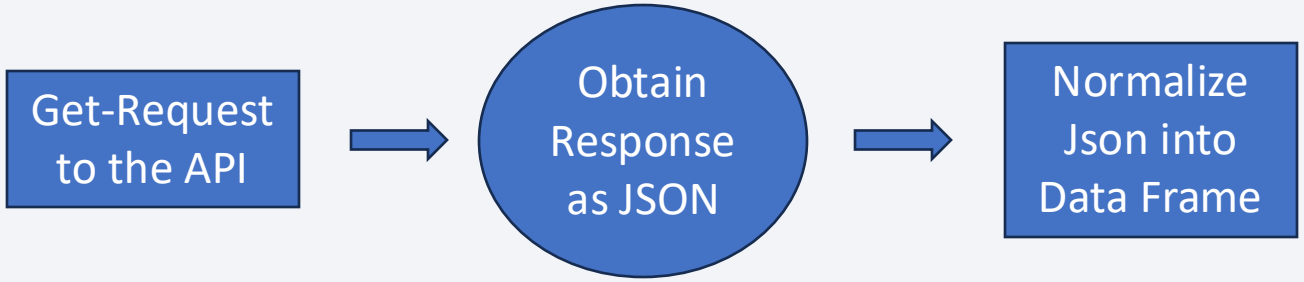
Executive Summary

- Data collection methodology:
  - Space X Rest API and Webscraping
- Perform data wrangling
  - One Hot Encoding and selection of relevant columns
- 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

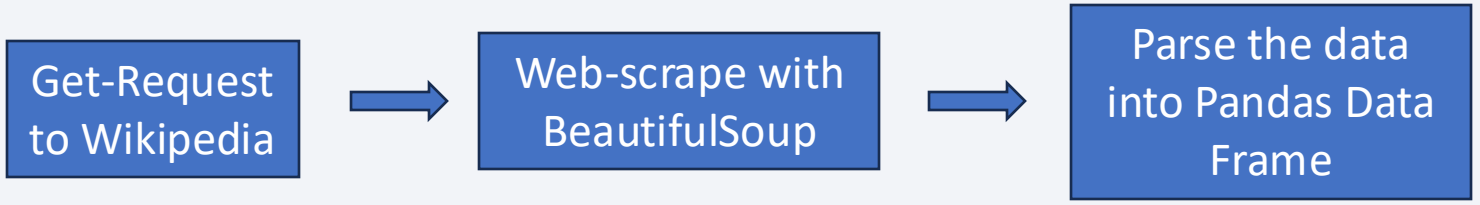
Data Collection with Space X Rest API:

A URL has been used to target a specific end point of the API to get past launch data. We used a get request to obtain the data and a response was obtained in the form of a list of JSON objects. The Json data have been normalized into a data frame.



Data Collection by Web Scraping Wikipedia with BeautifulSoup:

We used BeautifulSoup to web scrape HTML tables from the Wikipedia page. We then parsed the data and converted them into a Pandas data frame. Some columns had identifications number instead of actual data. We therefore needed to use the API again, , targeting another end point to gather specific data in order to build the dataset. Raw data was transformed into a clean dataset by wrangling, sampling and dealing with nulls.



Data Collection – SpaceX API

1. Request data from SpaceX API with the URL:

   spacex_url="https://api.spacexdata.com/v4/launches/past"

2. Use a get-request:

   response = requests.get(spacex_url)

3. Decode the response content as a Json and normalize it into a Pandas data frame:

   data = pd.json_normalize(response.json())

... Some of the columns have IDs, use the API again to get data and store it in lists.

Construct our dataset using the data we have obtained. Combine the columns into a dictionary.

Create a Pandas data frame from the dictionary launch_dict:

data = pd.DataFrame.from_dict(launch_dict)

Github: Data Collection API

Data Collection - Scraping

1. Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response:

   r = requests.get(static_url)

2. Create a BeautifulSoup object from the HTML response:

   soup = BeautifulSoup(r.text, 'html.parser')

3. Collect all relevant column names from the HTML table header, iterate through the <th> elements and extract column names.

4. Create an empty dictionary with keys from the extracted column names:

   launch_dict= dict.fromkeys(column_names)

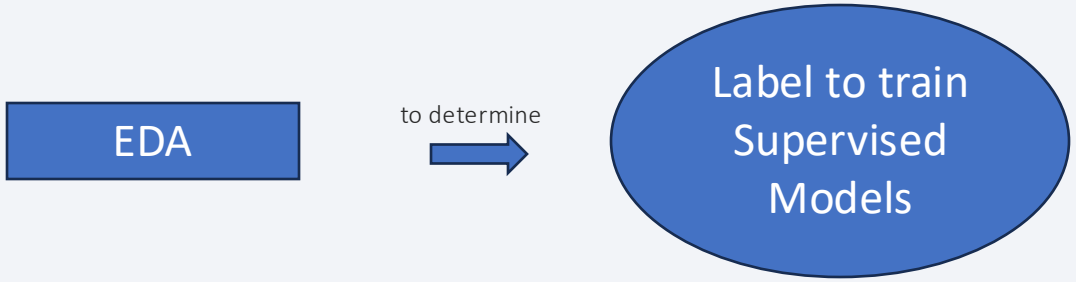
... Fill up the launch_dict with launch records extracted from table rows.

Create a dataframe:

df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })

Github: Data Collection Web Scraping

Data Wrangling



- True Ocean: successfully landed to a specific region of the ocean
- False Ocean: unsuccessfully landed to a specific region of the ocean
- True RTLS: successfully landed to a ground pad
- False RTLS: unsuccessfully landed to a ground pad
- True ASDS: successfully landed on a drone ship
- False ASDS: unsuccessfully landed on a drone ship.

Convert those outcomes into Training Labels: 1 means the booster successfully landed ,0 means it was unsuccessful.

The column 'Class' will represent the classification variable that represents the outcome of each launch.

Github: Data Wrangling

EDA with Data Visualization

Performed Data Analysis and feature Engineering using Pandas and Matplotlib

Scatter Plots: Flight Number vs Payload Mass, Flight Number vs Launch Site, Payload Mass vs Launch Site, Flight Number vs Orbit, Payload Mass vs Orbit, to see how these variables influence the outcome of the launches.

Bar Chart: Orbit vs Success Rate, to visually check if there is any relation between success rate and orbit type.

Line Chart: Year vs Average Success Rate, to get the average launch success trend.

Github: EDA Data Visualization

EDA with SQL

Performed SQL queries for EDA:

- 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 successful landing outcome in ground pad was achieved
- 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
- 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: EDA with SQL

Build an Interactive Map with Folium

Interactive Visual Analytics with Folium:

- Mark all launch sites on the map: **folium.Circle** and **folium.Marker** using site's latitude and longitude coordinates
- Mark the success/failed launches for each site on the map: folium.Marker to **marker.cluster** with green markers if the launch was successful and red markers if it was not successful
- Calculate the distances between a launch site to its proximities: points on the closest coastline, city, railway, highway ecc. using **MousePosition** to calculate the distance between the coastline point and the launch site drawing a **PolyLine**

Github: Site Launch Location with Folium

Build a Dashboard with Plotly Dash

Interactive Visual Analytics in real time:

Input Components:

- Launch Site Drop-down: We have four different launch sites and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate (class=0 vs. class=1)
- Payload Range Slider: we want to find if variable payload is correlated to mission outcome. We want to be able to select different payload range and see if we can identify some visual patterns

Graphs:

- Pie Chart: to visualize success rate of all or of a selected launch site
- Scatter Plot: to visualize how the payload might be correlated with mission outcome for selected launch site(s)

Callback Functions:

- Callback Function 1: to render success_pie_chart based on the selected site(s) from the drop-down
- Callback Function 2: to render success-payload-scatter-chart based on the selected payload range from the slider

Github: Dashboard

Predictive Analysis (Classification)

Building the Model:

- Standardize the data
- Split into training and test data
- Create Logistic Regression/Support Vector Machine/Decision Tree Classifier/K Nearest Neighbors objects

Evaluation:

- Calculate the accuracy on the test data using the method score
- Create and examine the Confusion matrix

Improvement:

- Create a GridSearchCV object to find best parameters

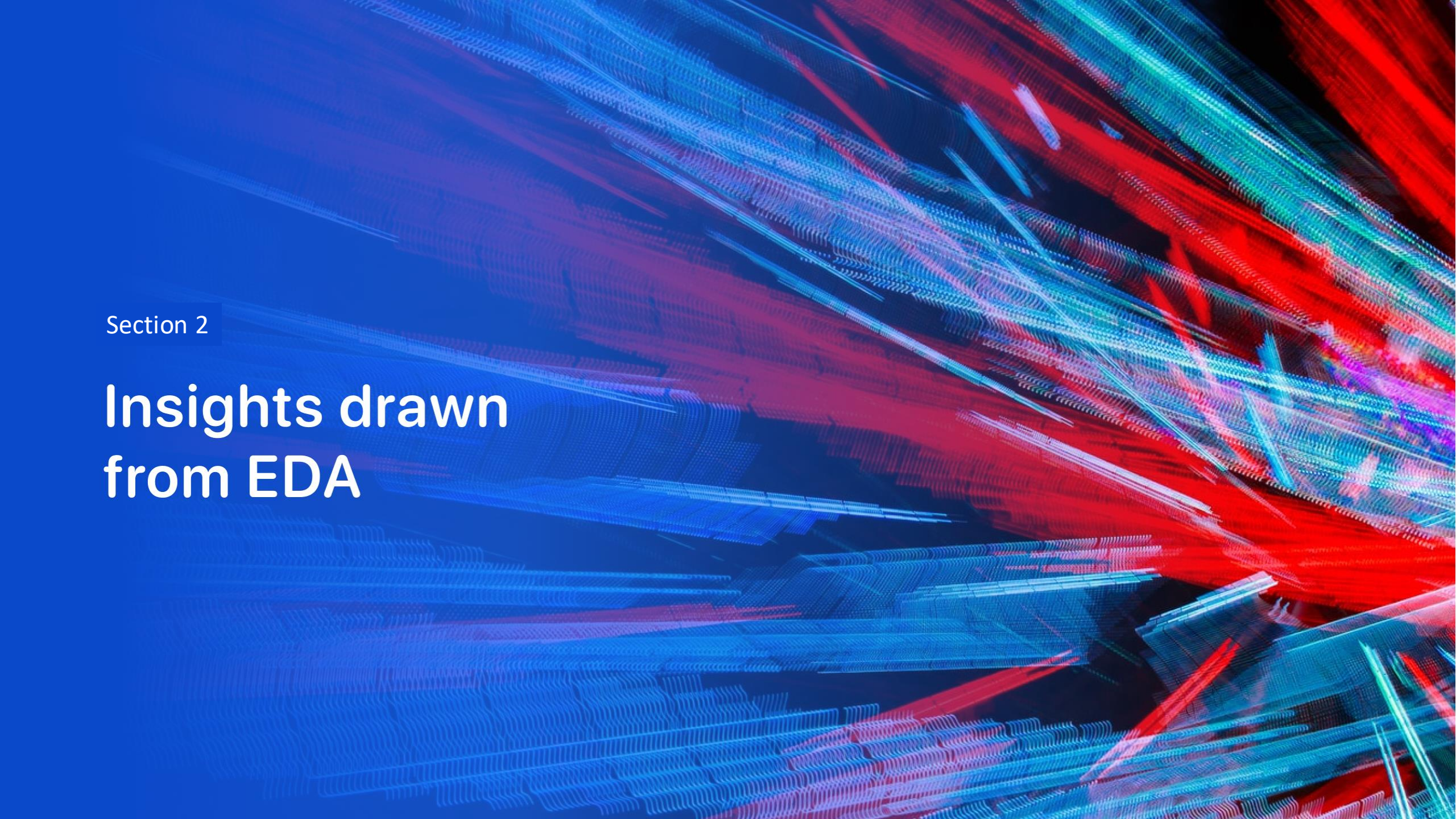
The best performing Classification Model:

- Find the model with the best accuracy score

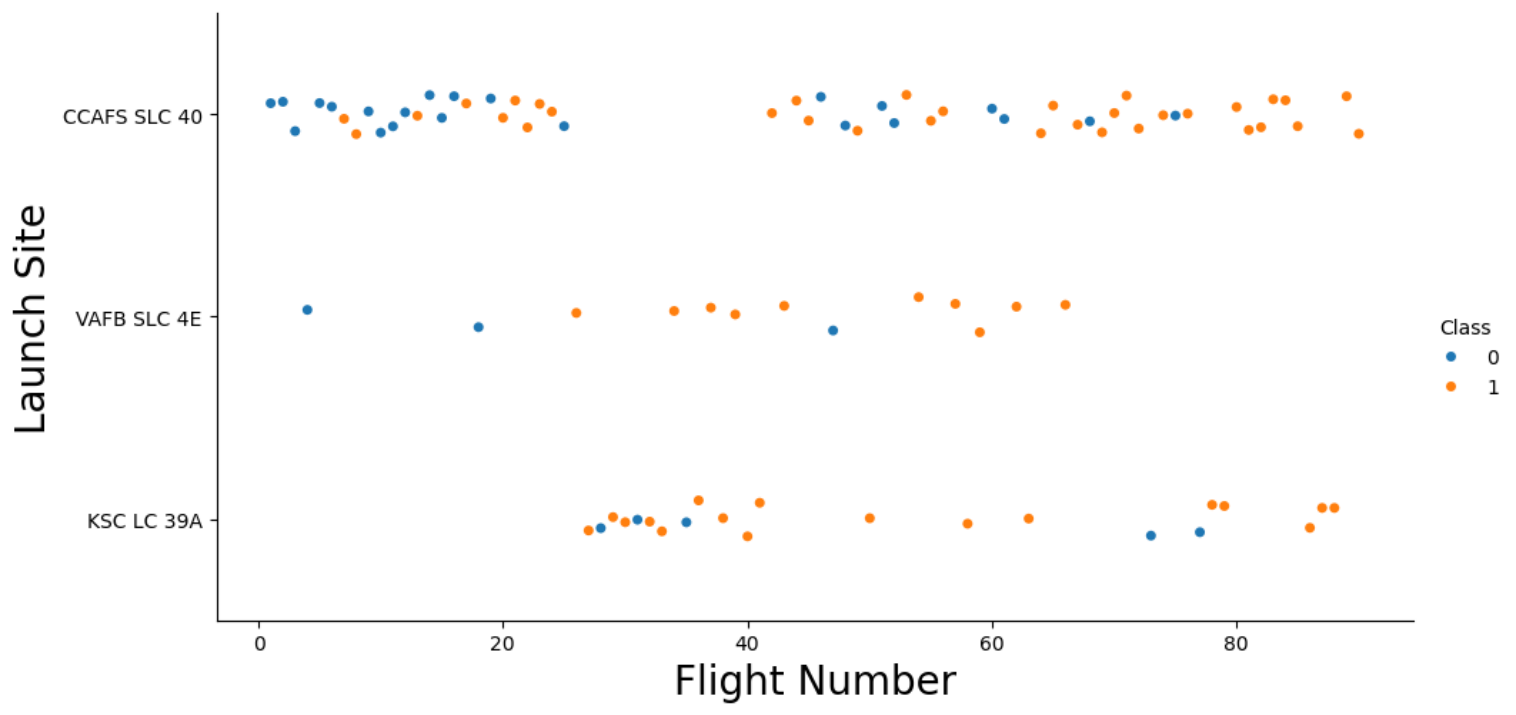
Github: Predictive Analysis

Results

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

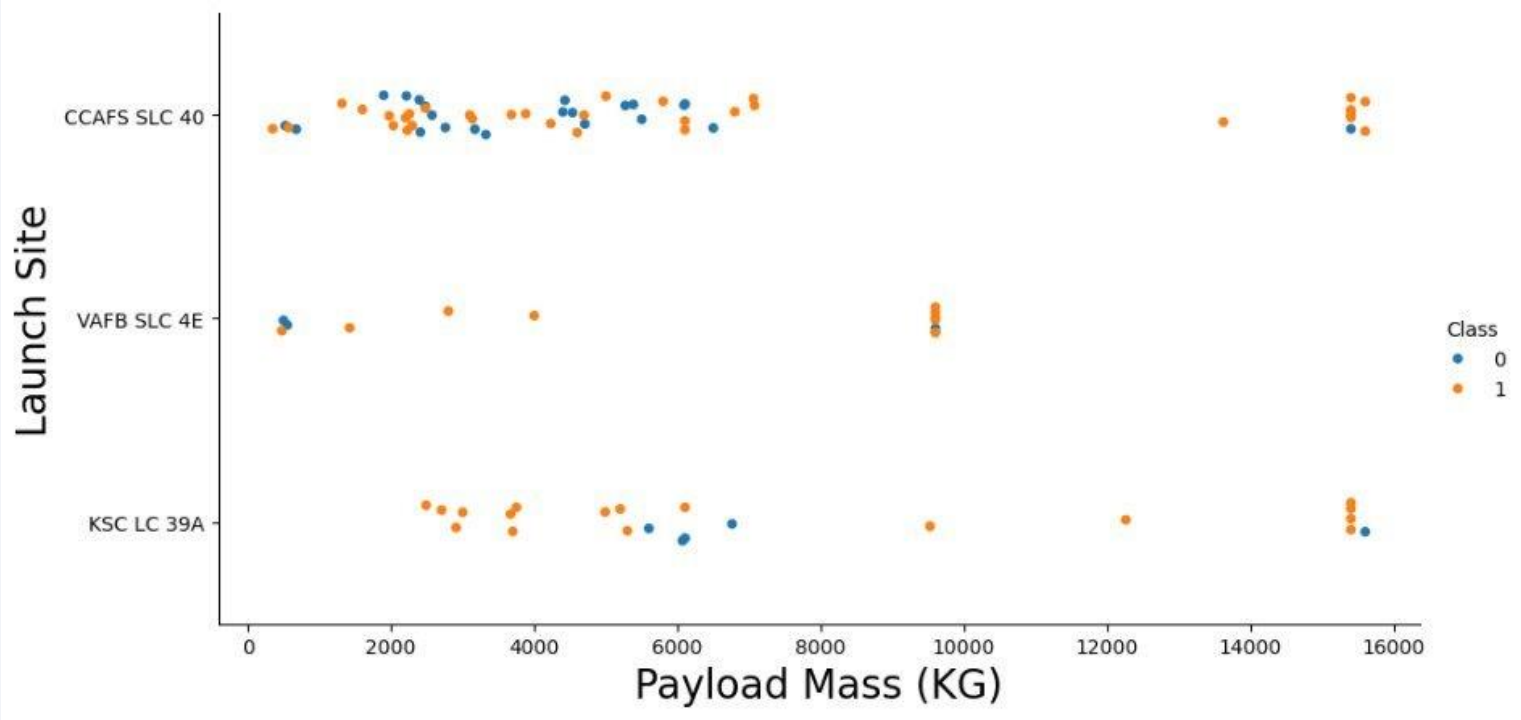

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



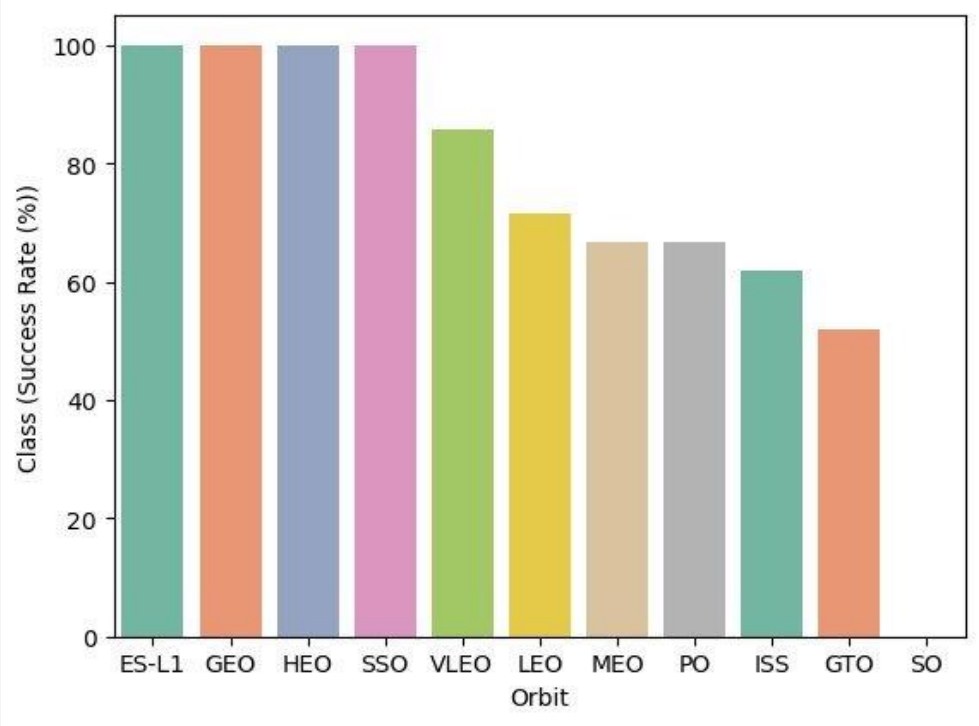
Flight Number vs Launch Site: As the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

Payload vs. Launch Site



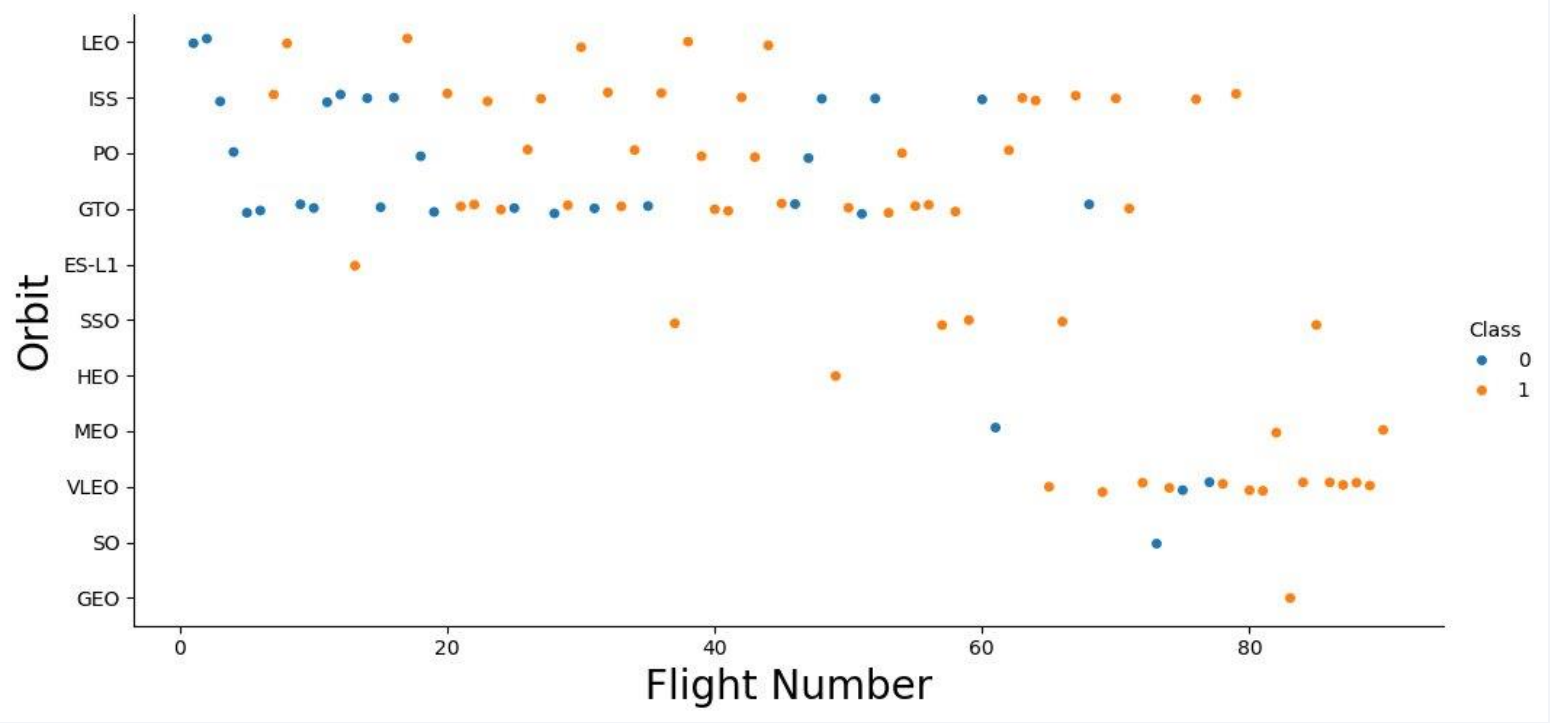
Payload Mass vs Launch Site: for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



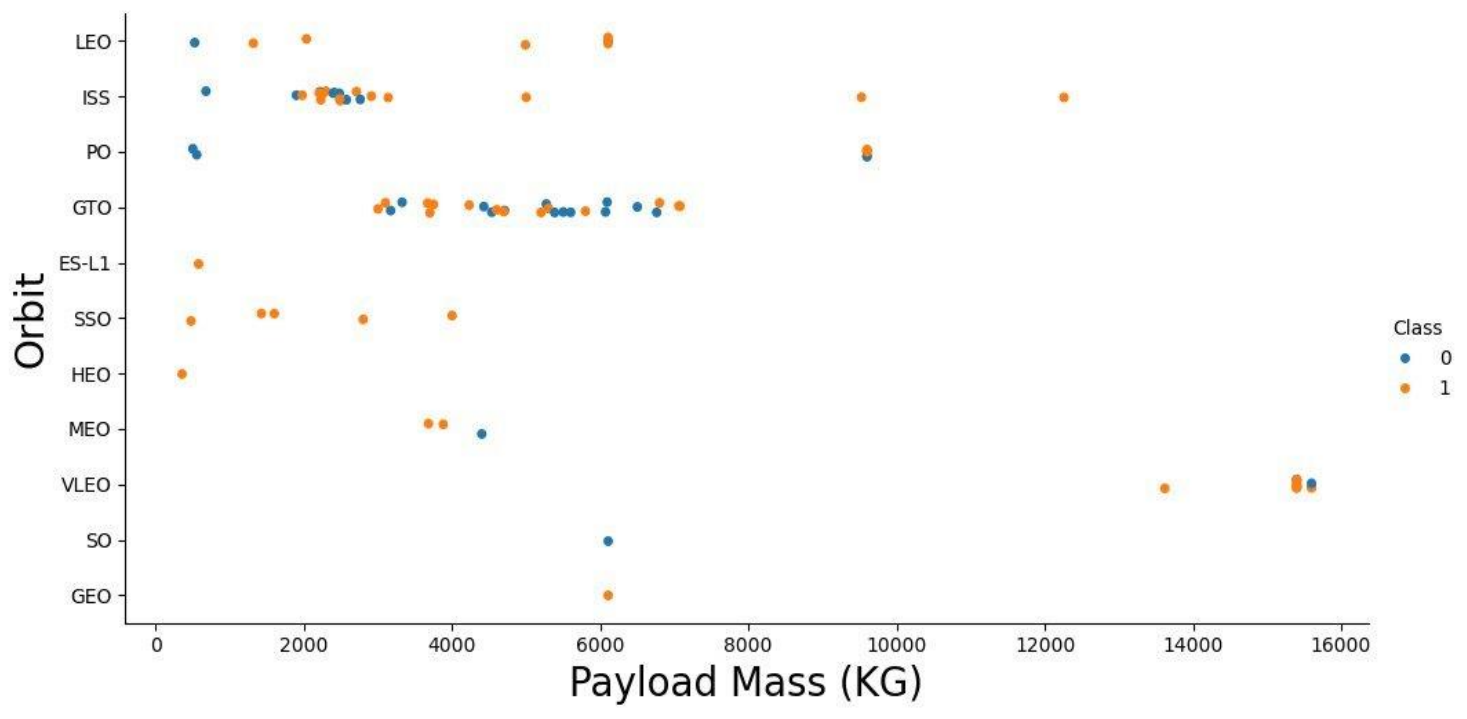
Orbit vs Success Rate: Orbits ES-L1, GEO, HEO and SSO have the highest success rates at 100%, GTO orbit has the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type



Flight Number vs Orbit: in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

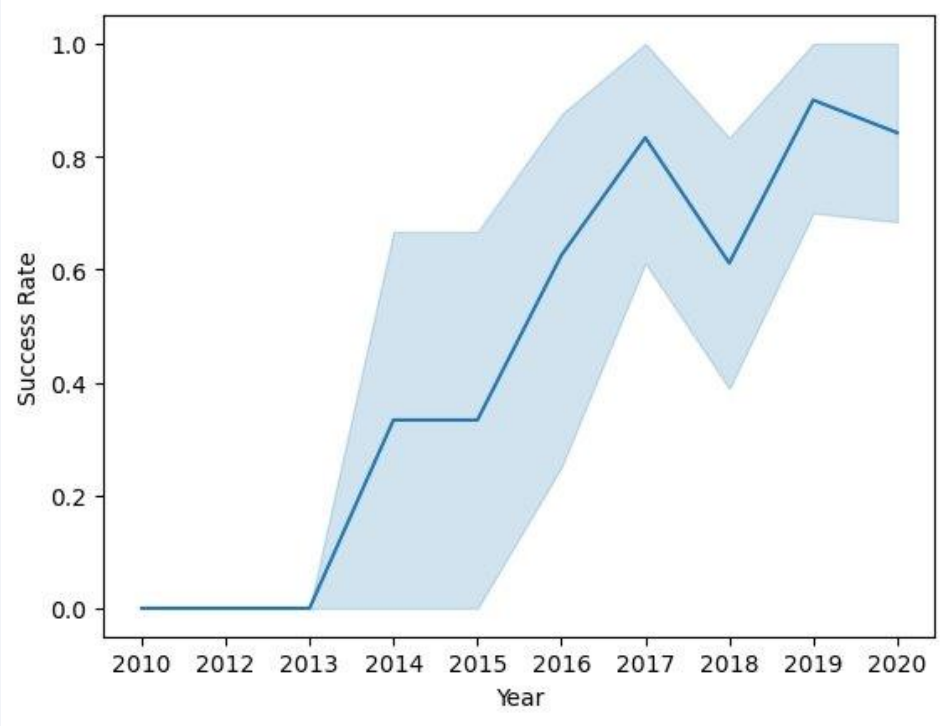
Payload vs. Orbit Type



Payload Mass vs Orbit: With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



Year vs Success Rate: success rate since 2013 kept increasing till 2020, with a drop in 2018.

All Launch Site Names

Unique Launch Sites:

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

Query:

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

Only the distinct values of the Launch Site columns are displayed.

Launch Site Names Begin with 'CCA'

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

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

Query:

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

Only 5 values are shown, whose Launch Site column begins with 'CCA'.

Total Payload Mass

Total payload mass carried by boosters launched by NASA (CRS):

NASA (CRS): 45596 KG

Query:

```
%sql SELECT sum(PAYLOAD_MASS__KG_) AS "Total_Payload_Mass", Customer FROM SPACEXTBL WHERE Customer='NASA (CRS)';
```

The dataset is filtered in order to sum the payload mass data related to the Customer Nasa.

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1:

F9 v1.1: 2928.4 KG

Query:

```
%sql SELECT avg(PAYLOAD_MASS__KG_) AS "Avg_Payload_Mass", "Booster_Version" FROM SPACEXTBL WHERE "Booster_Version"='F9 v1.1';
```

The dataset is filtered in order to sum the payload mass data related to the Booster Version F9.

First Successful Ground Landing Date

The date when the first successful landing outcome in ground pad was achieved:

First Success (ground pad): 2015-12-22

Query:

```
%sql SELECT MIN(DATE), "Landing_Outcome" FROM SPACEXTBL WHERE "Landing_Outcome"='Success (ground pad)';
```

The dataset is filtered in order to find the first successful landing outcome on ground pad .

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Query:

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

The dataset is filtered to select the Boosters with payload mass between 4000 and 6000 Kg and that landed successfully on a drone ship.

Total Number of Successful and Failure Mission Outcomes

Total number of successful and failure mission outcomes:

Successful Mission Outcomes: 100

Failure Mission Outcomes: 1

Query:

```
%sql SELECT(SELECT COUNT("Mission_Outcome") FROM SPACEXTBL WHERE "Mission_Outcome" LIKE 'Success%') AS "Successful_Mission_Outcomes", (SELECT COUNT("Mission_Outcome") FROM SPACEXTBL WHERE "Mission_Outcome" LIKE 'Failure%') AS "Failure_Mission_Outcomes";
```

Successful and Failure mission outcomes are counted from the column Mission Outcome.

Boosters Carried Maximum Payload

The names of the booster_versions which have carried the maximum payload mass:

Booster_Version	Payload
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21

Query:

```
%sql SELECT "Booster_Version", "Payload" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

The subquery finds the maximum payload mass, the query filters the names of the Booster Versions and Payload corresponding to the max mass.

2015 Launch Records

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

SUBSTR(Date,6,2)	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Query:

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

The dataset is filtered to find the month, booster version and launch site for the failure landing outcomes in drone ship for the year 2015.

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

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:

Landing_Outcome	Landing_Outcomes_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

Query:

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

Each 'typology' of landing outcome is listed and ranked in descending order based on the landing outcomes count. Only landing that have taken place between two specific date have been taken into account.

Section 3

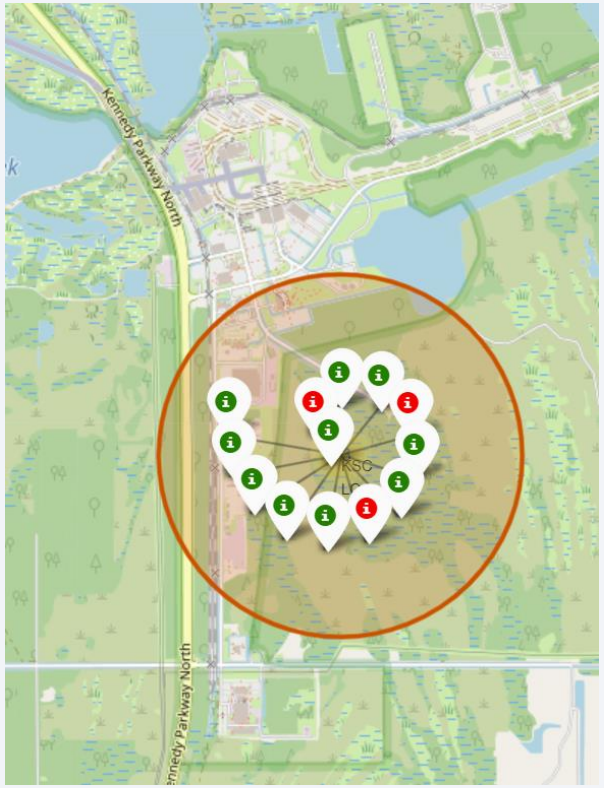
Launch Sites Proximities Analysis

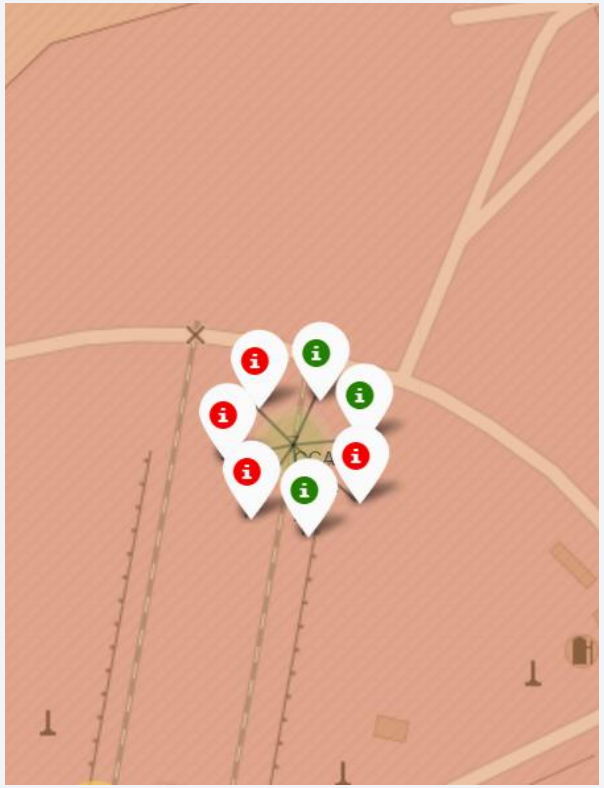
All launch sites on global map

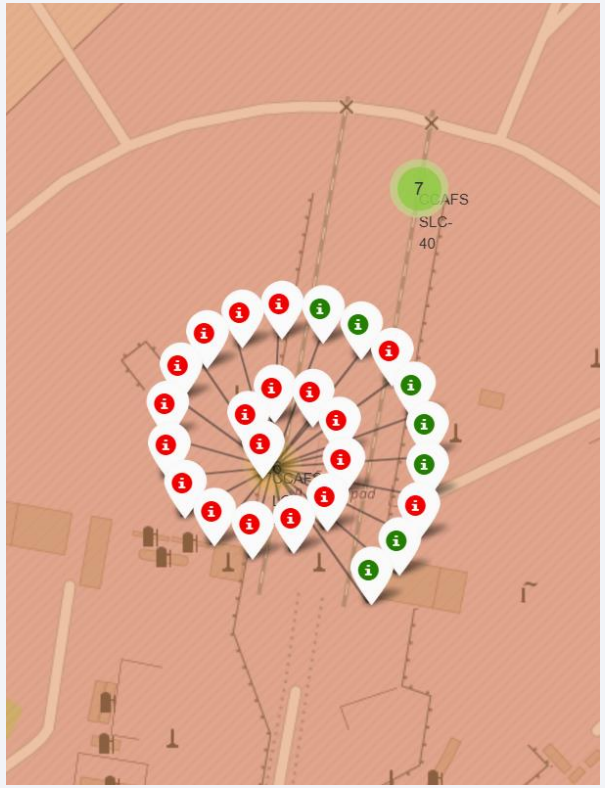
- All launch sites are in proximity to the Equator
- All launch sites are in very close proximity to the coast

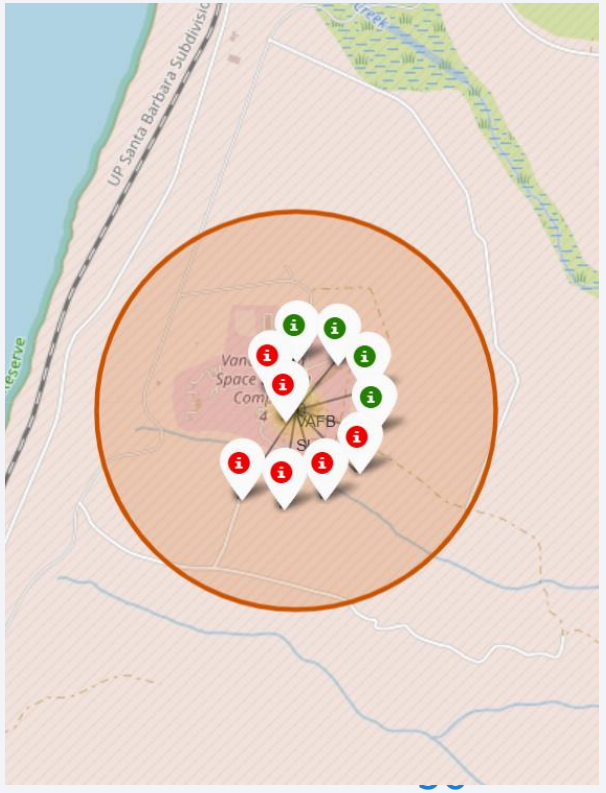
Success/Failed Launches Map

If a launch was successful (class=1), then we used a green marker and if a launch was failed, we used a red marker (class=0), in this way it is easy to understand the success rate of a specific site.



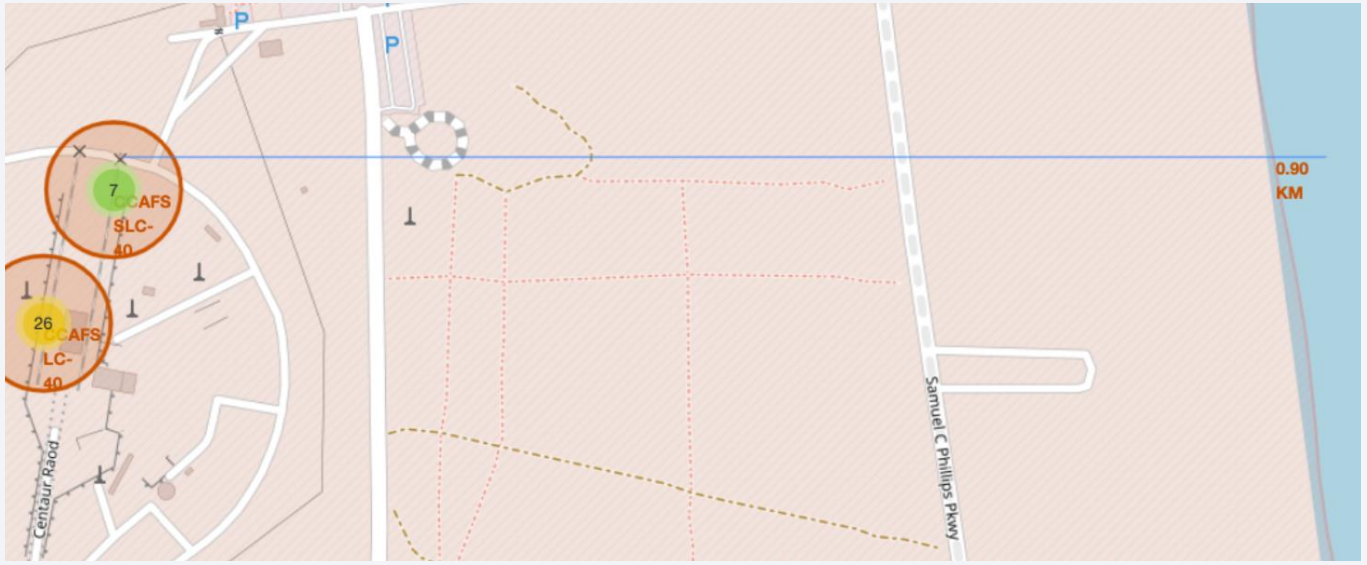


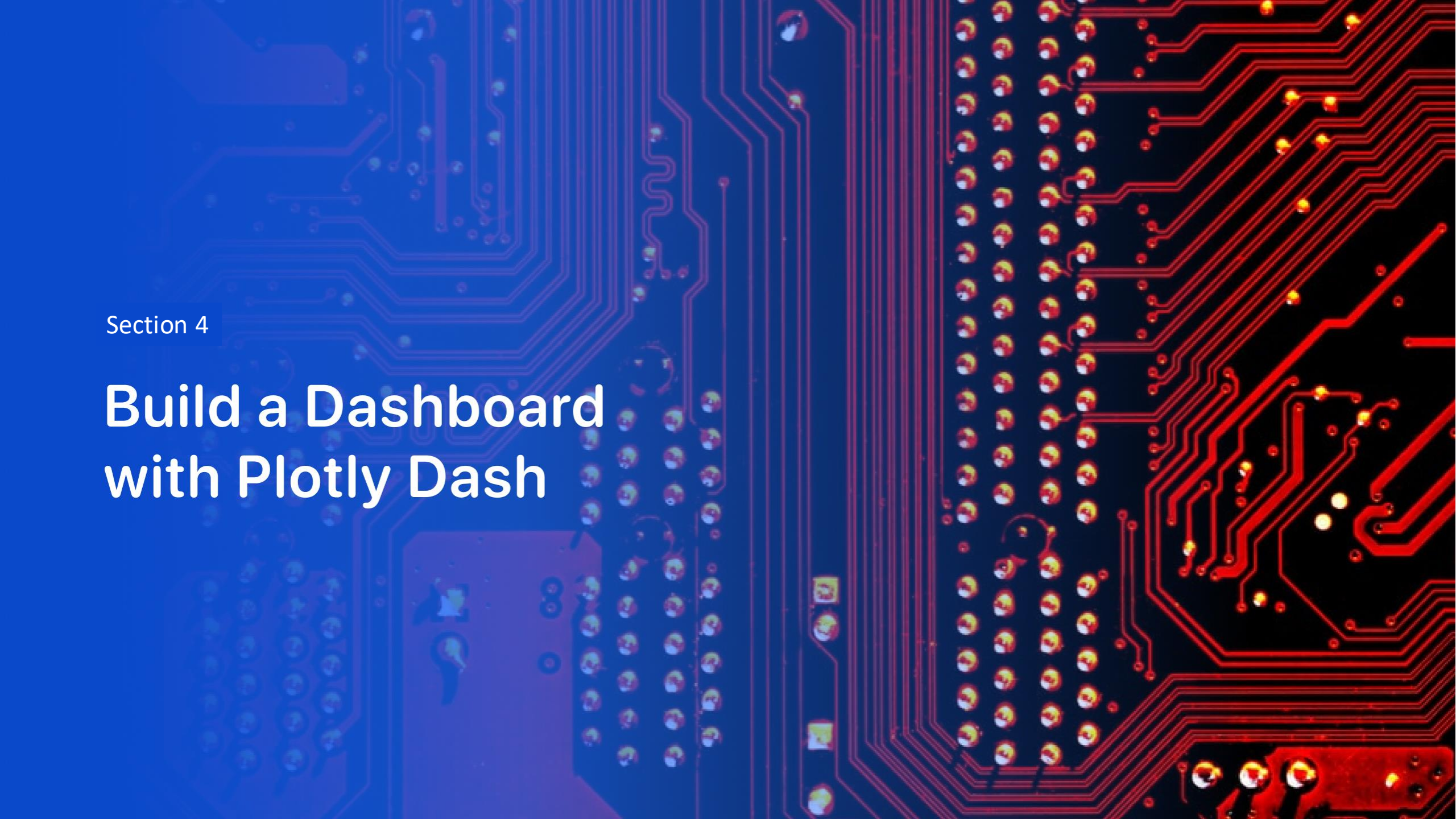




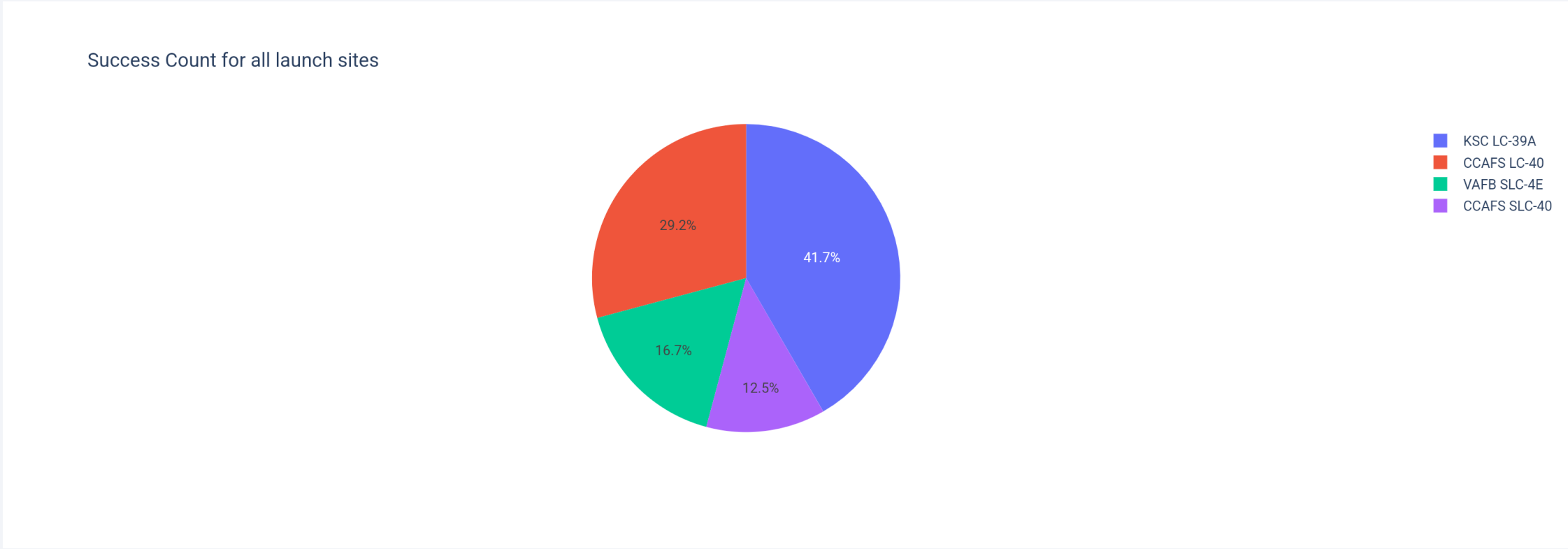
Distance between a launch site and its proximities

- Are launch sites in close proximity to railways? No.
- Are launch sites in close proximity to highways? No.
- Are launch sites in close proximity to coastline? Yes.
- Do launch sites keep certain distance away from cities? Yes.



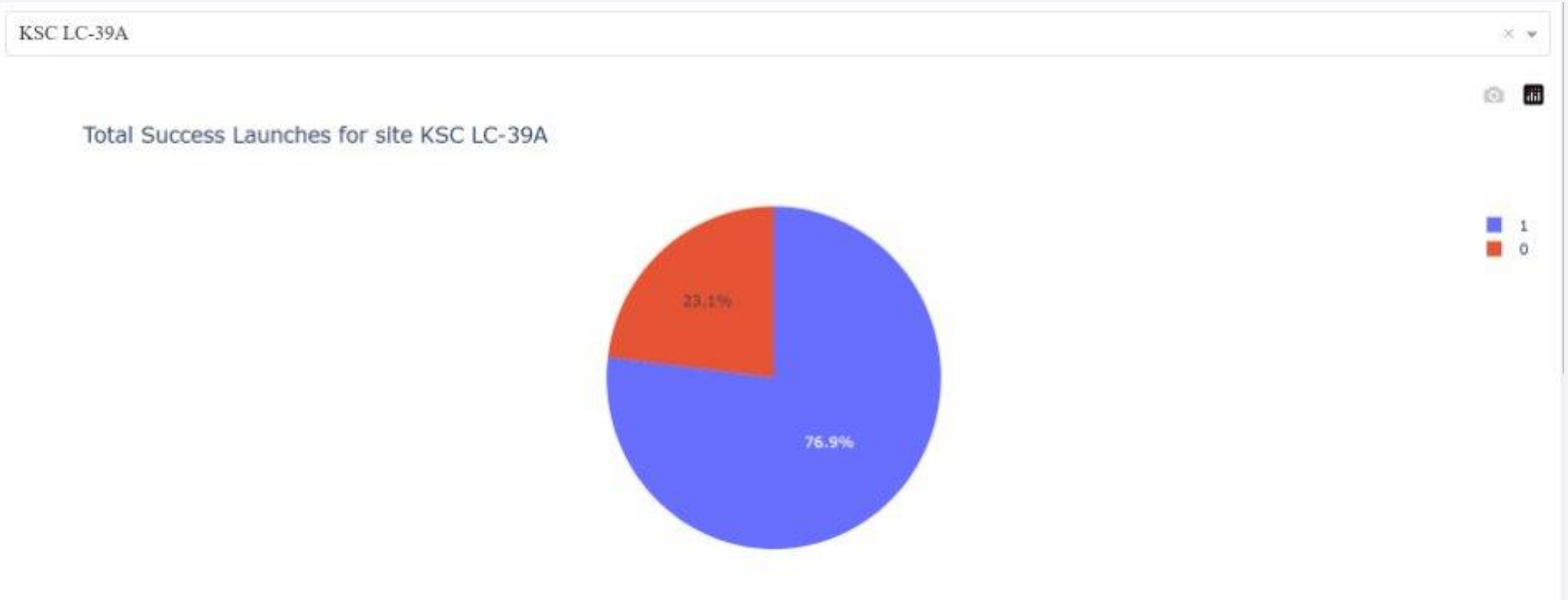

Section 4
Build a Dashboard with Plotly Dash

Dashboard Graph: launch success rate for all sites



The site KSC LC-39A has the best success rate, followed by CCAFS LC-40, VAFB SLC-4E and CCAFS SLC-40.

Launch Site with the highest launch success ratio

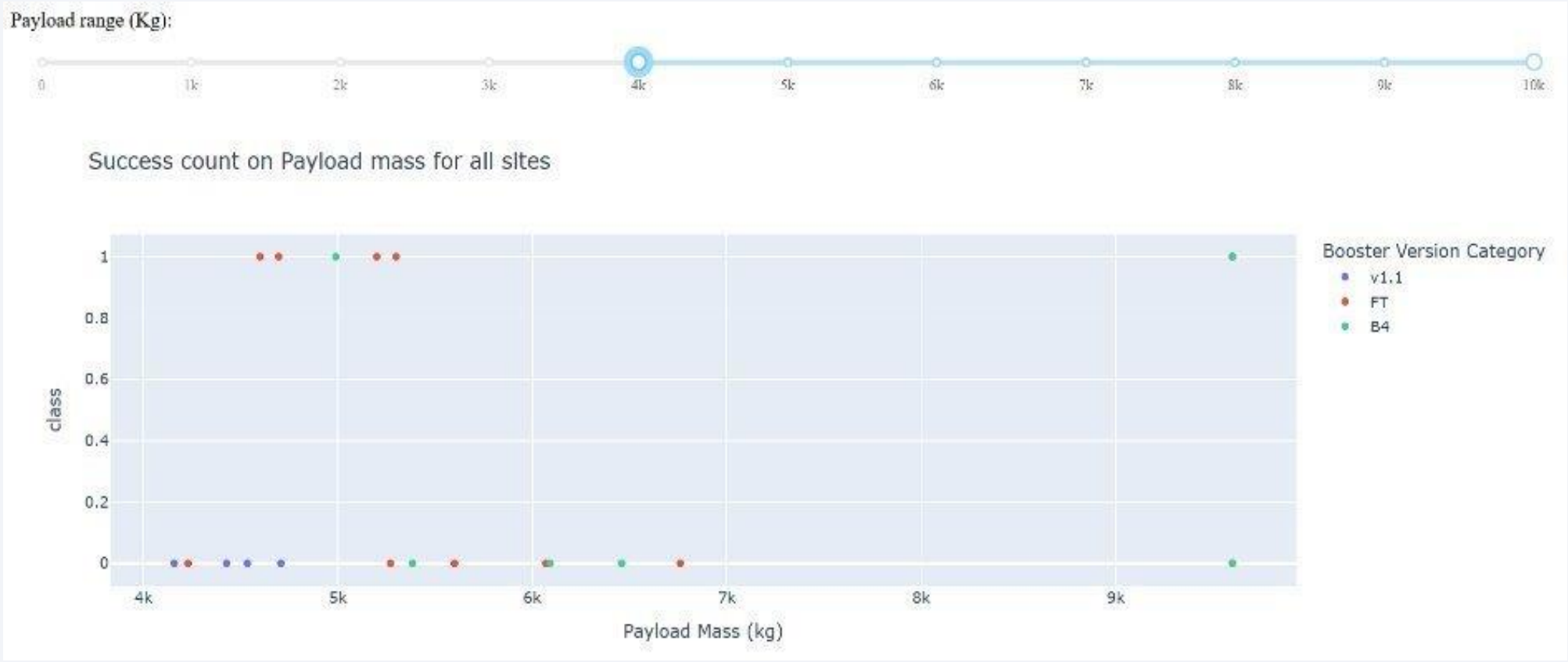


The site KSC LC-39A has the best success rate, equal to 76.9%.

Payload vs Launch Outcome, all sites, 0-4k KG

Lighter payload launches are more frequent and more successful, especially with the FT booster version.

Payload vs Launch Outcome, all sites, 4k-10k KG

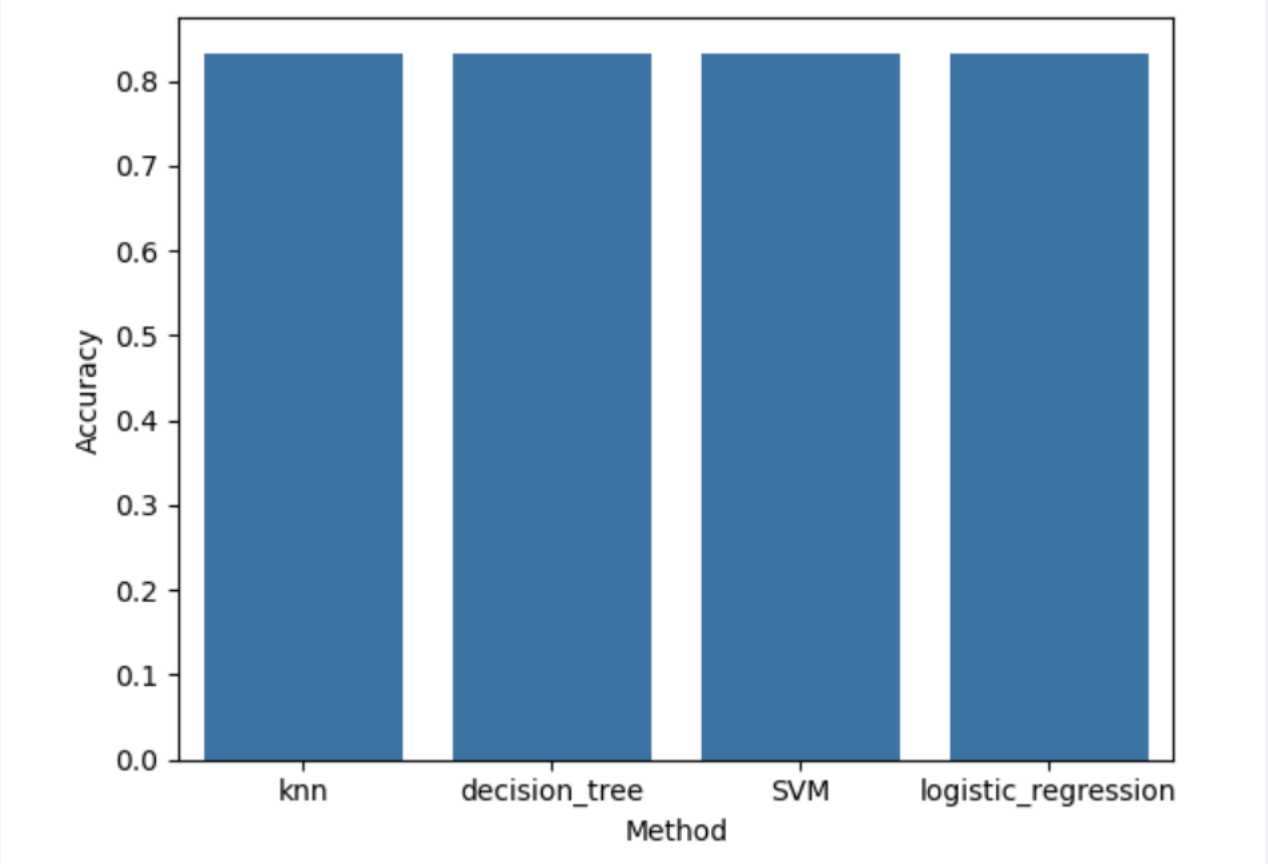


Heavier payload launches are less frequent and less successful, and are made with fewer booster versions. Version FT is the most successful with heavier payload mass.

Section 5

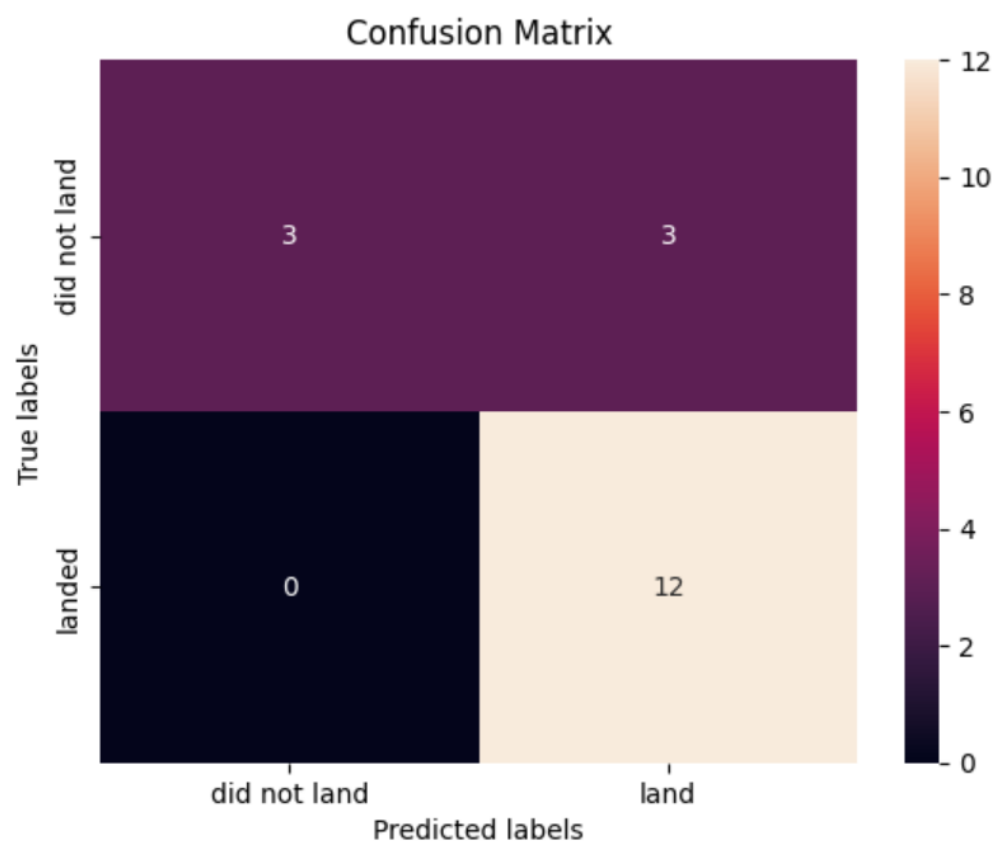
Predictive Analysis (Classification)

Classification Accuracy



Accuracy is the same for all the models.

Confusion Matrix



True Positive - 12 (True label is landed, Predicted label is also landed)

True Negative - 3 (True label is did not land, Predicted label is also did not land)

False Negative – 0 (True label is landed, Predicted label is did not land)

False Positive - 3 (True label is not landed, Predicted label is landed)

Accuracy= (12+3)/18= 0.83

Recall= 12/(12+0)=1

Precision= 12/(12+3)=0.8

Conclusions

- All models have the same accuracy, and the same confusion matrix, that means they all perform in a similar way.
- The accuracy is 83% that means that the models are correct 83 times out of 100.
- Recall is equal to 100% that means that the models are able to recognize a landed launch every time that there is one.
- Precision is equal to 80% that means that the positive predictions are correct 80 times out of 100.

Appendix

For the complete list of notebooks, for the datasets and scripts, follow the link:

Github: Applied Data Science Capstone

Thank you!