IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Luis Prieto
July/12/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection SpaceX API
  - Data collection with web scraping
  - Data wrangling
  - Exploratory analysis using SQL
  - Exploratory analysis using Pandas and Matplotlib
  - Exploratory data analysis for data visualization
  - Machine learning prediction
- Summary of all results
  - EDA Result
  - Visual analytics and dashboards
  - Predictive analysis

Introduction

- Project background and context

  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.

- Problems you want to find answers

  In this project, we will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  I worked with SpaceX launch data that is gathered from an API, specifically the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. The goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.

- Perform data wrangling

  I had to request and parse the SpaceX launch data using the GET request, then I had to filter the dataframe to only include falcon 9 launches, after that I had to deal with missing values and formatting the data.

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

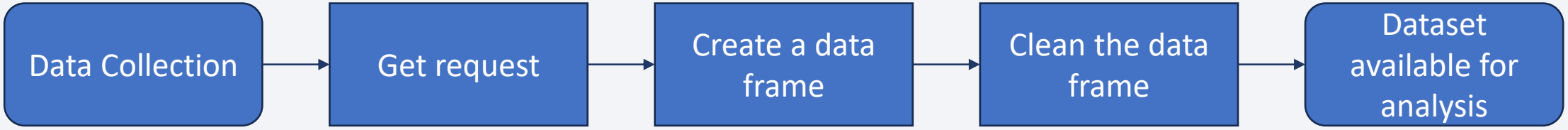
  I used machine learning to determine if the first stage of Falcon 9 will land successfully. I split the data into training data and test data to find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression. Then I had to find the method that performs best using test data.

Data Collection

- How data sets were collected.

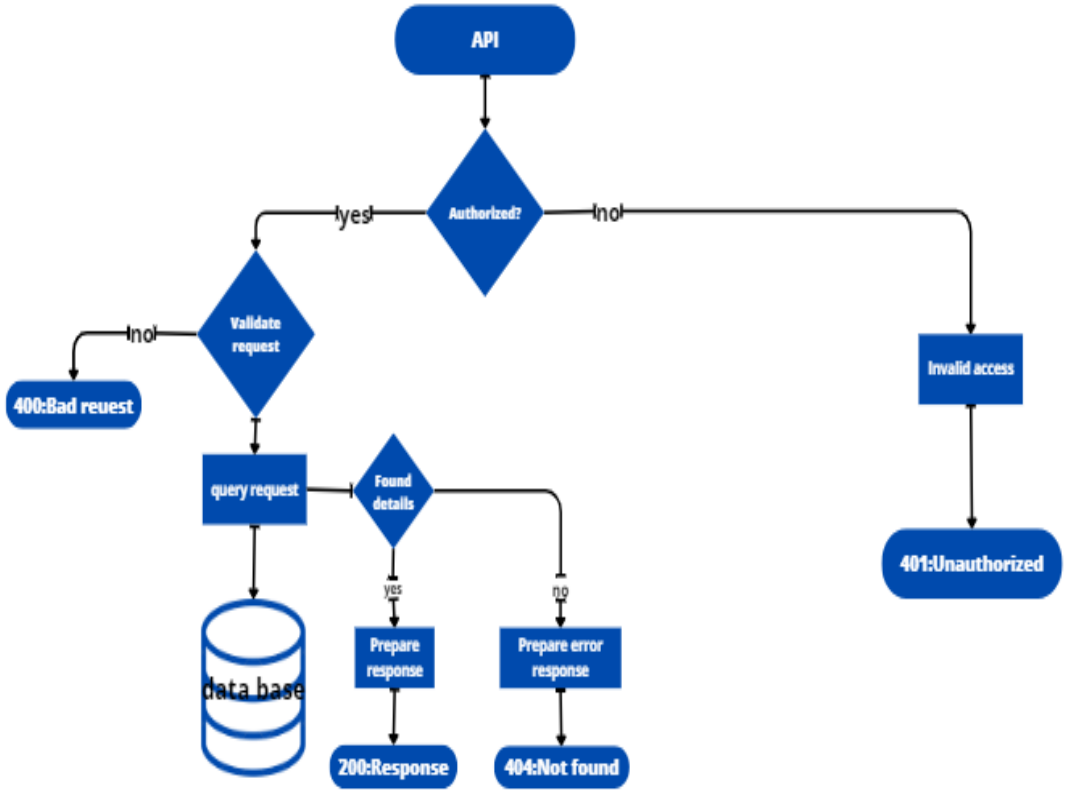
  I had to find relevant information by requesting and parsing the SpaceX launch data using the get request, the data from these requests had to be stored in lists and was used to create a new data frame, then the data frame was filter to only include falcon 9 launches, finally I had to deal with missing values and formatting to create a clean data frame.

- Data collection process flowcharts.



Data Collection – SpaceX API

- I started requesting the data from SpaceX API using the URL, then I waited the get response, once I got the successful response, I parse the SpaceX data using the GET request, then I created the data frame.
- The GitHub URL of the completed SpaceX API https://github.com/lprietol80/IBM-Data-Science-Capstone/blob/main/Lab%201_Collecting%20the%20data.ipynb



Data Collection - Scraping

- I had to used the Get request method and wait for a successful response, then I created a BeautifulSoup object from the HTML response, next I ft to extract all the columns I needed to work with, and finally I created a data frame.
- The GitHub URL: https://github.com/lprietol80/IBM-Data-Science-Capstone/blob/main/Lab%202%20Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

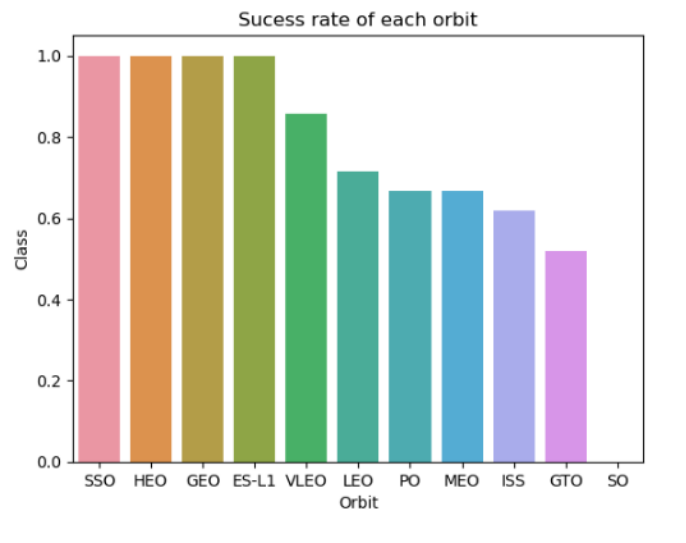
Data Wrangling

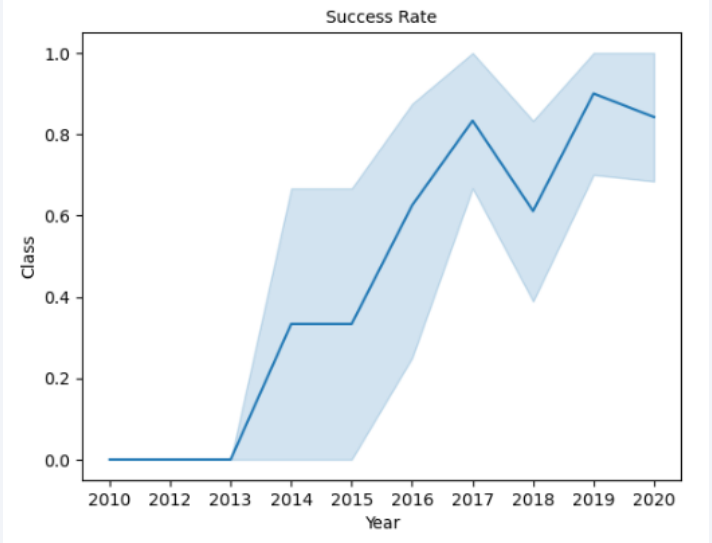
- How data were processed:
- I had to perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models. In the data set, there are several different cases where the booster did not land successfully. I n this lab I converted those outcomes into Training Labels with 1 means the booster successfully landed, and with 0 meaning it was unsuccessful. The first step was to Identify and calculate the percentage of the missing values, the next step was to identify the Space X launch facilities and calculate the number of launches on each site. Each launch aims to an orbit, so it was important to calculate the number and occurrence of each orbit and the outcome per orbit type, then it was needed to create a landing outcome label from the outcome column.
- The GitHub URL of your completed data wrangling: https://github.com/Iprietol80/IBM-Data-Science-Capstone/blob/main/Lab%203_Data%20Wrangling.ipynb

EDA with Data Visualization

- What charts were plotted and why you used those charts

  It was important to visually check if there were any relationship between success rate and orbit type, so it was created a bar chart for the success rate of each orbit, then a line chart was plotted to visually check any trend on the launches.





- The GitHub URL of the EDA with data visualization: https://github.com/lprietol80/IBM-Data-Science-Capstone/blob/main/Lab%205%20EDA%20with%20Visualization.ipynb

EDA with SQL

- The main SQL queries performed were to:
  - Display the names of the unique launch sites in the space mission.
  - 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 display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015
  - Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order
- The GitHub URL of the EDA with SQL: https://github.com/Iprietol80/IBM-Data-Science-Capstone/blob/main/Lab%204%20jupyter%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Objects I created and added to a folium map
  - ✓ I marked all the launch sites, and I created a circle around each one to highlight them on the map.
  - ✓ I created lines between the launch sites and some important point around the site such as roads, highways and cities.
  - ✓ I marked the success or failed launches for each site on the map.
- I added all theses objects to visually identify the success launch site.
- The GitHub URL of the completed interactive map with Folium map: https://github.com/Iprietol80/IBM-Data-Science-Capstone/blob/main/Lab%206%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

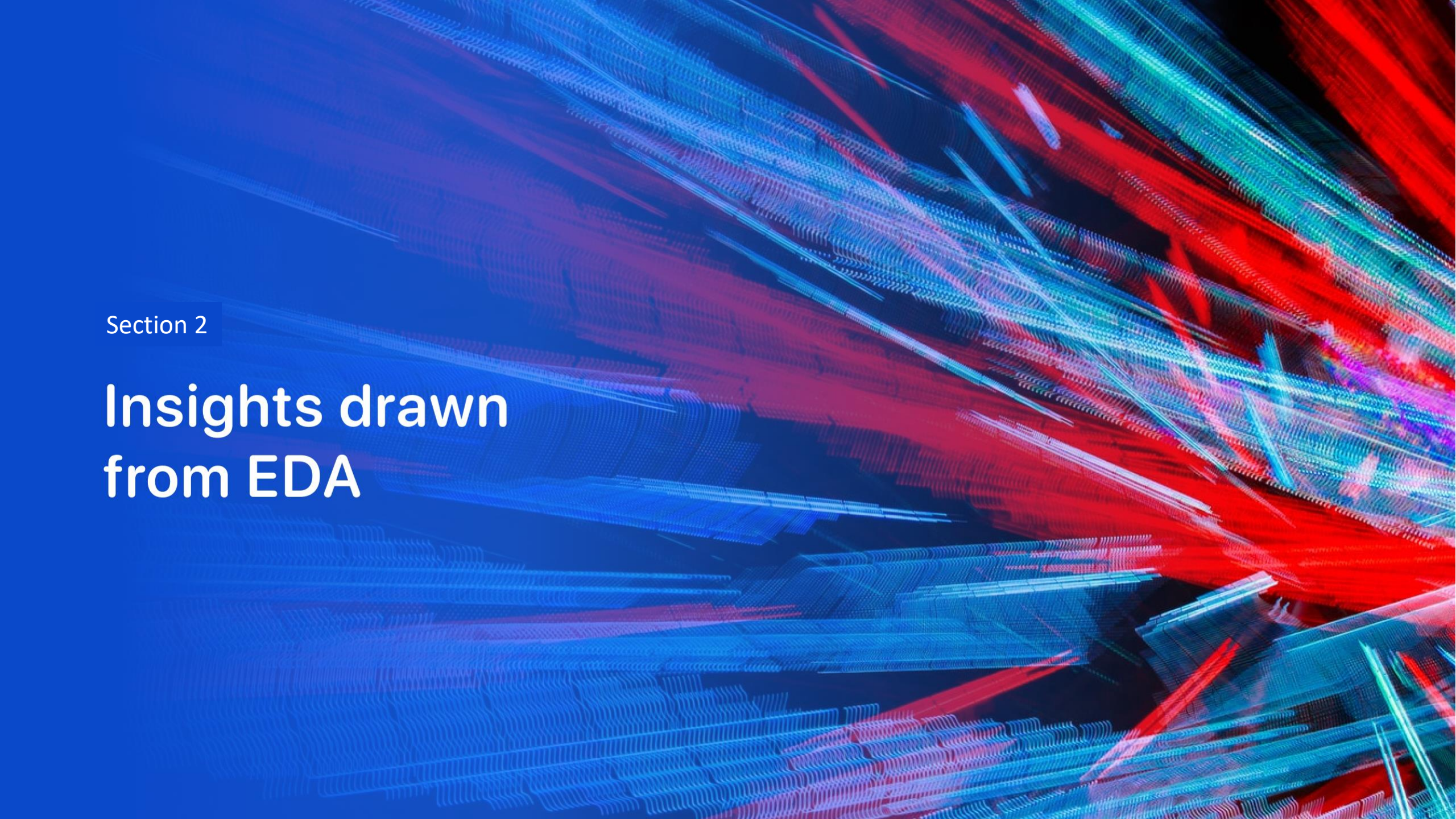
- What plots/graphs and interactions it was added to a dashboard
  - Add a Launch Site Drop-down Input Component
  - Add a callback function to render success-pie-chart based on selected site dropdown
  - Add a Range Slider to Select Payload
  - Add a callback function to render the success-payload-scatter-chart scatter plot
- The GitHub URL of the completed Plotly Dash lab, https://github.com/lprietol80/IBM-Data-Science-Capstone/blob/main/Lab7%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Predictive Analysis (Classification)

- How you built, evaluated, improved, and found the best performing classification model
  - It was loaded the data using the pandas and NumPy libraries, then the data was transformed and split into training and testing sets.
  - Different machine learning models were built and tune the hyperparameters using GridSearchCV.
  - It was used the accuracy as the metric for the model, and it was improved with models.
  - The best performing classification model was found.
- The GitHub URL of the completed predictive analysis lab, https://github.com/lprietol80/IBM-Data-Science-Capstone/blob/main/Lab8%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

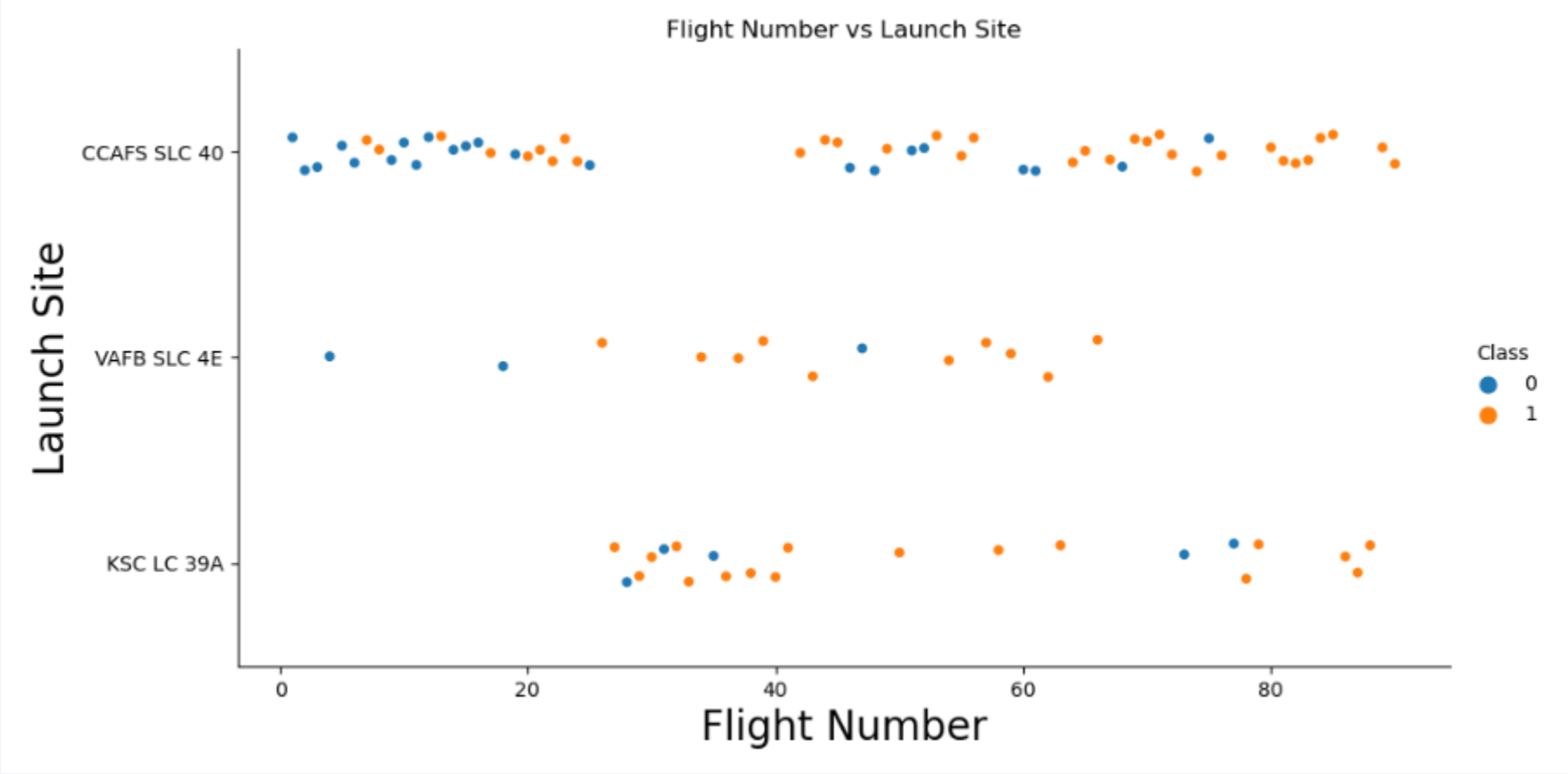
Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

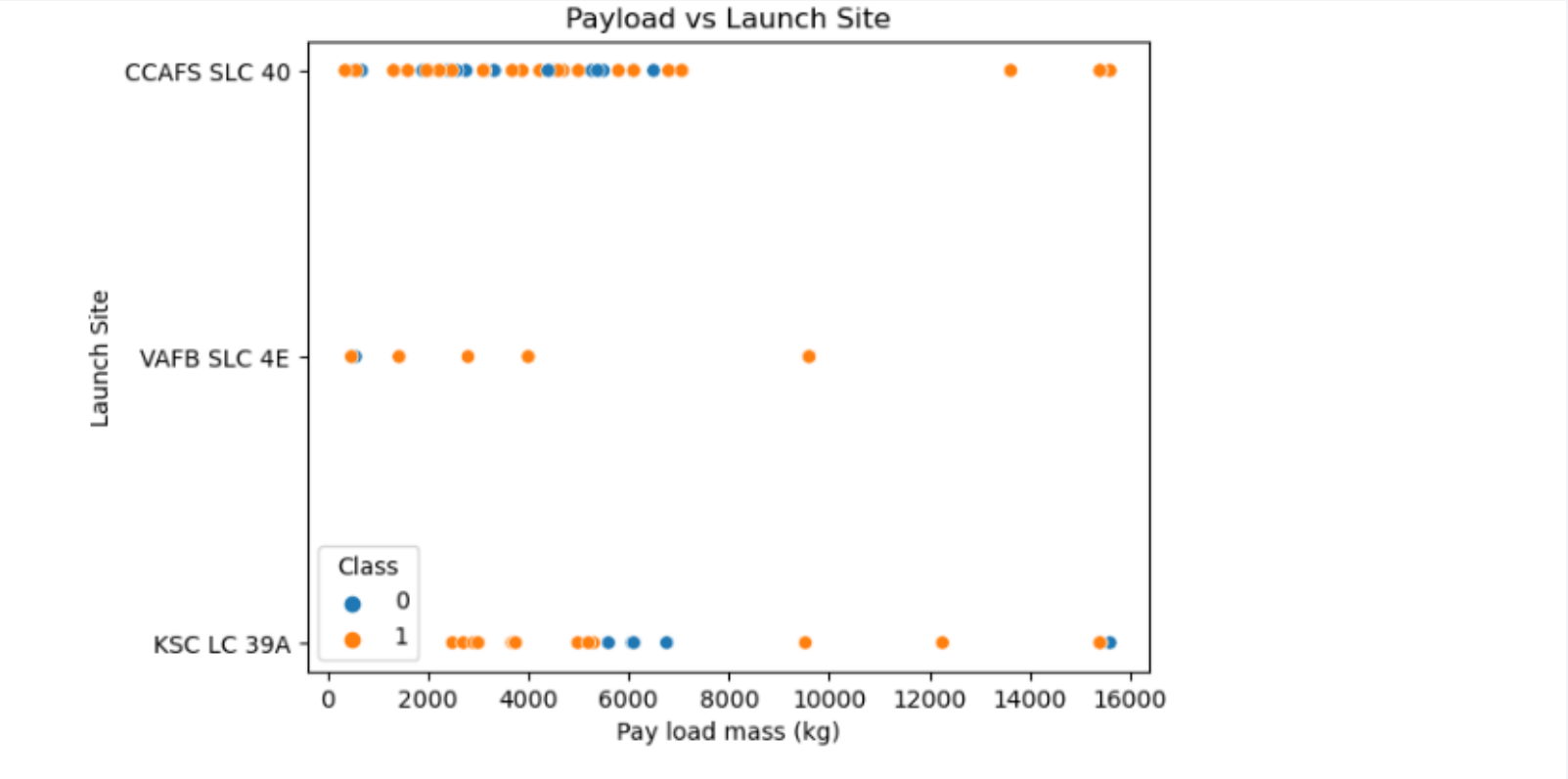
- The sites with best success rate was CCAFS SLC 40, especially after flight number 80.



Payload vs. Launch Site

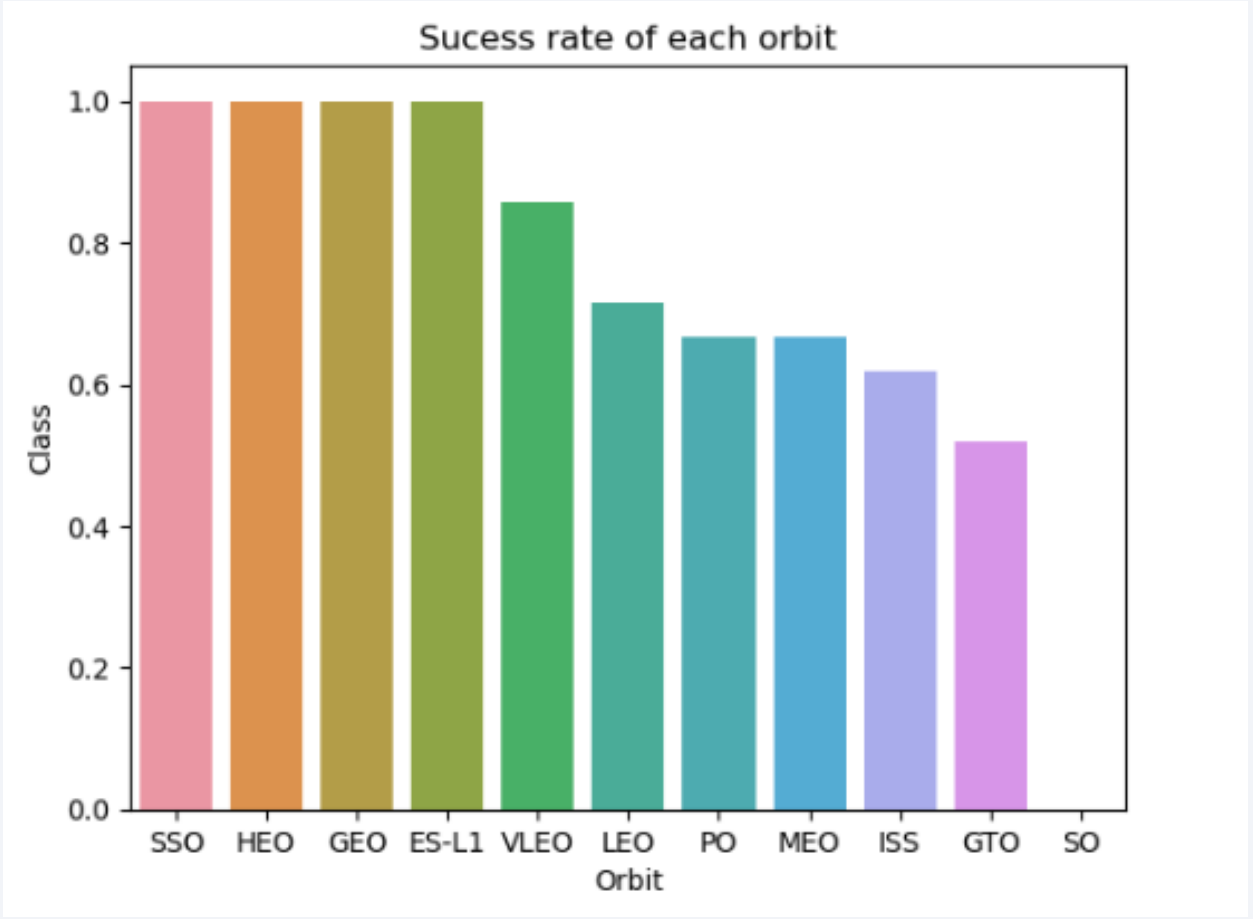
- Show a scatter plot of Payload vs. Launch Site

The launch site with highest payload success is CCAFS SLC 40



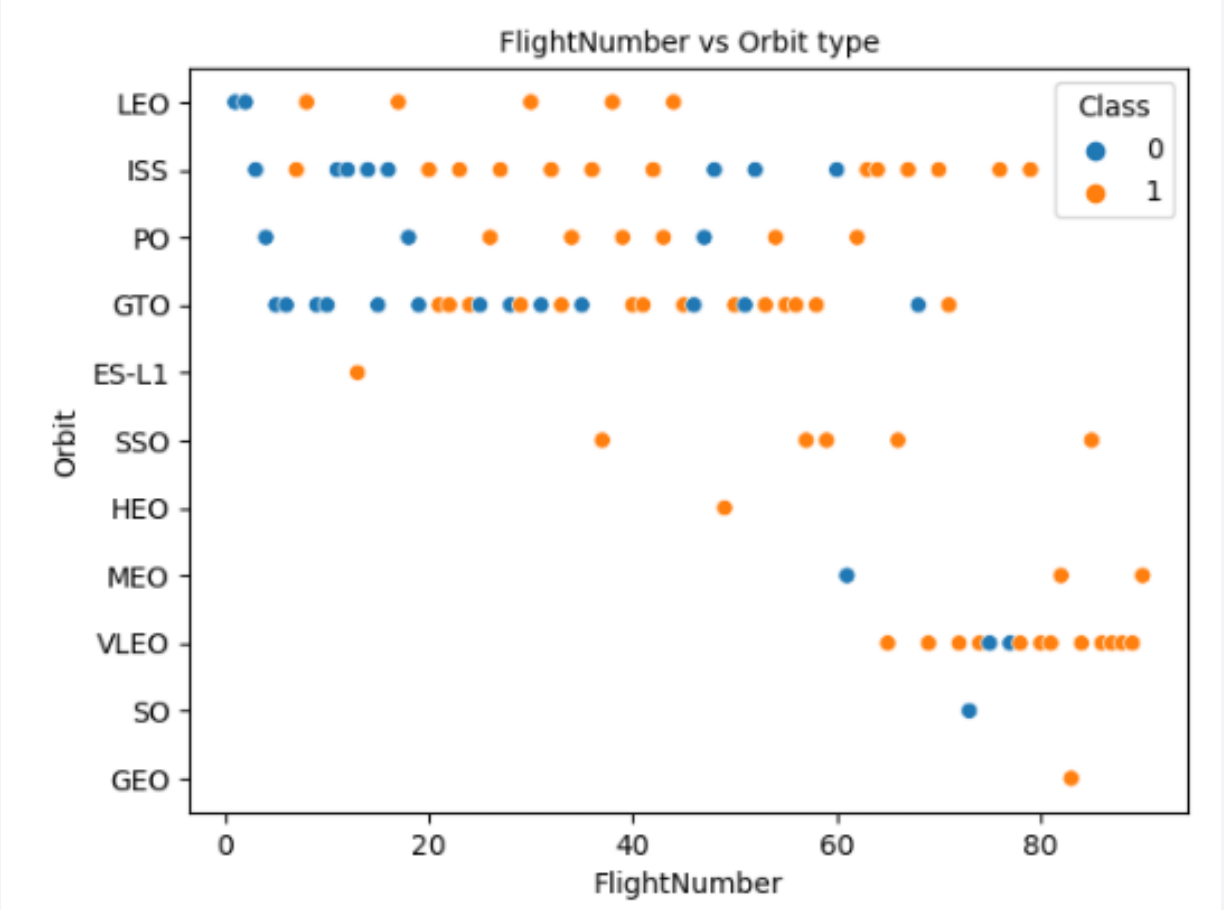
Success Rate vs. Orbit Type

- In this visualization we can see the success rate is for launches to the orbits SSO,HEO,GEO and ES-L1



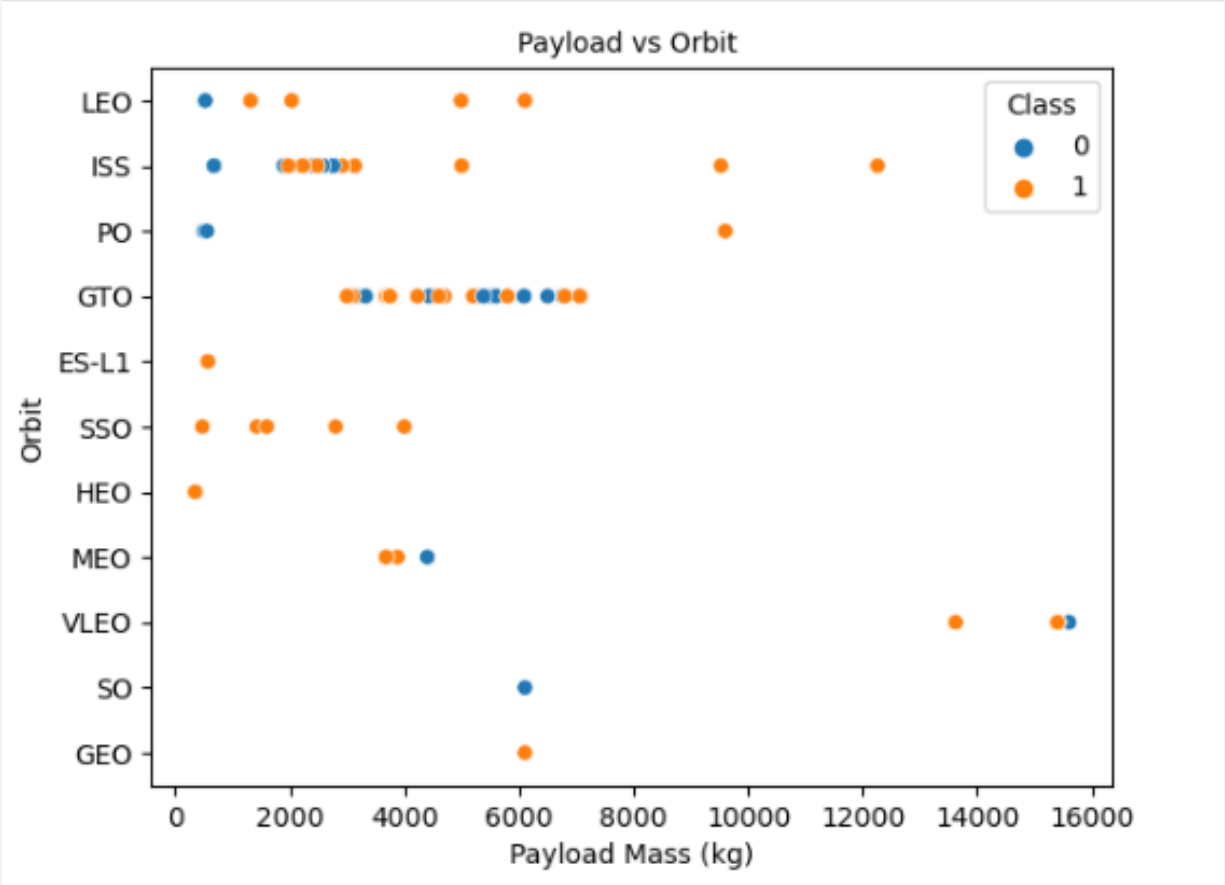
Flight Number vs. Orbit Type

- We can observe that most of the flight were to orbits ISS and GTO and the they are the orbits with the highest number of successful launches.



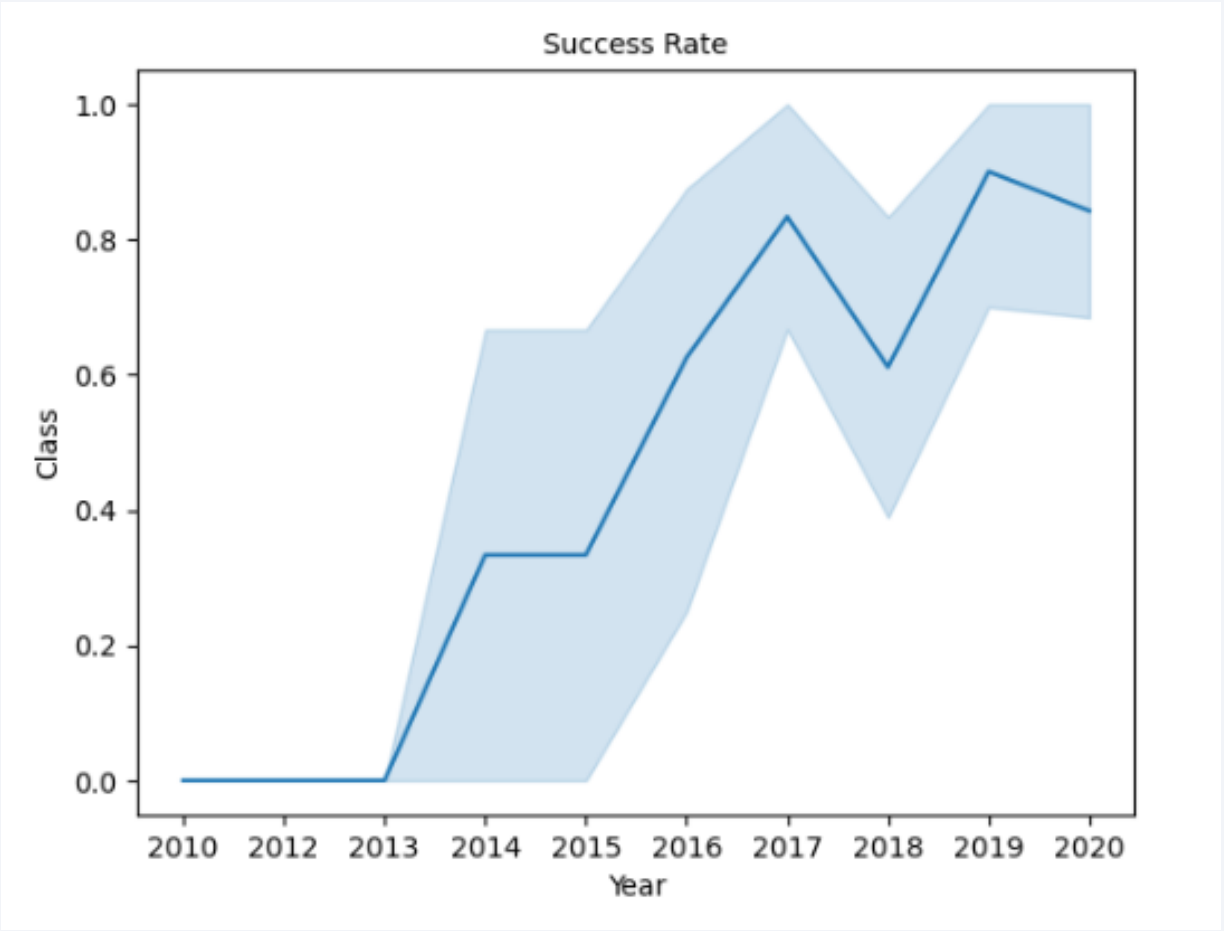
Payload vs. Orbit Type

- The highest payload mass was launched to orbits VLEO and ISS and most of the launches were successful.



Launch Success Yearly Trend

- Since 2013 the success rate of launches increased each year until 2020.



All Launch Site Names

- It was used the DISTINCT statement to return only the distinct values from the "Launch_Site" column.

```
1 %sql SELECT DISTINCT "Launch_Site" FROM "SPACEXTBL"
2
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Here we used the WHERE clause to filter the launch sites beginning with “CCA” and we also used the LIMIT to view only 5 records.

```
1 %sql SELECT "Launch_Site" FROM "SPACEXTBL" WHERE "Launch_Site" LIKE "CCA%" LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Here we used the SUM function to get the total payload carried by boosters from the customer NASA.

```
1 %sql SELECT SUM(`PAYLOAD_MASS__KG_`) AS Total_Payload FROM `SPACEXTBL` WHERE `Customer`="NASA (CRS)"
```

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

Total_Payload
45596.0

Average Payload Mass by F9 v1.1

- It was used the AVG function to calculate the average payload mass carried by booster version F9 v1.1

```
1 %sql SELECT avg(`PAYLOAD_MASS__KG_`) AS Average_Payload FROM `SPACEXTBL` WHERE `Booster_Version`="F9 v1.1"
```

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

Average_Payload
2928.4

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad is 01/08/2018

```
1 %sql SELECT MIN(`Date`) AS Early_Date FROM `SPACEXTBL` WHERE `Landing_Outcome`="Success (ground pad)"
```

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

Early_Date
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
1 %sql SELECT `Customer`,`Booster_Version`,`PAYLOAD_MASS__KG_`, `Landing_Outcome` FROM `SPACEXTBL` WHERE `Landing_Outcome`="S
```

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

Customer	Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
SKY Perfect JSAT Group	F9 FT B1022	4696.0	Success (drone ship)
SKY Perfect JSAT Group	F9 FT B1026	4600.0	Success (drone ship)
SES	F9 FT B1021.2	5300.0	Success (drone ship)
SES EchoStar	F9 FT B1031.2	5200.0	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- It was used the count function and the GROUP BY statement to group the result by column “Mission_Outcome” to get the number of success and failures.

```
1 %sql SELECT `Mission_Outcome`, count(*) as Total FROM `SPACEXTBL` GROUP BY `Mission_Outcome`
```

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

Mission_Outcome	Total
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- I was used DISTINCT and WHERE and a subquery to find he maximum payload mass.

```
1 %sql SELECT DISTINCT `Booster_Version`,`PAYLOAD_MASS__KG_` FROM `SPACEXTBL` WHERE `PAYLOAD_MASS__KG_`=(SELECT MAX(`PAYLOAD_
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

- It was used the substr (date,4,2) to find the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.

```
]:  1 %sql SELECT substr(Date, 4,2) as Months,`Booster_Version`,`Launch_Site`, `Landing_Outcome` FROM `SPACEXTBL` where `Landing_O
```

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

]:

Months	Booster_Version	Launch_Site	Landing_Outcome
10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- To 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 it was used Count, Where, like, between, group by and order by to get he results.

```
1 %sql SELECT `Landing_Outcome`, COUNT(`Landing_Outcome`) as Total_Success_Landing FROM SPACEXTBL WHERE `Landing_Outcome` li
```

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

Landing_Outcome	Total_Success_Landing
Success	20
Success (drone ship)	8
Success (ground pad)	7

Section 3

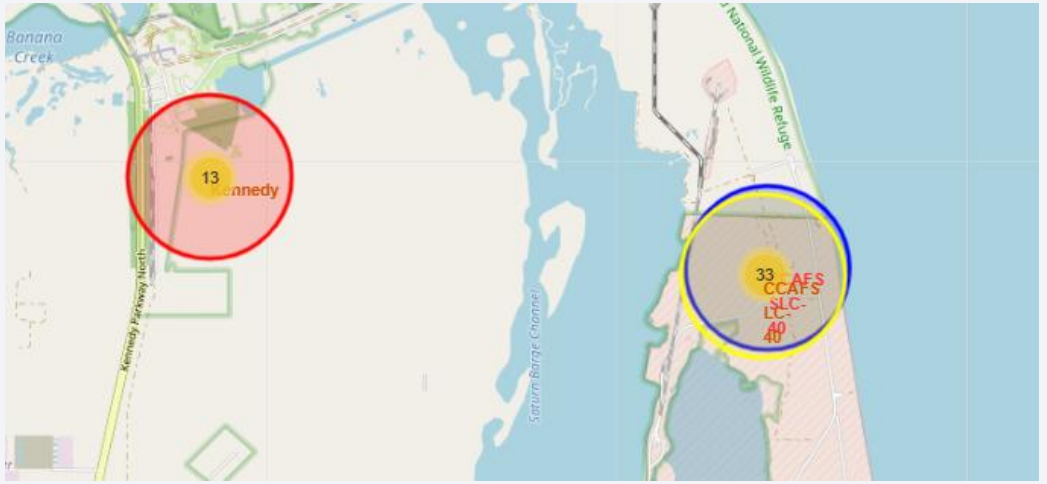
Launch Sites Proximities Analysis

Mark all launch sites on a map

- Folium was used to mark all launch sites on the map, in the visual we can see that all launch sites are very close to the coast.

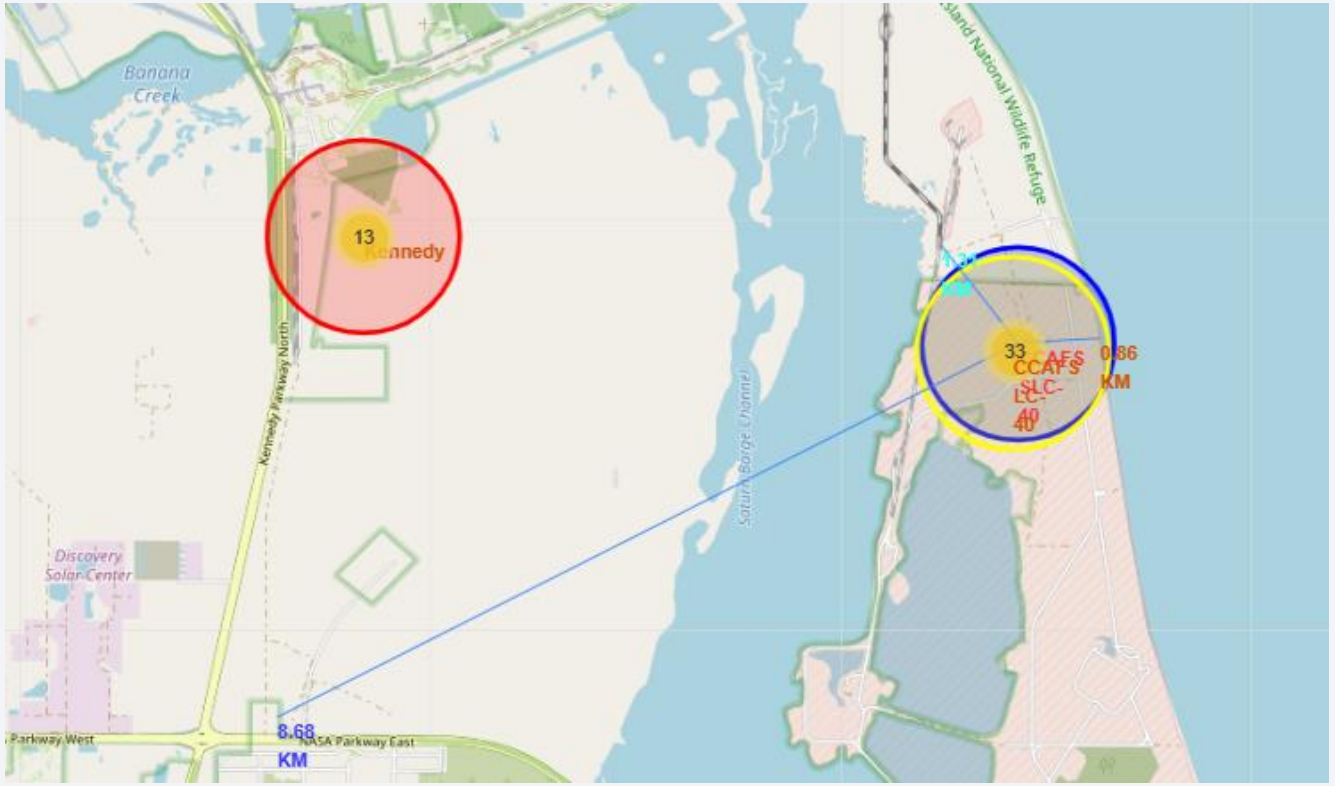
Mark outcomes for each site on the map

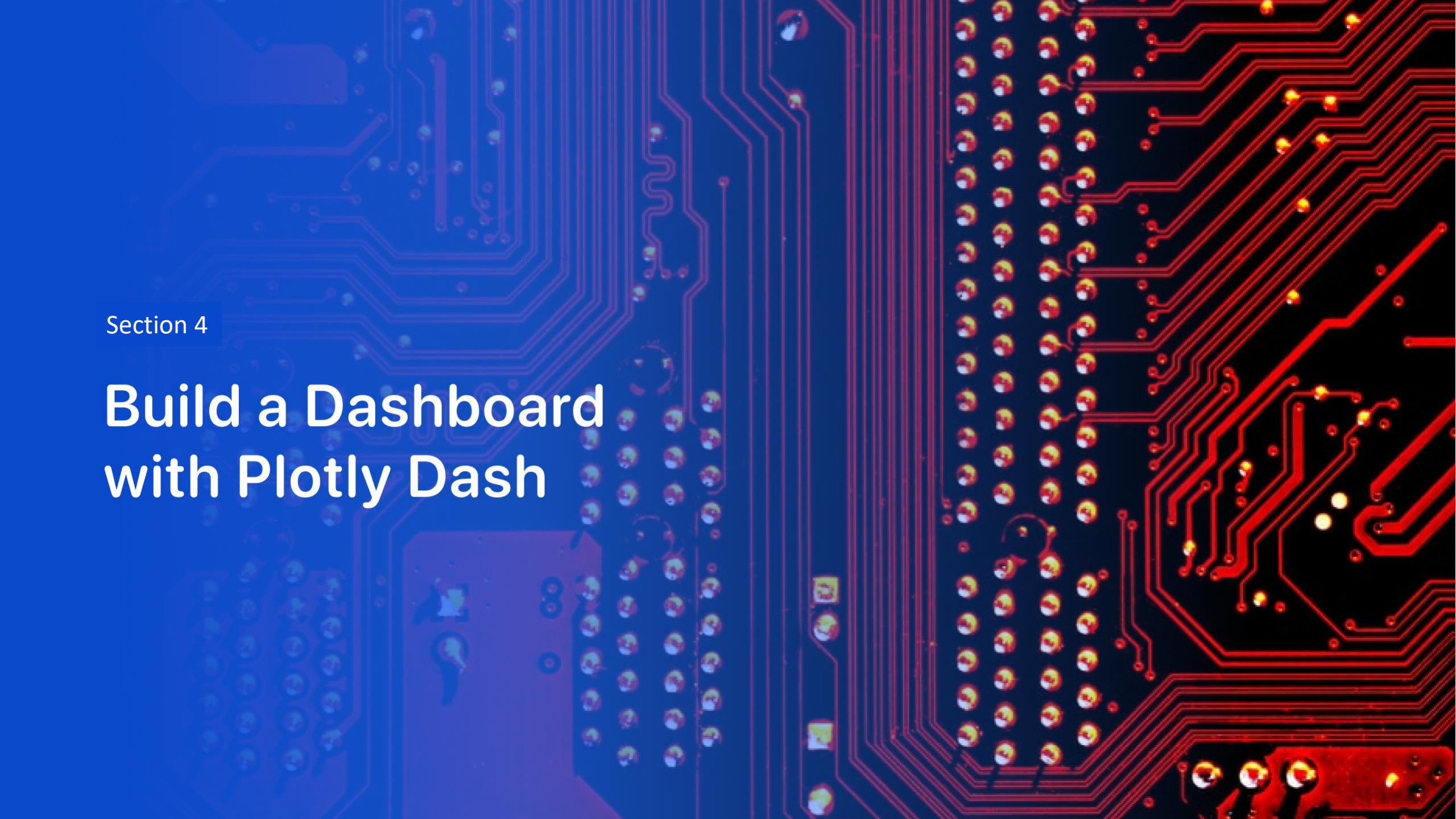
- The map shows the color-labeled launch outcomes on the map.



Distances between a launch site to its proximities

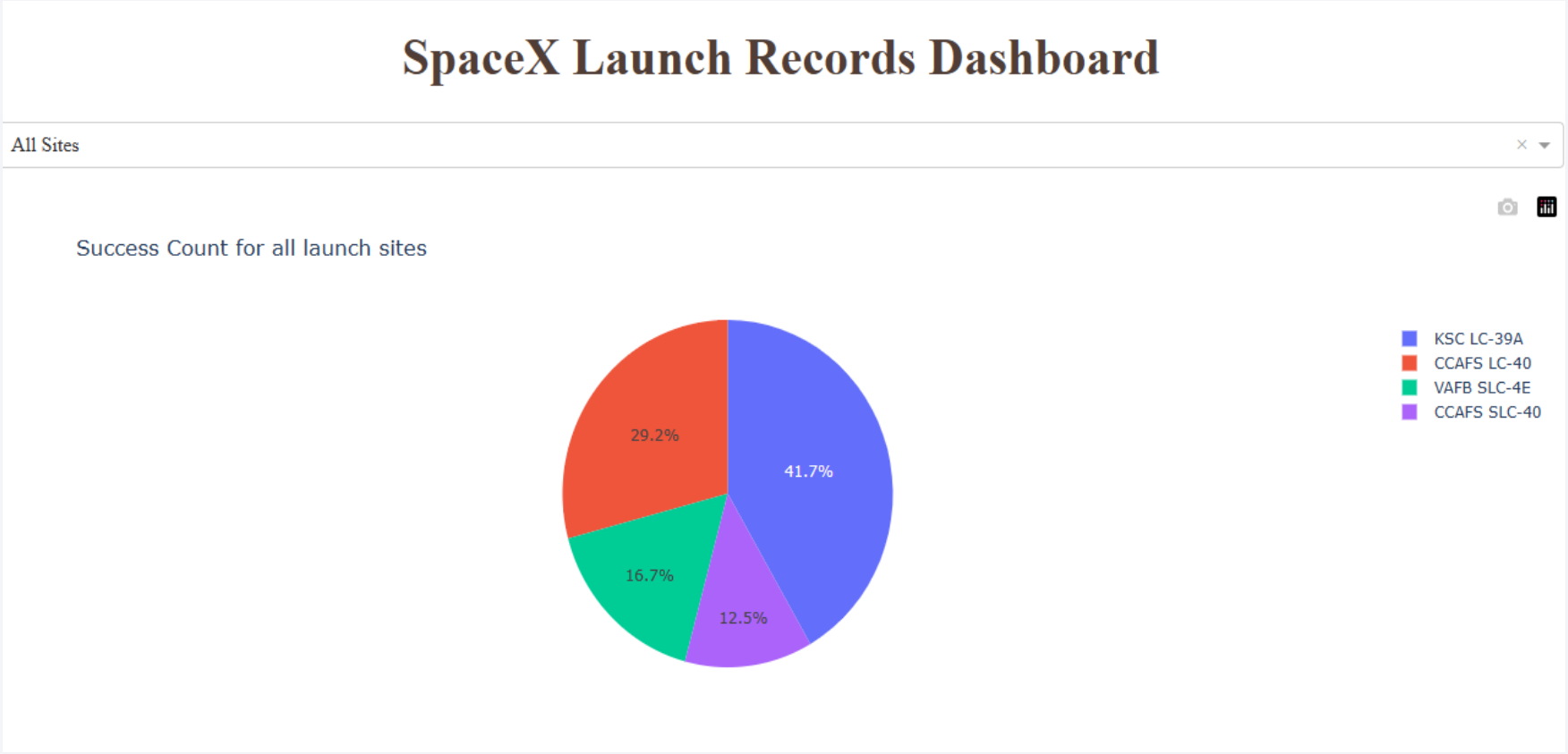
- The map shows CCAFS SLC 40 launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed.




Section 4
Build a Dashboard with Plotly Dash

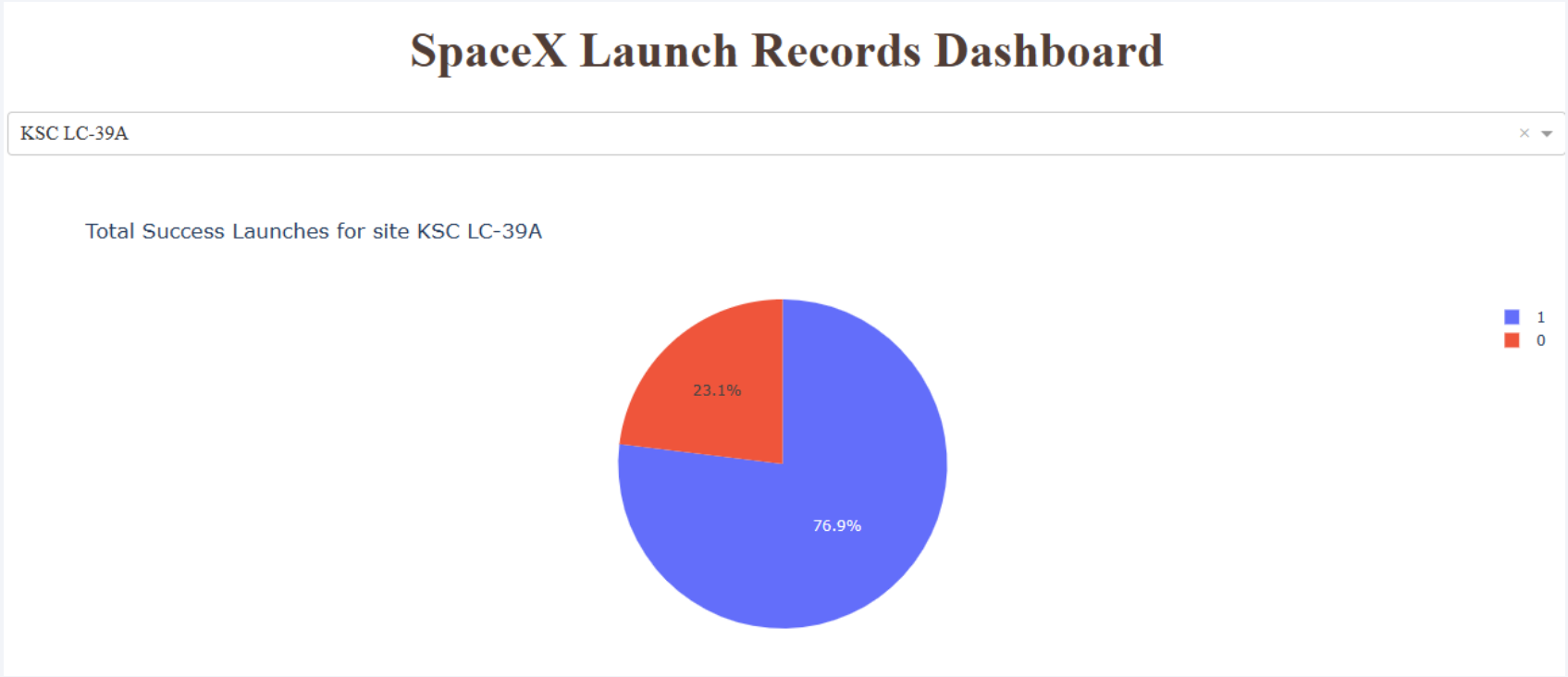
Launch success count

- The launch site KSC LC 39A has the 41.7% of success while launch site CCAFS SLC 40 has the 12.5% of success.



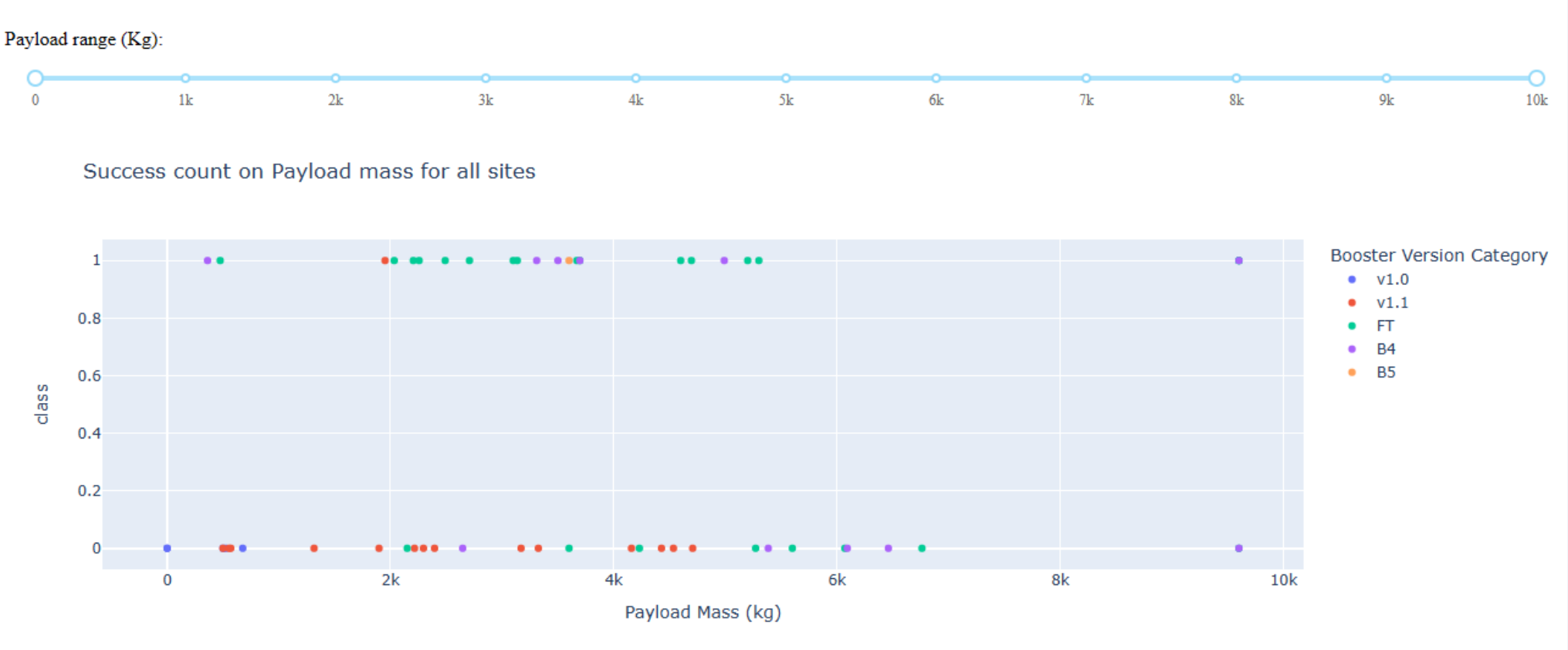
Launch site with highest launch success ratio

- The success ratio is 76.9% while the failures is only 23.1%.



Payload vs Launch Outcome

- The Booster version B4 has the largest success rate which payload range is between 8k to 10k.

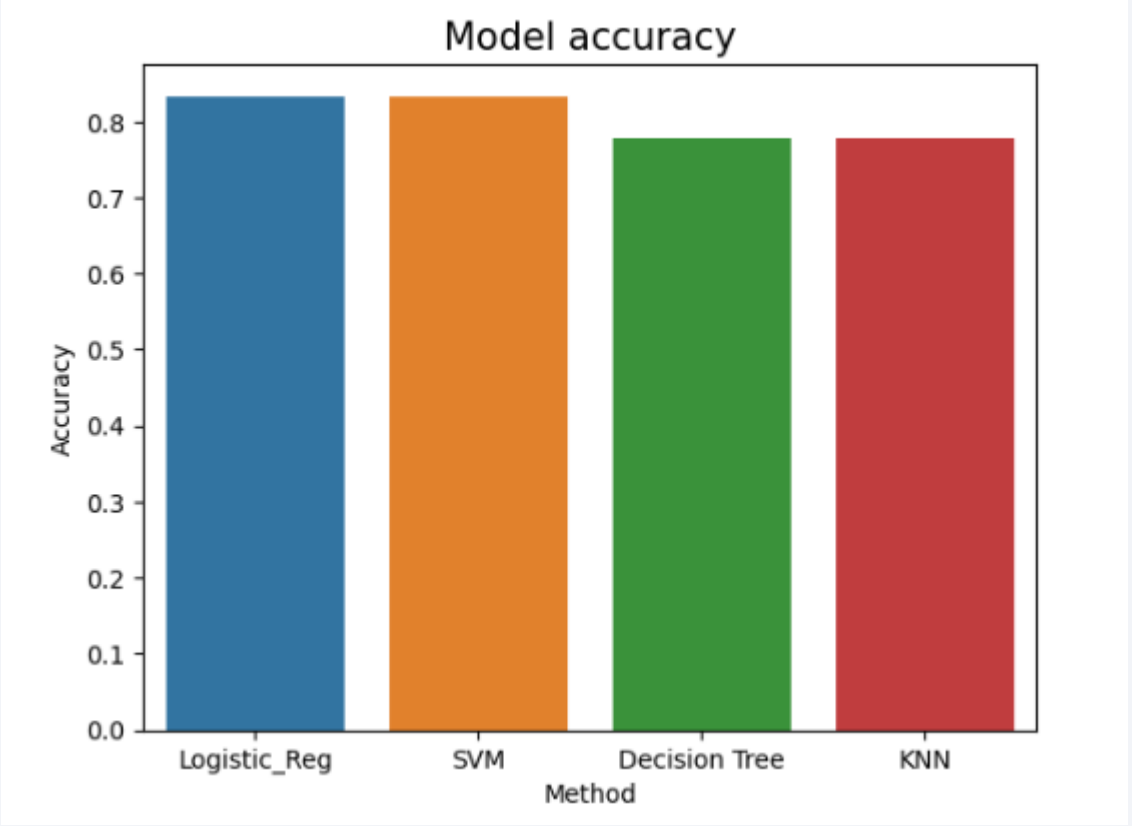


Section 5

Predictive Analysis (Classification)

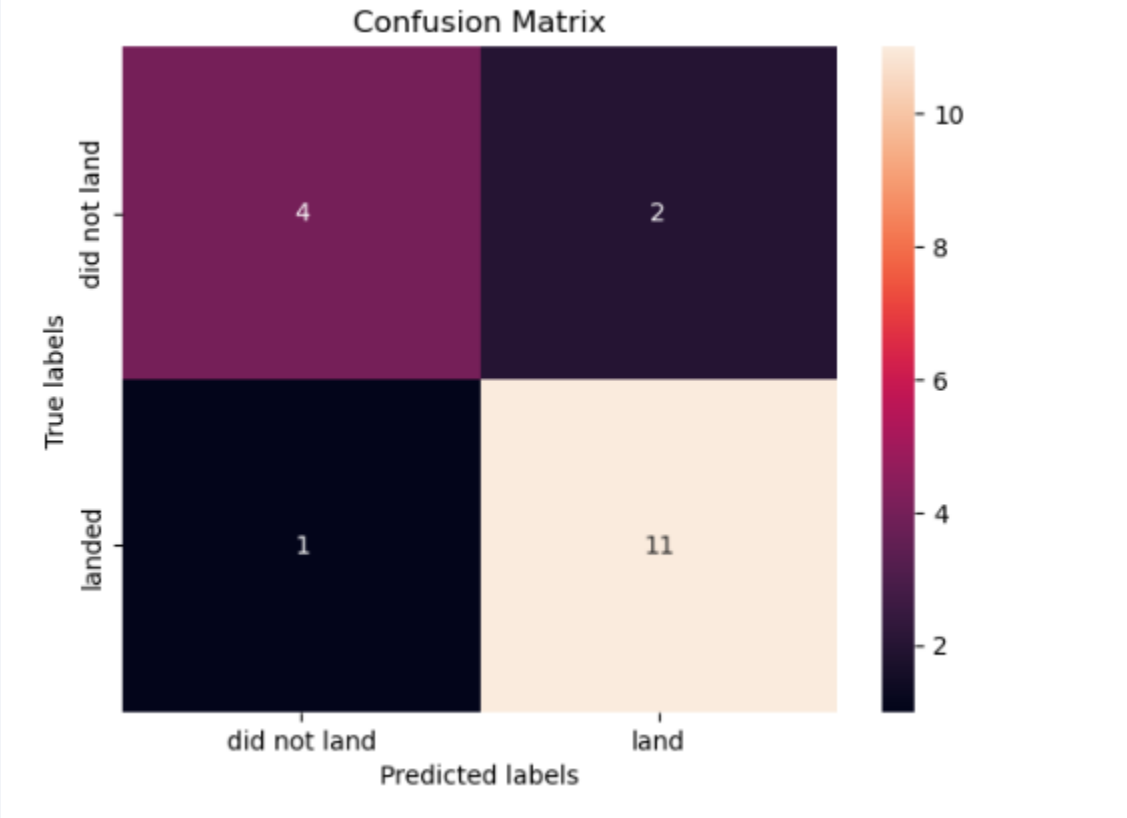
Classification Accuracy

- The models with the highest classification accuracy are Logistic Regression and SVM



Confusion Matrix

- Confusion matrix of the best performing model



Conclusions

- Models Logistic Regression and SVM have the best accuracy.
- We can predict the success of the launch of the competitor and determine the cost of the launch.

Appendix

- Python code
- SQL query
- Pandas library
- NumPy

Thank you!