IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dung Minh Nguyen
28/05/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with API and Web Scraping and Data Wrangling
  - Exploratory Data Analysis (EDA) using SQL and Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Data collection results
  - EDA results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- Project background and context

  The objective of this project is to make predictions about the successful landing of the Falcon 9 first stage. SpaceX, on its website, states that the cost of launching a Falcon 9 rocket is $62 million, while other providers charge more than $165 million for each launch. The significant price difference is attributed to SpaceX's ability to reuse the first stage. By determining the likelihood of a successful landing, we can estimate the cost of a launch. This information holds importance for any company intending to challenge SpaceX in the rocket launch market.

- Problems you want to find answers
  - What are the main characteristics of a successful or failed landing?
  - What are the effects of each relationship of the rocket variables on the success or failure of a landing ?
  - What are the conditions which will allow SpaceX to achieve the best landing success rate?

Section 1

Methodology

Methodology

Executive Summary

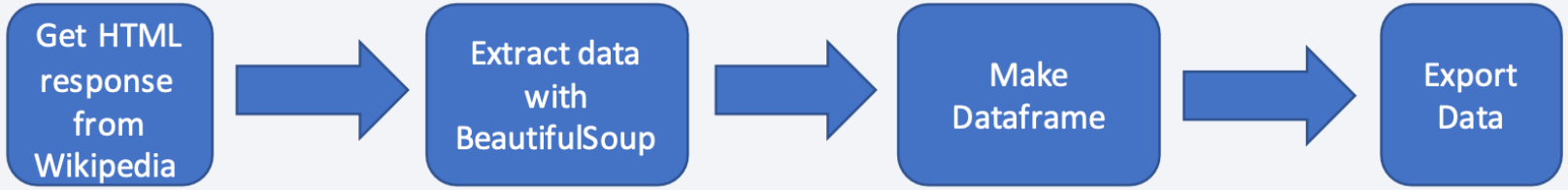
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Datasets are collected via Rest SpaceX REST API and Web Scraping on Wikipedia

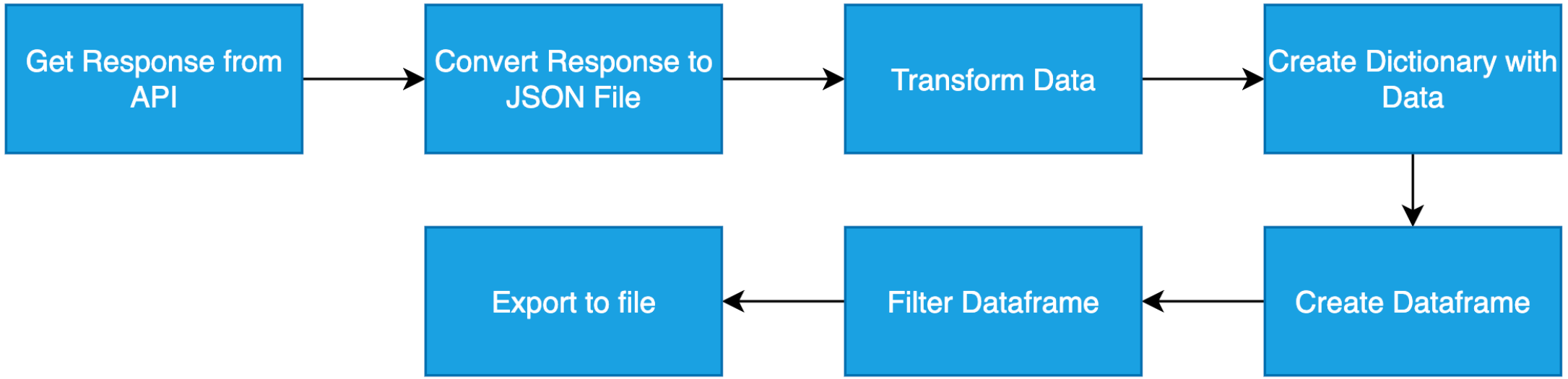
1. Rest SpaceX REST API: https://api.spacexdata.com/v4/rockets

2. Web Scraping: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

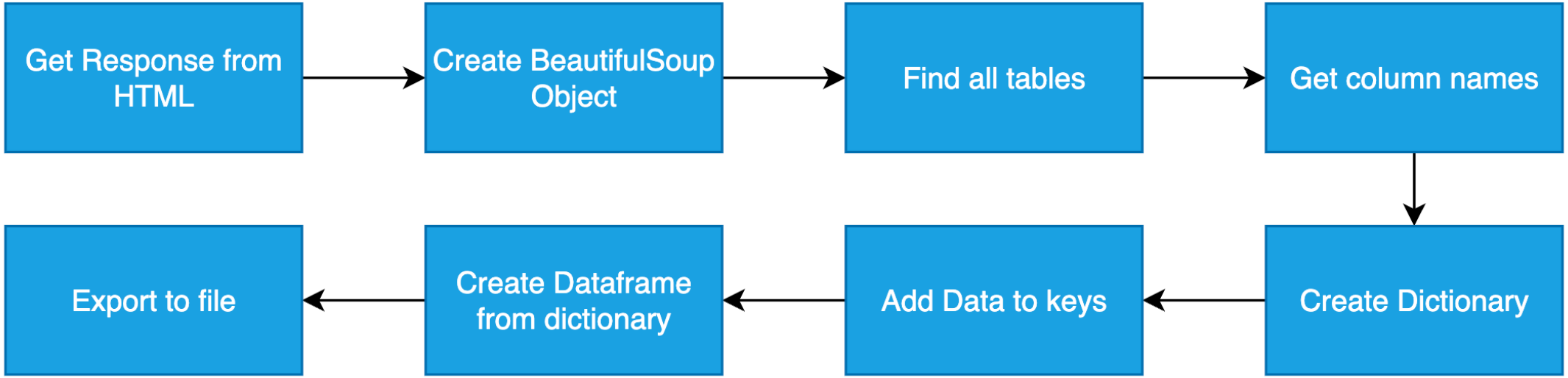
Flowcharts of data collection with SpaceX REST:



GitHub URL of the completed SpaceX API calls notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

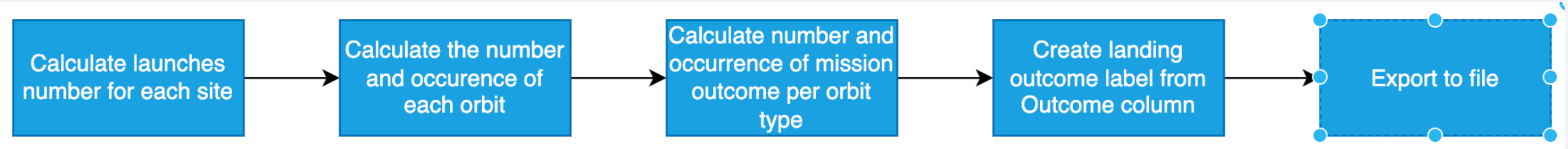
Flowcharts of data collection with Web Scraping:



GitHub URL of the completed web scraping notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Convert the outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Flowcharts of data wrangling:



GitHub URL of the completed web wrangling notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- What charts were plotted:
  - Scatter Graphs: Scatter plots show relationship between variables. This relationship is called the correlation.
  - Bar Graphs: Bar graphs show the relationship between numeric and categoric variables.
  - Line Graphs: Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data

GitHub URL of the completed EDA with data visualization notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Summarize the SQL queries you performed:
  - 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)
  - List the date when the first successful landing outcome in ground pad was achieved
  - 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 successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL of the completed EDA with SQL notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas
  - Red circle at NASA Johnson Space Center's coordinate with a label showing its name (fo/ium.Circle, folium.map.Marker).
  - Red circles at each launch site coordinate with label showing launch site name (fo/ium.Circle, folium.map.Marker, folium.features.Divicon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, folium./con).
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.Divicon)
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings, and the number of successful and unsuccessful landings.

GitHub URL of the completed Interactive Map with Folium notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

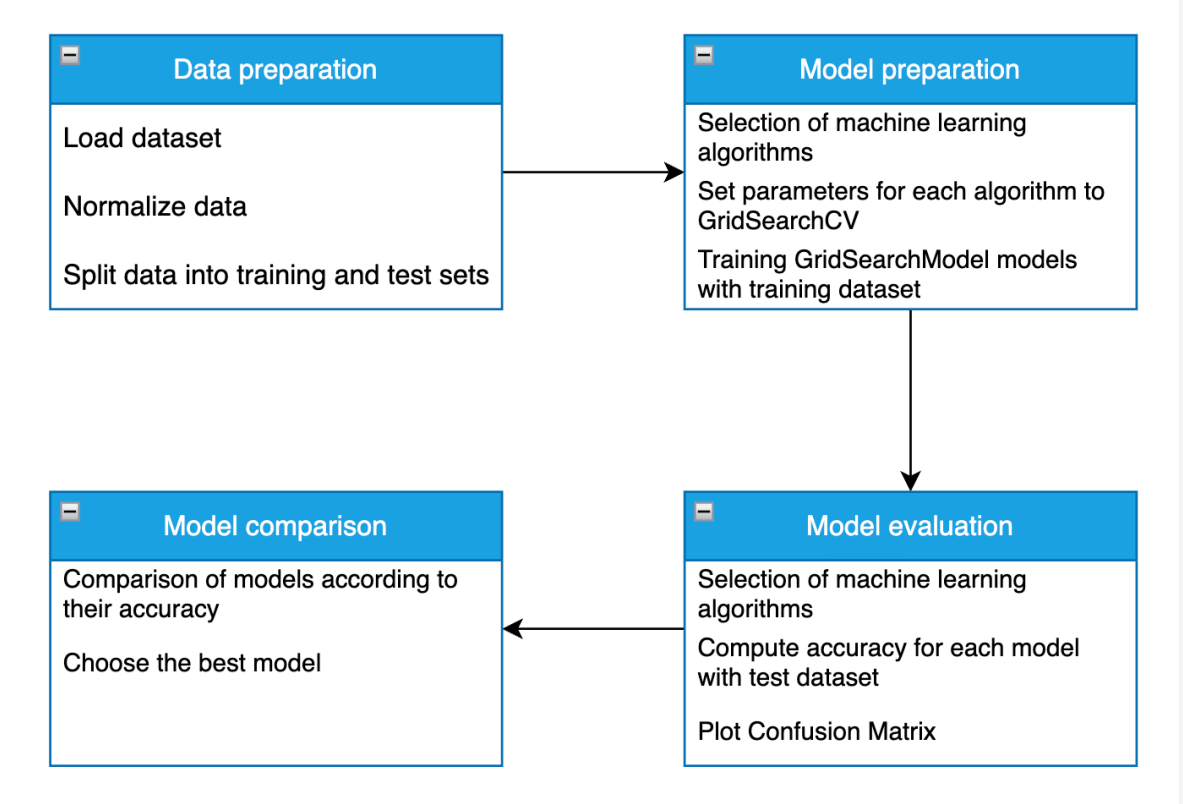
Build a Dashboard with Plotly Dash

- Build a Dashboard with Plotly Dash
  - Dashboard has dropdown, pie chart, range slider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (p/otly. express.pie)
  - Rangeslider allows a user to select a payload mass in a fixed range (dash_core_components.KangeSlider).
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (p/otly.express.scatter).

GitHub URL of the completed Interactive Map with Folium notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

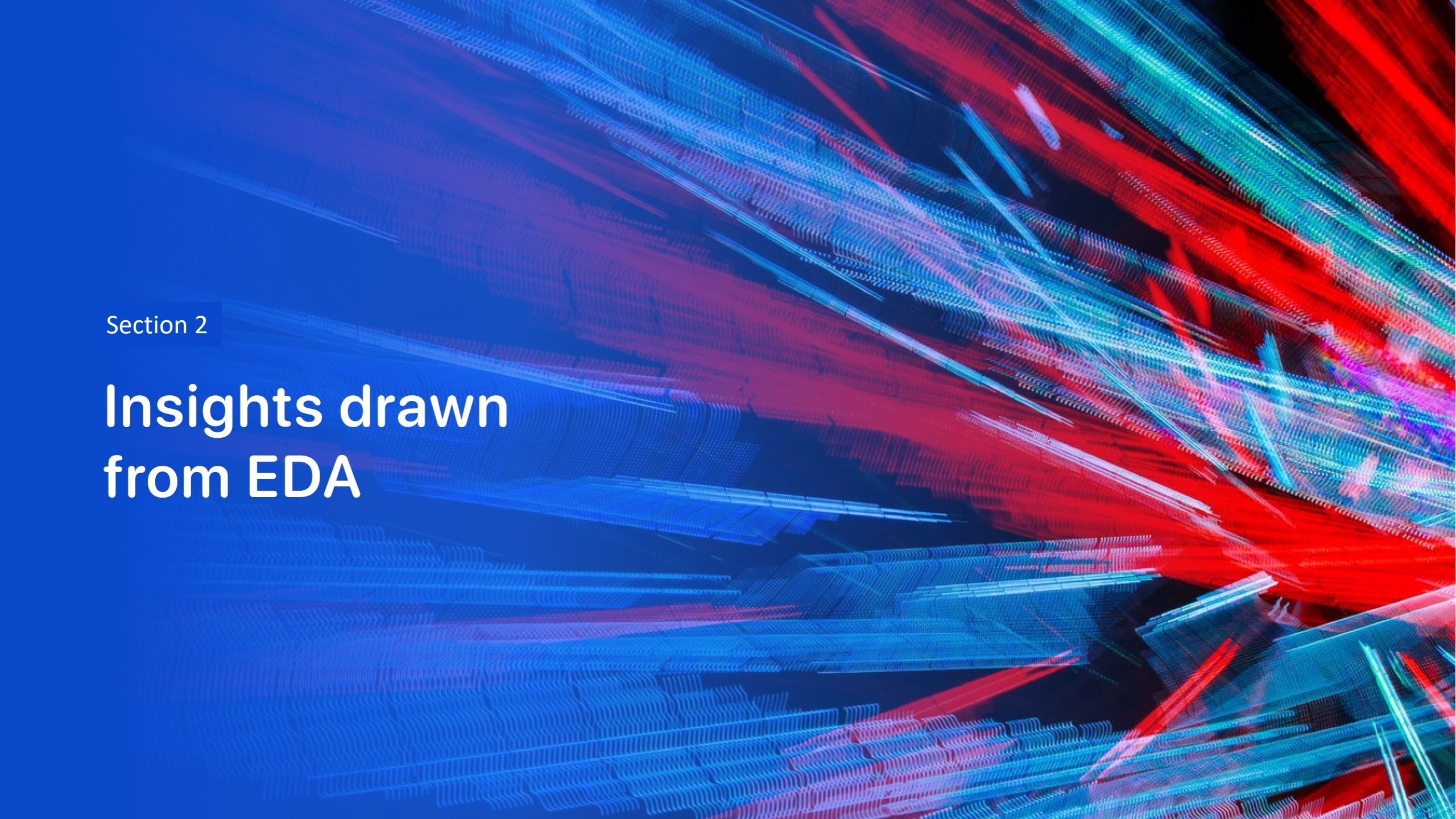
Flowcharts of Predictive Analysis:



GitHub URL of the completed Predictive Analysis notebook:
https://github.com/dungmn/Data-Science-Final-Project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

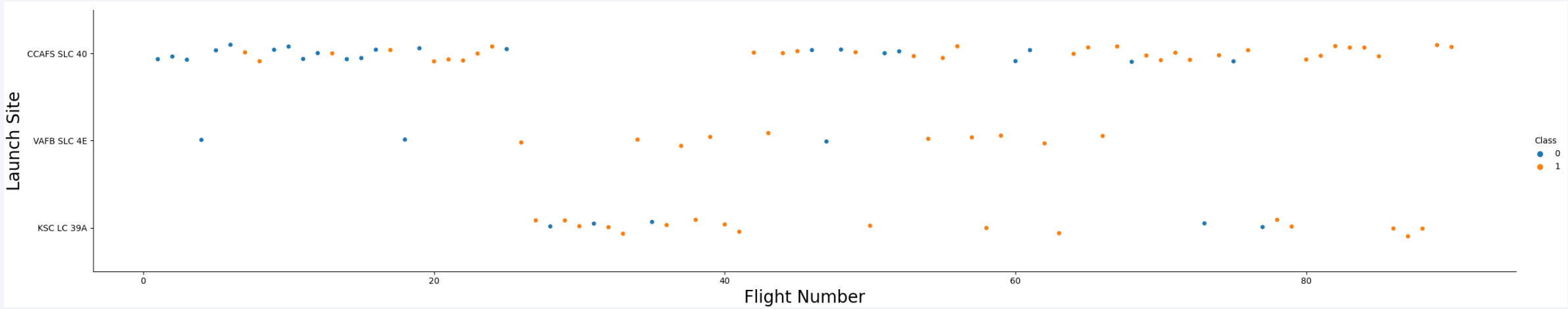
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- The scatter plot of Flight Number vs. Launch Site

```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

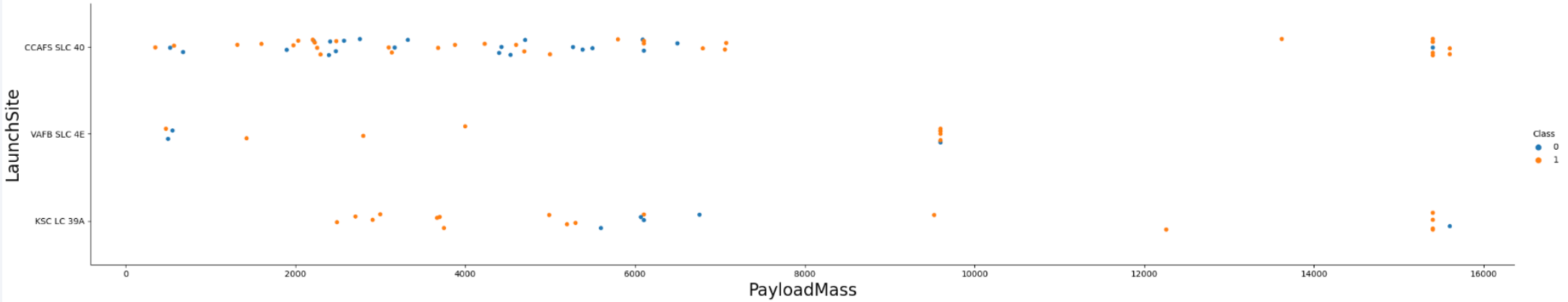


- We observe that, for each site, the success rate is increasing.

Payload vs. Launch Site

- The scatter plot of Payload vs. Launch Site

```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

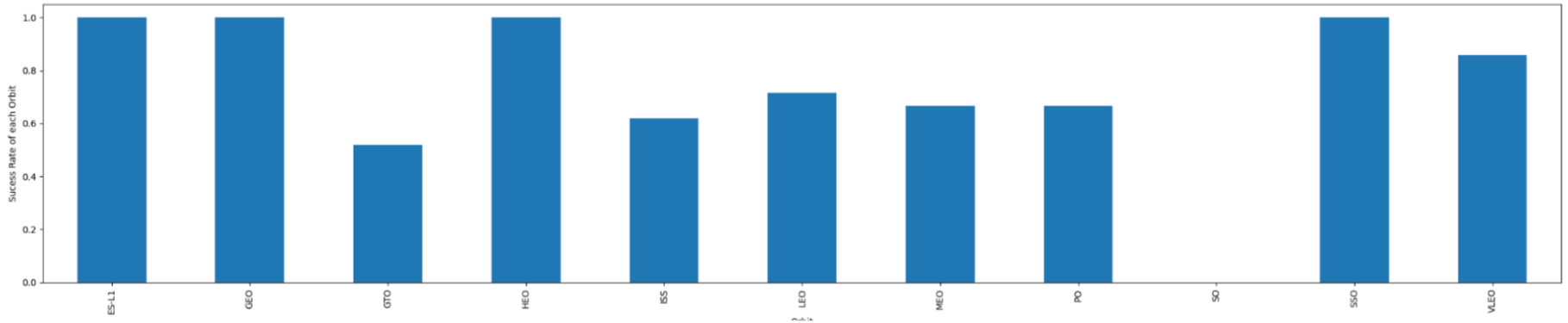


- Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)

Success Rate vs. Orbit Type

- The bar chart for the success rate of each orbit type

```
### TASK  3: Visualize the relationship between success rate of each orbit type
pl = df.groupby('Orbit')['Class'].mean()
ax = pl.plot(kind='bar')
ax.set_xlabel("Orbit")
ax.set_ylabel("Sucess Rate of each Orbit")
plt.show()
```

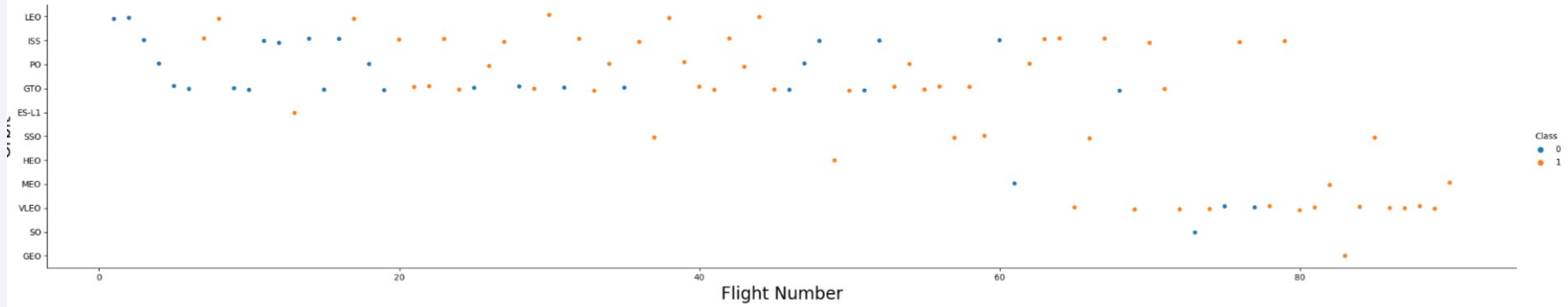


- Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type

- The scatter point of Flight number vs. Orbit type

```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

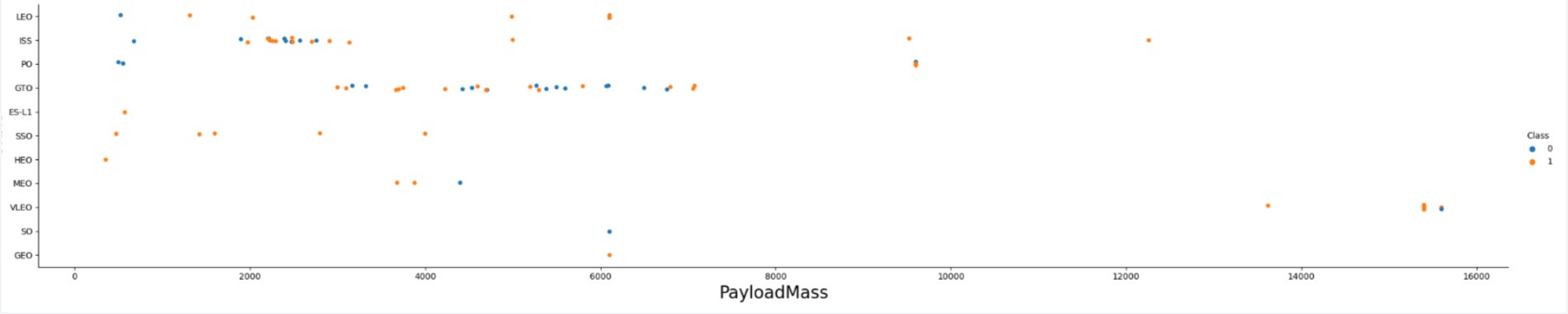


- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- The scatter point of payload vs. orbit type

```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



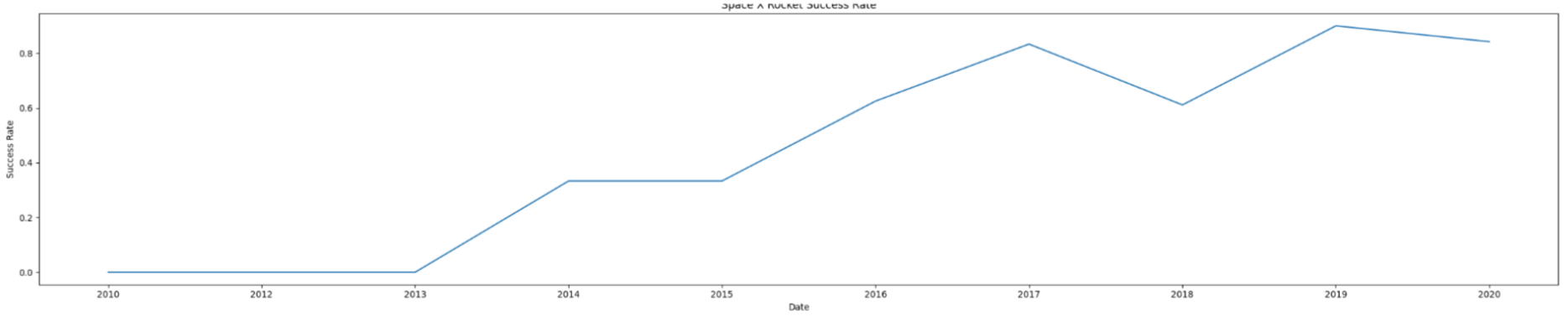
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- The line chart of yearly average success rate

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate

df_groupby_year = df.groupby("Date",as_index=False)["Class"].mean()
sns.lineplot(data = df_groupby_year, x="Date", y="Class")
plt.xlabel("Date")
plt.title('Space X Rocket Success Rate')
plt.ylabel("Success Rate")
plt.show()
```



- You can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- The names of the unique launch sites

```
%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

- The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

- The 5 records where launch sites begin with `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	Lan
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Fai
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Fai
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	

- The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering

Total Payload Mass

- The total payload carried by boosters from NASA

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

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

SUM("PAYLOAD_MASS__KG_")
45596.0

- This query returns the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

- The 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

- This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad

```
%sql SELECT MIN("Date") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success%'
```

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

MIN("Date")
01/07/2020

- With this query, we select the oldest successful landing.
- The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

SUCCESS	FAILURE
100	1

- With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass
- We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

```
%sql SELECT DISTINCT "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

2015 Launch Records

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

MONTH	Booster_Version	Launch_Site
10	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

- 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("LANDING_OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING_OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING_OUTCOME" \
ORDER BY COUNT("LANDING_OUTCOME") DESC ;
```

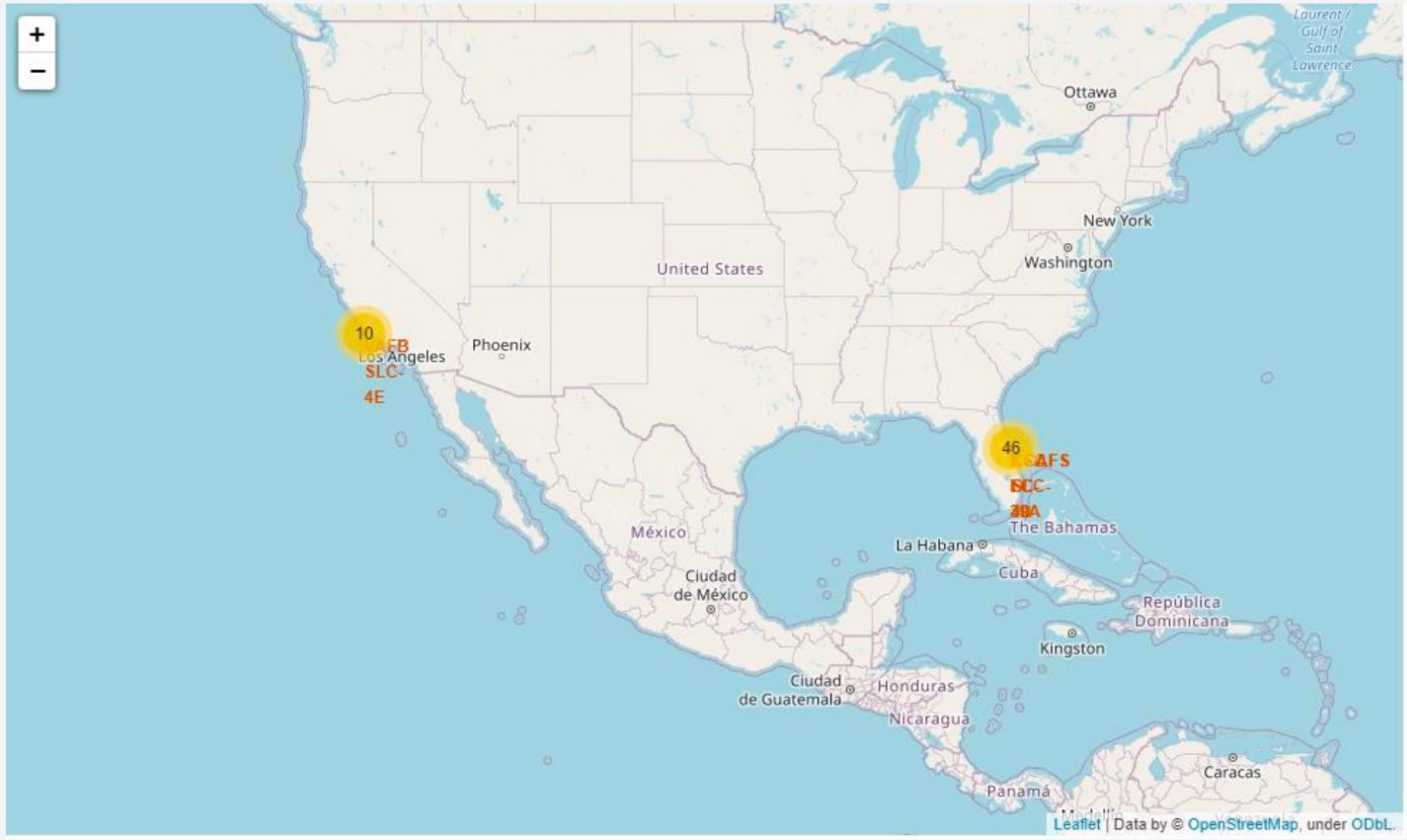
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT("LANDING_OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	7

- This query returns landing outcomes and their count where mission was successful, and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

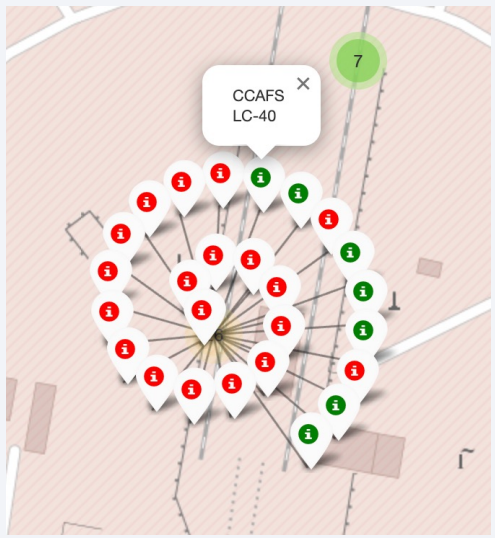

Section 3
Launch Sites
Proximities Analysis

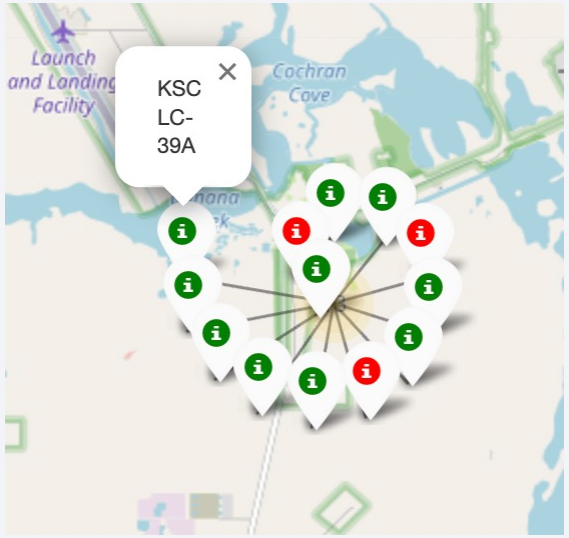
Folium map – Ground stations

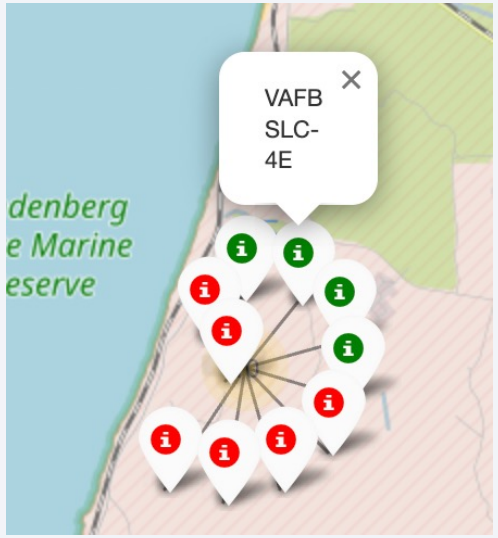


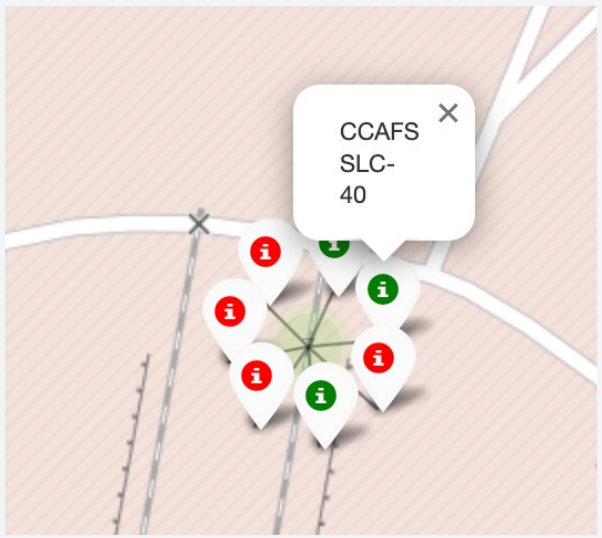
- We see that Space X launch sites are located on the coast of the United States

Folium map – Color Labeled Markers



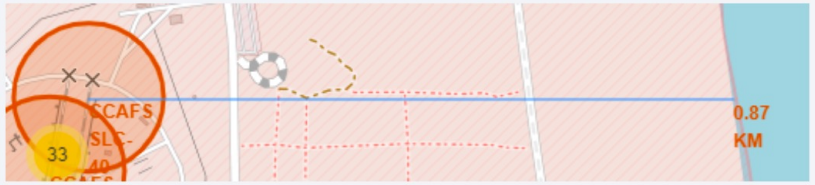


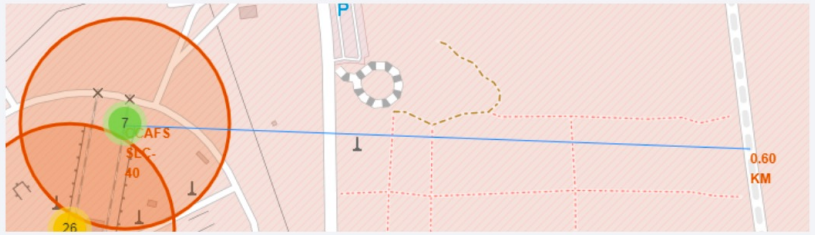




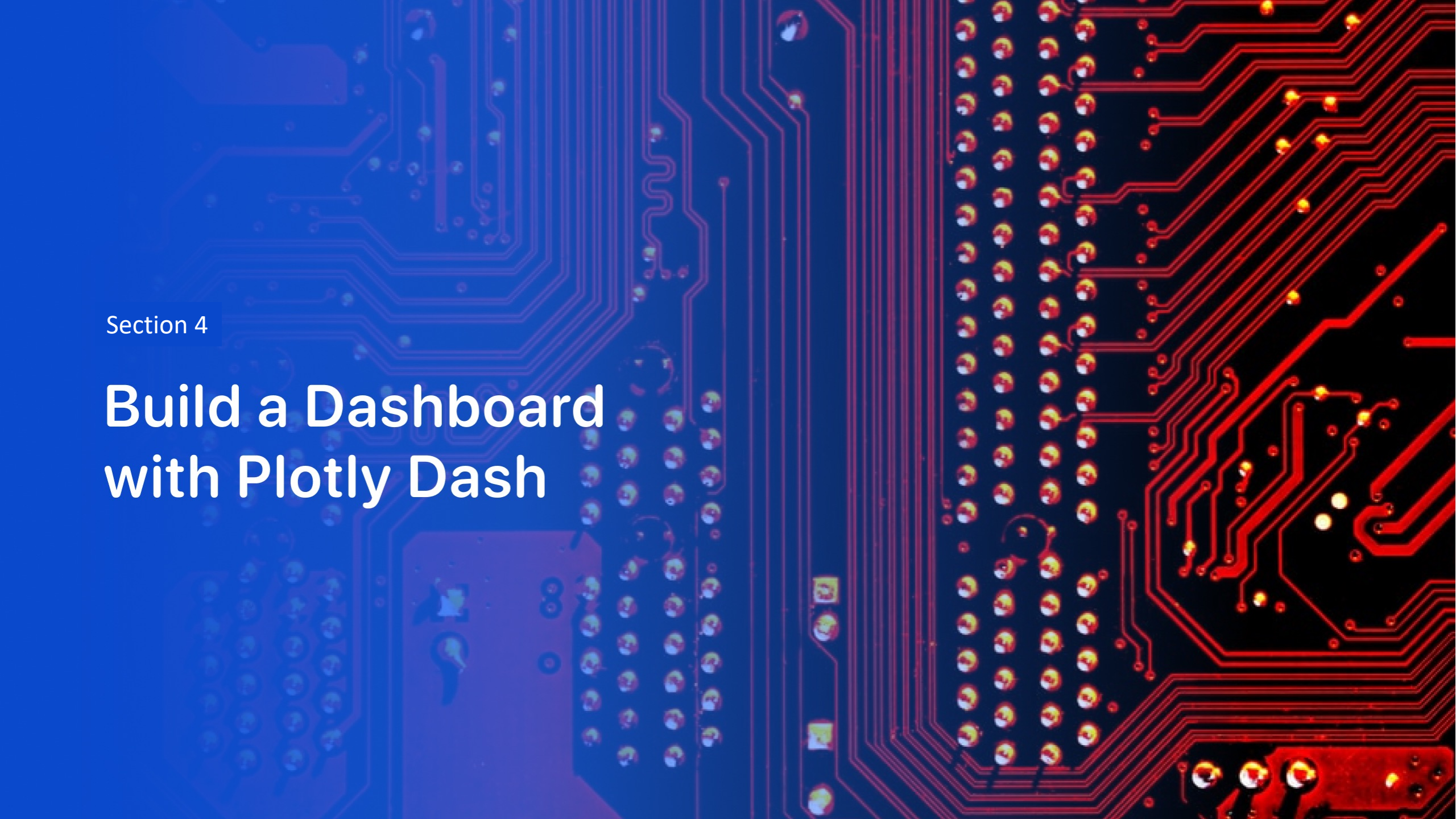
- Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate

Folium Map – Distances between CCAFS SLC-40 and its proximities





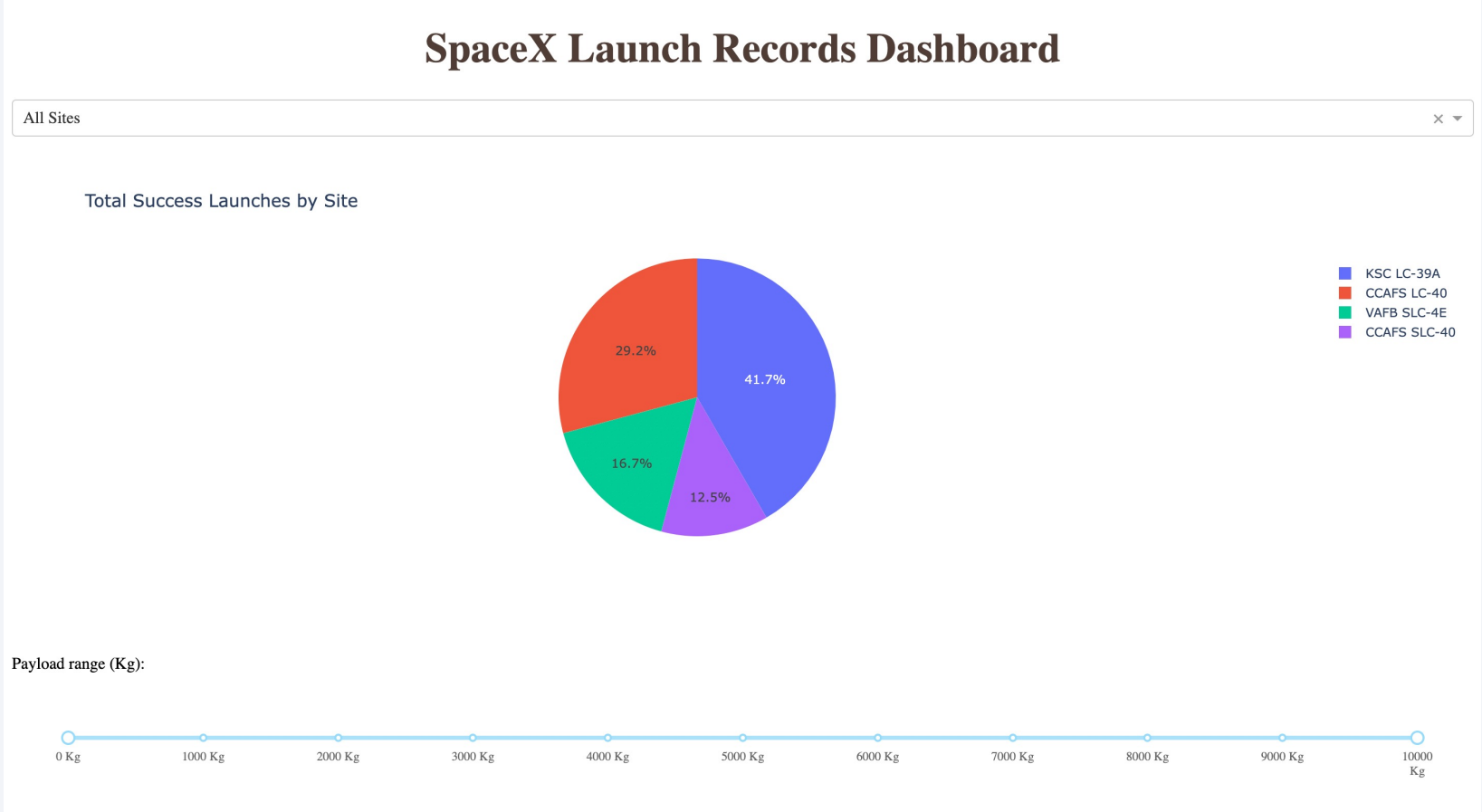
- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40 in close proximity to highways ? Yes
- Is CCAFS SLC-40 in close proximity to coastline ? Yes
- Do CCAFS SLC-40 keeps certain distance away from cities ? No



Section 4

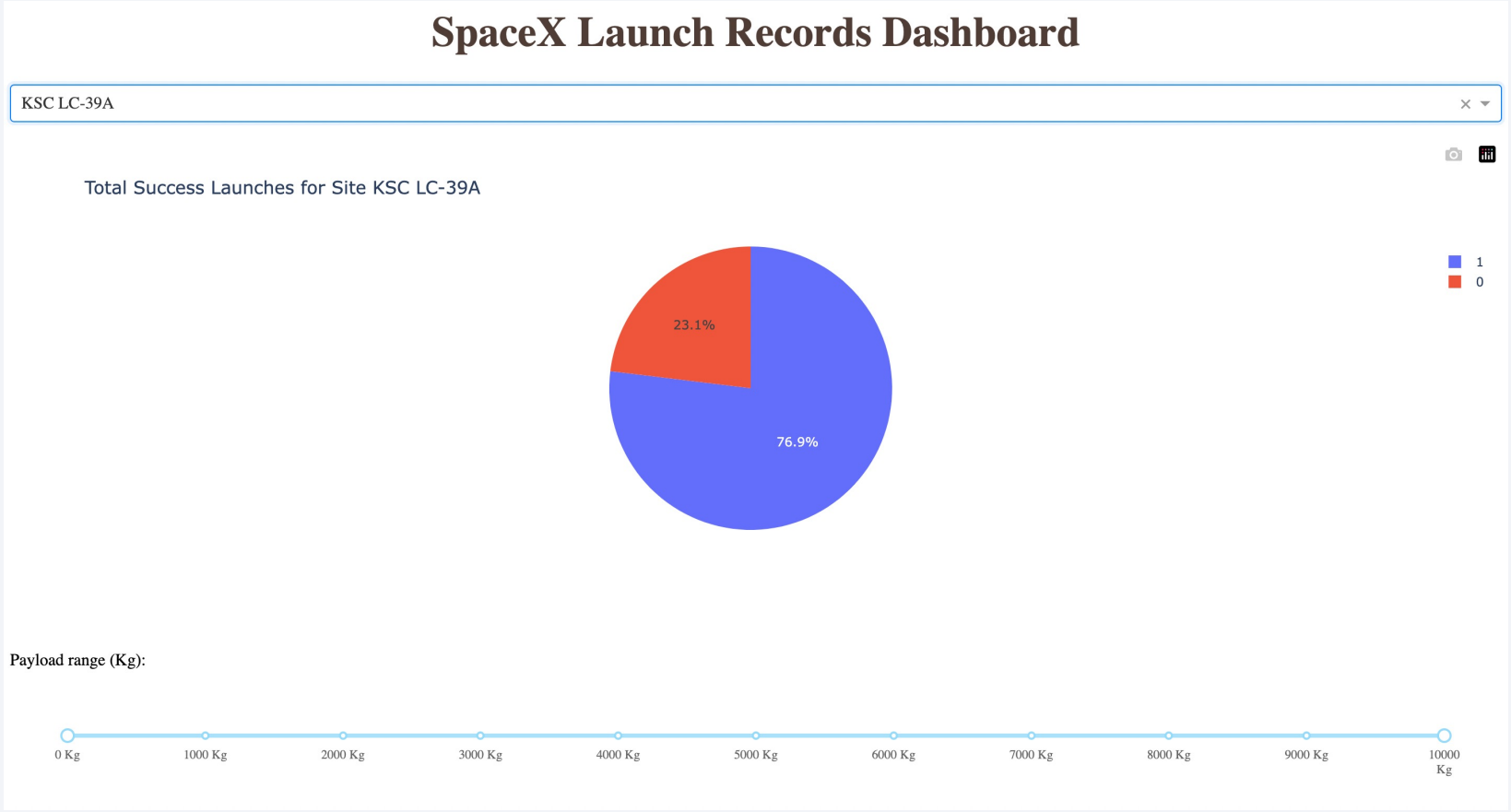
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



- We see that KSC LC-39A has the best success rate of launches

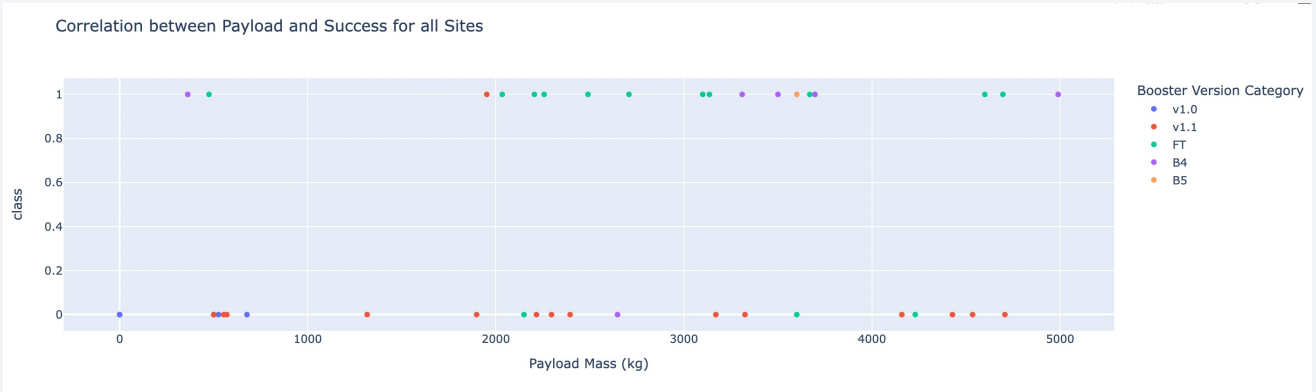
Dashboard – Total success launches for Site KSC LC-39A



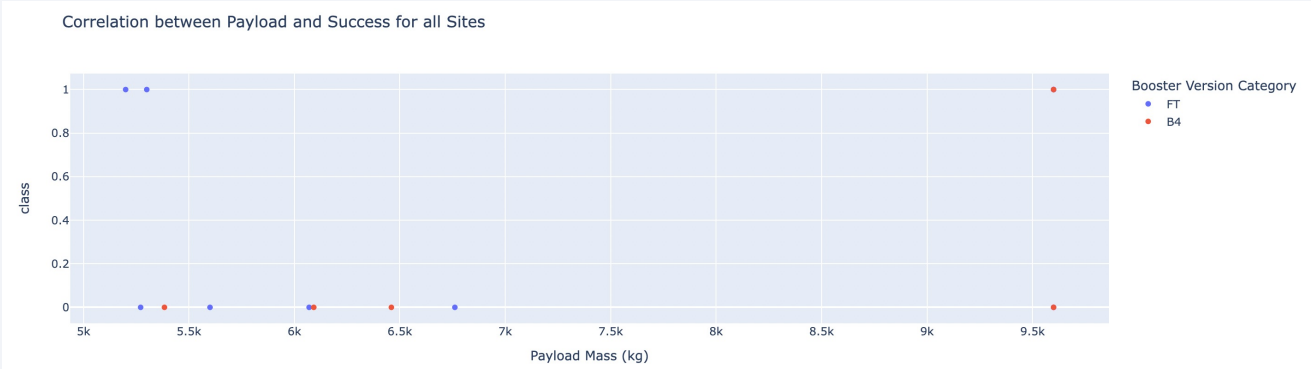
- We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected

Payload 0 - 5000 kg



Payload 5000 - 10000 kg



- Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.885714	0.833333
Knn	0.848214	0.833333

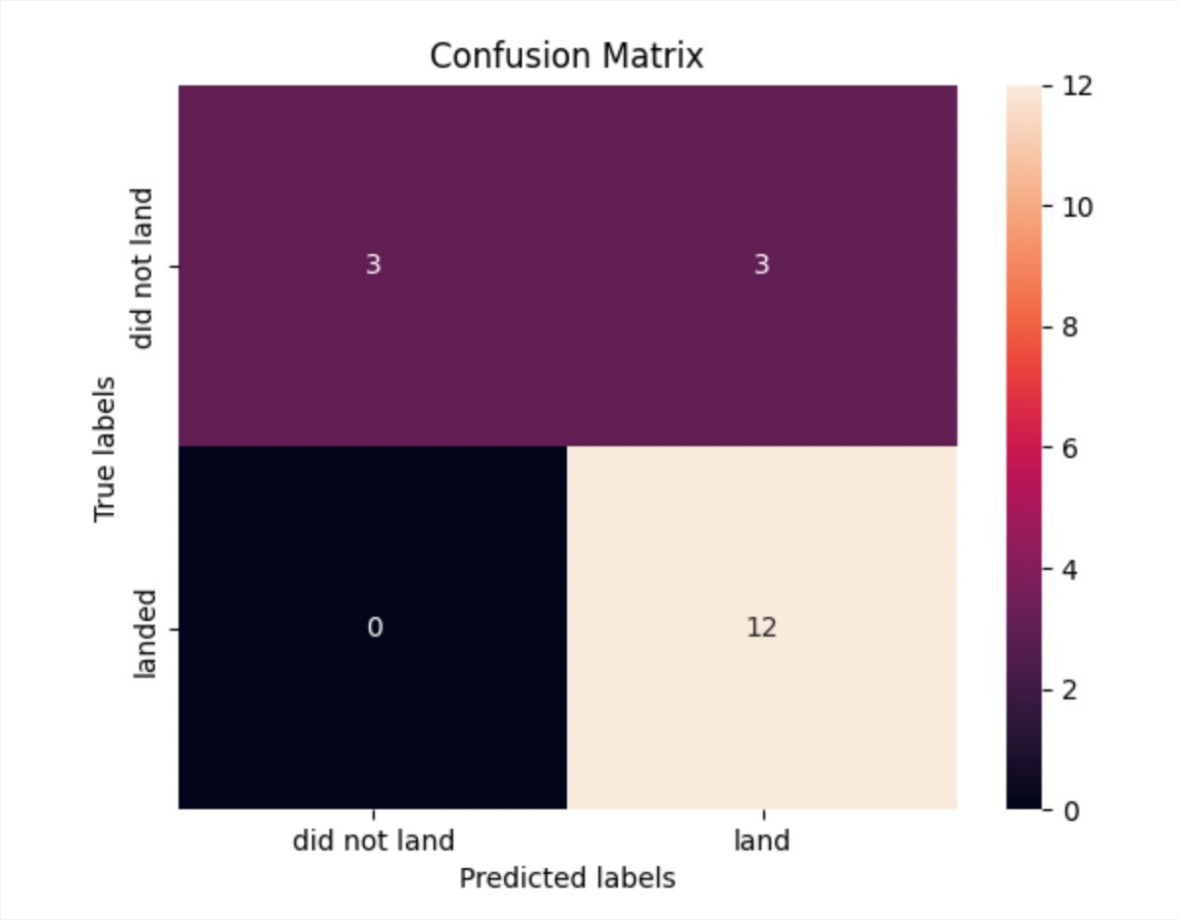
- For accuracy test, all methods performed similar. We could get more test data to decide between them.
- But if we really need to choose one right now, The decision tree is the best model

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples
_leaf': 4, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.8857142857142856
```

Confusion Matrix

- The confusion matrix of the best performing model – Decision Tree

As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives



Conclusions

- The aim of the project is to predict the successful landing of the Falcon 9 first stage.
- SpaceX's Falcon 9 rocket launch costs $62 million, while other providers charge over $165 million per launch.
- The price difference is due to SpaceX's ability to reuse the first stage.
- Factors such as the launch site, orbit, and the number of previous launches contribute to the success of a mission.
- Orbits with the best success rates include GEO, HEO, SSO, and ES-L1.
- Payload mass is an important criterion for mission success, with lighter payloads generally performing better.
- The reasons behind certain launch sites being better than others, such as KSCLC-39A, remain unexplained and may require additional atmospheric or relevant data for analysis.
- The Decision Tree Algorithm is chosen as the preferred model for this dataset, primarily due to its higher train accuracy.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!