IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alan
15 October 2024

Outline

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

Executive Summary

This presentation outlines the strategic application of data science within our organization to drive innovation, enhance decision-making, and optimize operational efficiency. We will discuss key projects, methodologies, and outcomes that demonstrate the transformative power of data analytics.

Introduction

- In this capstone, we will take the role of a data scientist working for a new rocket company.
- The job is to determine the price of each launch.
- This is done this by gathering information about Space X and creating dashboards for your team.
- We will also need to determine if SpaceX will reuse the first stage.
- Instead of using rocket science to determine if the first stage will land successfully, a machine learning model will be trained using public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

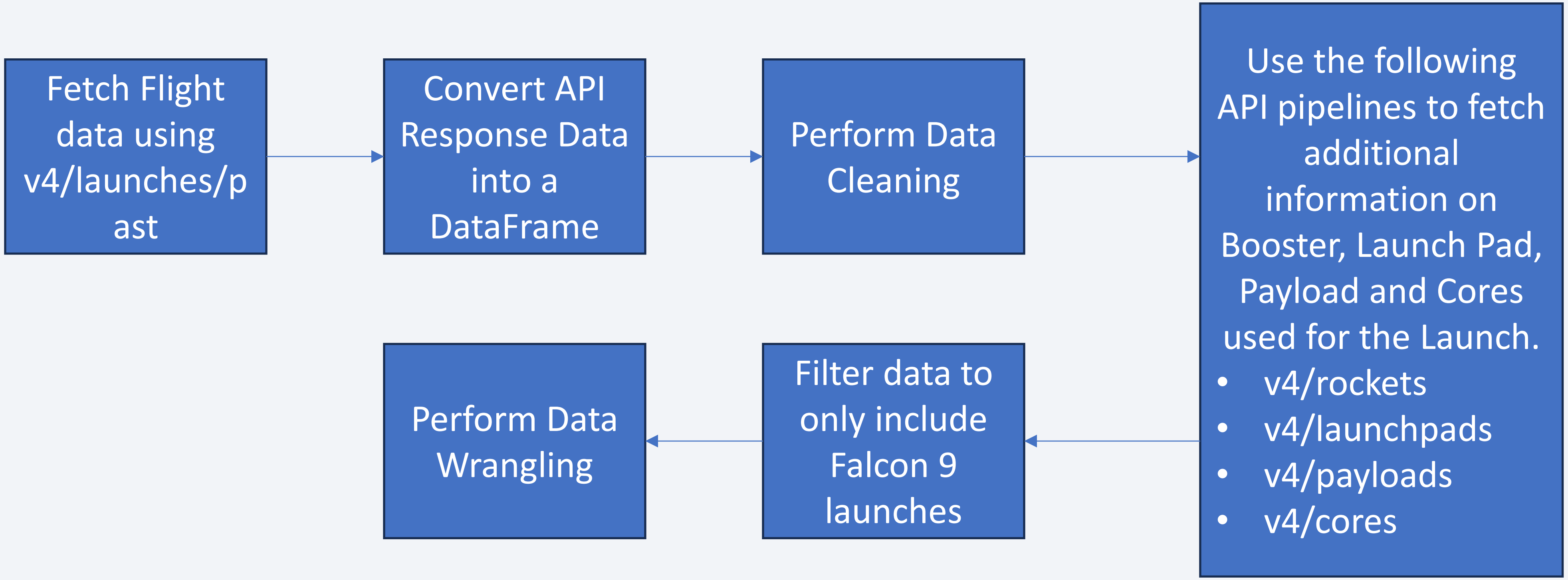
Executive Summary

- Data collection methodology:
  - Utilizing the API endpoint api.spacexdata.com/v4/launches/past
- Perform data wrangling
  - Cleaning the API Data, Sampling Relevant portions, and Dealing with Null values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

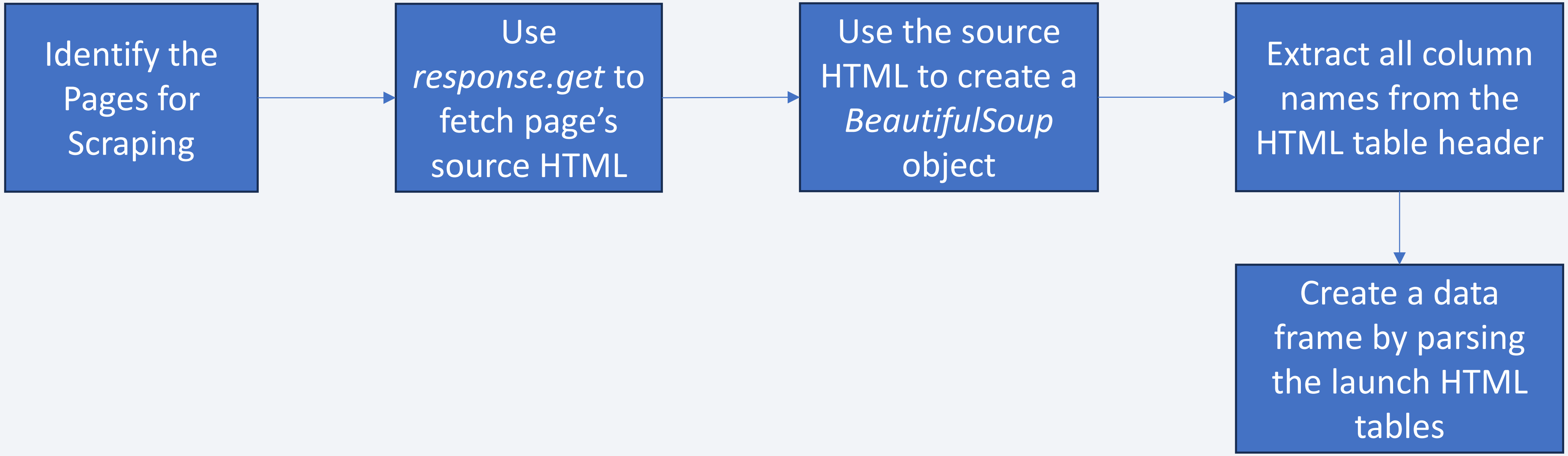
- We collect the data using two methods
  - Using API endpoint (api.spacexdata.com/v4/launches/past)
  - Web scraping related Wiki pages

Data Collection – SpaceX API



GitHub URL

Data Collection - Scraping



[GitHub URL]

Data Wrangling

Effective data wrangling is crucial for ensuring that the data is reliable and ready for analysis, ultimately leading to more accurate insights and decision-making. Some of the ways that we have achieved this is

- Filtering only Falcon 9 launch data
- Data exploration on null values, data types, value distribution of key attributes
- Create a landing outcome label from Outcome column
- Dealing with Missing Values in PayloadMass column using mean value

GitHub URL

EDA with Data Visualization

Scatter Plot

- Allows for deep analysis of value distribution over 2 selected axes

Bar Plot

- Allows for a comparative analysis on categorical data

Line Chart

- Allows for analysis of continuous data over a selected axis

GitHub URL

EDA with SQL

- 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, failure outcomes, booster versions, launch site for the months in year 2015.
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub URL

Build an Interactive Map with Folium

The Folium used in the map are as follows:

Marker – To add an interactive element to the map

Circle - To add a highlighted circle area with a text label on a specific coordinate

Popup – To add a interactive popup to the map element such as circle

MarkerCluster – Allows for the creation of marker groups

PolyLine – To add a line from one coordinate to another

GitHub URL

Build a Dashboard with Plotly Dash

- A pie chart was added to view the success/failure counts of individual launch sites and overall success counts of all sites.
- A scatter plot was added to view coorelation of payload mass with launch outcomes.

GitHub URL

Predictive Analysis (Classification)

- Using GridSearchCV, Models are built for logistic regression, support vector machine, decision tree classifier and knn.
- These models are trained using the same training set and values are predicted for the same test set.
- They are then evaluated using the model object score metric.

GitHub URL

Results

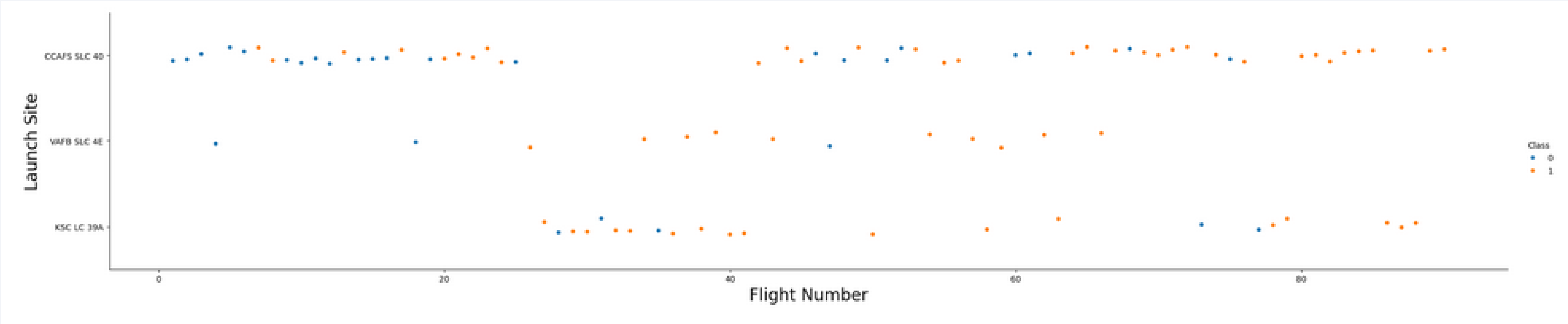
- The most relevant attributes to success outcomes were identified using data analysis.
- Interactive demo allowed for dynamic analysis of how launch site and payload mass affected launch outcome.
- Predictive analysis provided a foundational model for predicting future outcome.

Section 2

Insights drawn from EDA

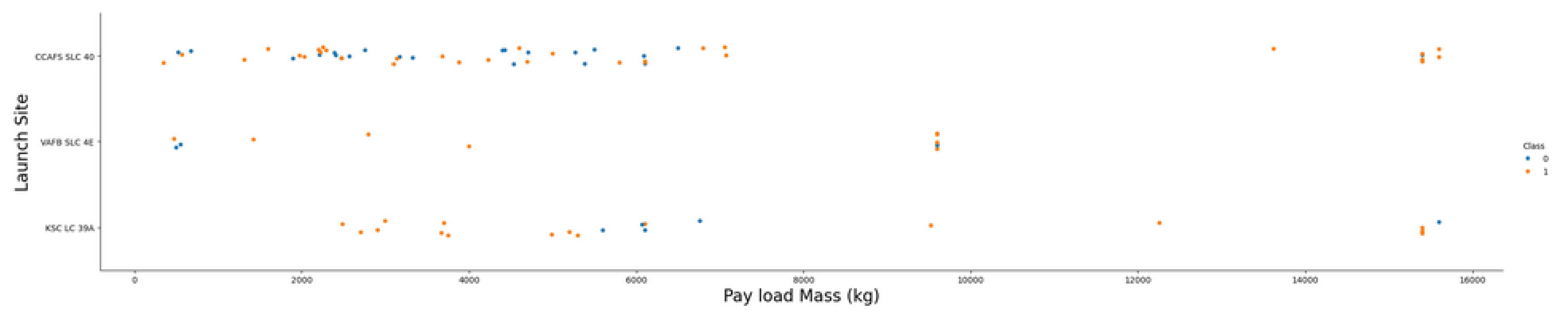
Flight Number vs. Launch Site

- The relation between flight number and launch site is established using a scatter plot.
- It is observed that initial launches were limited to one site but expanded to three.
- As number of available launch sites increased, success ratio also increased.



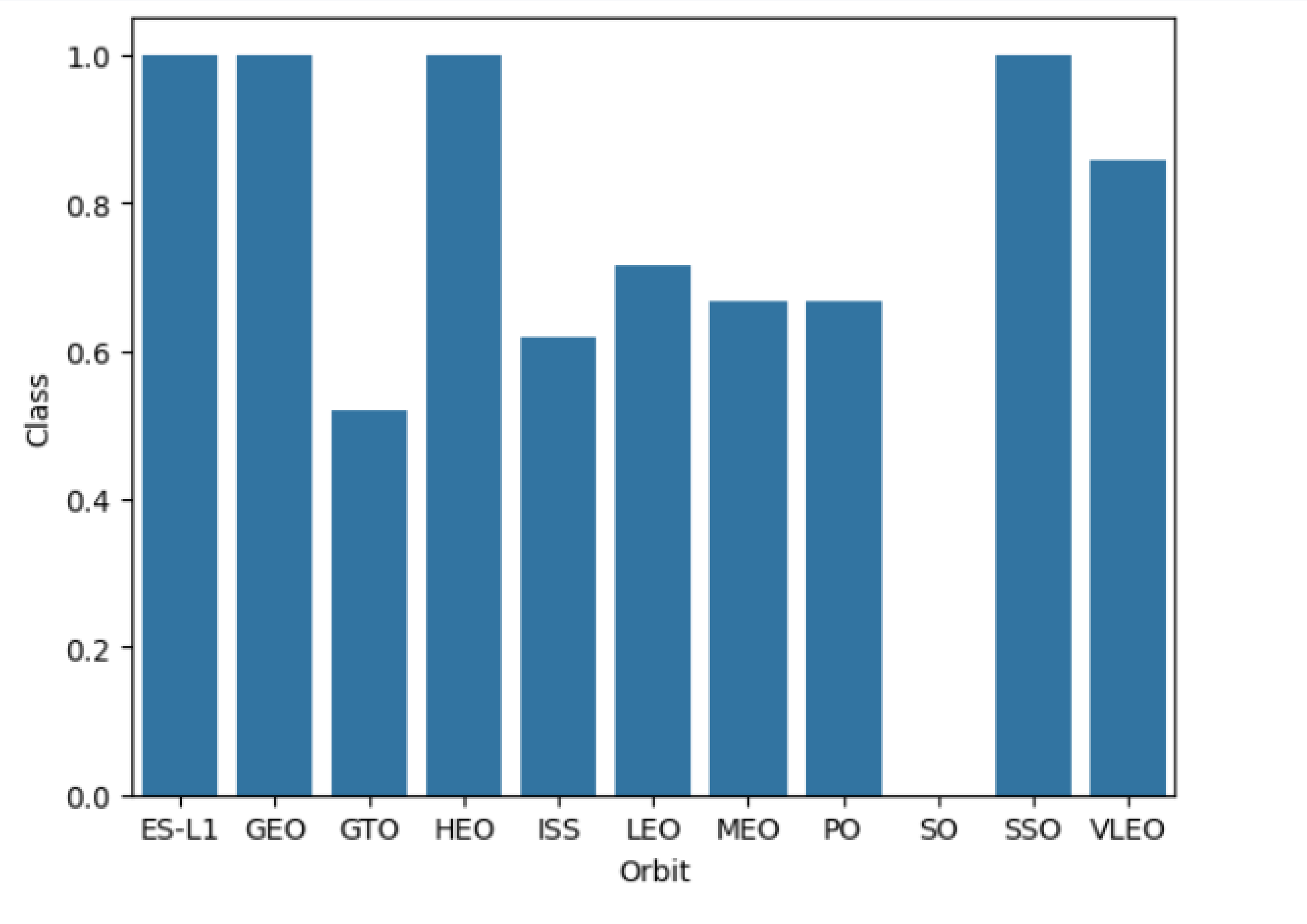
Payload vs. Launch Site

- The relation between payload and launch site is established using a scatter plot.
- For the VAFB-SLC launch site, there are no rockets launched for payload mass greater than 10000.



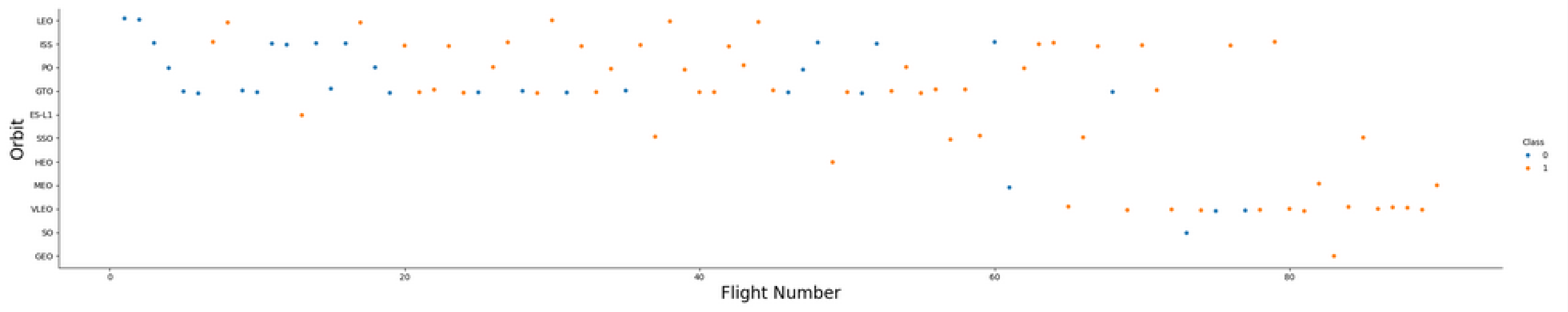
Success Rate vs. Orbit Type

- It is observed that ES-L1, GEO, HEO and SSO orbits have the most success.



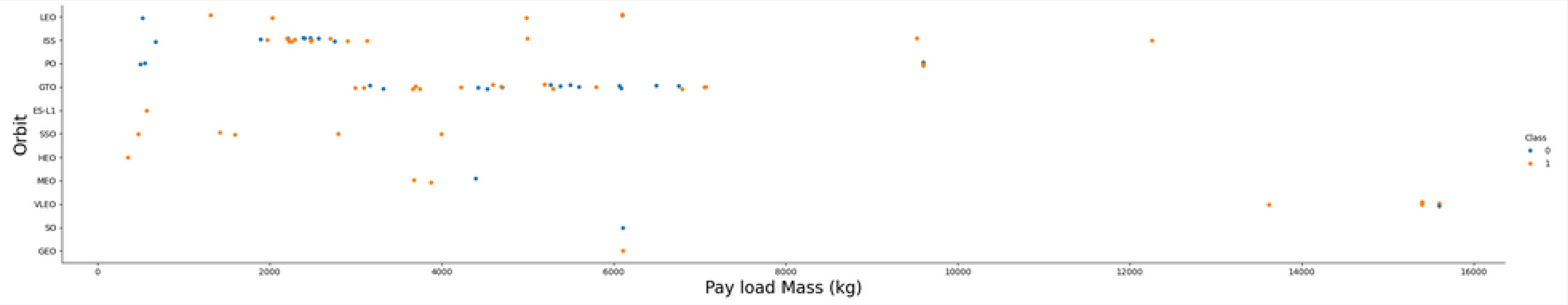
Flight Number vs. Orbit Type

- The relation between flight number and orbit type is established using a scatter plot.
- For the LEO orbit, success is observed to be related to the number of flights.
- For the GTO orbit, there appears to be no relationship between flight number and success.



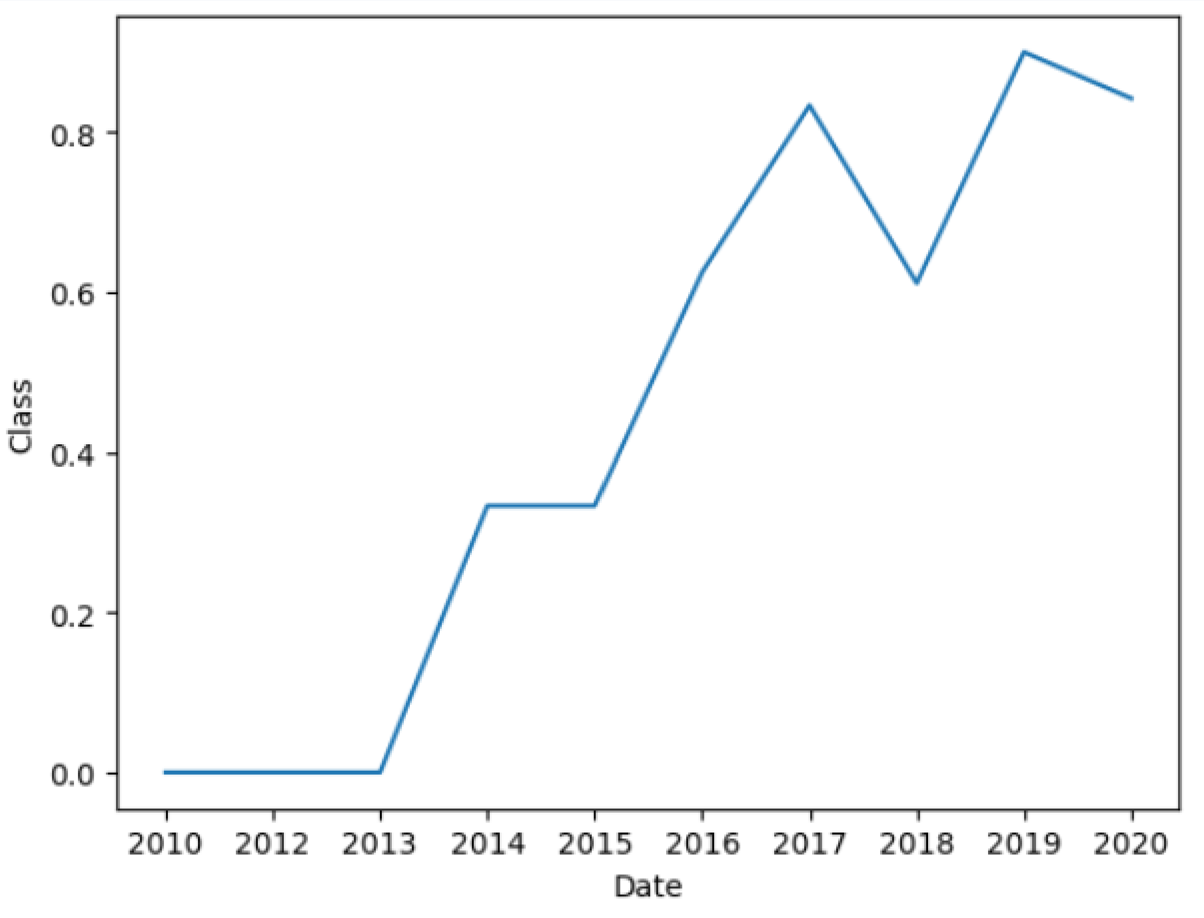
Payload vs. Orbit Type

- The relation between flight number and orbit type is established using a scatter plot.
- With heavy payloads, the successful landing rate increases for Polar, LEO and ISS.



Launch Success Yearly Trend

- It is observed that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

- The distinct names within the field are selected using the query

```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- The query will filter launch sites using the `CCA` string and select the first 5 records.

```
%sql SELECT * FROM SPACEXTABLE 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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- The query takes the sum of all payloads of launches for NASA.

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

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- The query finds the average payload mass carried for all launches using the booster version F9 v1.1

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

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

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- The query uses the function min to find the first successful landing outcome

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success"
```

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

MIN(Date)
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query finds names of the boosters which have had successful launches with a payload mass greater than 4000 but less than 6000.

```
%sql SELECT DISTINCT("Booster_Version") FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND  ("PAYLO
```

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

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

Total Number of Successful and Failure Mission Outcomes

- The query present total success and failure counts from all launches.

```
%%sql SELECT SUM(CASE WHEN "Mission_Outcome" = "Success" THEN 1 ELSE 0 END) as success_count,
    SUM(CASE WHEN "Mission_Outcome" = "Success" THEN 0 ELSE 1 END) as failure_count
    FROM SPACEXTABLE
```

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

success_count	failure_count
98	3

Boosters Carried Maximum Payload

- The query finds the names of all the boosters that have carried the maximum payload mass attempted so far.

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

```
 * 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 query lists the failed landing outcomes, their booster versions, and launch site names for all launches in the year 2015.

```
%%sql Select substr(Date, 6,2) as month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Failure (drone ship)" and substr(Date,0,5)='2015'
```

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

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

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

- The query returns a list of the possible landing outcomes and their counts between the date 2010-06-04 and 2017-03-20, ranked by the counts in descending order.

```
%%sql SELECT "Landing_Outcome", landing_count, RANK() OVER(ORDER BY landing_count DESC) as ranked_landings
    FROM (SELECT SUM(1) as landing_count, "Landing_Outcome"
    FROM SPACEXTABLE
    GROUP BY "Landing_Outcome"
    HAVING Date BETWEEN "2010-06-04" AND "2017-03-20")
```

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

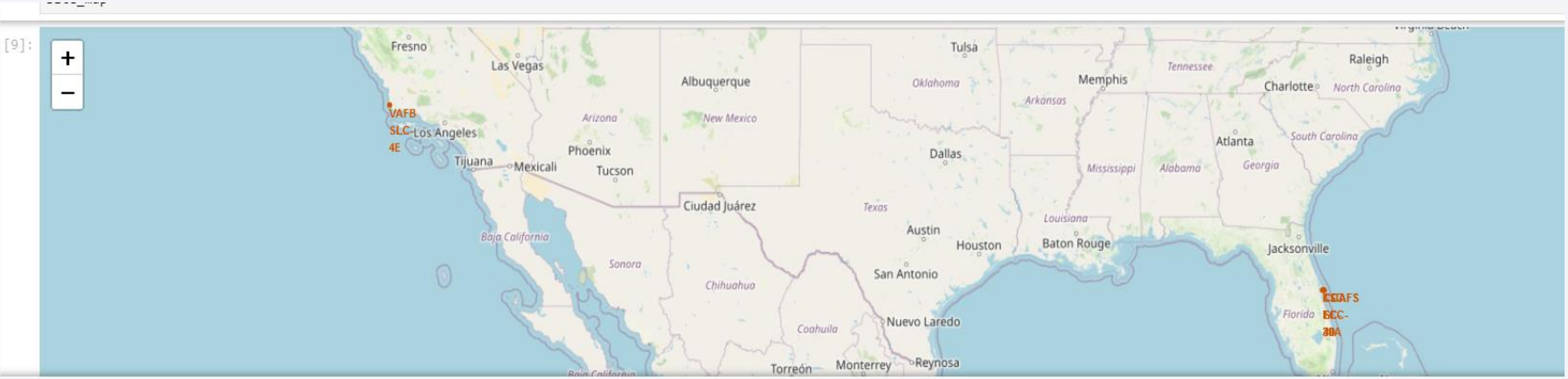
Landing_Outcome	landing_count	ranked_landings
No attempt	21	1
Success (drone ship)	14	2
Success (ground pad)	9	3
Controlled (ocean)	5	4
Failure (drone ship)	5	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

Section 3

Launch Sites Proximities Analysis

Landing Sites

- It is observed that all the launch sites are located near coastal areas

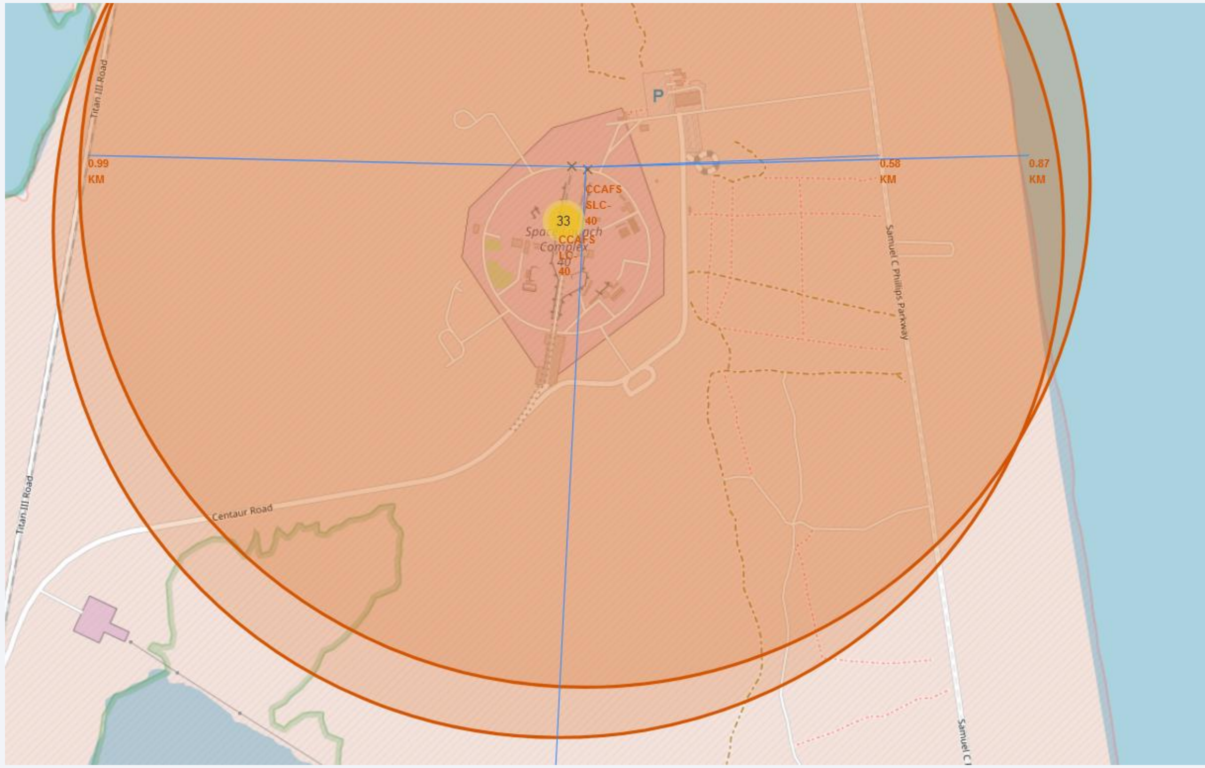


Grouped Launches

- We observed the various launches as groups of markers over the various launch sites

Distance Analysis

- It is observed that the launch site is close to railway, highway and a water body but is located far away from any major cities.

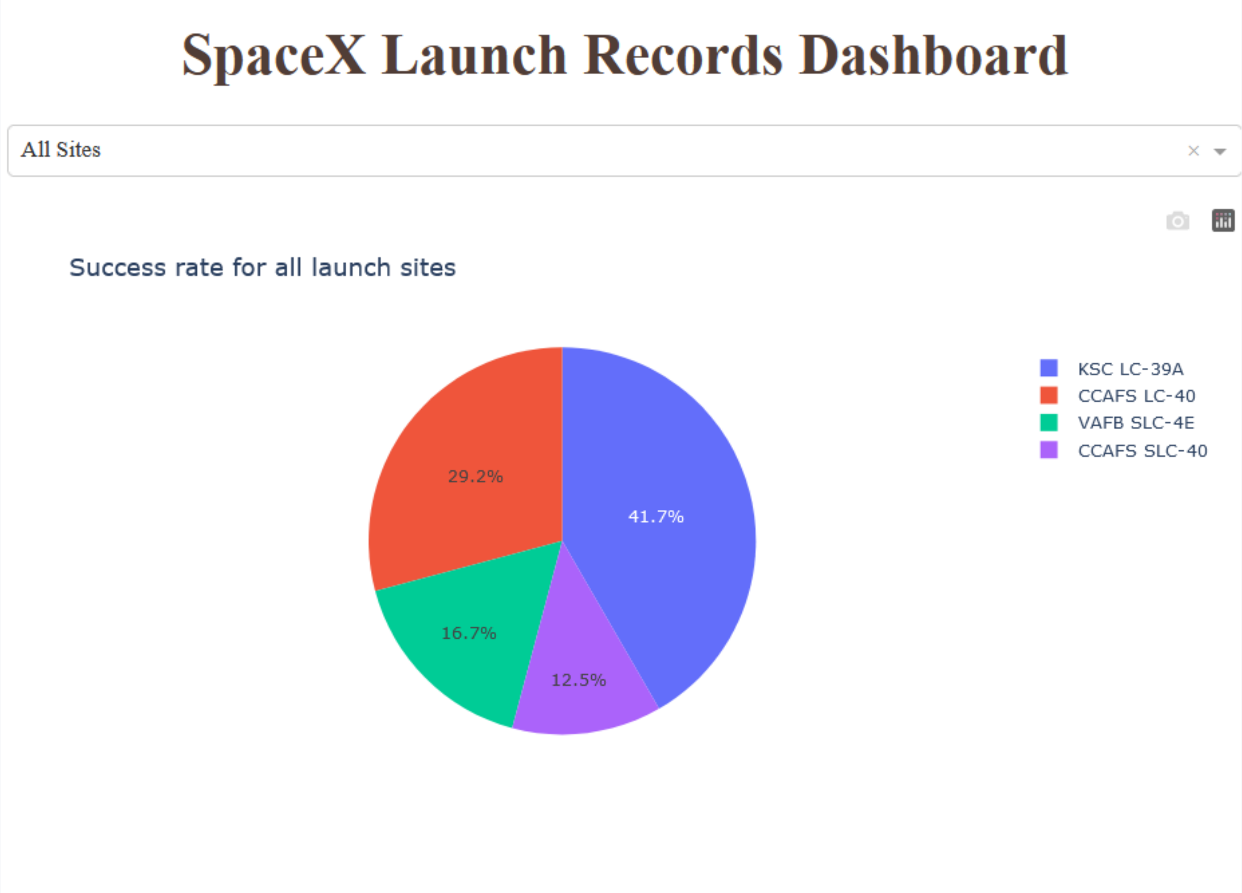


Section 4

Build a Dashboard with Plotly Dash

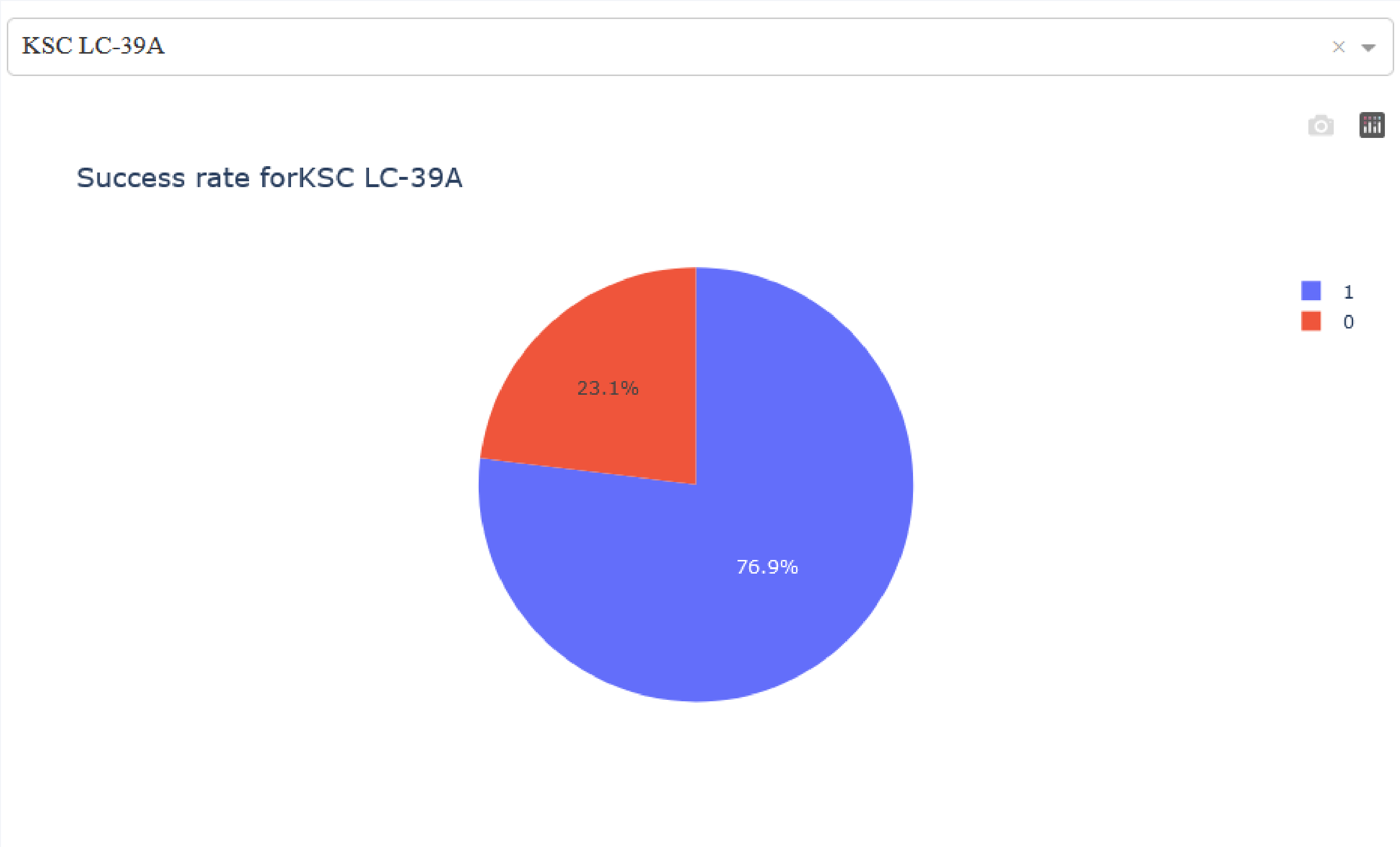
Success rates for all launch sites

- It is observed that KSC LC-39A has the highest number of successful outcomes.



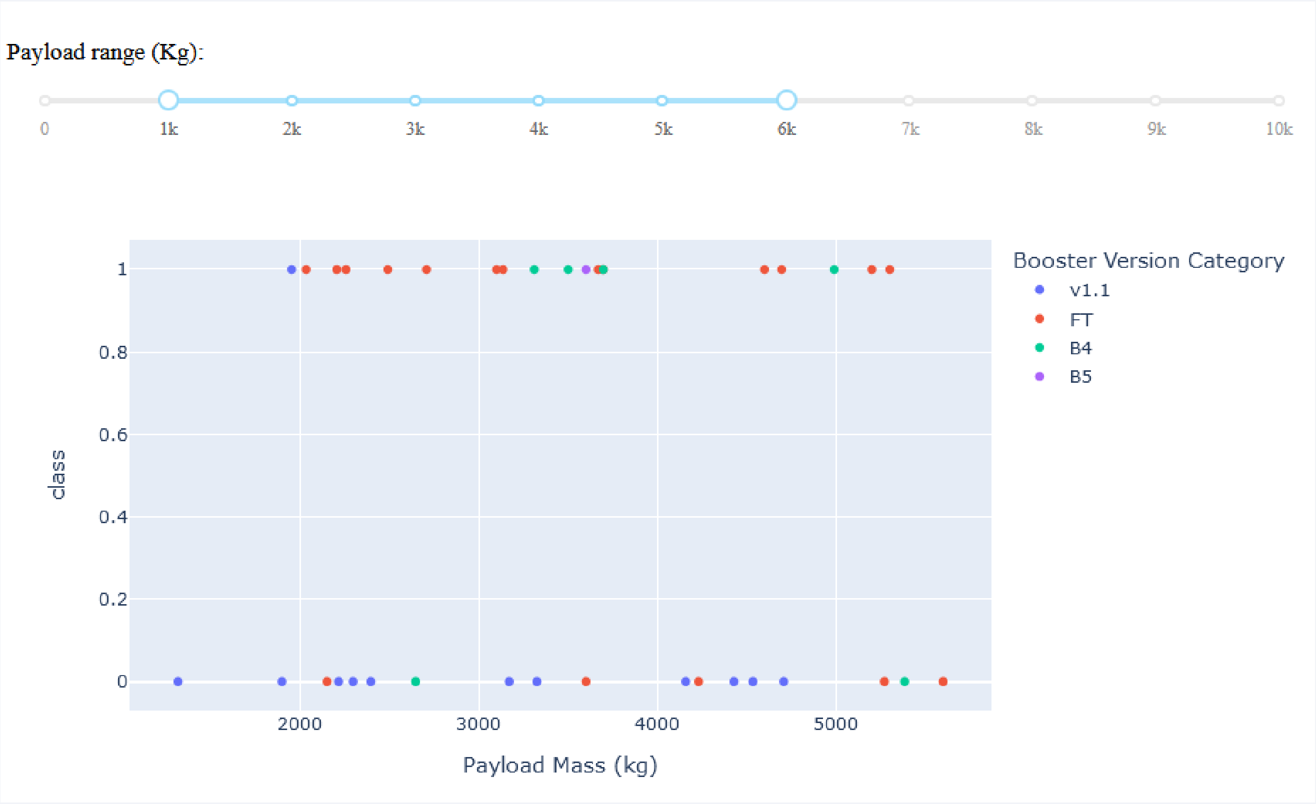
Success Analysis of KSC LC-39A

- It is observed the 77% of all launches from the site has resulted in successful outcomes.



Payload Mass's affect on Outcomes

- It is observed that some boosters are more effective for higher payloads



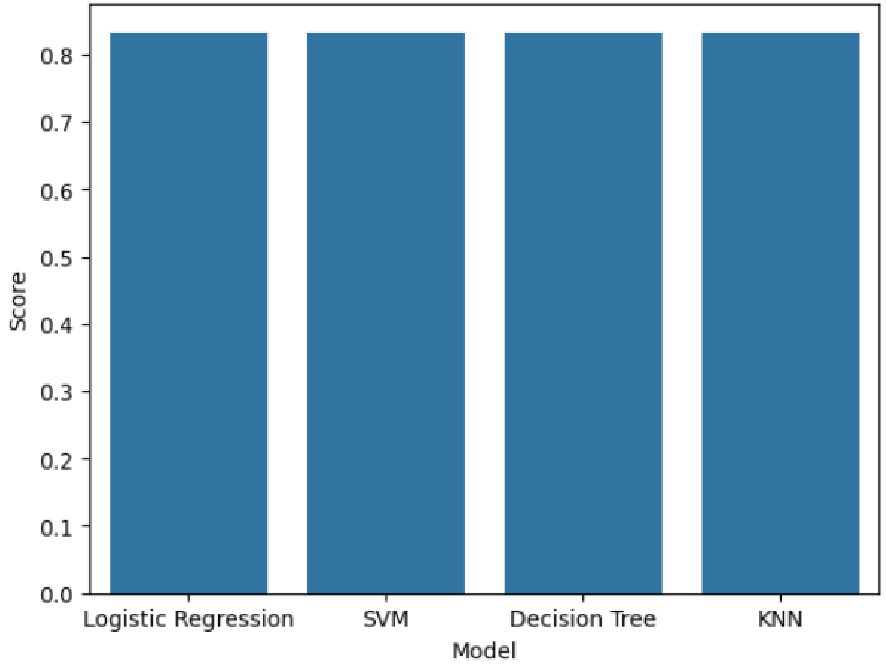
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All models are observed to have similar accuracy score.
- Additional metrics can be used to determine the best fit.
- KNN was determined to the best fit based on additional analysis.

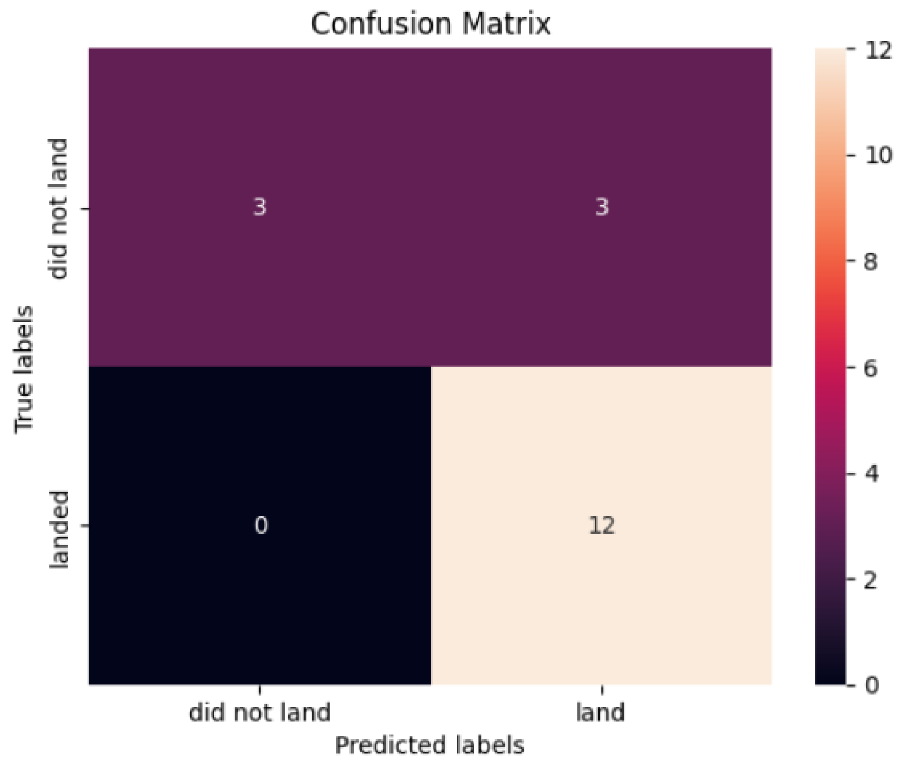
```
score_df = pd.DataFrame({'Model': ['Logistic Regression', 'SVM', 'Decision Tree', 'KNN'],
                         'Score': [logreg_score, svm_score, tree_score, knn_score]})
sns.barplot(data=score_df, x=score_df['Model'], y=score_df['Score'])
plt.show()
```



Confusion Matrix

- The KNN model accurately identified all cases of successful landing as success.
- However it only had a 50% success rate for identifying failure outcomes.

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

By analysis of the SpaceX data, we have observed the following:

- Having a booster specialize in a payload range is the best approach to its design.
- Having multiple launch site options allow for more varied testing and robust design.
- Identifying the Orbit type goal early for a booster can help boost its effectiveness.

Appendix

- All the relevant assets such as Python code snippets, SQL queries, charts, Notebook outputs that have been created during this project can be viewed in https://github.com/alankoshy/ds_capstone/tree/main

Thank you!