IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Abhineeth Anoop>
<7th May 2024>

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

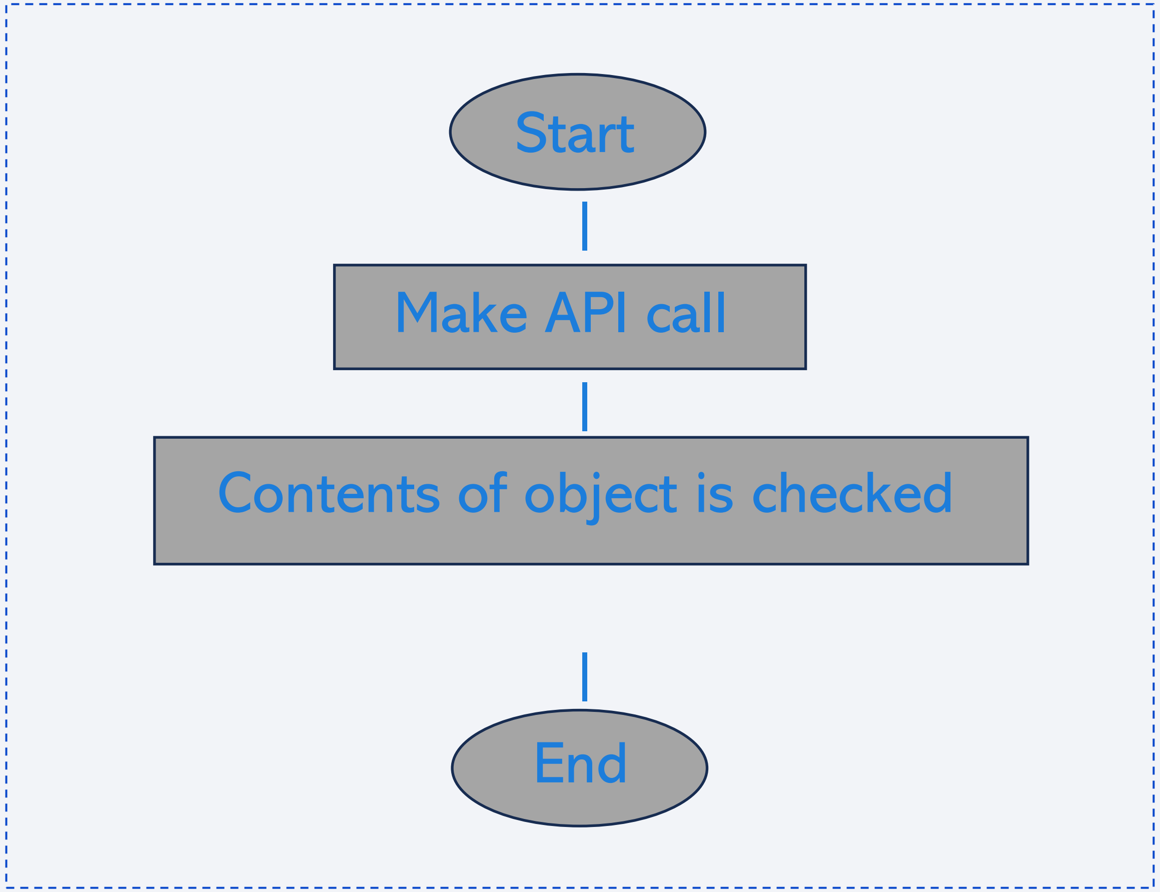
Methodology

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

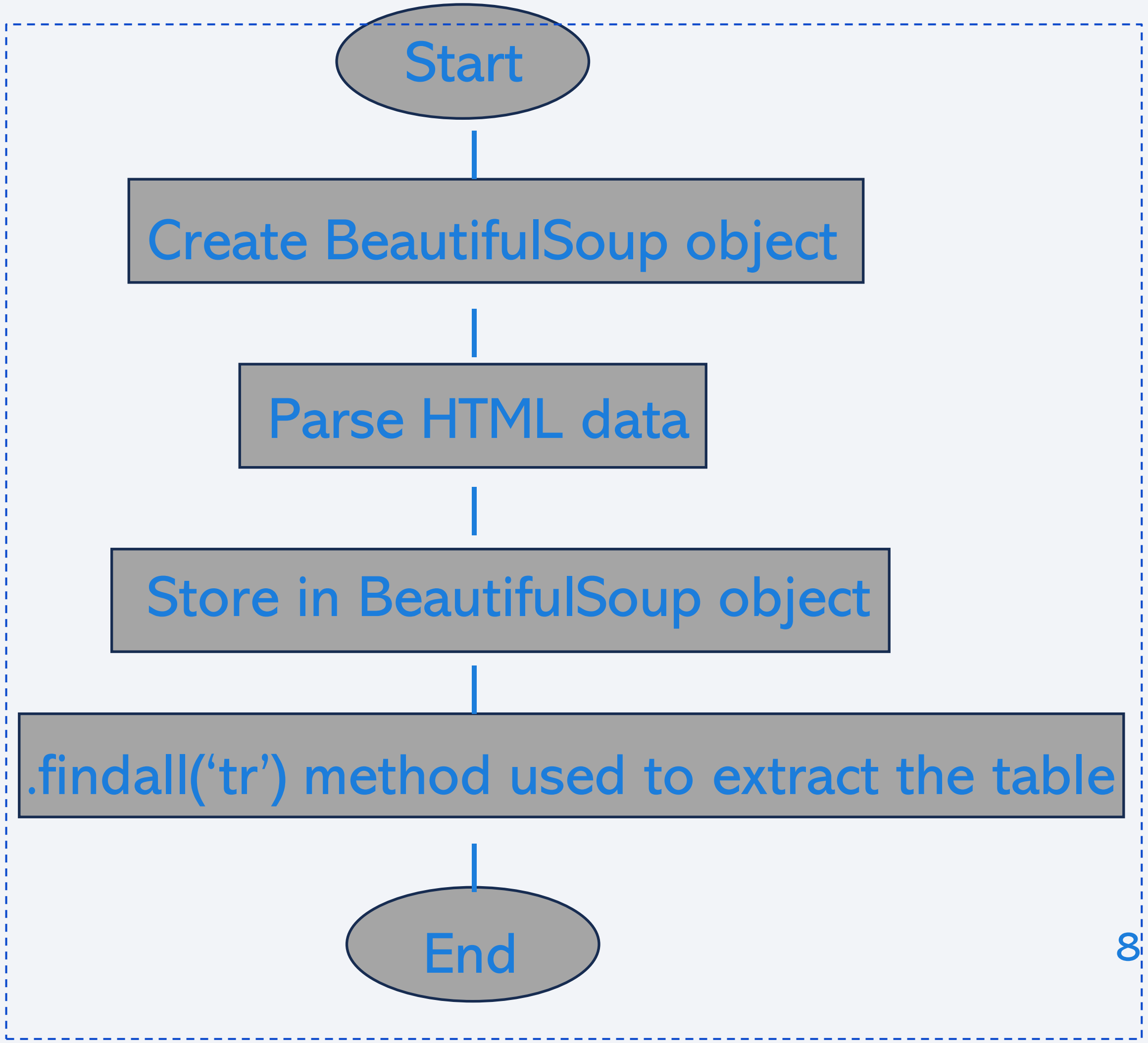
Data Collection – SpaceX API

- The dataset of SpaceX launches were extracted using an API - https://api.spacexdata.com/v4/launches/past
- The requests object is used to make the API call. The requests.get() method is used here and the required URL is given to the method and the response object is checked by using the '.content' method
- GitHub URL- https://github.com/abhineeth817/CourseraAssignment/blob/2c4582a566c6a1b428e1e4848a48bdeb78355a63/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- After a static website is extracted using the response object, the content parsed using BeautifulSoup html parser and is converted to a BeautifulSoup object.
- From the BeautifulSoup object, the ‘.findall(“tr”)’ method is used to extract the required table.
- Then the column names of the table are extracted using ‘for’ loop and ‘.findall(‘th’)’ method



Start

Create BeautifulSoup object

Parse HTML data

Store in BeautifulSoup object

.findall(‘tr’) method used to extract the table

End

Data Collection - Scraping

- A dataframe is created from a dictionary using these column names and the relevant data is loaded into the dictionary, which is then converted into a pandas DataFrame
- GitHub URL - https://github.com/abhineeth817/CourseraAssignment/blob/2c4582a566c6a1b428e1e4848a48bdeb78355a63/jupyter-labs-webscraping.ipynb

Data Wrangling

- First, the amount of missing values in the dataframe is checked and the missing values is replaced with their mean or the common value.
- The number of Launches in each site and the number and occurrence of each Orbit type is calculated using the '.value_counts()' method.
- Then to create a new column the number and occurrence of mission outcomes is also calculated.
- A new column 'Class' is created from this data, in which the value 0 refers to an unsuccessful outcome and the value 1 refers to a successful outcome

EDA with Data Visualization

- A cat plot in the seaborn library is used to find the relationship the flight number ,payload mass and the mission outcome.
- Scatter plots are plotted against flight number, launch site and color representing the mission outcome, to find the relationship of success rate of the mission and the launch site. It is also used to plot the relationship between payload mass, launch site, mission outcome and the relationships of many other variables.
- The relationship between orbit type and success rate is also visualized using scatter plot and bar graphs.
- A line graph is plotted to find the relationship between the launch year and its success rate
- GitHub URL - https://github.com/abhineeth817/CourseraAssignment/blob/2c4582a566c6a1b428e1e4848a48bdeb78355a63/edadataviz.ipynb

EDA with SQL

- The names of unique launch sites are extracted using the DISTINCT query
- Names of 5 launch sites whose names begin with ‘CCA’ is extracted using the LIKE query
- The total payload mass carried by NASA (CRS) is calculated using the SUM query
- The average payload carried by specific booster version is calculated using AVG query
- The date at which the first successful landing outcome is found using MIN query
- The booster versions which have successful outcomes and payload mass between two specific values is found using the BETWEEN query

EDA with SQL

- The total number of successful and failed mission outcomes is calculated using a subquery
- The booster versions which have carried the maximum payload mass is found using a subquery
- The records for failed landing in the year 2015 is listed
- The landing outcomes between two specific dates is ranked according to their count
- GitHub URL- https://github.com/abhineeth817/CourseraAssignment/blob/55aa8ca409d1e8f03e16bc84c5de87c2fc9cd270/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A circle marker is used to mark the launch sites in the map to find the approximate proximity of the launch sites and to approximate their distance to the nearest coast and the equator
- A marker cluster object is used to mark the sites with their color representing the success of the mission
- The distance between the closest site and the coast and the railway is measured and marked on the map with a Polyline representing the distance.
- GitHub URL - https://github.com/abhineeth817/CourseraAssignment/blob/55aa8ca409d1e8f03e16bc84c5de87c2fc9cd270/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

- A pie chart is used to visualize the ratio is the total successful launches to their launch site which can be used to find the percent of successful launches and in specific sites
- A scatter plot between the payload mass and success of mission is plotted to find how the payload mass affects the outcome of a mission
- GitHub URL- https://github.com/abhineeth817/CourseraAssignment/blob/45f9f16bdef1c3ac9b5dcfaaa1917700c60c4680/spacex_dash_app.py

Predictive Analysis (Classification)

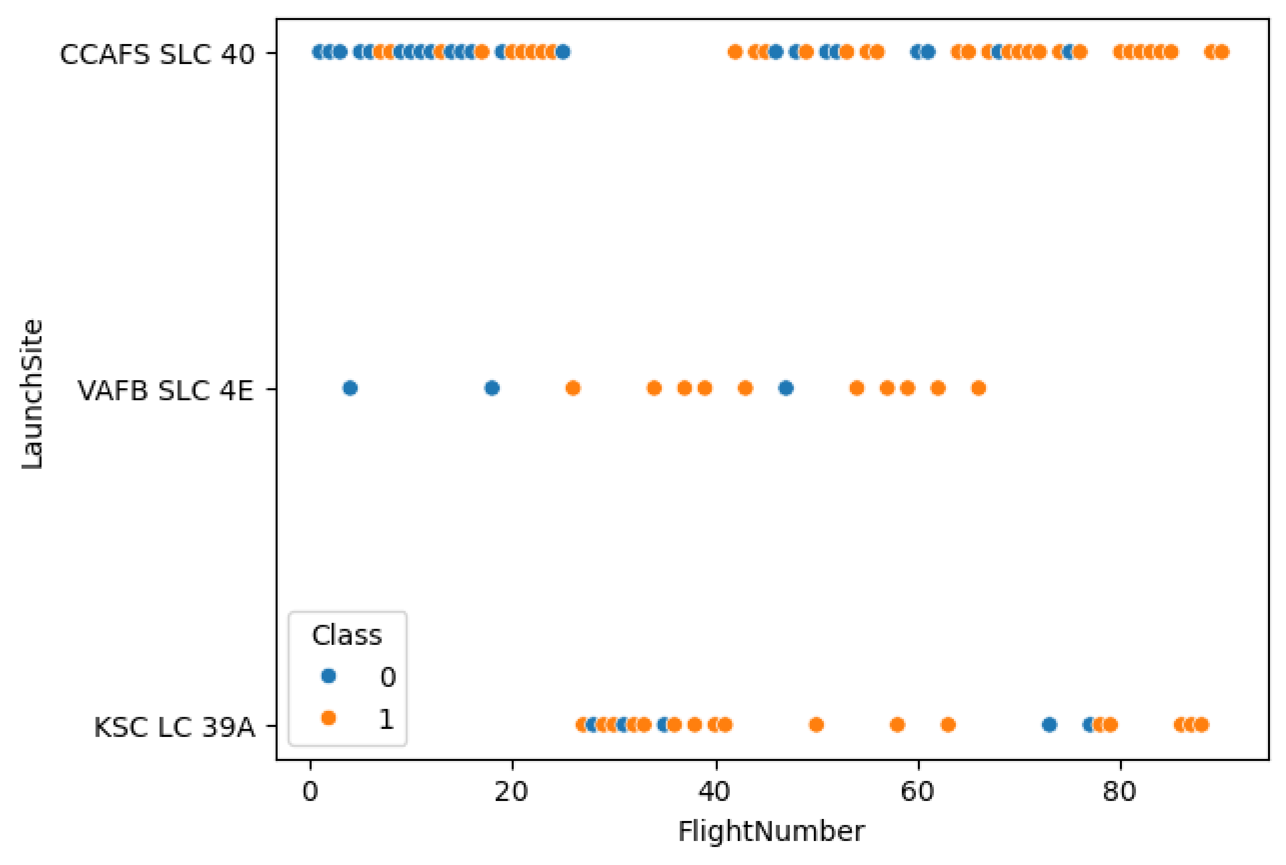
- The outcome column in the dataframe is first converted to a numpy array and the rest of the df is standardized using the StandardScaler object. The df is then split into train and test sets using the train_test_split() function.
- Then the GridSeachCV object is used with various models like LogisticRegression, SVC, DecisionTreeClassifier and KNeighboursClassifier and the results of each variation is tested to find the one with the least error
- The error of each variation is tested using the '.score()' method and by plotting a confusion matrix
- Add the GitHub URL- https://github.com/abhineeth817/CourseraAssignment/blob/45f9f16bdef1c3ac9b5dcfaaa1917700c60c4680/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Section 2

Insights drawn from EDA

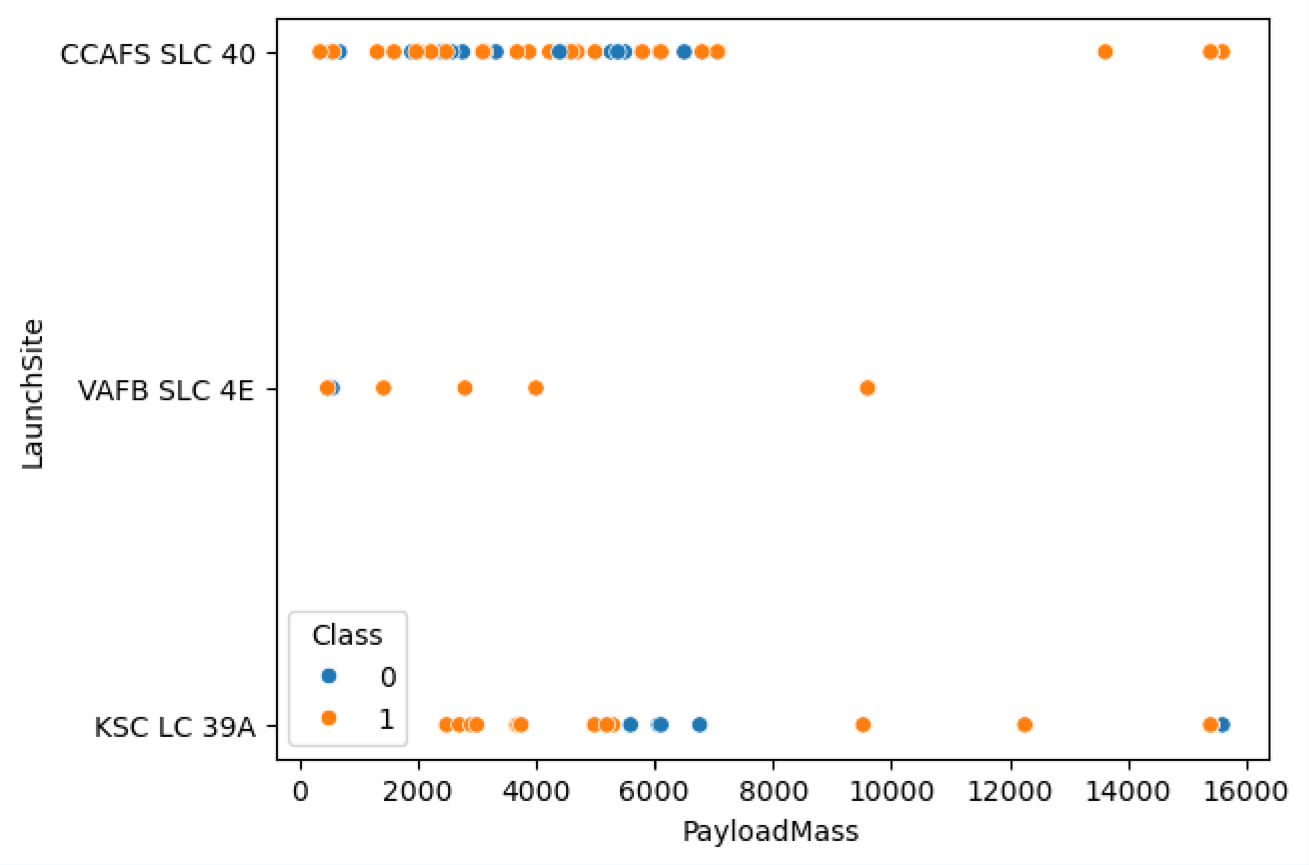
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site
- It can be inferred that the least launches where from the site VAFB SLC 4E. And that site KSC LC-39A has relatively highest number of launches and success rate



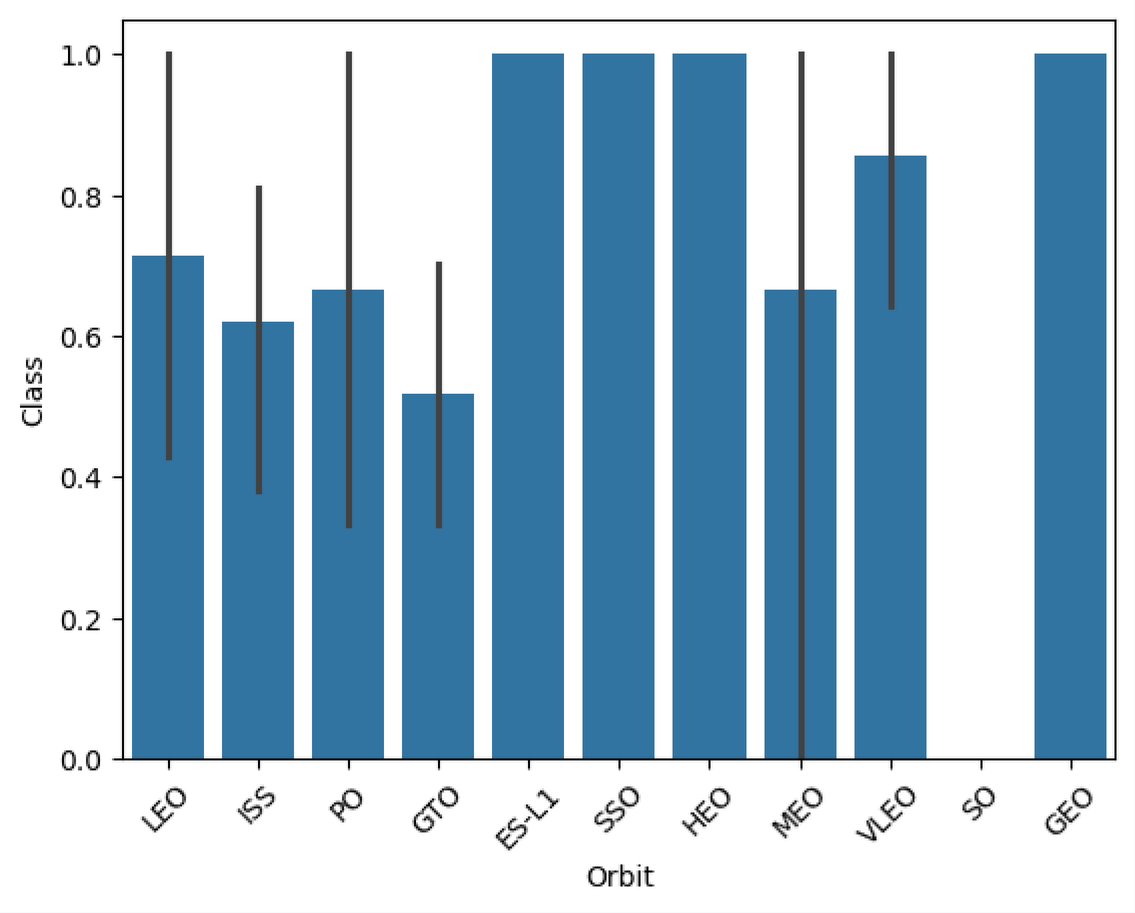
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site
- It can be inferred that the number of launches with payload mass more than 8000 which failed is only one. And for a relatively high number of launches in site CCAFS LC-40, almost all the launches have a payload below 8000



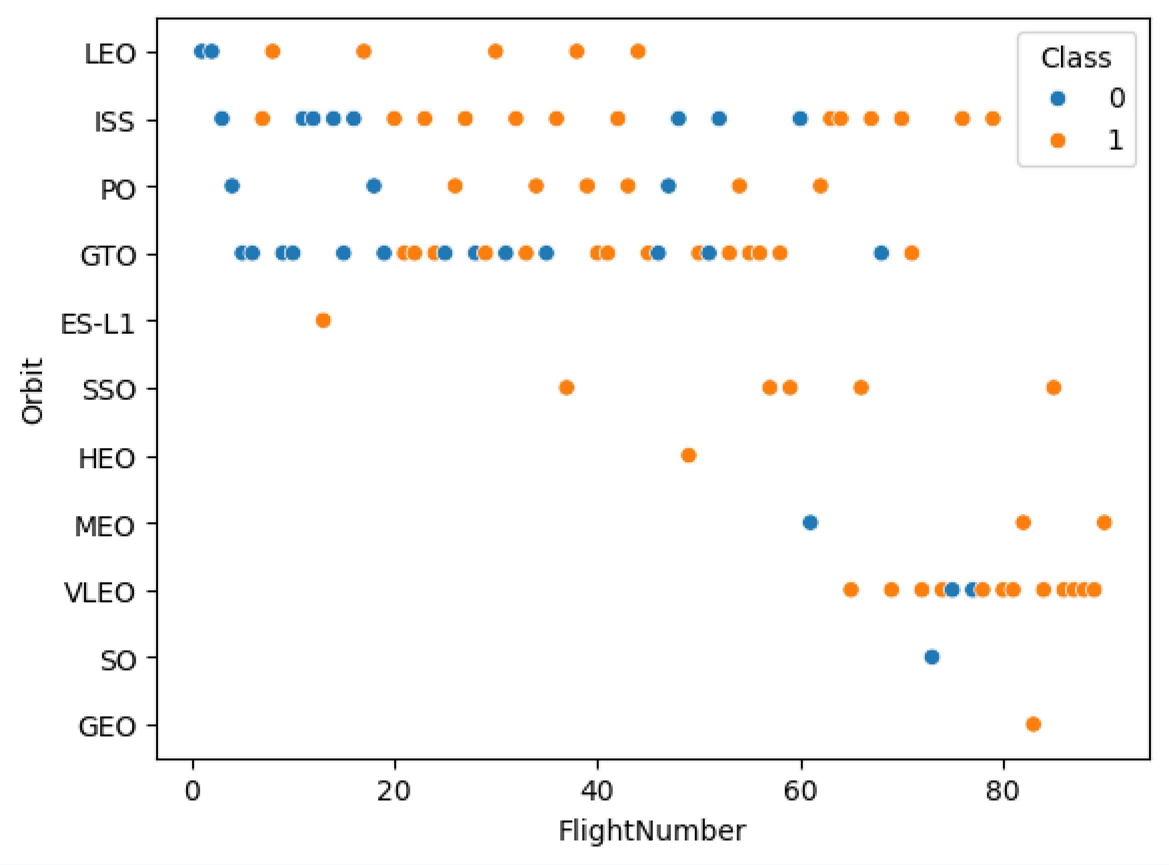
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- It can be inferred that the lowest success rate is for orbit GTO and 4 orbits have a success rate of 100%, they are ES-L1,SSO, HEO and GEO.



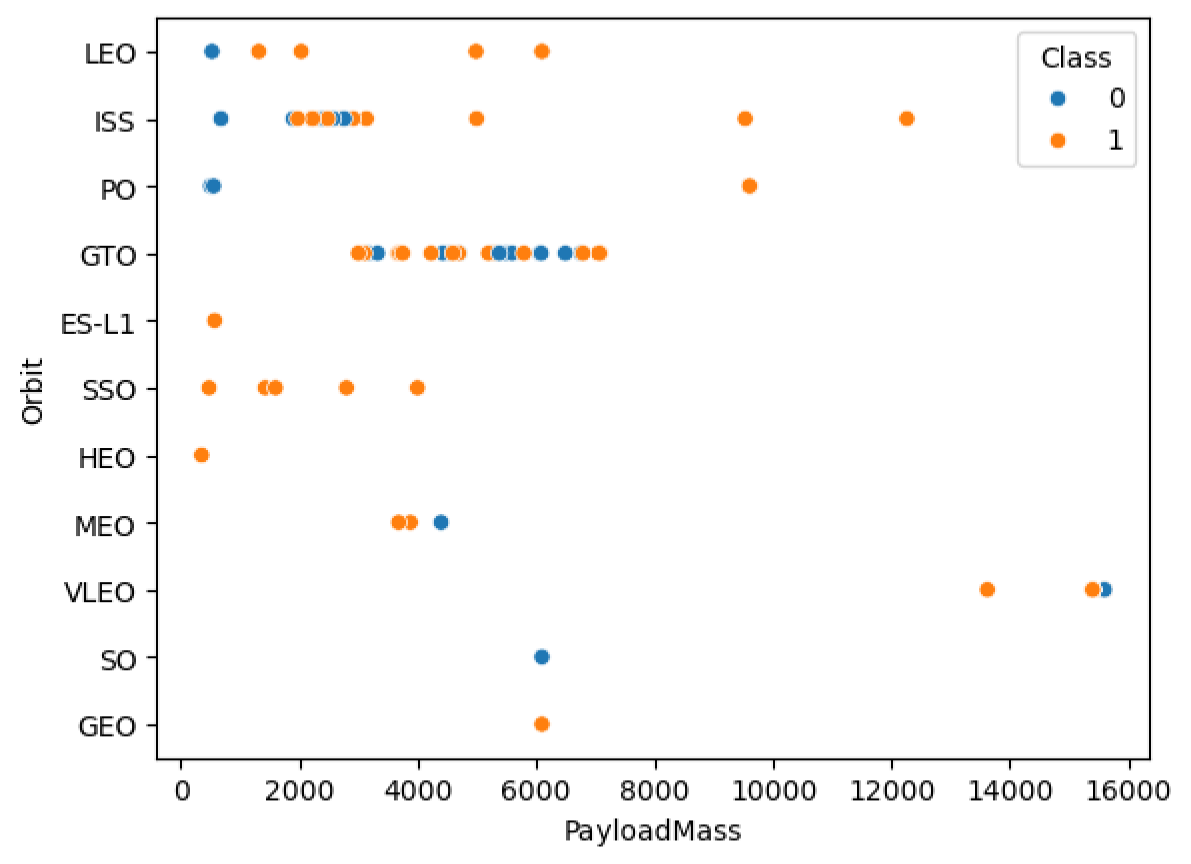
Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type
- It can be inferred that the orbits which have a success rate of 100% is because of their low number of launches and hence is unreliable data for future prediction.



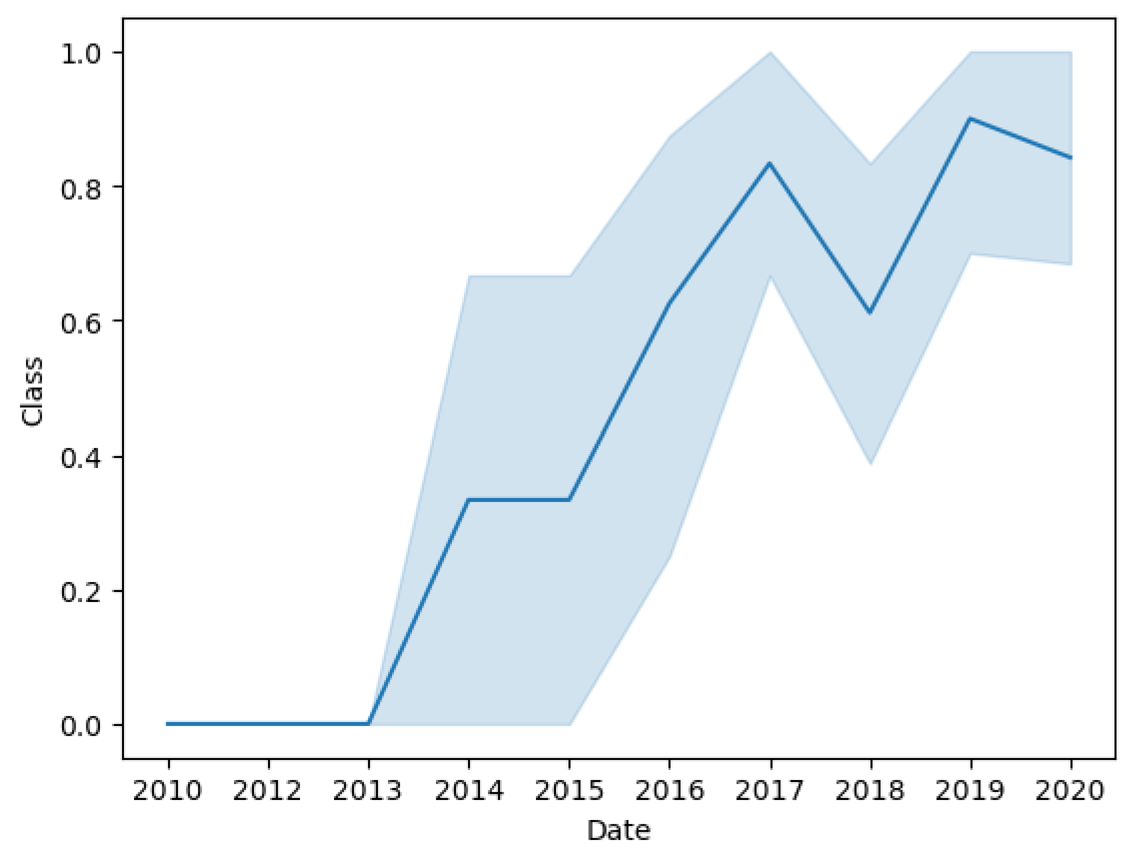
Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type
- It can be inferred that for high payload mass the success rate is high for PO, LEO and ISS



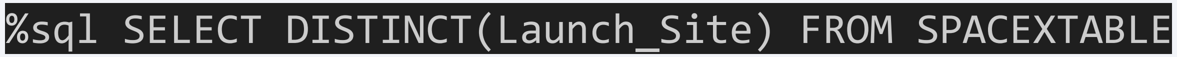
Launch Success Yearly Trend

- Line chart of yearly average success rate
- The success rate has seen a steady increase since 2013 with the peak being at 2019



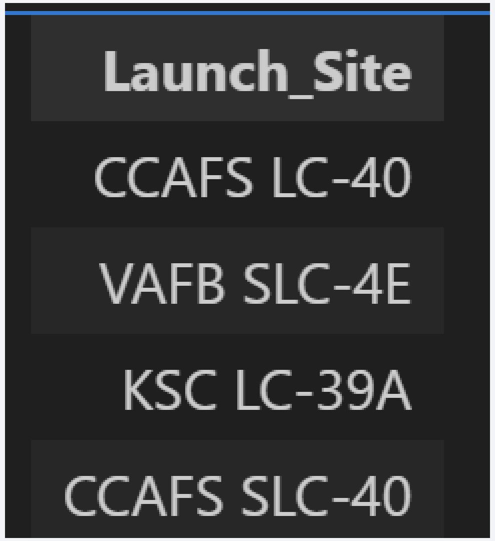
All Launch Site Names

- Find the names of the unique launch sites



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

Result

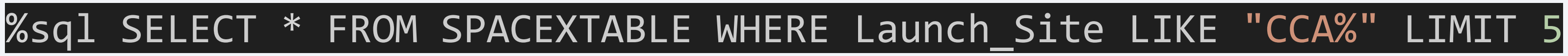


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

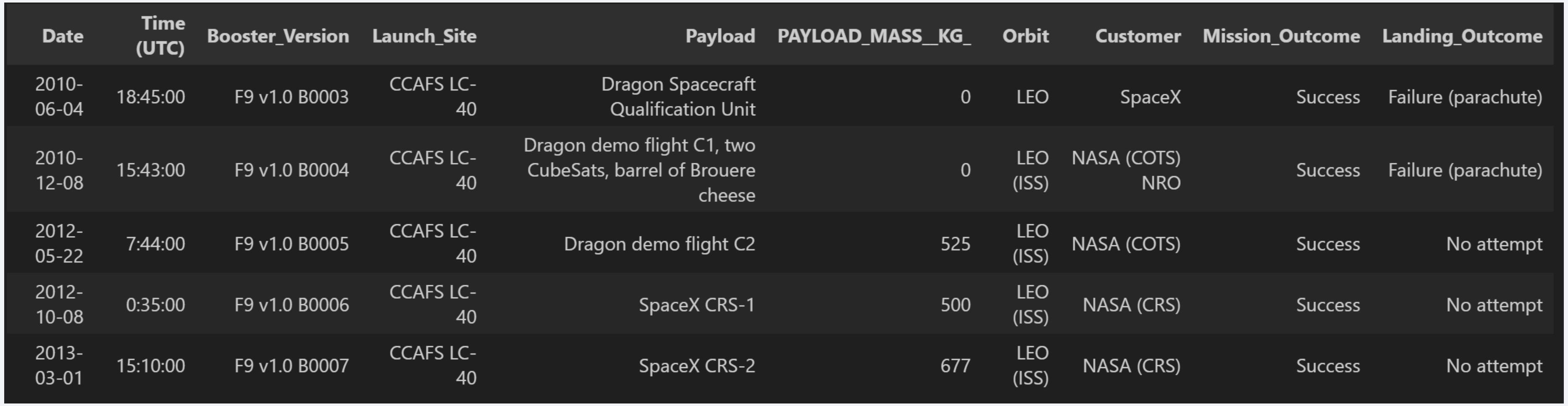
- DISTINCT keyword is used to get the unique site names from the column Launch_Site

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

- 

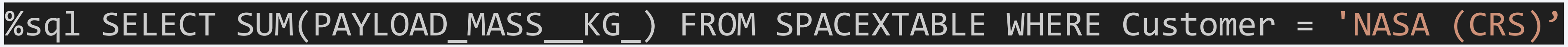
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5
```



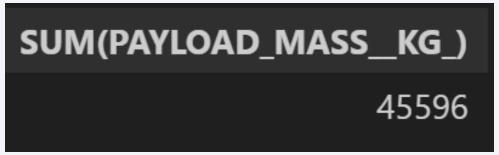
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

- The LIKE keyword is used to get the get the words that start with CCA from the column Launch_Site

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- 

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

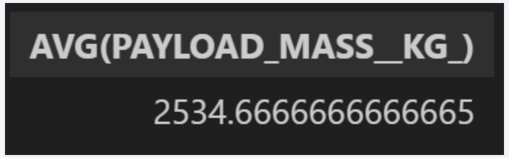


SUM(PAYLOAD_MASS__KG_)
45596

- The SUM and WHERE keywords are used to get the sum of the PAYLOAD_MASS_KG__ column only where the customer is NASA (CRS)

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
-
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%’
```



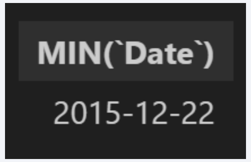
AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- The AVG, WHERE and LIKE keywords are used to get the average value of the column PAYLOAD_MASS_KG__ where the column Booster_Version has a value that starts F9 v1.1

First Successful Ground Landing Date

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

-
  ```
  %sql SELECT MIN(`Date`) FROM SPACEXTABLE WHERE Landing_Outcome LIKE "Success (ground pad%”);
  ```



MIN(`Date`)
2015-12-22

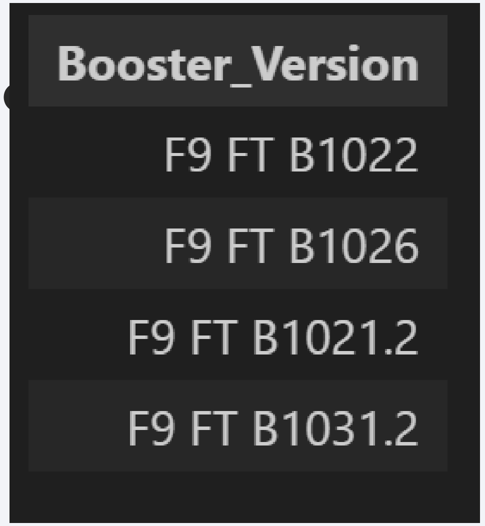
- The MIN ,WHERE and LIKE keywords are used to get the minimum date where the Landing_Outcome column has success listed

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

```
%%sql SELECT Booster_Version FROM SPACEXTABLE
WHERE (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000) AND Landing_Outcome LIKE "Success (drone ship)%";
```

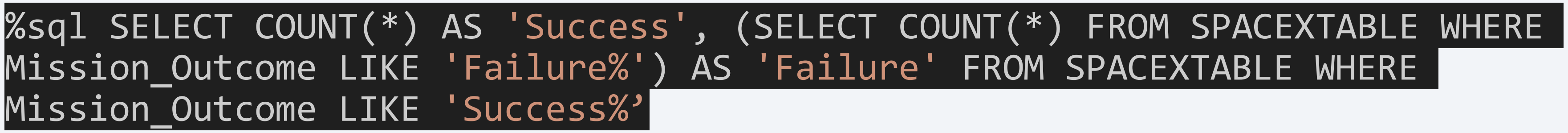
- The BETWEEN, LIKE, AND and WHERE keywords are used to get the values is the Booster_Version column where the value of the PAYLOAD_MASS_KG__ are between 4000 and 6000 and the value of the Landing_Outcome column starts with Success (drop ship)



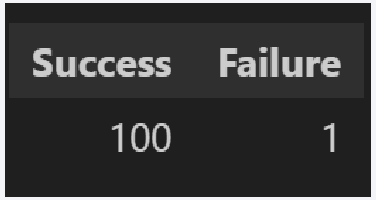
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

- 

```
%sql SELECT COUNT(*) AS 'Success', (SELECT COUNT(*) FROM SPACEXTABLE WHERE
Mission_Outcome LIKE 'Failure%') AS 'Failure' FROM SPACEXTABLE WHERE
Mission_Outcome LIKE 'Success%’
```



Success	Failure
100	1

- Here a subquery is used to evaluate a WHERE keyword for two different conditions: checking the contents of the Mission_Outcome column for value that starts with success and failure

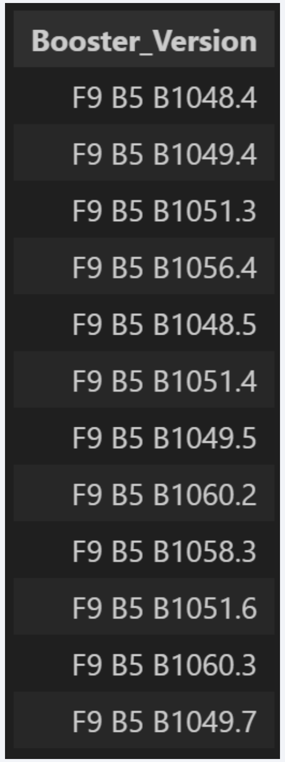
Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

-
  ```
  %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT
  MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
  ```
- Here a subquery is used to find the maximum mass using the MAX

Keyword which is then equated to the column value which corresponds

To the max value

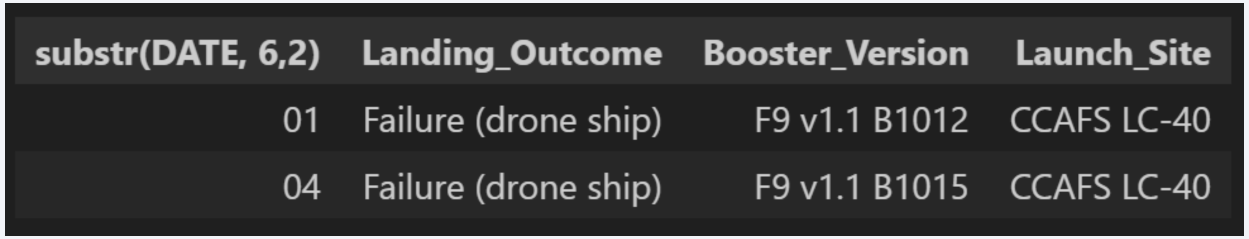


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
SELECT substr(DATE, 6,2),Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
WHERE substr(Date, 0,5) = '2015' AND Landing_Outcome LIKE 'Failure%’
```



substr(DATE, 6,2)	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

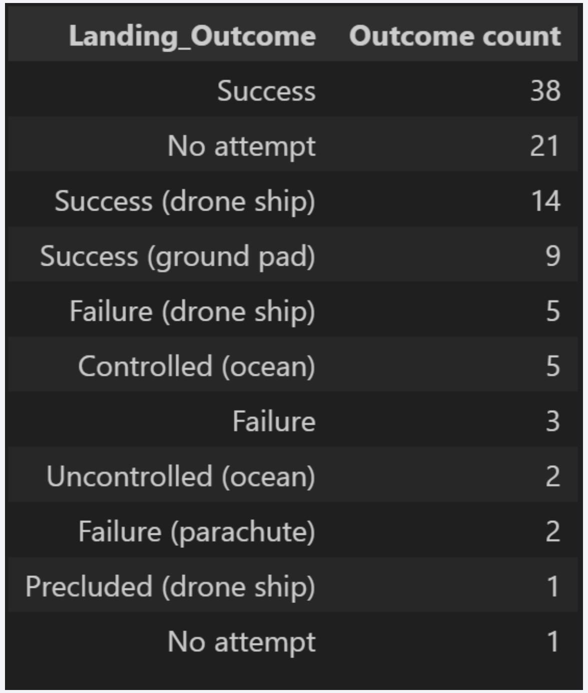
- Here substr function is used to access specific parts of the date value and WHERE , LIKE and AND keywords are used to get the result that fit the required condition

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
- `%%sql`
- `SELECT Landing_Outcome, COUNT(*) AS "Outcome count" FROM SPACEXTABLE`
- `GROUP BY Landing_Outcome`
- `ORDER BY COUNT(*) DESC;`
- Here the GROUP BY and ORDER BY keywords are used to get

The unique values of the colum Landing_Outcome and then order

It in descending order.



Landing_Outcome	Outcome count
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

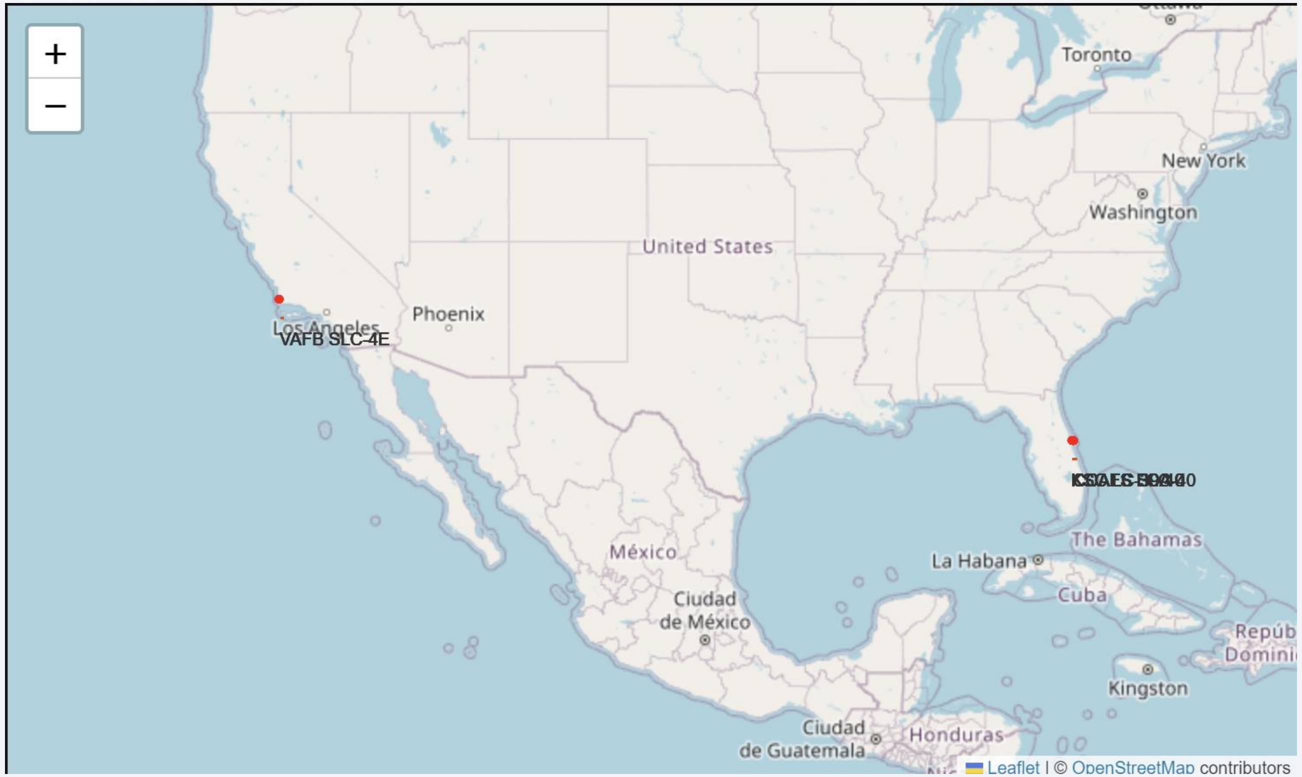
Section 3

Launch Sites Proximities Analysis

SpaceX Launch Sites

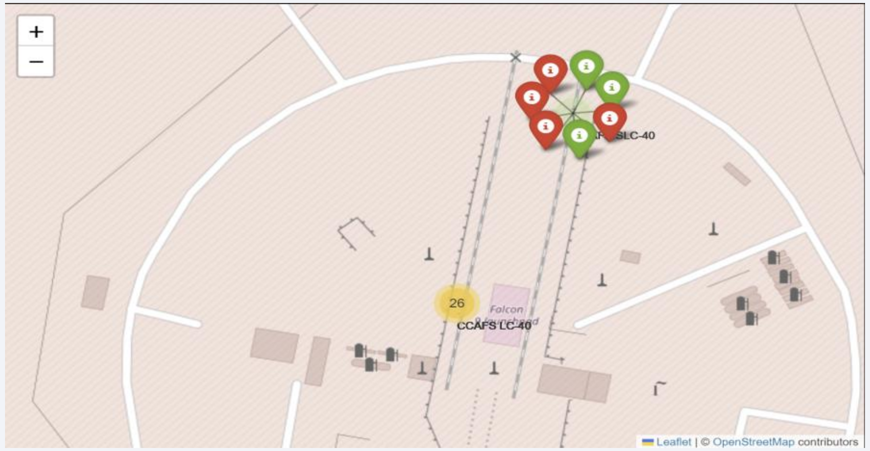
- One of the launch sites is near Santa Maria in Los Angeles , 2 of them are in Cape Cannaveral and the last one is at Merritt Island in Florida.

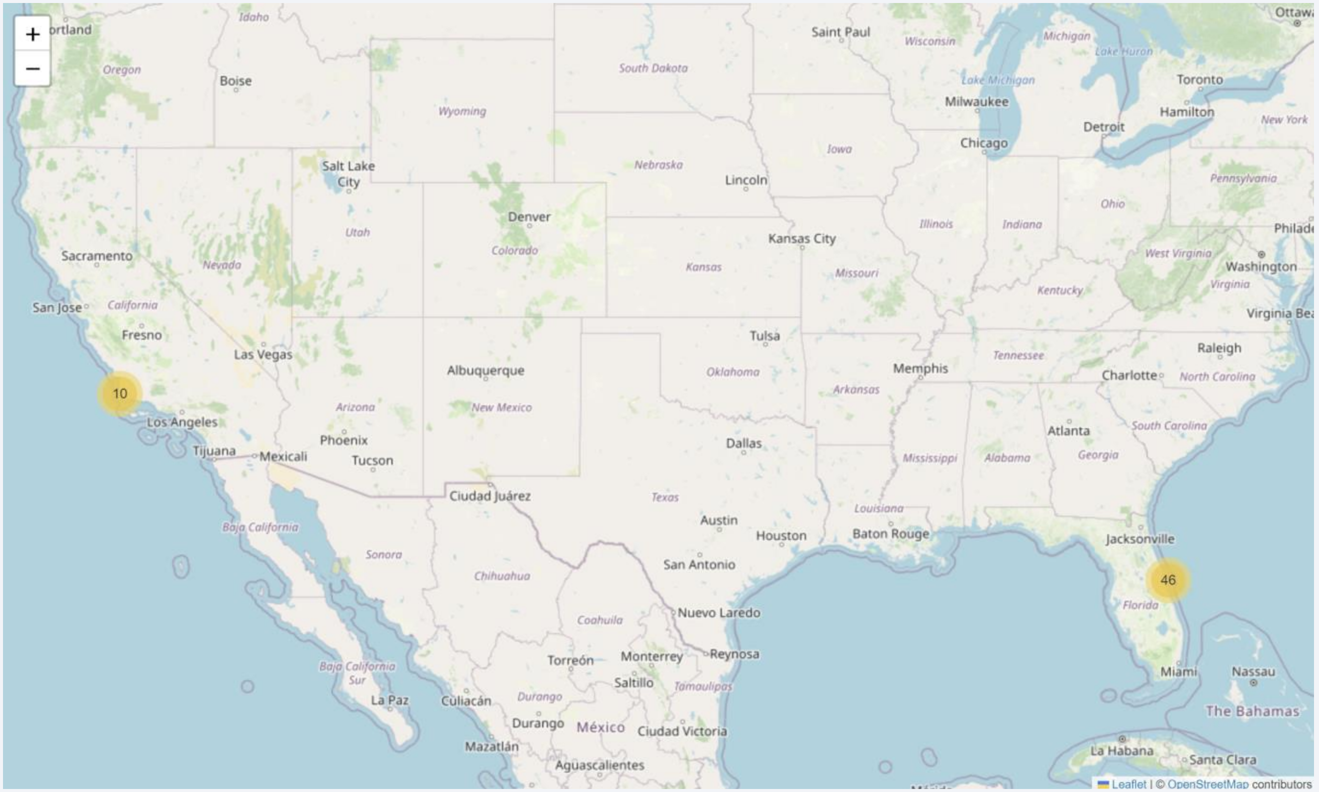
All of the sites are relatively close to the equator



SpaceX Launch Sites Cluster

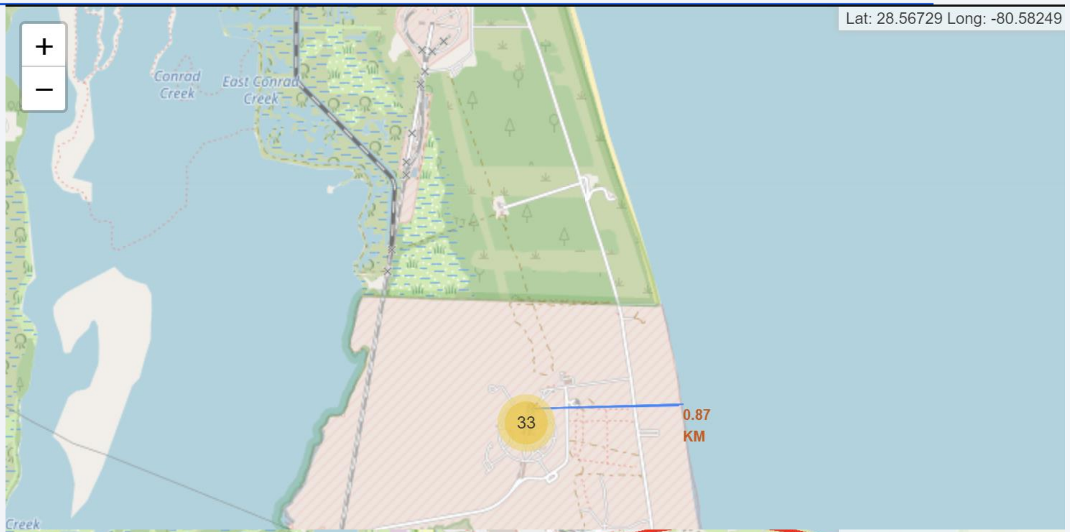
- The site KSC LC- 39A is the one with the relatively highest success rate while the site CCAFS LC-40 has a relatively low success rate

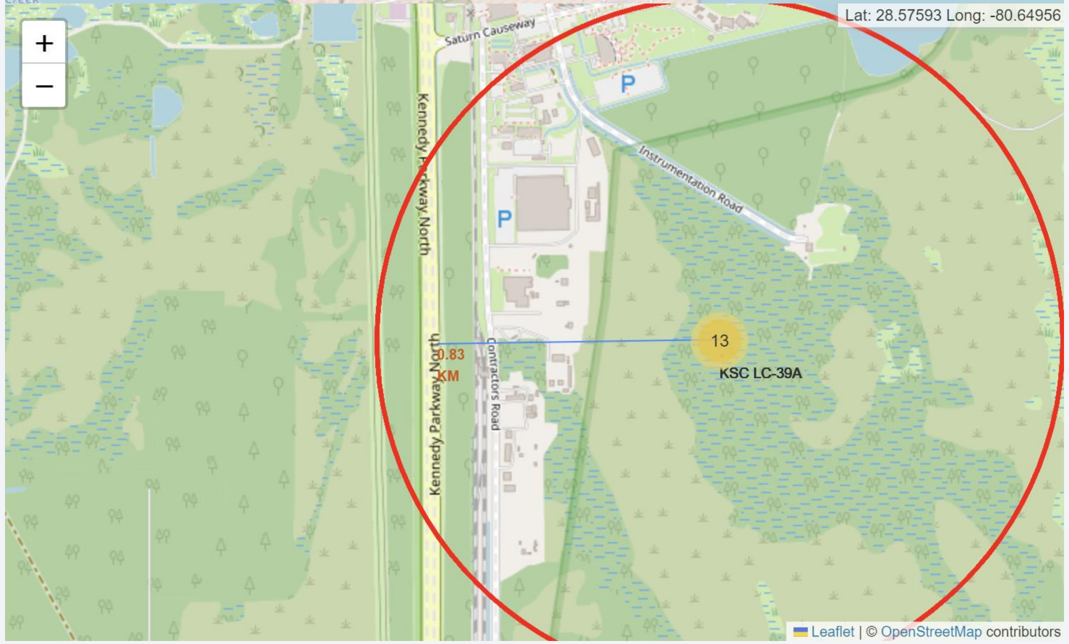


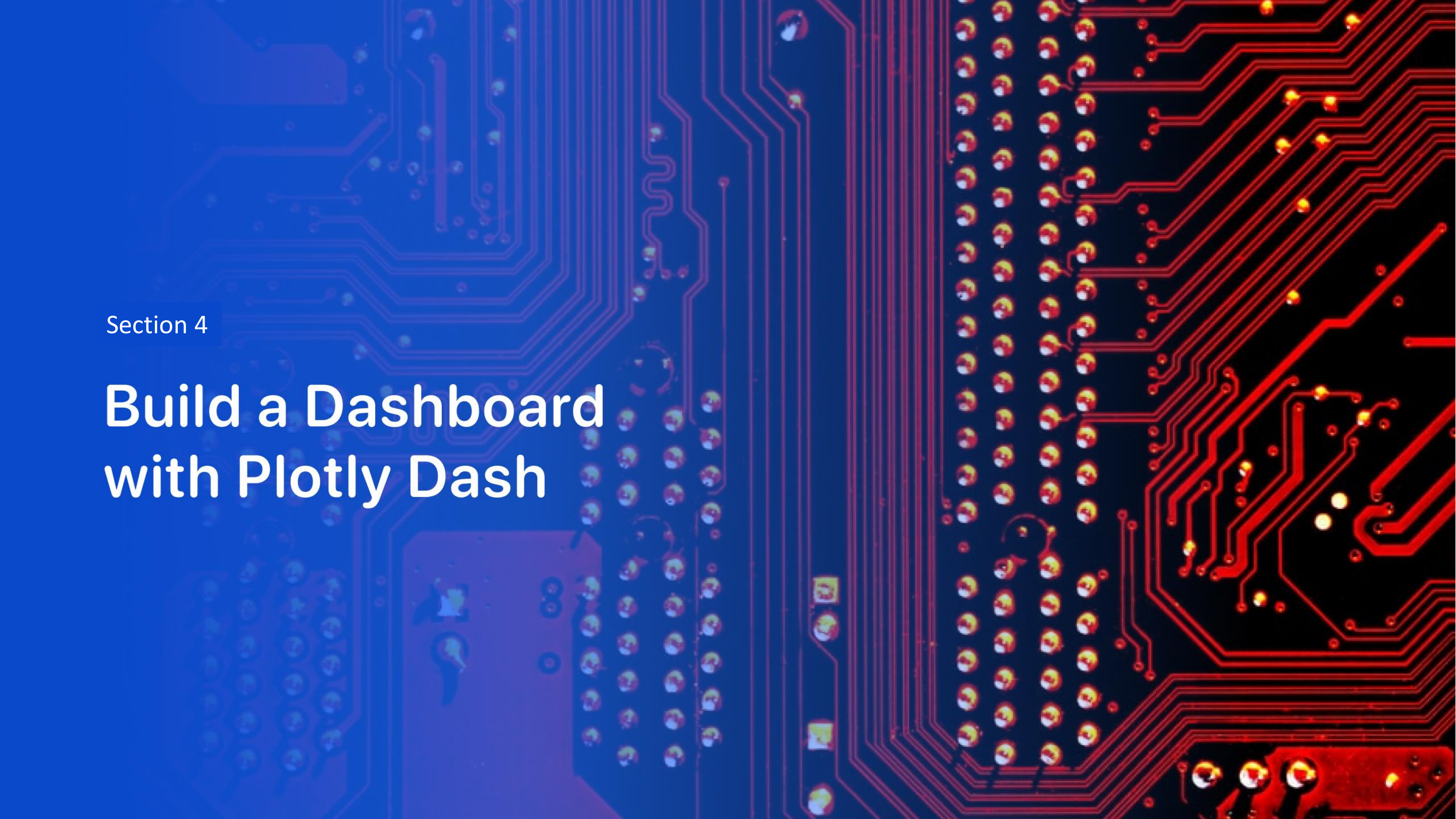


SpaceX Site’s relative Isolation

- The distance between the nearest SpaceX launch site and the coastline is 0.87km, the launch centre is CCAFS SLC-40
- The distance between the nearest SpaceX launch site and the railway line is 0.83 km, the launch centre is KSC LC- 39A





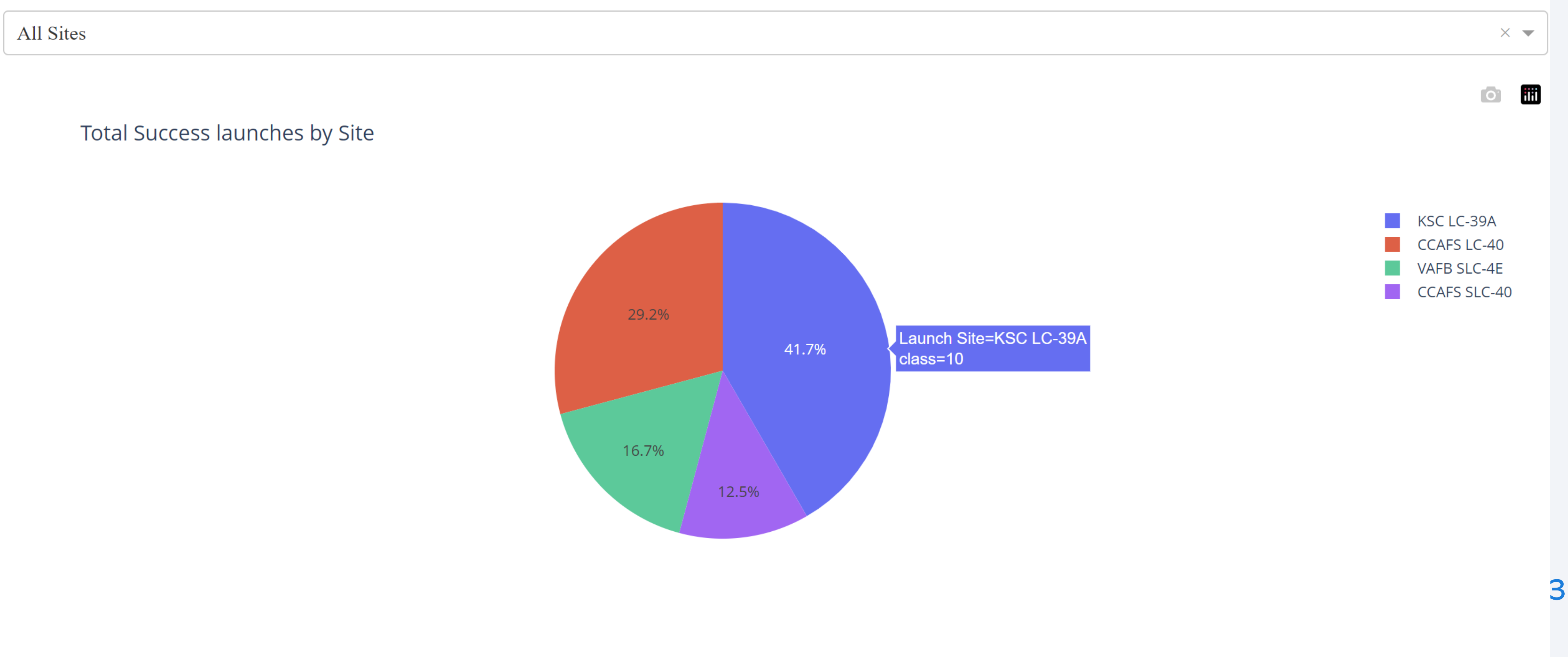


Section 4

Build a Dashboard with Plotly Dash

Pie Chart - Total Successful Launches by Site

- The site with the most successful launches is KSC LC- 39A and the site with the least successful launches is CCAFS LC-40. With this information we can predict that the site with the probability that the launch will be successful is KSC LC- 39A

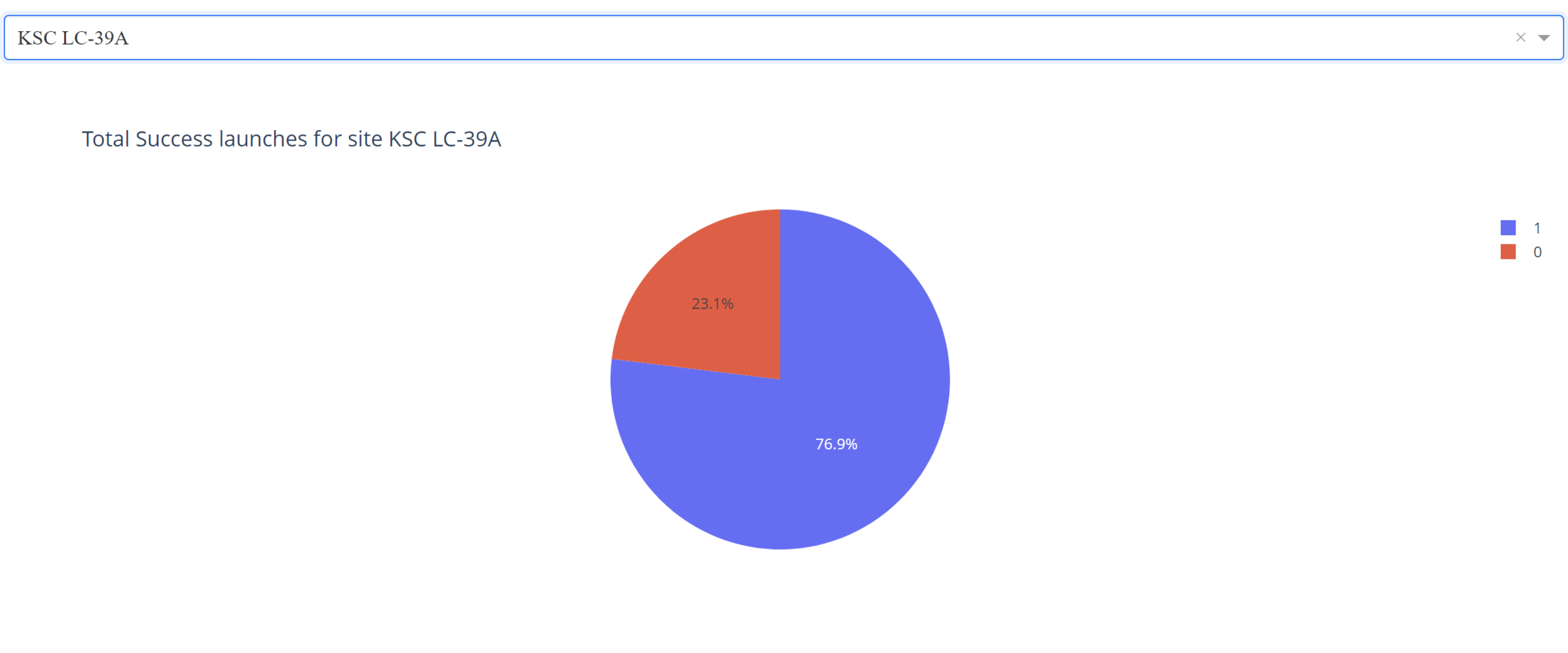


All Sites

Total Success launches by Site

Pie Chart - Launch Success Ratio of a site

- The pie chart shows that the success percentage of the launches in site KSC LC- 39A is 76.9 % and the failure percentage is 23.1%

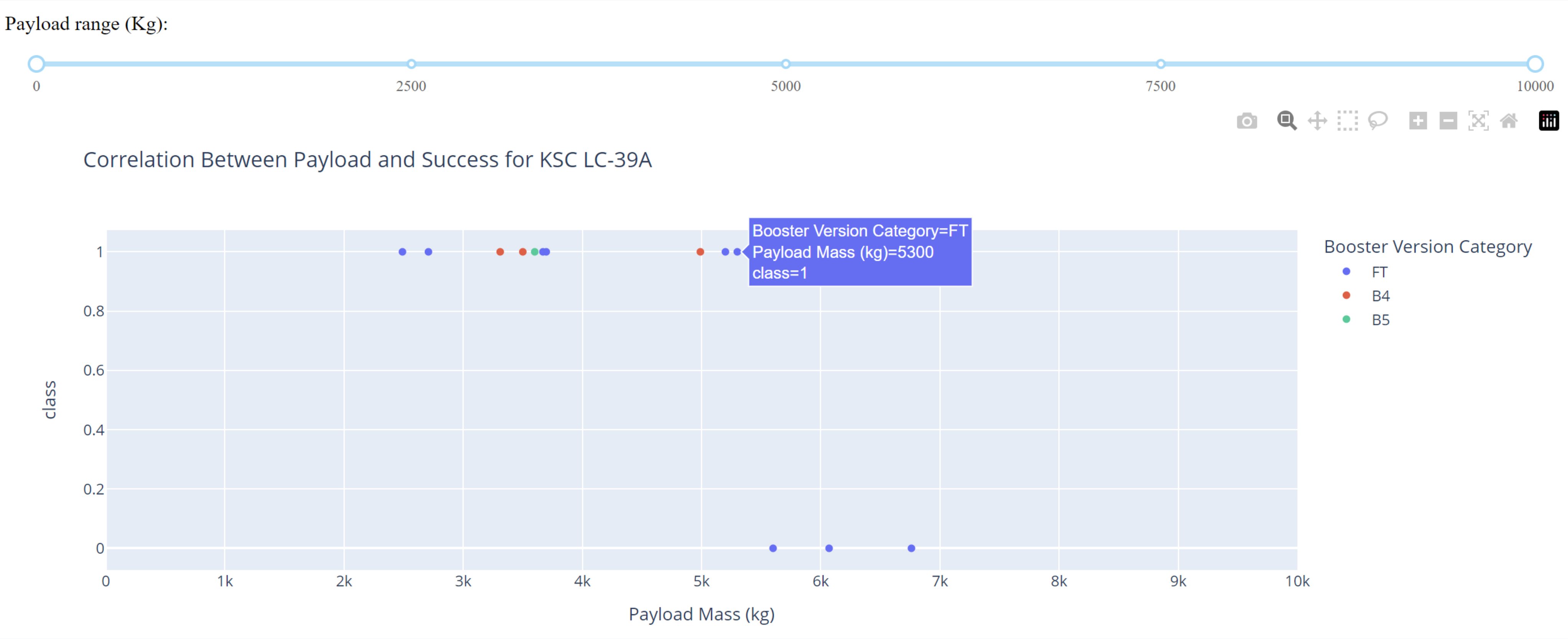


KSC LC-39A

Total Success launches for site KSC LC-39A

Scatter Plot – Payload Mass vs Success rate

- From the below graph we can see that all of the failures have been from Booster Version Category FT which carries a payload of above 5300kg and the rest of the launches are a success, but their payload masses are below 5300kg

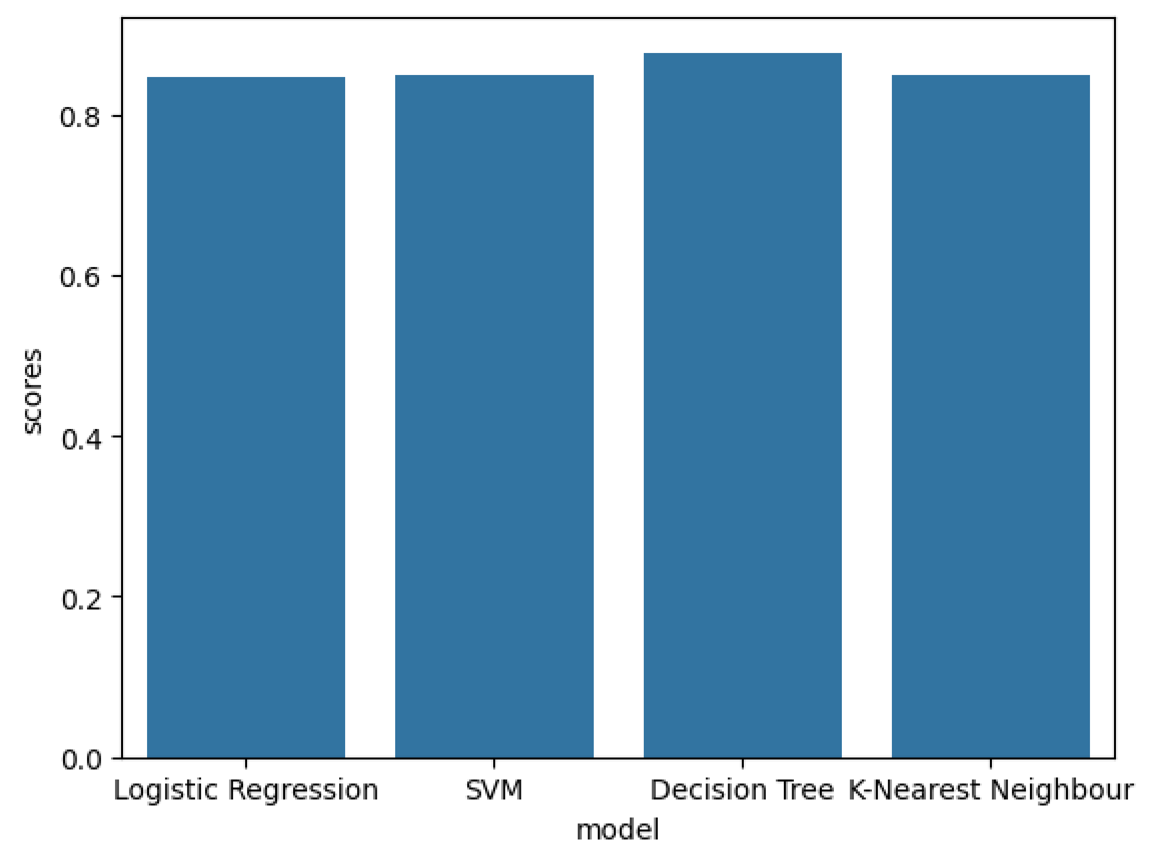


Section 5

Predictive Analysis (Classification)

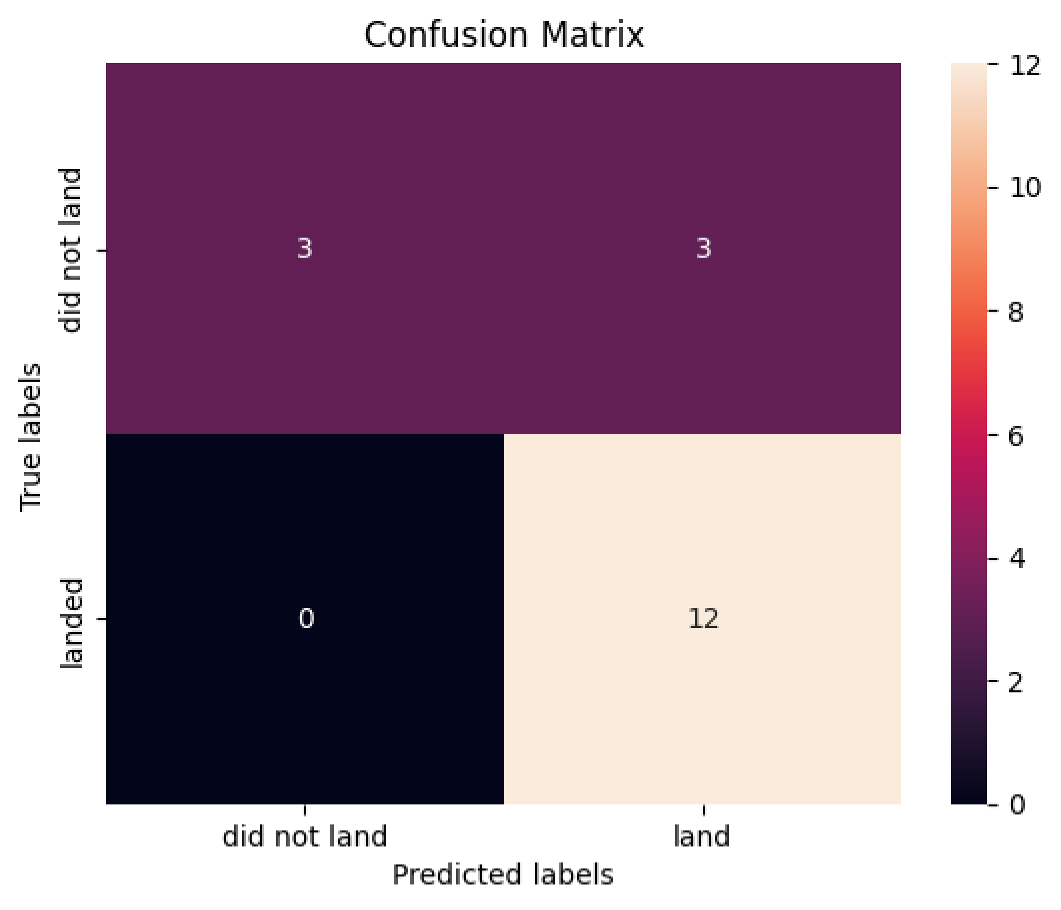
Classification Accuracy

- The highest best accuracy when calculated using the '.best_score_' method of the GridSearchCV object is given by the Decision Tree model
- But it can be inferred that the difference in accuracy of the models is not significant coming in less than 0.3 score



Confusion Matrix

- The best performing model is the Decision Tree model with an best accurary score of 0.876
- The confusion matrix shows us that the model predicted the outcomes of a successful landing with an accuracy of 100% while the accuracy of prediction of an unsuccessful launch is only 50%,ie the model gives False Positive values



Conclusions

- The best accuracy score is held by the Decision Tree model while using the ‘.best_score_’ method
- The accuracy score of the all the models are same when testing with the test dataset , coming in 0.8333333333333334
- The Confusion matrix of all the models are also the same with all of the models showing the same False positive values
- So it can be concluded that all the four models work well with this SpaceX dataset with the Decision Tree barely coming to the top spot, and all these models can be used to predict the future launch outcomes and can provide satisfying results

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!