IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

YETTOU FARES
10/12/2022

Outline

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

Executive Summary

- **Summary of methodologies :**

1) Data collection via API, SQL and Web Scraping
2) Data wrangling and Analysis
3) EDA with data visualization
4) EDA with SQL
5) Building an Interactive map with folium
6) Building a Dashboard with Plotly Dash
7) Predictive Analysis -classification

- **Summary of all results :**

1) Exploratory data analysis results
2) Interactive Visualizations in screenshots
3) Predictive Analysis results

Introduction

Project background and context :

The aim of this project is to predict if the falcon 9 first stage will land successfully. SpaceX advertises Falcon 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 the launch. This information can used if an alternate company wants to bid against SpaceX

Problems to be addressed :

- What influences if the rocket will land successfully?
- What is the effect of each relationship of rockets variables on outcomes?
- What are the Conditions which will aid SpaceX to achieve the best results?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology :
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling :
  - One Hot Encoding data fields for Machine learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL :
  - Plotting : Scatter and bar graphs to show relationship between variables and to show patterns of data
- Perform interactive visual analytics using Folium and Plotly Dash :
  - Using Folium and Plotly Dash visualizations
- Perform predictive analysis using classification models :
  - Build, tune, evaluate classification models

Data Collection

- ## How data sets were gathered :

I worked with SpaceX launch data that is gathered from SpaceX REST API.

The API gives us data about launches, including information about the rocket used, payload delivered, launches specifications, landing specifications and landing outcome.

Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.

The SpaceX REST API endpoints, or URL, start with Api. SpaceX data t.come/v4/.

The second data source for obtaining falcon 9 launch data is through web scrapping Wikipedia using Beautiful Soup.(A popular python library use for web scrapping)

- ## SpaceX API :

Use SpaceX RESTAPI=> API returns SpaceX detain JSON =>Normalize data into flat data file such as.csv

- ## Web Scrapping :

Get HTML Response from Wikipedia= > Extract using beautiful soup library = > Normalize into flat data such as .csv

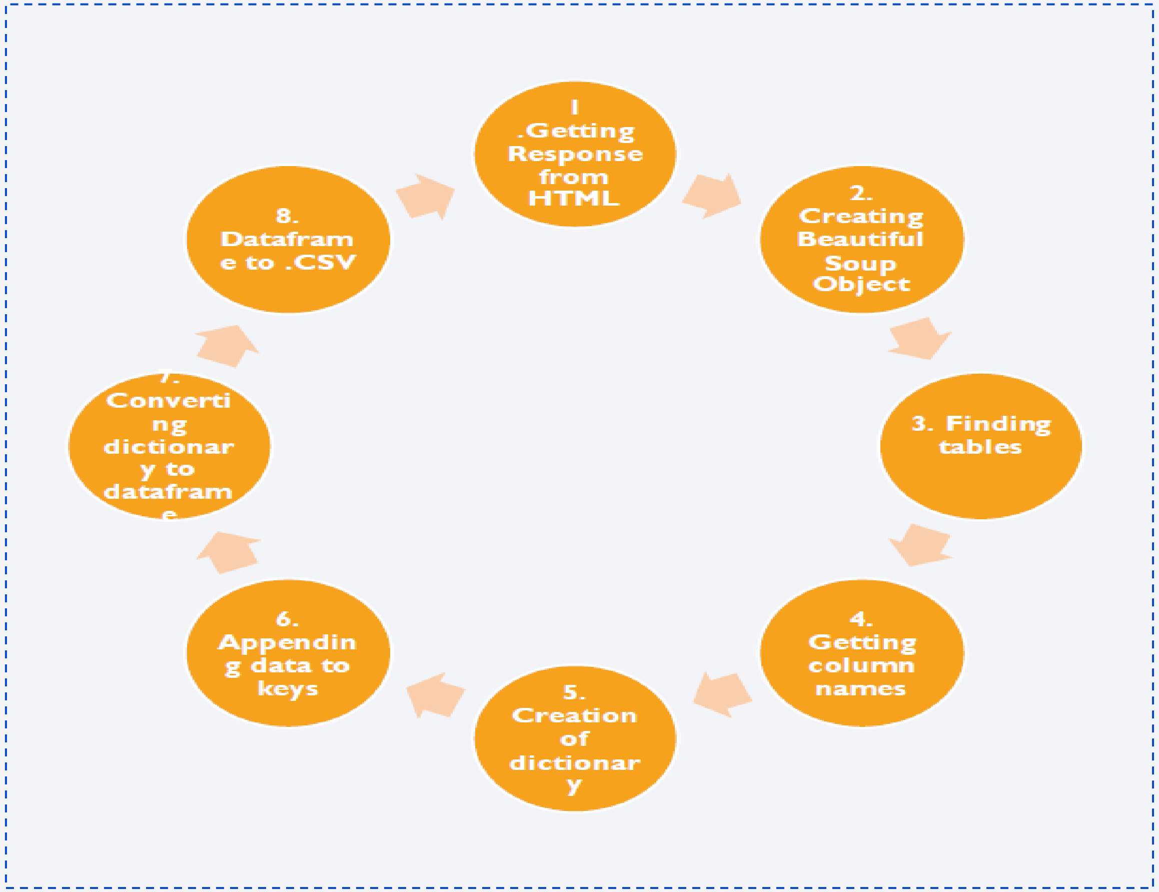
Data Collection – SpaceX API

1) Getting Response from API
2) Converting Response to a .json file
3) Apply custom functions to clean data
4) Assign list to dictionary then data frame
5) Filter data frame and export to flat file (.csv)

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



1 .Getting Response from HTML

2. Creating Beautiful Soup Object

3. Finding tables

4. Getting column names

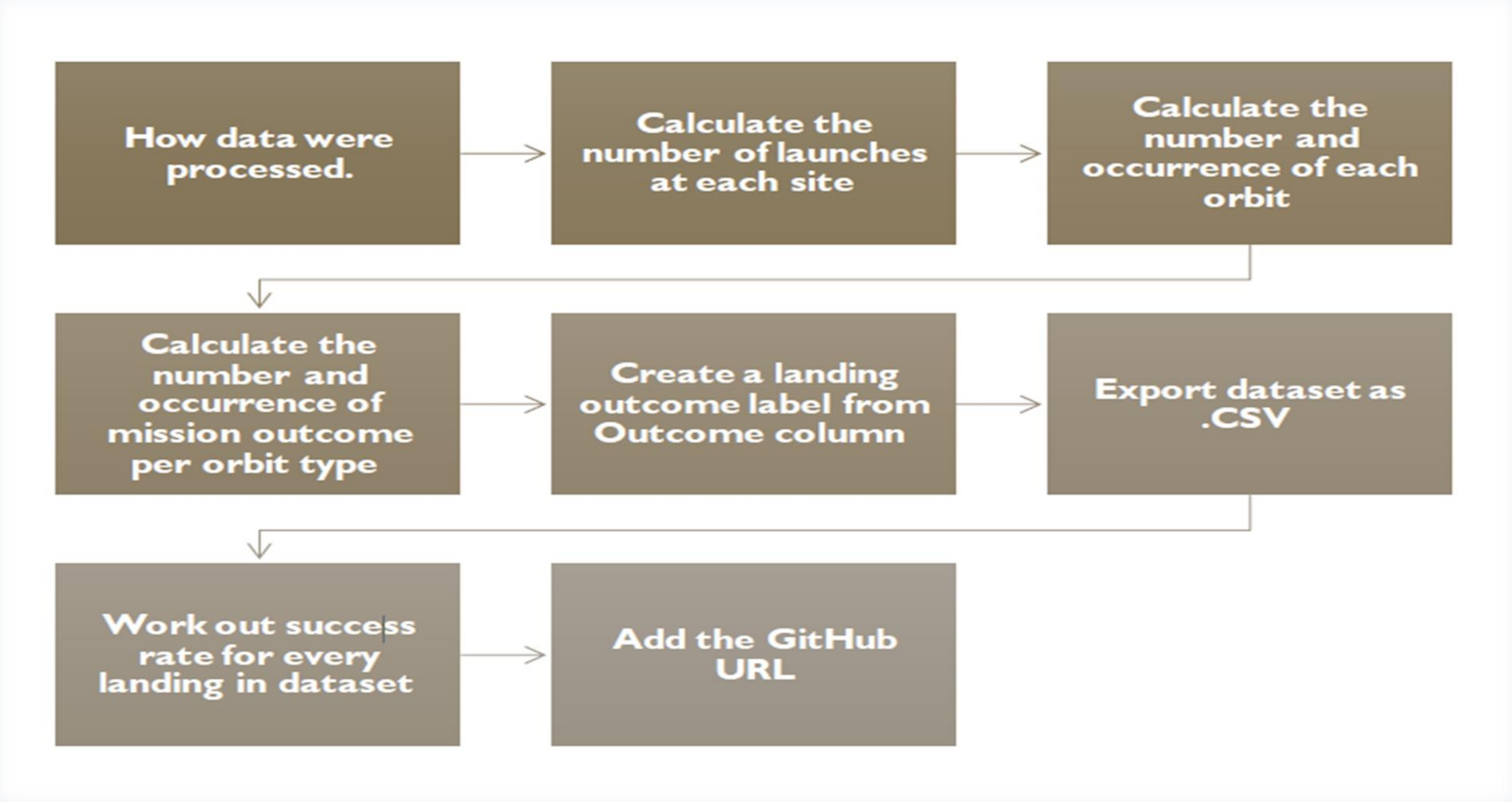
5. Creation of dictionary

6. Appending data to keys

7. Converting dictionary to dataframe

8. Dataframe to .CSV

Data Wrangling



How data were processed.

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Export dataset as .CSV

Work out success rate for every landing in dataset

Add the GitHub URL

EDA with Data Visualization

Scatter Graphs being drawn :

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data.

- Flight Number VS. Payload
- Mass Flight Number VS.
- Launch Site Payload VS. Launch Site-Orbit VS. Flight Number Payload VS. Orbit Type
- Orbit VS. Payload Mass

Bar Graph being drawn :

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

- Mean VS. Orbit

Line Graph being drawn :

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

- Success Rate VS. Year

EDA with SQL

Summary of the SQL queries I performed :

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass.
- Listing the records which will display the month names, successful landing outcomes in ground pad, booster versions, launch site for the months in year 2017
- Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

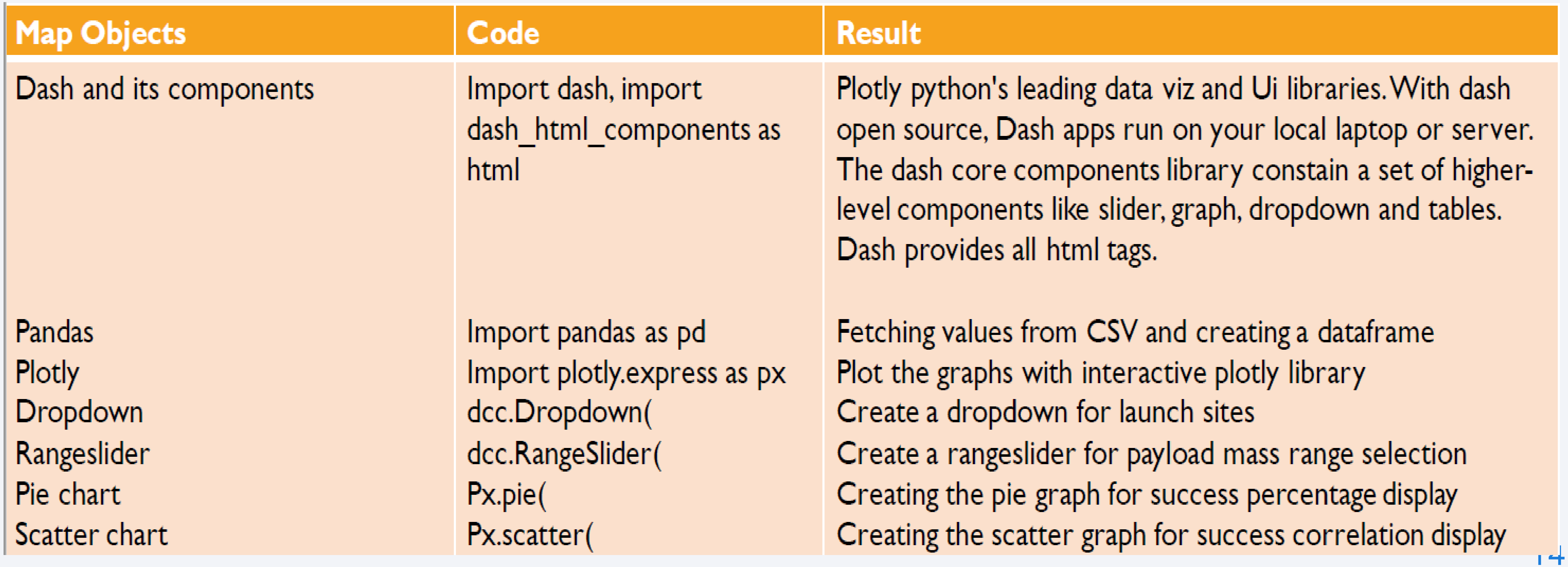
Build an Interactive Map with Folium

Folium makes it easy to visualize manipulated data in python on an interactive leaflet map. We use the latitude and longitude coordinates for each launch site and added a Circle Marker around each launch site with a label of the name of the launch site. Also, it is easy to visualize the number of success and failure for each launch site with green and Red markers on the map

Map Objects	Code	Result
Map Marker	folium.Marker(	Map object to make a mark to map
Icon Marker	folium.icon(	Create an icon on map
Circle Marker	folium.Circle(	Create a circle where marker is being placed
Polyline	folium.Polyline(	Create a line between points.
Marker Cluster Object	MarkerCluster()	This is a good way to simplify a map containing many markers having the same coordinate.
AntPath	Folium.plugins.AntPath(	Create an animated line between points

Build a Dashboard with Plotly Dash

- Pie chart shows the total success for all sites / by certain launch site
- Scatter graph shows the correlation between payload and success for all sites



Map Objects	Code	Result
Dash and its components	Import dash, import dash_html_components as html	Plotly python's leading data viz and Ui libraries. With dash open source, Dash apps run on your local laptop or server. The dash core components library constain a set of higher-level components like slider, graph, dropdown and tables. Dash provides all html tags.
Pandas	Import pandas as pd	Fetching values from CSV and creating a dataframe
Plotly	Import plotly.express as px	Plot the graphs with interactive plotly library
Dropdown	dcc.Dropdown(	Create a dropdown for launch sites
Rangeslider	dcc.RangeSlider(	Create a rangeslider for payload mass range selection
Pie chart	Px.pie(	Creating the pie graph for success percentage display
Scatter chart	Px.scatter(	Creating the scatter graph for success correlation display

Predictive Analysis (Classification)

- **Building Model:**

Load our dataset into NumPy and Pandas

Transform data

Split our data into training and test data sets

Check how many test samples we have

Decide which type of machine learning algorithms we want to use

Set our parameters and algorithms to GridSearch_Cv

Fit our datasets into the GridSearch_Cv objects and train our model

- **Evaluating Model :**

Check accuracy for each model

Get tuned hyperparameters for each type of algorithms

Plot confusion Matrix

Improving Model

Feature Engineering

Algorithm Tuning

Finding the best performing classification Model

The model with the best accuracy score wins the best performing model

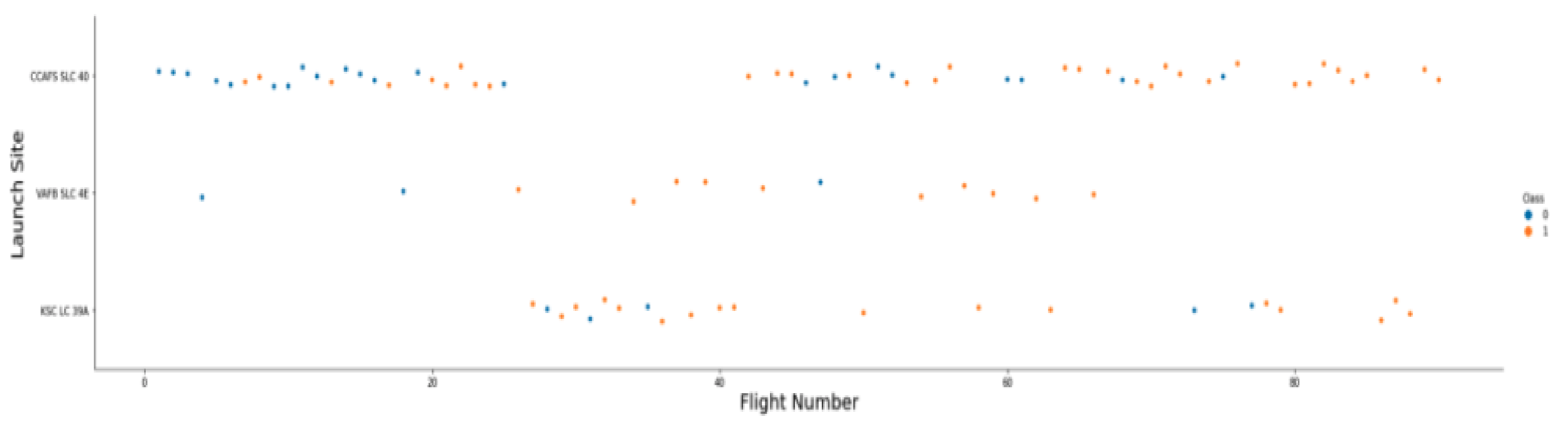
Results

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

Section 2

Insights drawn from EDA

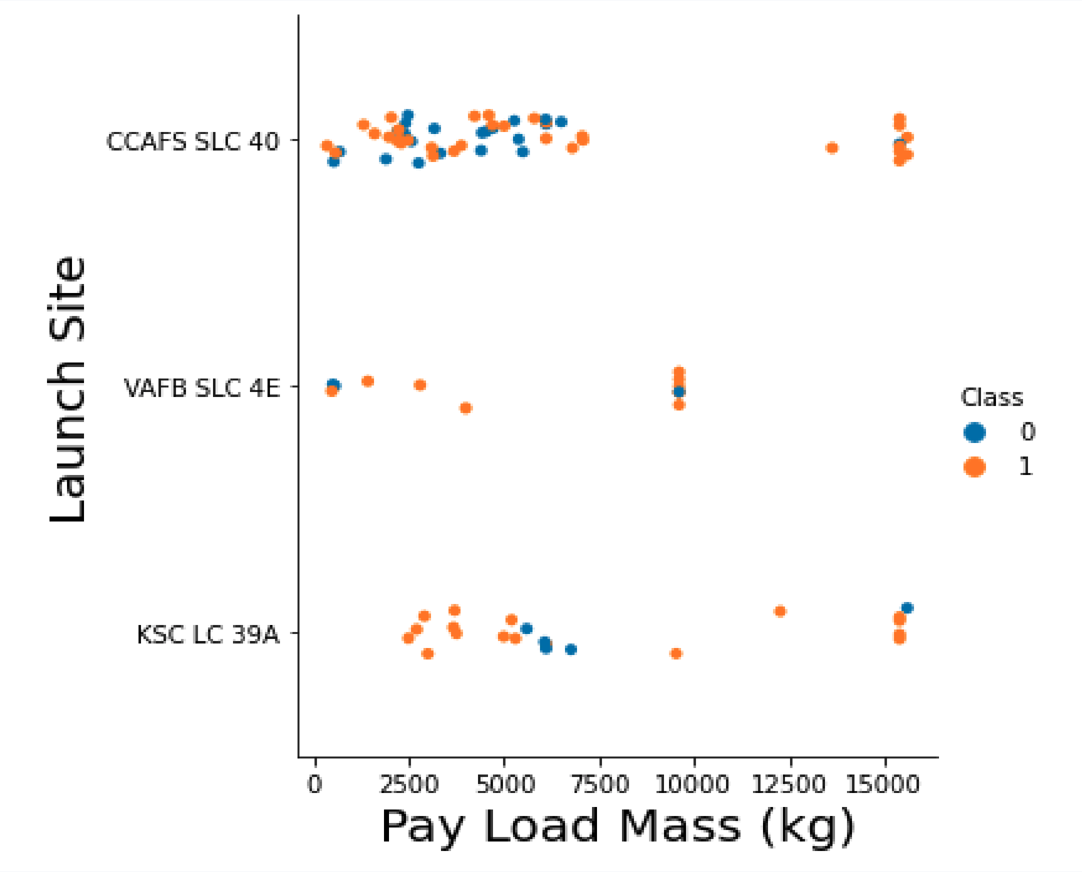
Flight Number vs. Launch Site



- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations

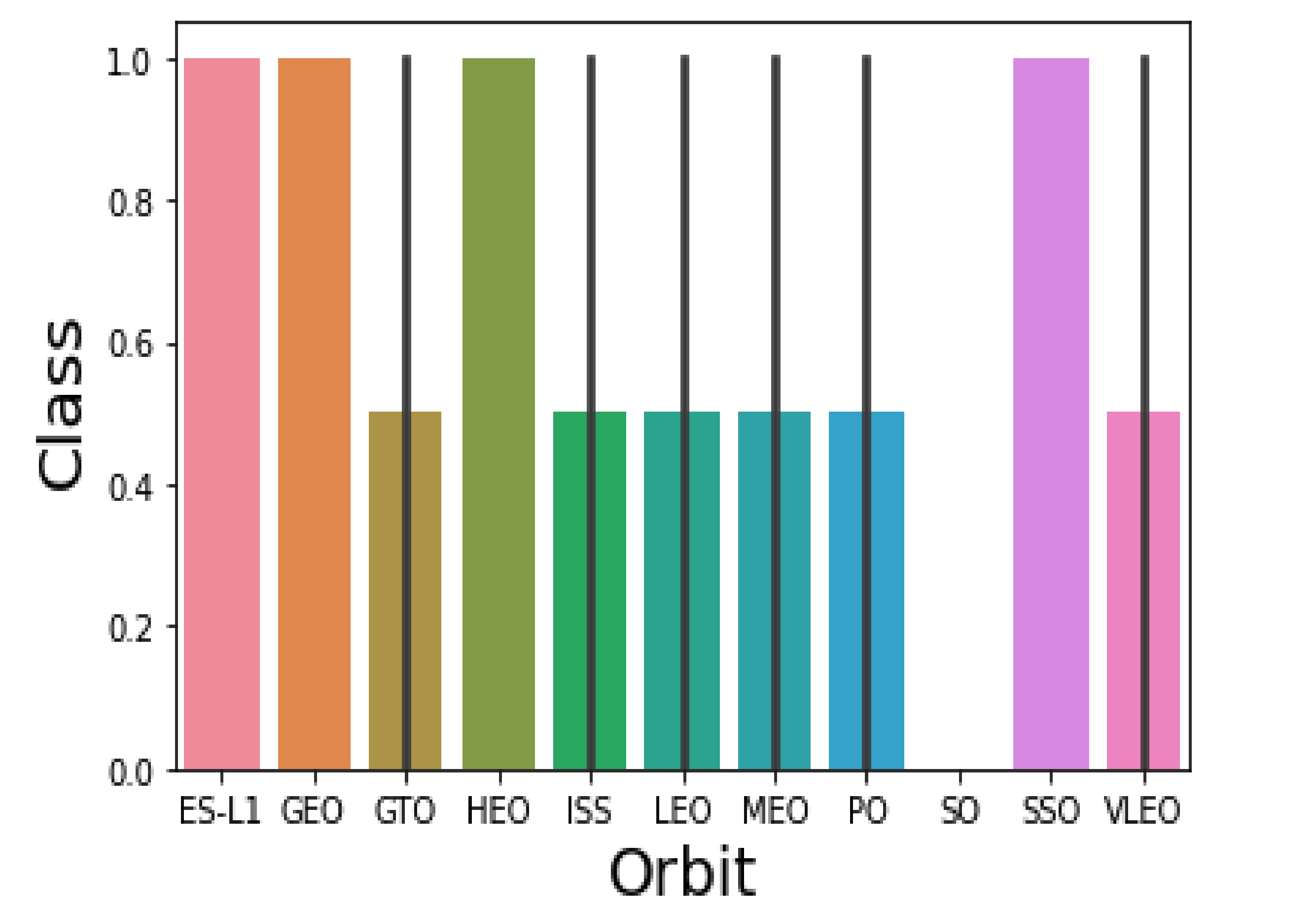
Payload vs. Launch Site

The more the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependent on Pay Load Mass for a success launch.



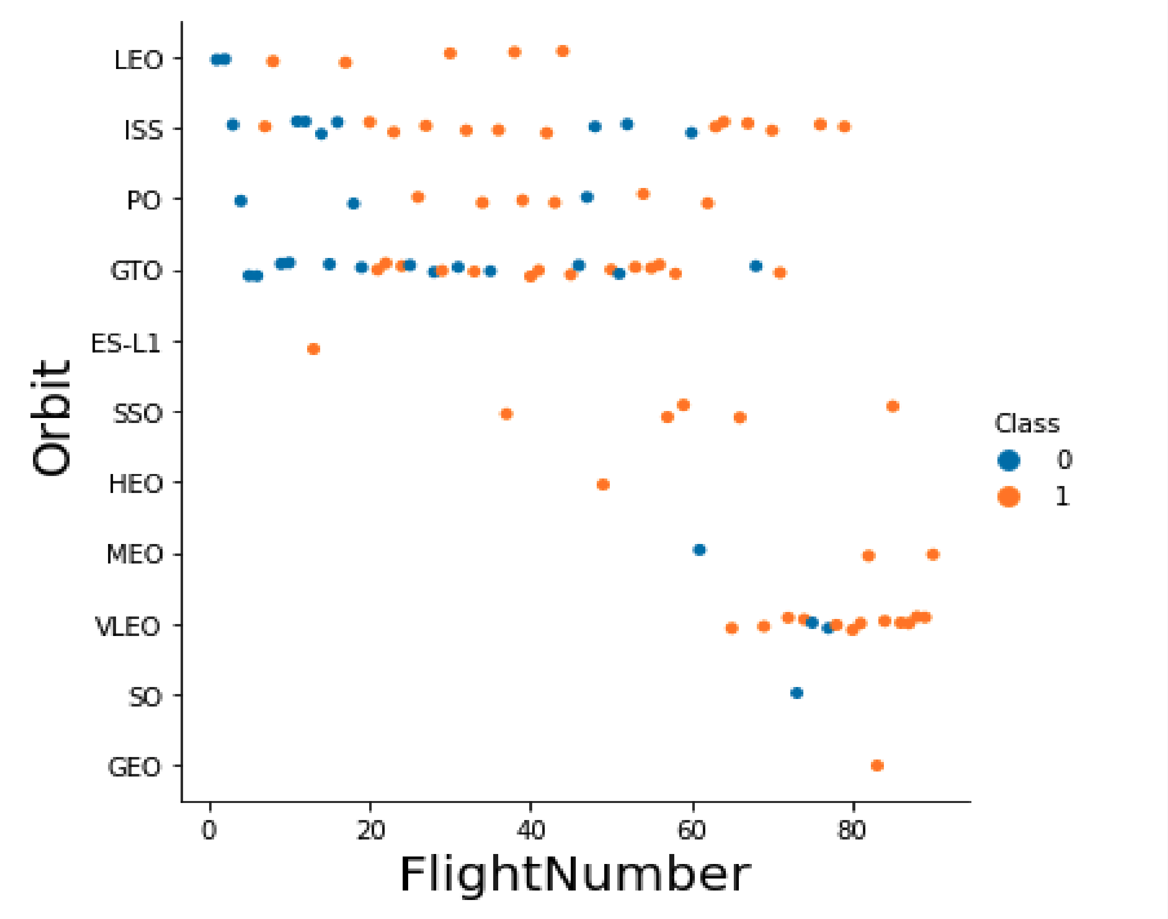
Success Rate vs. Orbit Type

- The Orbit
- GEO,HEO,SSO,ES-L1 has the best Success Rate



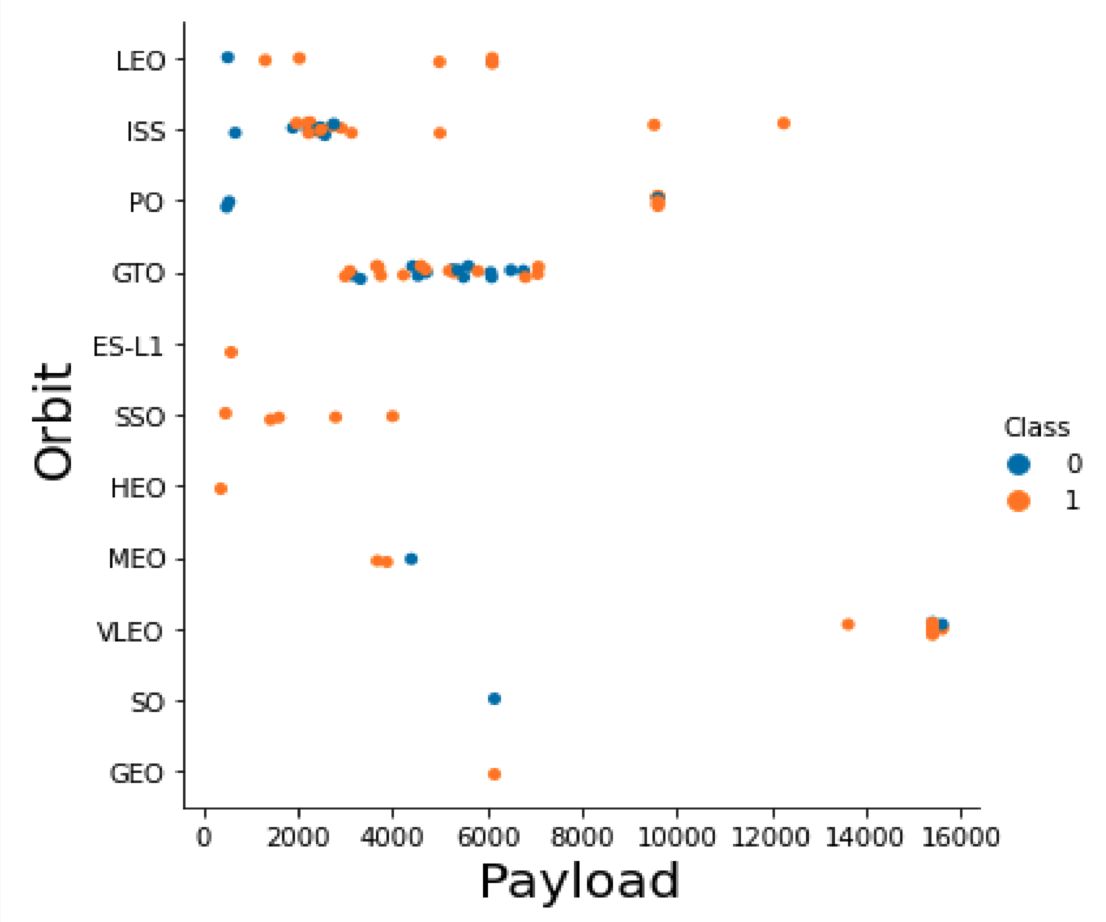
Flight Number vs. Orbit Type

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.



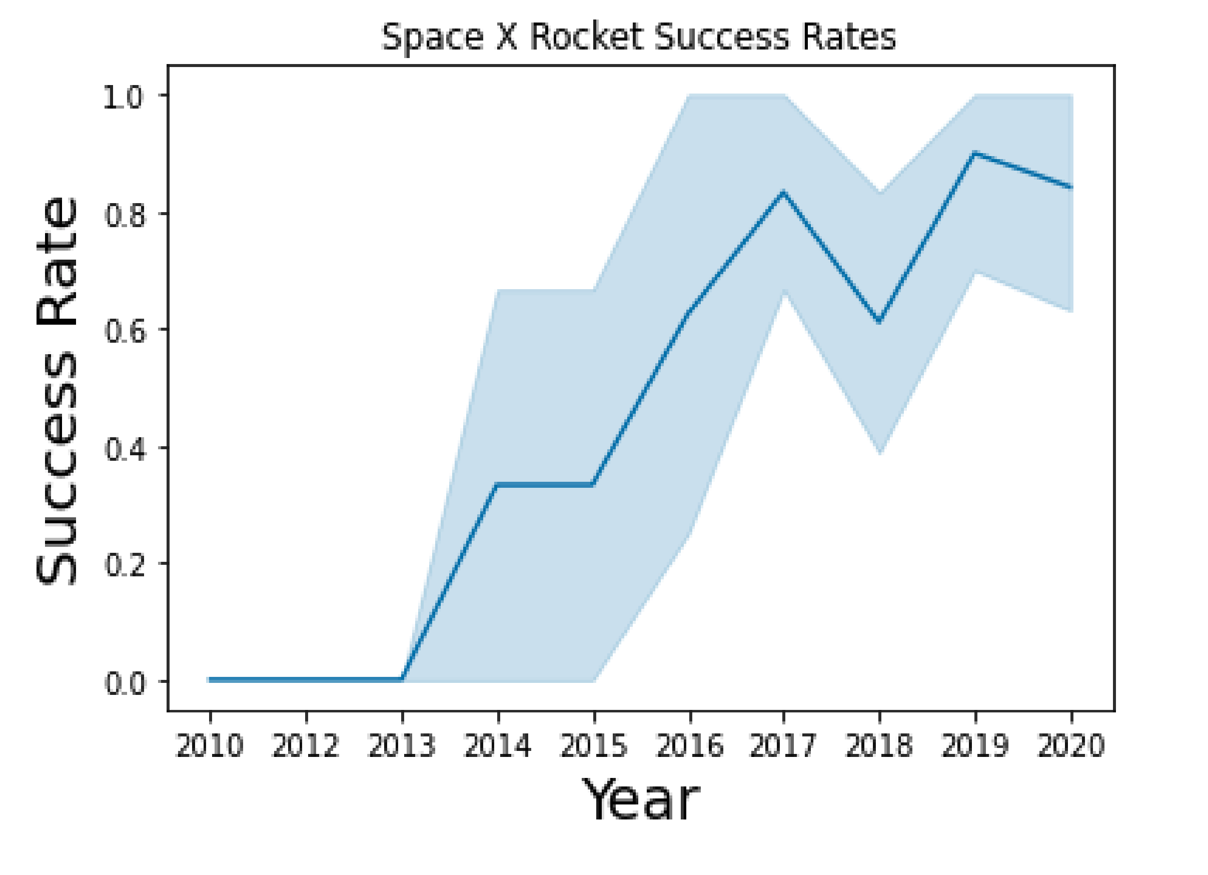
Payload vs. Orbit Type

We can observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



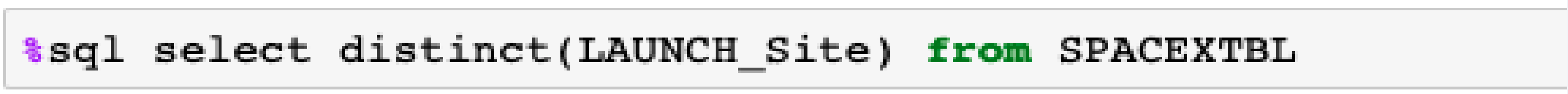
Launch Success Yearly Trend

As we can see that the success rate since 2013 kept increasing till 2020



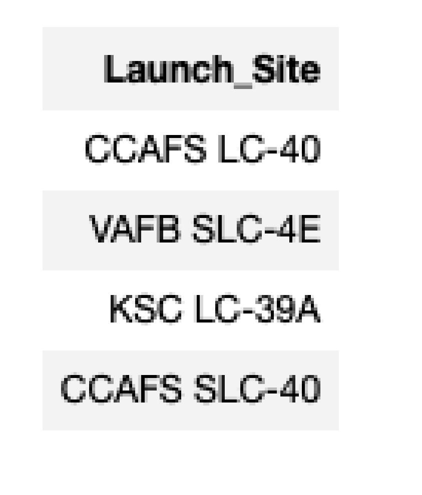
All Launch Site Names

SQL Query :



```
%sql select distinct(LAUNCH_Site) from SPACEXTBL
```

SQL result :

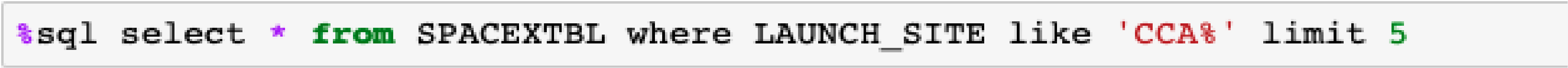


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Description: Using the word distinct in the query will pull the unique values for the launch Site column from the table SPACEXTBL

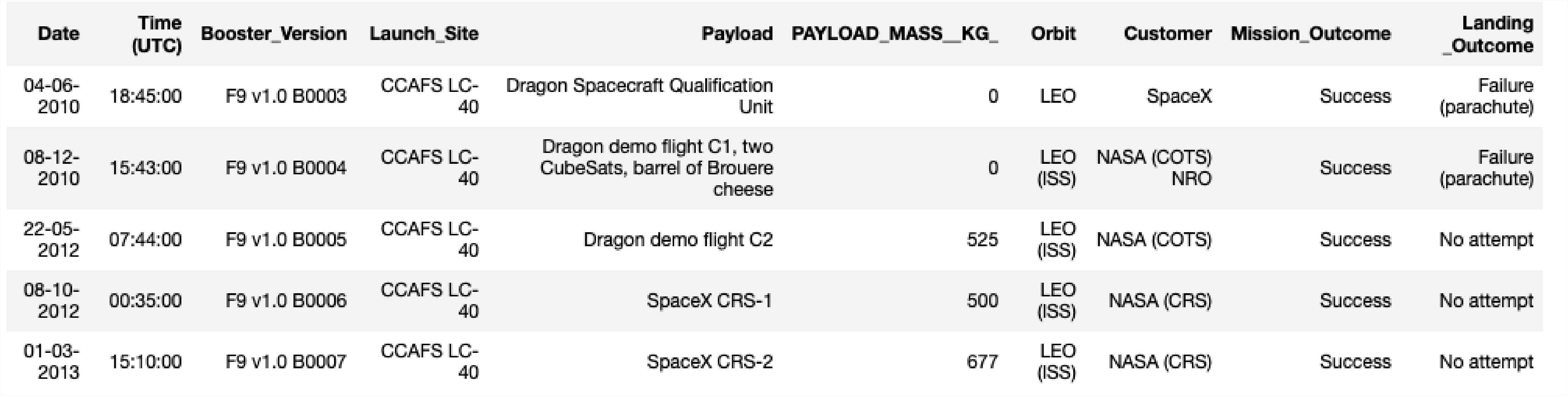
Launch Site Names Begin with 'CCA'

SQL Query :



```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

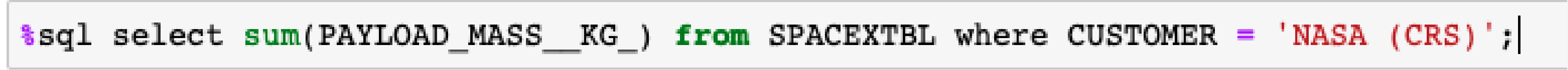
Description: Using keyword “Limit 5” in the query will fetch 5 records from table SpaceX, condition LIKE keyword with wild card “CCA%”. The percentage in the end suggest that the launch site name must start with CCA.



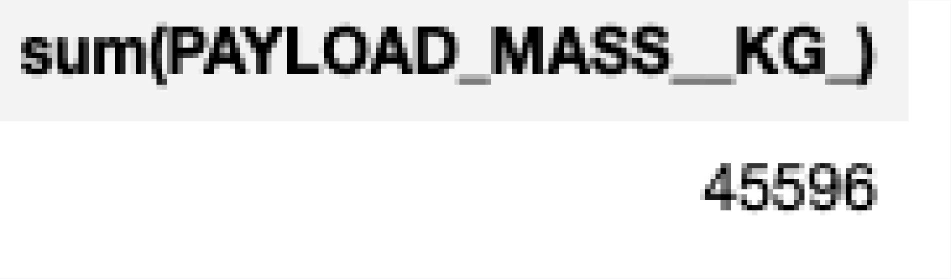
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

SQl Query :



```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

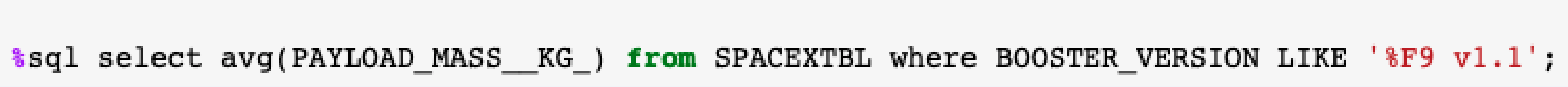


sum(PAYLOAD_MASS__KG_)
45596

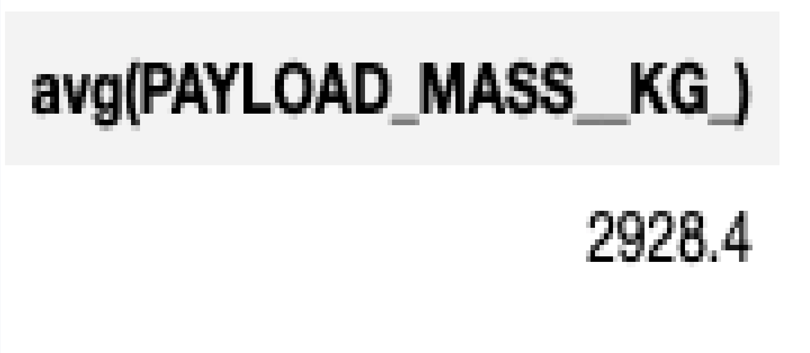
Description: Using function Sum summate the total in the column PAYLOAD_MASS_KG and the WHERE clause filters the dataset to only perform calculations on customer NASA (CRS)

Average Payload Mass by F9 v1.1

SQl Query :



```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION LIKE '%F9 v1.1';
```



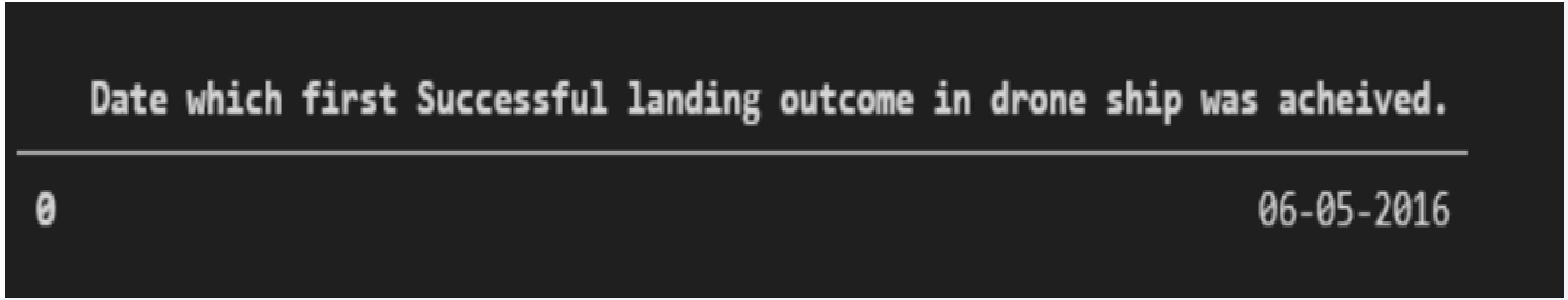
avg(PAYLOAD_MASS__KG_)
2928.4

Description :

Using the function AVG works out the average in the column PAYLOAD_MASS_KG
The WHERE clause filters the dataset to only perform calculations on Booster_versionf9 v1.1

First Successful Ground Landing Date

SQL Query:
select MIN(Date) SLO from table SpaceX where Landing Outcome= ”Success (drone ship)”

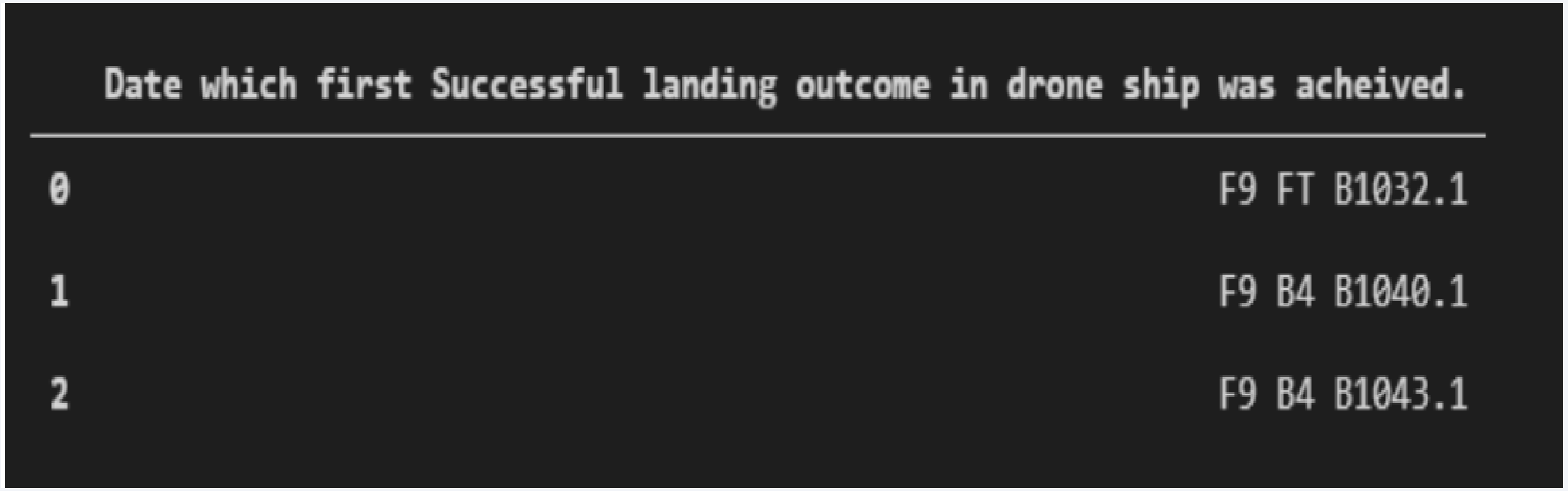


```
   Date which first Successful landing outcome in drone ship was acheived.
0                                                               06-05-2016
```

Description: The function MIN works out the minimum date in the column date while the WHERE clause filters the dataset to only perform calculations on Landing outcome success

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL Query:
- select Booster Versionfrom tbl SpaceX where Landing Outcome = 'Success (ground pad)’ AND Payload_MASS_KG_ > 4000 AND Payload MASS_KG_ < 6000



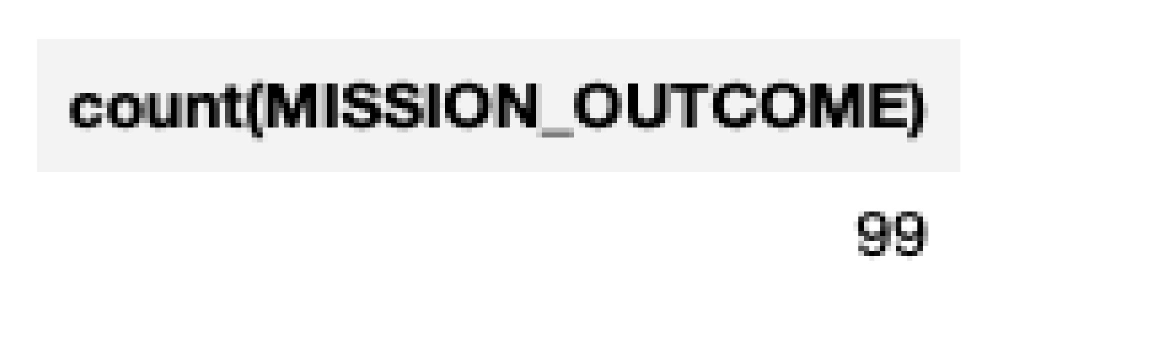
```
   Date which first Successful landing outcome in drone ship was acheived.
0                                                         F9 FT B1032.1
1                                                         F9 B4 B1040.1
2                                                         F9 B4 B1043.1
```

Description: Selecting only Booster Version. WHERE clause filters AND clause specifies additional filter.

Total Number of Successful and Failure Mission Outcomes

SQL Query:

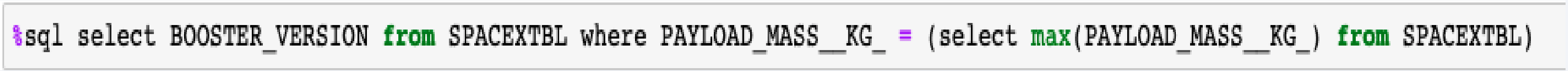
```
%sqlselect count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)’
```



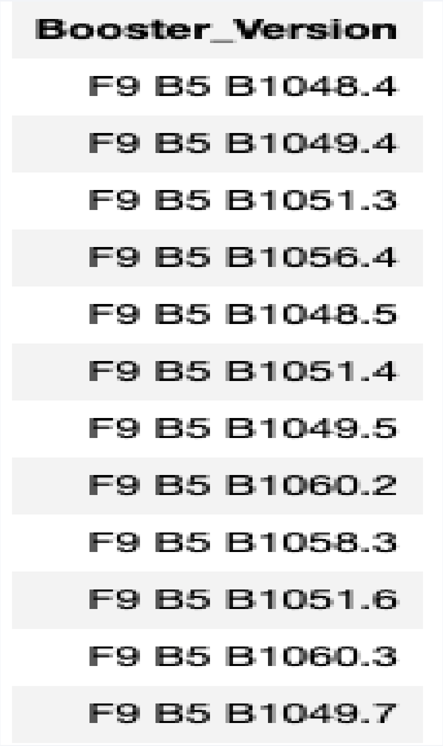
count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

SQL Query :



```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```



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

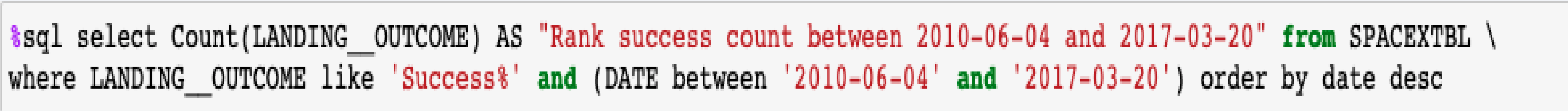
Description: Using the function MAX works out the maximum payload in the column PAYLOAD_MASS_KG_ in the sub query and WHERE clause filters Booster Version which had that maximum payload.

2015 Launch Records

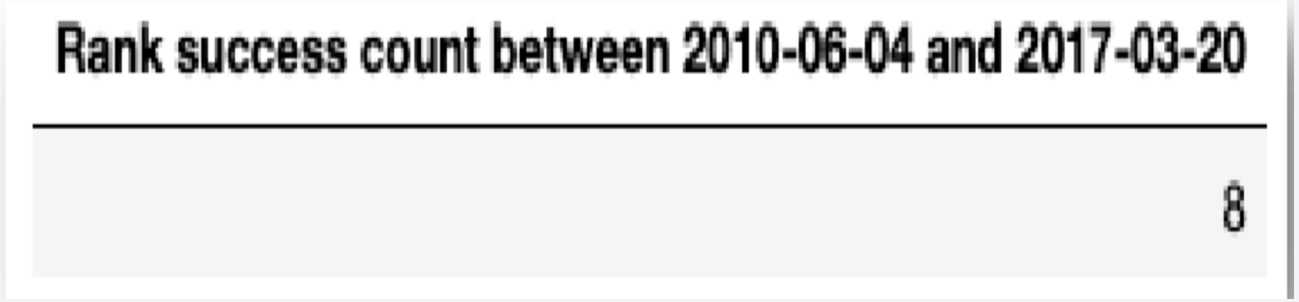
- 33
- SQL Query
- %SQL SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
- LANDING__OUTCOME = 'Failure (drone ship)';

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

SQ query :



```
%sql select Count(LANDING__OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-20" from SPACEXTBL \
where LANDING__OUTCOME like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```



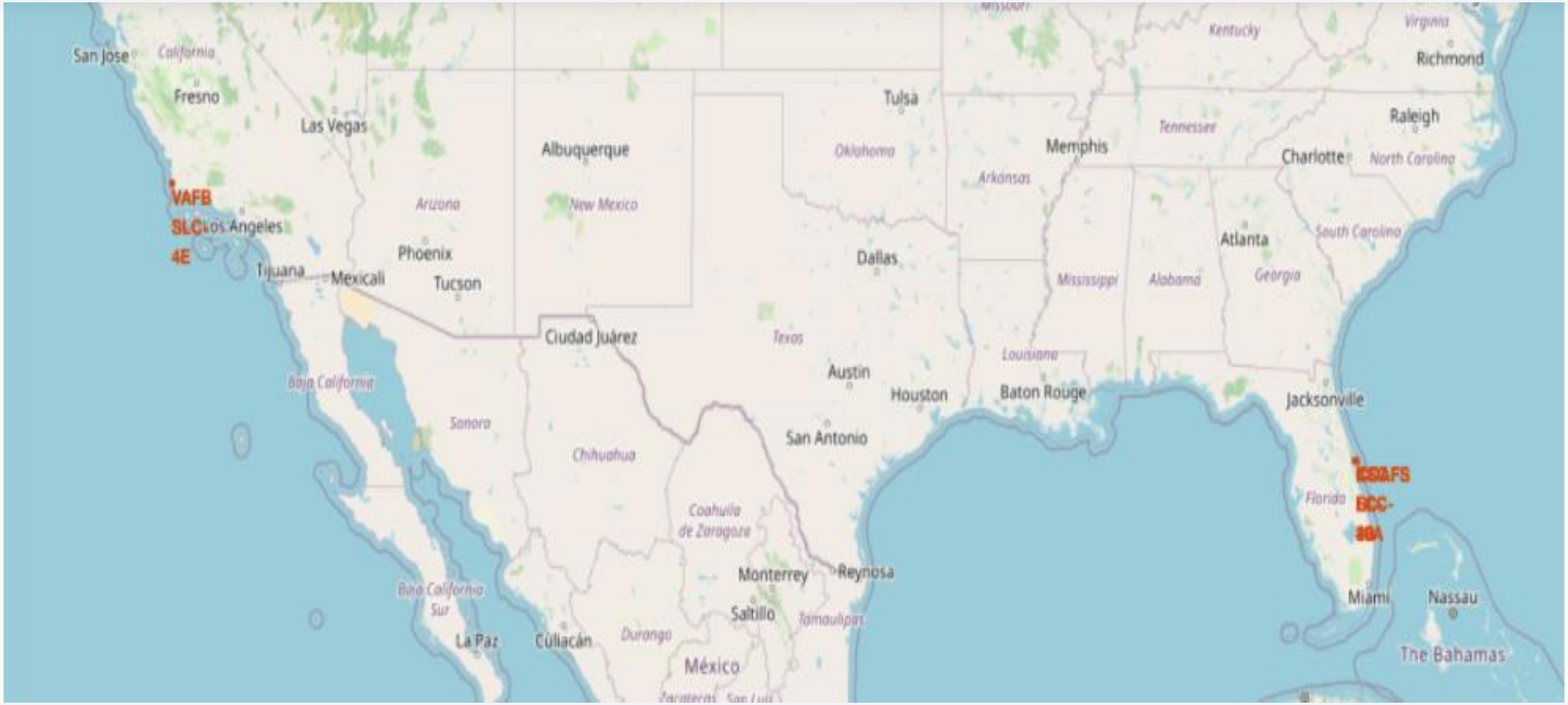
Rank success count between 2010-06-04 and 2017-03-20
8

Description:
COUNT counts records in column LANDING__OUTCOME. WHERE filters data with '%Success%'

Section 3

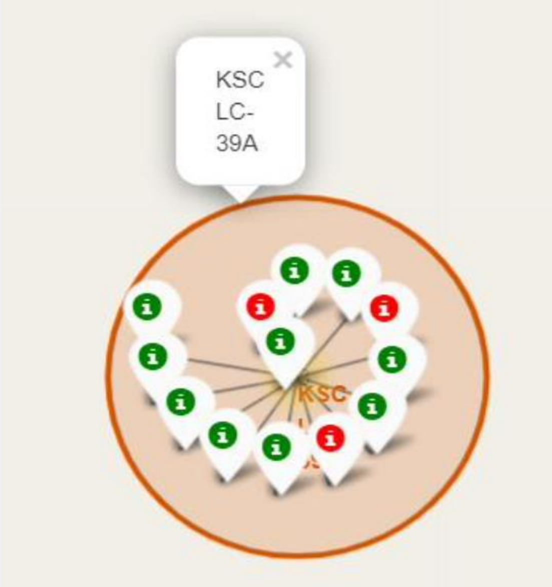
Launch Sites Proximities Analysis

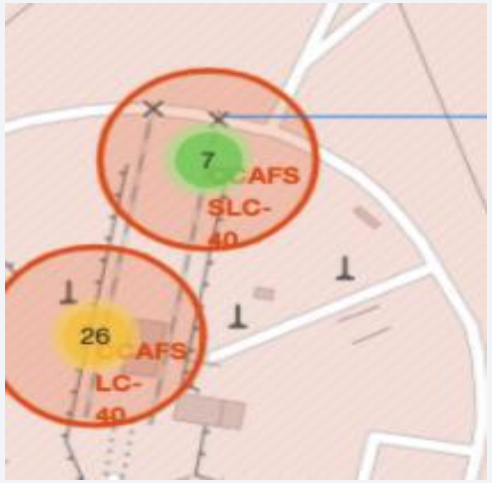
<Folium Map Screenshot 1>



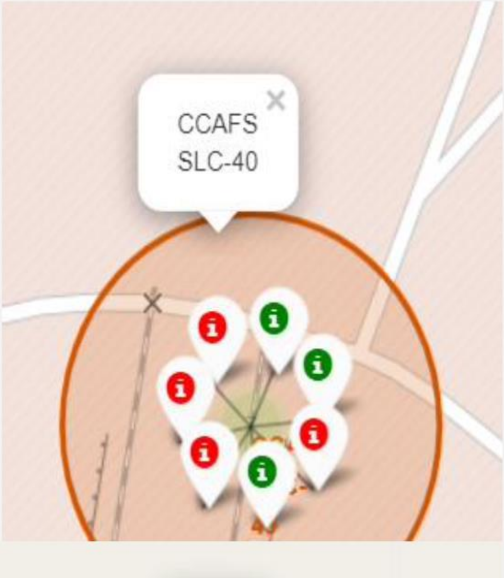
According to the map :
The SpaceX launch sites are in the united states of America coast.
Florida and California

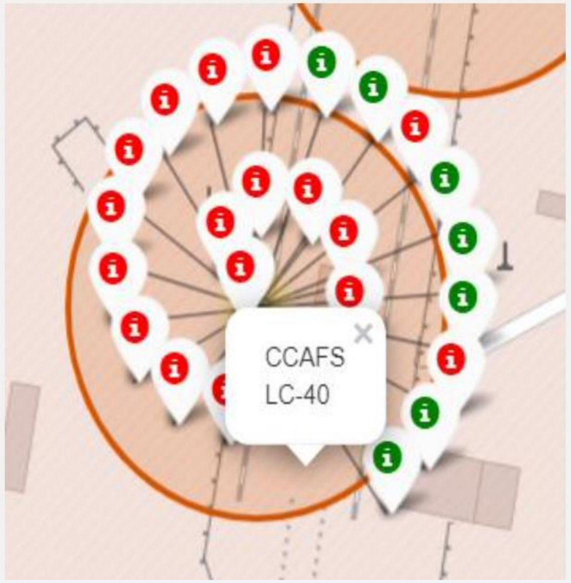
<Folium Map Screenshot 2>



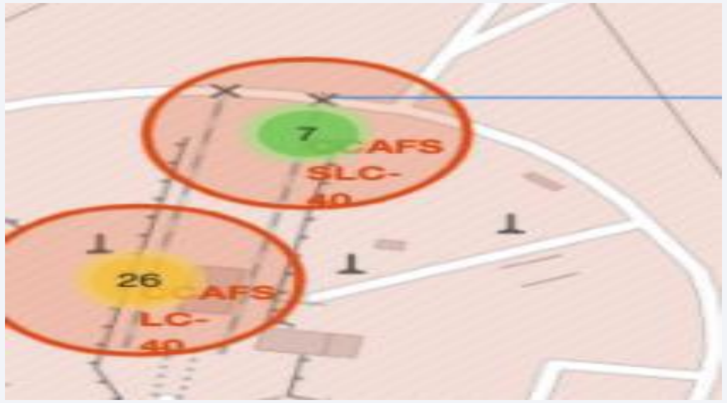


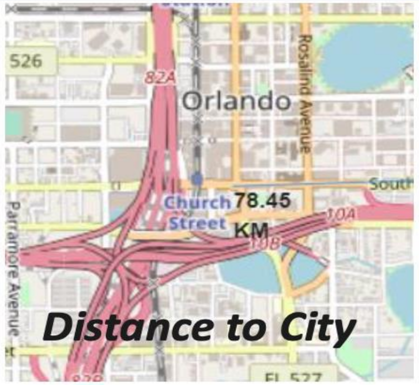
Green Marker shows successful launches and Red Marker shows Failures

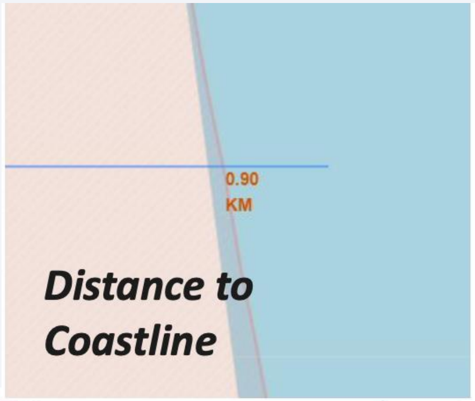


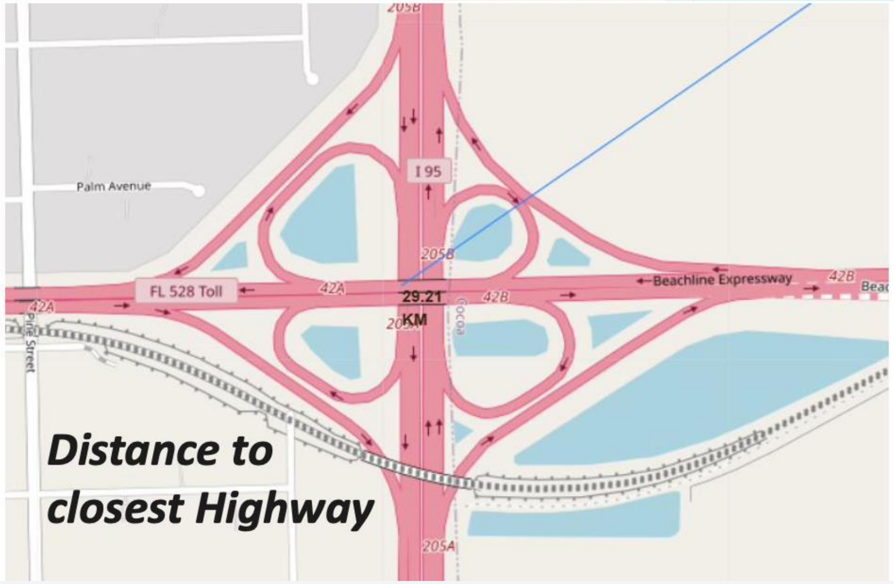


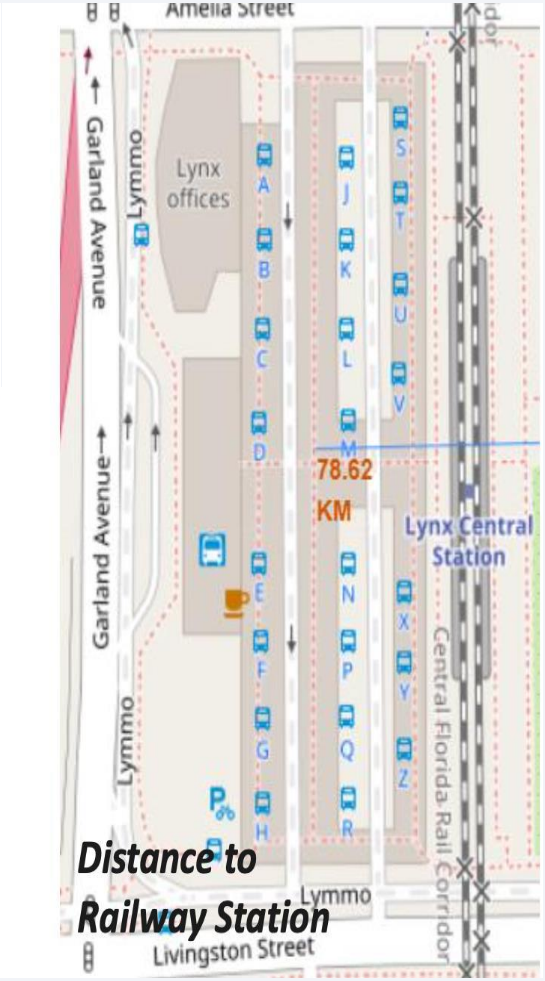
<Folium Map Screenshot 3>



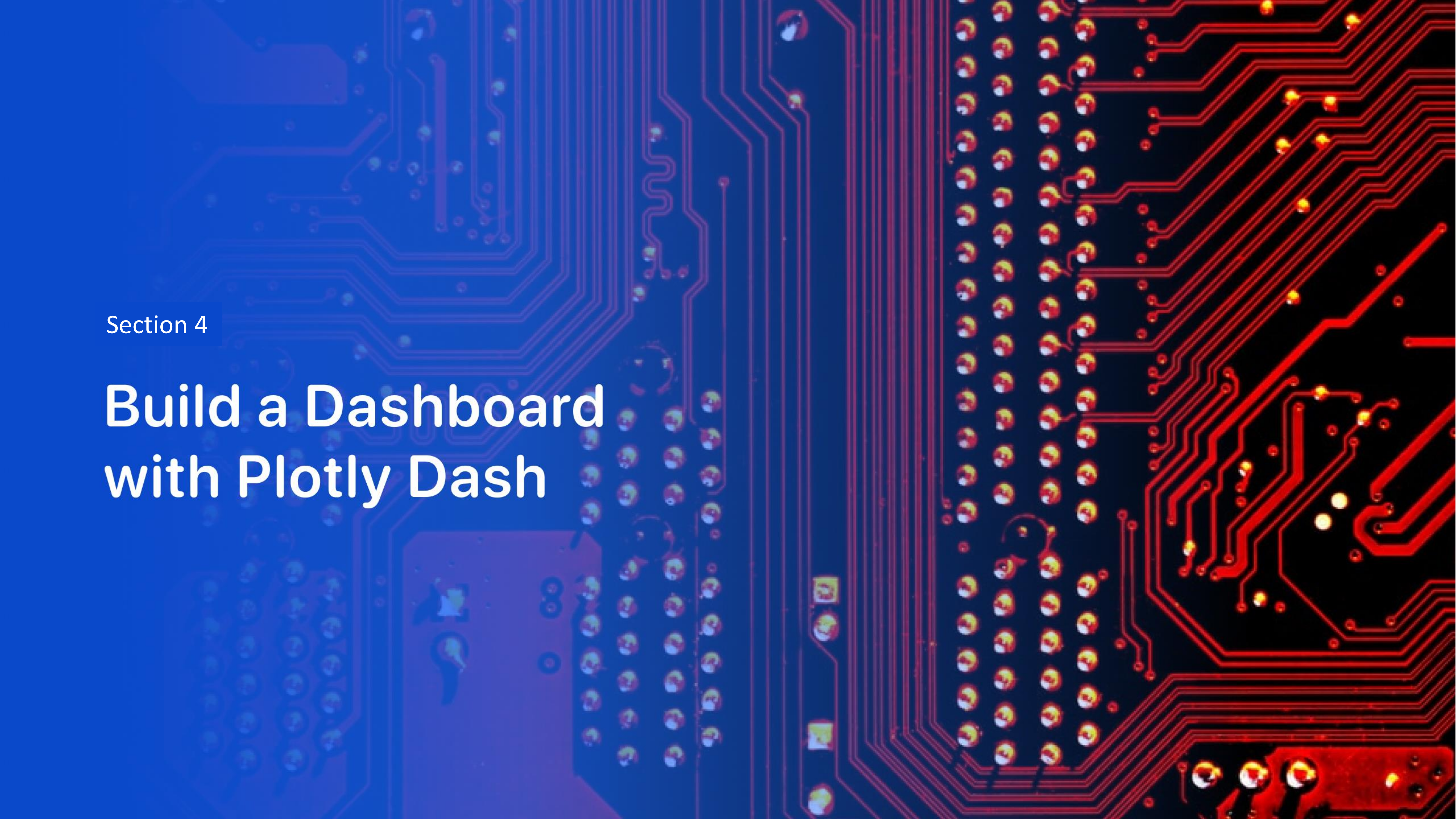








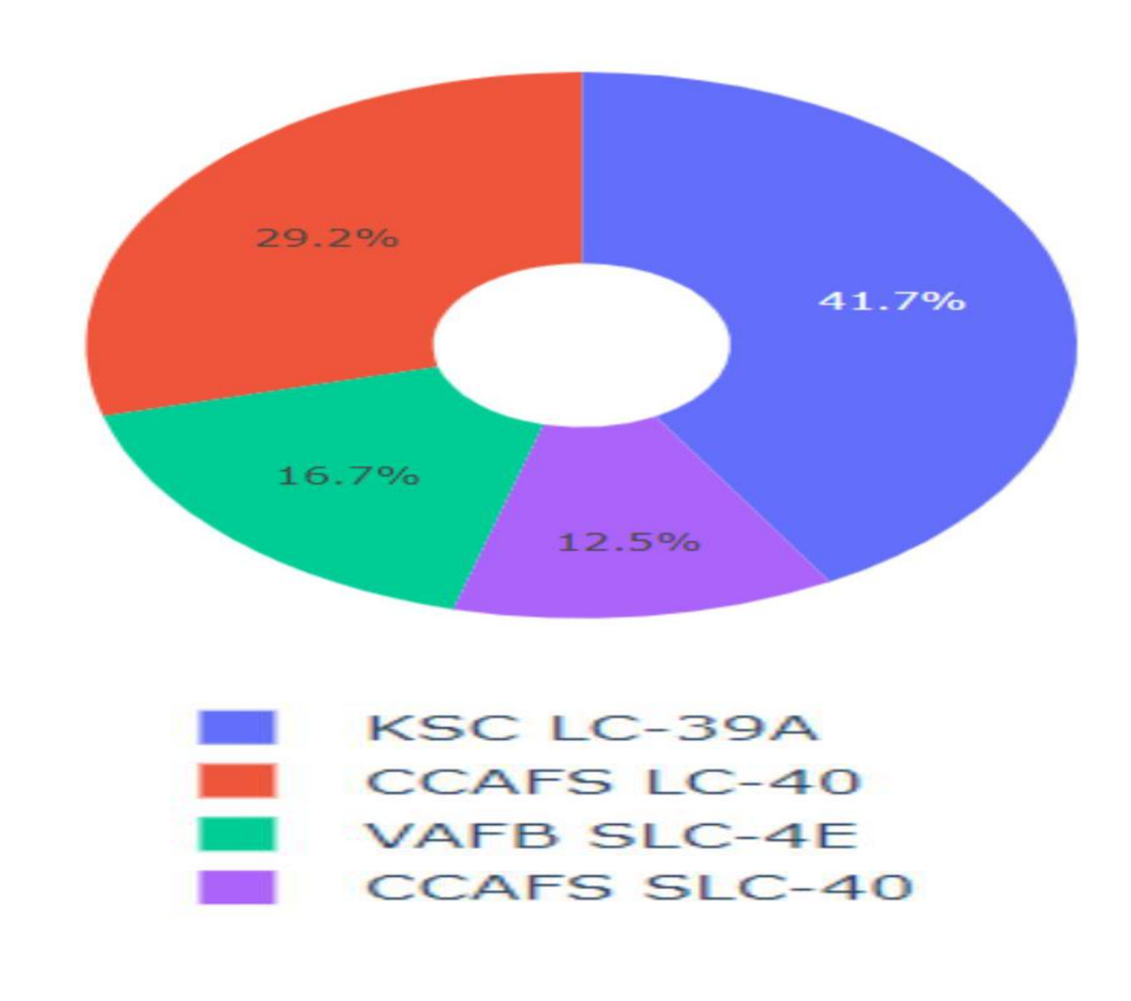
Are launch sites in close proximity to railways? No
Are launch sites in close proximity to highways? No
Are launch sites in close proximity to coastline? Yes
Do launch sites keep certain distance away from cities?
yes



Section 4

Build a Dashboard with Plotly Dash

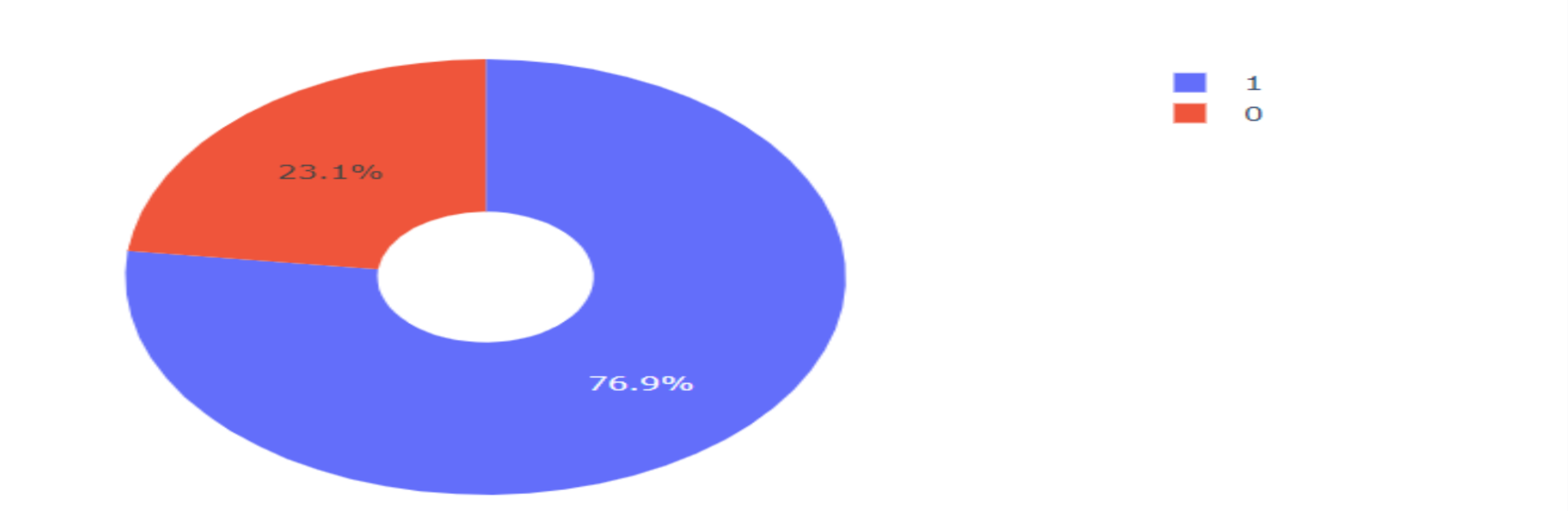
<Dashboard Screenshot 1>



As we can see from the pie chart

KSC LC-39A had the most successful launches from all the sites

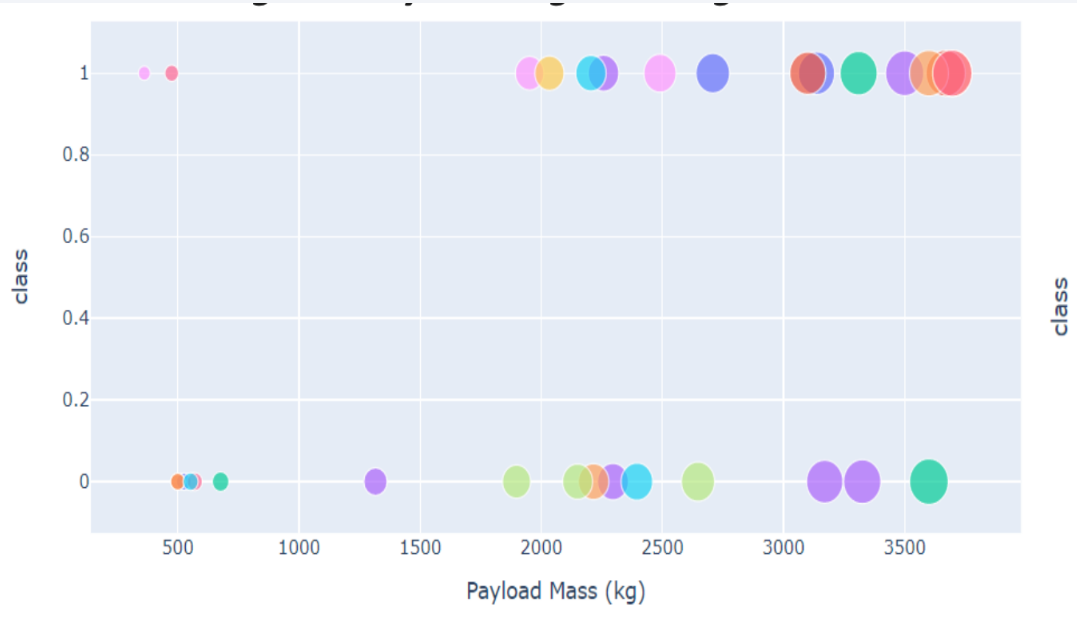
<Dashboard Screenshot 2>



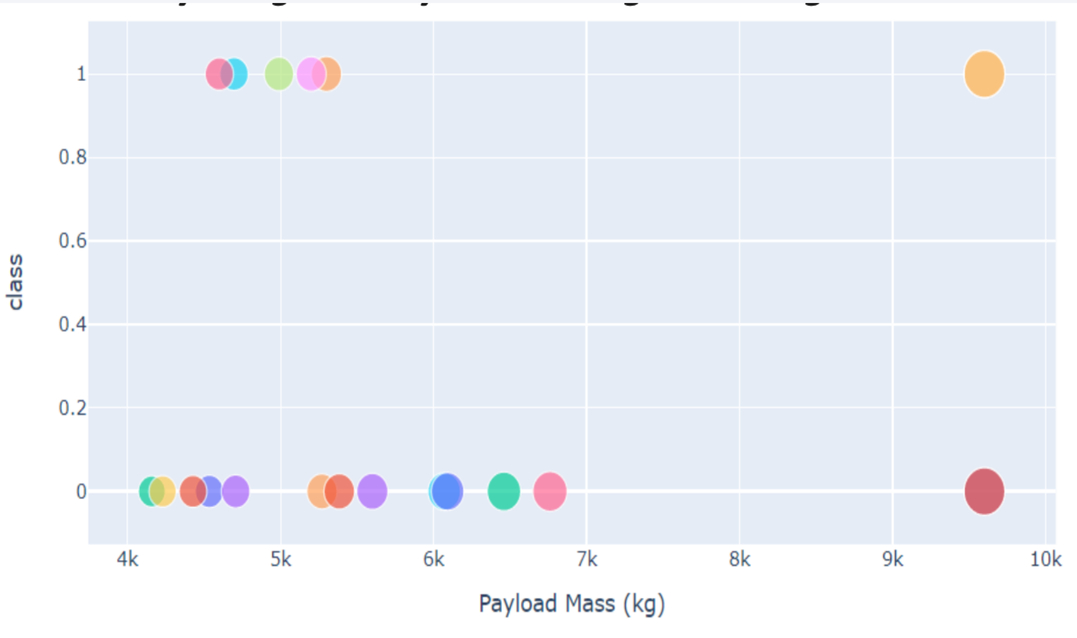
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>

Low weighted payload 0kg –4000kg



Heavy weighted payload 4000kg –10000kg



As we can see the success rates for low weighted payloads is higher than the heavy weighted payloads

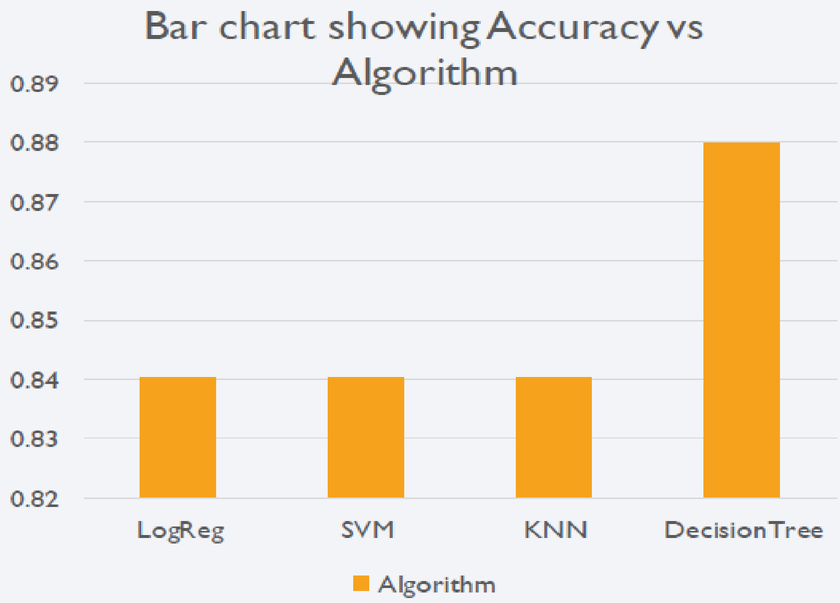
Section 5

Predictive Analysis (Classification)

Classification Accuracy

The best perform accuracy is Decision tree with a score of 0.88. We trained four model and none of them had anything less than 0.83

Algorithm	Accuracy	Accuracy on test data	
Logistic Regression	0.8464	0.83334	
SVM	0.84821	0.83334	
KNN	0.8482	0.83334	
Decision Tree	0.8892	0.7222	



Confusion Matrix

All models used have the same confusion matrix.

Accuracy: (TP+TN)/total =(12+3)/18=0.8333

Misclassification Rate(FP+FN)/total=(3+0)/18=0.1667

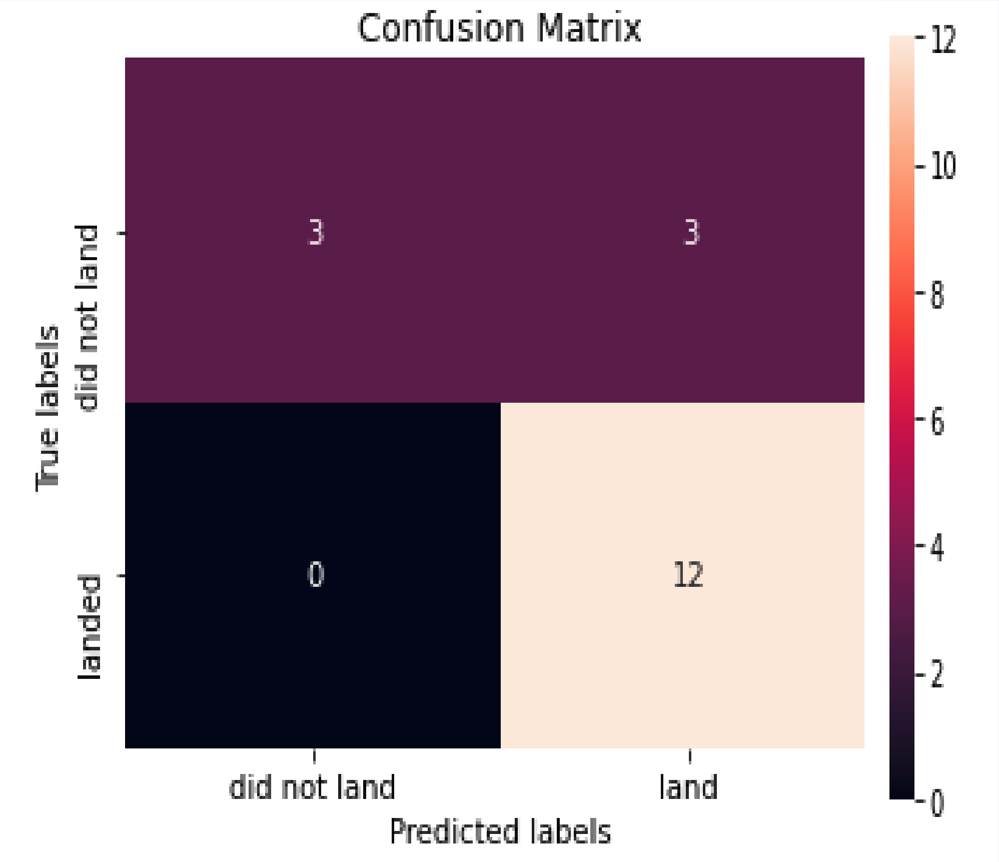
True Positive Rate: TP/Actual Yes = 12/12 =1

False Positive Rate: FP/Actual No=3/6=0.5

True Negative Rate: TN/Actual No=3/6=0.5

Precision: TP/Predicted Yes = 12/15=0.8

Prevalence: Actual Yes/total=12/18 = 0.6667



Conclusions

- 46 Orbits ES L1, GEO, HEO, SSO has highest success rates
- The Success rates for SpaceX launches has been increasing relatively with time, they will eventually perfect the launches.
- KSC LC_39A had the most successful launches from all the sites
- Low weighted payloads perform better than the heavier payloads
- The tree classifier Algorithm is the best machine learning model for this datatset

Appendix

- Interactive plotly
- Python Anywhere
- Folium Measure Control Plugin Tool
- Basic Decision Tree Construction
- IBM Cognos Visualization Tool

Thank you!