IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ahmed Lotfi Alqnatri
2021-10-09

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)

Summary of all results

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

Introduction

- Project background and context

  We anticipated if the Falcon 9 first stage will successfully land. SpaceX promotes Falcon 9 rocket launches on its website for 62 million dollars; other companies charge up to 165 million dollars each launch. Much of the savings are due to SpaceX's ability to reuse the first stage. As a result, if we can predict whether the first stage will land, we can calculate the cost of a launch This data can be useful if another business wants to compete with SpaceX for a rocket launch.

- Problems you want to find answers

  - What factors impact whether the rocket lands successfully?
  - The influence that each association with certain rocket factors will have on the success rate of a successful landing.
  - What parameters must SpaceX meet in order to get the greatest outcomes and assure the highest rate of rocket landing success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Wikipedia using soup for Web Scrapping
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning
  - Dropping irrelevant (not needed) columns
- 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

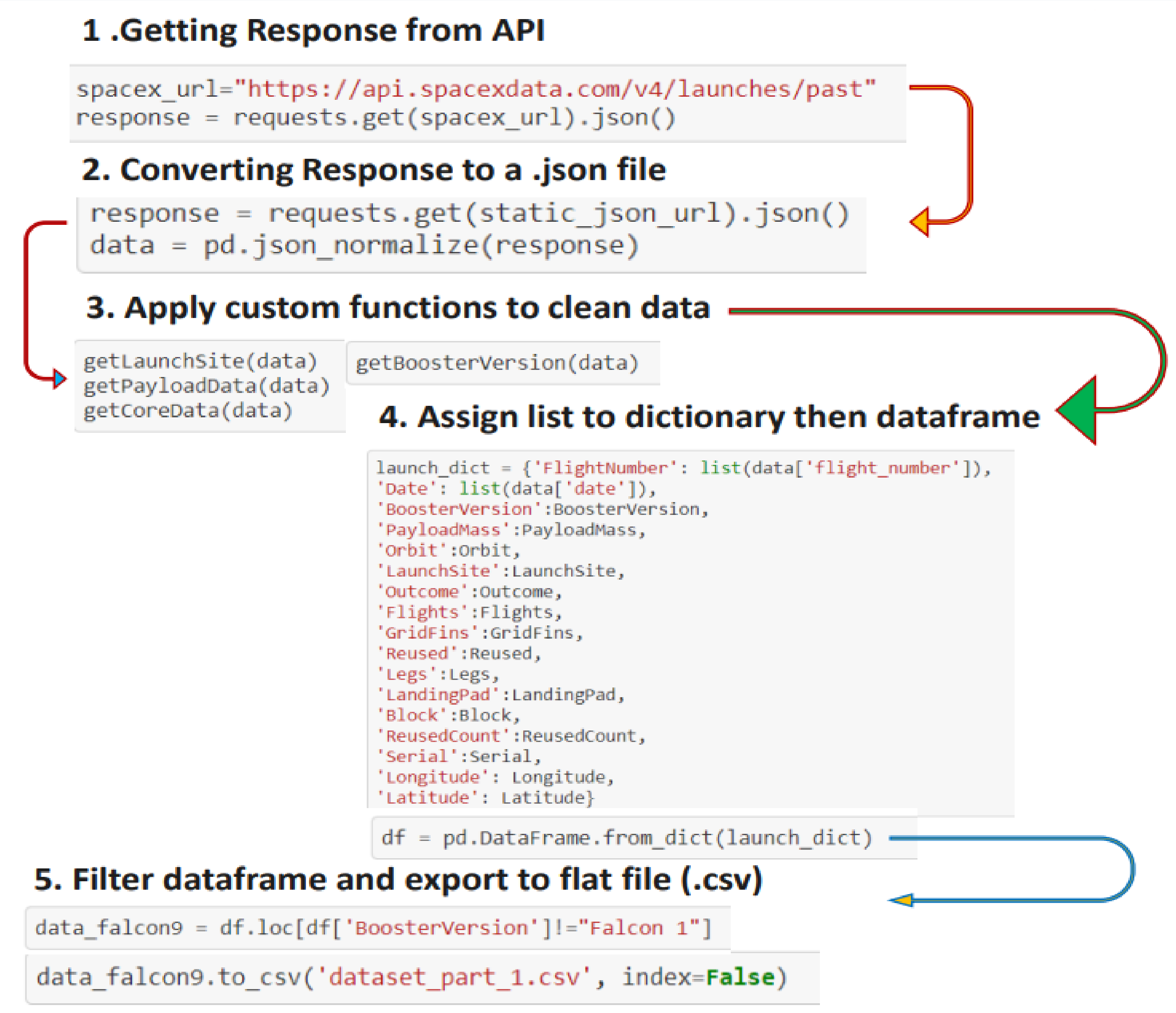
The following datasets were gathered by:

- ❑ We worked with SpaceX launch data obtained via the SpaceX REST API.
- ❑ This API will provide us with information regarding launches, such as the rocket utilized, the cargo delivered, the launch specs, the landing specifications, and the landing outcome.
- ❑ Our objective is to utilize this information to forecast whether or not SpaceX will attempt to land a rocket.
- ❑ The URL for the SpaceX REST API endpoints begins with api.spacexdata.com/v4/.
- ❑ Web scraping Wikipedia using BeautifulSoup is another common data source for collecting Falcon 9 Launch data.

Data Collection – SpaceX API

- Presenting data collection with SpaceX REST calls using key phrases and flowcharts

- GitHub URL



1 .Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filter dataframe and export to flat file (.csv)

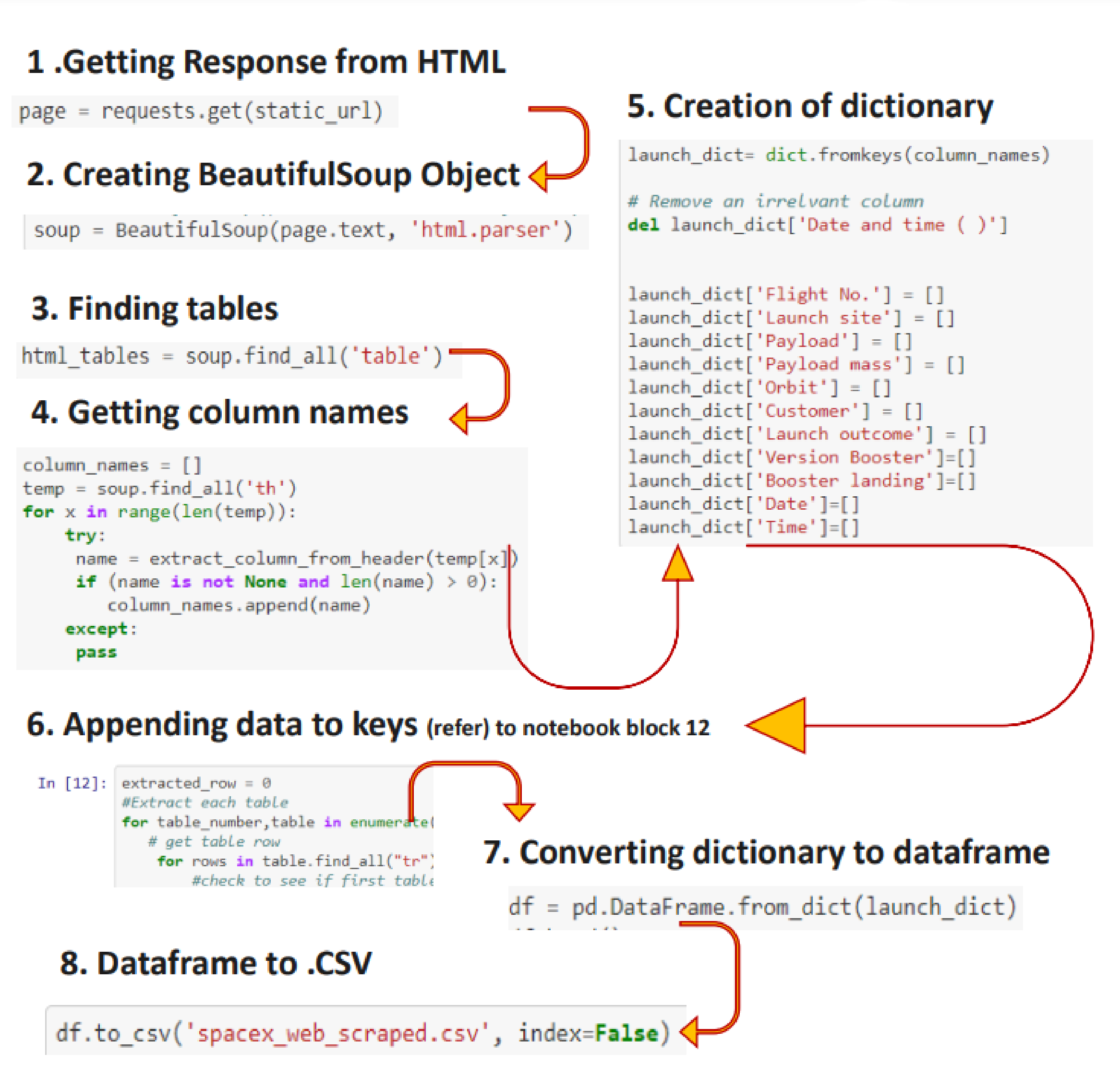
```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Presenting web scraping process using key phrases and flowcharts

- GitHub URL



1 .Getting Response from HTML

```
page = requests.get(static_url)
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

```
html_tables = soup.find_all('table')
```

4. Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']


launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
             # get table row
              for rows in table.find_all("tr")
                  #check to see if first table
```

7. Converting dictionary to dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Dataframe to .CSV

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

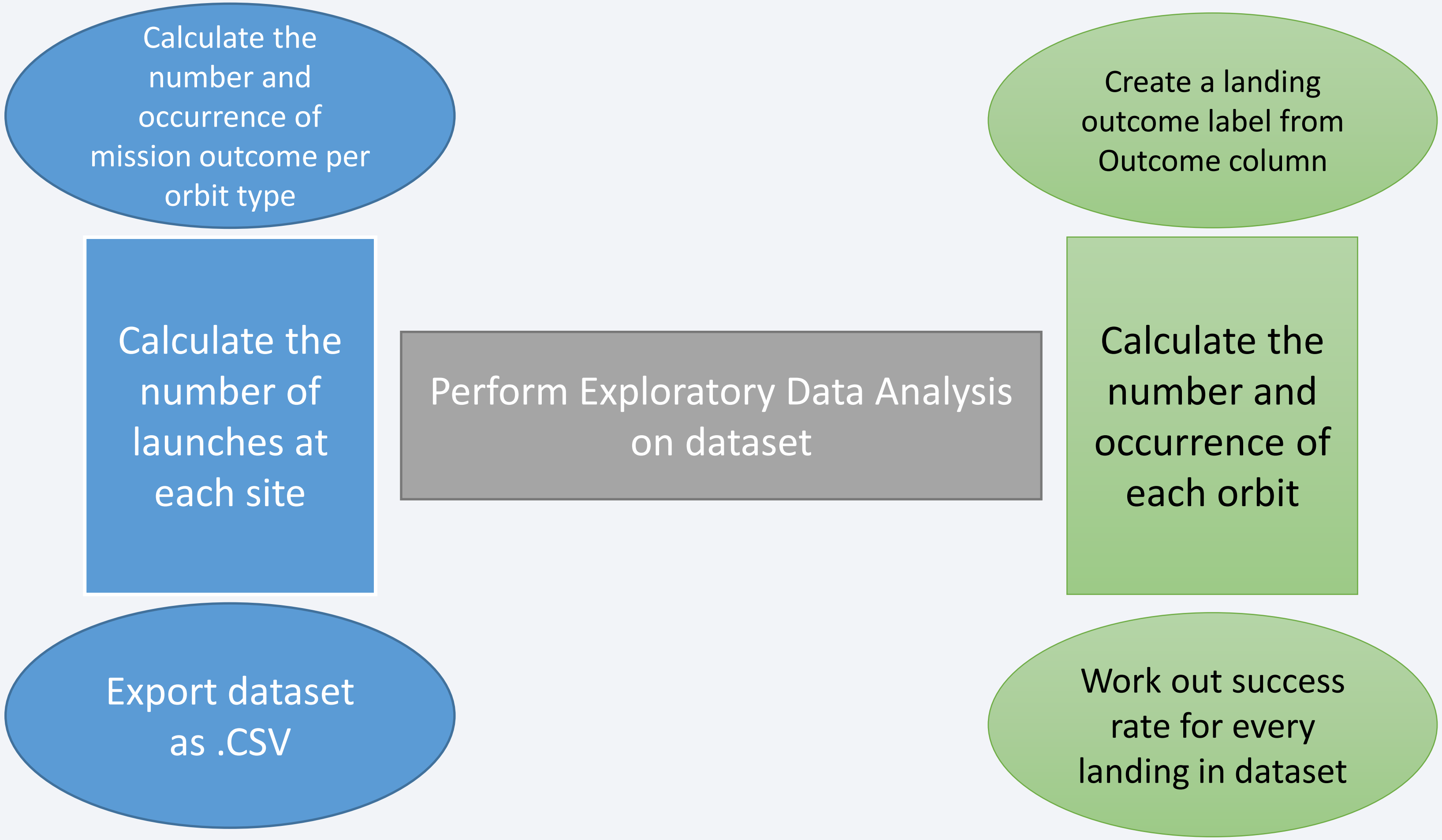
Introduction

There are numerous distinct situations in the data set when the booster did not successfully land. A landing may have been attempted but failed due to an accident for example, True Ocean indicates that the mission outcome was successfully landed to a specific location of the ocean, whereas False Ocean indicates that the mission outcome was unsuccessfully landed to a specific region of the ocean.

True RTLS indicates that the mission resulted in a successful landing on a ground pad. False RTLS indicates that the mission was unsuccessfully landed on a ground pad. True ASDS indicates that the mission result was successfully landed on a drone ship. False ASDS indicates that the mission outcome was a failed landing on a drone ship.

We primarily translate such results into Training Labels, with 1 indicating a successful landing and 0 indicating a failed landing.

Data Wrangling



Calculate the number and occurrence of mission outcome per orbit type

Calculate the number of launches at each site

Export dataset as .CSV

Perform Exploratory Data Analysis on dataset

Create a landing outcome label from Outcome column

Calculate the number and occurrence of each orbit

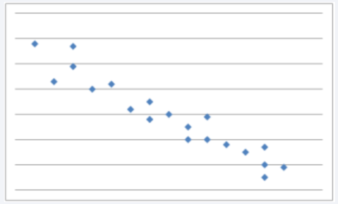
Work out success rate for every landing in dataset

GitHub URL

EDA with Data Visualization

Scatter Graphs being drawn:

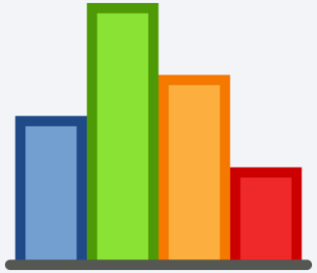
- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit
- Type Orbit VS. Payload Mass



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.

Bar Graph being drawn:

- Mean VS. Orbit



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.

Line Graph being drawn:

- Success Rate VS. Year

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

GitHub URL

EDA with SQL

Performed SQL queries to gather information about the dataset.

For example, of some questions we were asked about the data we needed information about. Which we are using SQL queries to get the answers in the dataset :

- 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.

GitHub URL

Build an Interactive Map with Folium

- To make an interactive map out of the Launch Data. We used the Latitude and Longitude Coordinates for each launch site to create a Circle Marker with the name of the launch site labeled around it.

- Classes were assigned to the data frame launch outcomes(failures, successes).In a MarkerCluster, 0 and 1 with Green and Red markers on the map ()

- We computed the distance from the Launch Site to the Launch Site using Haversine's formula. Numerous landmarks to discover distinct patterns concerning what is happening in the vicinity of the Launch Site to Patterns are measured. Lines are drawn on the map to measure distance to landmarks

Example of some trends in which the Launch Site is situated in.

- 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

GitHub URL

Build a Dashboard with Plotly Dash

Graphs

Pie Chart

- Pie Chart showing the total launches by a certain site/all sites
- display relative proportions of multiple classes of data.
- size of the circle can be made proportional to the total quantity it represents.

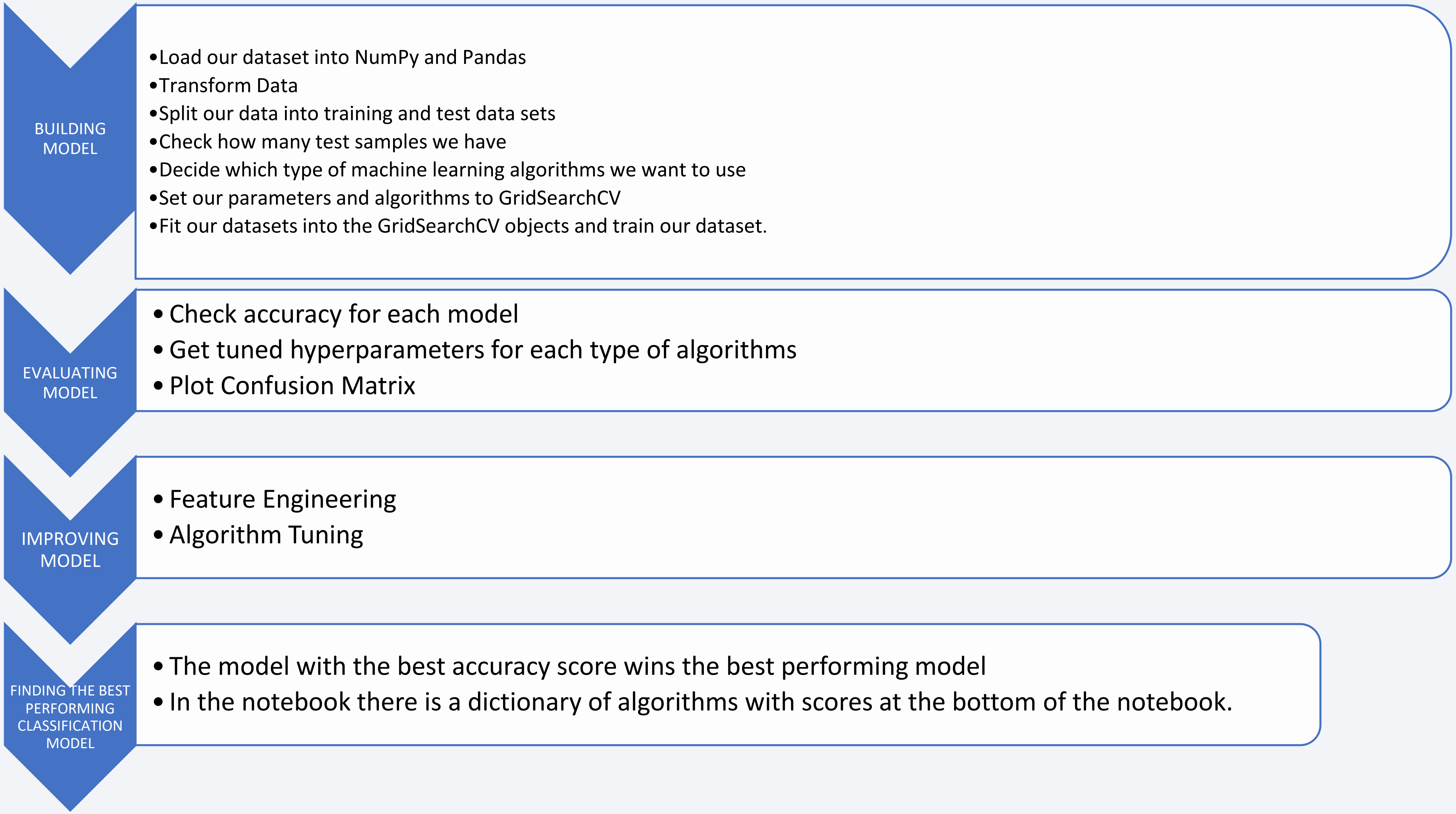
Scatter Graph

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

- It shows the relationship between two variables.
- It is the best method to show you a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

GitHub URL

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 GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

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
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

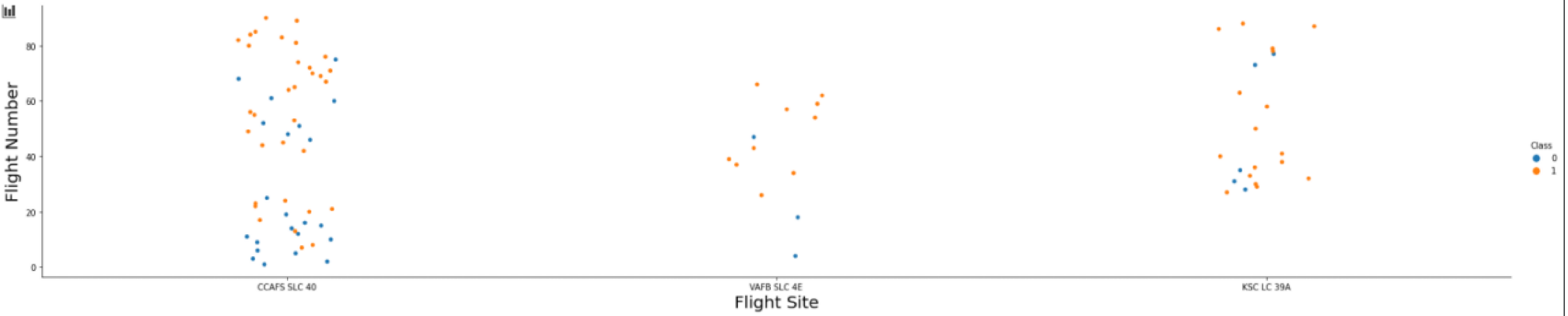
GitHub URL

Results

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

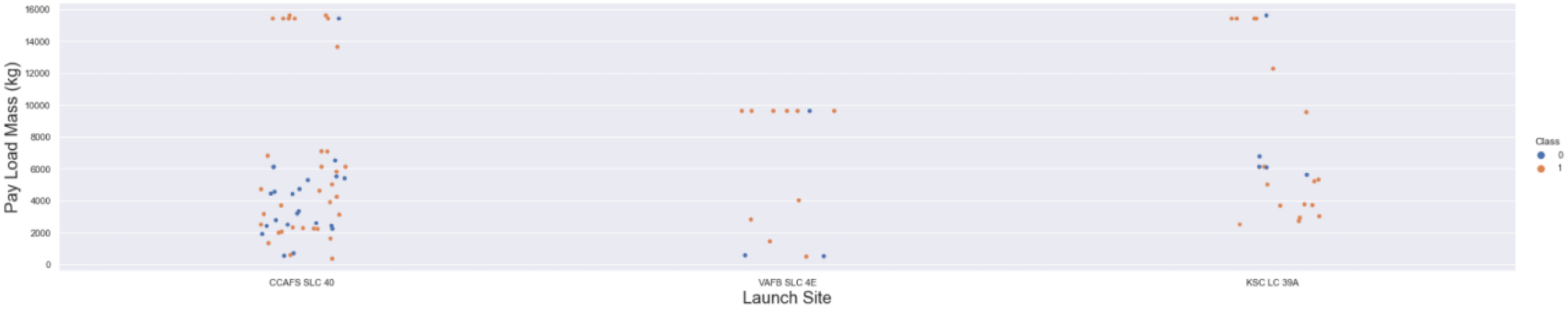
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- The larger the number of flights at a launch location, the higher the success rate for that launch site.

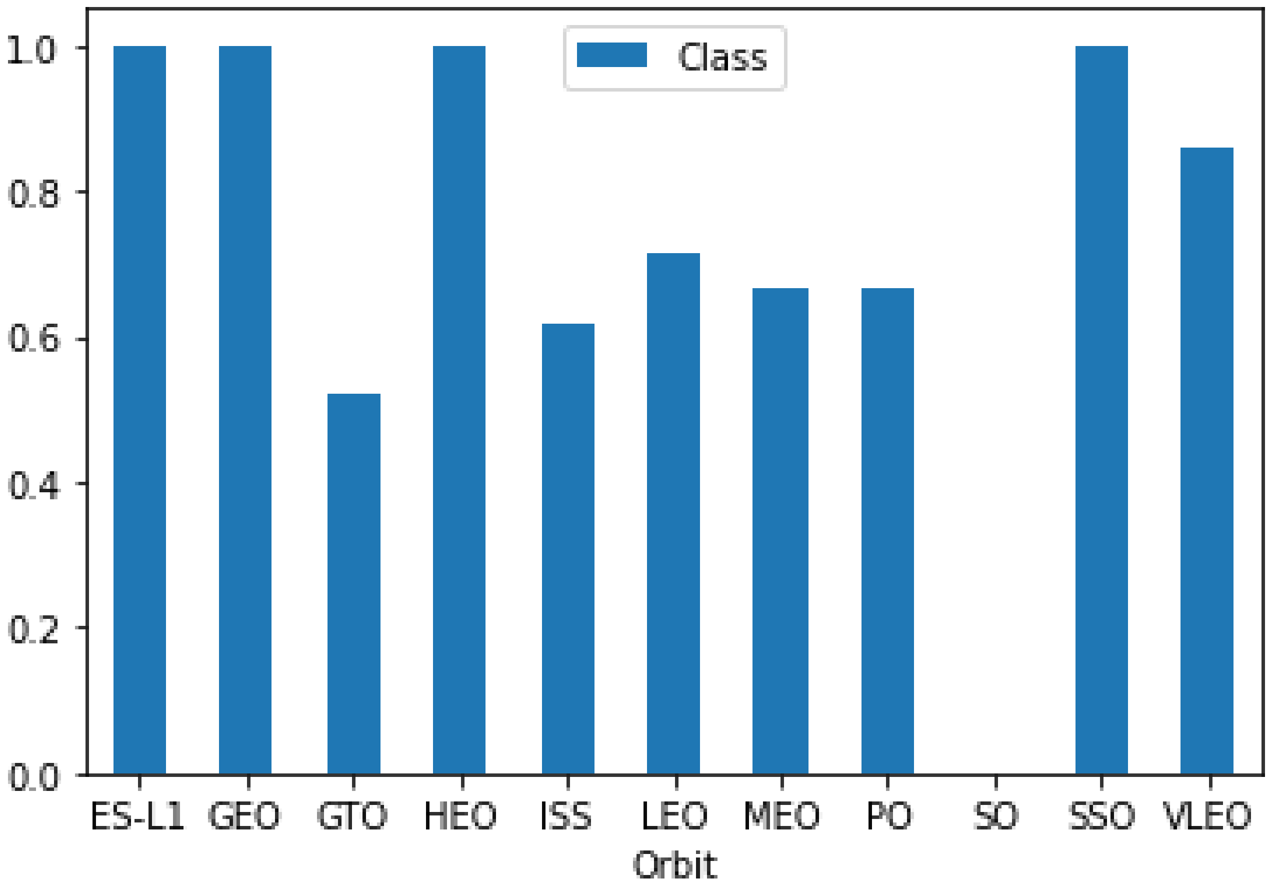


Payload vs. Launch Site

- The bigger the payload tonnage for Launch Site CCAFS SLC 40, the better the Rocket's success rate.

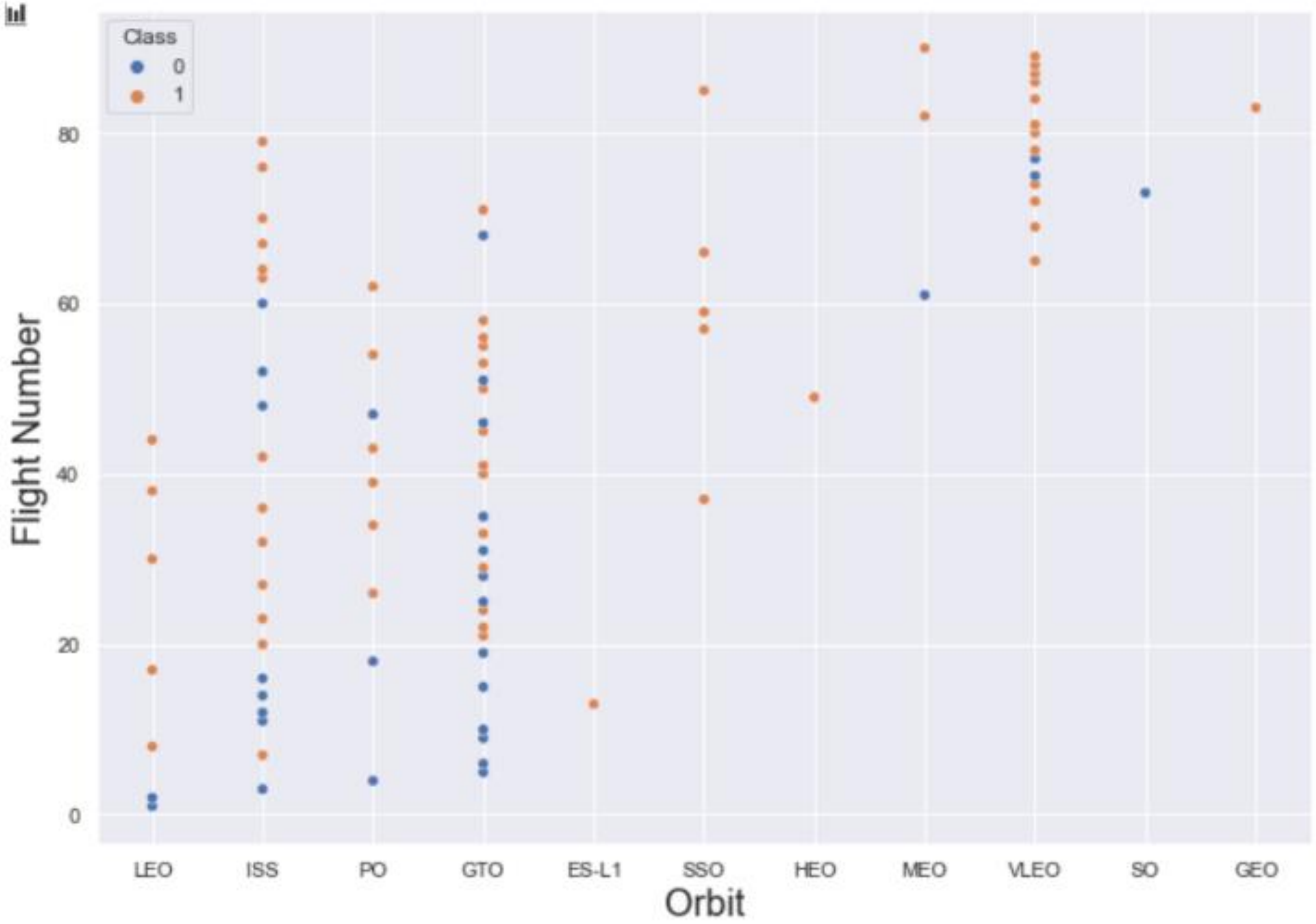
Success Rate vs. Orbit Type

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



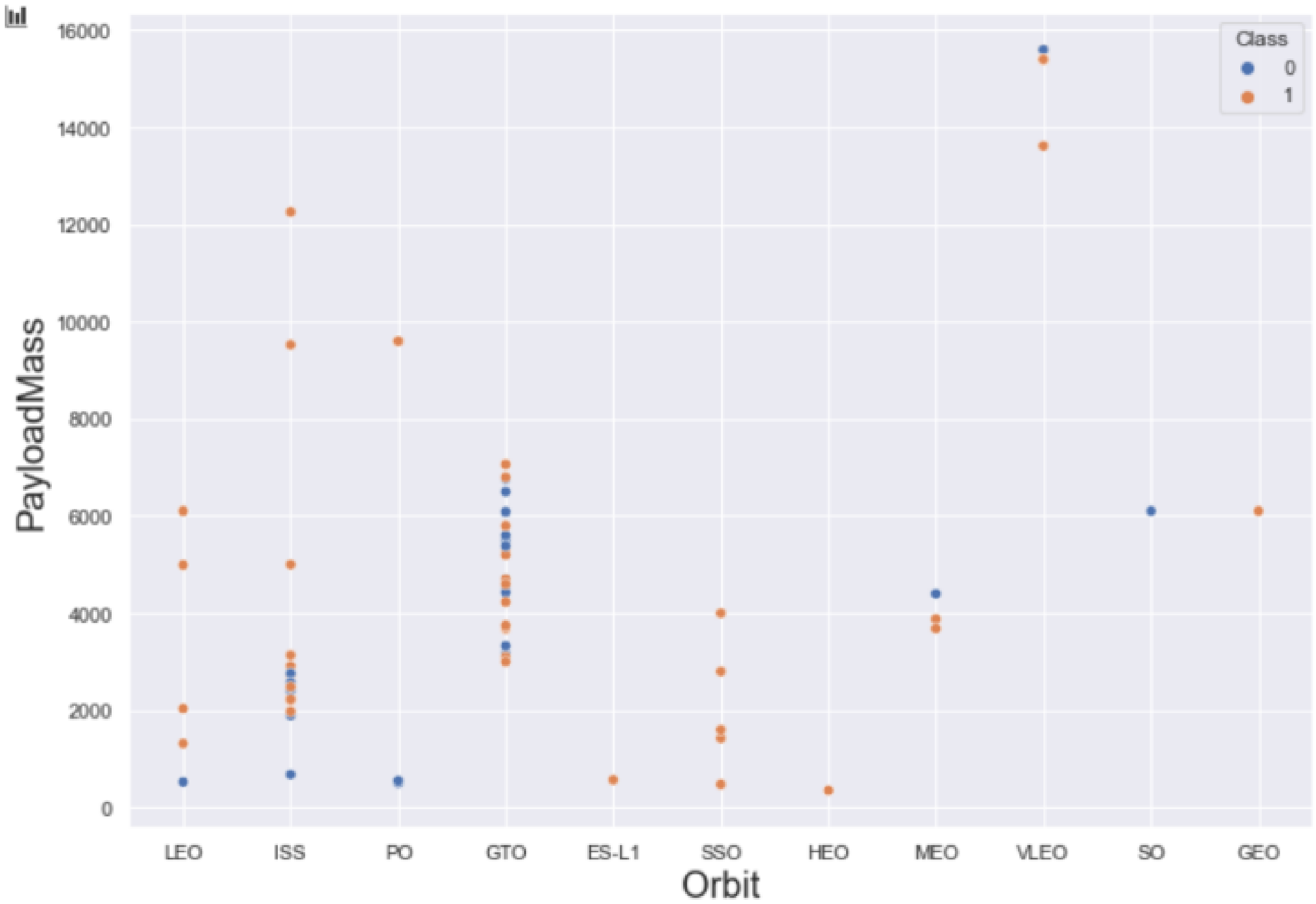
Flight Number vs. Orbit Type

- You can see that in LEO orbit, success appears to be proportional to the number of flights; but, in GTO orbit, there appears to be no relationship between flight number.



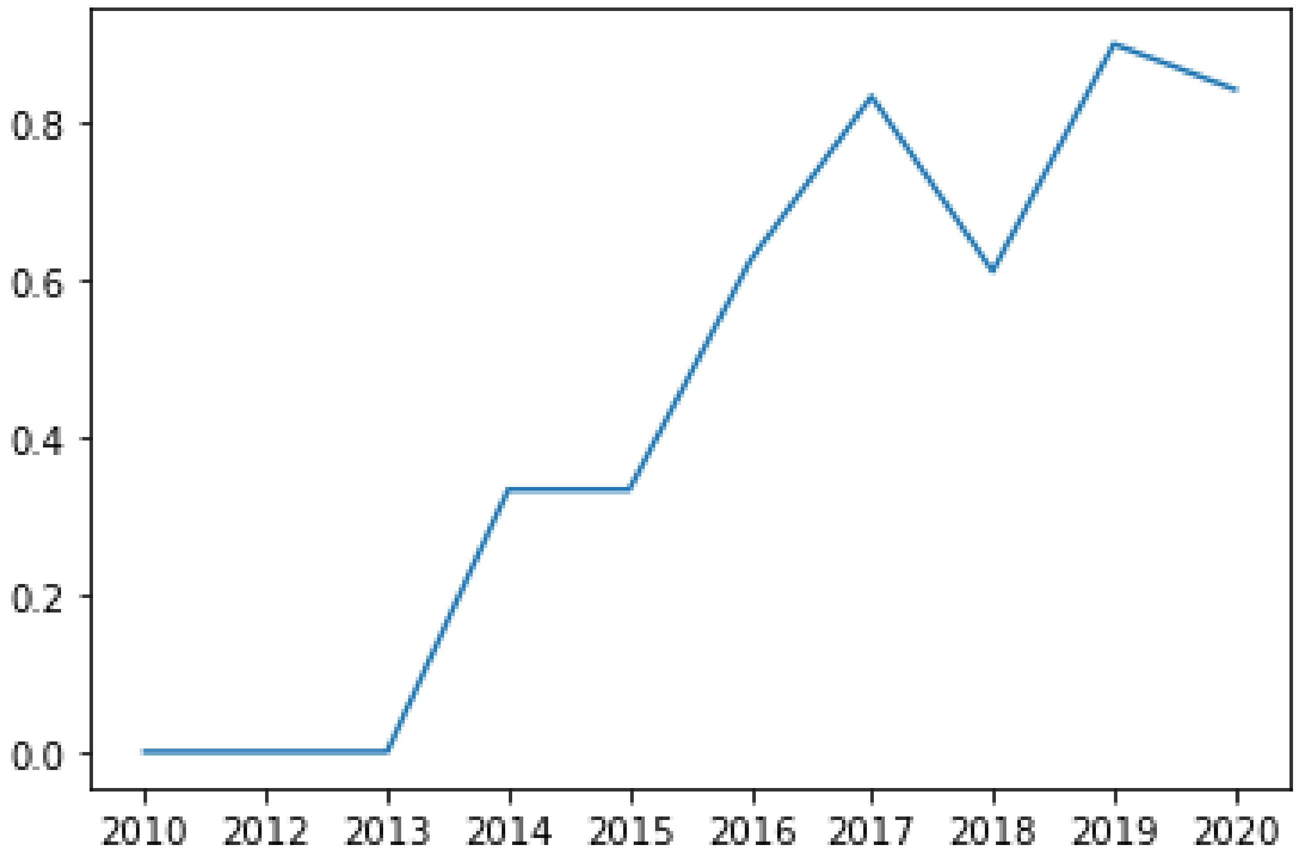
Payload vs. Orbit Type

- Heavy payloads have a detrimental impact on GTO orbits but a favorable impact on GTO and Polar LEO (ISS) orbits.



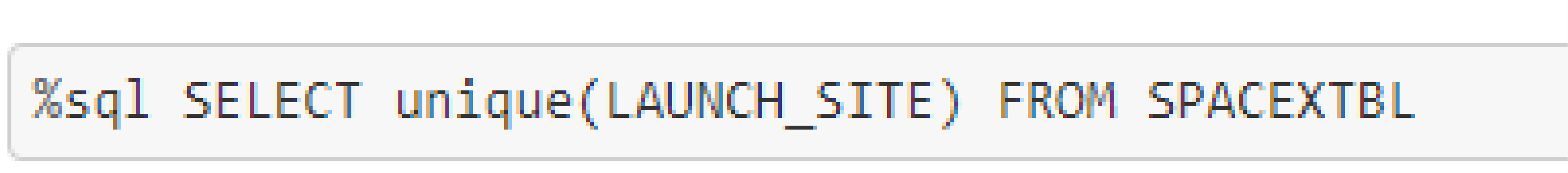
Launch Success Yearly Trend

- you can observe that the success rate since 2013 kept increasing till 2020



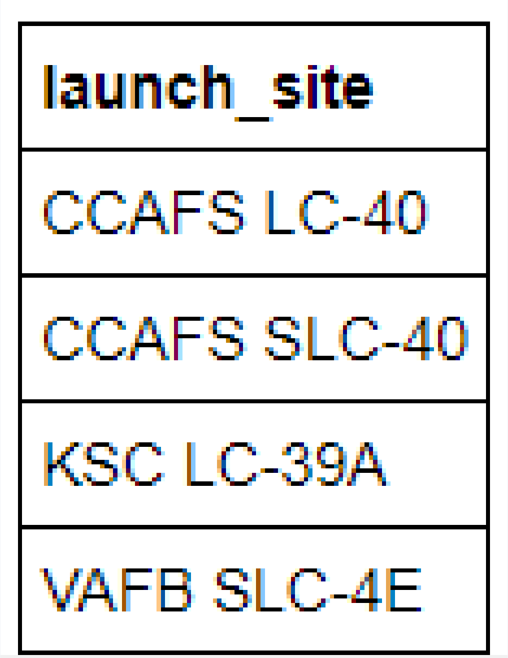
Section 3
EDA with SQL

All Launch Site Names



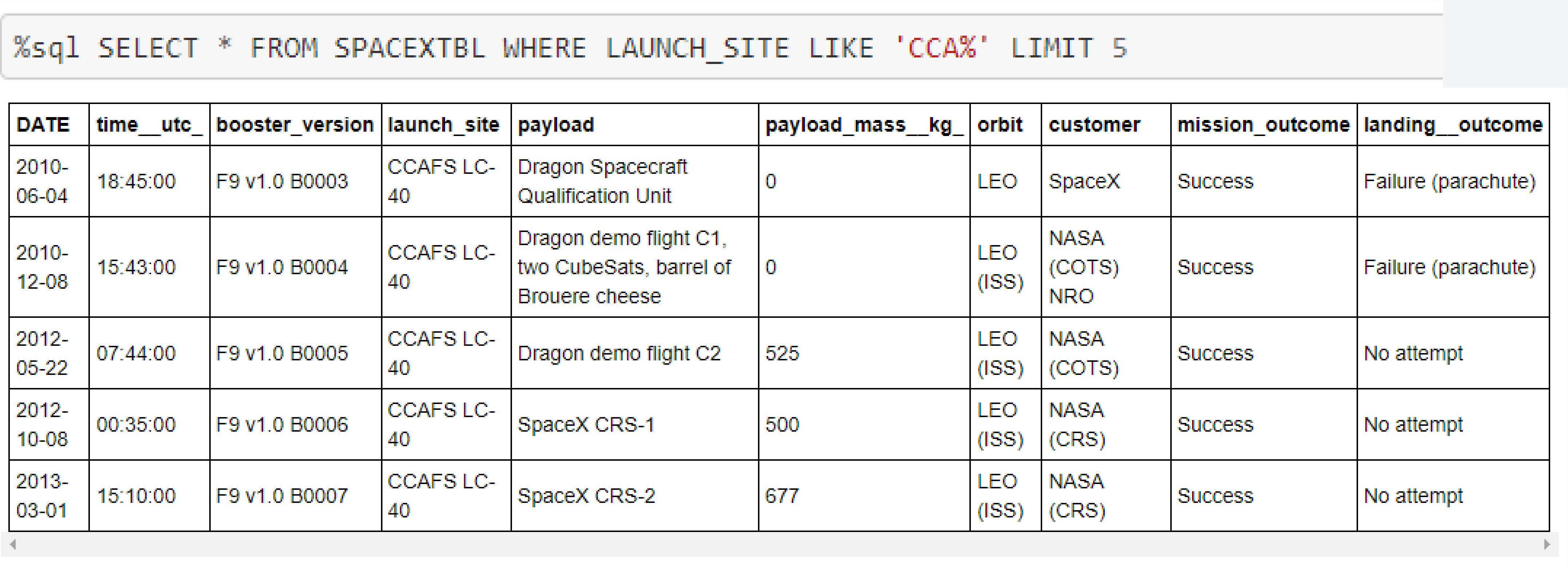
```
%sql SELECT unique(LAUNCH_SITE) FROM SPACEXTBL
```

- Using the word UNIQUE in the query means that it will only show Unique values in the Launch_Site column from SPACEXTBL



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

Launch Site Names Begin with 'CCA'

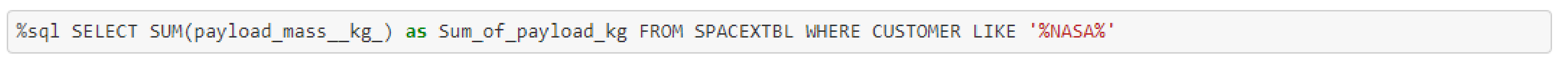


```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

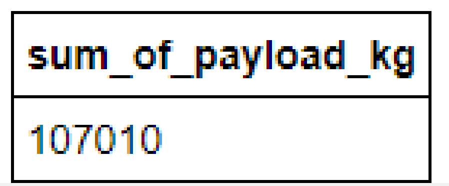
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Using the Keyword LIKE and % in SQL means to find a string that starts with the value in ‘CCA’

Total Payload Mass



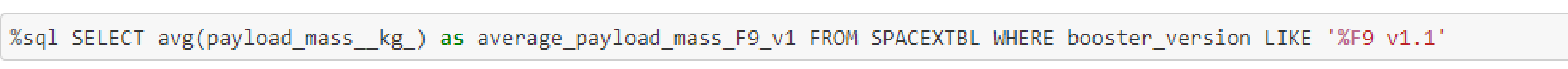
```
%sql SELECT SUM(payload_mass__kg_) as Sum_of_payload_kg FROM SPACEXTBL WHERE CUSTOMER LIKE '%NASA%'
```



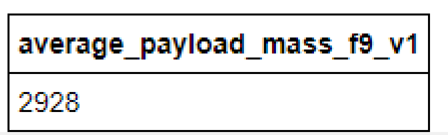
sum_of_payload_kg
107010

- Using the function SUM summates the total in the column PAYLOAD_MASS_KG_ The WHERE clause and LIKE filters the dataset to only perform calculations on Customer that has NASA in it

Average Payload Mass by F9 v1.1



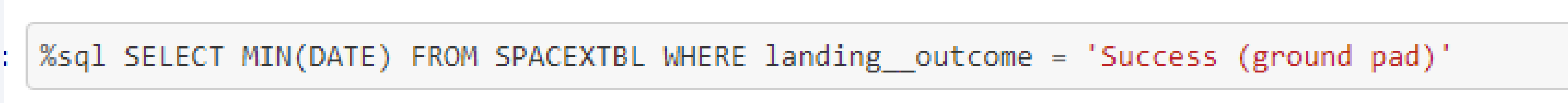
```
%sql SELECT avg(payload_mass__kg_) as average_payload_mass_F9_v1 FROM SPACEXTBL WHERE booster_version LIKE '%F9 v1.1'
```



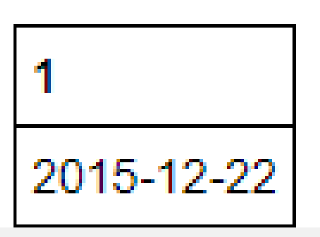
average_payload_mass_f9_v1
2928

- 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_version F9 v1.1

First Successful Ground Landing Date



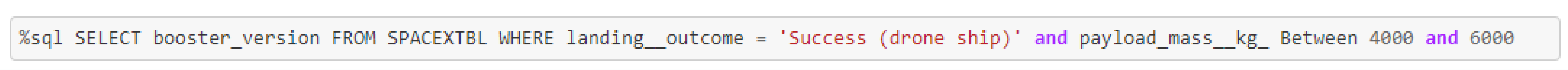
```
: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```



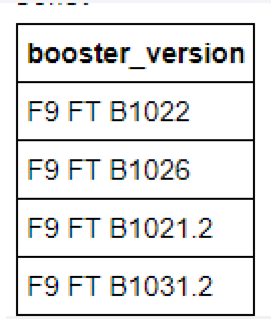
1
2015-12-22

- Using the function MIN works out the minimum date in the column Date
- The WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000



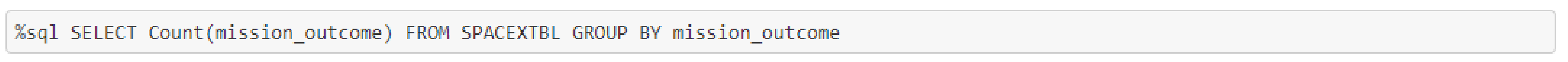
```
%sql SELECT booster_version FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' and payload_mass__kg_ Between 4000 and 6000
```



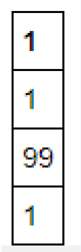
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Selecting only Booster_Version
- The WHERE clause filters the dataset to Landing_Outcome = Success (drone ship)
- The AND clause specifies additional filter conditions Payload_MASS_KG_ > 4000 AND Payload_MASS_KG_ < 6000

Total Number of Successful and Failure Mission Outcomes



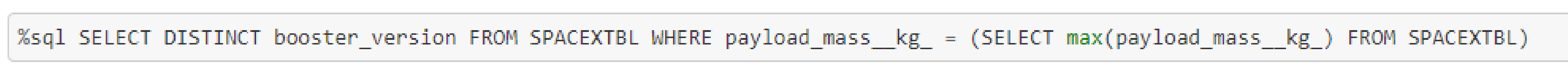
```
%sql SELECT Count(mission_outcome) FROM SPACEXTBL GROUP BY mission_outcome
```



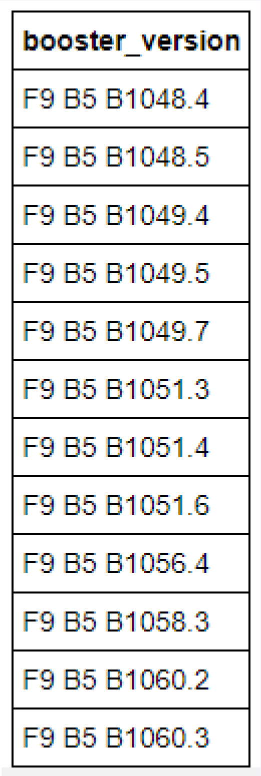
```
1
1
99
1
```

- Using the count function on mission_outcome and then using the GROUP BY function we were able to get 100 succesful and 1 failure mission

Boosters Carried Maximum Payload



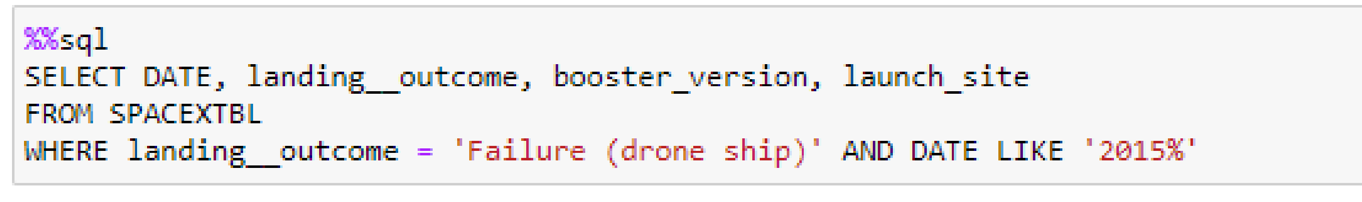
```
%sql SELECT DISTINCT booster_version FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEXTBL)
```



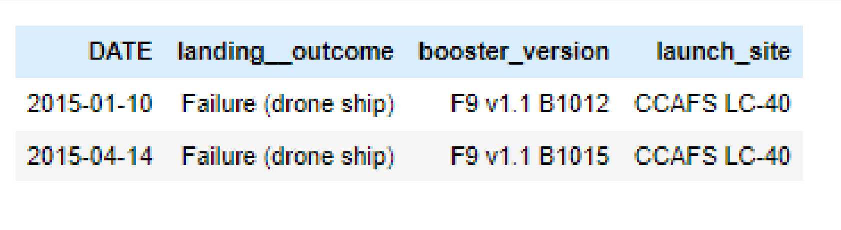
booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

- First we used a subquery to get the mas payload and then we applied the function MAX on it after that we applied another query to get booster_version from the max payload_mass and used the keyword DISTINCT to list a list of the names without dupicates

2015 Launch Records



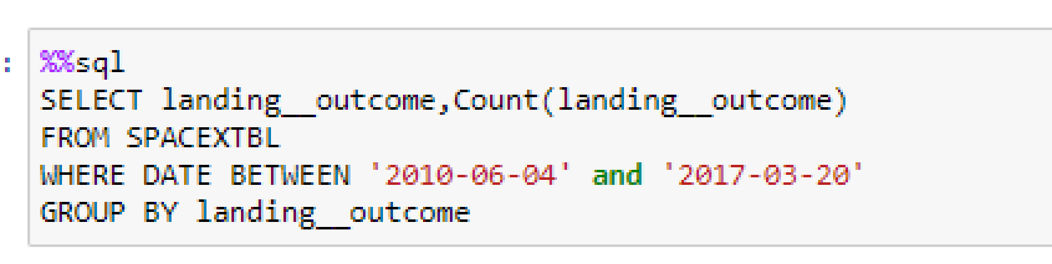
```
%%sql
SELECT DATE, landing__outcome, booster_version, launch_site
FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' AND DATE LIKE '2015%'
```



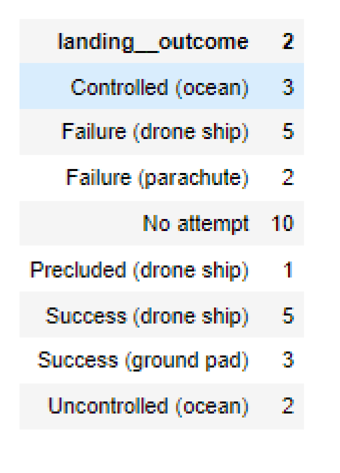
DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- We queried the landing_outcome if it was “Failure (drone ship)” and used the LIKE keywords to filtter on DATE field for 2015 value

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



```
%%sql
SELECT landing__outcome,Count(landing__outcome)
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome
```



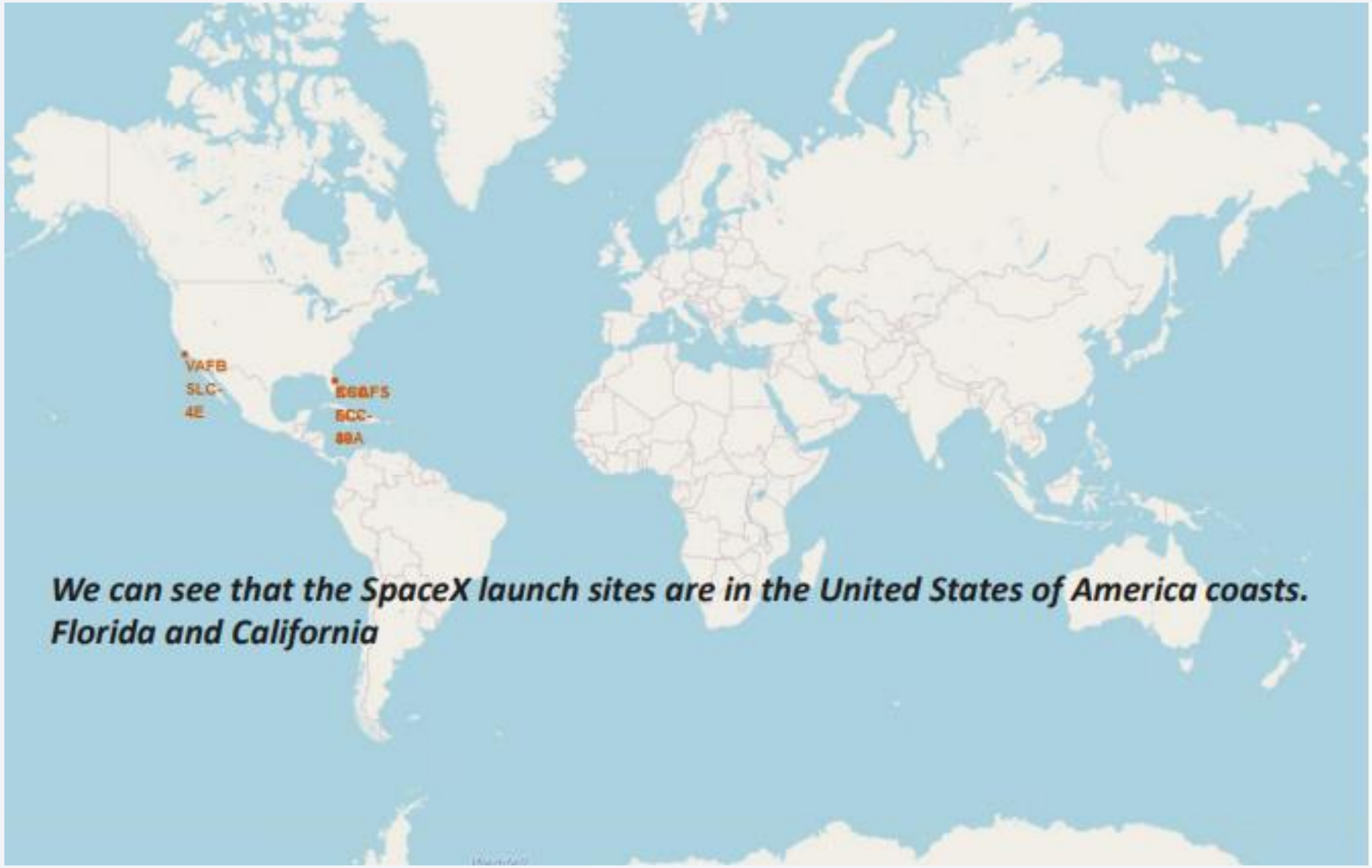
landing__outcome	2
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	10
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	3
Uncontrolled (ocean)	2

- We Queried the landing_outcome and using the Count function to return its count
- Applied WHERE clause on the date and used the BETWEEN function between two given dates and then grouped them by landing_outcome using the GROUP BY function

Section 4

Launch Sites Proximities Analysis

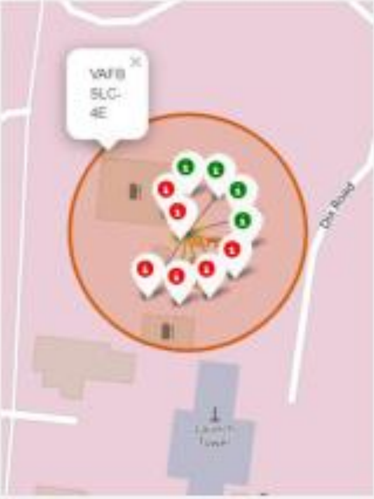
Marked Launches Sites on Global Map



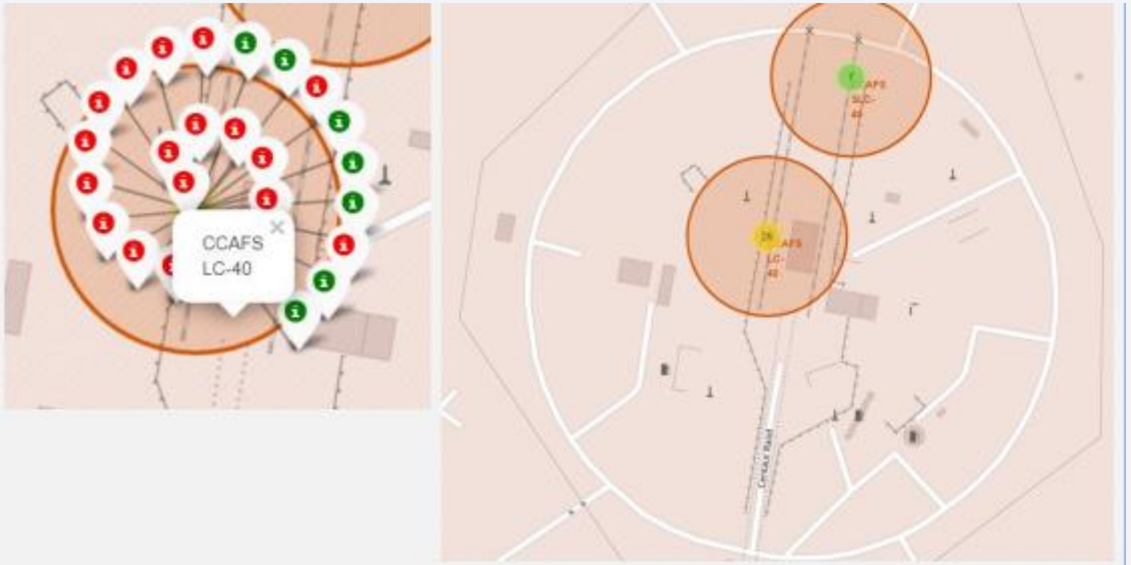
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Color Labelled Markers on launch sites

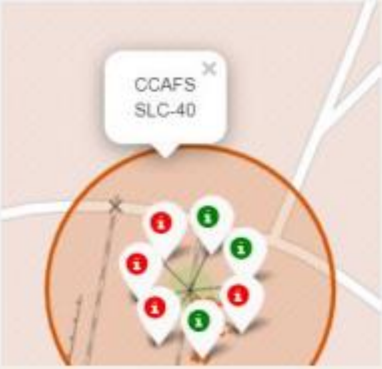
- Green Marker shows successful Launches and Red Marker shows Failed Launches

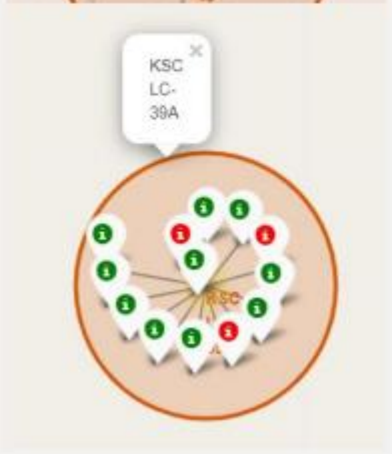


California Launch Site

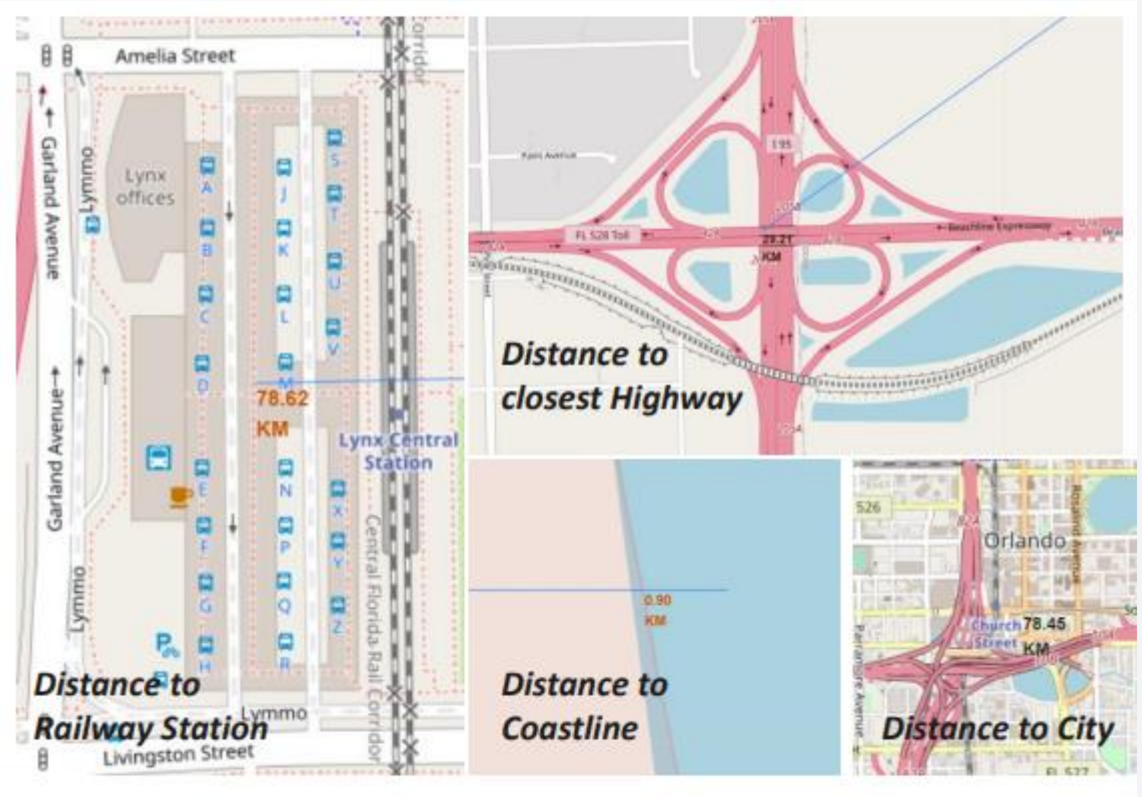


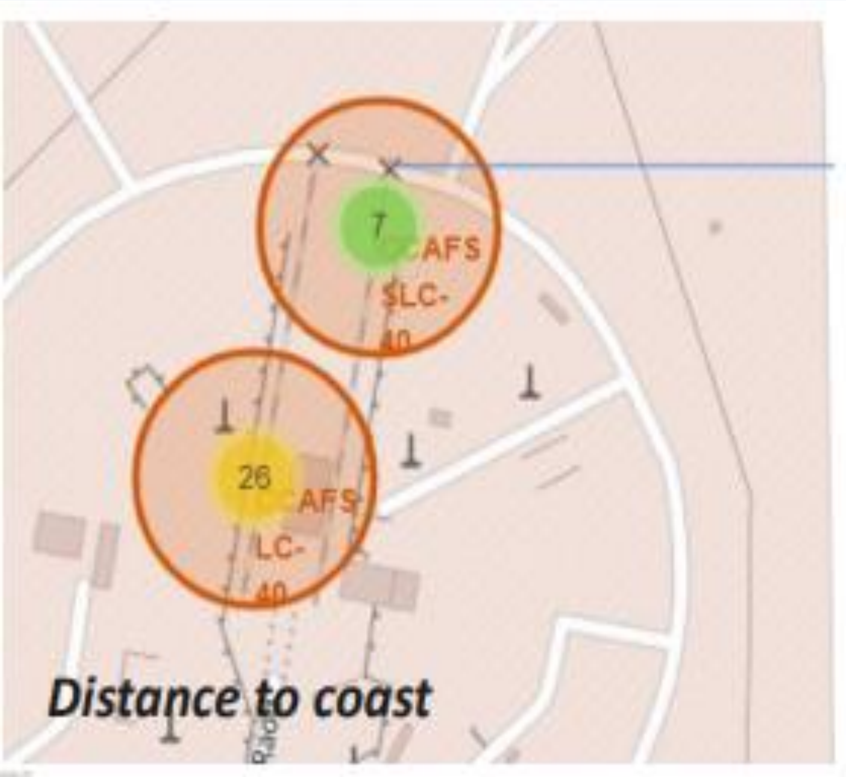
Florida Launch Sites





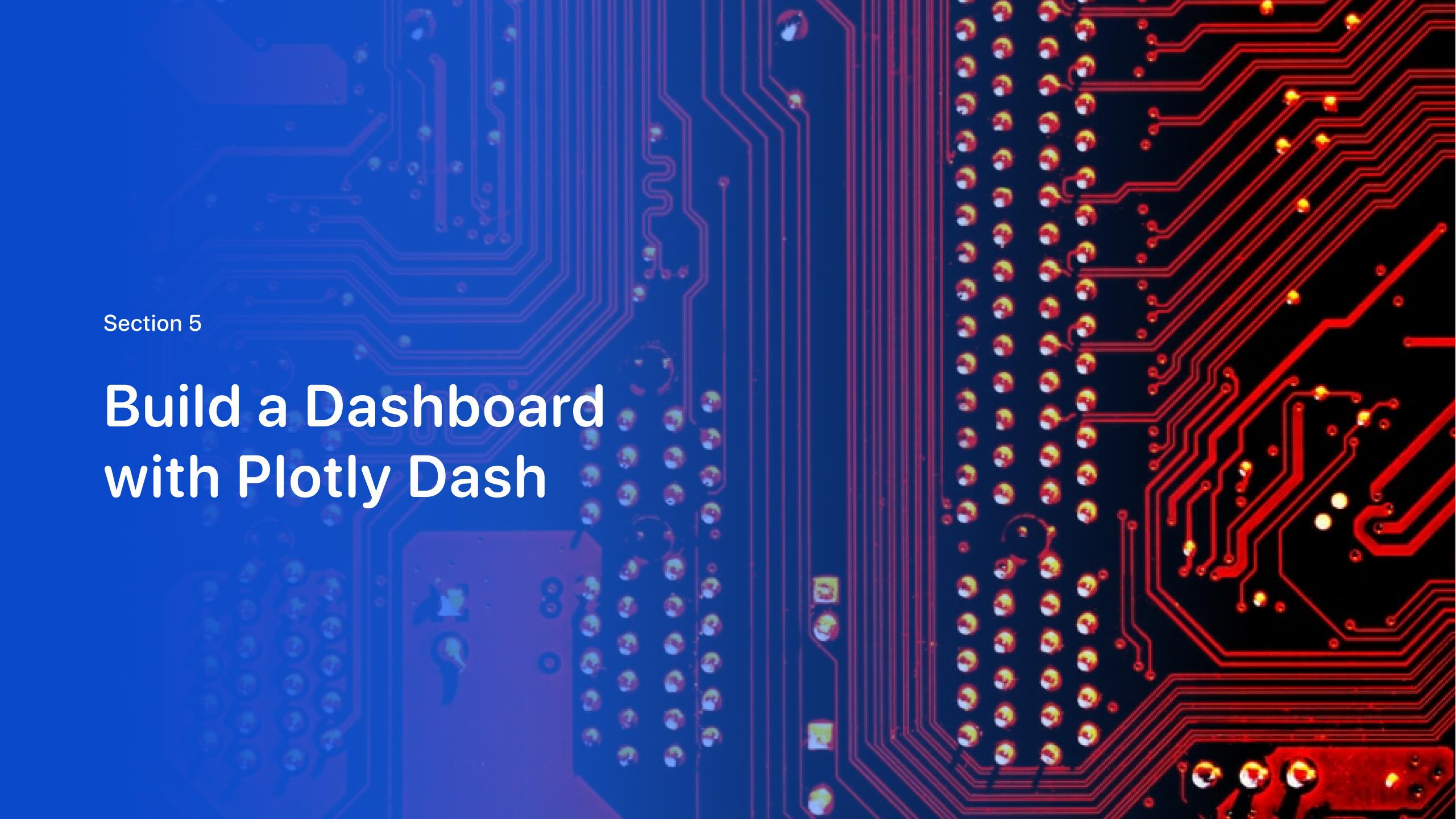
Launch Sites distance to landmarks





We had some questions and those are the answers to them:

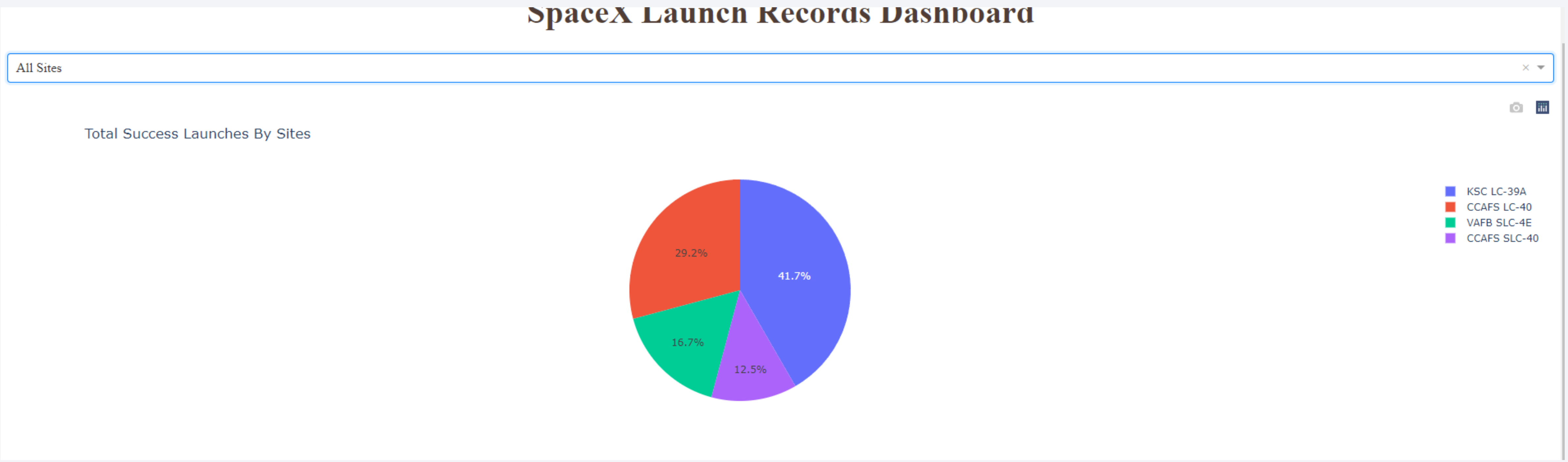
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 5

Build a Dashboard with Plotly Dash

Pie Chart for Total Success Launches by all Sites



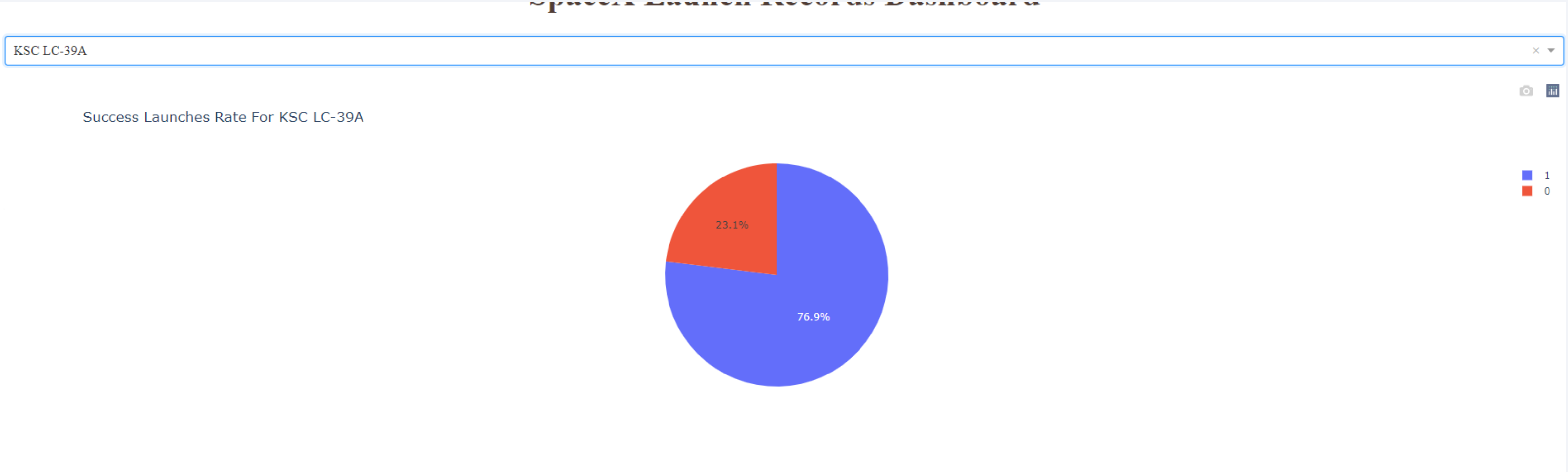
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Sites

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

Pie chart for the launch site with highest launch success ratio

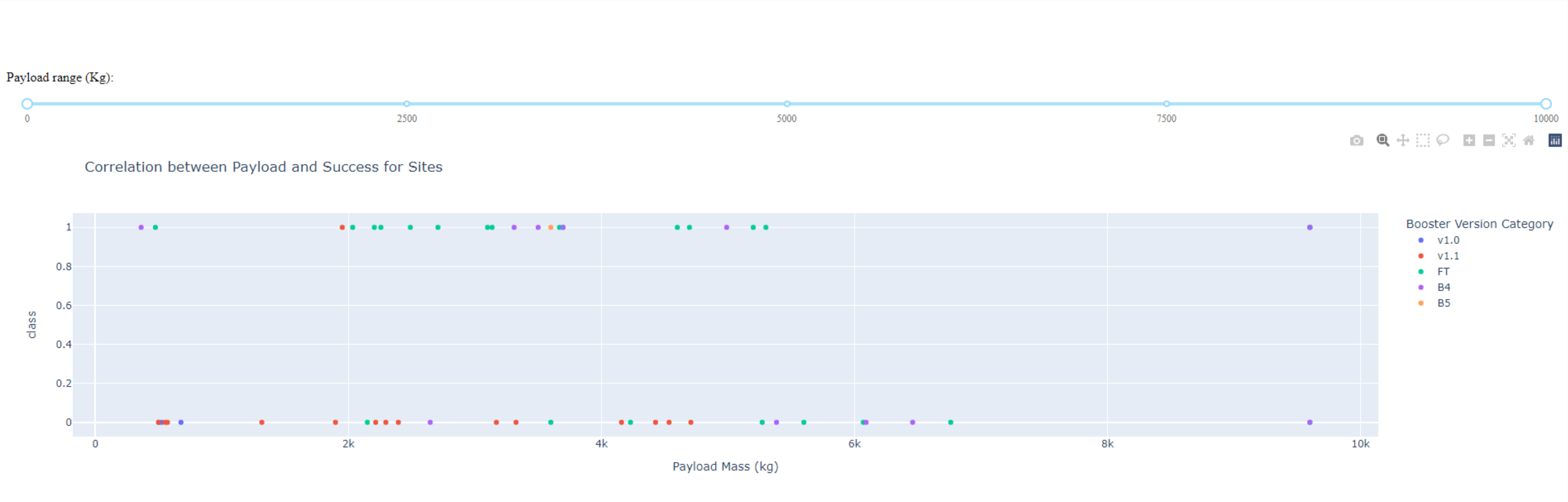


SpaceX Launch Records Dashboard

KSC LC-39A

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

Payload vs. Launch Outcome scatter plot for all sites

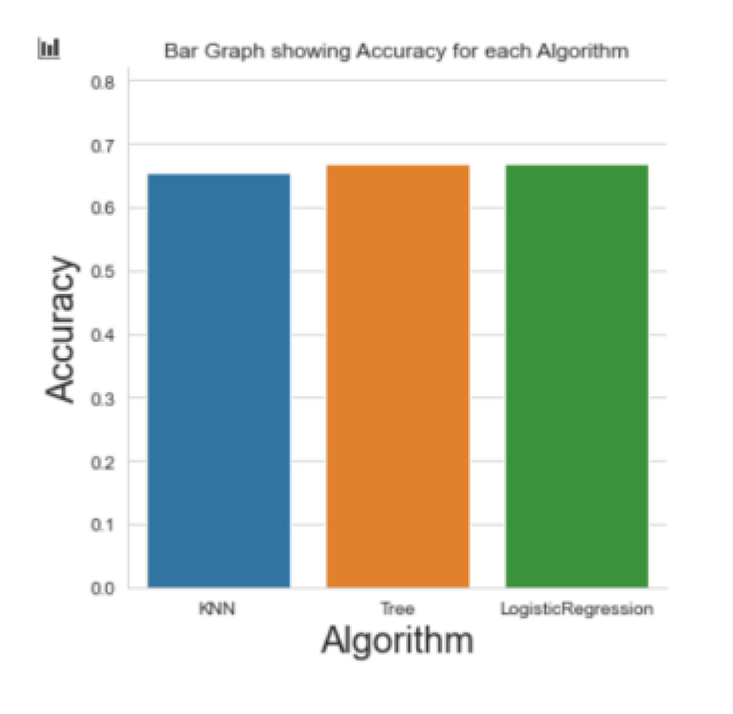


- success rates for low weighted payloads is higher than the heavy weighted payloads

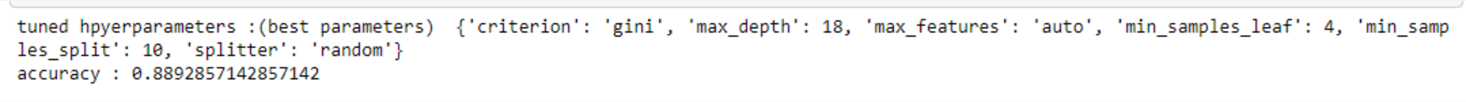
Section 6

Predictive Analysis (Classification)

Classification Accuracy

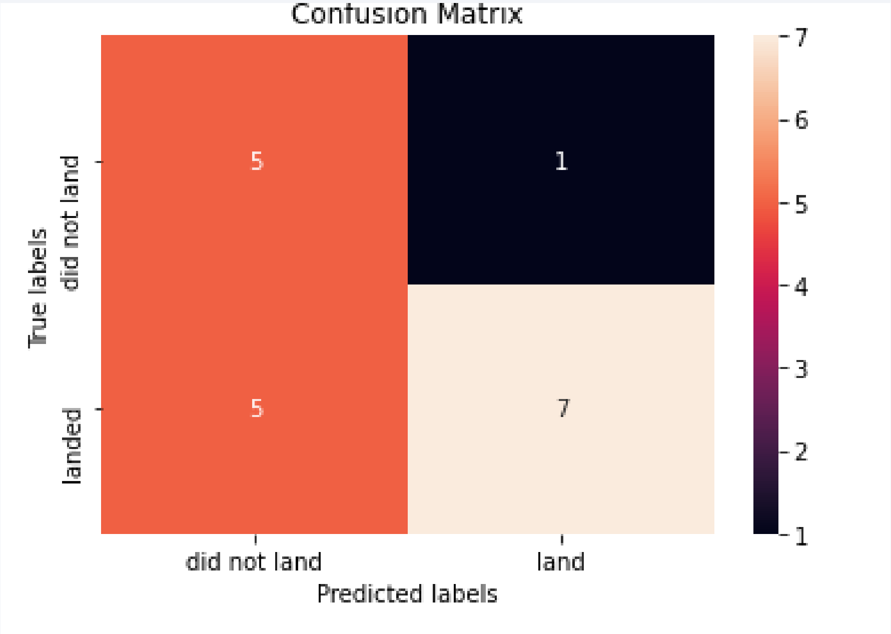


All the three algorithms have similar performance but Decision Tree algorithm was the best and on both train and test data



```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 18, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samp
les_split': 10, 'splitter': 'random'}
accuracy : 0.8892857142857142
```

Confusion Matrix



- Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives

Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset.
- Low weighted payloads outperform bigger payloads.
- The success rate of SpaceX launches is exactly related to the time in years it will take to perfect the launches.
- We can observe that KSC LC-39A had the most successful launches of any location.
- Orbit GEO,HEO,SSO,ES-L1 has the highest success rate.

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!