IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Milad Iranpour
9 September 2022

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 predicted if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 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 a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Common problems that needed solving.

  - What influences if the rocket will land successfully?

  - The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.

  - What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and dropping irrelevant 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

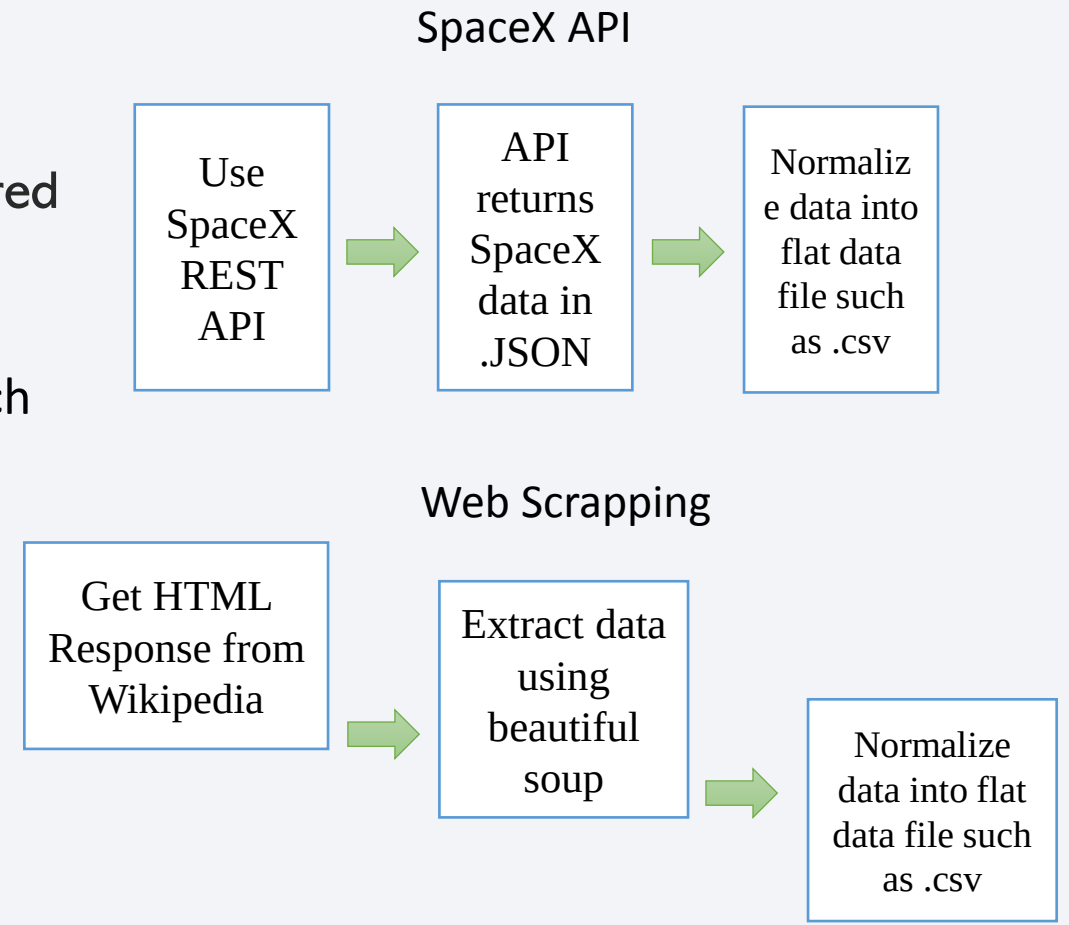
- The following datasets was collected by

  - We worked with SpaceX launch data that is gathered from the SpaceX REST API.

  - This API will give us data about launches, including information about the rocket used, payload delivered, launch 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, starts with api.spacexdata.com/v4/.

• Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup



Data Collection – SpaceX API

1. Getting Response from API

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

2. Converting Response to a .json file

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdc
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3. Apply custom functions to clean data

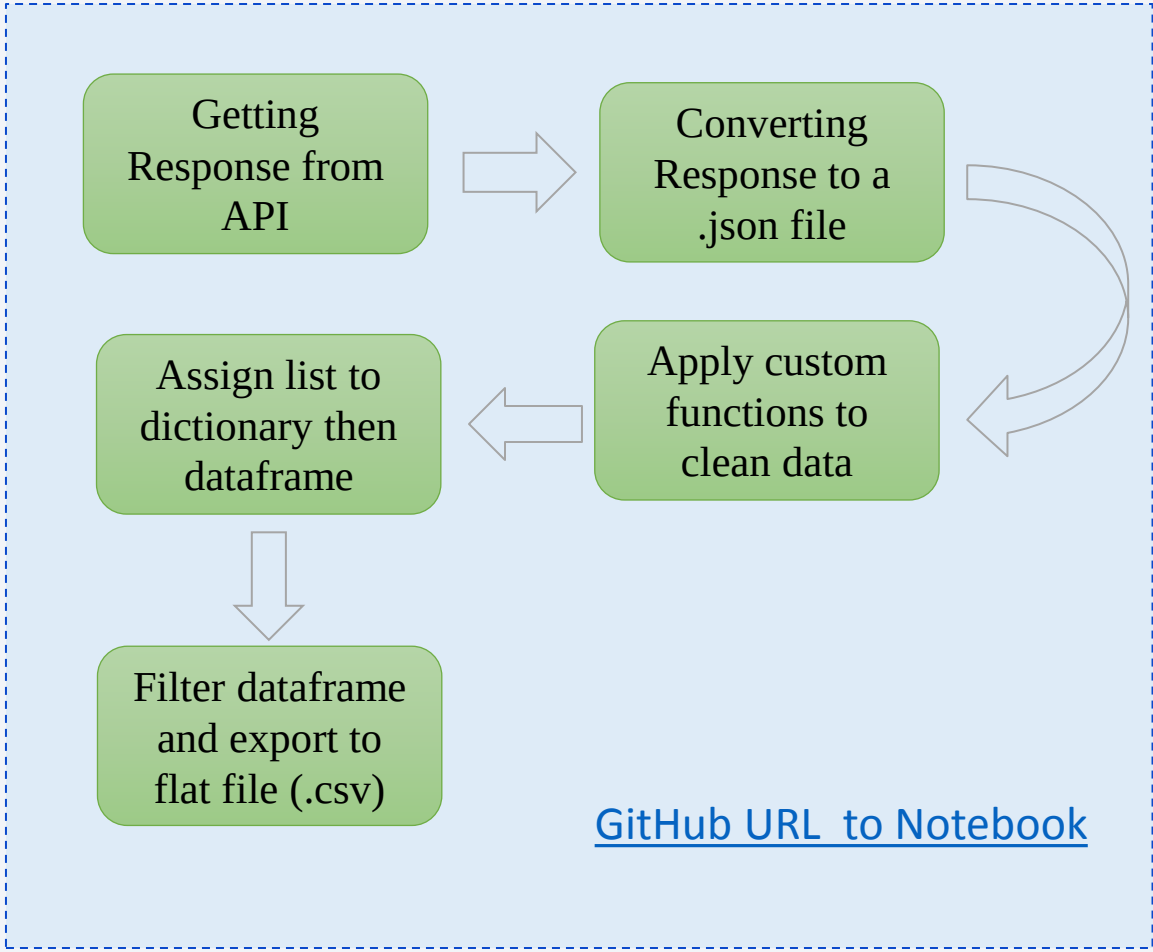
```
getBoosterVersion(data)      getPayloadData(data)
getLaunchSite(data)          getCoreData(data)
```

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}
```

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

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



Data Collection - Scraping

1 .Getting Response from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_o
data = requests.get(static_url).text
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(data, "html.parser")
```

3. Finding tables

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

4. Getting column names

```
column_names = []

# Apply find_all() function with `th` elemer
rows = first_launch_table.find_all('th')
# Iterate each th element and apply the pro
# Append the Non-empty column name (`if nam
for row in rows:
    name = extract_column_from_header(row)
    if(name != None and len(name) >0):
        column_names.append(name)
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each val
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys

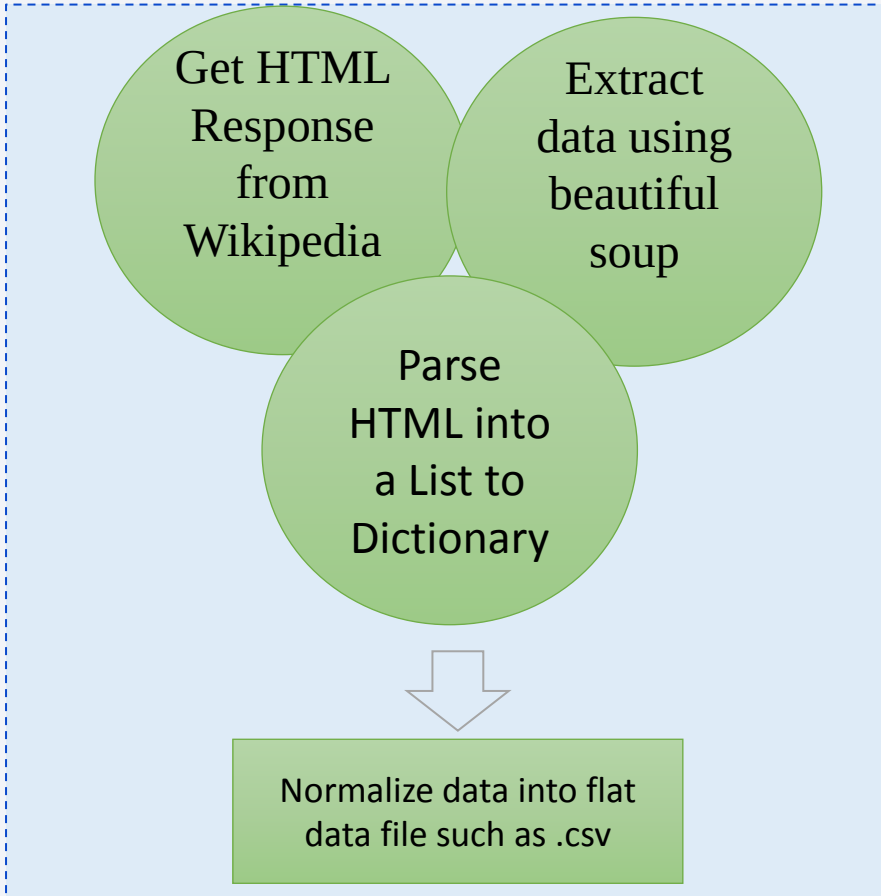
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(html_file.find_al
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as nu
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
```

7. Converting dictionary to dataframe

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

8. Dataframe to .CSV

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



GitHub URL to Notebook

Data Wrangling

Introduction

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful

Process

Perform Exploratory Data Analysis EDA on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

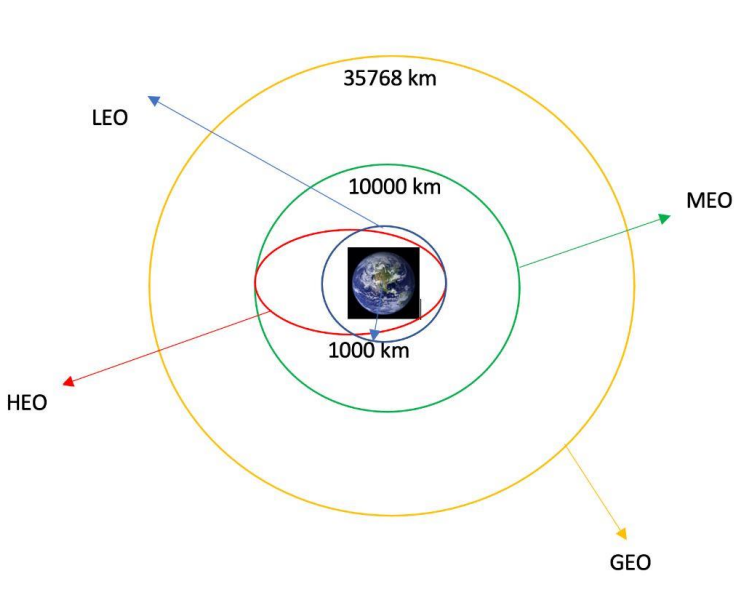
Calculate the number and occurrence of mission outcome per orbit type

Export dataset as .CSV

Create a landing outcome label from Outcome column

Work out success rate for every landing in dataset

Each launch aims to an dedicated orbit, and here are some common orbit types:



GitHub URL to Notebook

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 to Notebook

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 to Notebook

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.

We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()

Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. 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 to Notebook

Build a Dashboard with Plotly Dash

Used Python Anywhere to host the website live 24/7 so your can play around with the data and view the data

- The dashboard is built with Flask and Dash web framework.

  Graphs

  - 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 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 to Notebook

Predictive Analysis (Classification)

1. 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.

2. EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

3. IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning

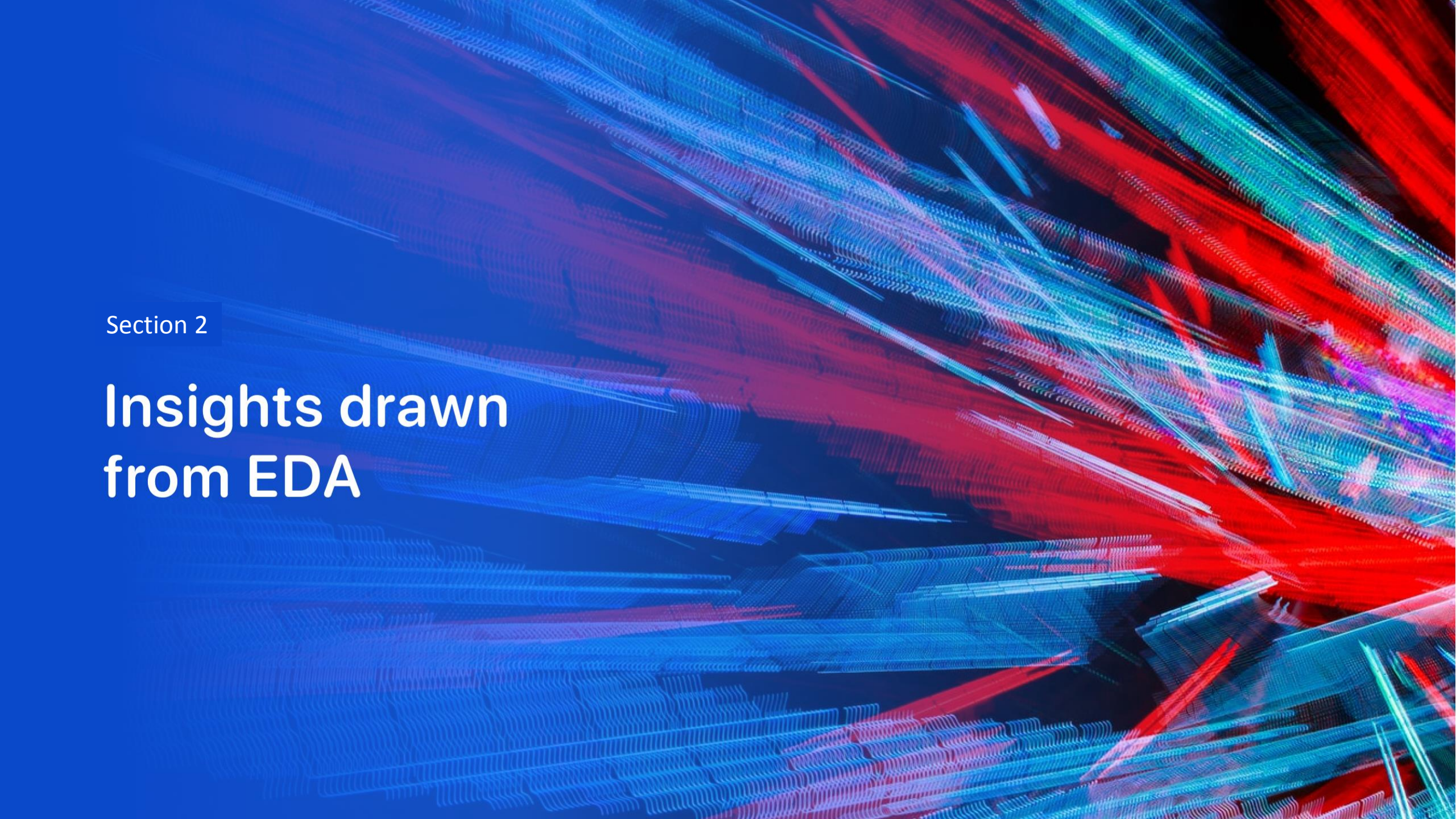
4. FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- 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.

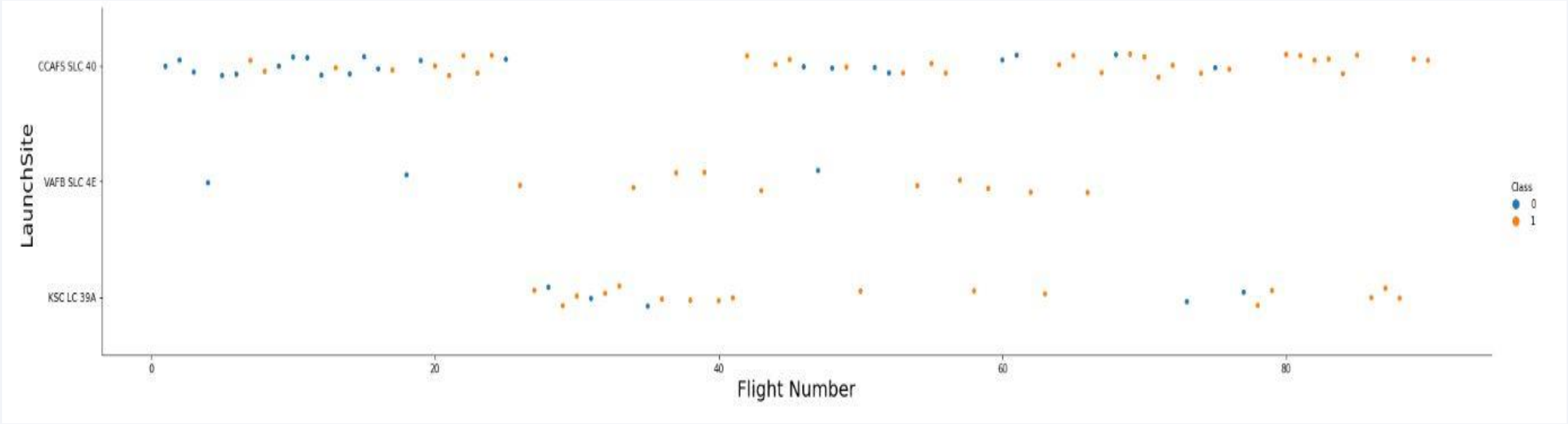
GitHub URL to Notebook

Results

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

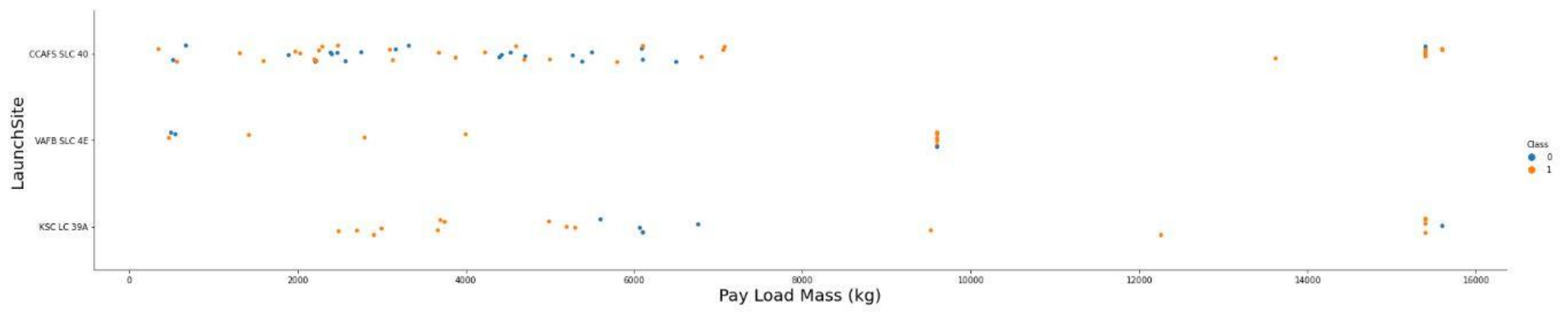

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



The more amount of flights at a launch site the greater the success rate at a launch site.

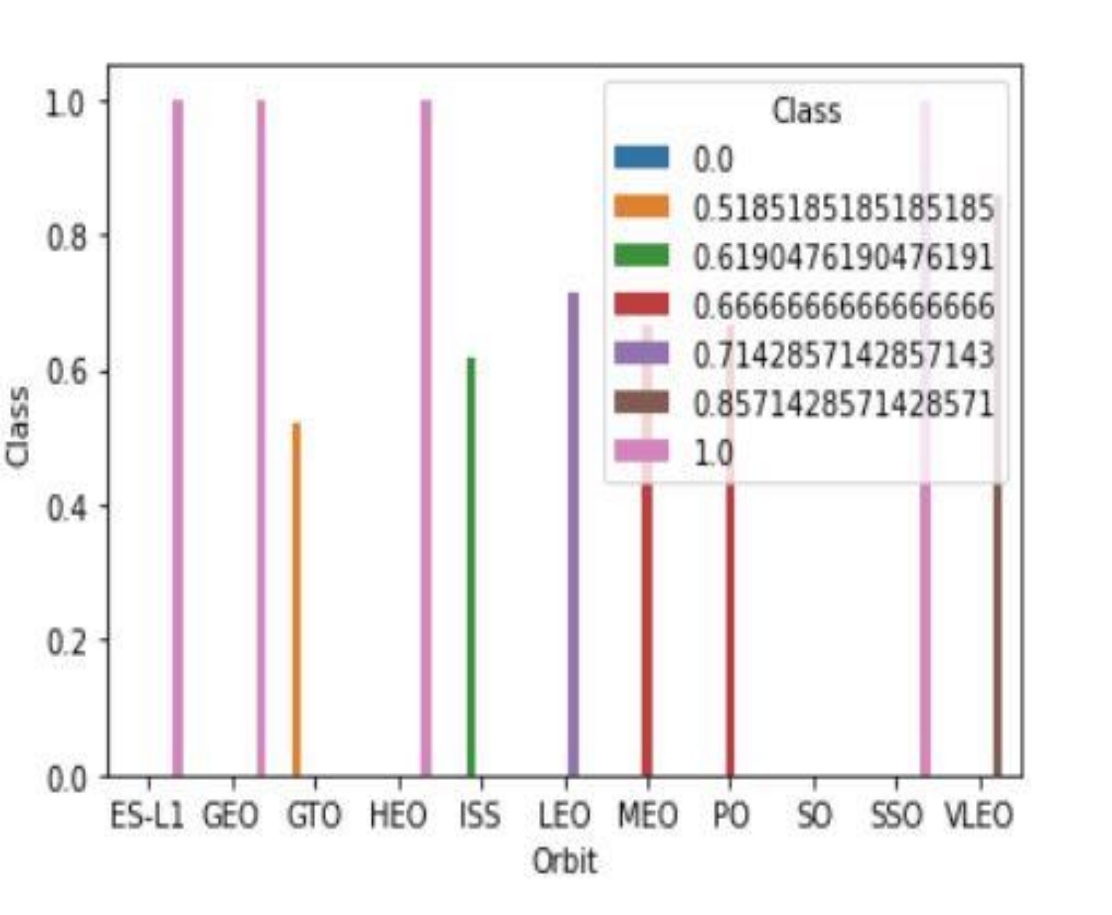
Payload vs. Launch Site



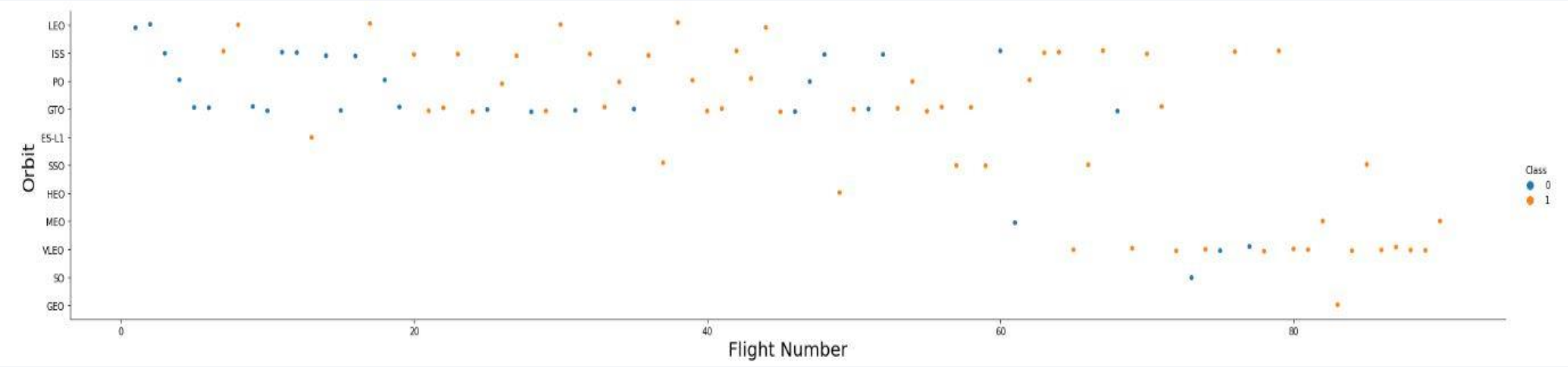
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch. if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

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

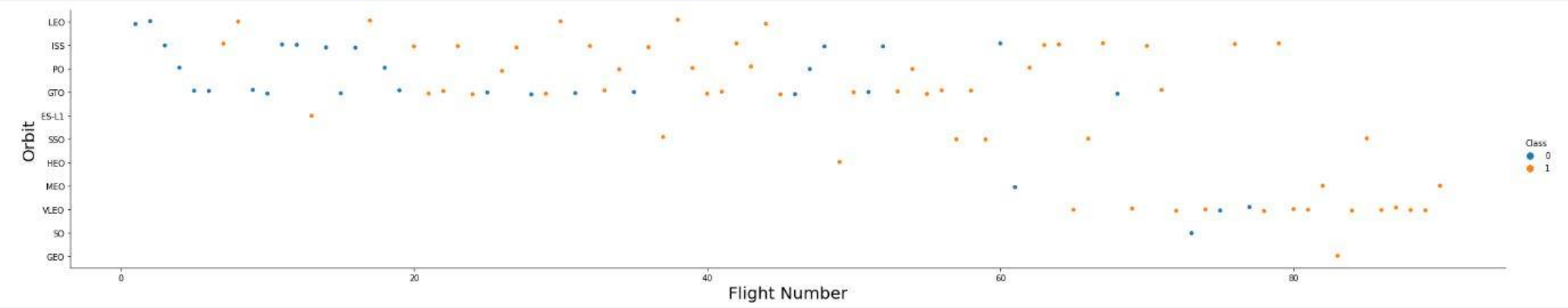


Flight Number vs. Orbit Type



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

Payload vs. Orbit Type

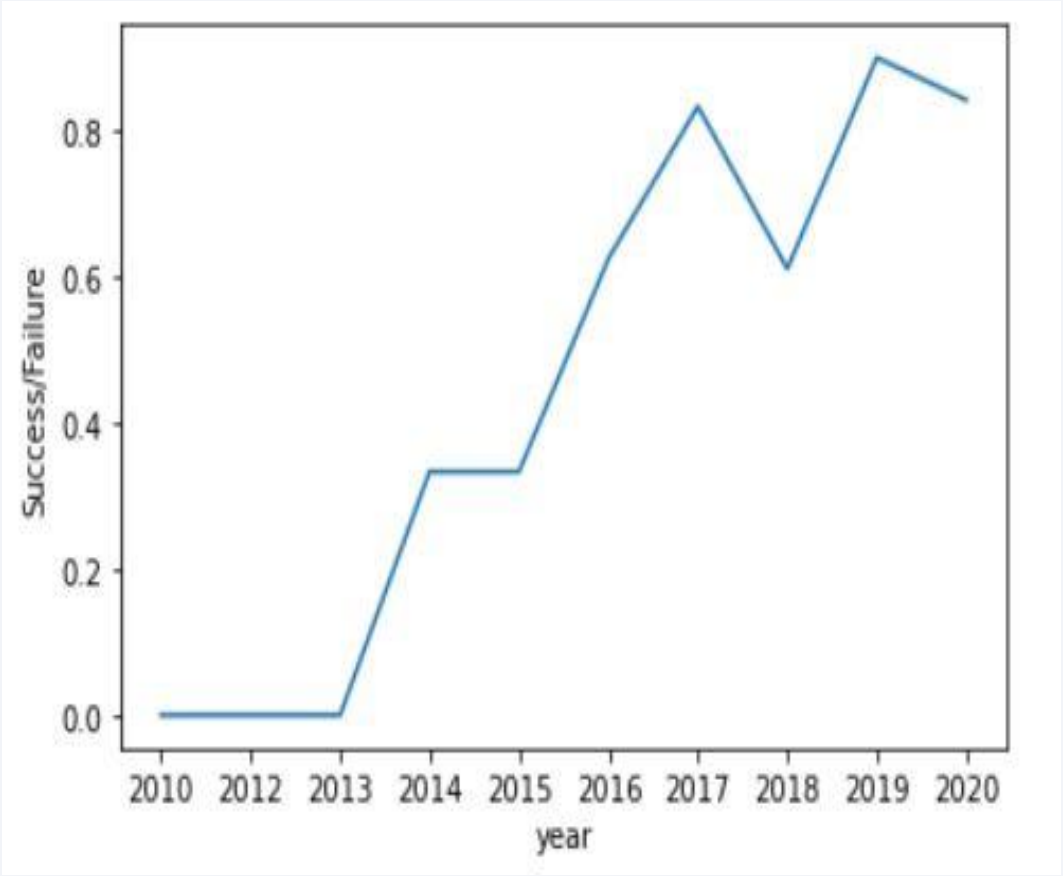


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

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



%sql select distinct(LAUNCH_SITE) from SPACEXTBL;
%sql select distinct(LAUNCH_SITE) from SPACEXTBL;

All Launch Site Names

SQL QUERY

```
select distinct(LAUNCH_SITE) from
SPACEXTBL;
```

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

QUERY EXPLAINATION

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

Launch Site Names Begin with 'CCA'

SQL QUERY

```
select LAUNCH_SITE from
SPACEXTBL where LAUNCH_SITE
like 'CCA%' limit(5)
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

QUERY EXPLAINATION

We display 5 records where launch sites begin with the string 'CCA'

Total Payload Mass

SQL QUERY

```
select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer == 'NASA (CRS)'
```

SUM(PAYLOAD_MASS__KG_)
45596

We display the total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

SQL QUERY

```
select avg(PAYLOAD_MASS__KG_) as
payloadmass from SPACEXTBL where
Booster_Version == 'F9 v1.1'
```

payloadmass
2928.4

QUERY EXPLAINATION

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 QUERY

```
select min(Date) from SPACEXTBL where
Mission_Outcome == 'Success'
```

min(Date)
01-03-2013

QUERY EXPLAINATION

Using the function MIN works out the minimum date in the column Date

The WHERE clause filters the dataset to only perform calculations on Mission_Outcome Success

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY

```
select Booster_Version from SpaceXtbl where Landing_Outcome = 'Success
(ground pad)' AND Payload_MASS_KG_ > 4000 AND Payload_M
```

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

```
select count(MISSION_OUTCOME) as
missionoutcomes from SPACEXTBL GROUP BY
MISSION_OUTCOME;
```

missionoutcomes
1
98
1
1

QUERY EXPLAINATION

Using the word count in the query means that it will only show number in the Missin_outcome column from tblSpaceX

GROUP BY puts the list in order set to a certain condition.

Boosters Carried Maximum Payload

```
select BOOSTER_VERSION as
boosterversion from SPACEXTBL
where PAYLOAD_MASS__KG_=(select
max(PAYLOAD_MASS__KG_) from
SPACEXTBL);
```

QUERY EXPLAINATION

Using the subquery for getting maximum payload mass from SPACEXTBL table.

Then List the names of the booster_versions which have carried the maximum payload mass

boosterversion
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

```
SELECT DATE AS
D,MISSION_OUTCOME,BOOSTER_VERSION,
LAUNCH_SITE FROM SPACEXTBL where
(substr(Date,7,4)='2015')
```

D	Mission_Outcome	Booster_Version	Launch_Site
10-01-2015	Success	F9 v1.1 B1012	CCAFS LC-40
11-02-2015	Success	F9 v1.1 B1013	CCAFS LC-40
02-03-2015	Success	F9 v1.1 B1014	CCAFS LC-40
14-04-2015	Success	F9 v1.1 B1015	CCAFS LC-40
27-04-2015	Success	F9 v1.1 B1016	CCAFS LC-40
28-06-2015	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
22-12-2015	Success	F9 FT B1019	CCAFS LC-40

QUERY EXPLAINATION

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

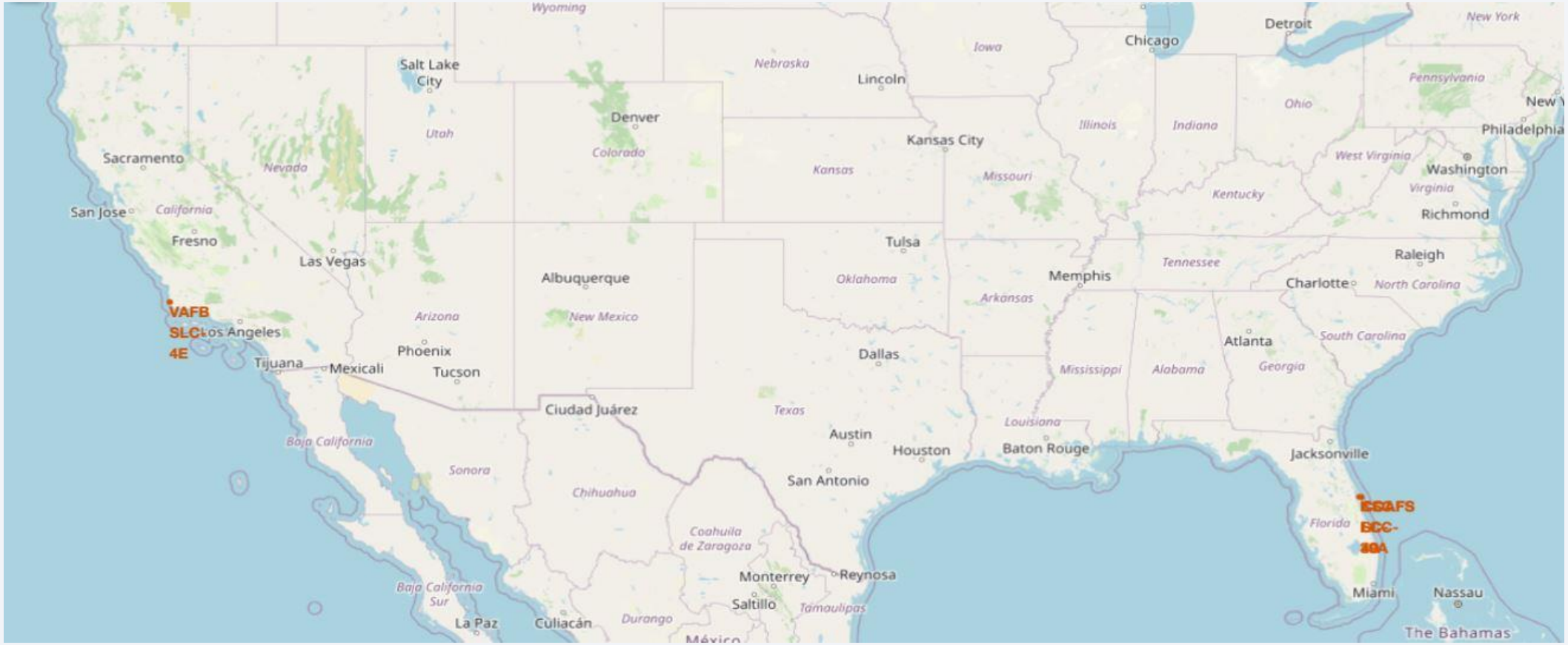
Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

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

- %sql SELECT Landing_Outcome FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

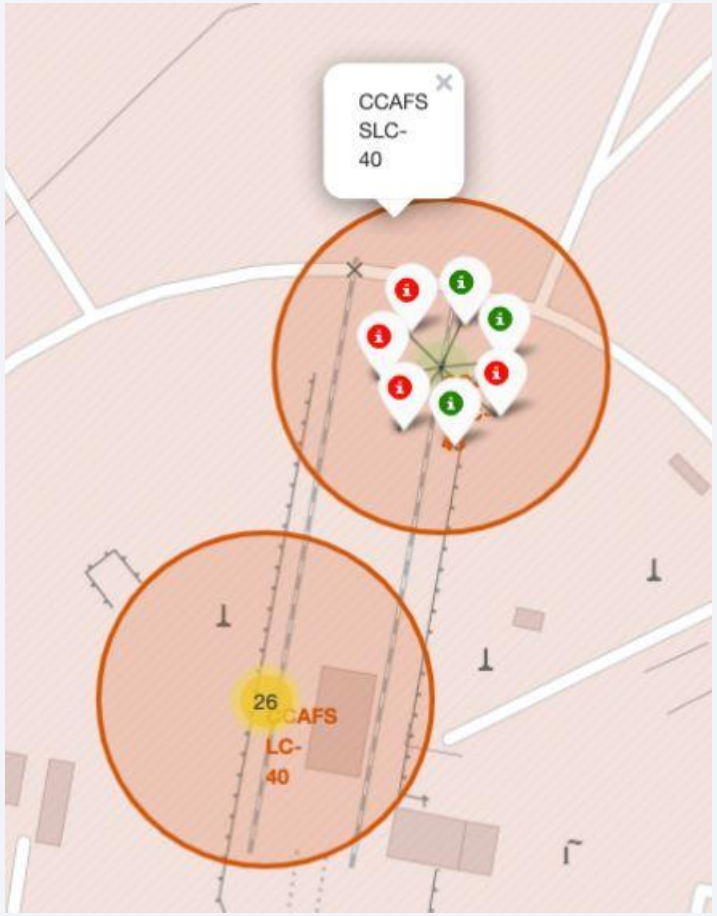

Section 3
Launch Sites
Proximities Analysis

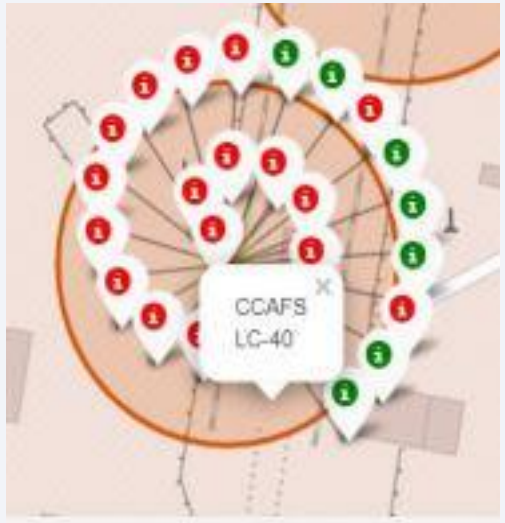
All launch sites global map markers



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

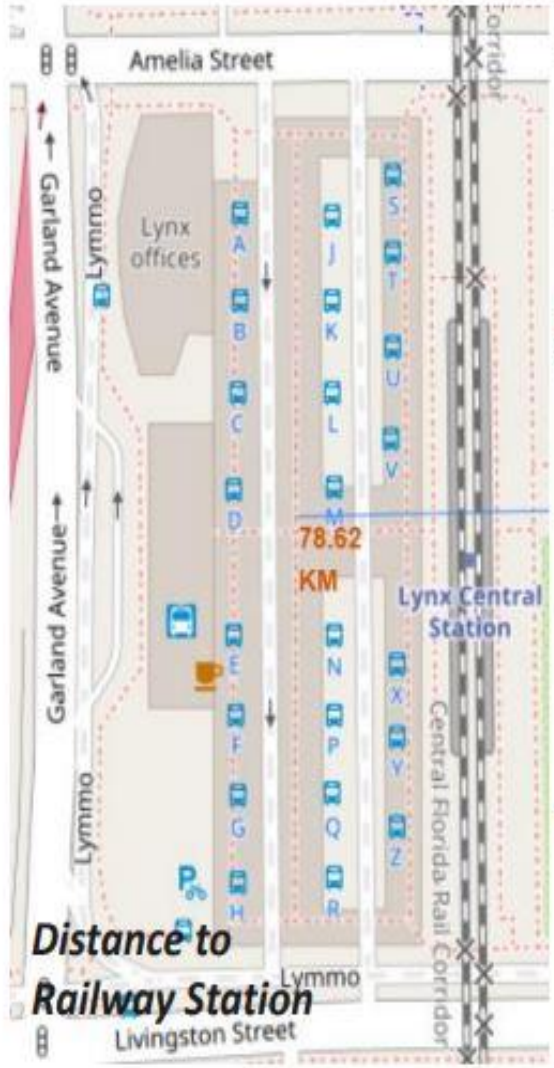
Mark the success/failed launches for each site on the map

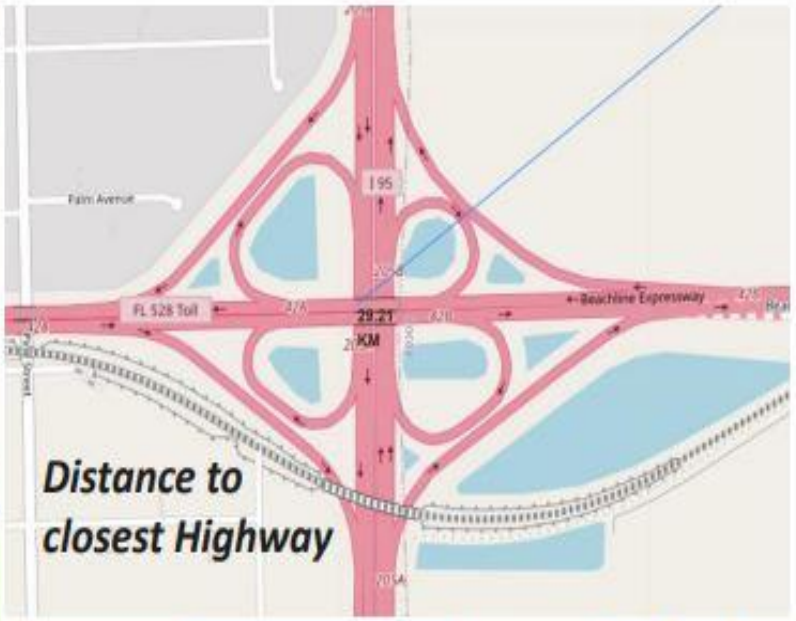


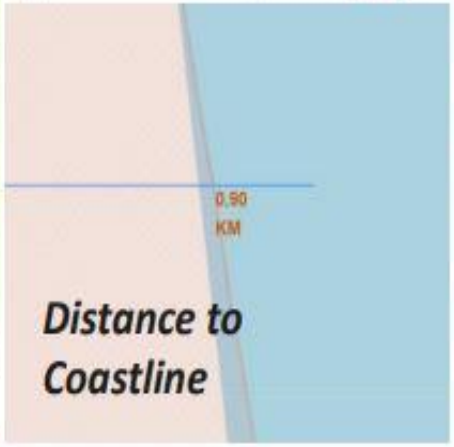


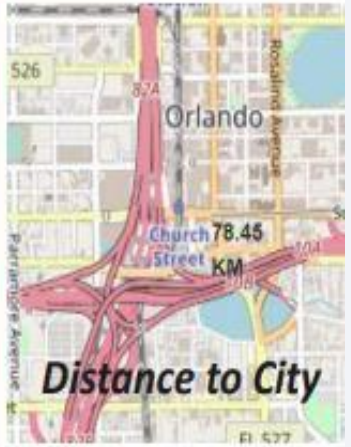
Green Marker shows successful Launches and Red Marker shows Failures

Calculate the distances between a launch site to its proximities

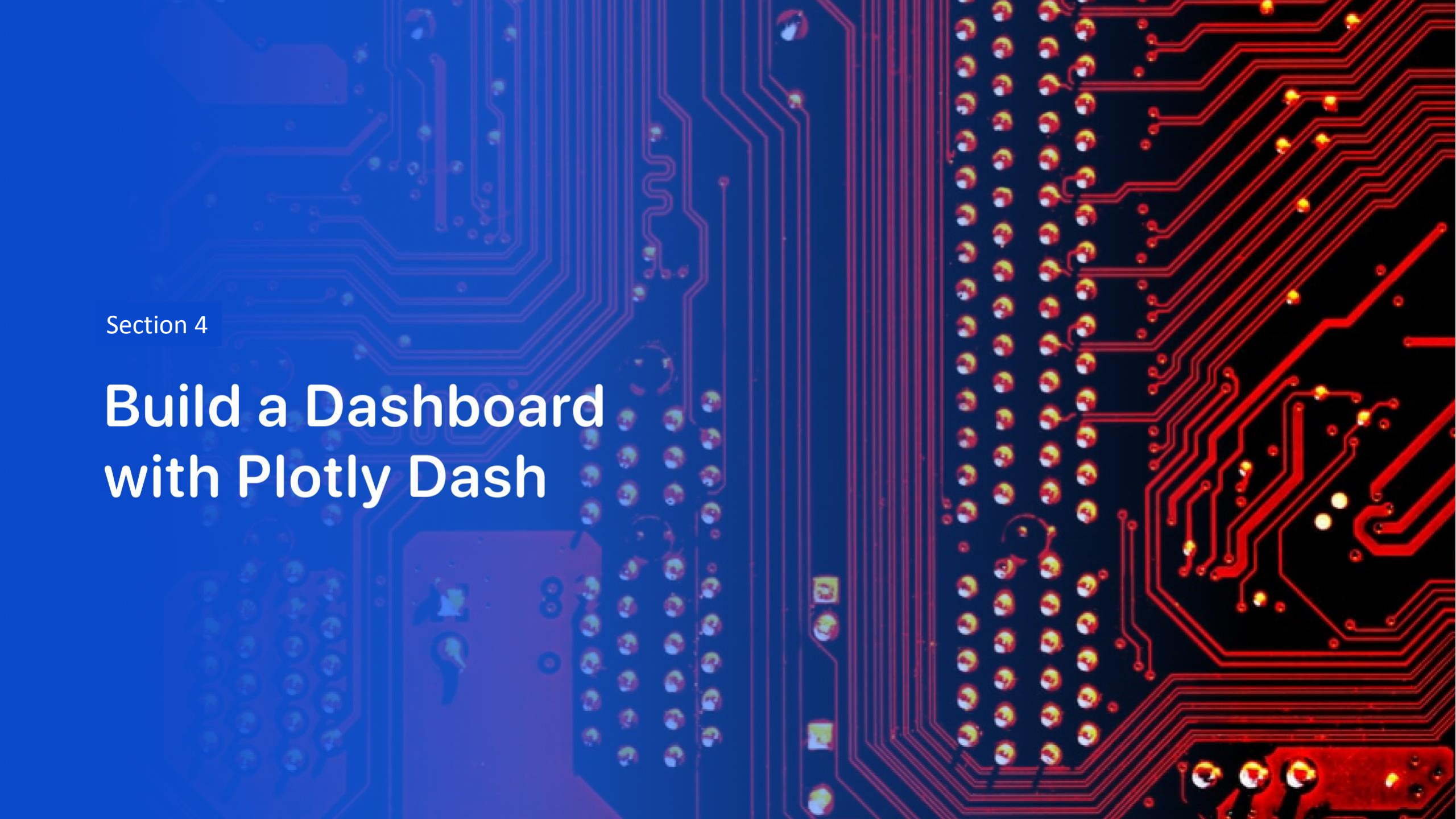




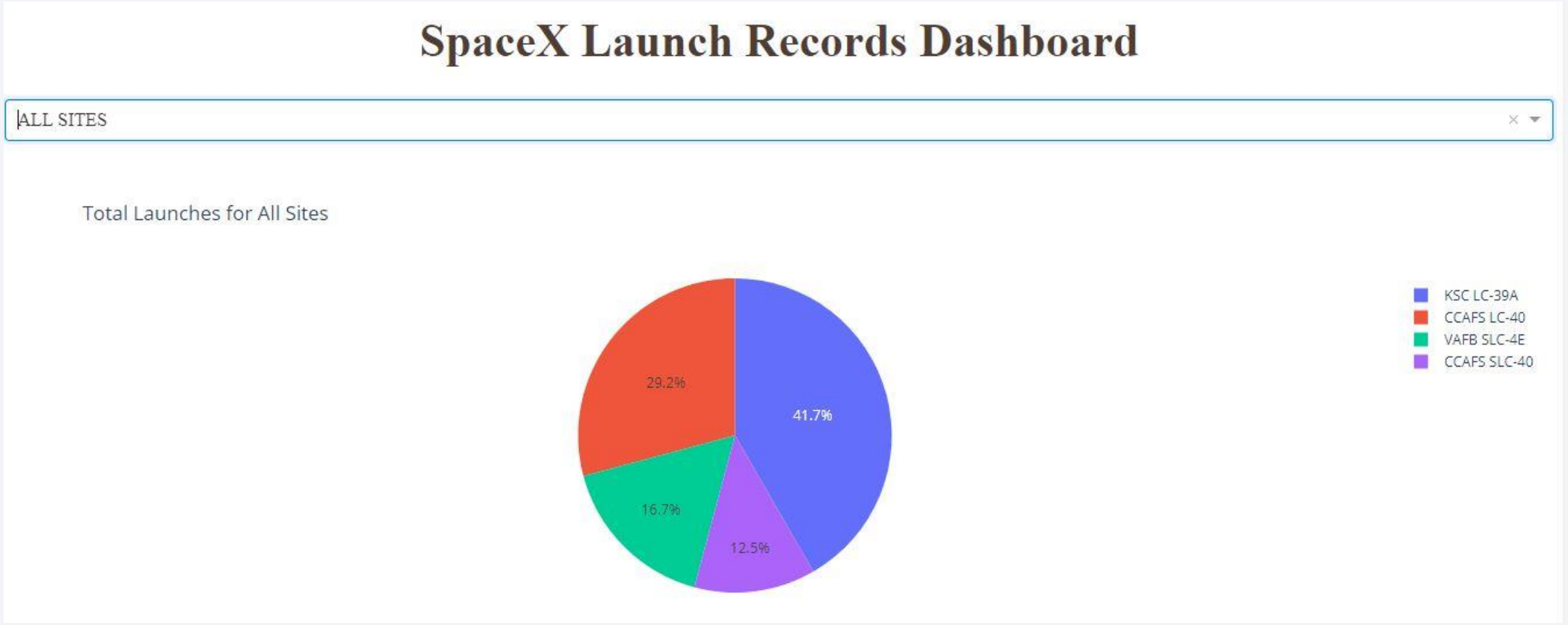




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

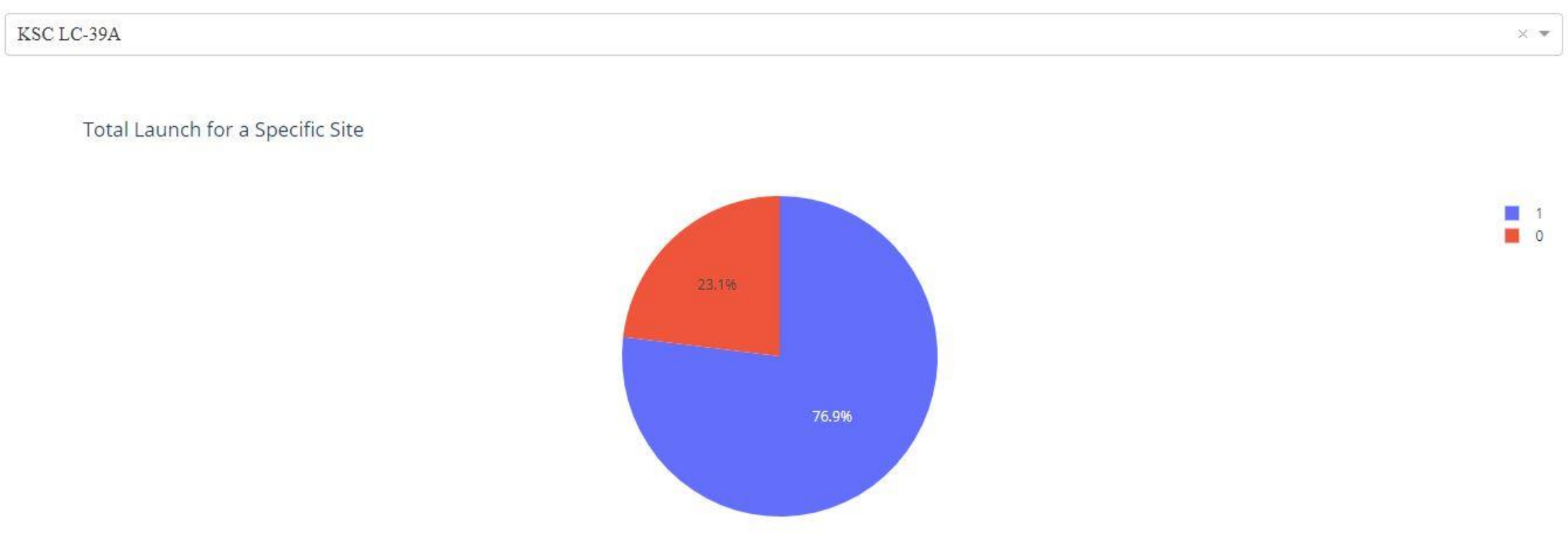

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard – Total Launches for All Sites



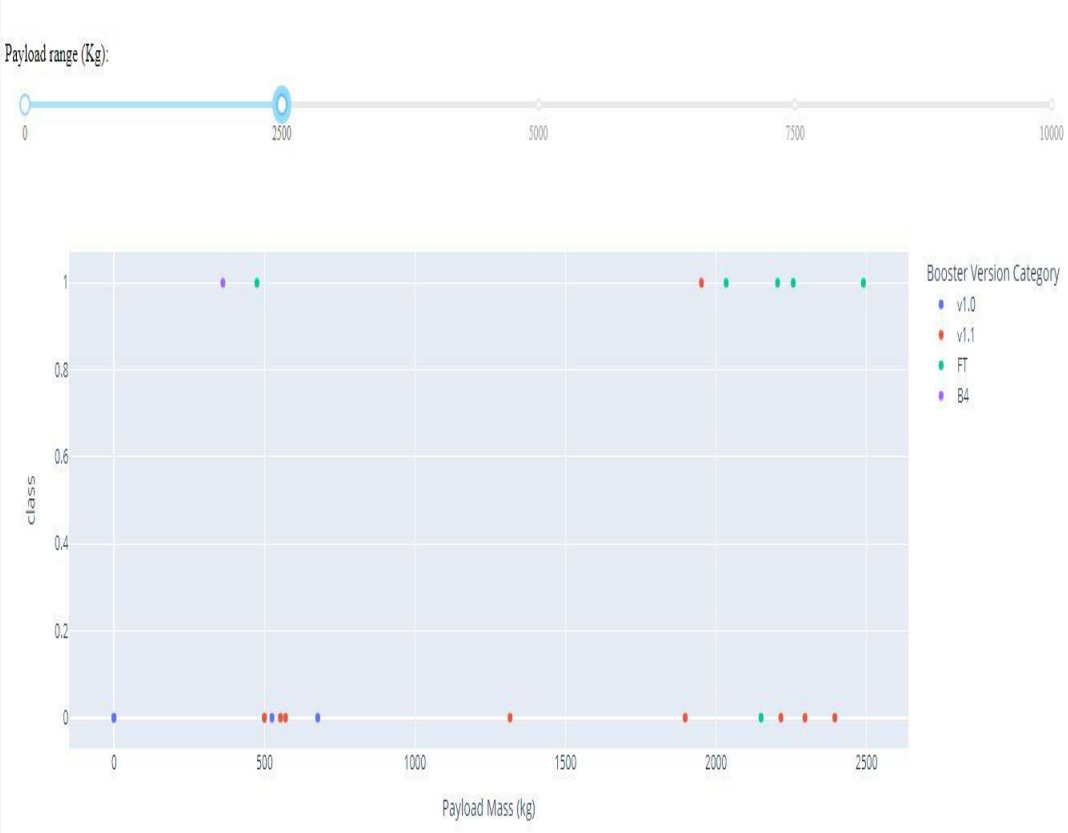
We can see that KSL LC-39A had the most successful launches from all the sites

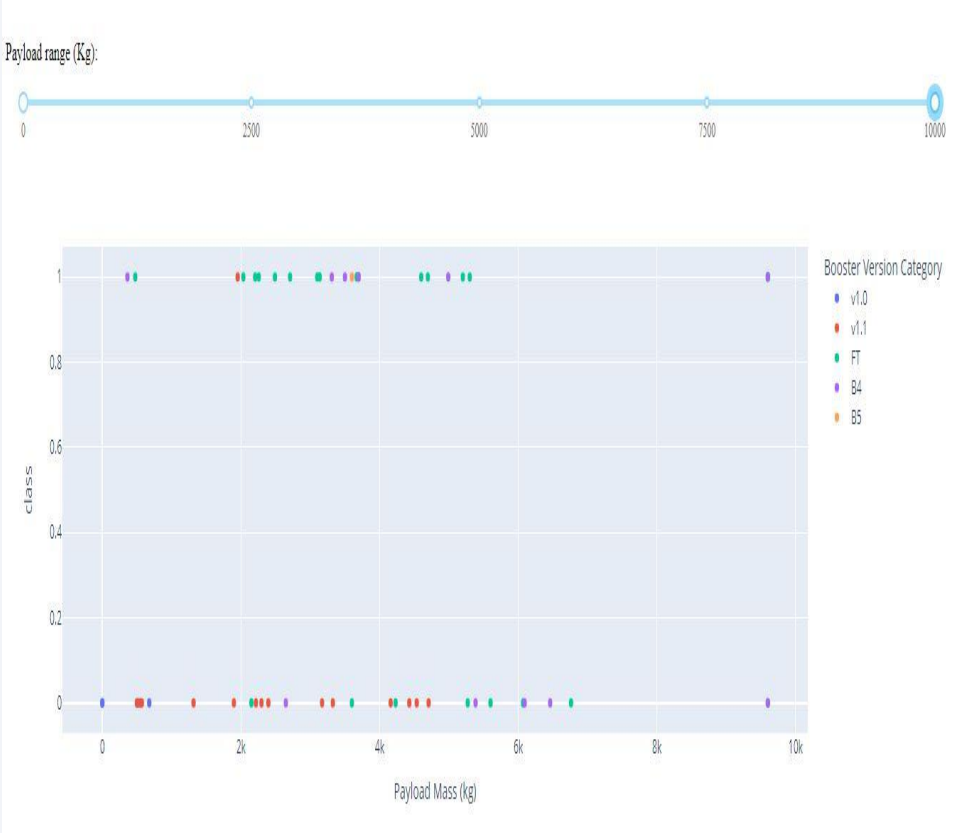
SpaceX Launch Records Dashboard – Total Launches for a Sites



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

DASHBOARD – Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider





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

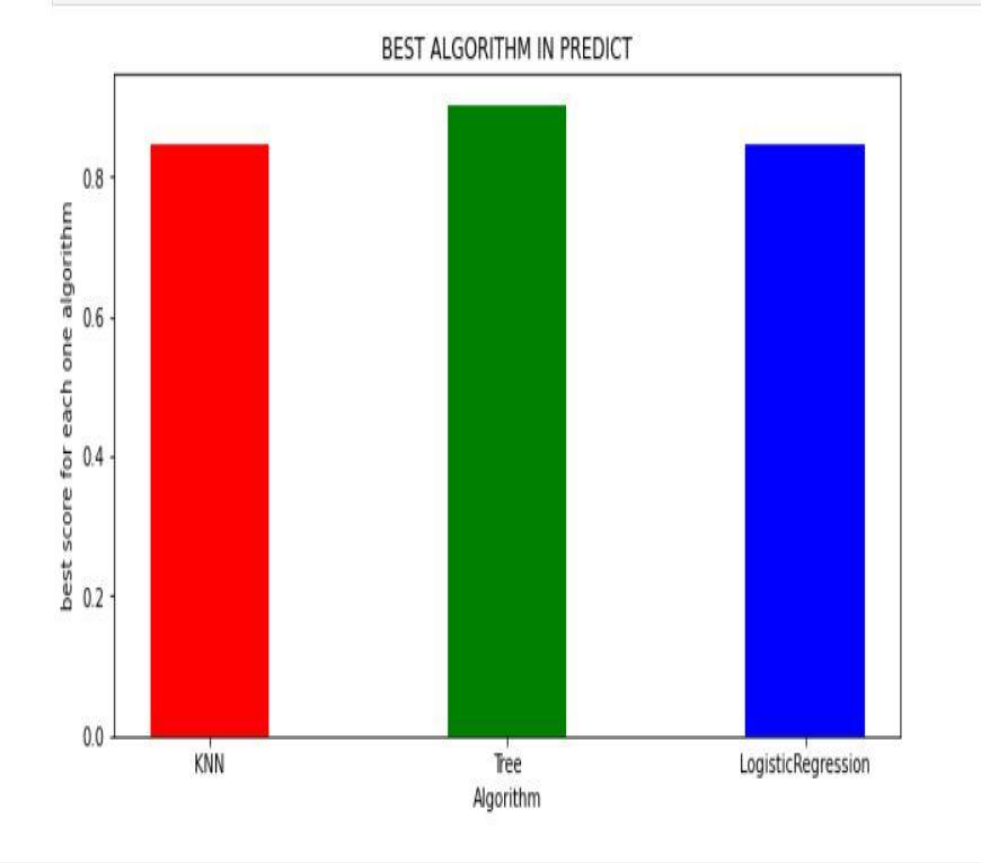
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier model has the highest classification accuracy

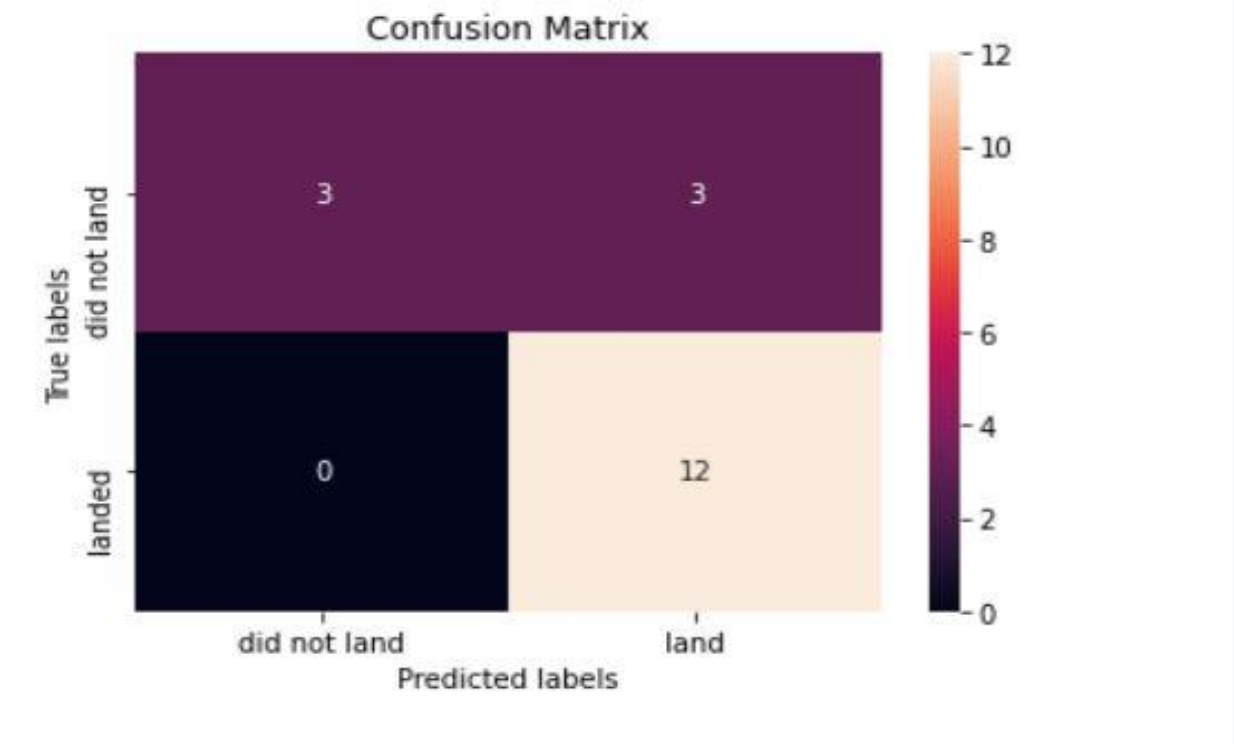
```
{'KNN': 0.8482142857142858,
 'Tree': 0.9028571428571428,
 'LogisticRegression': 0.8464285714285713}
```




```
Best Algorithm is Tree with a score of 0.9028571428571428
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix for the Tree

- A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm. So decision tree classifier model has the best performance.



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

- 1-1-jupyter-labs-spacex-data-collection-api
- 1-2-jupyter-labs-webscraping
- 1-3-labs-jupyter-spacex-Data wrangling
- 2-1-jupyter-labs-eda-sql-coursera_sqllite
- 2-2-jupyter-labs-eda-dataviz
- 3-1-lab_jupyter_launch_site_location
- 3-2-spacex_dash_app
- 4-1-SpaceX_Machine Learning Prediction_Part_5

Thank you!