IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mutholib Yusira
14th of October, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX is a revolutionary company that has disrupted the space industry by offering rocket launches specifically Falcon 9 for as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this savings is due to SpaceX astounding idea to reuse the first stage of the launch on the next mission. Repeating this process will make the price reduce even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variables and how it is affecting the outcome.
- The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling:
    - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

Data Collection

- Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia

- For REST API, the get request method was used. Then the response content was decoded as Json and turn it into a pandas dataframe using json_normalize(). The data was then cleaned, checked for missing values and filled with whatever needed.

- For web scrapping, BeautifulSoup was used to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

Get request for rocket launch data using API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

From:
https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Collection.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want a
nd the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rocket
s with 2 extra rocket boosters and rows that have multiple payloads in a
single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the s
ingle value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then ex
tracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

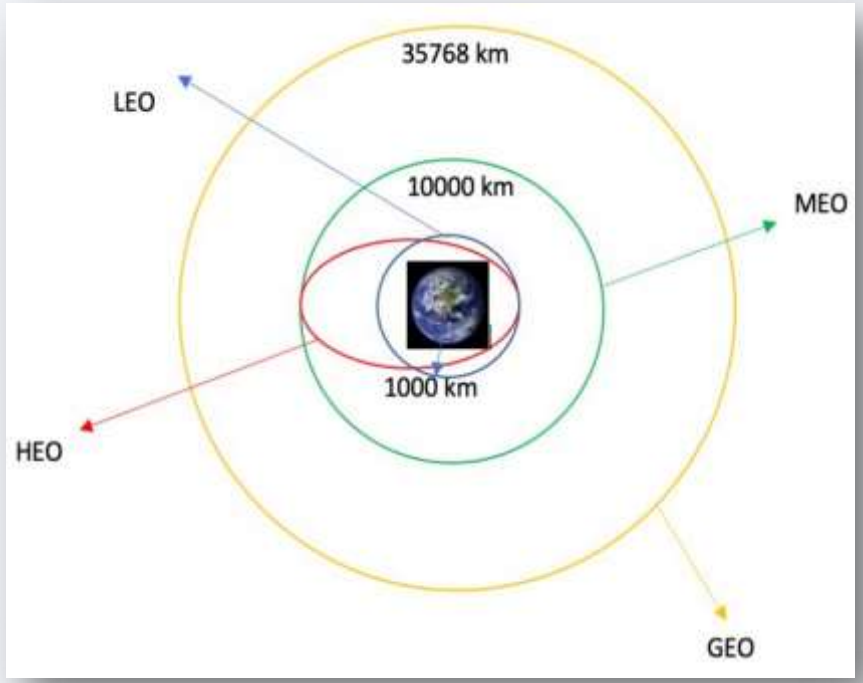
From:
https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response te
xt content
soup = BeautifulSoup(data,'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Data Wrangling



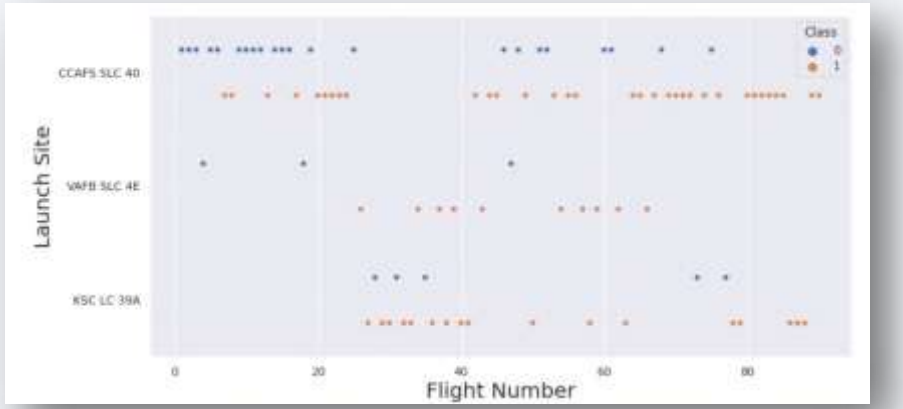
From:
https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Wrangling.ipynb

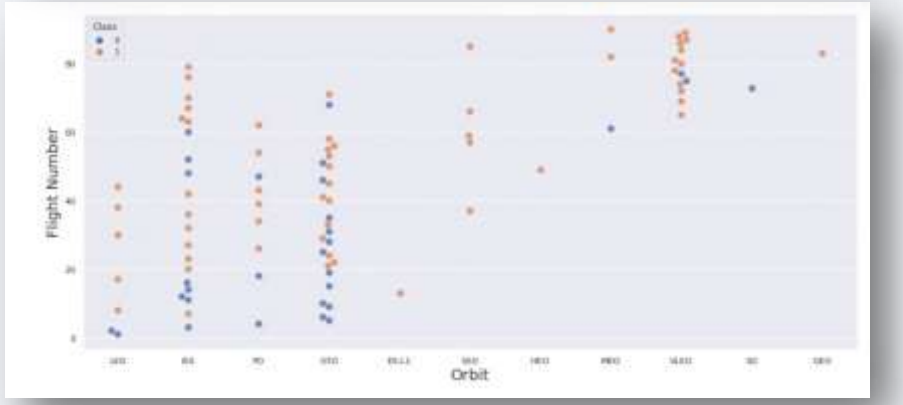
Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

The number of launches on each site was first calculated, then the number and occurrence of mission outcome per orbit type.

A landing outcome label from was feature engineered from the outcome column. This will make it easier for further analysis, visualization, and machine learning. Lastly, the result was exported to a CSV.

EDA with Data Visualization





Scatter graphs was used to find the relationship between the attributes such asbetween:

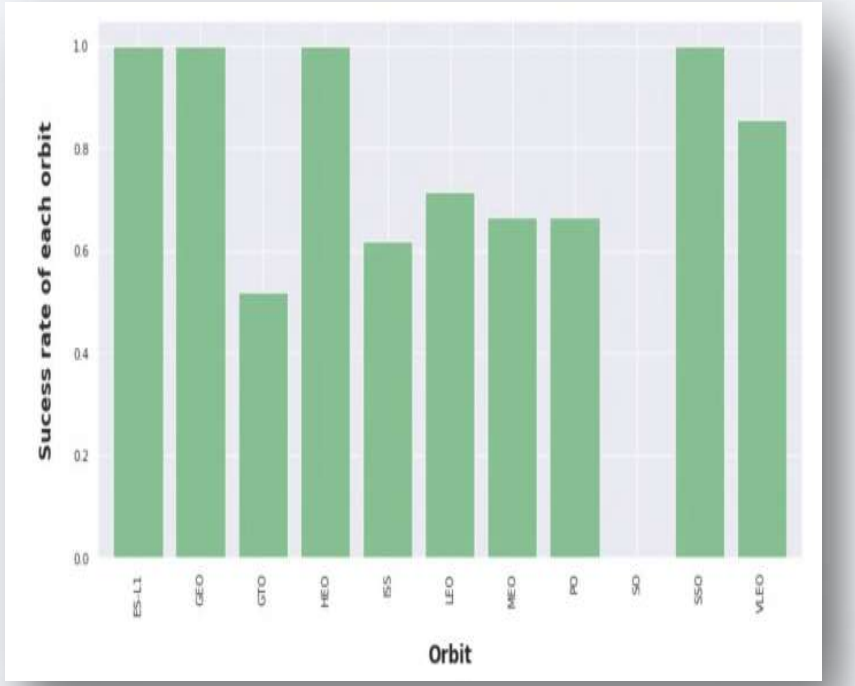
- Payload and Flight Number.
- Flight Number and LaunchSite.
- Payload and LaunchSite.
- Flight Number and OrbitType.
- Payload and Orbit Type.

Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs, it's very easy to see which factors affect the success of the landing outcomes the most.

https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20Visualization.ipynb

1
2

EDA with Data Visualization



After getting a hint of the relationships using scatter plot. Other visualization tools such as bar graph was used for furtheranalysis.

Bar graphs is one of the easiest way to interpret the relationship between the attributes. In this case, the bar graph was used to determine which orbits have the highest probability of success.

Then Feature Engineering was used to created the dummy variables for the categorical columns.

https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20Visualization.ipynb

Build an Interactive Map with Folium

To visualize the launch data into an interactive map, the latitude and longitude coordinates at each launch site was taken, and a circle marker around each launch site was added with a label of the name of the launch site.

Then the dataframe launch_outcomes(failure,success) was assigned to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().

The Haversine's formula was used to calculated the distance of the launch sites to various landmark to find answer to the questions of:

- How close the launch sites with railways, highways and coastlines?
- How close the launch sites with nearby cities?

From: https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was built with Plotly dash, allowing the users to play around with the data.
- Pie charts showing the total launches by a certain sites were plotted
- Scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version was also plotted

The link of the app.py:: https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- Set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- Plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

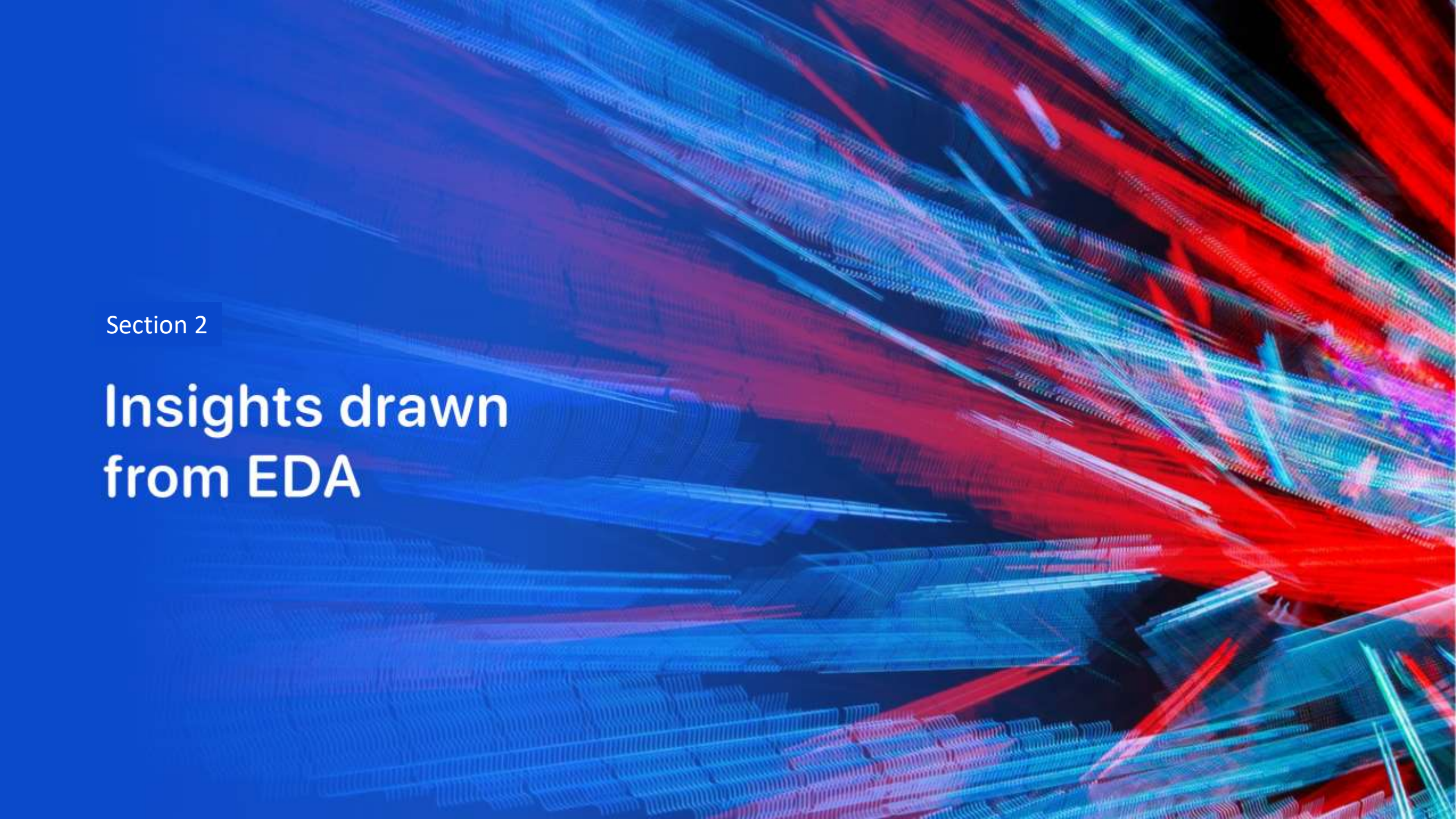
- The model with the best accuracy score will be the best performing model.

From: https://github.com/Yusira/Applied-Data-Science-Capstone-SpaceX/blob/main/Machine%20Learning%20Prediction.ipynb

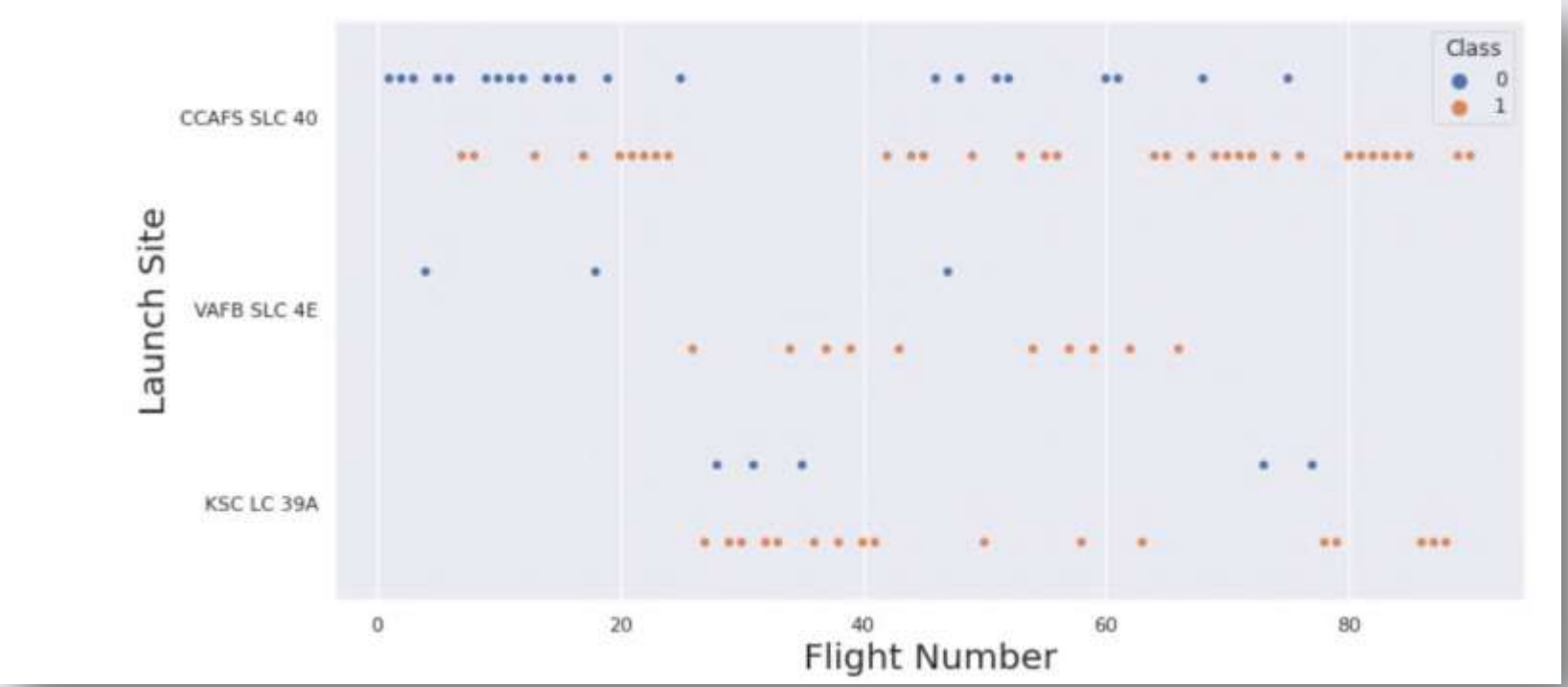
Results

The results will be categorized to 3 main results which is:

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

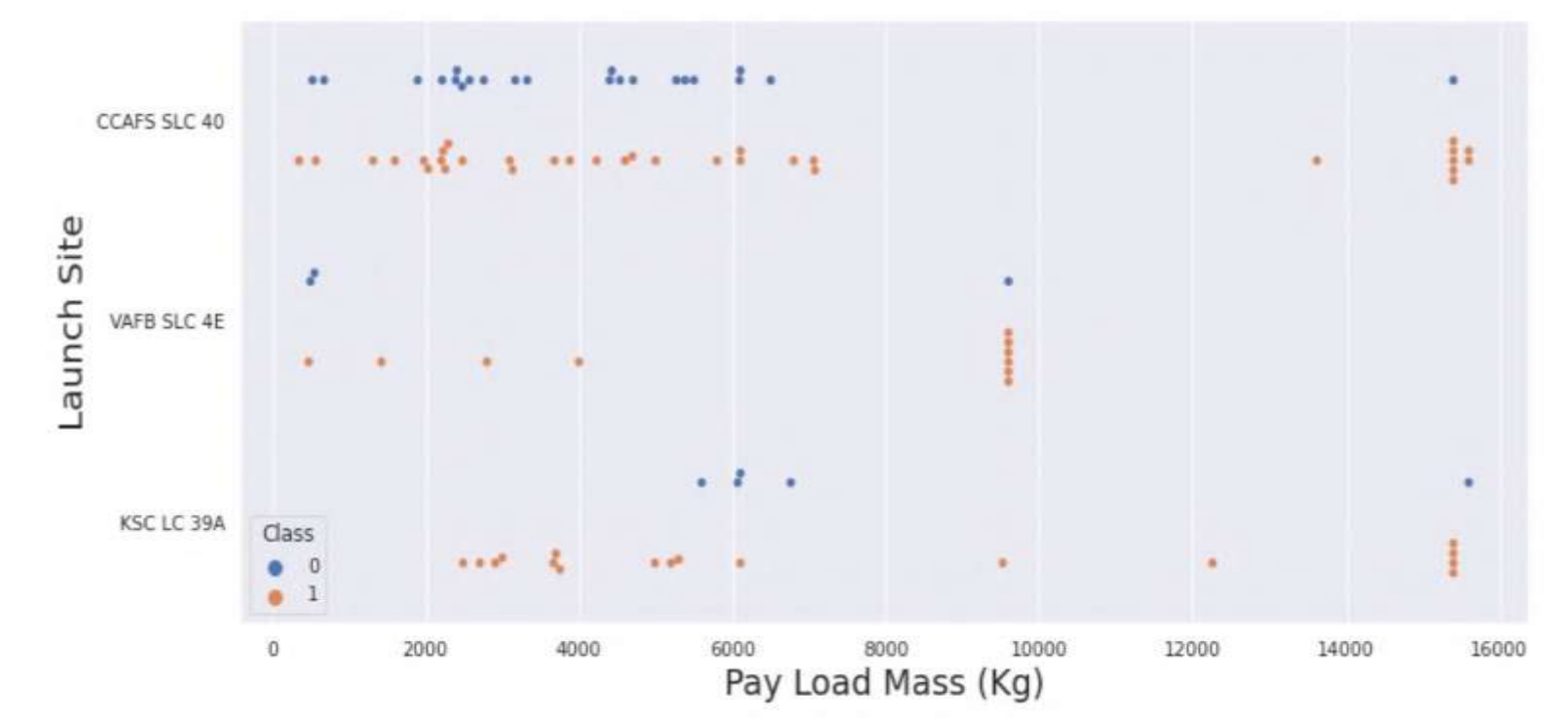

Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site



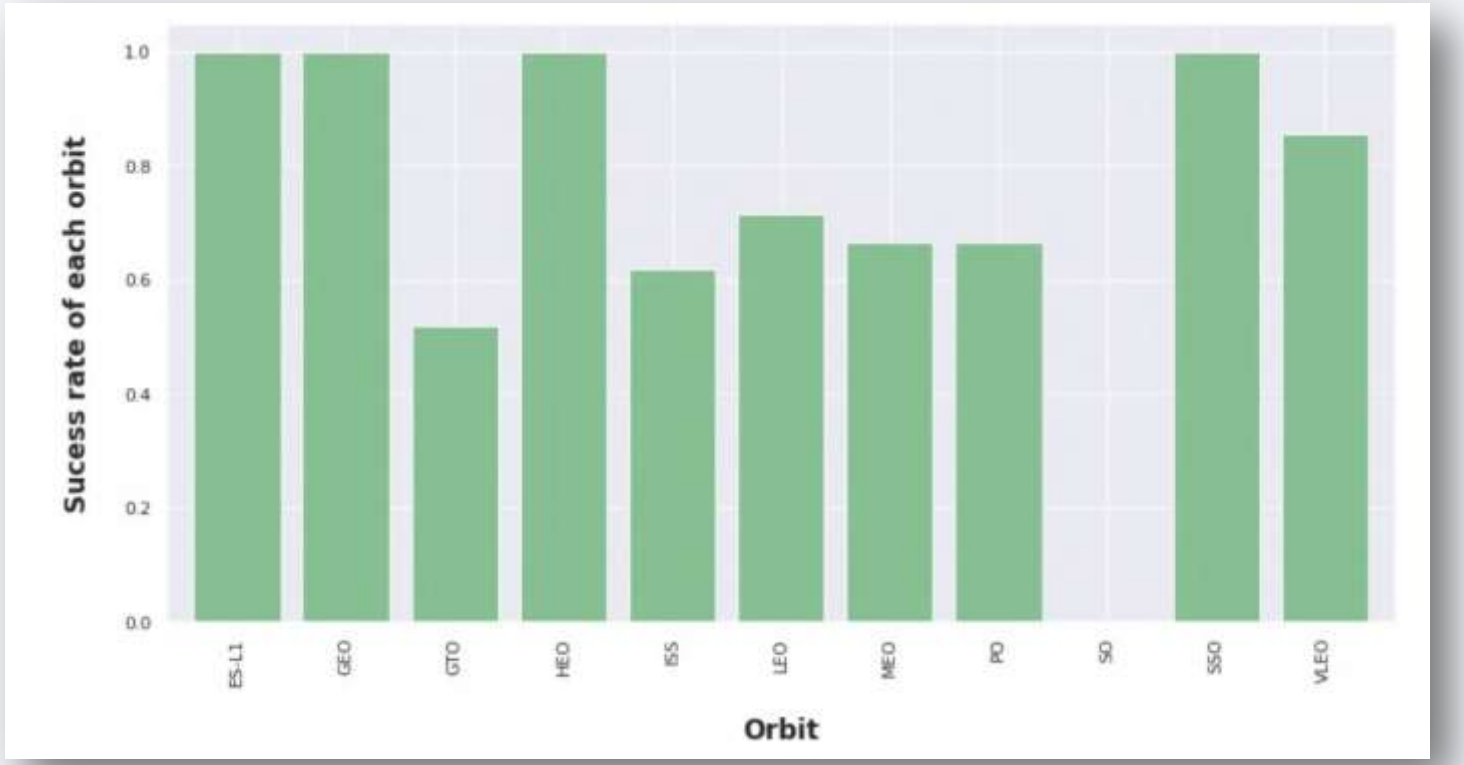
- This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be.
- However, site CCAFS SLC40 shows the least pattern of this.

Payload vs. Launch Site



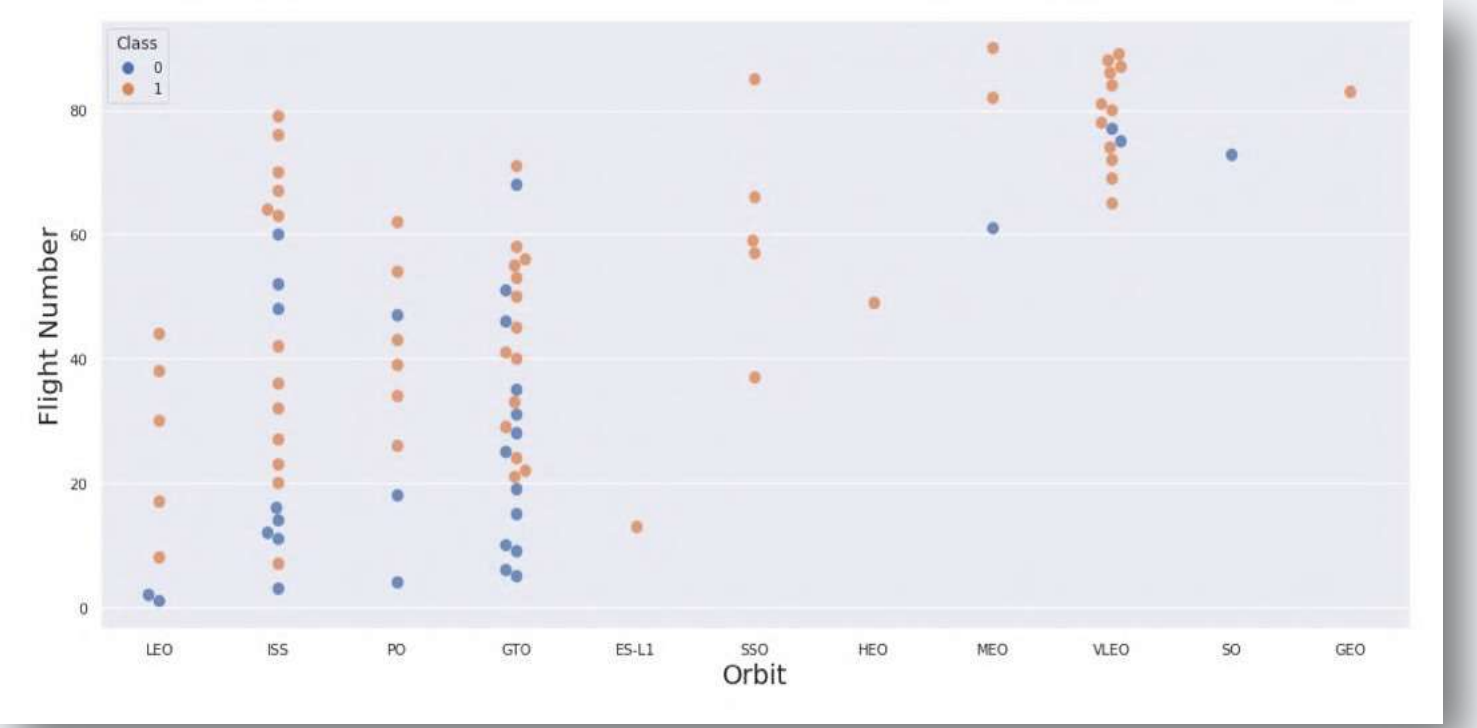
- This scatter plot shows once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.
- However, there is no clear pattern to say the launch site is dependent to the pay load mass for the success rate.

Success Rate vs. Orbit Type



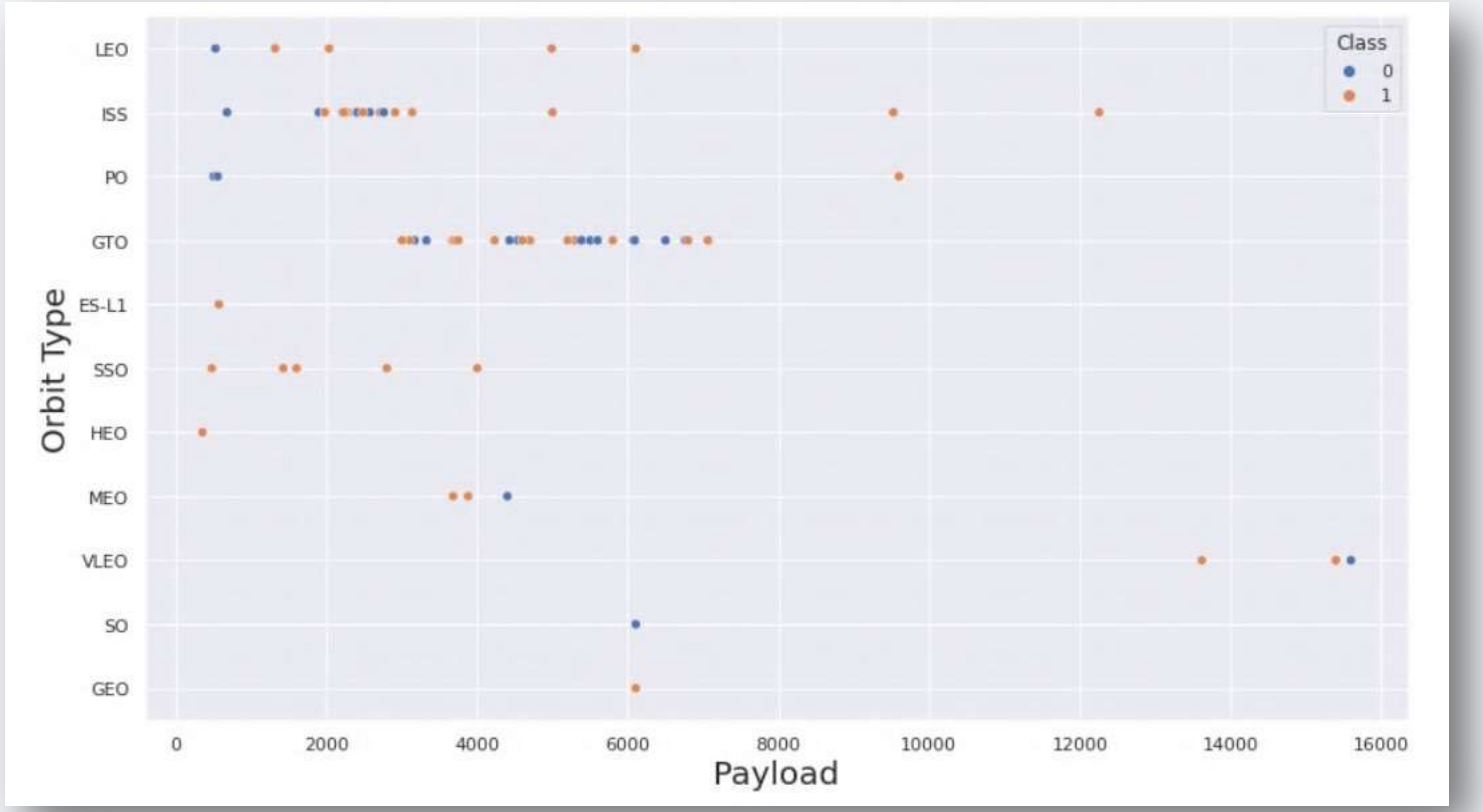
- This figure depicted the possibility of the orbits influencing the landing outcomes, as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1, while SO orbit produced 0% rate of success.
- However, deeper analysis show that some of these orbits have only 1 occurrence such as GEO, SO, HEO and ES-L1, which means this data need more dataset to see pattern or trend before any conclusion can be drawn.

Flight Number vs. Orbit Type



- This scatter plot shows that the larger the flight number on each orbits, the greater the success rate (especially LEO orbit), except for GTO orbit which depicts no relationship between both attributes.
- Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.

Payload vs. Orbit Type



- Heavier payload has positive impact on LEO, ISS and P0 orbit. However, it has negative impact on MEO and VLEO orbit.
- GTO orbit seem to depict no relation between the attributes.
- Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.

Section 3

Launch Sites Proximities Analysis

Location ofall the Launch Sites

All the SpaceX launch sites are located inside the United States

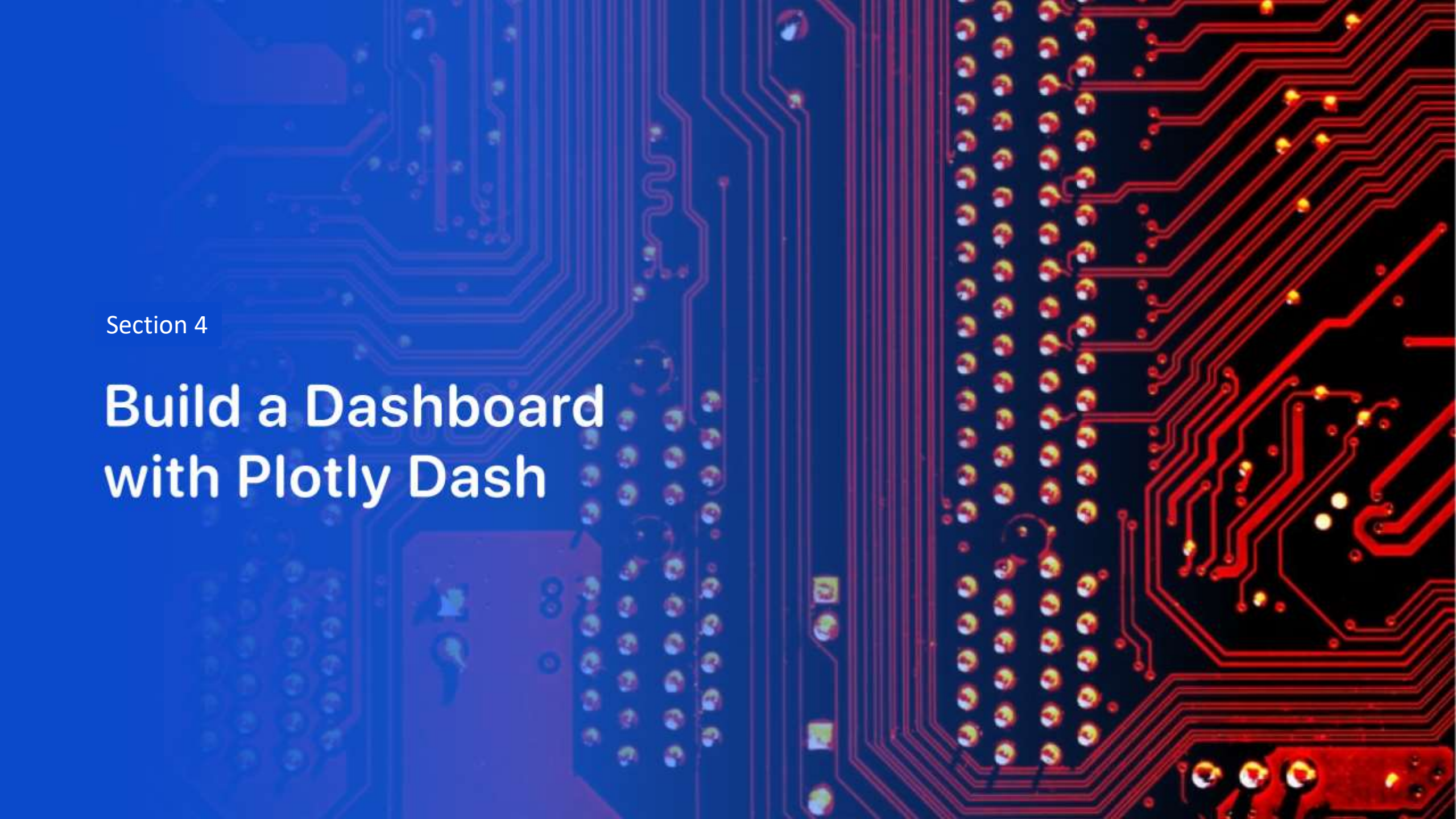
Markers showing launch sites with color labels

Launch Sites Distance to Landmarks

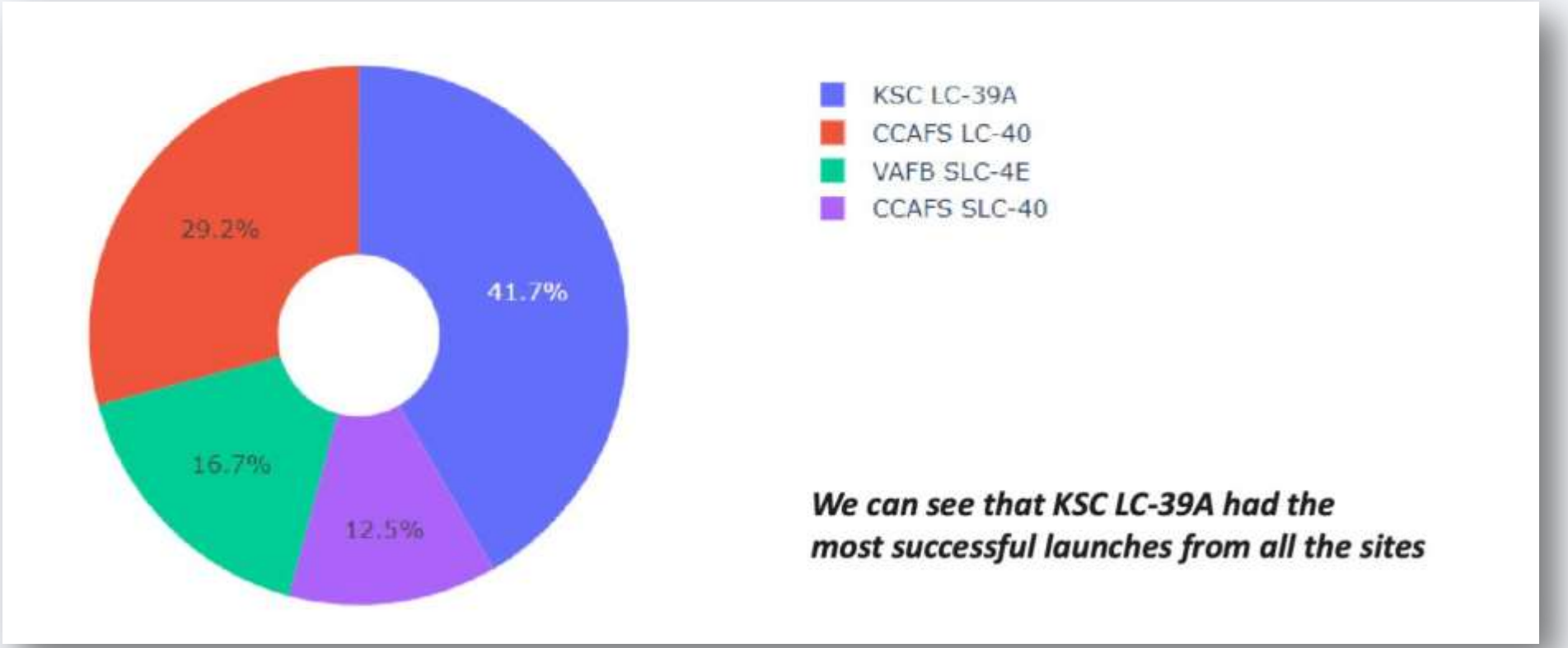
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

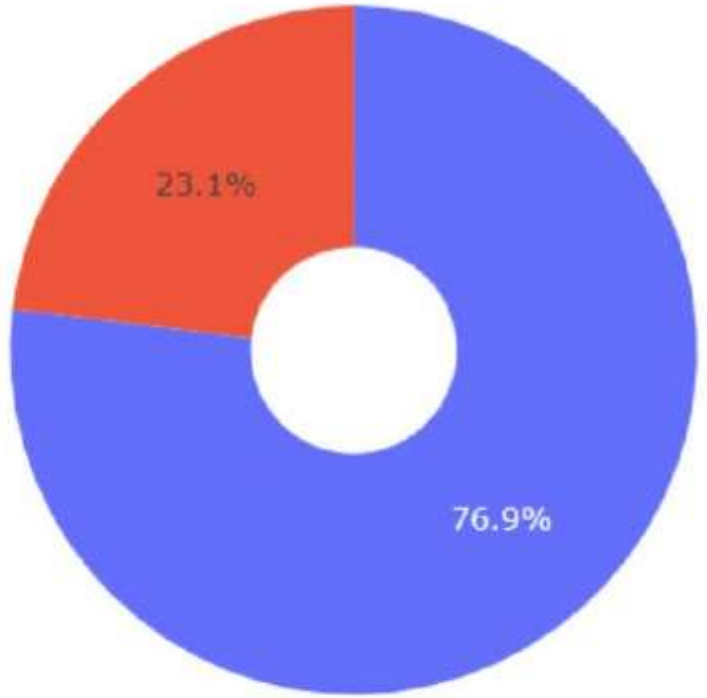
Build a Dashboard with Plotly Dash



The success percentage by each sites.



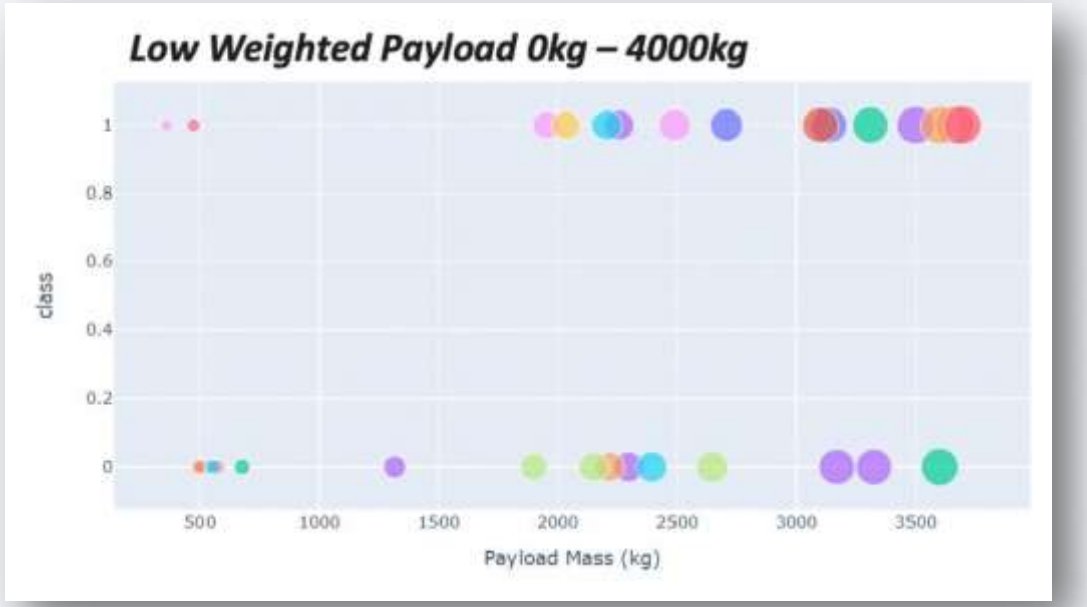
The highest launch-success ratio: KSCLC-39A

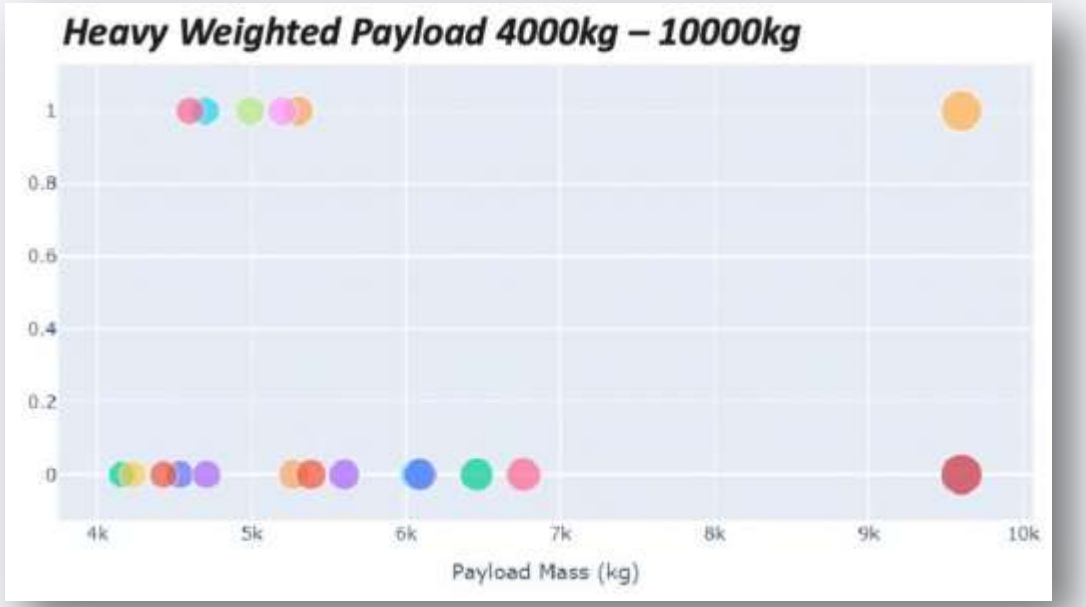


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot

All the success rate for low weighted payload are higher than heavy weighted payload





Section 5

Predictive Analysis (Classification)

Classification Accuracy

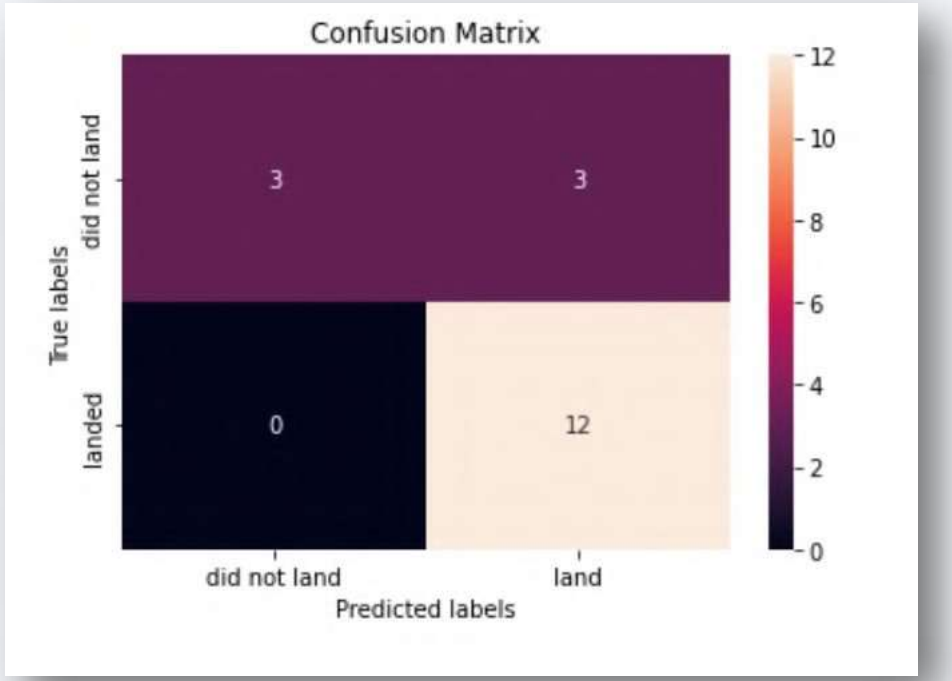
By using the code as below: we could identify that the best algorithm was the Tree Algorithm which have the highest classification accuracy.

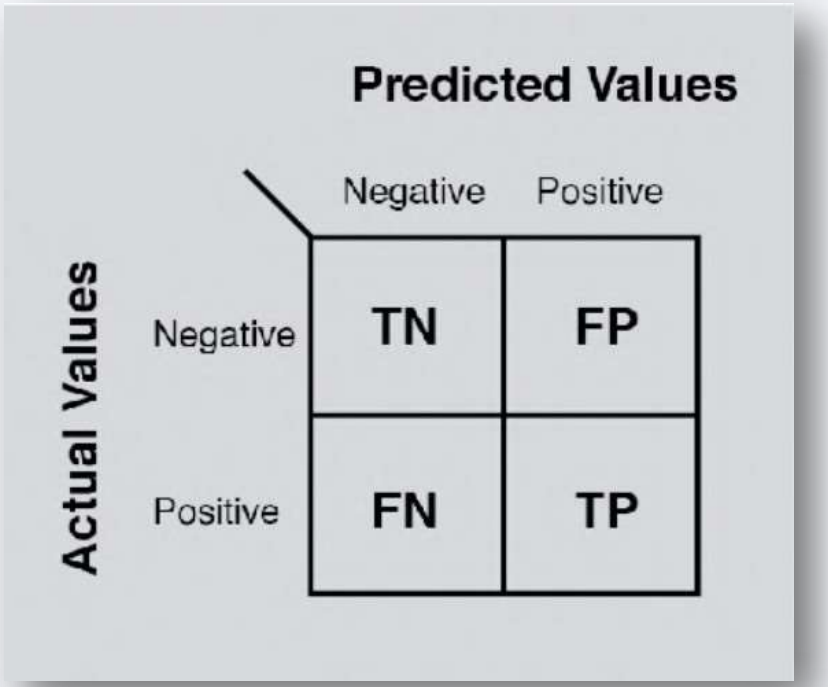
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9017857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_sampl
es_split': 10, 'splitter': 'random'}
```

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





Conclusions

We can conclude that:

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!