Winning Space Race with Data Science

Debanjan Shil
04-04-2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

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 who offer launching rocket specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price down even further. As a data scientist of, 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:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The dataset was collected by REST API and Web Scrapping from Wikipedia

- For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values and fill with whatever needed.

- For web scrapping, we will use the BeautifulSoup 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

```
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)]
```

https://github.com/Deba199/testrepo/blob/master/jupyter_labs_spacex_data_collection_api.ipynb

Data Collection - Scraping

Request the Falcon 9 URL data from wikipedia

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

Create a BeautifulSoup from the HTML response

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

https://github.com/Deba199/testrepo/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

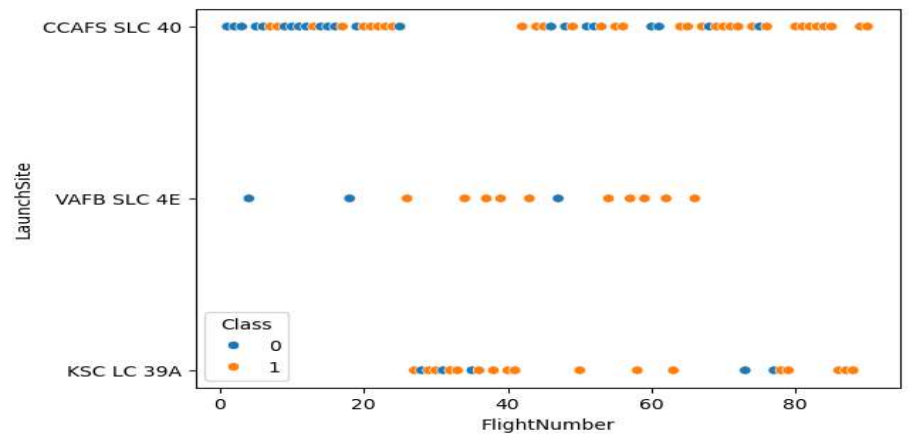
- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

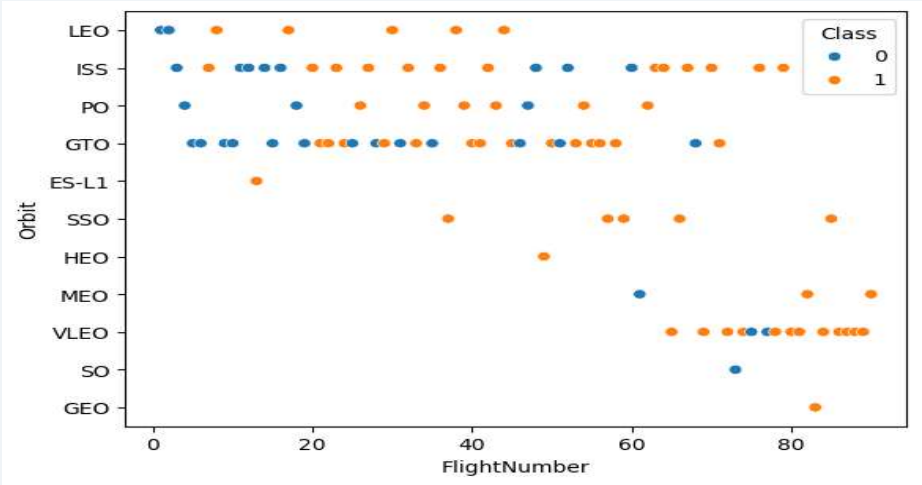
- We first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type.

- We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

https://github.com/Deba199/testrepo/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization





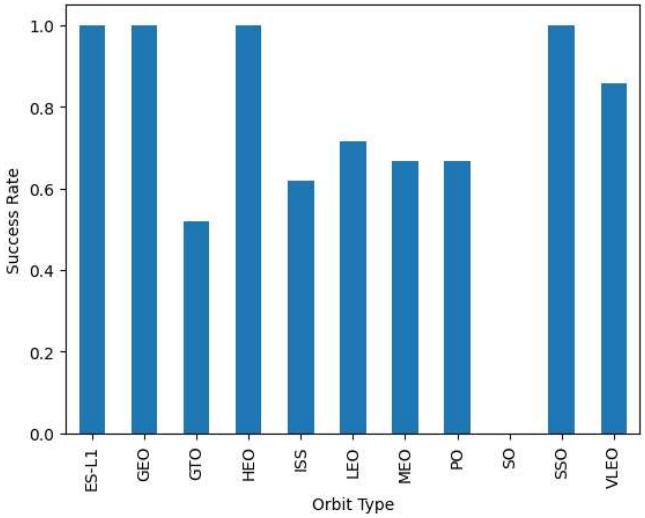
We first started by using scatter graph to find the relationship between the attributes such as between:

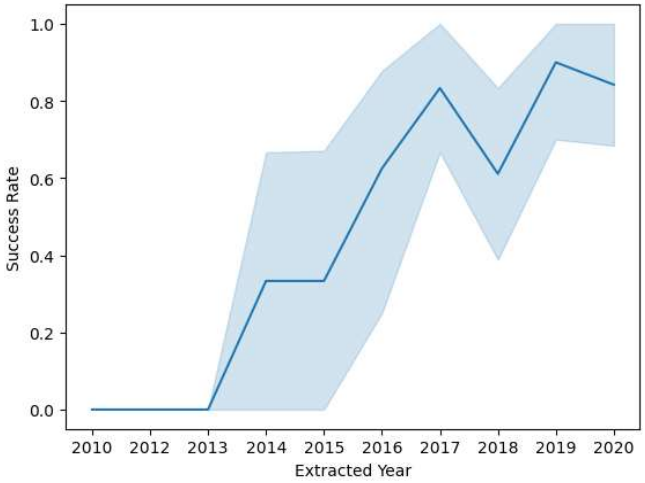
- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

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 affecting the most to the success of the landing outcomes.

https://github.com/Deba199/testrepo/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with Data Visualization





Once we get a hint of the relationships using scatter plot. We will then use further visualization tools such as bar graph and line plots graph for further analysis.

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

We then use the line graph to show a trends or pattern of the attribute over time which in this case, is used for see the launch success yearly trend.

We then use Feature Engineering to be used in success prediction in the future module by created the dummy variables to categorical columns.

https://github.com/Deba199/testrepo/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Using SQL we performed exploratory data analysis to understand dataset:

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string 'CCA'.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship 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 failed landing outcomes in drone ship, their booster versions, and launch sites names in year 2015
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/Deba199/testrepo/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance: - How close the launch sites with railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?

https://github.com/Deba199/testrepo/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

https://github.com/Deba199/testrepo/blob/master/Interactive%20Dashboard%20with%20Ploty%20Dash

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

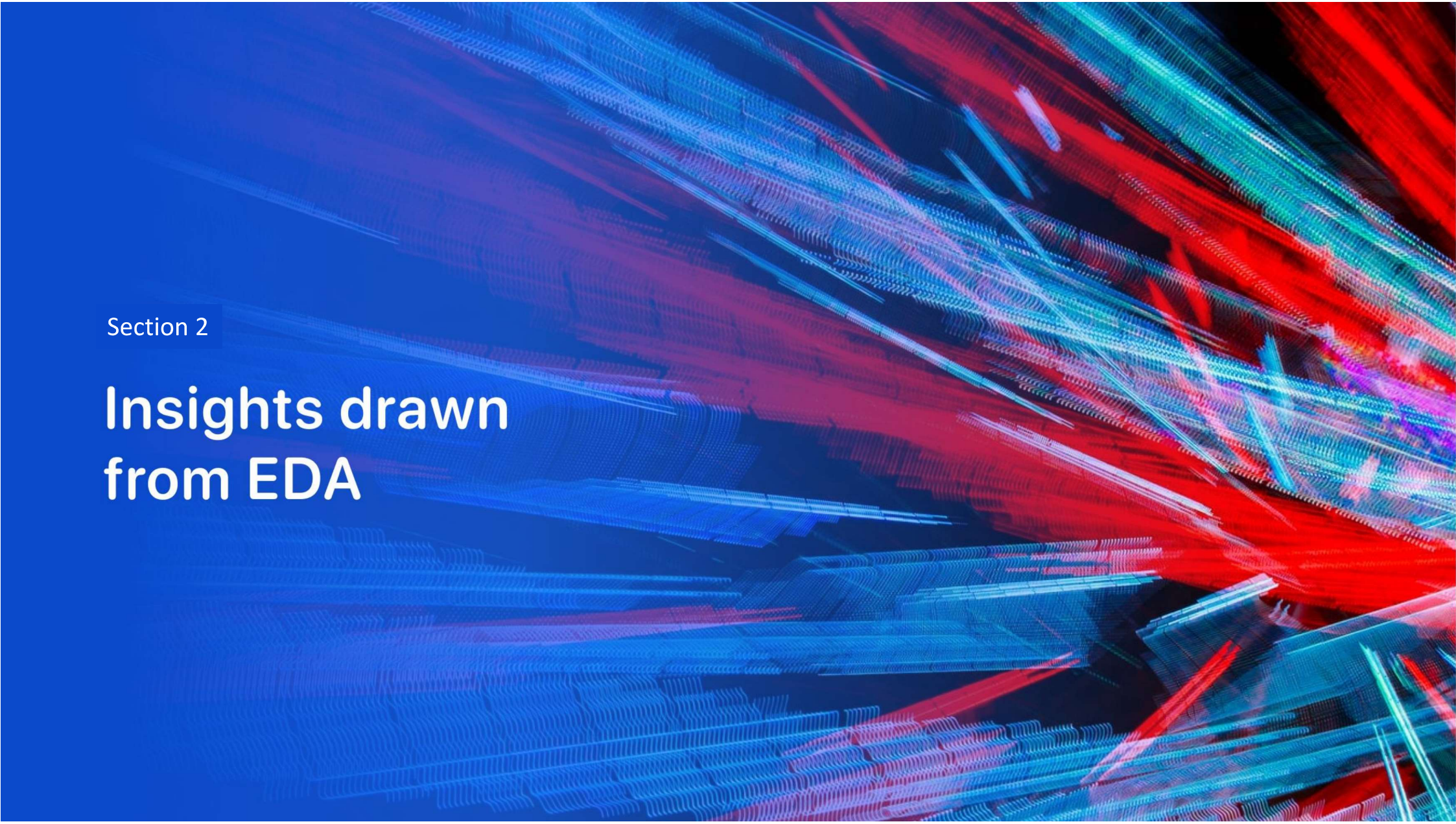
Find the Best Model

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

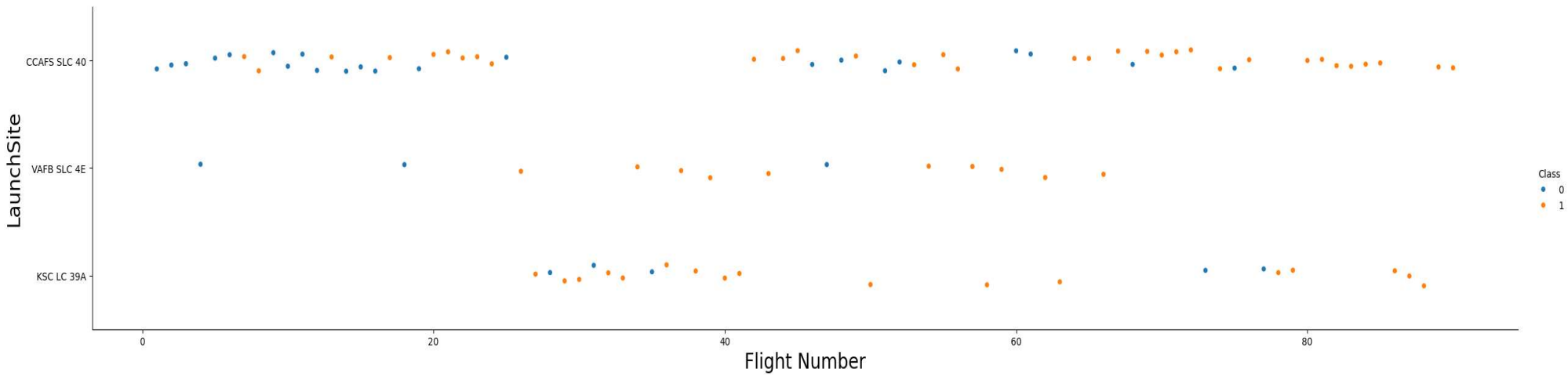
https://github.com/Deba199/testrepo/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- The results will be categorized to 3 main results which is:
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

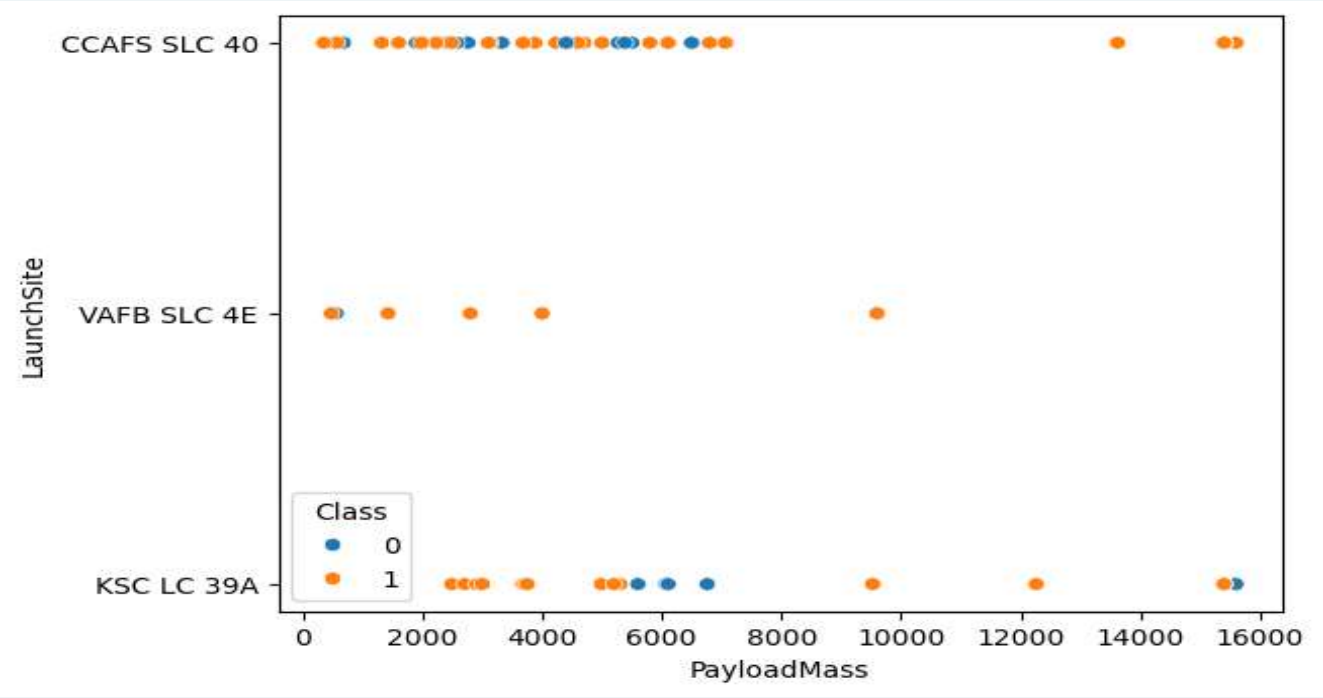

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



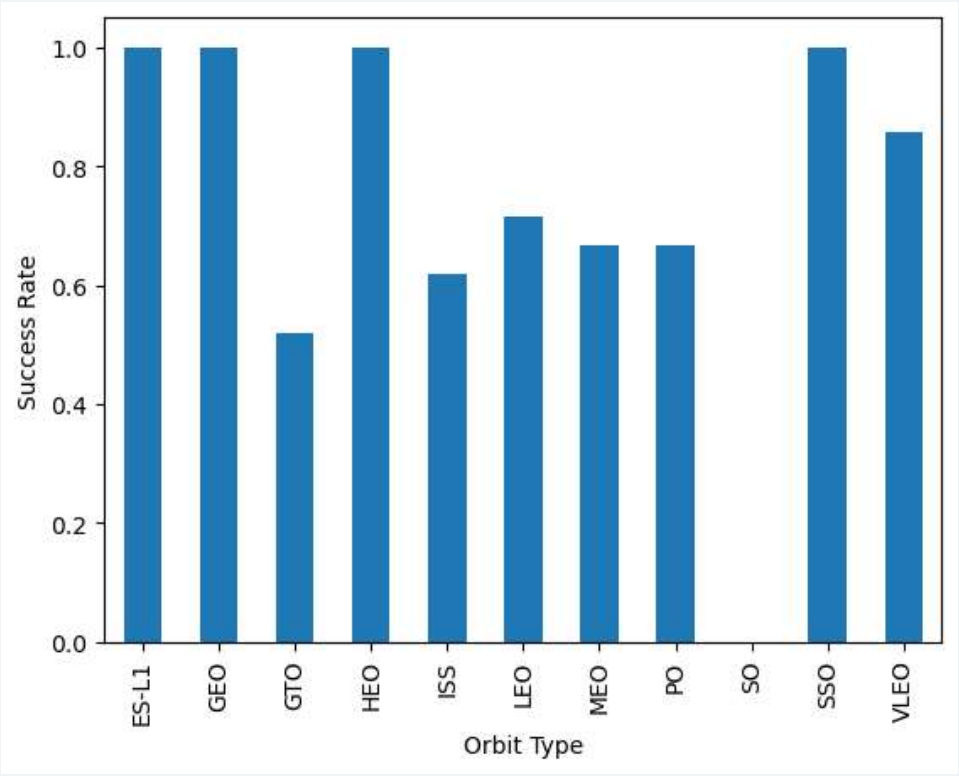
This scatter plot shows that the larger the flights amount of the launch site, the greater the success rate will be.
Unlikely site CCAFS SLC40 shows the least pattern of this

Payload vs. Launch Site



This scatter plot shows if the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased. However there is no clear pattern visible between LaunchSite and PayloadMass

Success Rate vs. Orbit Type

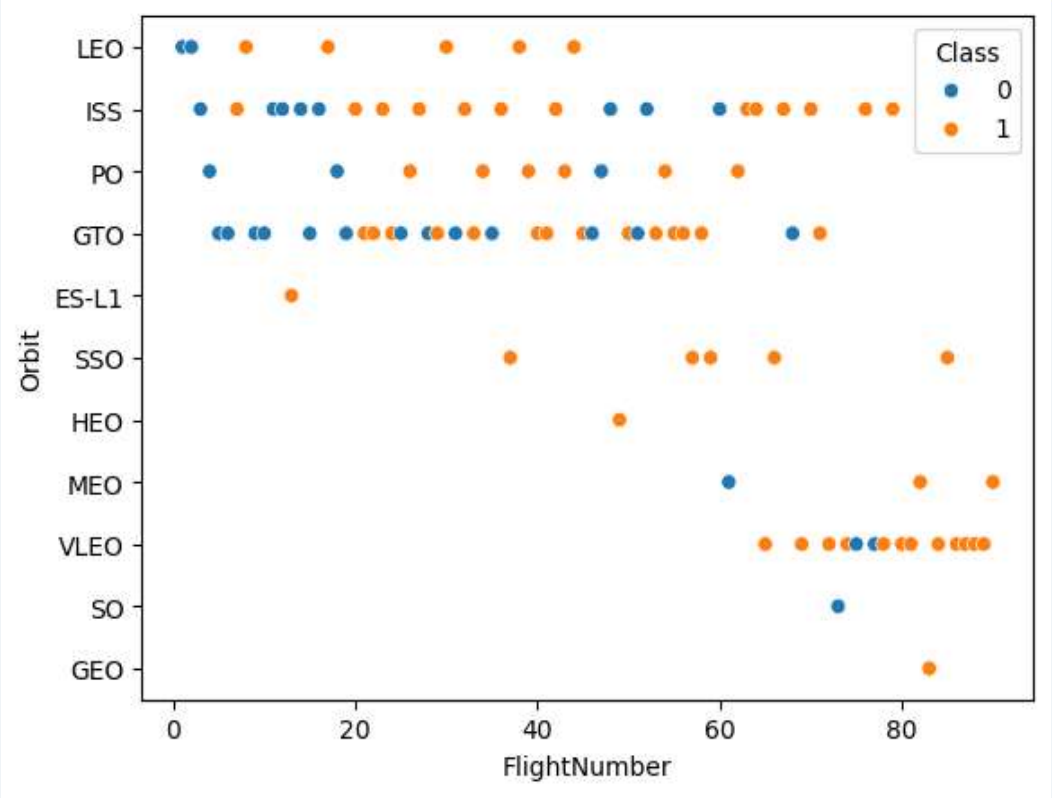


- ES-L1, GEO, HEO and SSO have a success rate of 100%, however, SO has a success rate of 0%

• To deeper understanding, we need more dataset to know the trend or pattern of the orbits types

This bar plot show the correlation between type of orbits and their success rate

Flight Number vs. Orbit Type

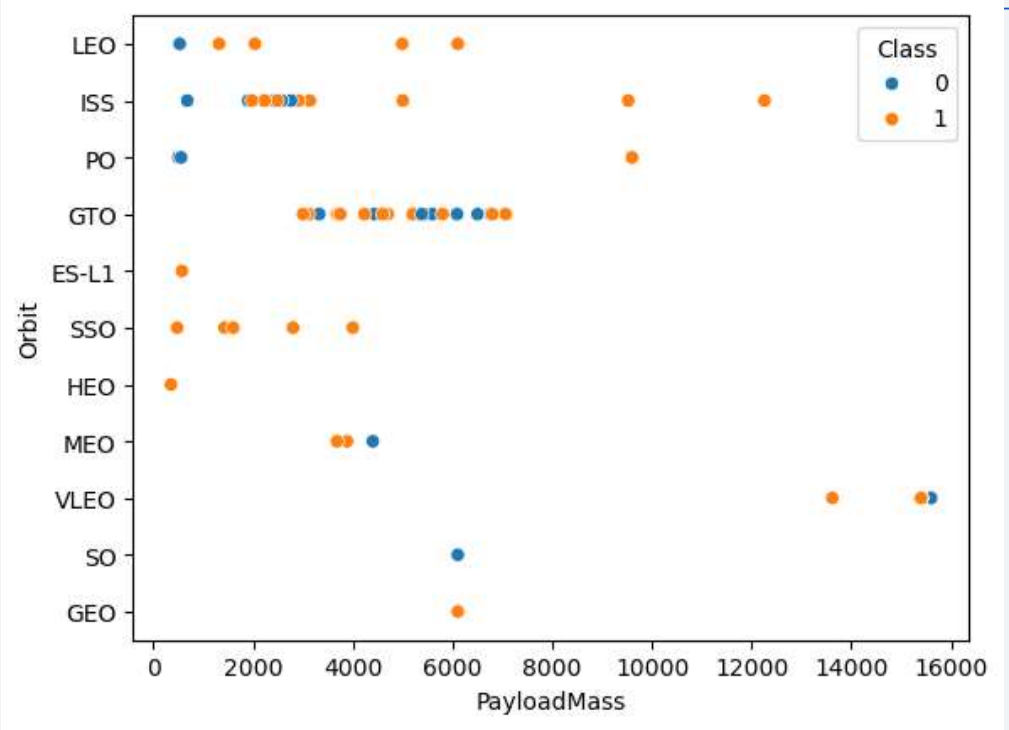


- The larger the flight numbers on each orbits, the greater the success rate especially LEO.

- There is no relationship between Flight number and Orbit

- Orbit that has only 1 occurrence maybe given the less accuracy results, however, we need more dataset for conclusion

Payload vs. Orbit Type

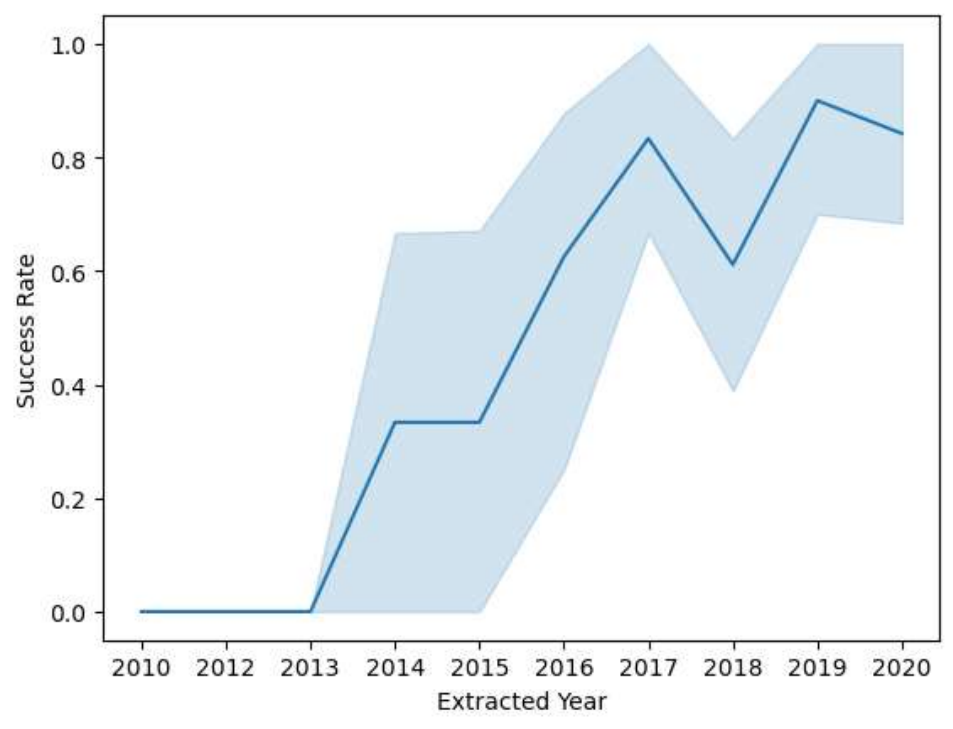


- Heavier payload mass has positive impact on LEO, ISS and PO orbit, but it has negative impact on MEO and VLEO orbit.

- There is no relation between orbit type and Payload mass for GTO orbit

- Meanwhile, SO, GEO, HEO orbit need more data to understand their pattern

Launch Success Yearly Trend



There is an increasing trend from the year of 2013 to 2020 for Space X Rocket Success Rate, although in 2018, the success rate have a slightly decrease.

All Launch Site Names

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with `CCA`

```
%sql select Launch_Site from SPACEXTBL where Launch_Site like '%CCA%' limit 5;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum(PAYLOAD_MASS__KG_) as TotalPayLoadMass from SPACEXTBL Group By Customer = '%NASA(CRS)%';
```

```
 * sqlite:///my_data1.db
Done.
```

TotalPayLoadMass
619967

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql Select Booster_Version, AVG(PAYLOAD_MASS__KG_) from SPACEXTBL Where Booster_Version like '%F9 V1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1 B1003	2534.6666666666665

First Successful Ground Landing Date

- Here we use the min() function to find the result
- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql SELECT MIN(Date) AS FirstSuccessfulLandingDate FROM SPACEXTBL WHERE Landing_Outcome LIKE '%Success (ground pad)%';
```

```
 * sqlite:///my_data1.db
Done.
```

FirstSuccessfulLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Weused the WHERE clause for filter the boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql Select Booster_Version from SPACEXTBL Where (PAYLOAD_MASS__KG_ between 4000 and 6000) and \
Landing_Outcome like '%Success (drone ship)%';
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

We used wildcard like filter for **WHERE** Mission Outcome was a success or a failure

```
%sql SELECT Count("MISSION_OUTCOME") AS "SUCCESSFUL MISSION" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * sqlite:///my_data1.db
Done.
```

SUCCESSFUL MISSION
100

```
%sql SELECT Count("MISSION_OUTCOME") AS "FAILURE MISSION" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * sqlite:///my_data1.db
Done.
```

FAILURE MISSION
1

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
%sql SELECT Distinct Booster_Version as "Booster_Versions Which have Carried the Maximum Payload" from SPACEXTBL where \
PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Versions Which have Carried the Maximum Payload
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

We used a combinations of the WHERE clause, LIKE, AND conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql Select substr(Date, 6,2) as Month, substr(Date,0,5) as Year, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL \
where substr(Date, 0, 5) = '2015' and Landing_Outcome like '%Failure (drone ship)%' group by Date;
```

```
 * sqlite:///my_data1.db
Done.
```

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2017-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%sql Select Date , Landing_Outcome , Count(Landing_Outcome) as Count from SPACEXTBL \
Where Date between ' 2010-06-04' and '2017-03-20' group by Landing_Outcome order by Count desc;
```

```
 * sqlite:///my_data1.db
Done.
```

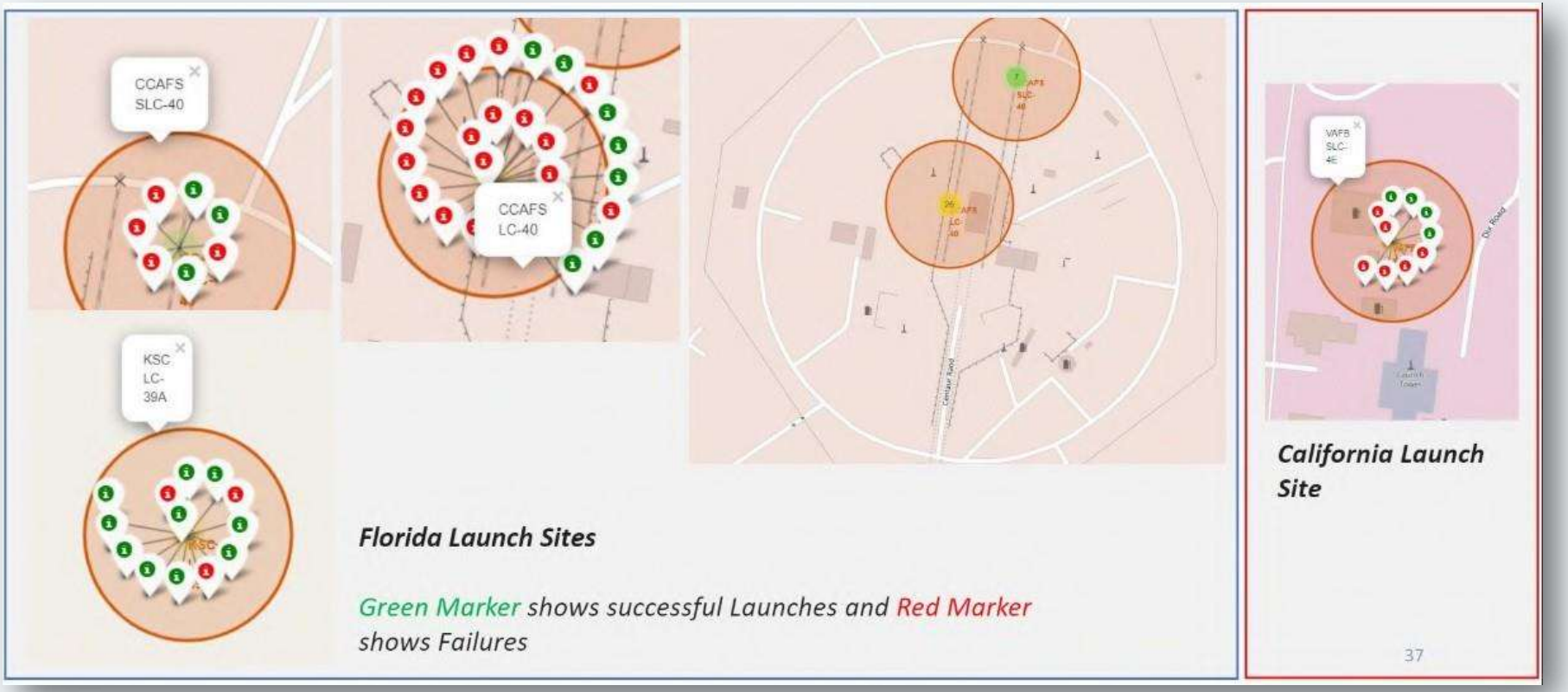
Date	Landing_Outcome	Count
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

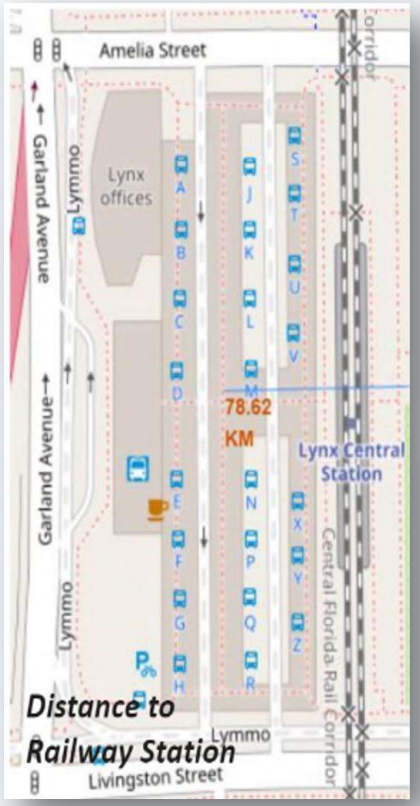
All Launch Sites Location Markers

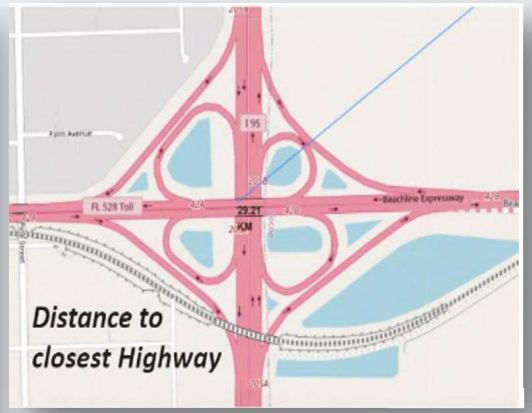
We can see that all the SpaceX launch sites are located inside the United States

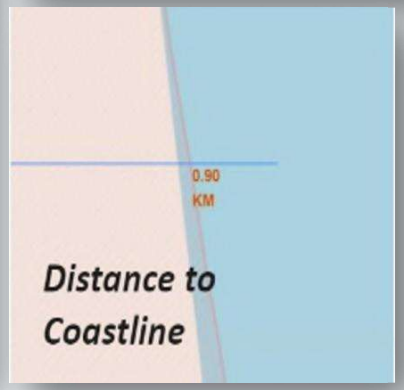
Color labeled Launch Outcomes



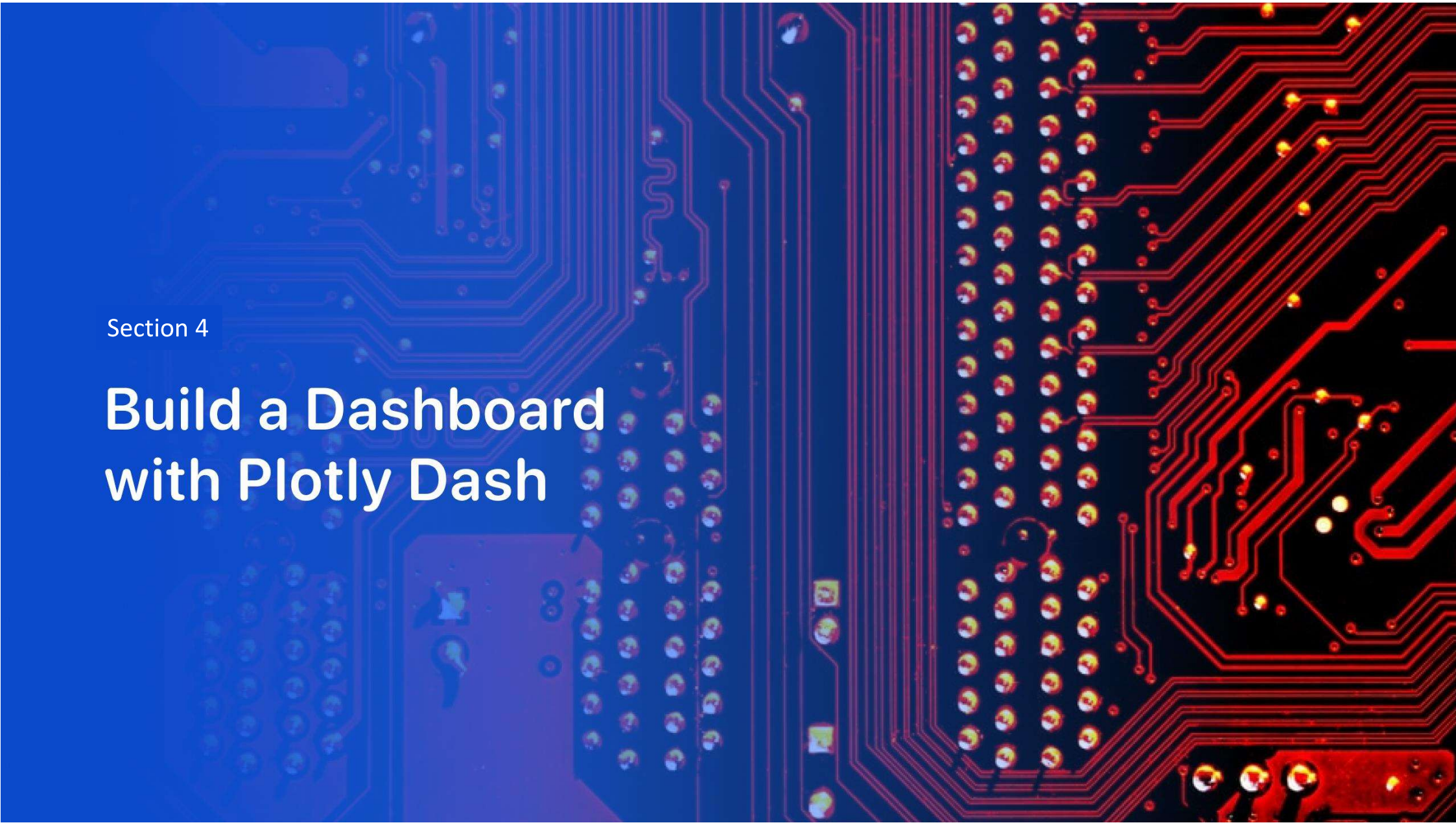
Launch Sites Distance to Landmarks



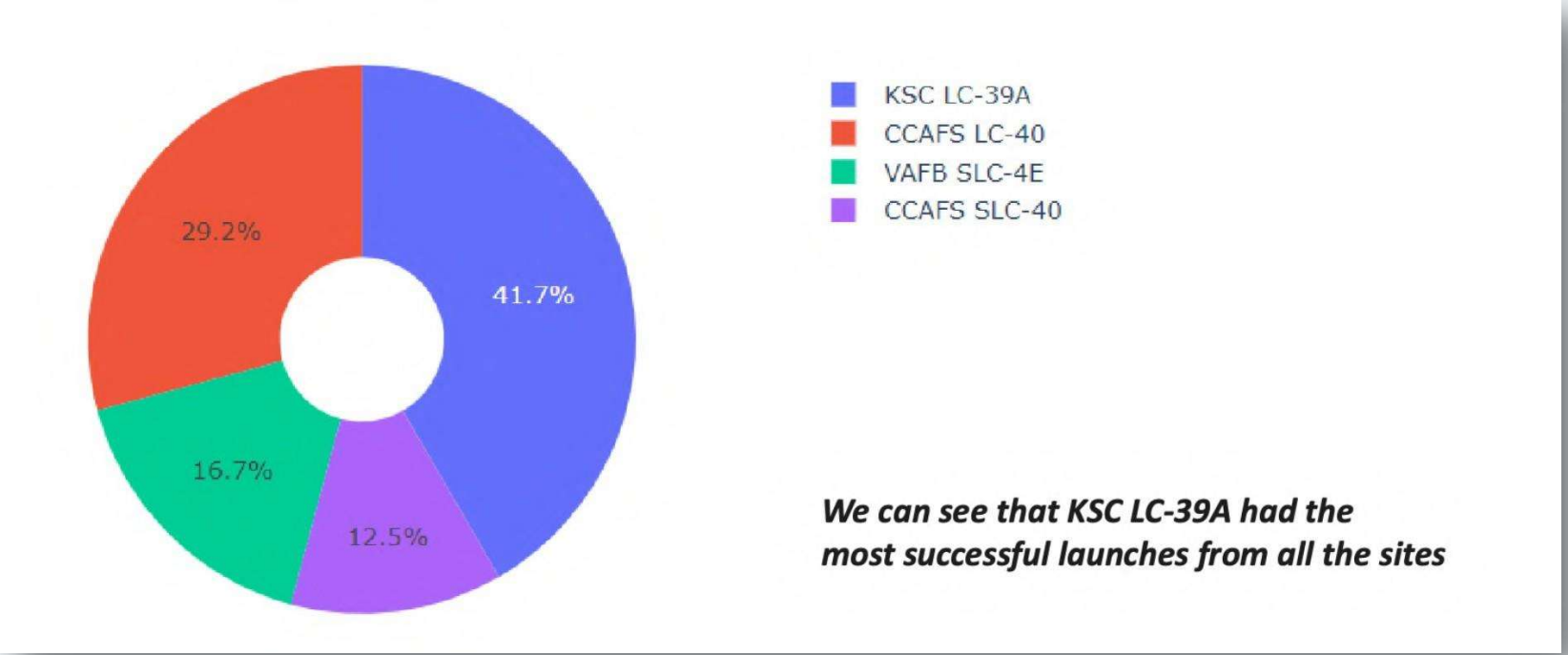




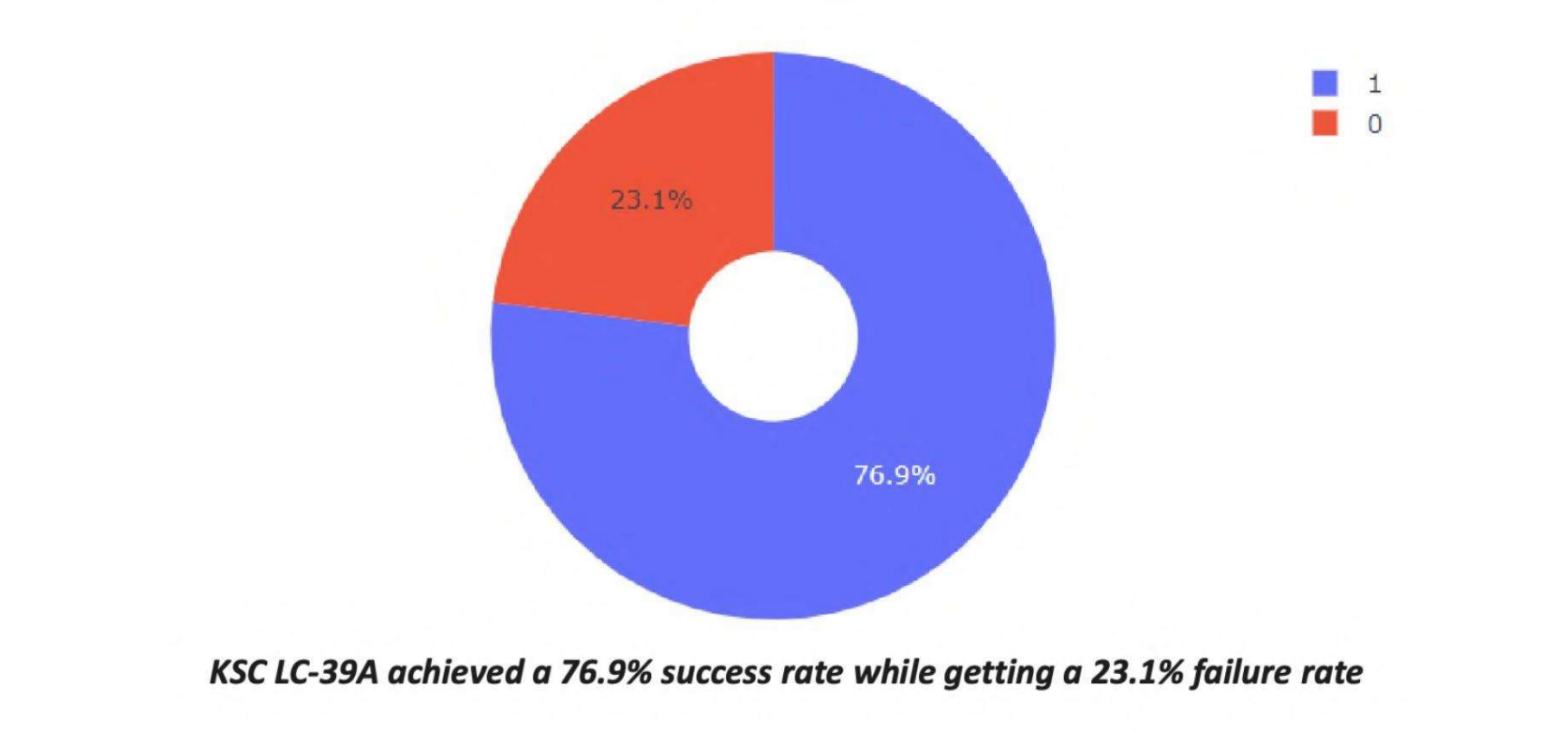
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites



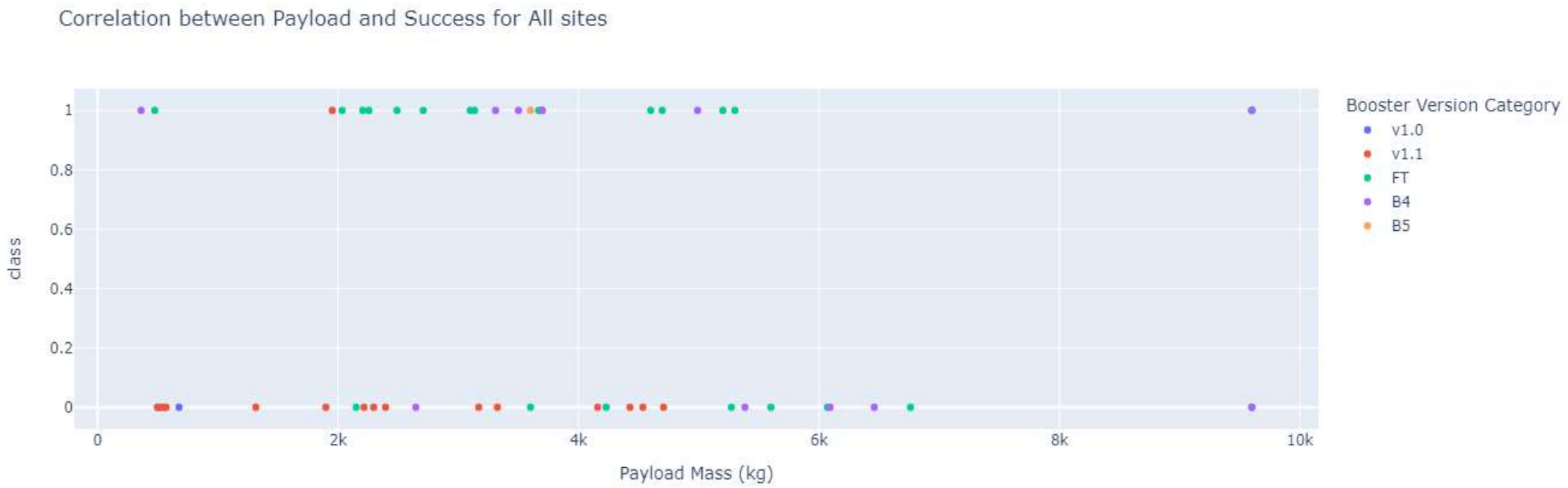
The highest launch-success ratio: KSC LC-39A



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

Payload vs Launch Outcome Scatter Plot

We can see that all the success rate for low weighted payload is higher than heavy weighted payload



Section 5

Predictive Analysis (Classification)

Classification Accuracy

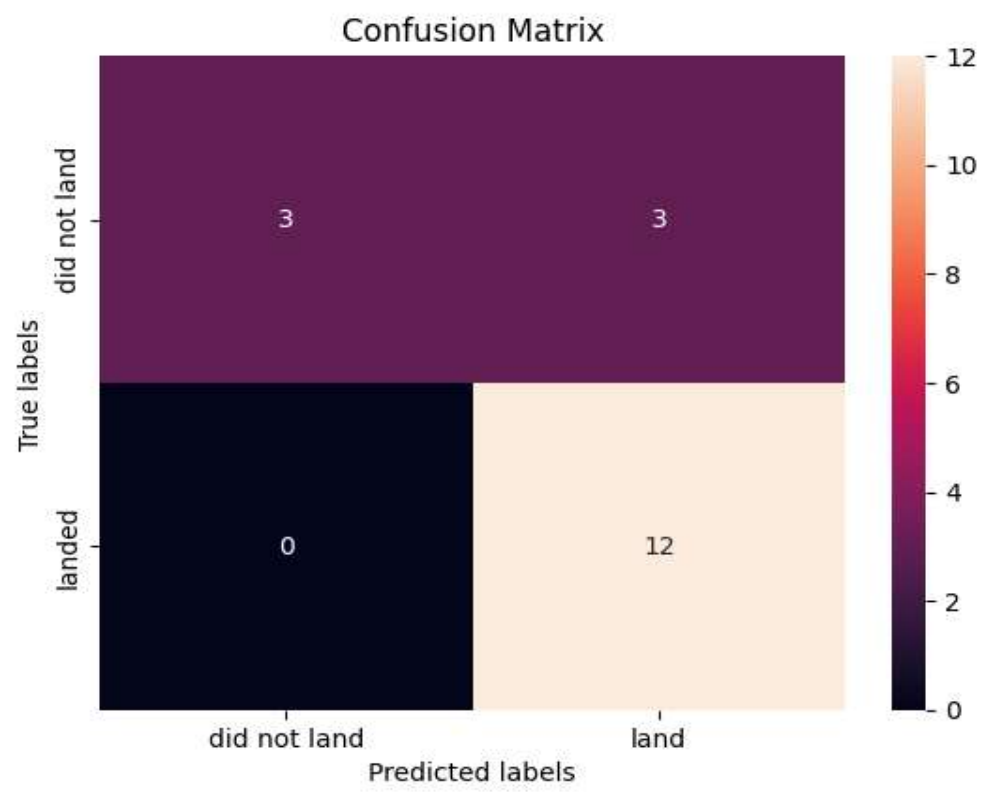
As we can see the best algorithm to be Tree Algorithm which have the highest classification accuracy

```
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=='KNN':
    print('Best param is:' , knn_cv.best_params_)
if bestalgorithm=='Tree':
    print('Best param is:' , tree_cv.best_params_)
if bestalgorithm=='LogisticRegression':
    print('Best param is:' ,logreg_cv.best_params_)
```

```
best algorithm is Tree with a score of 0.8625
Best param is: {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, '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.
- KSCLC-39A have the most successful launches of any sites; 76.9%
- SSOorbit have the most success rate; 100% and more than 1 occurrence.

Thank you!