IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Urvashi Devi>
<24 Nov 2023>

Outline

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

Executive Summary

- **Summary of methodologies : below methodologies were used to analyze and predict the Falcon9 first stage:**
- i) Data collection from an SpaceX API and web scrap to collect the Falcon9 historical launch records from a wikipedia page and parse the data and convert it to a dataframe.
- ii) Data wrangling to deal with null or missing values in the data and cleaning the data for further analysis
- iii) Exploratory data analysis using SQL and data visualization. Some attributes in the SpaceX data were determined if the first stage can be reused. These features can be used with machine learning to automatically predict if the first stage can land successfully. Various charts were created to explore the relationship between the attributes.
- iv) Interactive map using Folium to perform analysis on the launch site locations. SpaceX launch data was visualized to discover some preliminary correlation between the launch site and success rate.
- v) First stage Landing Prediction by creating a machine learning pipeline to predict if the first stage will land given the data
- **Summary of all results**
- As per the data present on the SpaceX API and historical data present on the SpaceX website, machine learning model was trained to determine whether the first stage will land which in turn will help in calculating the price of a launch. Various machine learning model were trained using the available data and method which perform best was found. All methods practically perform same.

Introduction

- **Project background and context:**

SpaceX is a revolutionary company who has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving thanks to SpaceX astounding idea 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 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:

Falcon 9 data was collected using Space X REST API and web scraping for the Falcon9 launch HTML table from wikipedia.

Perform data wrangling

Data was processed using one-hot encoding for categorical features. Data wrangling was done to deal with null or missing values in the data

Perform exploratory data analysis (EDA) using visualization and SQL

Exploratory data analysis using SQL and data visualization. Some attributes in the SpaceX data were determined if the first stage can be reused. These features can be used with machine learning to automatically predict if the first stage can land successfully. Various charts were created to explore the relationship between the attributes.

Perform interactive visual analytics using Folium and Plotly Dash

Interactive map using Folium to perform analysis on the launch site locations. SpaceX launch data was visualized to discover some preliminary correlation between the launch site and success rate.

Perform predictive analysis using classification models

Machine learning pipeline was created to predict if first stage will land based on the data already provided.

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, its started by using the get request. Then, the response content as Json was decoded and turned it into a pandas dataframe using json_normalize(). Then data was cleaned, checked for missing values and null values.

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 .

GitHub Link:

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api(1).ipynb

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)]
```

Github Link : https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api(1).ipynb

Data Collection - Scraping

Github Link :

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.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')
```

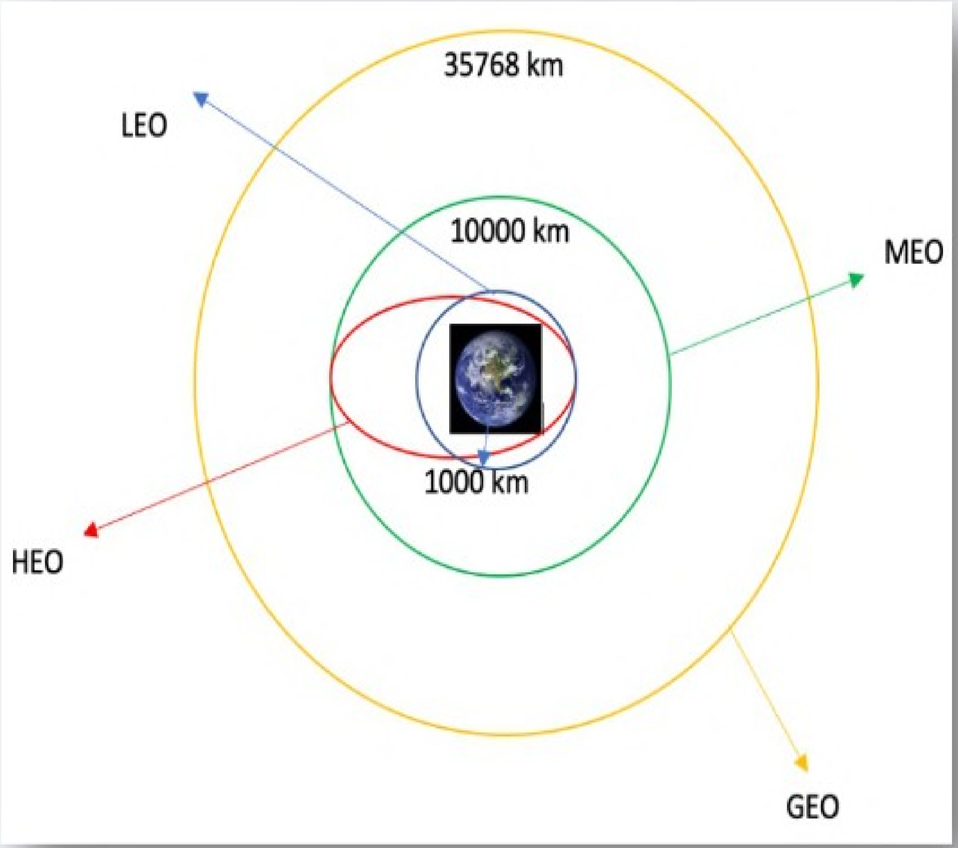
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
       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

Data Wrangling

Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA). Here, first number of launches on each site will be calculated, then the number and occurrence of mission outcome per orbit type will be calculated.

Then a landing outcome label from the outcome column will be created. This will make it easier for further analysis, visualization, and ML. Lastly, the result to a CSV will be exported.

Git Hub link : https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

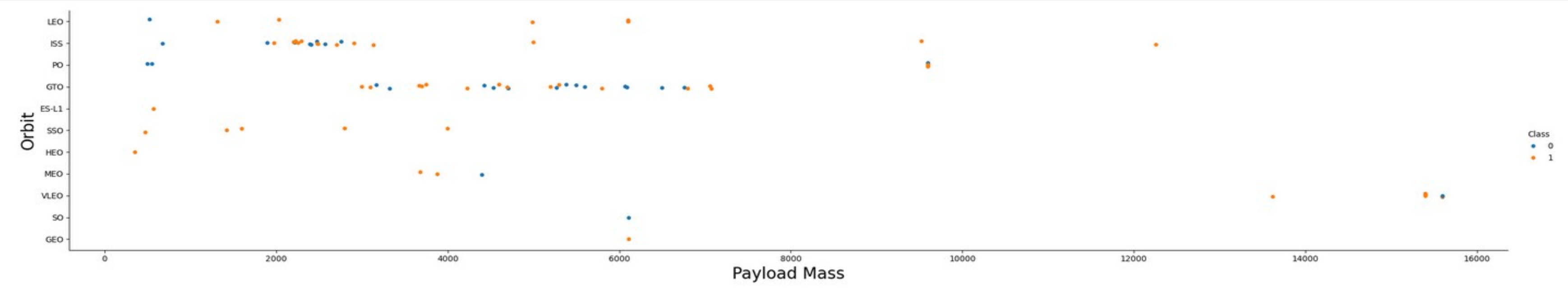
To predict the falcon 9 first stage whether it will land successfully, SpaceX data was analyzed. As mentioned below, various charts were created to explore various attributes and the relationship among them. We first started by using scatter graph to find the relationship between the attributes such 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.

GitHub URL of EDA with data visualization notebook:

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

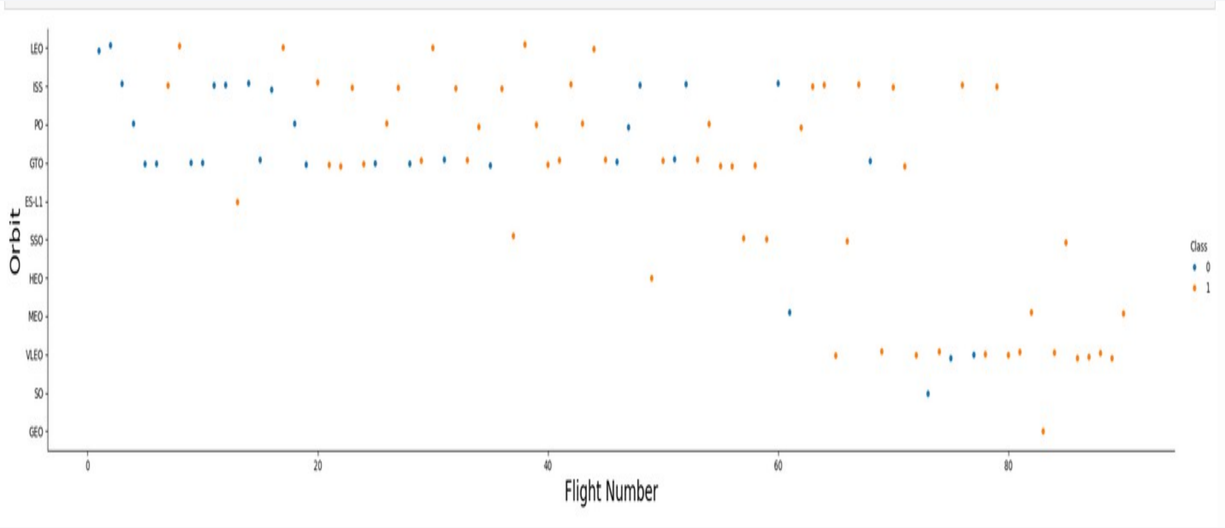


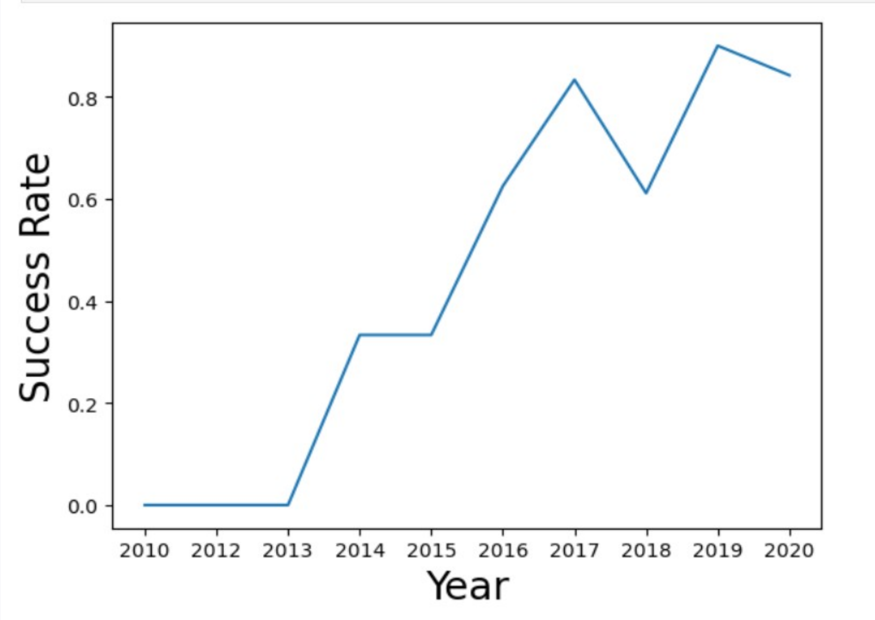
EDA with Data Visualization

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.

GitHub URL of EDA with data visualization notebook:

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

Below is the summary of SQL queries which were performed on the SpaceX dataset to get better understanding of the dataset :

1. Display the names of the unique launch sites in the space mission

Sql query : select distinct(Launch_Site) from SPACEXTABLE

2. Display 5 records where launch sites begin with the string 'CCA

*Sql Query: select * from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5*

3. Display the total payload mass carried by boosters launched by NASA (CRS)

Sql Quey: select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTABLE where Customer='NASA (CRS)'

4. Display average payload mass carried by booster version F9 v1.1

Sql Query: select avg(PAYLOAD_MASS__KG_) as Avg_Payload_Mass from SPACEXTABLE where Booster_Version like 'F9 v1.1%'

5. List the date when the first succesful landing outcome in ground pad was acheived.

Sql Query : select min(date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'

6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Sql Query : select booster_version,PAYLOAD_MASS__KG_, Landing_Outcome from SPACEXTBL where "Landing_Outcome"='Success (dro

EDA with SQL Continued..

7. List the total number of successful and failure mission outcomes

Sql Query : select Mission_Outcome,Count(*) as Total from SPACEXTBL group by Mission_Outcome

8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

Sql Query: select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ in (Select max(PAYLOAD_MASS__KG_) from SPACEX

9. 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.

Sql Query: select substr(Date,6,2)as Month , Landing_Outcome as Failure_Landing_Outcome ,Booster_Version , Launch_Site from SPACEXTBL where Landing_Outcome='Failure (drone ship)' and substr(Date,0,5)='2015'

10. Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

Sql Query: *SELECT "LANDING_OUTCOME", COUNT(*) as 'COUNT' FROM SPACEXTBL WHERE substr(Date,1,4) || substr(Date,6,2) || substr(Date,9,2) between '20100604' and '20170320' GROUP BY "Landing_Outcome" ORDER BY "COUNT" DESC;*

GitHub URL of EDA with SQL notebook :

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximity of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors . On the Folium interactive map , objects such as markers, circles , lines were created to showcase the launched sites .

As part of the lab, more interactive visual analytics using Folium have been created to mark all launch sites and success/failed launches for each site on the map using marker object. Further distance between a launch site to its proximity has been calculated.

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

We then used the Haversine’s formula 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?

External reference (Github URL for Folium Lab) :

https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite(1).ipynb

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard is a plotly dash application which contains input components such as a drop down list and a range slider to interact with a pie chart and a scatter point chart.

Drop down list contains the list of all launched sites and range slider showcases the payload mass range in Kg.
This application is built for users to perform interactive visual analytics on SpaceX launch data in real-time.

We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
• We plotted pie charts showing the total launches by a certain sites.
• We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

GitHub URL of Plotly Dash lab:
https://github.com/urvashisirohi8/Final-Applied-Data-Science-Capstone/blob/main/spacex_dash_app(1).py

Predictive Analysis (Classification)

Machine learning pipeline was created to predict if first stage will land based on the data already provided. To find out the best performing classification model, below steps were performed:

1) A Building the Model

• Load the data set into NumPy and Pandas

• Transform the data and then split into training and test data sets

• Decide which type of ML to use

• set the parameters and algorithms to GridSearchCV and fit it to data set..

2) Evaluating the Model

•Check the accuracy for each model

•Get tuned hyperparameters for each type of algorithms.

•plot the confusion matrix.

3) Improving the Model

• Use Feature Engineering and Algorithm Tuning

4) Find the Best Model

•The model with the best accuracy score will be the sbest performing model

External reference :

https://github.com/urvashisirohi8/Assignment-/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(3).ipynb

Results

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

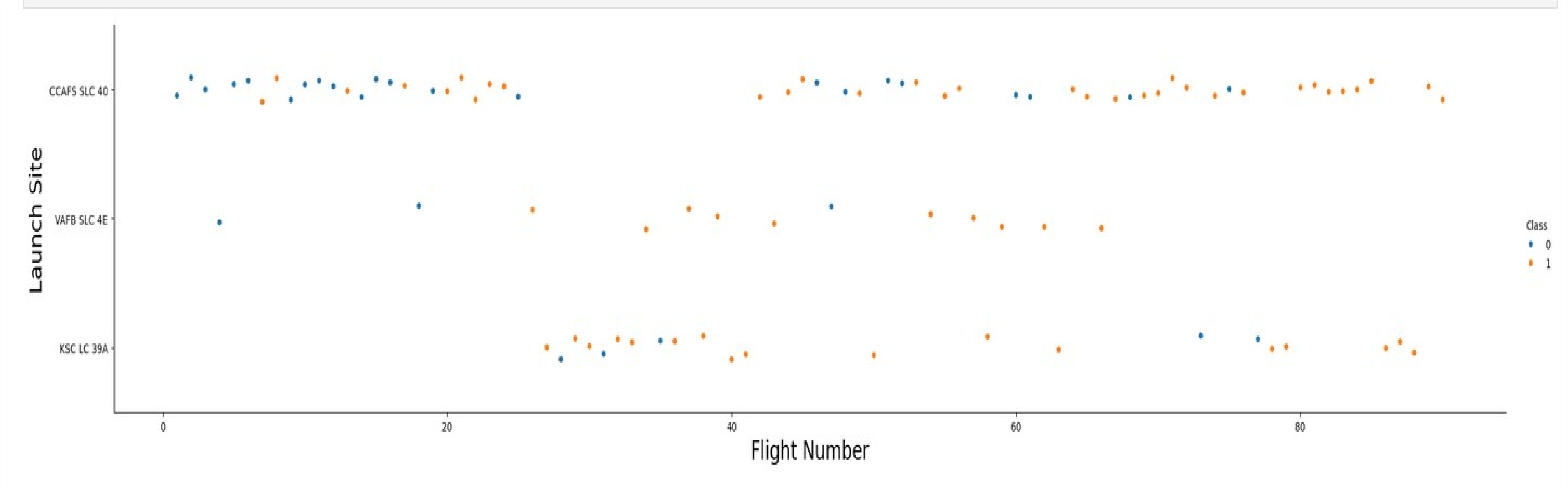
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

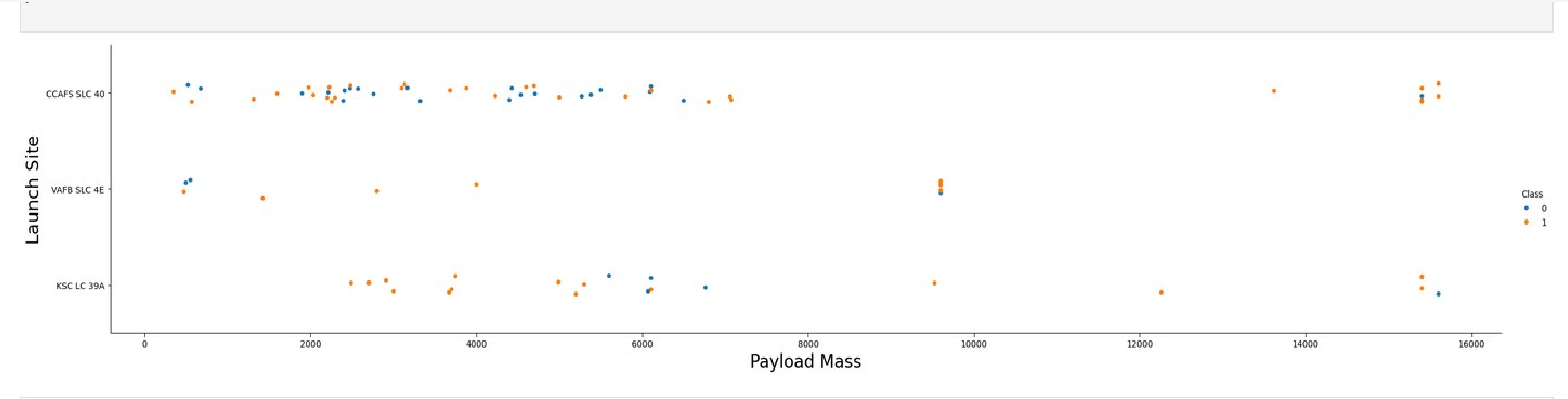
scatter plot of Flight Number vs. Launch Site is shown in the screenshot . This scatter plot shows that the larger the flights amount of the launch site, the greater the success rate will be. Launch site CCAFSSLC 40 shows the least pattern for this.



Payload vs. Launch Site

- Relationship between Payload and Launch Site has been represented through a scatter plot where x axis shows the Payload Mass in Kg and y axis shows the data for Launch Site

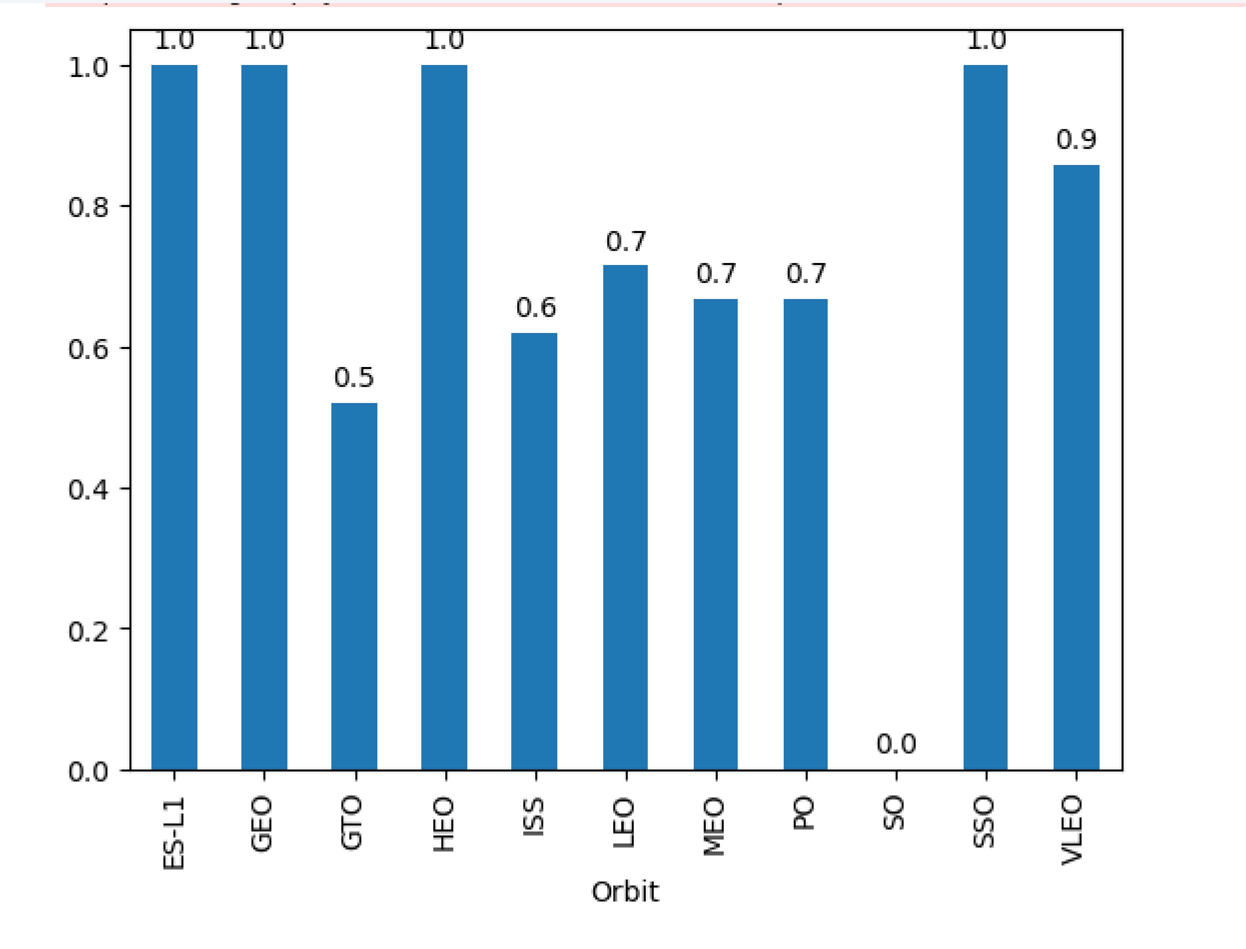
It can be clearly observed in the below screenshot that once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.for the VAFB-SLC launch site, there are no rockets launched for heavy payload mass(greater than 10000).



Success Rate vs. Orbit Type

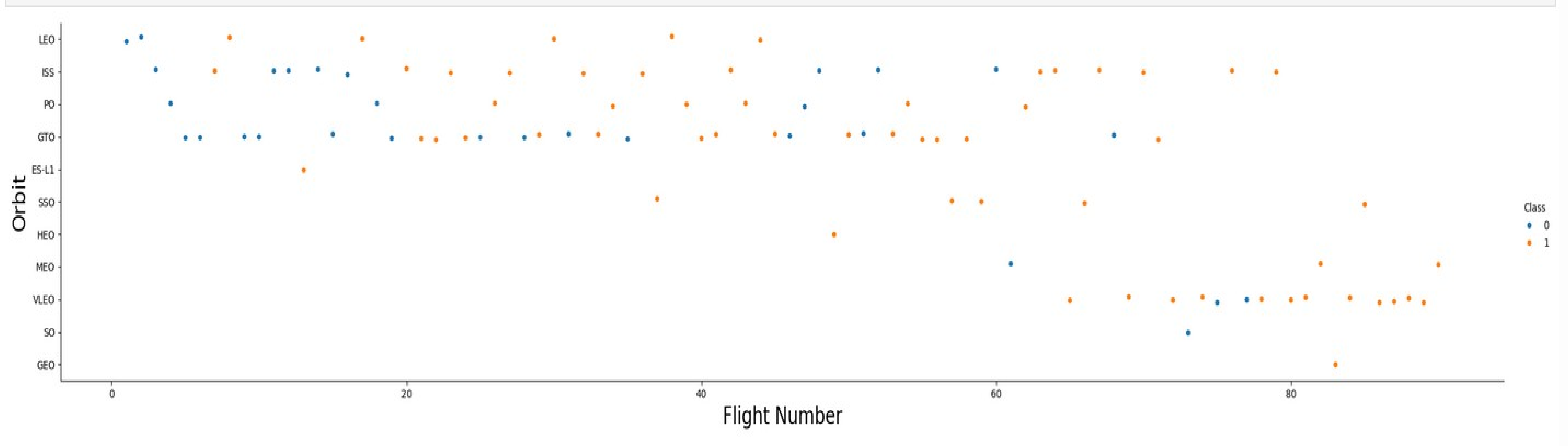
- Here is the bar chart to visualize the relationship between success rate and orbit type.

- It can be visualized his figure depicted the possibility of the orbits to influences the landing outcomes as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced0% rate of success.However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



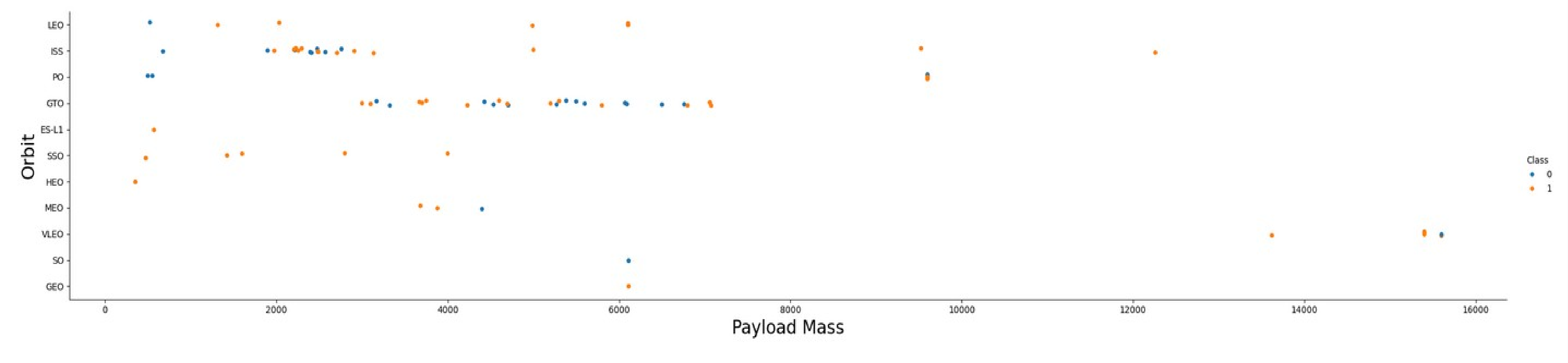
Flight Number vs. Orbit Type

- scatter point of Flight number vs. Orbit type in the below screenshot is represented to showcase the relationship between Flight number and Orbit type.
- It can be observed 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. generally, 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

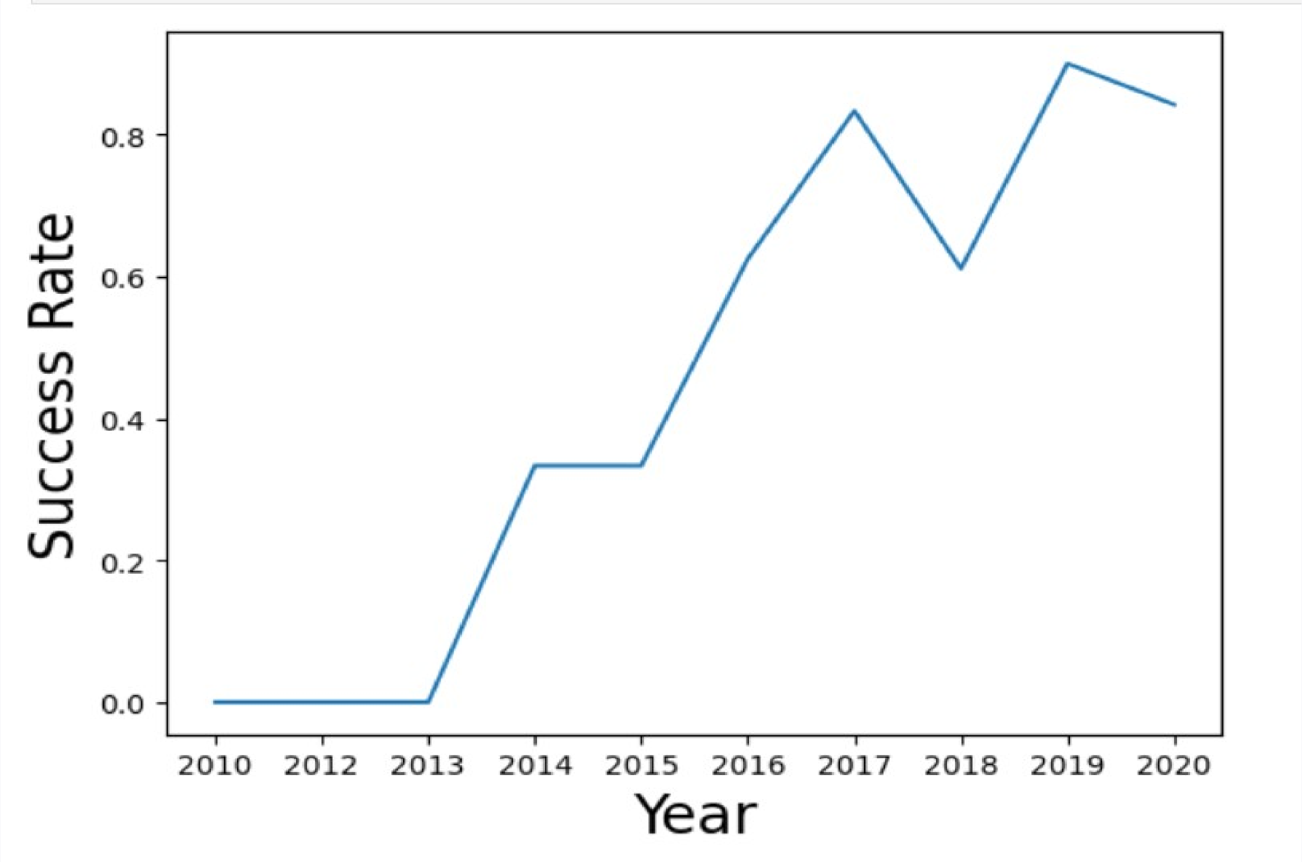
- scatter point chart is shown in the below screenshot to visualize the relationship between payload and orbit type
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- a line chart is represented here to visualize the yearly average success rate

- It can be seen clearly that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

Below screenshots has been captured from the EDA with SQL lab work and displays the names of all unique launch sites. There are four unique launch sites:

Display the names of the unique launch sites in the space mission

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

```
 * 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'

Below screenshot displays the sql query to find the 5 records where launch sites begin with `CCA` along with the result.

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
```

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

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

Below query result displays the total payload mass(in Kg) carried by boosters from NASA.

Here in the sql query, a sum function has been used to add the total payload for a specific customer named ‘NASA (CRS) :

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

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTABLE where Customer='NASA (CRS)'
```

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

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

As shown in the EDA with SQL lab result screenshot below, average payload mass carried by booster version F9 v1.1 is 2534.67 kg approx.

Display average payload mass carried by booster version F9 v1.1

```
]: %sql select avg(PAYLOAD_MASS__KG_) as Avg_Payload_Mass from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
```

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

]:

Avg_Payload_Mass
2534.6666666666665

First Successful Ground Landing Date

First successful landing outcome on ground pad was performed on 22 Dec 2015. Query result from EDA lab has been displayed in below screenshot:

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql select min(date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

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

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Below screenshot depicts the list of the boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 :

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT booster_version,PAYLOAD_MASS__KG_, Landing_Outcome from SPACEXTBL where  "Landing_Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ >4000

and PAYLOAD_MASS__KG_ < 6000
```

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

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

Below screenshot displays the sql query which results the total number of successful and failure mission outcomes.
It can be observed that there are total 100 successful mission outcomes with one failure as per the data given :

List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome,Count(*) as Total from SPACEXTBL group by Mission_Outcome
```

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

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Below screenshot represents the names of the booster which have carried the maximum payload mass. Here , it can be clearly seen that there are 12 such boosters which carries the max payload:

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ in (Select max(PAYLOAD_MASS__KG_ ) from SPACEXTBL)
```

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

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

2015 Launch Records

Below screenshot displays the sql query and its results with list of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015. There are two such failures observed in 2015 and both occurred on same site:

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql select substr(Date,6,2)as Month , Landing_Outcome as Failure_Landing_Outcome ,Booster_Version , Launch_Site from SPACEXTBL
where Landing_Outcome='Failure (drone ship)' and substr(Date,0,5)='2015'
```

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

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

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

Below screenshot displays the sql query result for the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order:

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
%sql SELECT "LANDING_OUTCOME", COUNT(*) as 'COUNT' FROM SPACEXTBL WHERE substr(Date,1,4) || substr(Date,6,2) || substr(Date,9,2)
between '20100604' and '20170320' GROUP BY "Landing_Outcome" ORDER BY "COUNT" DESC;
```

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

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

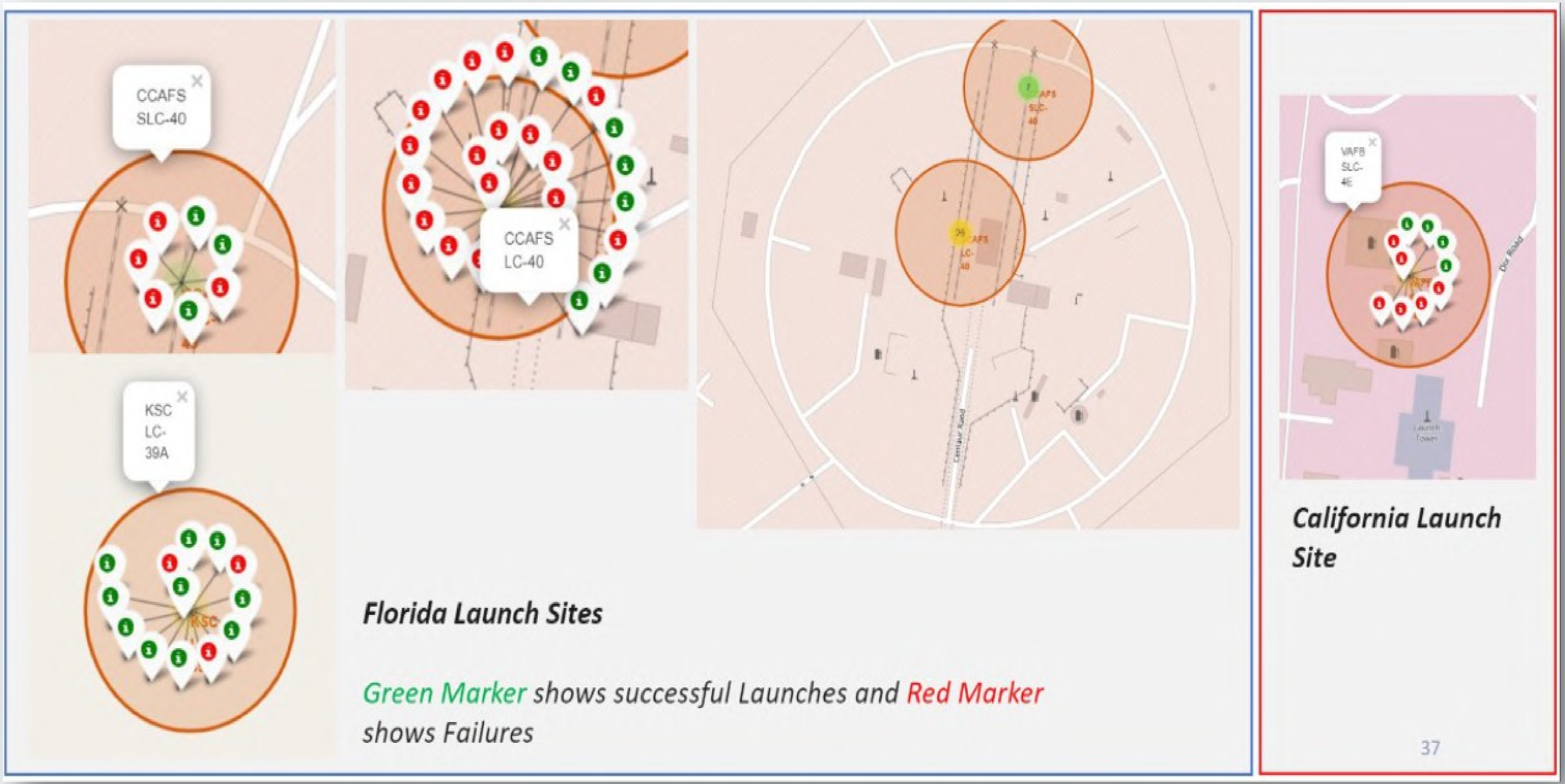
Section 3

Launch Sites Proximities Analysis

Launch Site Location

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

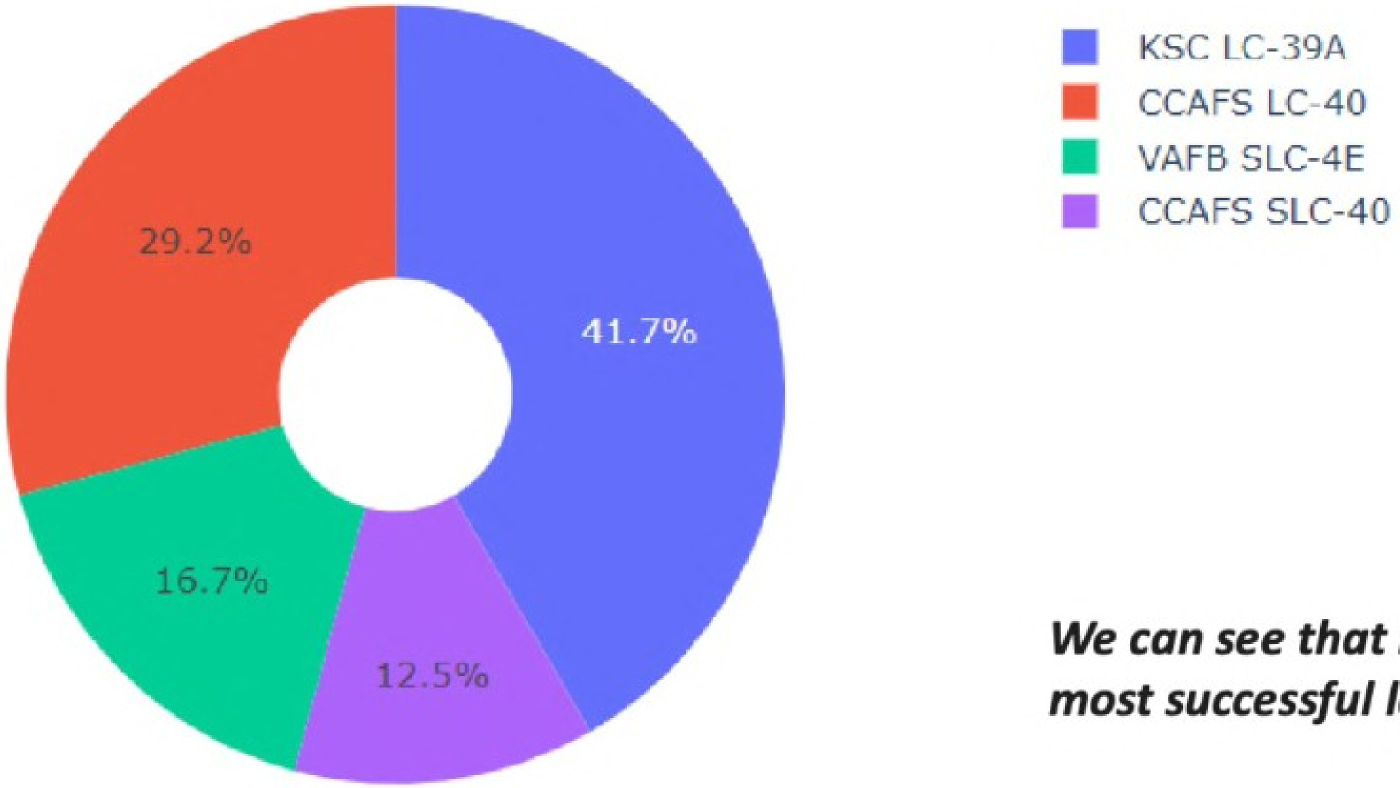
Marker showing launch site with color level



Section 4

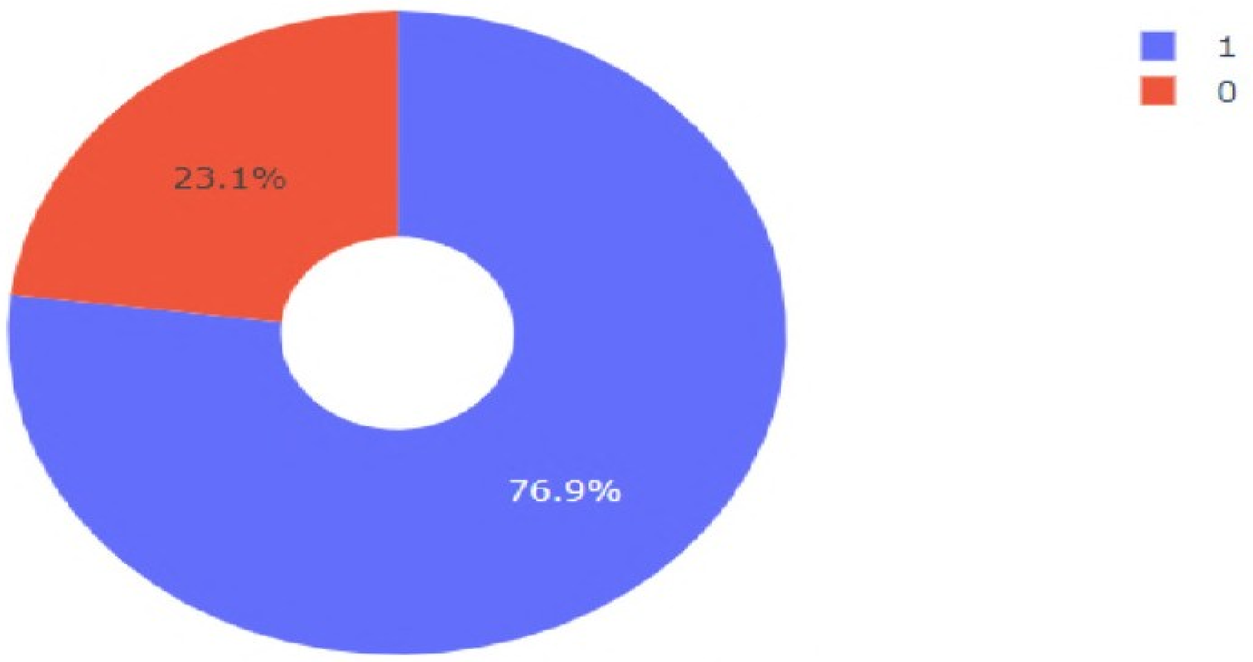
Build a Dashboard with Plotly Dash

Success percentage by each site



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

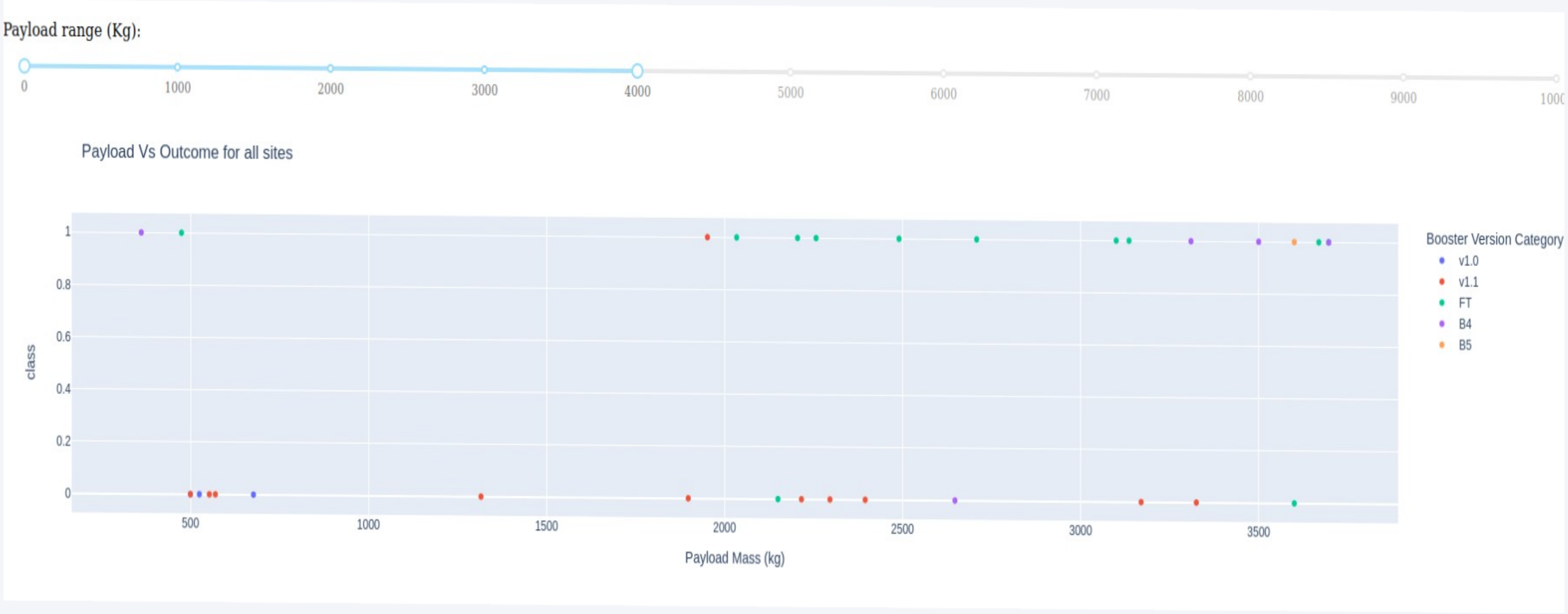
Highest launch success ratio: KSC LC-39A



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

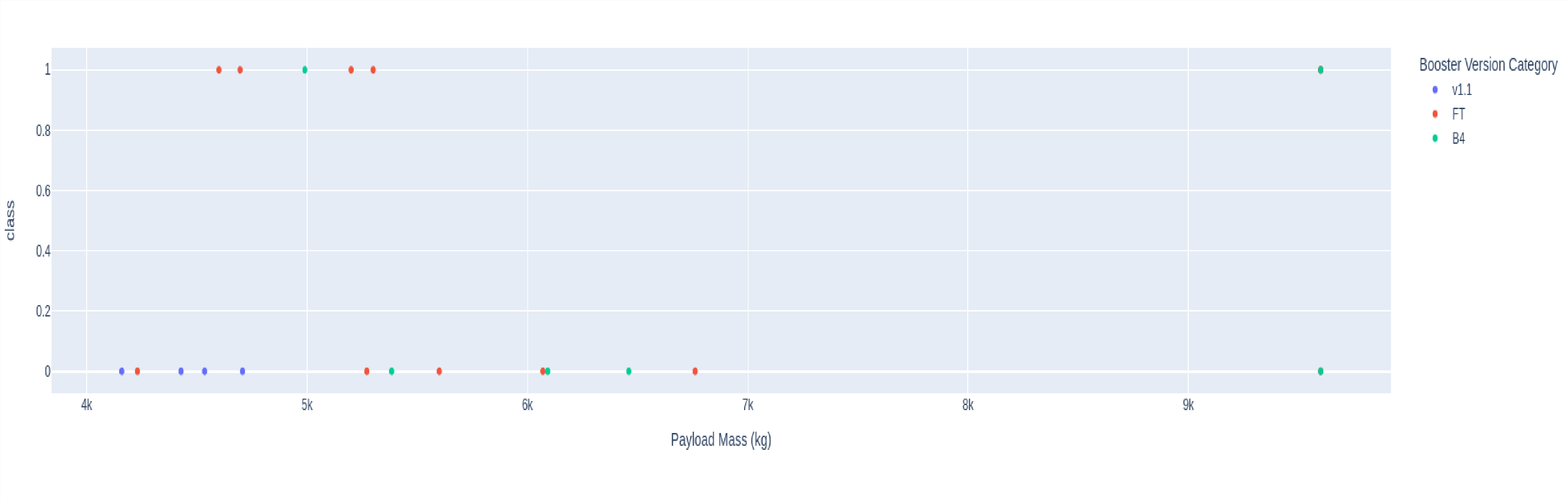
Payload Vs Launch Outcome Scatter Plot(for low weighted payload<4000)

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



Payload Vs Launch Outcome Scatter Plot(for high weighted payload>4000)

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

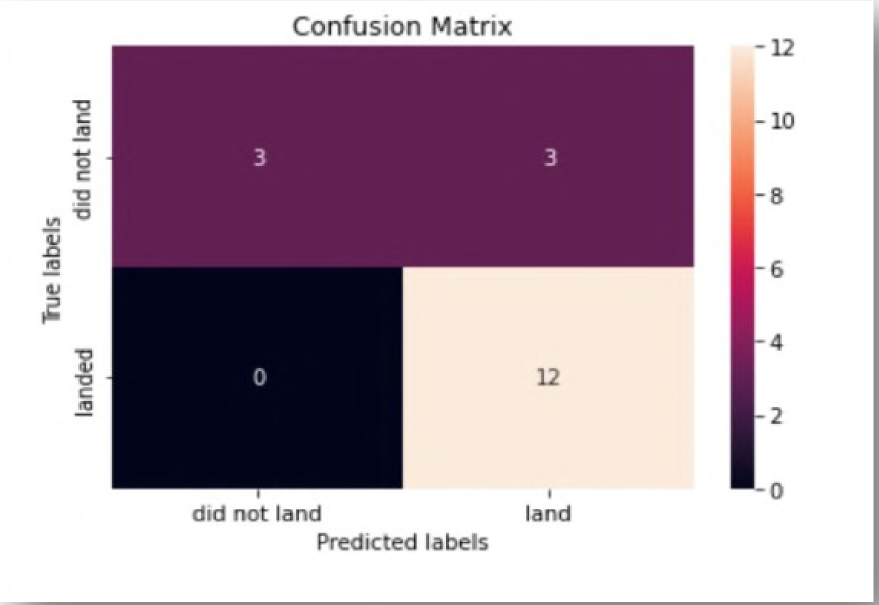
As we can see, by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy.

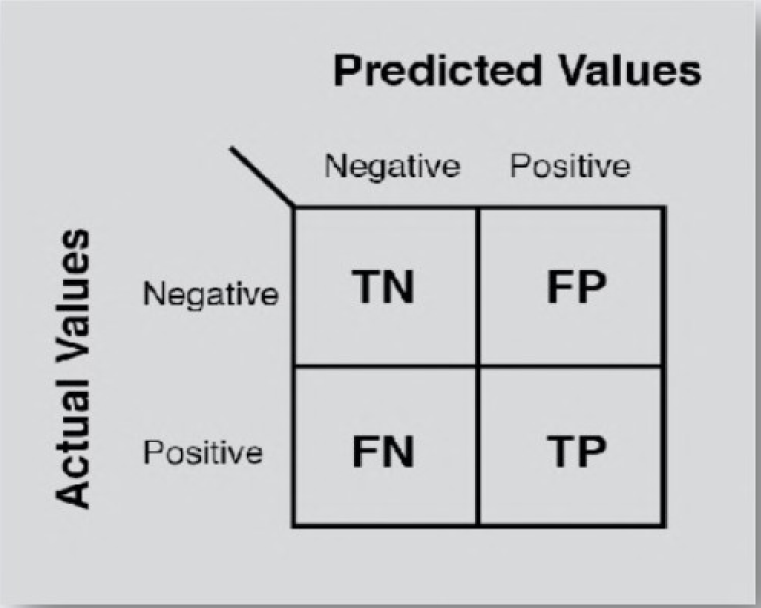
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params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

Below can be concluded :

- 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!