IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abhijit Rath
11-03-2022

Outline

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

Executive Summary

- From the dataset after the Vizualisation it is found that launch sites are located closer to Proximity of coastlines and Equators as well as rail road .
- Launch Site in the west of the MAP named Kennedy Space Center has higher success rate than other launch sites.
- After training Classifiers, Decision tree Classifier is found to have most accuracy over others.
- In Testing Data, All Classifiers has the same accuracy at 83.333%

Introduction

- From the given Dataset, We are trying to predict whether the next falcon9 rocket by spaceX will probably land successfully or not.
- We are going to find what variables like the Payload Mass or Launch Site or any variables be a good predictor .
- We collect the data, do the Exploratory Data Analysis then Visualize the data.
- We then train the Machine Learning Algorithm using the data and find which is the best one out of all.

Section 1

Methodology

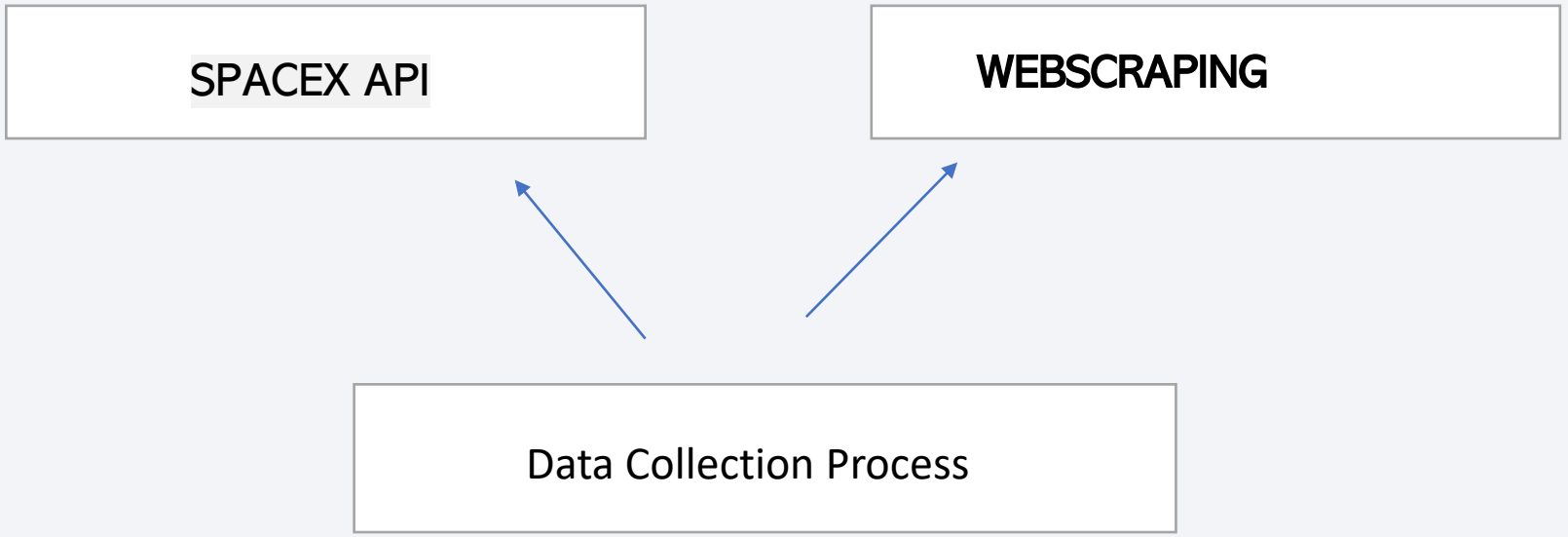
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from spacex API from the website and through Webscraping from Wikipedia and features were selected which are required,
- Data wrangling
  - Data was cleaned and pre processed the payload-mass mean was calculated to mill the None values
- Perform exploratory data analysis (EDA) using visualization and SQL
  - EDA was done using magic SQL in python to find out the meaningful insights
- Visualization was then carried out via plotly and dash using pie charts and scatter plot
- The Data was then trained and tested to find the accuracy and to find best classifier

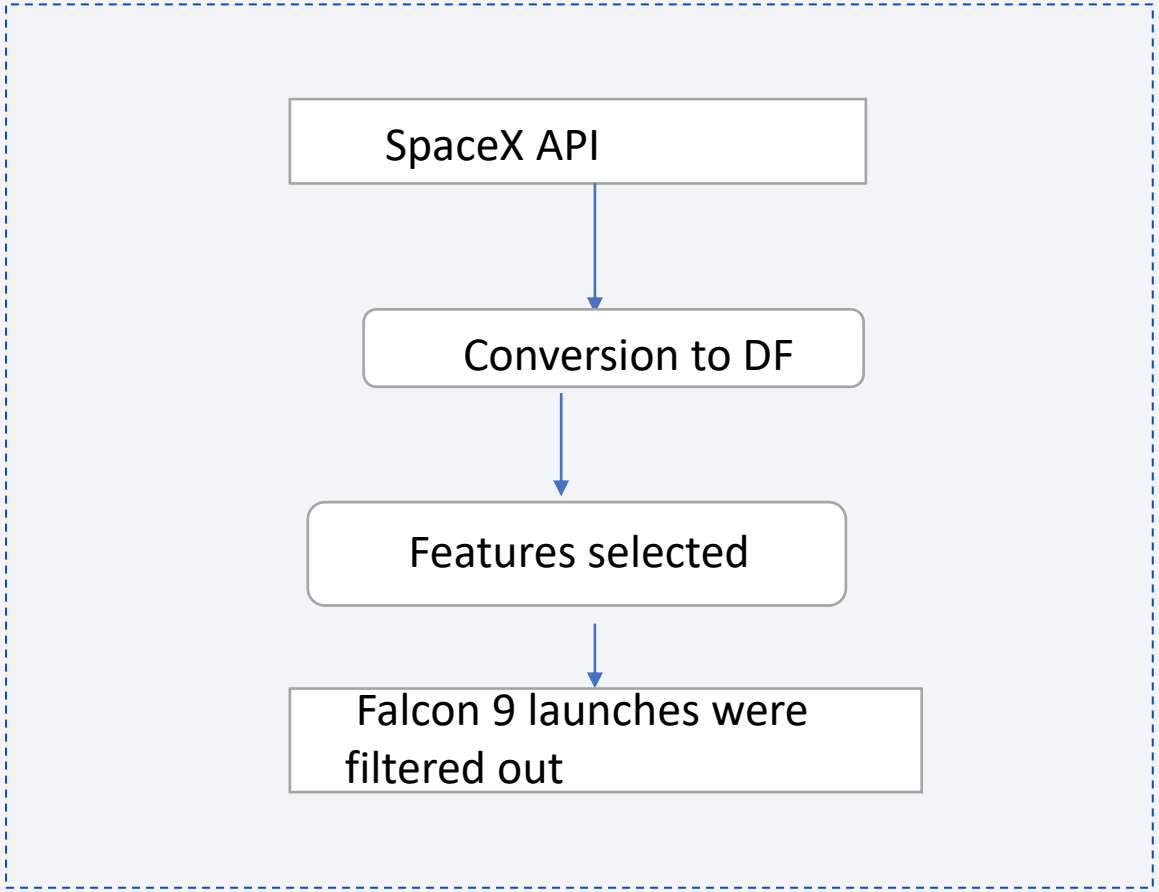
Data Collection

- Datasets were collected from SPACEXAPI via SpaceX website and Web scraping from Wikipedia.



Data Collection – SpaceX API

- The Data was collected using the api url https://api.spacexdata.com/v4/launches/past
- The response content was decoded in json format and normalized using pd.json_normalize function to convert to a Dataframe
- Required Features were extracted like Booster Version, Payload Mass etc. then Falcon 9 launches were filtered out.
- Github link to notebook: https://cutt.ly/uABJFh4



Data Collection – SpaceX API

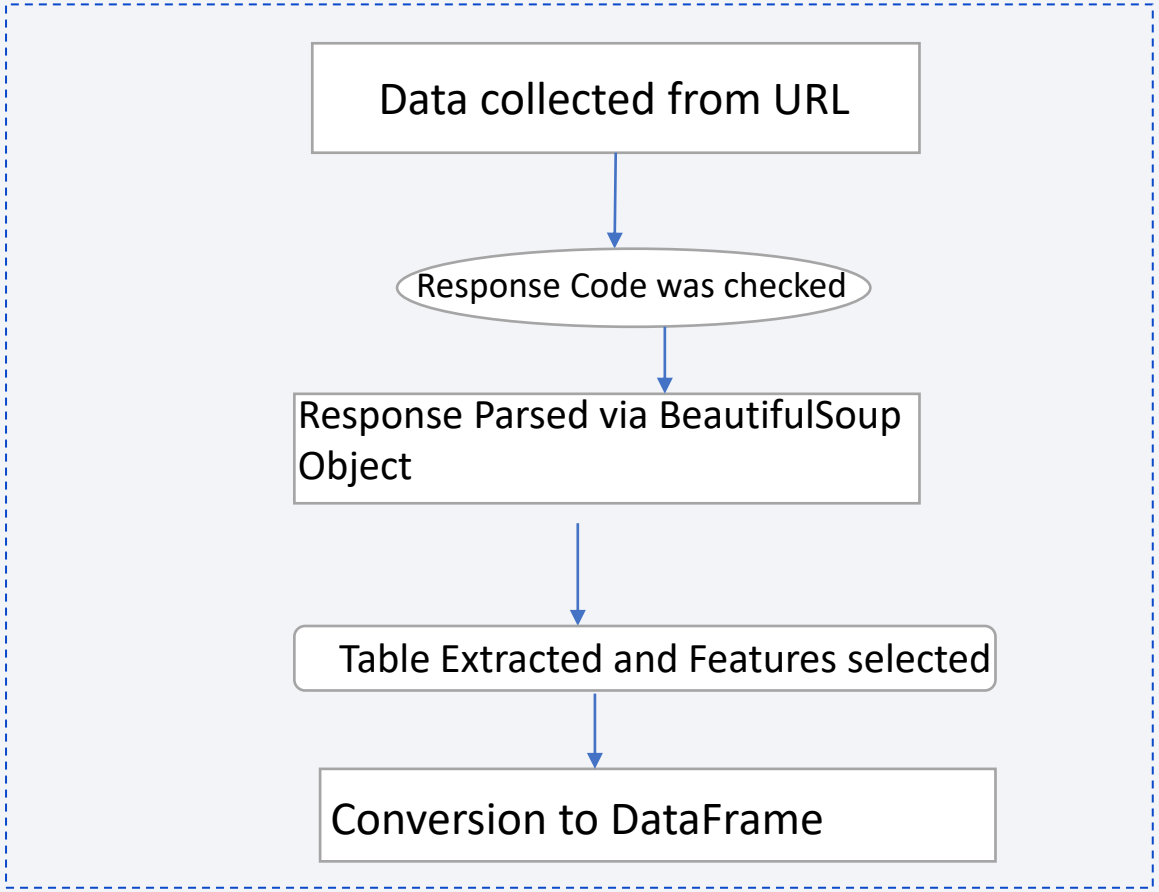
- Missing Values of Payload Mass was found out and the mean of the Payload Mass was Calculated and filled accordingly
- Github link to notebook: https://cutt.ly/uABJFh4

```
[59]: data_falcon9.isnull().sum()

Out[59]:
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       5
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

```
In [60]:
# Calculate the mean value of PayloadMass column
avgpayloadmass=data_falcon9["PayloadMass"].astype(float).mean(axis=0)

data_falcon9["PayloadMass"].replace(np.nan,avgpayloadmass,inplace=True)

# Replace the np.nan values with its mean value

/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/pandas/core/generic.py:6619:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
  return self._update_inplace(result)

In [61]:
data_falcon9.isnull().sum()

Out[61]:
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

Data Collection - Scraping

- Web scraping was done using the Wikipedia link from the table of Falcon 9 launches of spacex.
- Using BeautifulSoup Object, response was parsed , required table is extracted and features were selected and converted into a dataframe from dictionary.
- Github Link: https://cutt.ly/3ABCbyl



Data Wrangling

- After dataset is loaded and missing values is found , the categorical variables were then observed.
- The number of Launch sites were counted using value_counts() method

```
n [6]: # Apply value_counts() on column LaunchSite
       df['LaunchSite'].value_counts()

ut[6]: CCAFS SLC 40    55
       KSC LC 39A      22
       VAFB SLC 4E     13
       Name: LaunchSite, dtype: int64
```

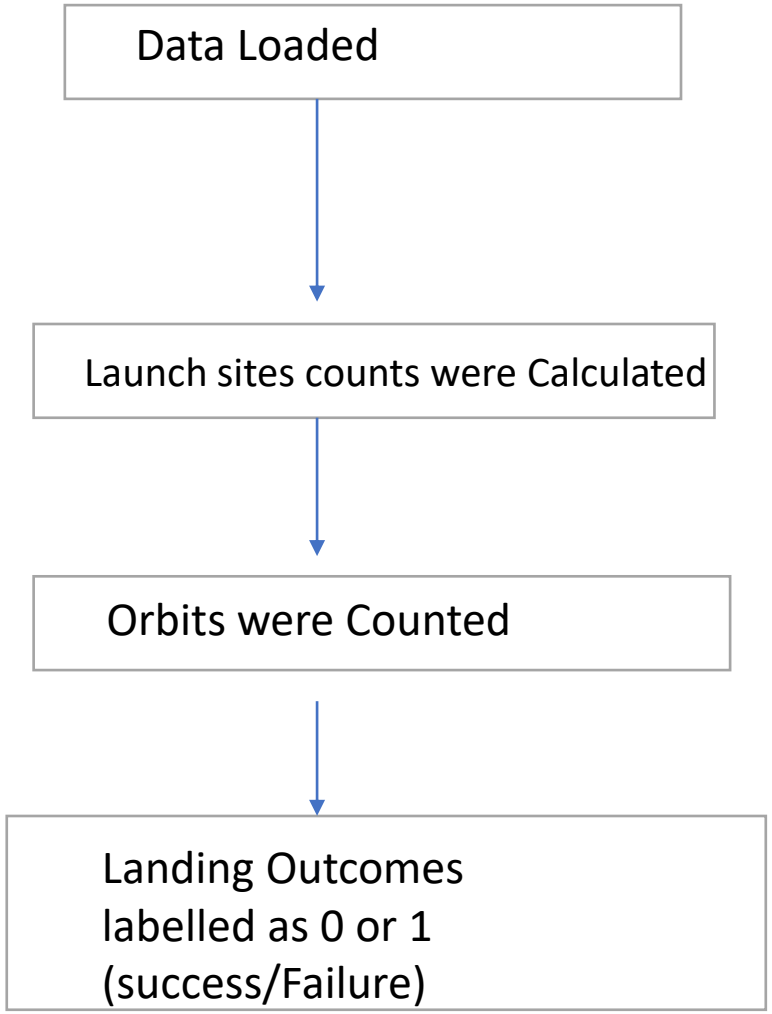
Data Wrangling

- The number of occurrence of each orbit is then calculated with value_counts() method.

```
In [7]: # Apply value_counts on Orbit column
        df['Orbit'].value_counts()

Out[7]: GTO      27
        ISS      21
        VLEO     14
        PO        9
        LEO       7
        SSO       5
        MEO       3
        ES-L1     1
        HEO       1
        SO        1
        GEO       1
        Name: Orbit, dtype: int64
```

- Landing Outcome is then found and then labeled as Success and Failure. With Class 0 for Failure and Class 1 for a success outcome
- *Github Notebook: https://cutt.ly/cAB1Mqy*

Data Wrangling



EDA with Data Visualization

- Scatter plot was used to find the relationships of variable like Flight Numbers with Lauch sites, Orbit types and Payload Mass with Launch Site and Orbit types
- Bar Plot is used to find the success rate of each orbit type
- Finally, We used Line Plot to find out the Success trends over time.

- *Github Notebook : https://cutt.ly/FAB2Q6M*

EDA with SQL

- First, we selected the distinct lauch sites using the query

select distinct(launch_site) from spacex;

- We have then calculated the total Payload mass using the SUM()

Function and Average Payload Mass using AVG() function

- We then checked the first landing also examined the outcomes of landing

- *Github Ref notebook: https://cutt.ly/1AB9o0d*

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, success) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()
- We the calculated the distance between rail roads, Coastline , Highway and nearest city added the lines with the help of polyline method to measure the distance.

Build an Interactive Map with Folium

- Observed that : It is at proximity to coastline and Rail Roads and Highways but Cities are farther from the launch site because of the safety reasons.

- *Github Notebook link: https://cutt.ly/pA8Gjm0*

Building a Dashboard with Plotly Dash

- The dashboard is built with Flask and Dash web framework provided by IBM Cognitive Class Framework
- **Graphs:**
  - Pie Chart showing the total launches by a certain site/all sites
  - display relative proportions of multiple classes of data.
- **Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions:** It shows the relationship between two variables. i.e, How variables are correlated to each other.

- *Github Link for Source Code: https://cutt.ly/sA8GtX9*

Predictive Analysis (Classification)

- It was done as per the step below:

- **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.

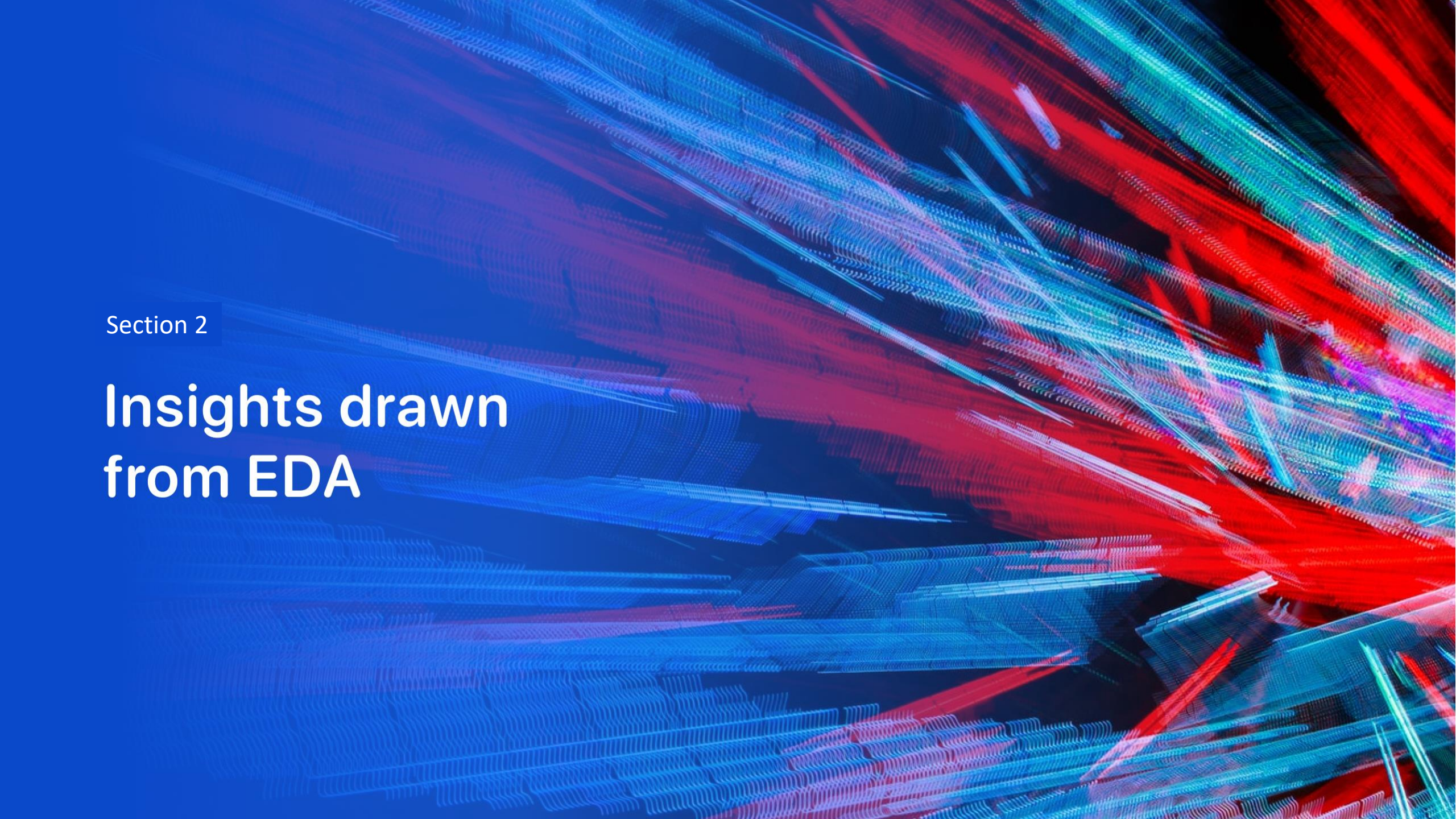
Predictive Analysis (Classification)

- **EVALUATING MODEL :**
  - Check accuracy for each model
  - Get tuned hyperparameters for each type of algorithms
  - Plot Confusion Matrix IMPROVING MODEL
  - Feature Engineering
  - Algorithm Tuning
- **FINDING THE BEST PERFORMING CLASSIFICATION MODEL**
  - The model with the best accuracy score wins the best performing model
  - In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.
- *Github Notbook Link: https://cutt.ly/fA8K7L4*

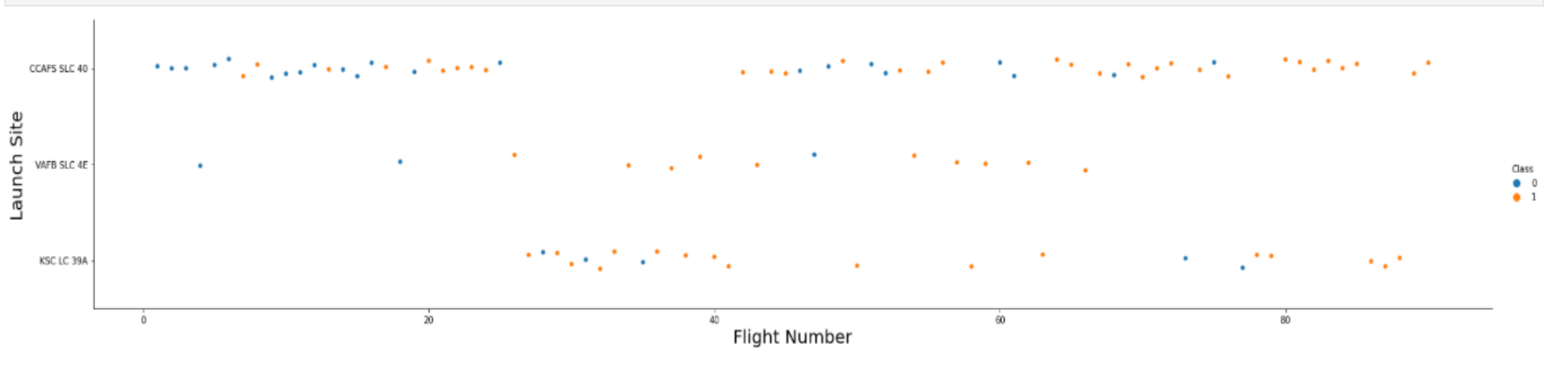
Results

In the Upcoming Slides we shall see:

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

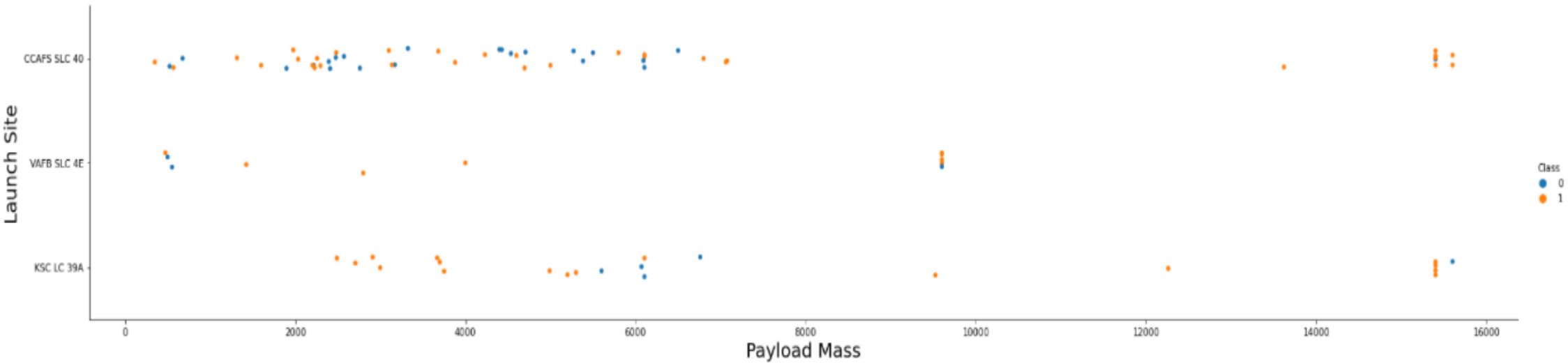

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



From the plot above we can see that as the Number of flights increases the Success rate also increases.

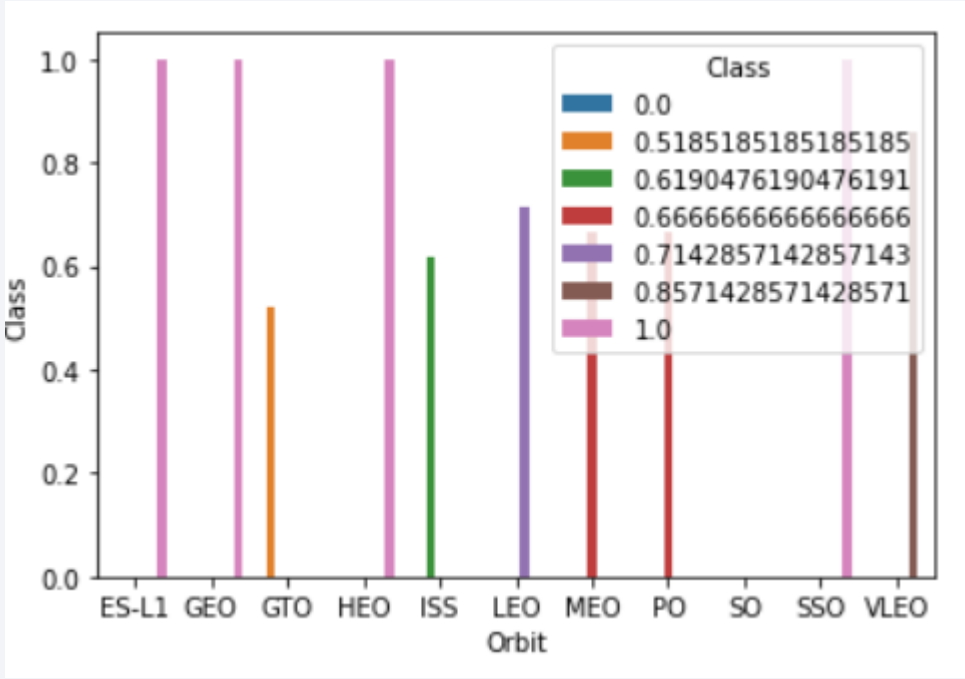
Payload vs. Launch Site



From Above Plot we can see that if the Payload Mass is greater than 10000 the success rate is much higher compared to failure. For VAFB SLC 4E launch site Payload Mass greater than 10000 is non-Existent.

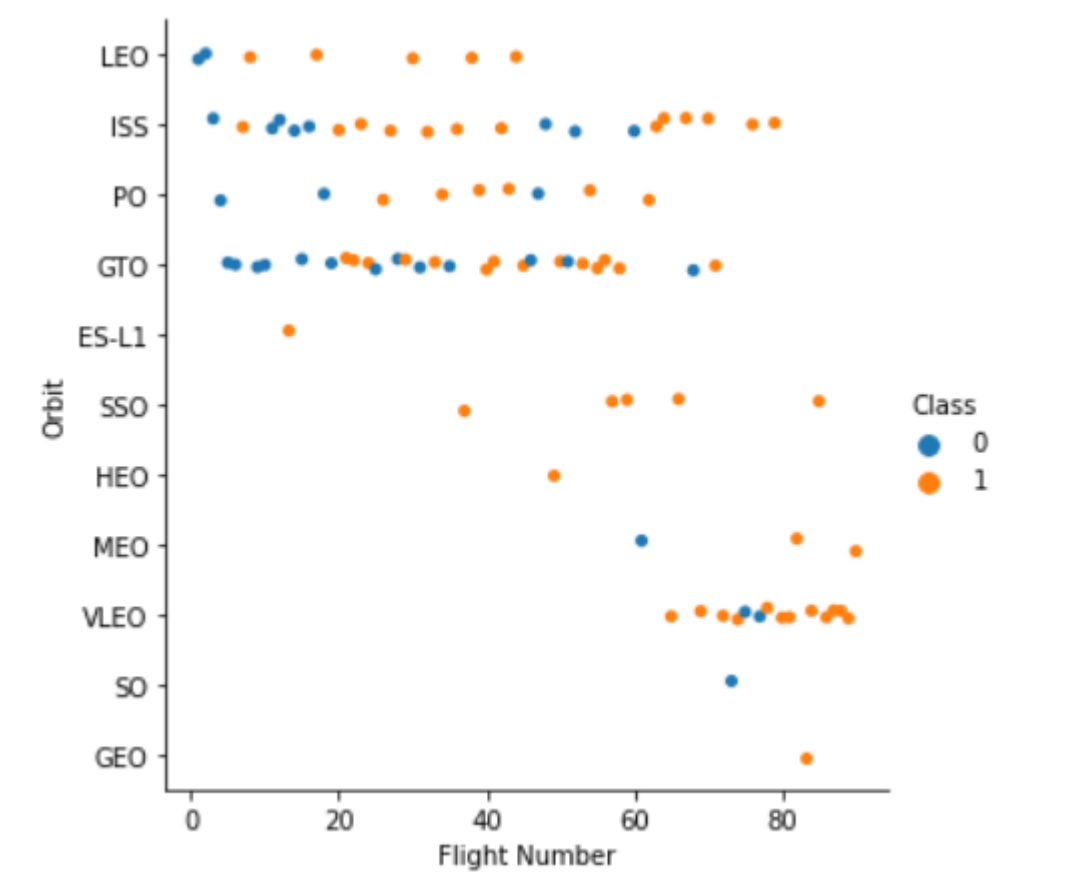
Success Rate vs. Orbit Type

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

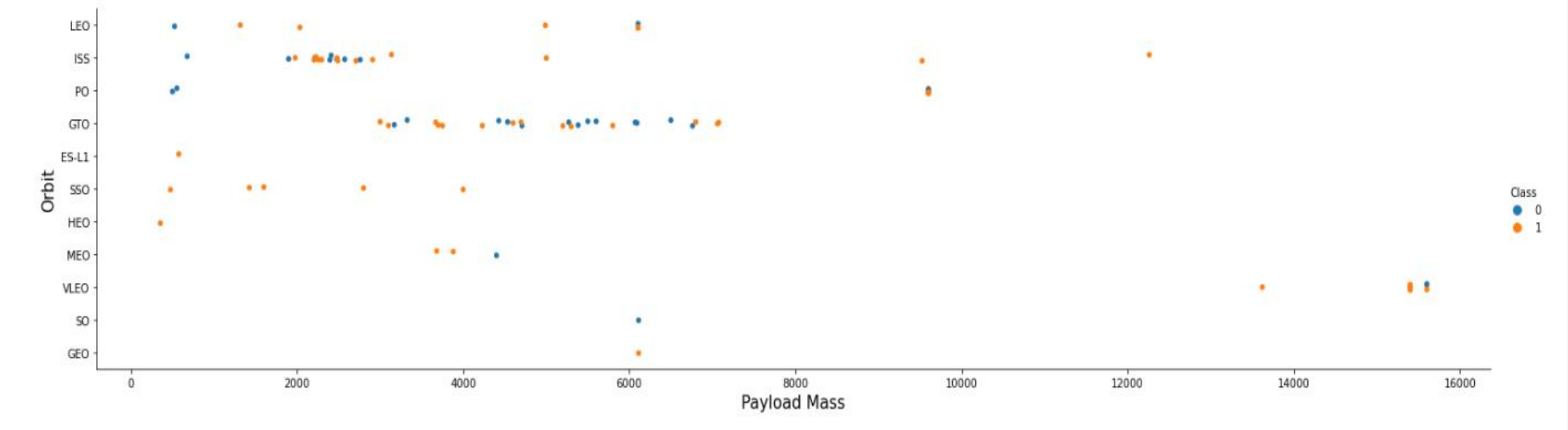


Flight Number vs. Orbit Type

- You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.
- 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.



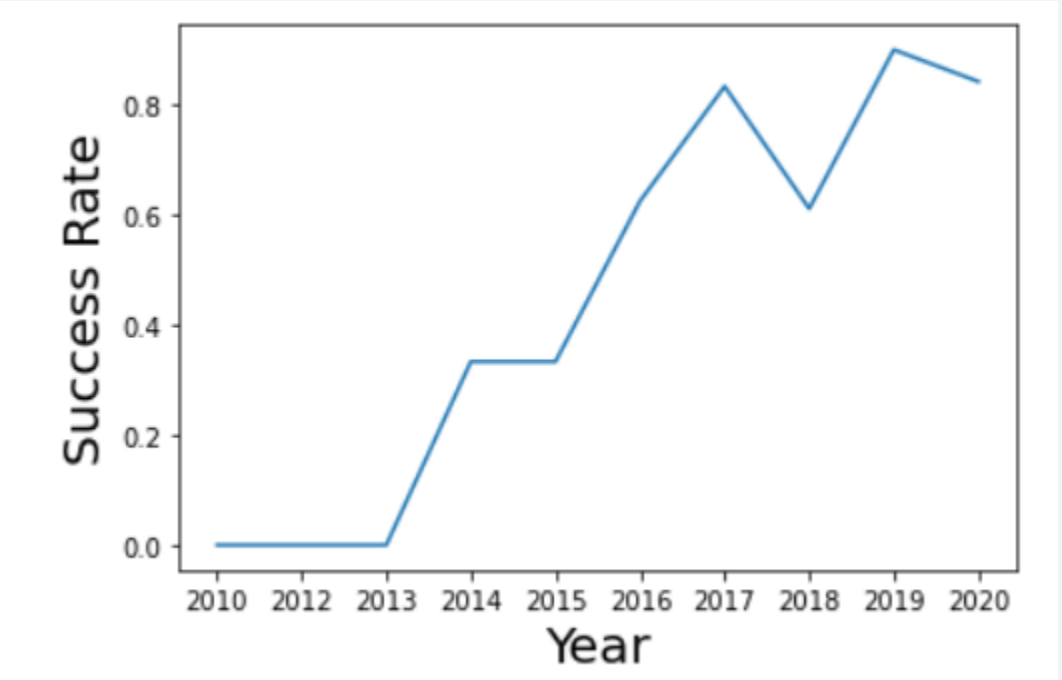
Payload vs. Orbit Type



- 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) both are there.

Launch Success Yearly Trend

- We can observe that the Success Rate increases as the Year Increases which signifies positive success rate with time.



All Launch Site Names

In [3]:
```
%sql select distinct(launch_site) from spacex;
```

```
 * ibm_db_sa://rpt40241:***@76[redacted]
Done.
```

Out[3]:

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

Unique Launch Sites were extracted using the above query.

Launch Site Names Begin with 'CCA'

```
%sql select * from spacex where launch_site like 'CCA%' limit 5;
```

Out[4]:

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Using the above query we found out the launch sites name starting with CCA

Total Payload Mass

Using the query below we found the total Payload Mass.

```
]: %sql select SUM(payload_mass__kg_) as Total_payload_mass from spacex where customer='NASA (CRS)';
```

Out[5]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

```
%sql select AVG(payload_mass__kg_) as average_payload_mass from spacex where booster_version like 'F9 v1.1%'
```

average_payload_mass
2534

Using the above query we found the average Payload Mass by F9 V1.1 booster versions.

First Successful Ground Landing Date

```
%sql select MIN(Date) as first_landing from spacex where mission_outcome='Success';
```

Done.

first_landing
2010-06-04

Using the above query we found out the first landing date using MIN() function.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
9]: %sql select booster_version,payload_mass__kg_ from spacex where landing__outcome='Success (drone ship)' and payload_mass__kg_ in (select payload_mass__
```

ut[19]:

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Using the query and sub-query we found the list of successful Drone ship landing between Payload Mass 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome as outcome,count(*) as count from spacex group by mission_outcome;
```

Out[26]:

outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Using the above query we grouped the mission_outcome as shown above

Boosters Carried Maximum Payload

```
%sql select booster_version from spacex where payload_mass__kg_=(select max(payload_mass__kg_) from spacex);
```

Out[27]:

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

Using the above query we found the Booster versions carrying maximum payload using a subquery.

2015 Launch Records

```
%sql select landing__outcome,booster_version,launch_site from spacex where landing__outcome='Failure (drone ship)' and YEAR(date)=2015;
```

Done.

Out[28]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Using the query above we found the list of failure launches in the year 2015

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

```
%sql select count(*) as count,landing__outcome from spacex where date between '2010-06-04' and  '2017-03-20' group by landing__outcome order by count(*
```

Done.

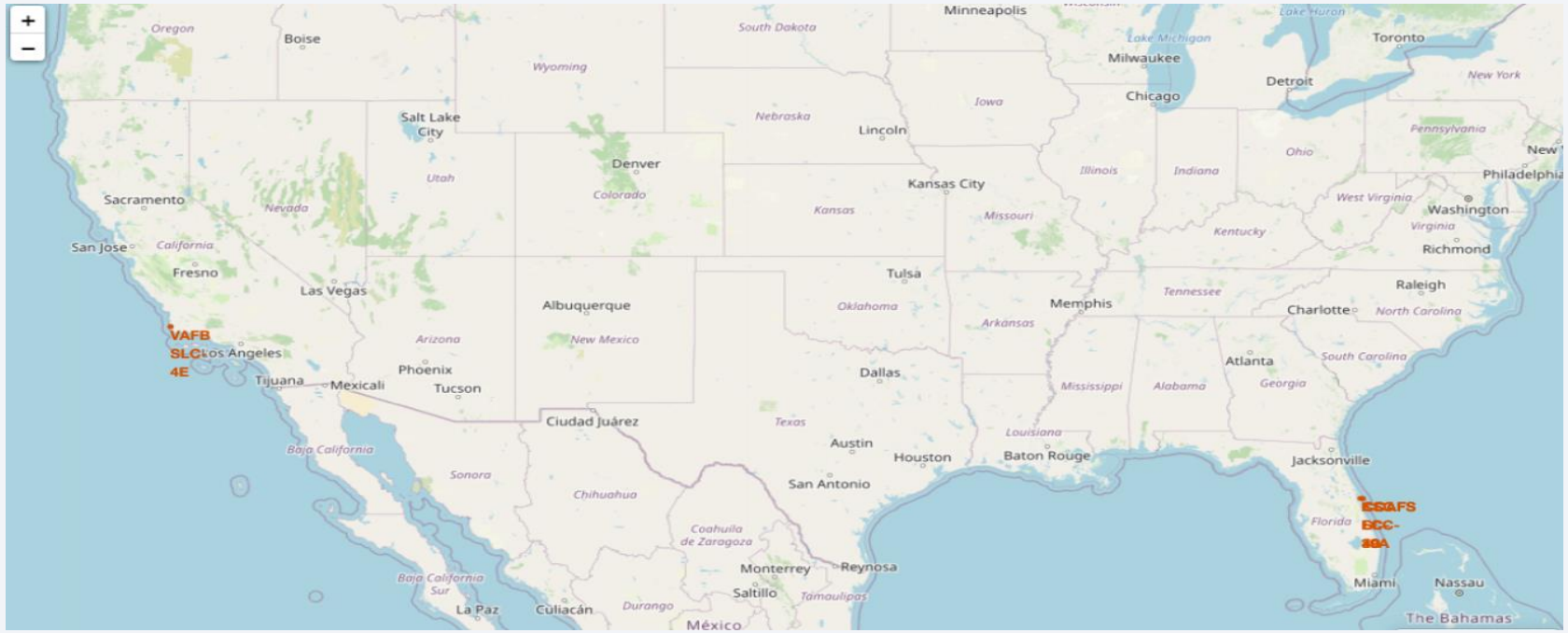
[33]:

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

Using the above query we found out the landing outcomes in descending order between the dates 2010-06-04 and 2017-03-20

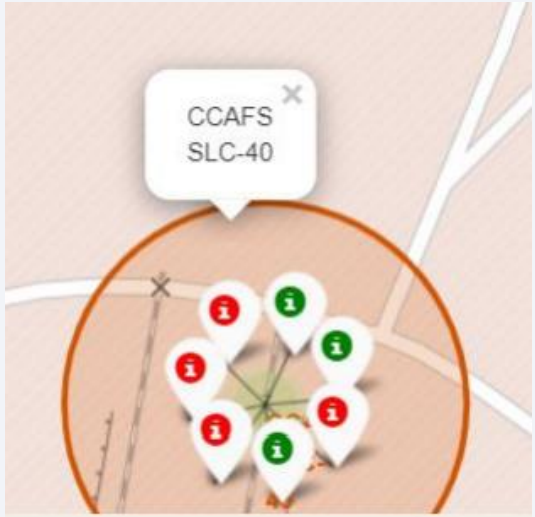

Section 3
Launch Sites
Proximities Analysis

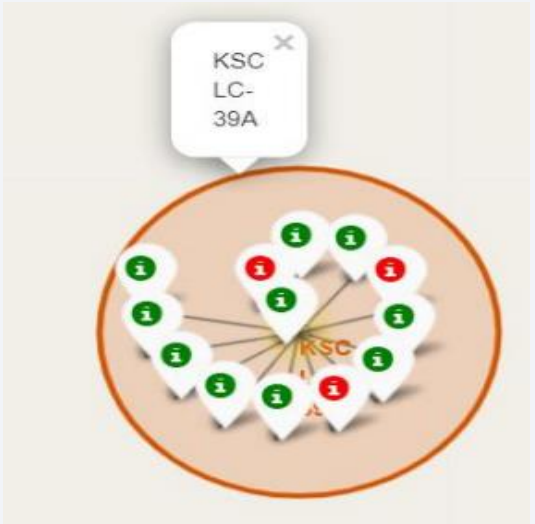
Launch Sites in the USA

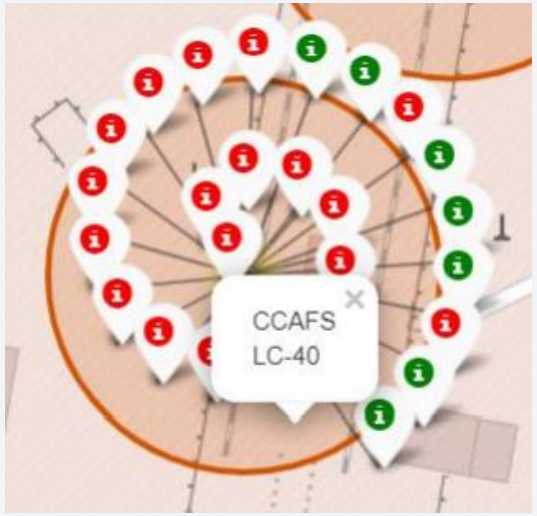


We can observe that the launch sites are near to the coastlines of USA which is near Florida and California

Checking Success Rates of Launches using Markers

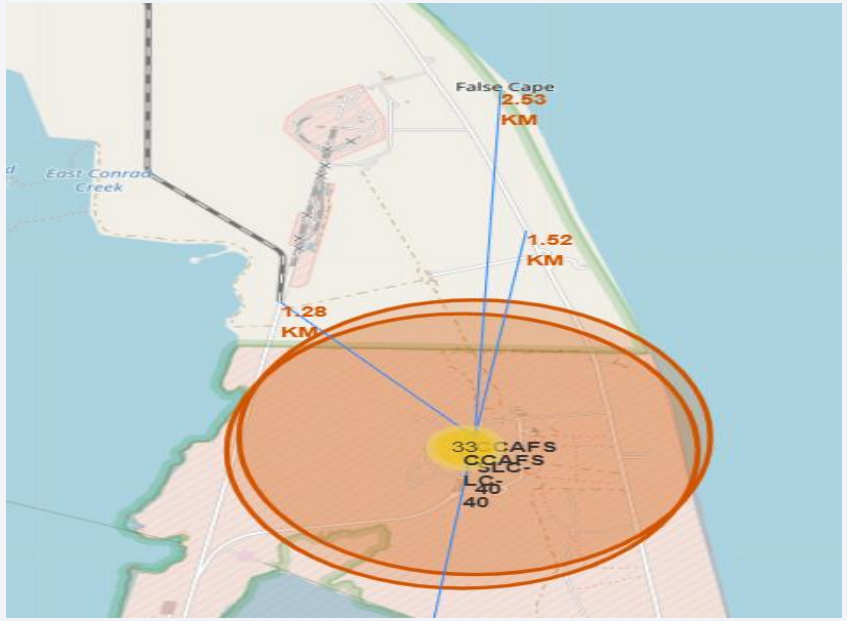


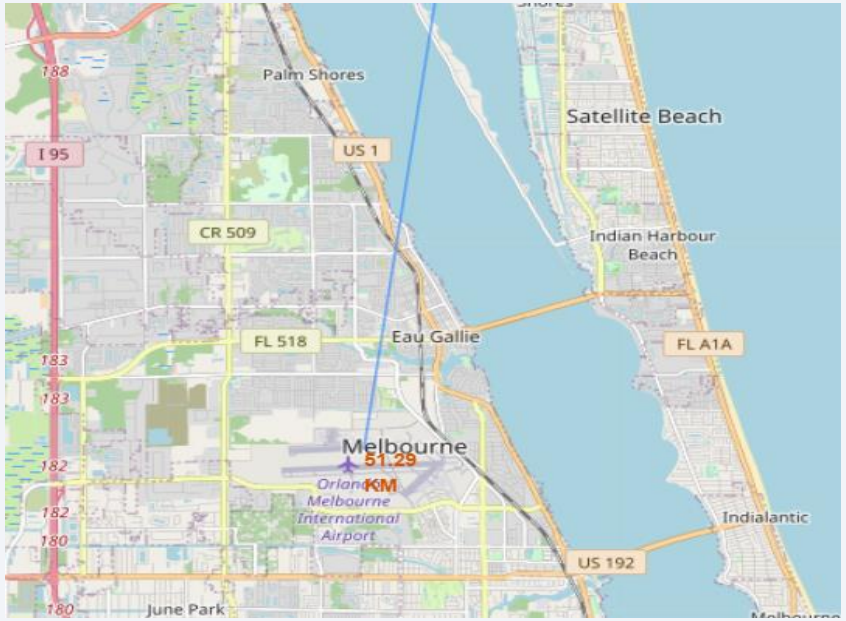




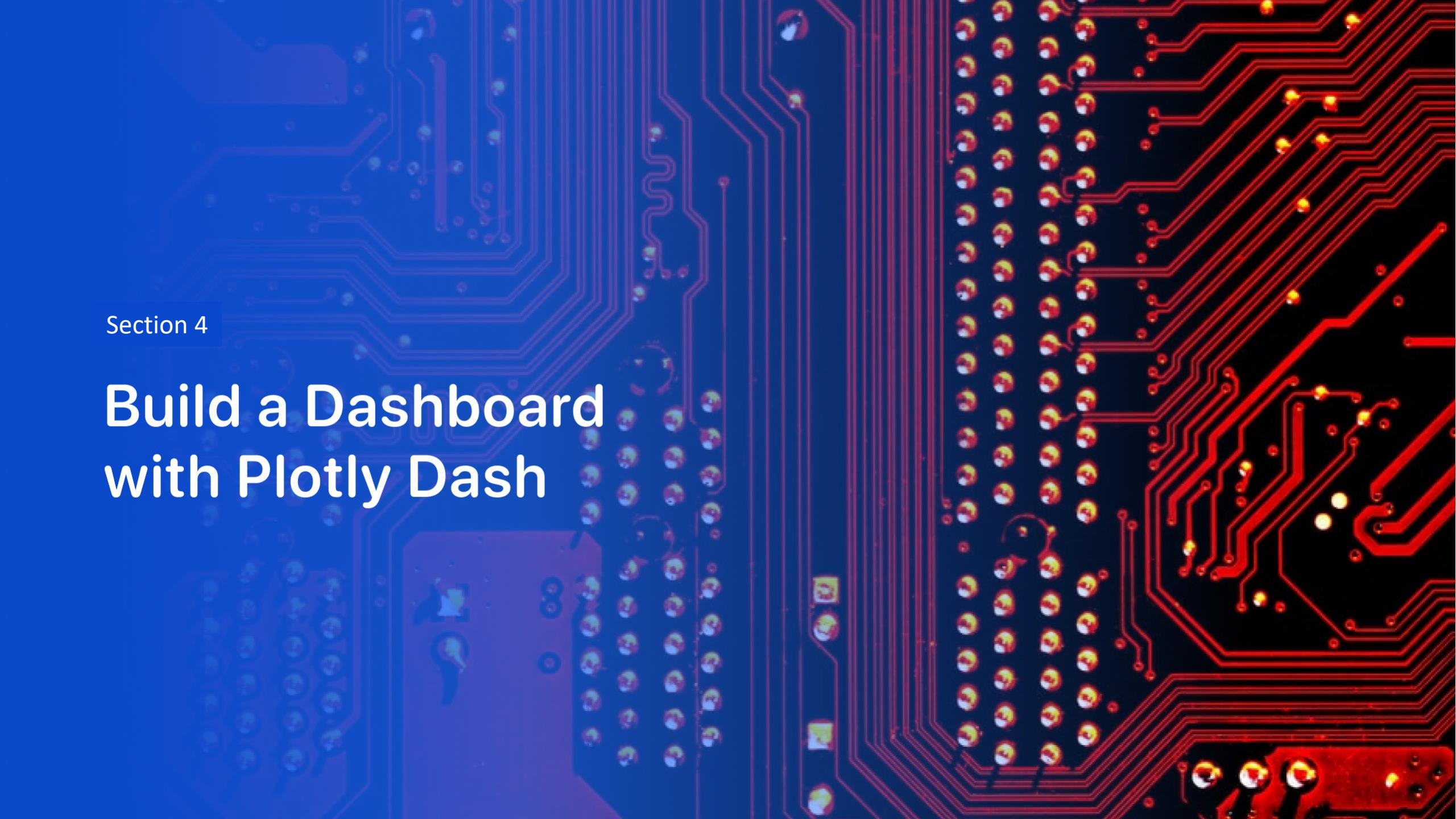
Using Folium we draw the cluster and add it to MarkerCluster() method. We observe that CCAFS-LC 40 has the highest launches but more failure rates, KSC LC 39-A has most number of successful launches. Green Marker shows Successful launches and Red Markers shows Failures

Checking Close Proximities with Polylines

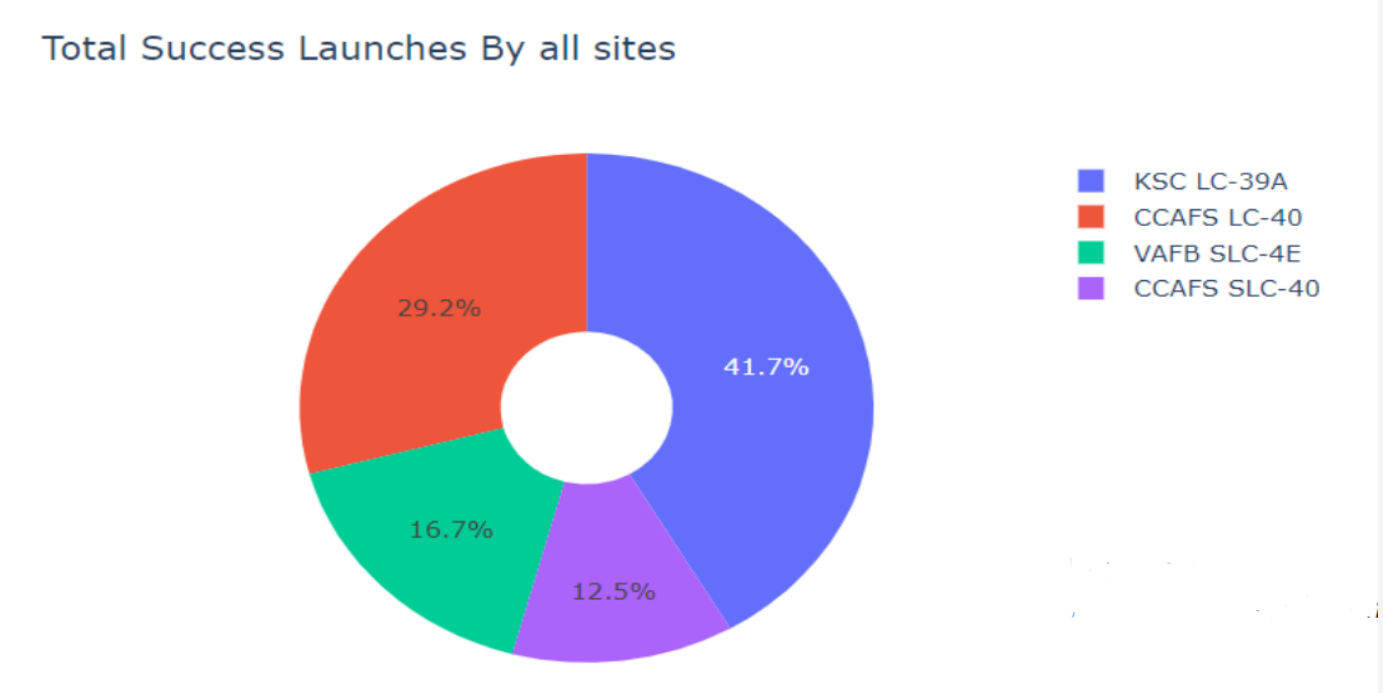




- From the above figures we can see that Coastline, Highways and Rail Roads are in close proximity because of the amount of goods needed to be transferred to launch sites and loads movement also it is safer to launch rockets near coast lines in an event of failed missions it can crash in the ocean.
- Cities are farther from the Launch site because of the safety reasons.

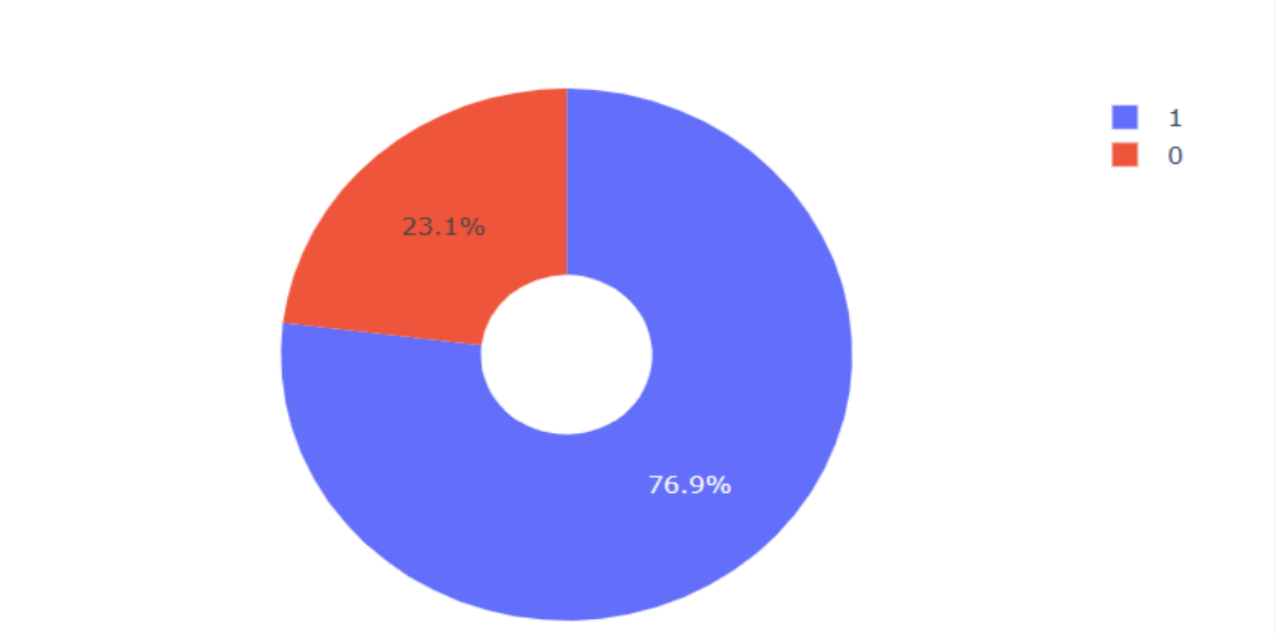

Section 4
Build a Dashboard with Plotly Dash

Success Rate- Pie Chart



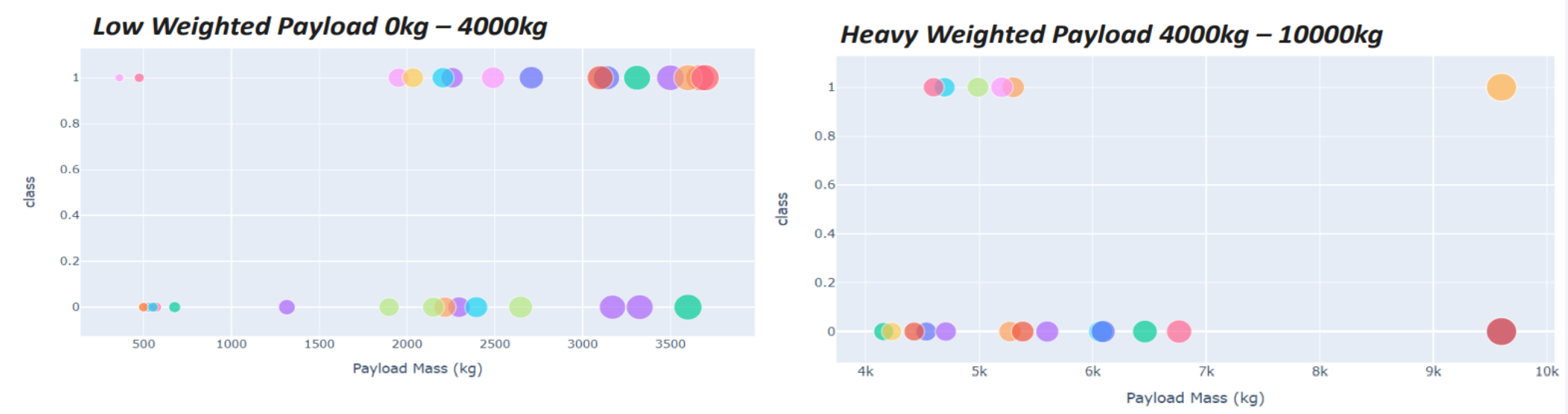
We can see that KSC-LC-39A has comparatively high Success rate than other sites.

Pie Chart With Highest Success rate Launch Site



KSC-LC 39 A has a success of 76.9% launches and failures of 23.1% launches

Payload Vs Launch Site comparison for different boosters



We can see that success of low weight payloads is high than Heavy Weight Payloads

Section 5

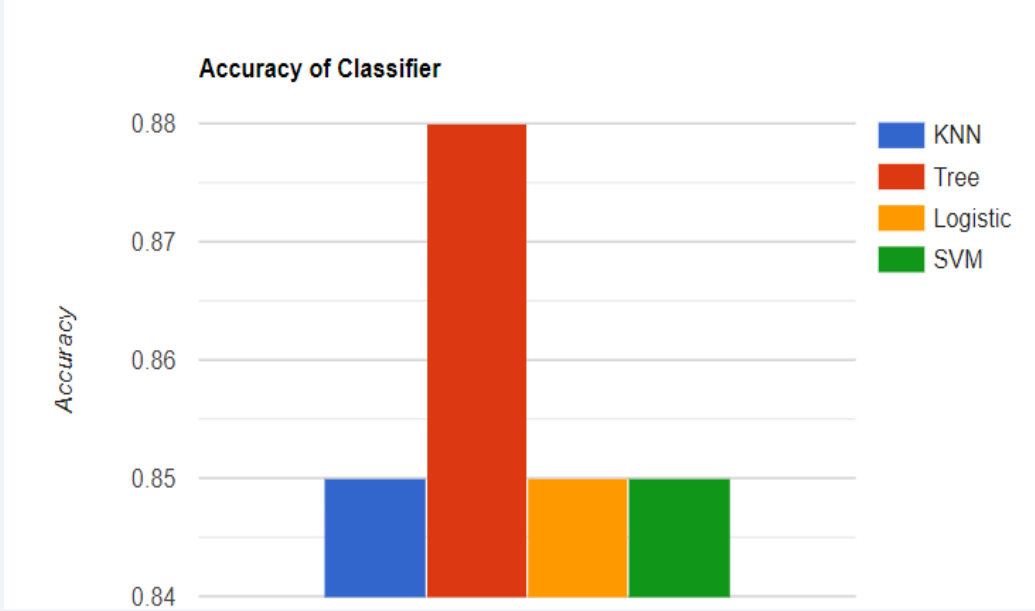
Predictive Analysis (Classification)

Classification Accuracy

- From the bar graph we see Tree Classifier has an accuracy of 88% when tuned with best parameters.
- Result below

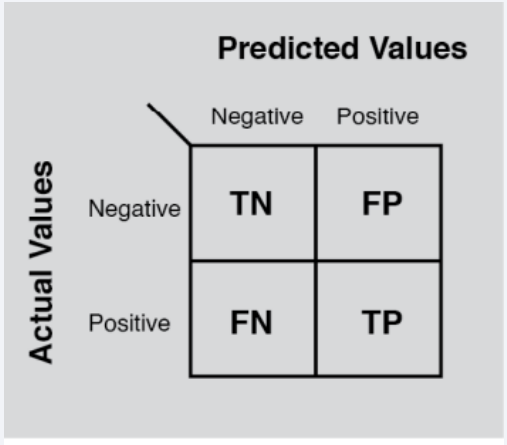
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 2, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.8857142857142858
```

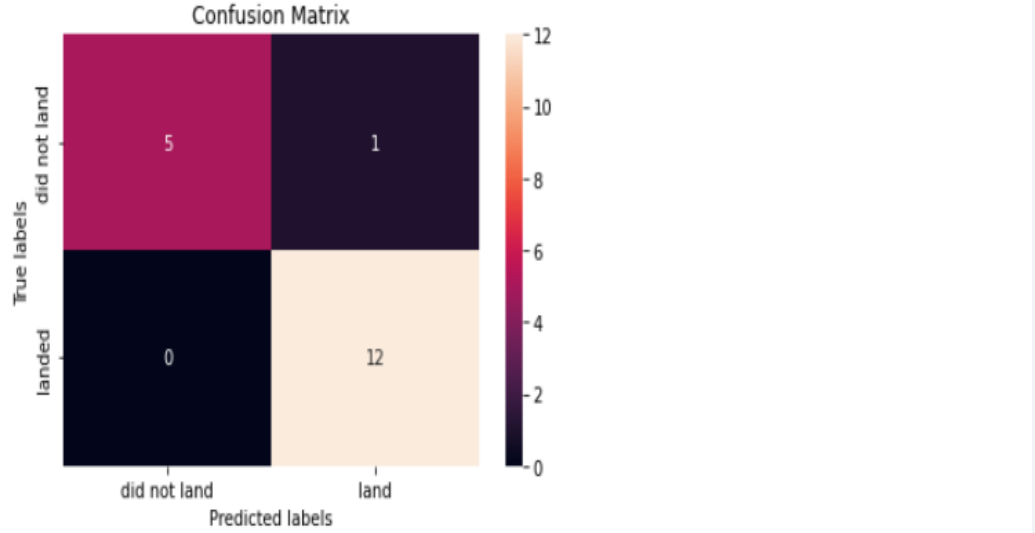


Confusion Matrix

From the Confusion Matrix we can see that the best performing model that is the decision tree classifier accurately predicts with only 1 False Positive.




```
In [26]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



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

- Confusion Matrix: https://cutt.ly/KA4ryTg
- Folium Map concepts: https://cutt.ly/HA4rpLG
- Marker Clusters: https://deparkes.co.uk/2016/06/24/folium-marker-clusters/

- Matplotlib Visualizations: https://cutt.ly/7A4rxzl
- Plotly is a module in python to make Dashboards

Thank you!