IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mozhgan Shahmohammadi
20 June 2023

Outline

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 company which advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars, while other providers cost upward of 165 million dollars each.

- Most of the savings is because SpaceX can reuse the first stage. Therefore if we can determine that the first stage will land, we can determine the cost of a launch.

- As a data scientist ,the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future.

Problems you want to find answers

- How to provide the data and clean and standardize the data?
- What features are the best and determine if the rocket will land successfully?
- Which method is the best which can predict the landing outcome ?
- How can we evaluate the selected method and find the best accuracy?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - ❖ using SpaceX API
  - ❖ web scrapping from Wikipedia
- Perform data wrangling
  - ❖ Using one-hot encoding for categorical features
- 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

Collecting the data using get request to the SpaceX Rest API

- Decoding the response content by using json() function call and turn it into a dataframe using a json.normalize()
- Cleaning the data with checking the missing values

Obtaining Falcon9 Launch data by web scraping Wikipedia pages

- Using python Beautifulsoup package to web scrape HTML tables
- Parsing the data from those tables and convert them into a Pandas dataframe for further analysis

Data Collection – SpaceX API

Using get request to the SpaceX API

Using a json.normalize() function to convert json response to dataframe

Cleaning and dealing with the missing values and data wrangling

Source Code:
https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

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

```
response = requests.get(spacex_url)
```

```
df=pd.json_normalize(response.json())
```

```
data_falcon9 = df[df.BoosterVersion == "Falcon 9"]
data_falcon9.isnull().sum()
payloadmean = data_falcon9['PayloadMass'].mean()
data_falcon9.replace(np.nan,payloadmean)
```

Data Collection - Scraping

Request Falcon 9 Launch data with scraping wiki page from url

```
response=requests.get("https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922")
data = requests.get(static_url).text
```

Use the python Beautifulsoup package to web scrape HTML tables

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

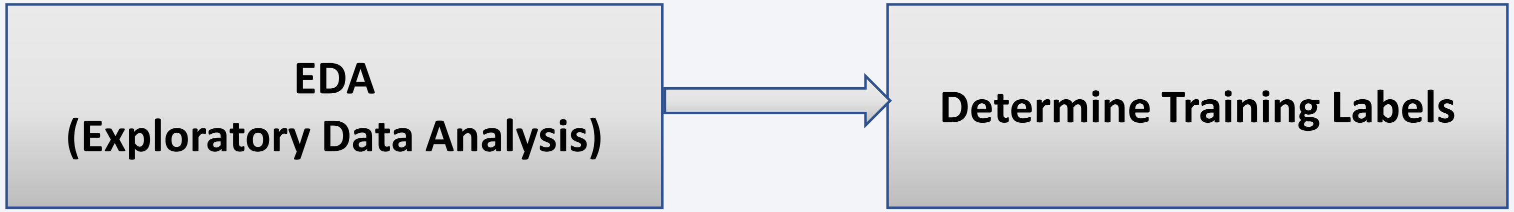
Parse the data from the tables and convert them to dataframe & Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
column_names = []
x= first_launch_table.find_all('th')
for th in x:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Source Code:
https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- First some Exploratory Data Analysis (EDA) was performed on the dataset.
- Second we calculated the number of launches on each site, the number and occurrence of each orbit and the number and occurrence of mission outcome of the orbits
- Third the landing outcome label was created from the outcome column



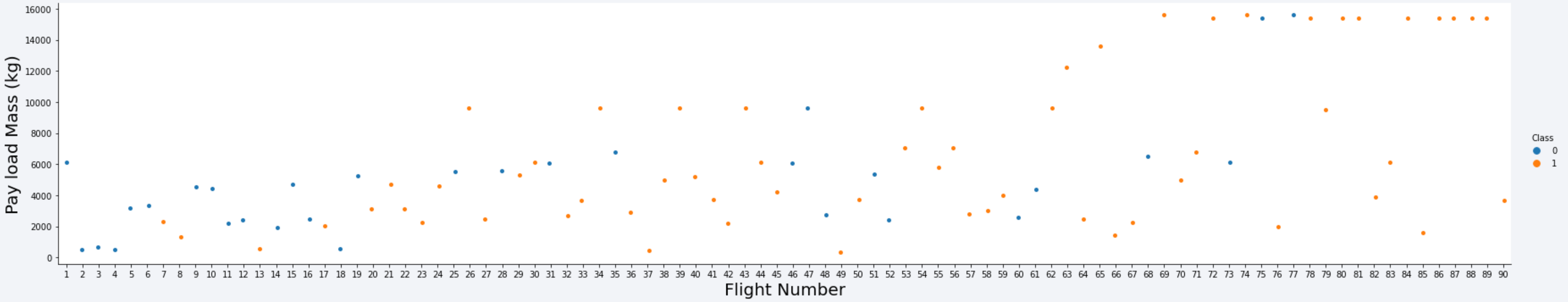
Source code:

https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

First we used scatterplots and bar plots to visualize the correlation between the features such as:

- ❖ Payload and Flight Number
- ❖ Flight Number and Launch Site
- ❖ Payload and Launch Site
- ❖ Flight Number and Orbit type
- ❖ payload and Orbit type



Source code: https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

The following SQL queries are performed:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery
- 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
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Source Code: https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We visualized the launch data into an interactive map.
- First, we marked the launch sites on a map by using sites' latitude and longitude coordinates and added a circle markers around each launch site with a label of the name of the launch site.
- Second, we marked the success/failed launches for each site to classes 0 and 1 with Red and Green markers on the map
- Third, we Calculated the distances between a launch site to its proximities such as railways, highways, coastlines and cities

Source Code:

https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allows users to explore and manipulate data in an interactive and real-time way.
- We built a dashboard application such as dropdown list, pie chart and scatter point chart.
- We plotted a pie chart showing total success launches by all sites.
- Then we plotted scatter graph showing the relationship between outcome and payload mass (Kg) for the different booster version.

Source Code:

https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/spaceX_dash.ipynb

Predictive Analysis (Classification)

Building the Model
Load the dataset with Pandas and Numpy Library.

create a column for the class

Standardize the data, Split the data into training and test data.

Train the model and use GridSearchCV to find the hyperparameters and fit it to dataset.

Evaluating the Model
Test of each model and check the accuracy.

Get tuned hyperparameters foe each type of algorithms.

Plot the confusion matrix.

Improving the Model

Find best Hyperparameter for classifiers and use feature Engineering

Find the best Model

The model with best accuracy score with be the best model.

https://github.com/Shahmohammadi-M/IBM-Data-Science-Professional-Certificate/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

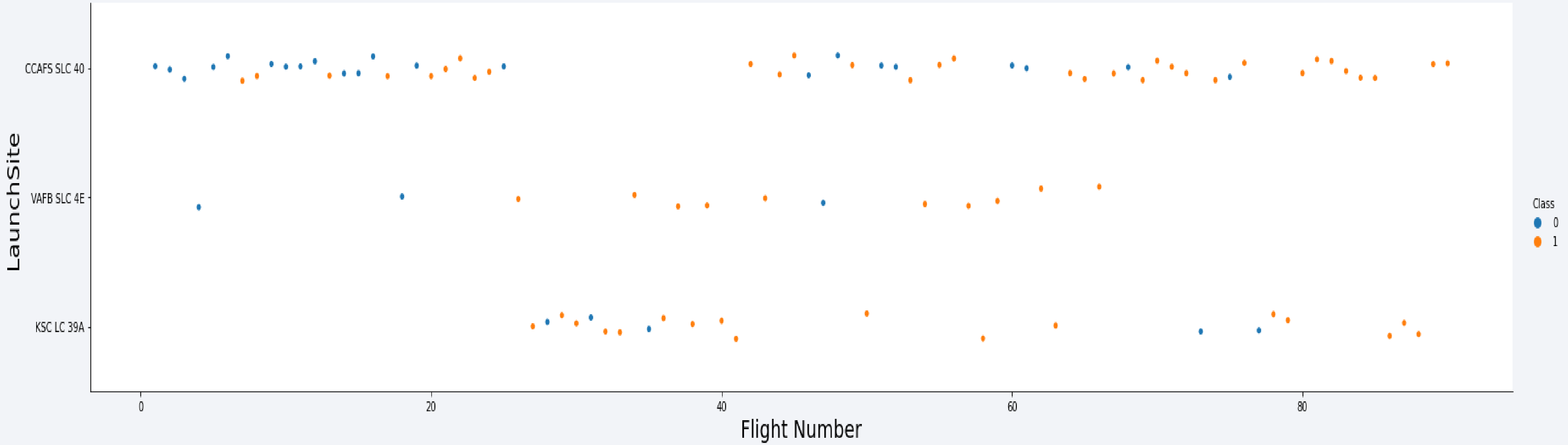
The results will be categorized to 3 main results which is:

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

Section 2

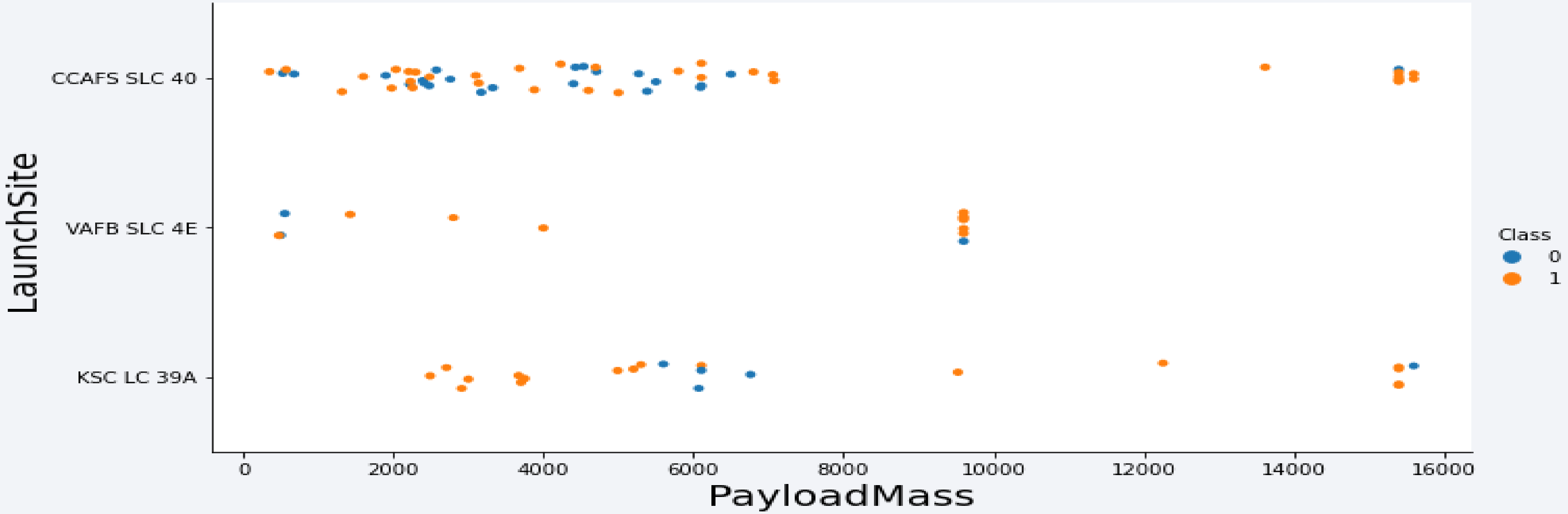
Insights drawn from EDA

Flight Number vs. Launch Site



We found that the larger the flight Number at a launch site, the greater success rate at a launch site will be.

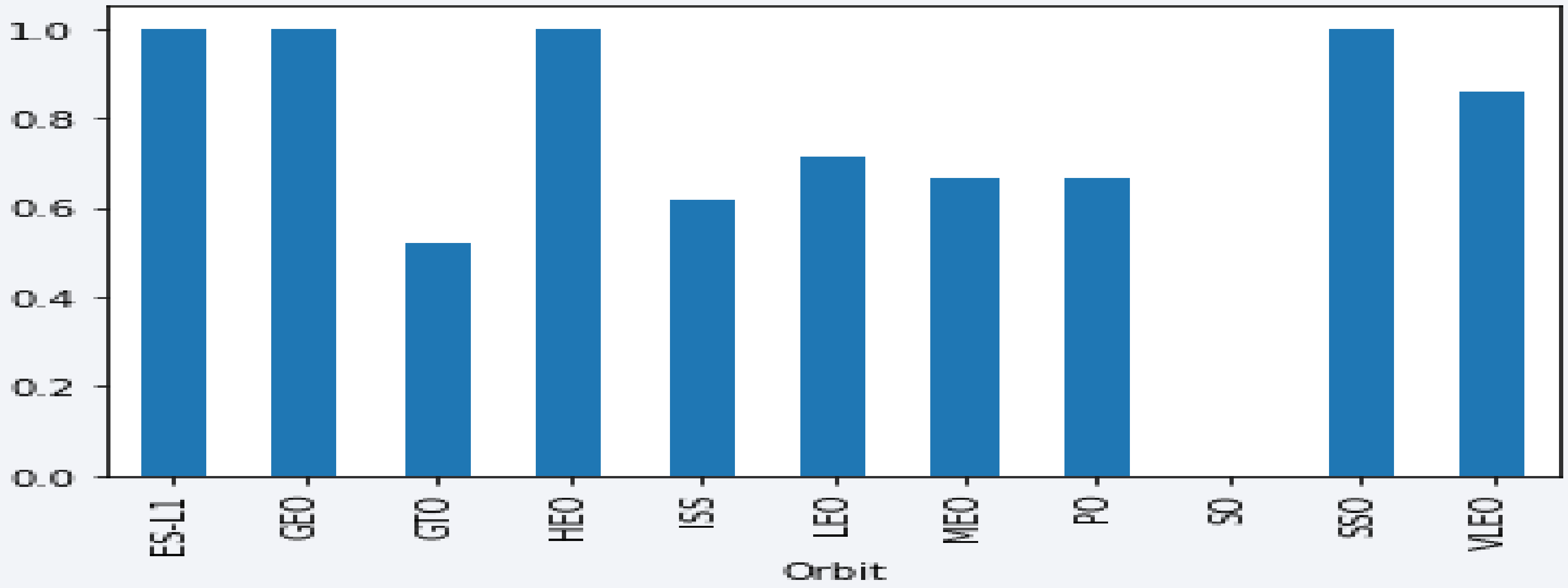
Payload vs. Launch Site



We found that the VAFB-SLC launch site, there are no rockets launched for heavy payload mass(greater than 10000).

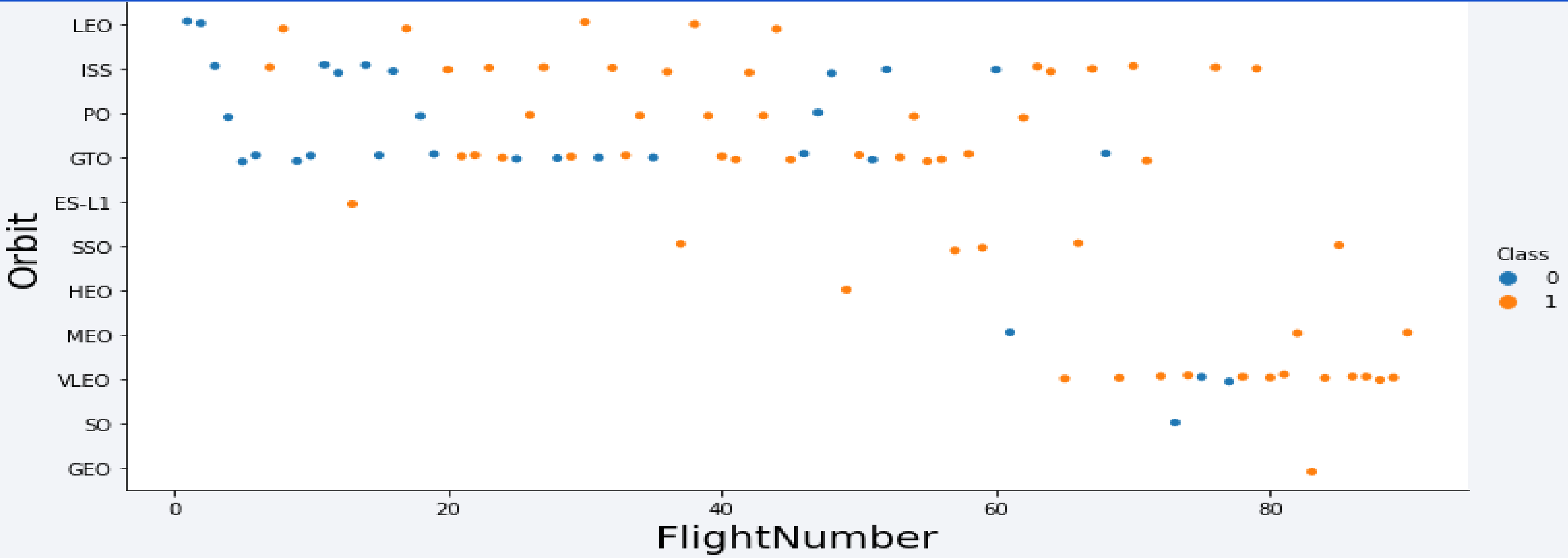
The scatter plot shows for the load mass more than 7000kg, the probability of the success rate will be increased.

Success Rate vs. Orbit Type



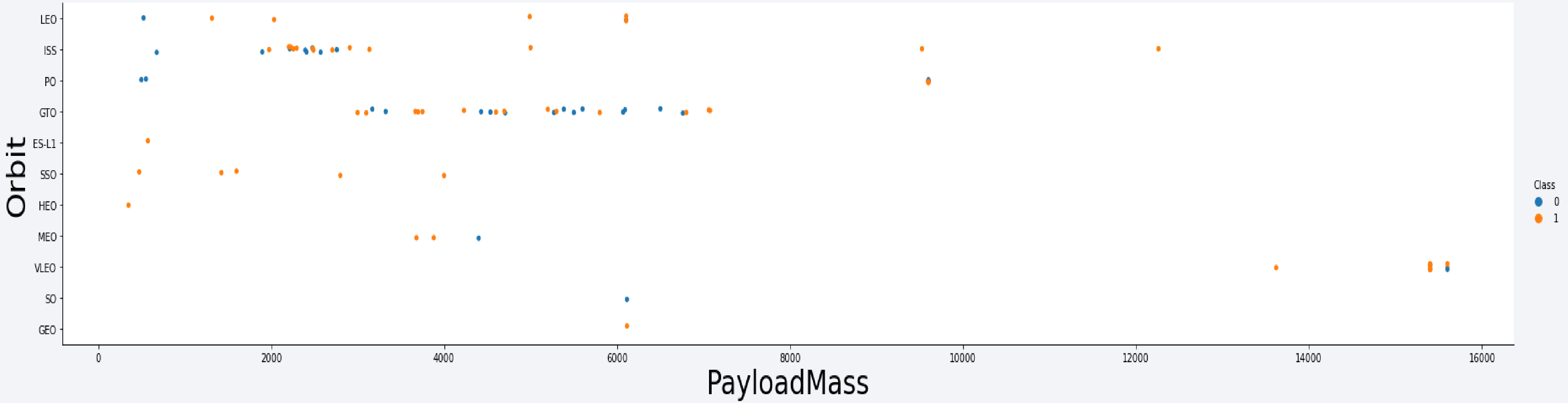
- In this figure, you can see some of the orbits have 100% success rate like ES-L1, GEO, HEO and SSO.
- The following orbit is VLEO with above 80% success rate.

Flight Number vs. Orbit Type



- This plot shows that generally, the larger flight number on each orbits have the more success rate(especially LEO) except GTO orbit that you can’t see any relation between the attributes

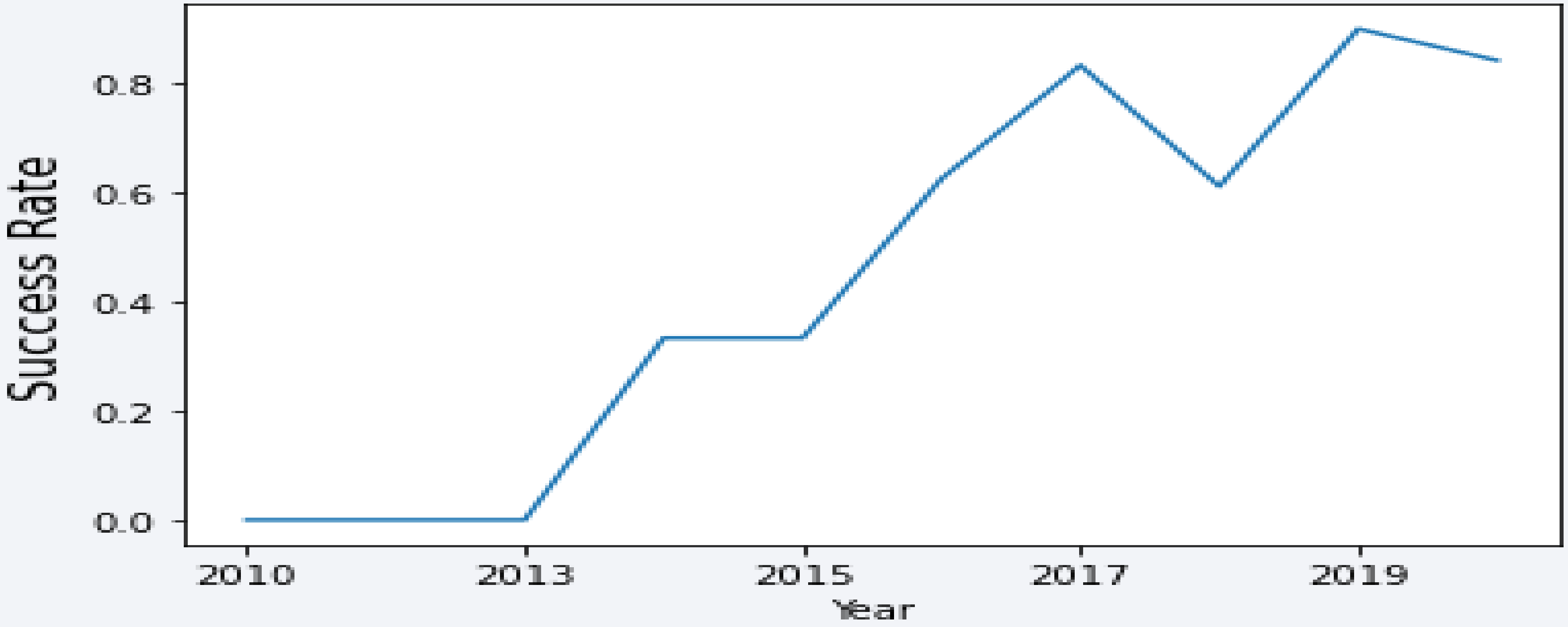
Payload vs. Orbit Type



- There is no relation between the payload and orbits for GTO.
- There are a few launches for SO, GEO and HEO to find the pattern.
- Heavier Payload has a positive impact on LEO and ISS orbits.

Launch Success Yearly Trend

- This figure clearly shows that the success rate has started increasing from year 2013 until 2020.
- The success rate can reach to 100% in the next following years.



All Launch Site Names

```
In [51]: %sql select Launch_Site from SPACEXTBL GROUP BY Launch_Site

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

Out[51]:

Launch_Site
None
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

We used group by to obtain the unique launch sites from the SpaceX dataset.

Launch Site Names Begin with 'CCA'

We displayed 5 records where launch sites begin with ‘CCA’.

In [24]:
```
%sql select launch_site from SPACEXTBL where (launch_site) like 'CCA%' limit 5;
```

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

Out[24]:

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

Total Payload Mass

```
In [49]: sql SELECT sum(PAYLOAD_MASS__KG_) as Total FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'

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

Out[49]:

Total
45596.0

- We calculated the total payload mass carried by boosters launched by NASA which was 45596.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Average Payload Mass by Booster Version F9 v1.1
2928

- Average Payload Mass carried by booster version F9 v1.1 is 2928.

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pac
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

We observed that the date of the first successful landing outcome in ground pad was 2015-12-22.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
In [37]: sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome =
'Success (drone ship)';

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

Out[37]:

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

Total Number of Successful and Failure Mission Outcomes

The number of successful and failure mission outcomes

```
In [38]: sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME

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

Out[38]:

Mission_Outcome	COUNT(*)
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Mission Outcome	Occurrences
Success	99
Success (payload status unclear)	1
Failure (in flight)	1

- **Counting the records for each group can let us know the total number.**

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass
- We used **where** clause and **Max()** function in our query

```
In [40]: sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM
         SPACEXTBL)

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

Out[40]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- 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 as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

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

Section 3

Launch Sites Proximities Analysis

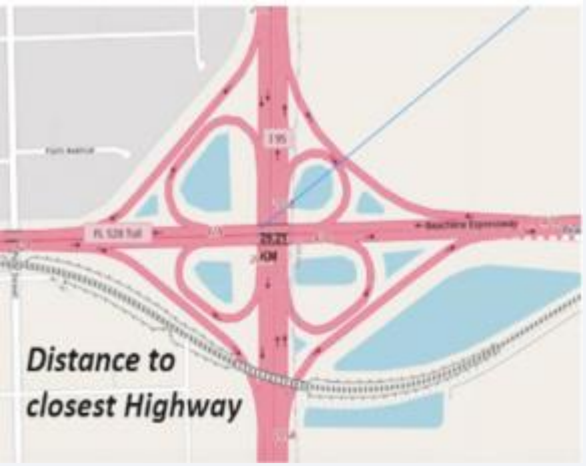
Location of all the Launch Sites

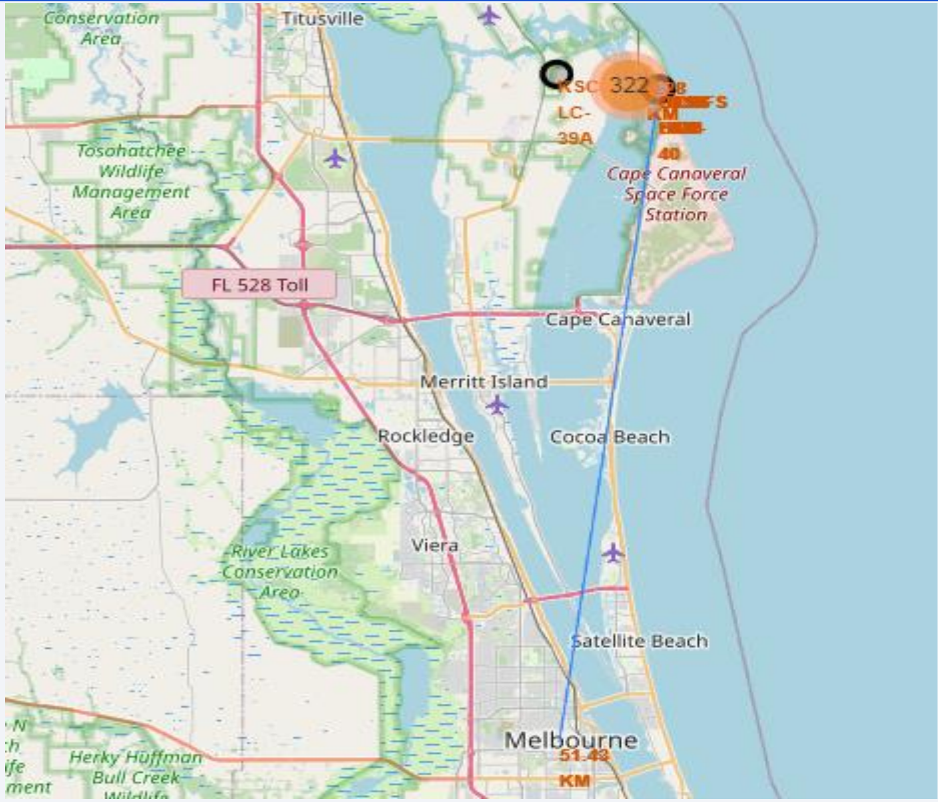
We can see that all the SpaceX launch sites are located in the United states.

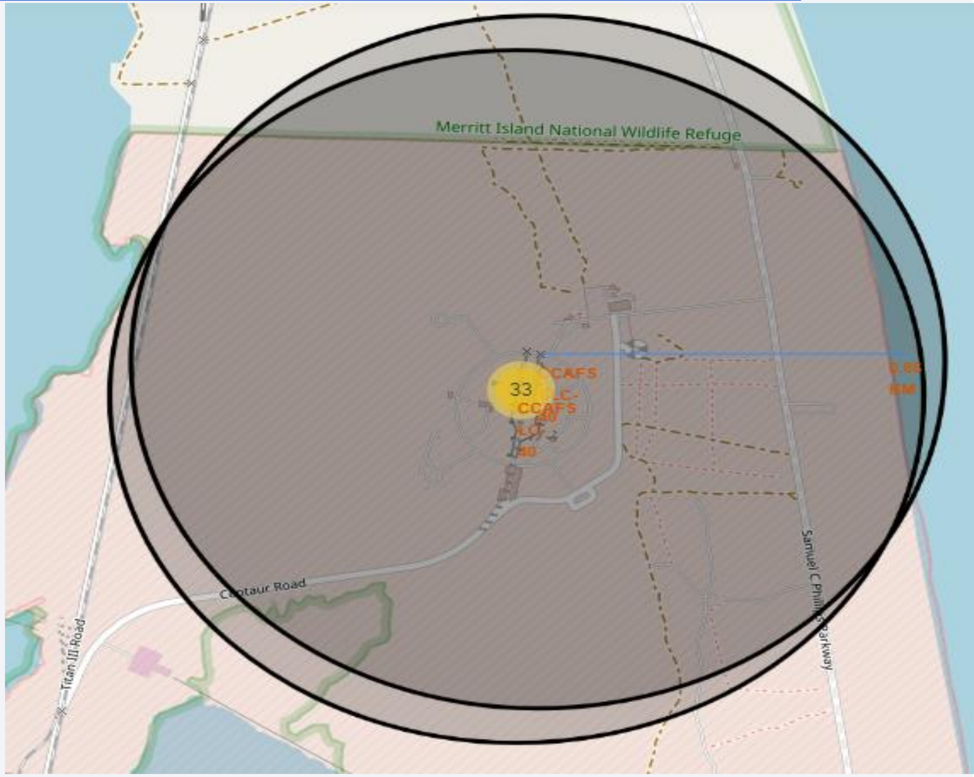
Launch sites outcome with color labels

Green Marker shows **successful** launche and **Red** Marker shows **Failure**

Launch sites Distance to Landmarks



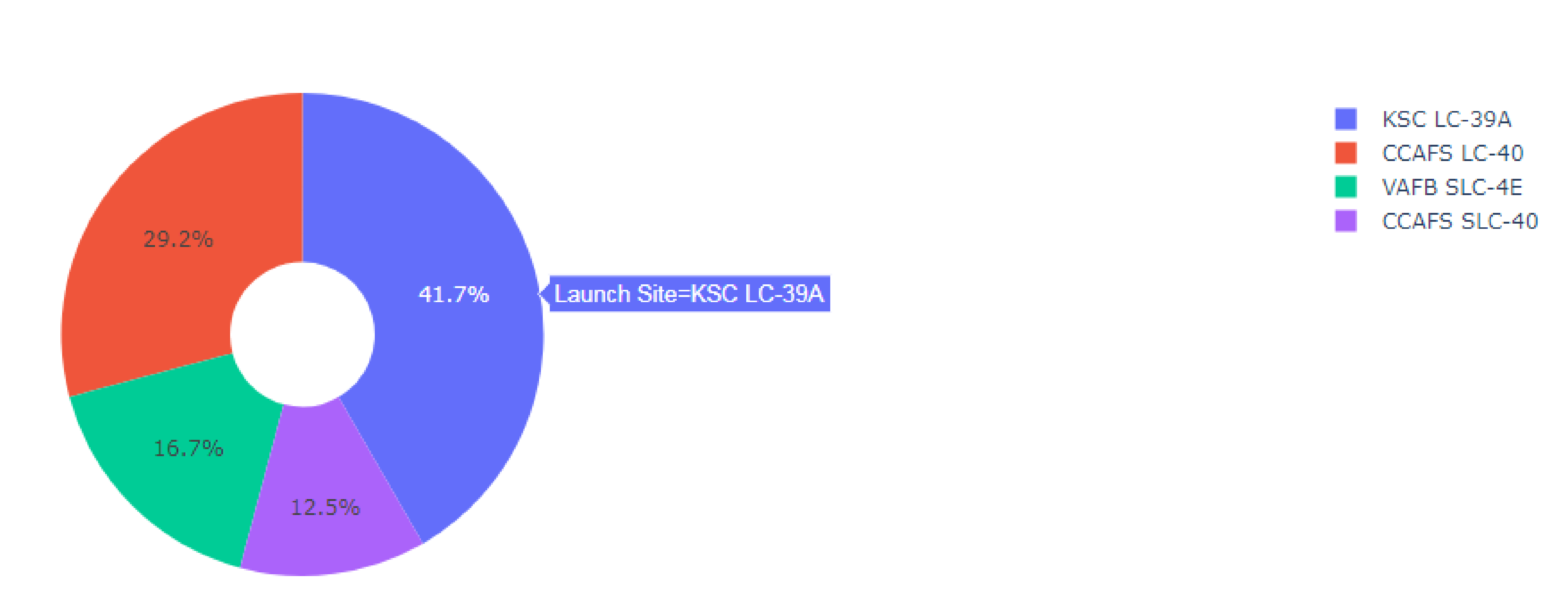




Section 4

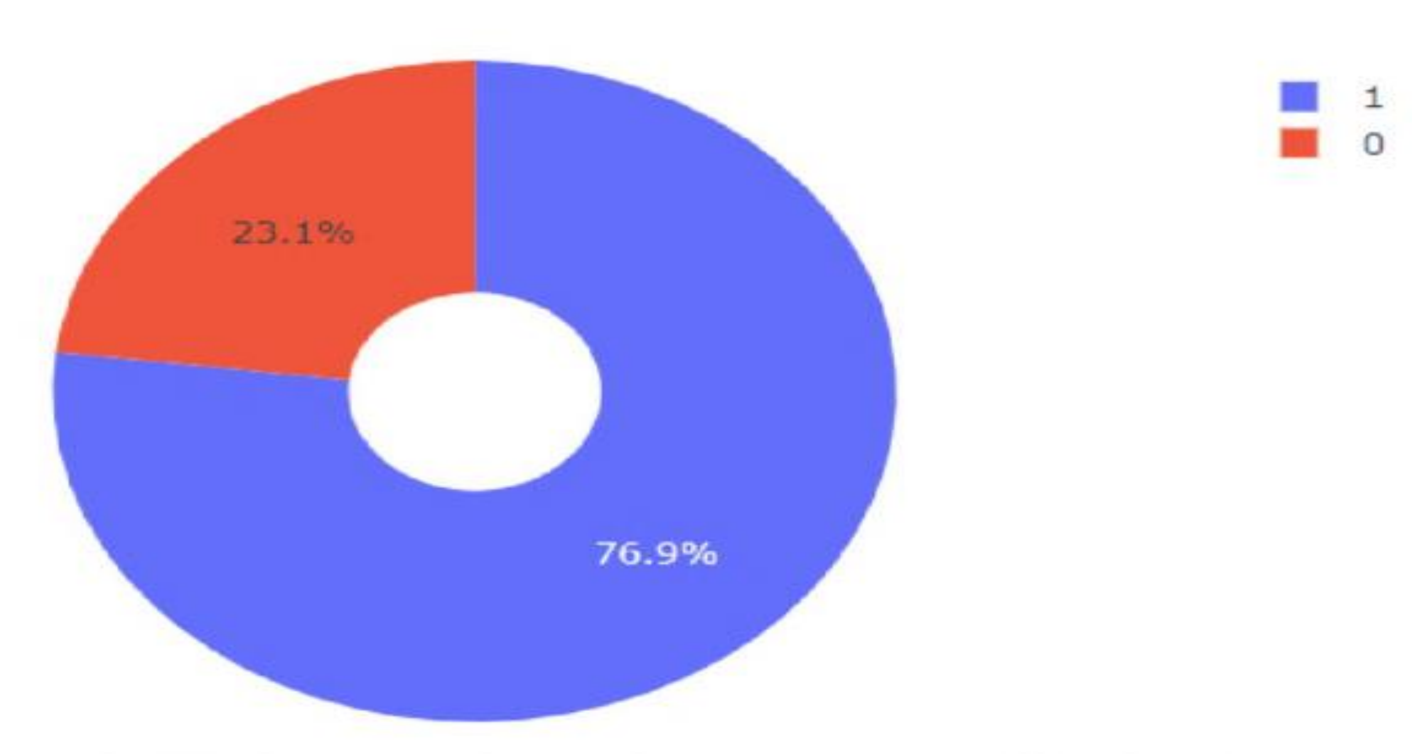
Build a Dashboard with Plotly Dash

The Success Percentage by each sites



We can see KSC LC-39A has the most successful rate in all the launch sites.

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



KSC LC-39A has a 76.9% success rate and 23.1% failure rate.

Payload Mass vs. Launch Outcome Scatter Plot

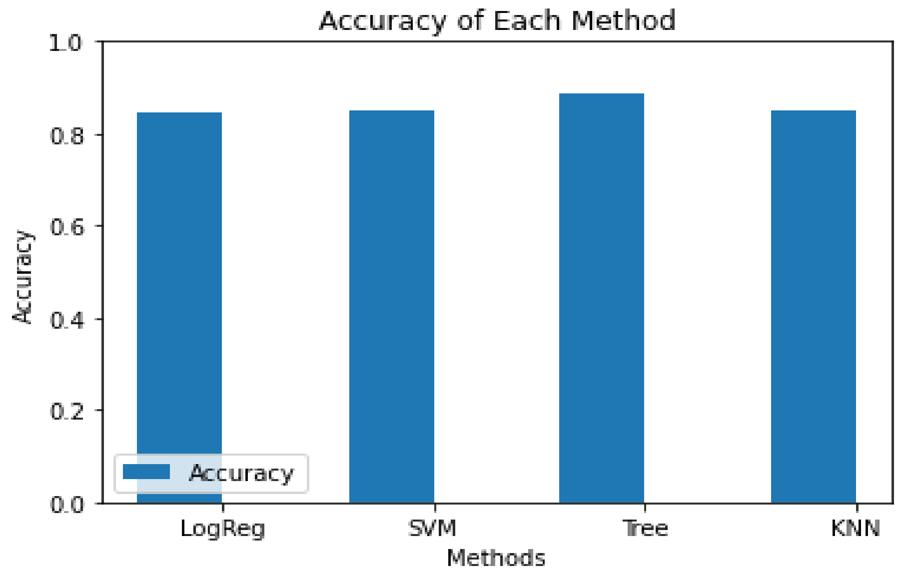
We can see all the success rate for low weighted payload (0 - 4000 kg) is higher than heavy weighted payload (4000 kg – 10000kg).

Section 5

Predictive Analysis (Classification)

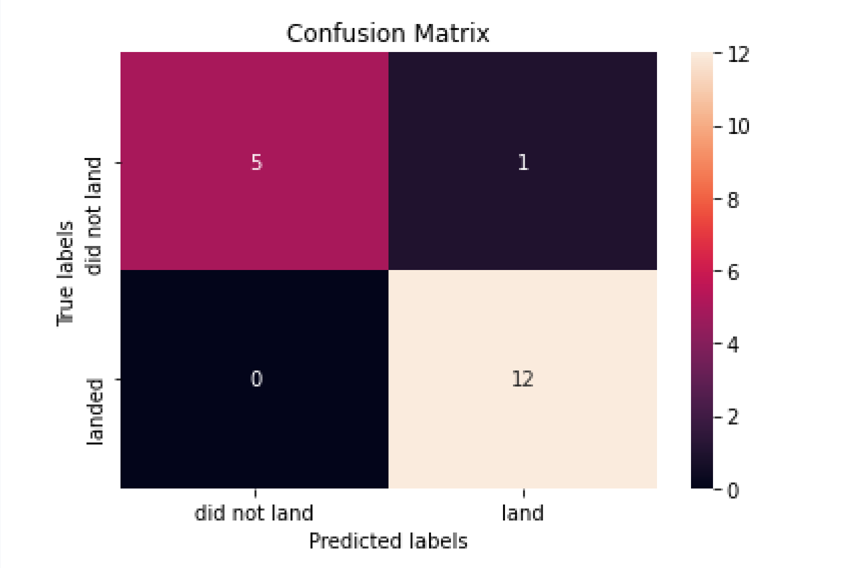
Classification Accuracy

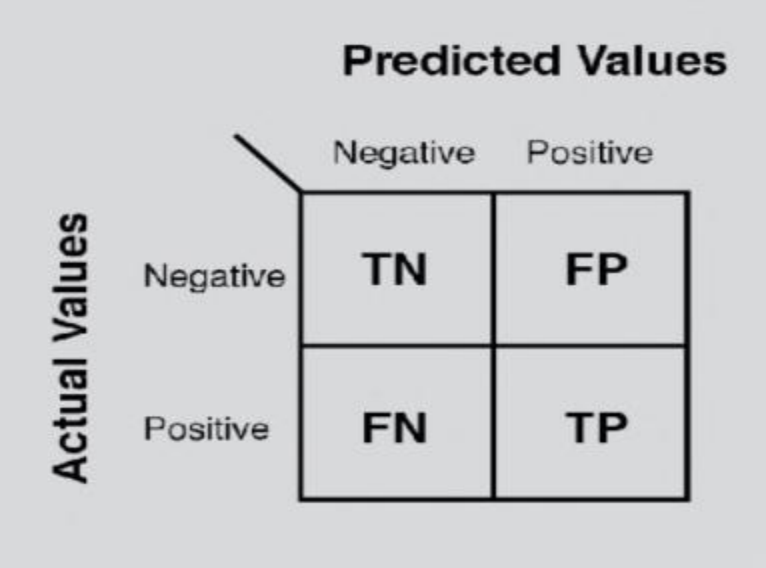
```
In [142]:

print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))

Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.9444444444444444
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```



- Four classification models tested and the accuracies are plotted (Logistics Regression, SVM, Decision Tree and K-nearest neighbors)
- The best algorithm with highest accuracy is Decision Tree.

Confusion Matrix





- The confusion matrix is the visualization of the performance of Decision Tree algorithm.
- FP: False Positive: The values which were actually negative but falsely predicted as positive.
- FN: False Negative: The values which were actually positive but falsely predicted as negative.
- The problem is the false positive which is the unsuccessful landing predicted as successful by the classifier.

Conclusions

The Decision Tree classifier is the best Machine Learning algorithms for this dataset.

The best launch site is KSC LC-39A.

The low weighted payload(4000kg and below) performed better than the heavy weighted payloads.

From the year 2013 until 2020, the success rate for SpaceX launches is increased and we can predict the success rate can reach to 100% in the next following years.

Thank you!