IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

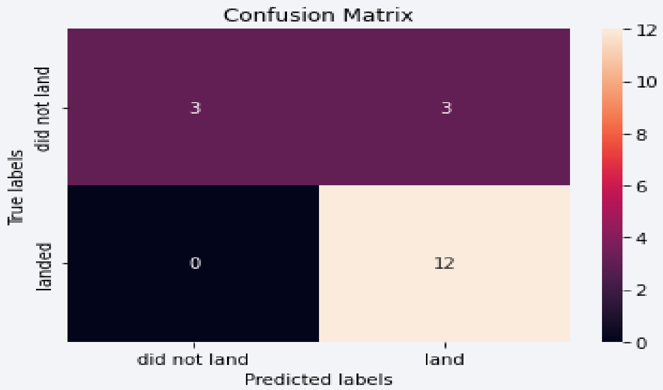
Tarun

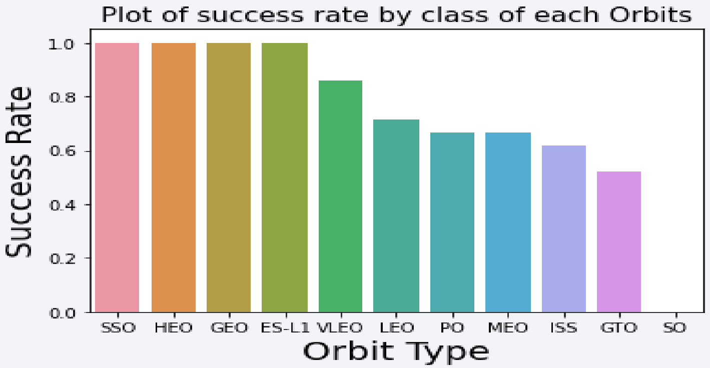
6th August, 2022

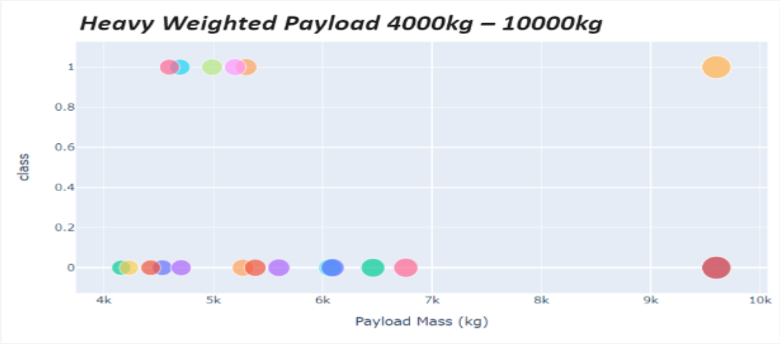
Outline

Executive Summary

- Summary of methodologies:-
    - Data Collection via API, SQL and Web Scraping
    - Data Wrangling and Exploratory Data Analysis
    - Interactive Visual Analytics with Folium
    - Machine Learning Prediction
- Summary of all results:-
    - Exploratory Data Analysis result
    - Predictive Analytics result







Introduction

- Project background and context
  - SpaceX is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. It advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.
  - Therefore if we can determine if the first stage will land, we can determine the cost of a launch.
  - This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
  - The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

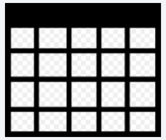
- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One hot encoding data fields for machine learning and dropping irrelevant columns (Transforming data for Machine Learning)
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and bar graphs to show patterns between data
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using Folium and Plotly Dash Visualizations
- Perform predictive analysis using classification models
  - Build and evaluate classification models

Data Collection

- Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

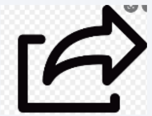
1. Getting Data from API or Web Page

2. Make Dataframe from it



3. Filter **Dataframe** as per requirement

4. Export to flat file



Data Collection – SpaceX API

- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

```
In [12]: # Use json_normalize method to convert the json result into a dataframe

         # decode response content as json
         static_json_df = res.json()
```

```
In [13]: # apply json_normalize
         data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

```
In [30]: rows = data_falcon9['PayloadMass'].values.tolist()[0]

         df_rows = pd.DataFrame(rows)
         df_rows = df_rows.replace(np.nan, PayloadMass)

         data_falcon9['PayloadMass'][0] = df_rows.values
         data_falcon9
```

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/jupyter-labs-webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]: 200

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

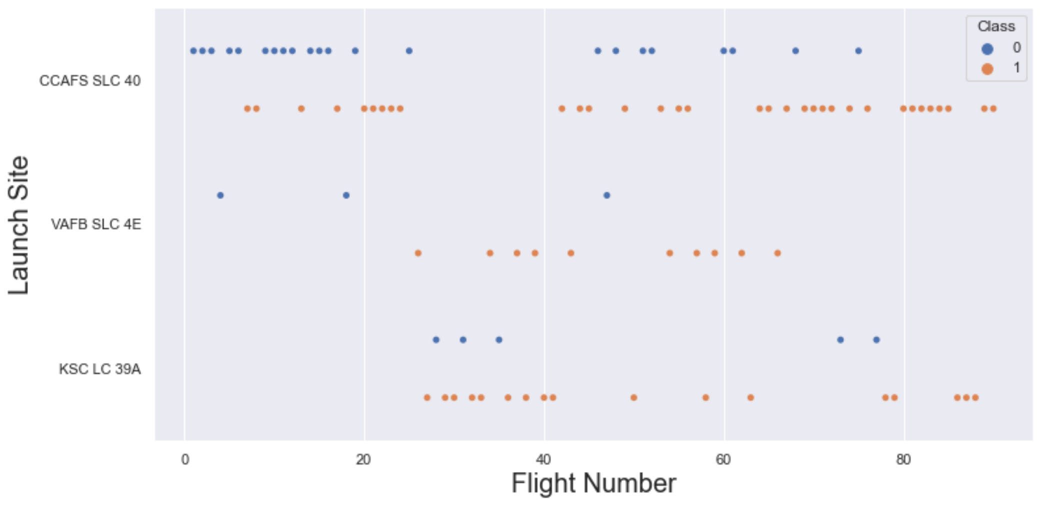
4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

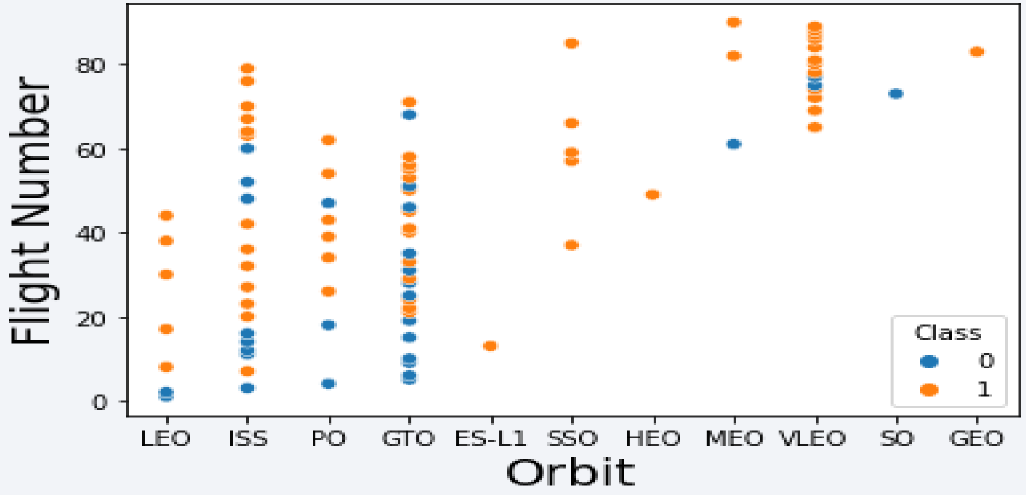
Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site and the number and occurrence of each orbits
- We created landing outcome label from outcome column.
- Separating the successful and unsuccessful landed and its result is stored in a variable which represent a classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

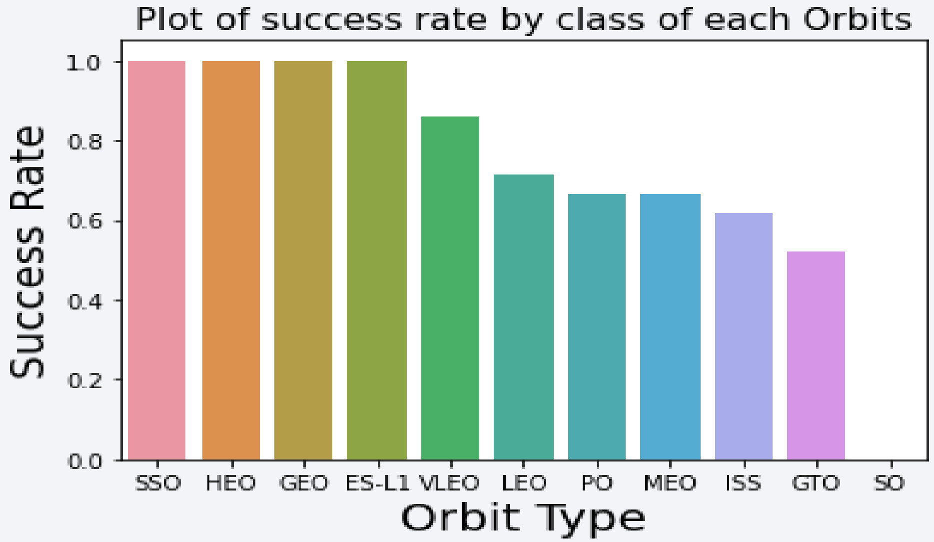
- We first started by using scatter graph to find the relationship between the attributes such as between:
  - Payload and Flight Number.
  - Flight Number and Launch Site.
  - Payload and Launch Site.
  - Flight Number and Orbit Type.
  - Payload and Orbit Type.
- Scatter plots show dependency of attributes 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.
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/jupyter-labs-eda-dataviz.ipynb

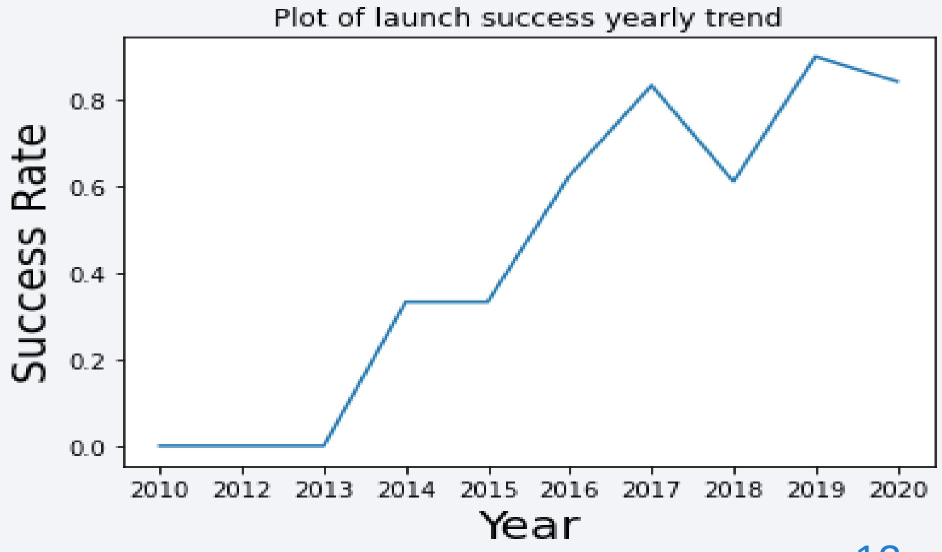




EDA with Data Visualization

- 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.
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/jupyter-labs-eda-dataviz.ipynb





EDA with SQL

- We loaded the SpaceX dataset using the database console LOAD tool in DB2
- SQL queries were performed to find:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The first successful landing outcome in ground pad
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
  - the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/jupyter-labs-eda-sql-coursera_sqllite.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:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We made a Launch Site Drop-down to select different sites
- We plotted pie charts showing the total launches by a certain or all sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- We also added Range slider to select different Payload.
- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

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

Evaluating the Model

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

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

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

- The link to the notebook is https://github.com/Tarun1204/Applied_Data_Science/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

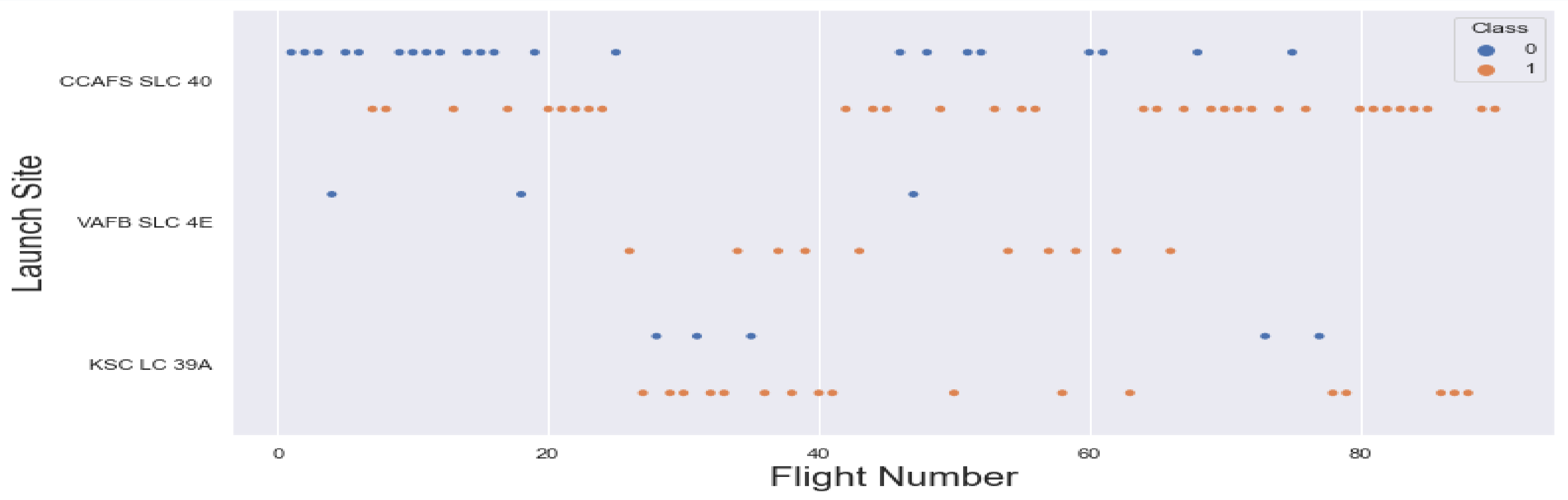
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

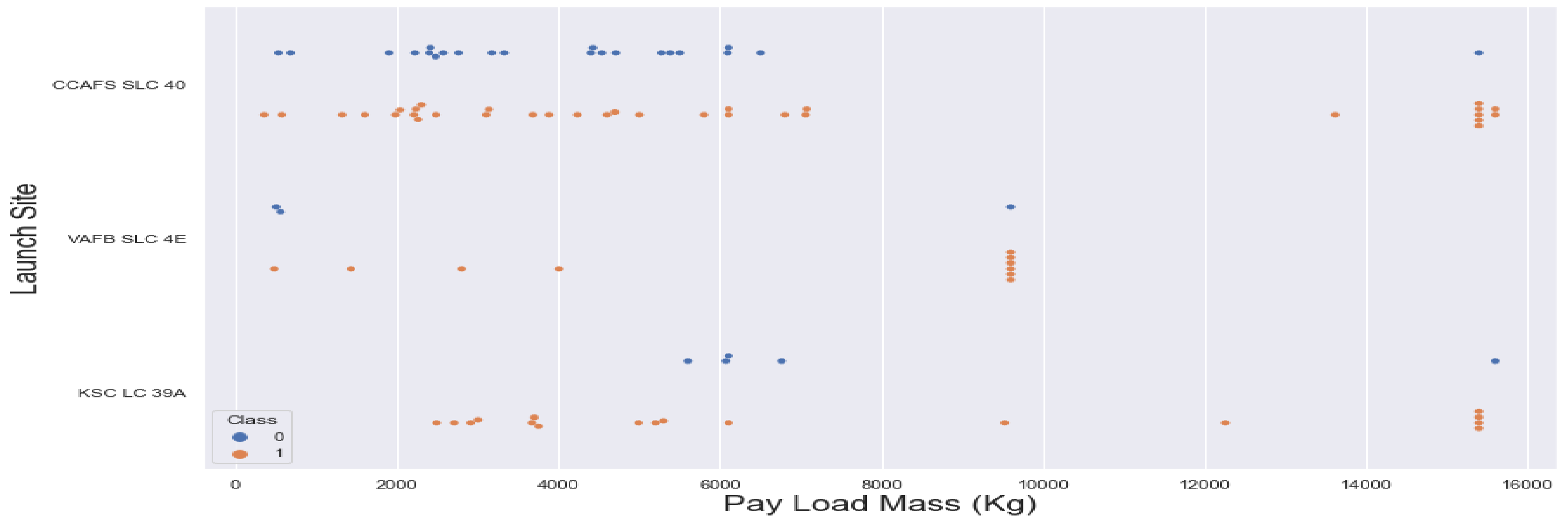
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.
- However, site CCAFS SLC40 shows the least pattern of this.



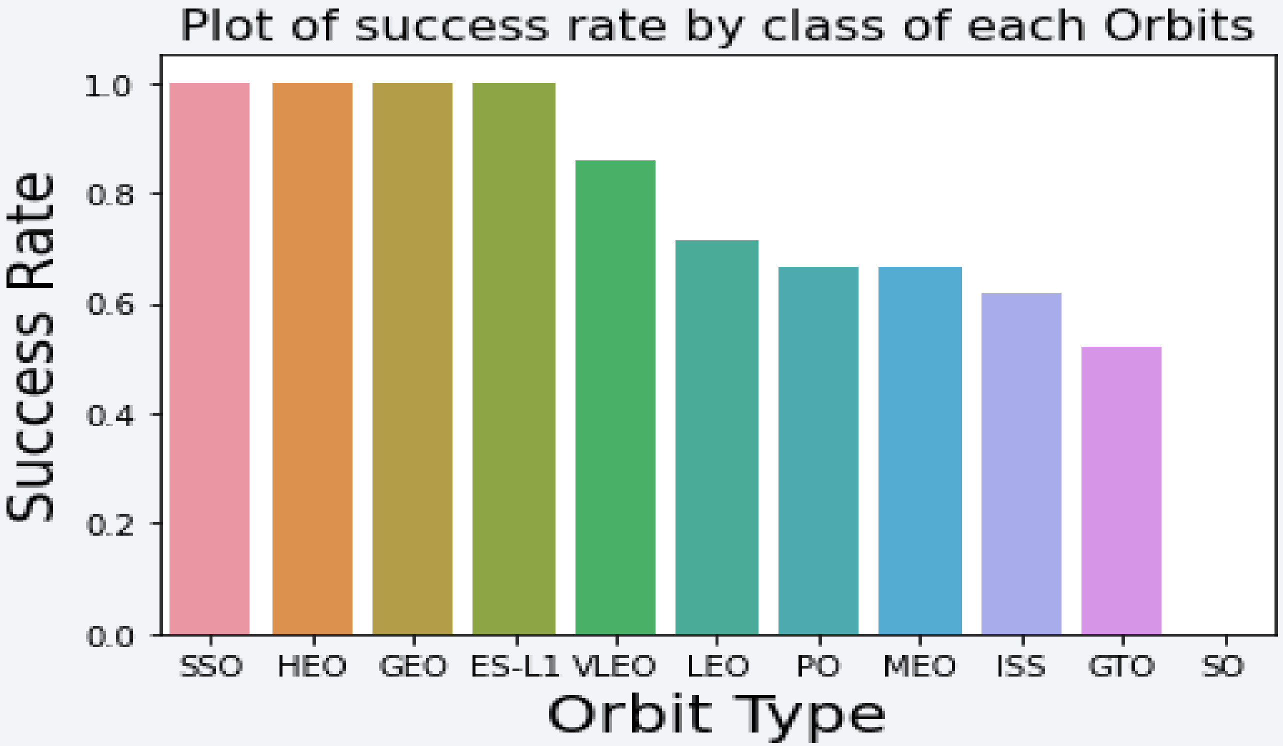
Payload vs. Launch Site

- The greater the payload for CCAFS SLC 40 the higher the success rate for the rockets.
- VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



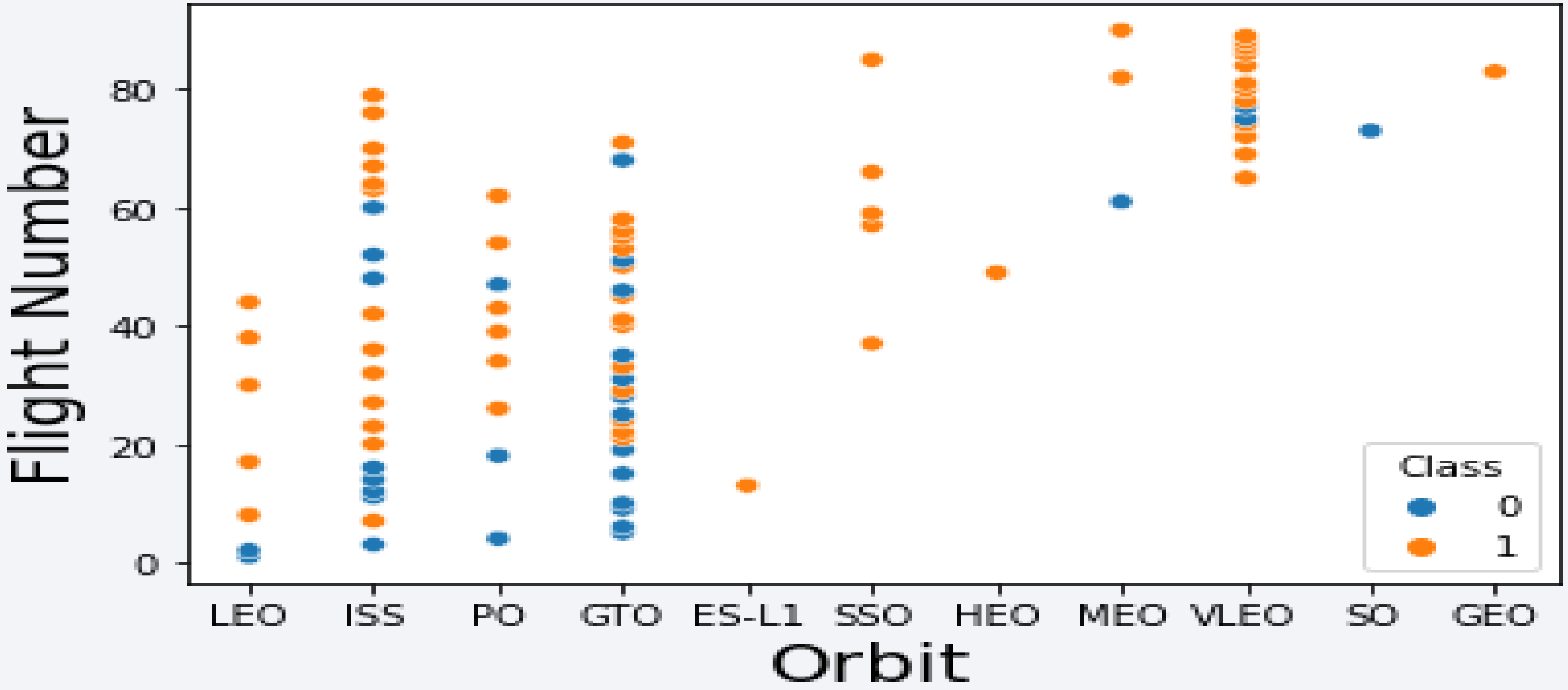
Success Rate vs. Orbit Type

- This 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 produced 0% 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.



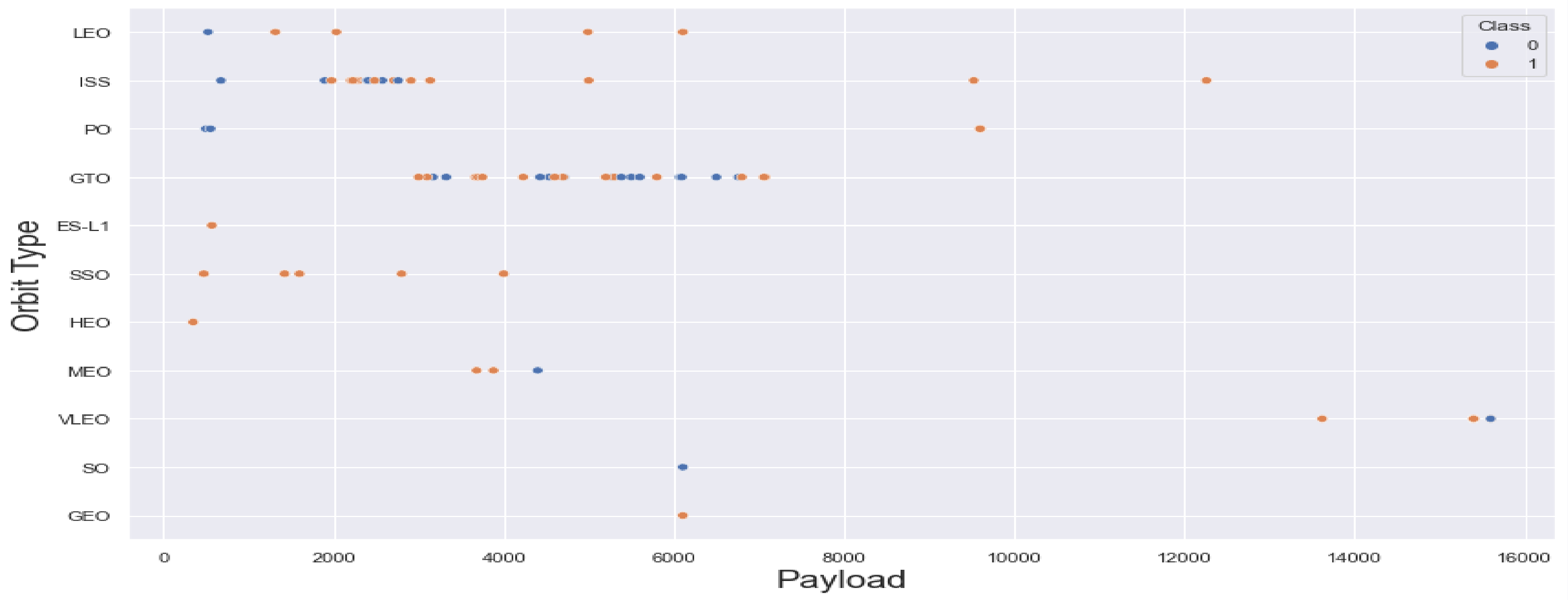
Flight Number vs. Orbit Type

- We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.
- Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.



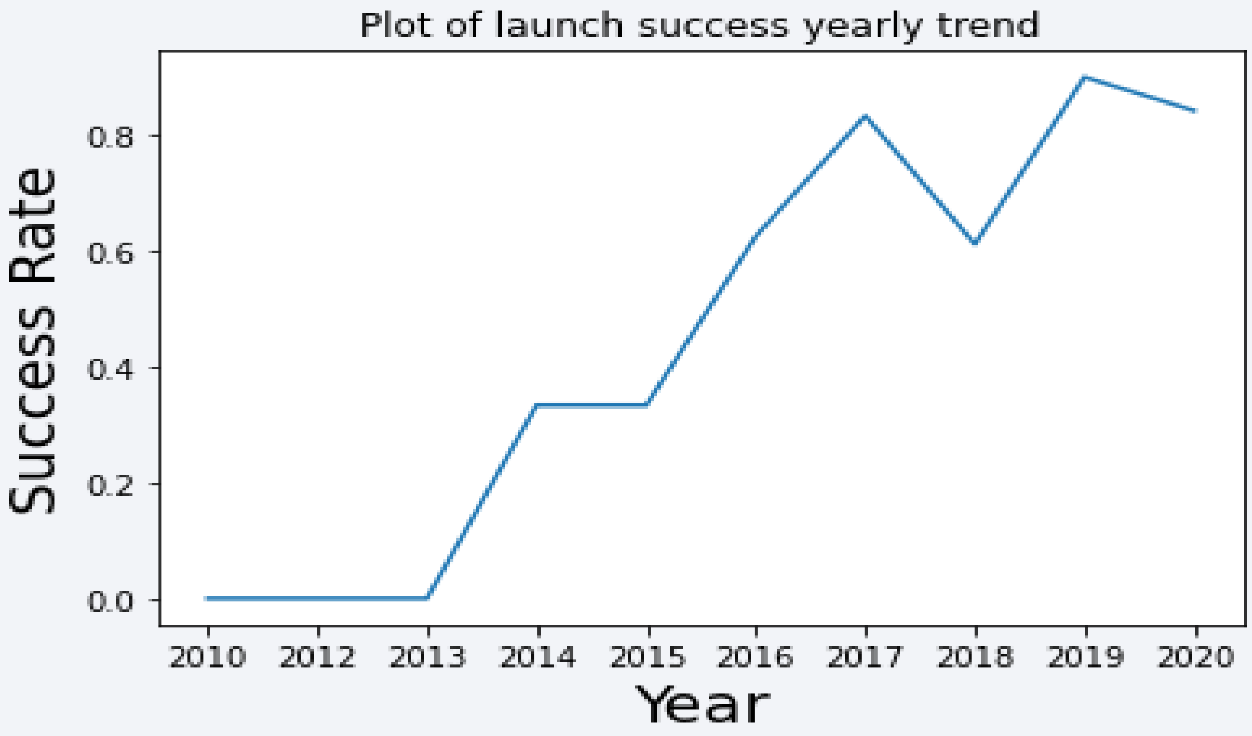
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing are both there.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.
- If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

- We used the key word **Unique** to show only unique launch sites from the SpaceX data

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

```
In [7]: %sql select unique(launch_site) from SPACEXTBL

 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

- We used the ‘like %’ and ‘limit’ functions in query to display 5 records where launch sites begin with `CCA`

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

In [8]:
```
%sql select * from SPACEXTBL where launch_site like 'CCA%' \
limit 5
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[8]:

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

Total Payload Mass

- We calculated the total payload carried by boosters by NASA(CRS) as 45596 using the sum and where functions in query below.

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

In [9]:
```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[9]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

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

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

In [10]:
```
%sql select avg(payload_mass__kg_) as avg_payload_mass from SPACEXTBL where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[10]:

avg_payload_mass
2928

First Successful Ground Landing Date

- We observed that the date of the first successful landing outcome on ground pad was 22nd December 2015

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

Hint:Use min function

```
In [11]: %sql select min(date) as First_Successfull_landing_date from SPACEXTBL where landing__outcome = 'Success (ground pad)'

 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[11]:

first_successfull_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for 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

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

In [12]:
```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)' and \
payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[12]:

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

Total Number of Successful and Failure Mission Outcomes

- Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.
- Case when MISSION OUTCOME LIKE '%Success%' then 1 else O end" returns a Boolean value which we sum to get the result needed.

In [34]:
```
%sql select sum(case when mission_outcome LIKE '%Success%' then 1 else 0 end) as Successful_Mission,\
sum(case when mission_outcome LIKE '%Failure%' then 1 else 0 end) as Failure_Mission \
from SPACEXTBL;
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[34]:

successful_mission	failure_mission
100	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.

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

```
In [14]: %sql select booster_version , payload_mass__kg_ from SPACEXTBL where \
payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL ) \
order by booster_version
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[14]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015 and for month we used **TO_CHAR(TO_DATE())** function.

In [18]:
```
%sql select TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME,booster_version ,\
launch_site,landing__outcome  from SPACEXTBL where landing__outcome = 'Failure (drone ship)'\
and date LIKE '%2015%'
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[18]:

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

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 2010-03-20.
- 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.

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

In [20]:
```
%sql select 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(landing__outcome) DESC
```

```
 * ibm_db_sa://hlj04941:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

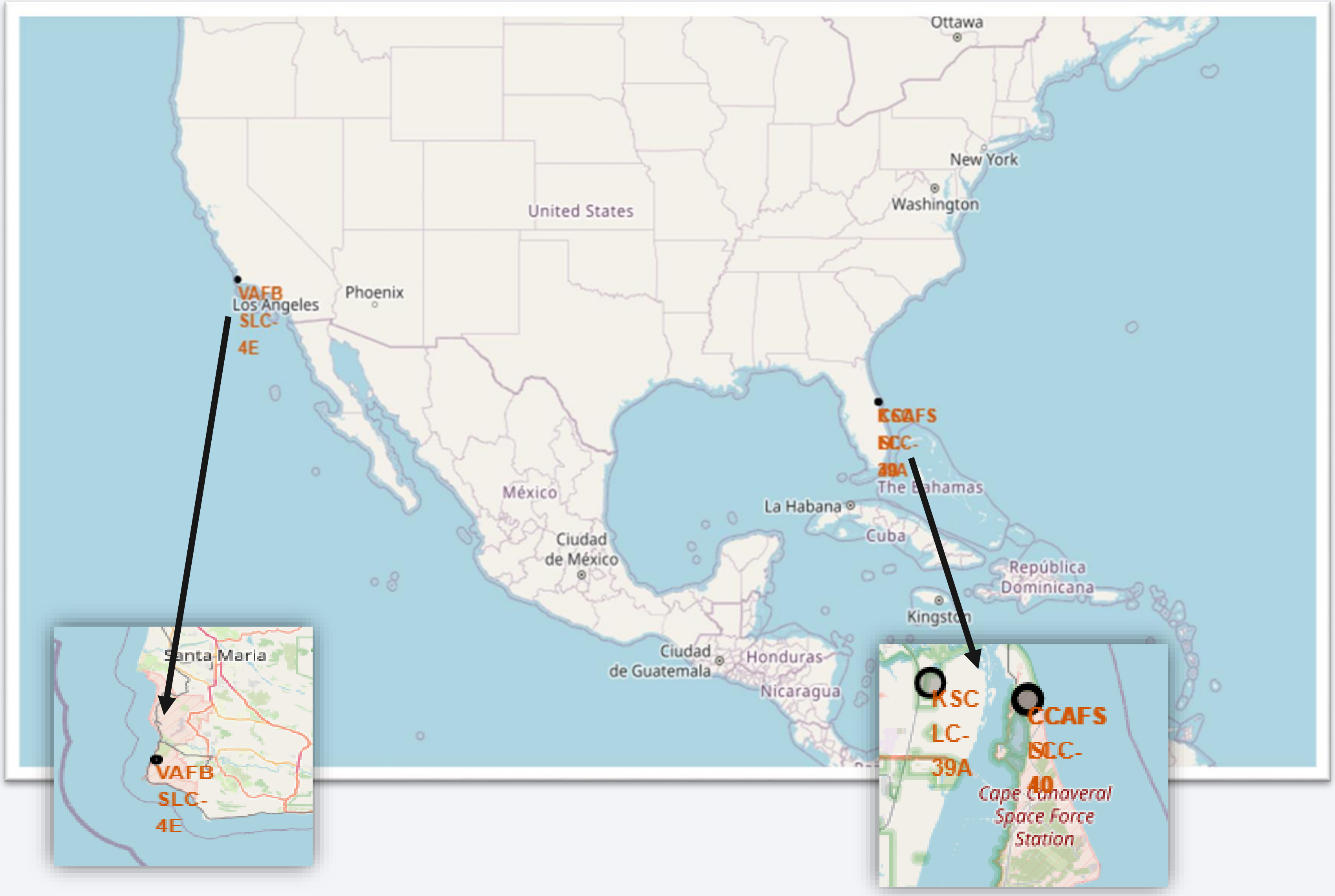
Out[20]:

landing__outcome	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

Section 3

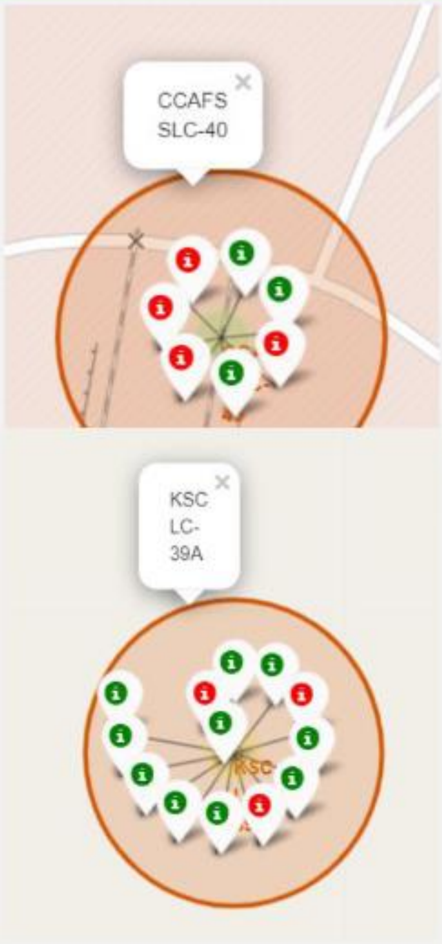
Launch Sites Proximities Analysis

All launch sites global map markers



- We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions.

Markers showing launch sites with color labels



Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

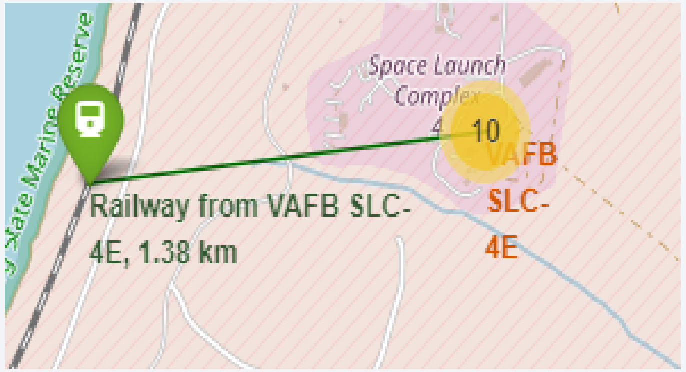
California Launch Site

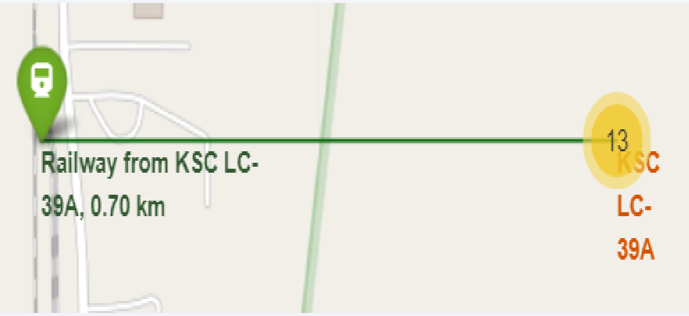
37

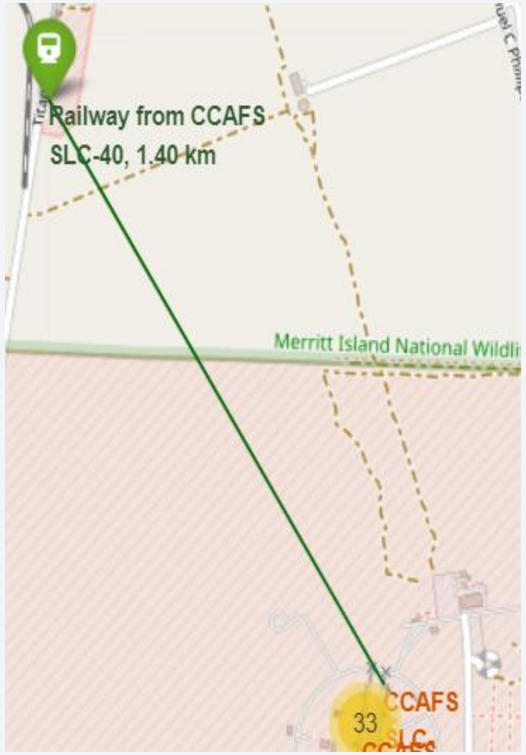
Launch Site Distances from Equator & Railway

Distance from Equator is greater than 3000 Km for all sites.

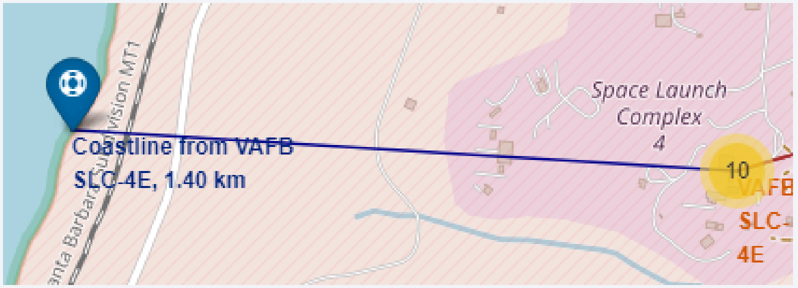
Distance for all launch sites from railway tracks are greater than .7 Km for all sites. So, launch sites are not so far away from railway tracks.

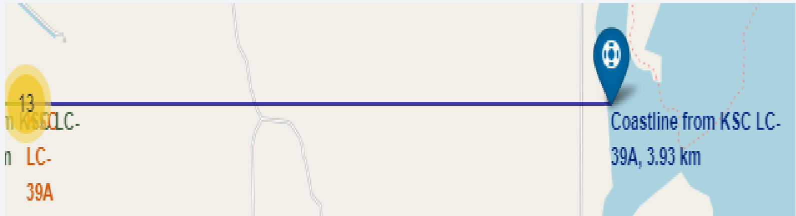






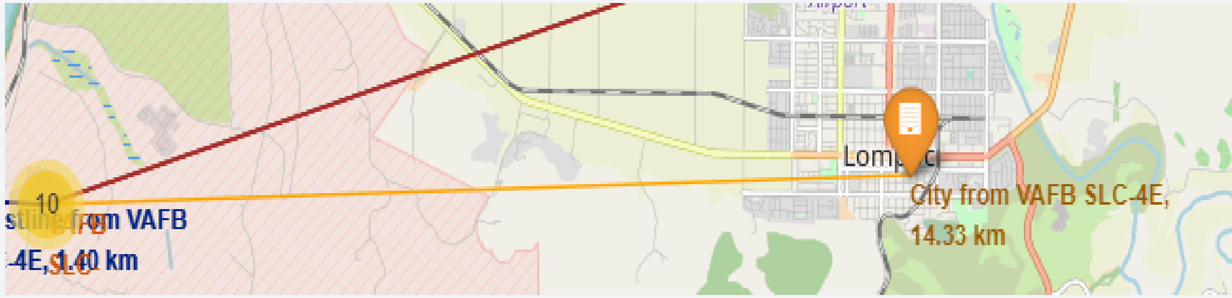
Launch Site Distances from Coastlines & Cities

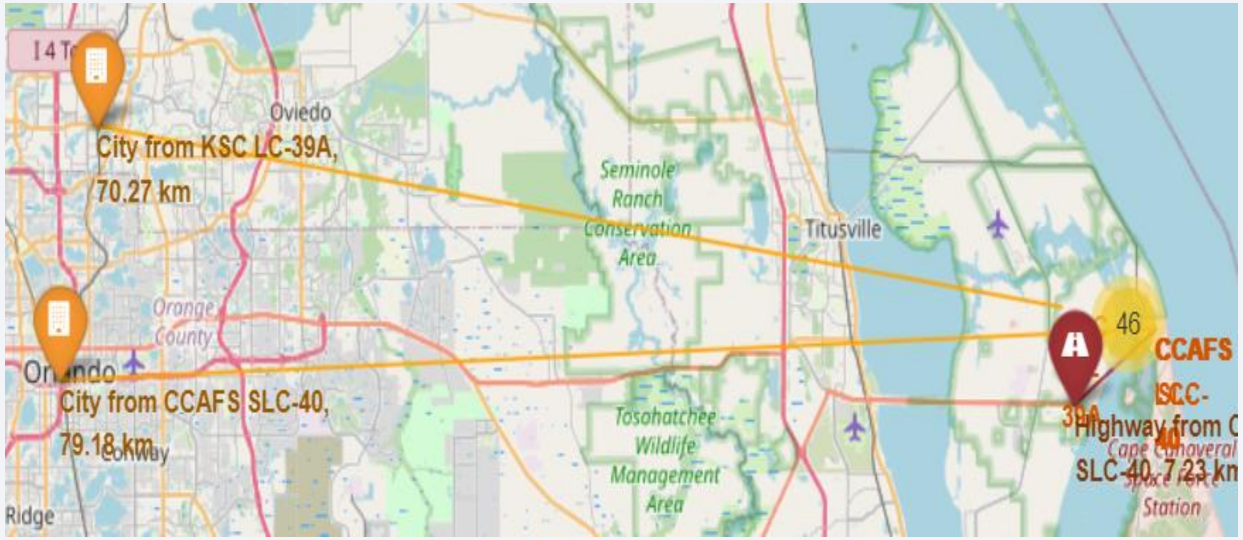




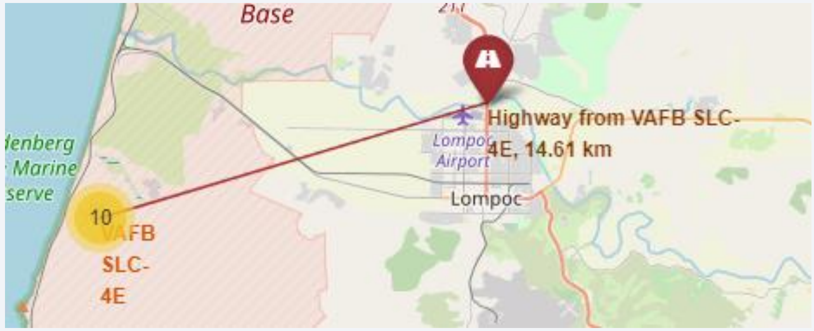
Distance for all launch sites from coastline is less than 4 Km.

Distance for all launch sites from cities is greater than 14 Km for all sites. So, launch sites are far away from cities.





Launch Site Distances from Highways



Distance for all launch sites from highways is greater than 5 Km for all sites. So, launch sites are relatively far away from highways.

Conclusion:

- Are all launch sites in proximity to the Equator line?

  No (4000 Km> distance> 3000 Km}
- Are launch sites in close proximity to railways?

  Yes {2 Km> distance> .5 Km)
- Are launch sites in close proximity to highways?

  No {15 Km> distance> 5 Km)
- Are launch sites in close proximity to coastline?

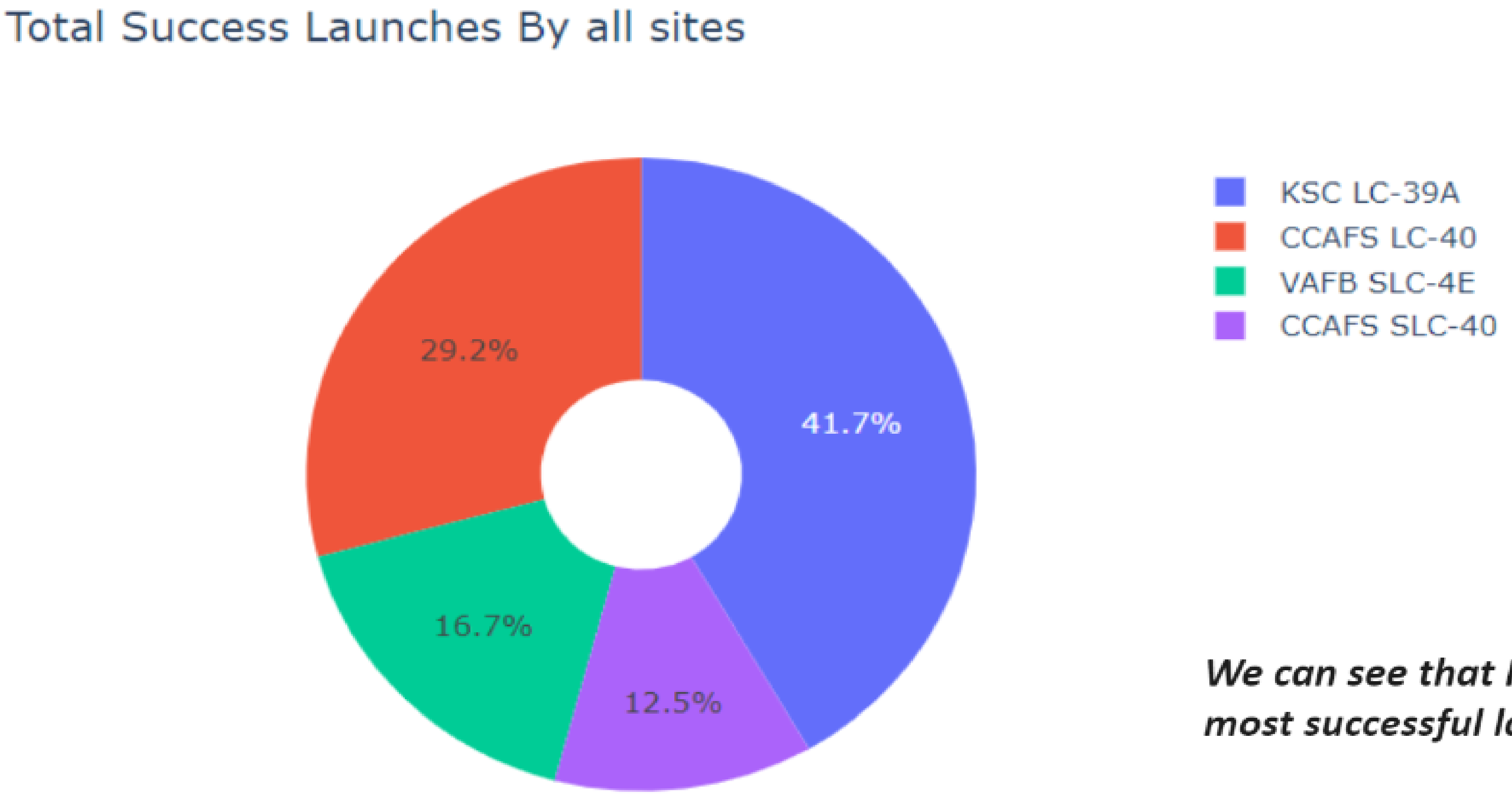
  Yes (5 Km> distance> .5 Km)
- Do launch sites keep certain distance away from cities?

  Yes (15 Km> distance> 80 Km)

Section 4

Build a Dashboard with Plotly Dash

Success percentage achieved by each launch site

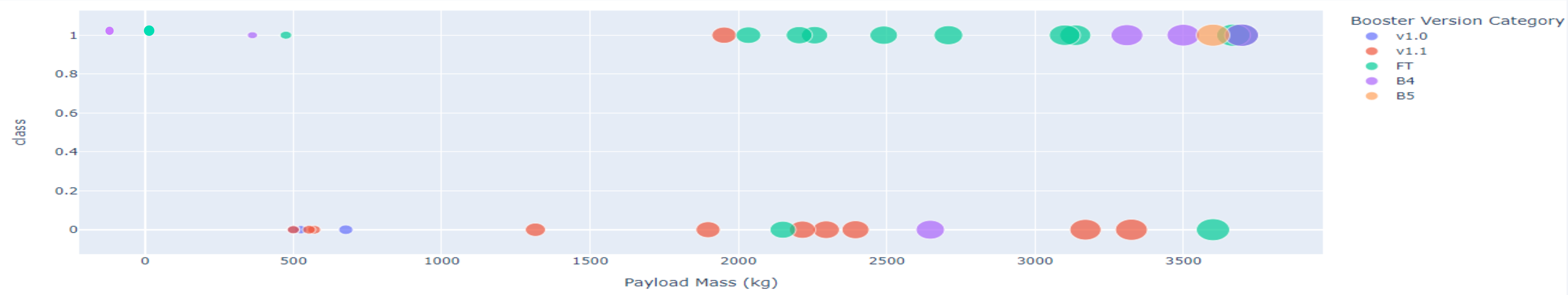


We can see that KSC LC-39A had the most successful launches from all the sites

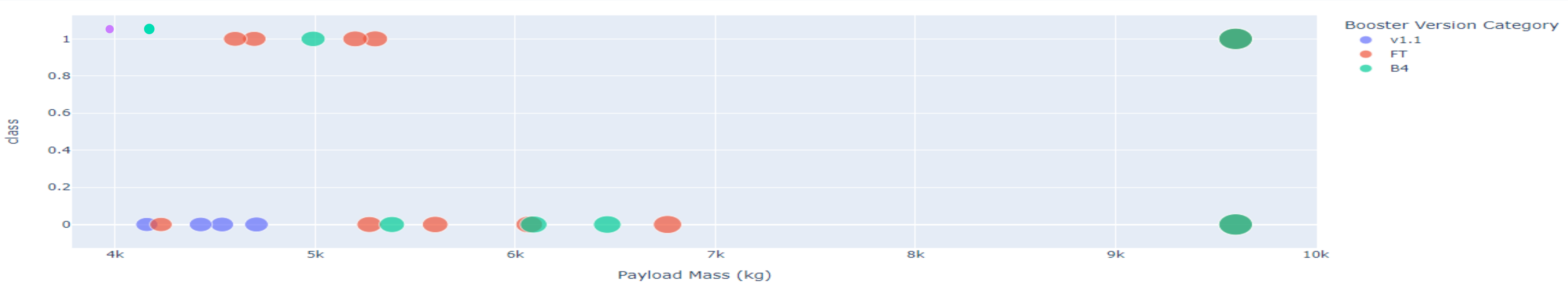
Scatter plot of Payload vs Launch Outcome

- We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Low Weighted Payload 0kg-4000kg



High Weighted Payload 4000kg – 10000kg



Launch site with the highest launch success ratio

After visual analysis using the dashboard, we are able to obtain some insights to answer these questions:

- Which site has the highest launch success rate?

  KSC LC-39A

- Which payload range(s) has the highest Launch success rate?

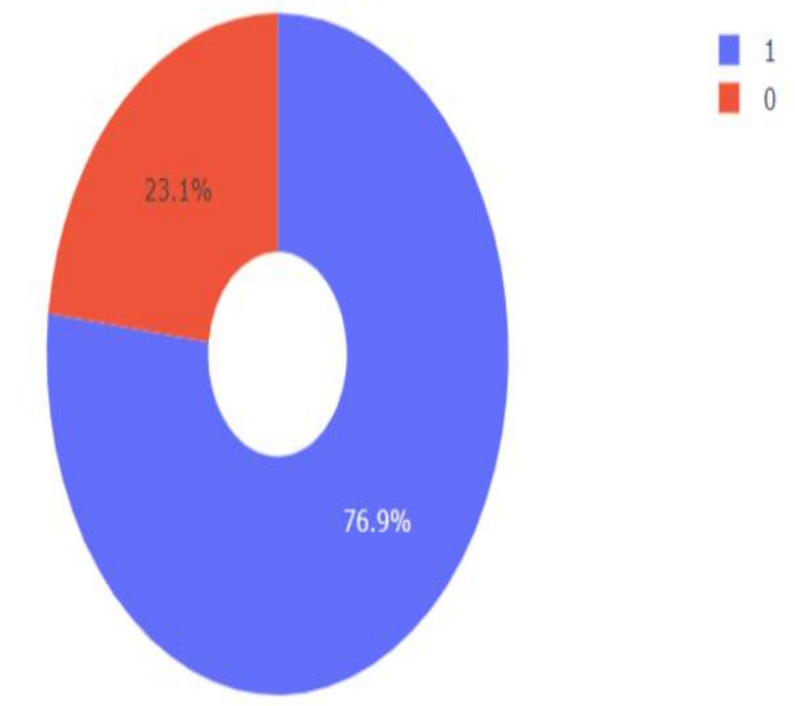
  2000 Kg- 10000 Kg

- Which payload range(s} has the lowest launch success rate?

  0 Kg-1000 Kg

- Which F9 Booster version (vl.O, vl.1, FT, B4, BS, etc.} has the highest launch success rate?

  FT



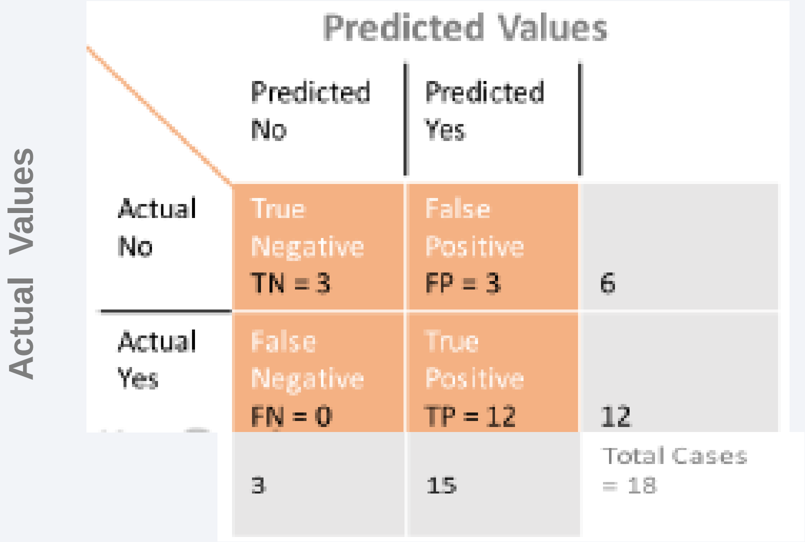
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Section 5

Predictive Analysis (Classification)

Confusion Matrix

Out here for all models unfortunately, we have same confusion matrix.



Actual Values	Predicted Values: Predicted No	Predicted Yes	
Actual No	True Negative TN = 3	False Positive FP = 3	6
Actual Yes	False Negative FN = 0	True Positive TP = 12	12
	3	15	Total Cases = 18

Accuracy: {TP+TN)/Total = {12+3)/18 = 0.83333

Misclassification Rate: (FP+FN)/Total = (3+0)/18 = 0.1667

True Positive Rate: TP/Actual Yes= 12/12 = 1

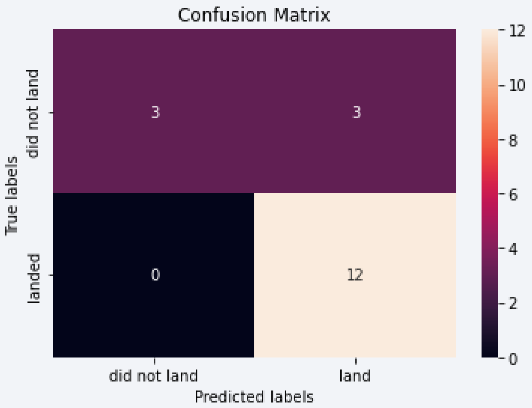
False Positive Rate: FP/Actual No= 3/6 = 0.5

True Negative Rate: TN/Actual No= 3/6 = 0.5

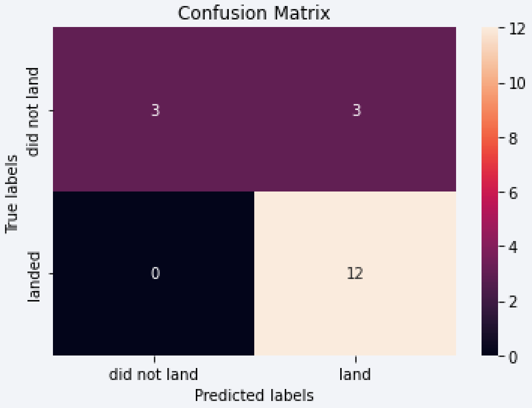
Precision: TP/Predicted Yes= 12/15 = 0.8

Prevalence: Actual yes/Total= 12/18 = 0.6667

Logistic Regression



SVM



Decision Tree

KNN

Classification Accuracy

- The **Decision tree** classifier is the model with the highest classification accuracy of 88.9% on training data.
- We trained four different models which each had an 83% accuracy rate on test data.

Find the method performs best:

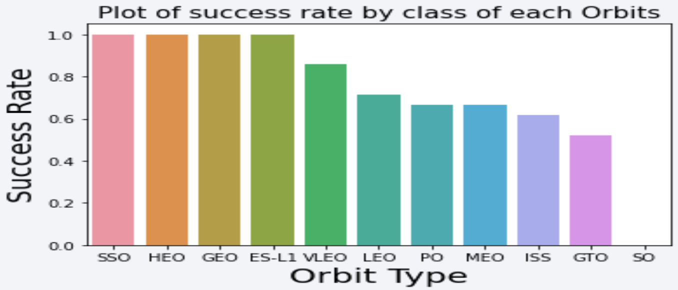
In [30]:
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_,
              'SVM': svm_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
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_)
if bestalgorithm == 'SVM':
    print('Best Params is :',svm_cv.best_params_)
```

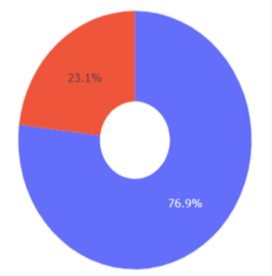
```
Best Algorithm is Tree with a score of 0.8892857142857145
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2,
'splitter': 'random'}
```

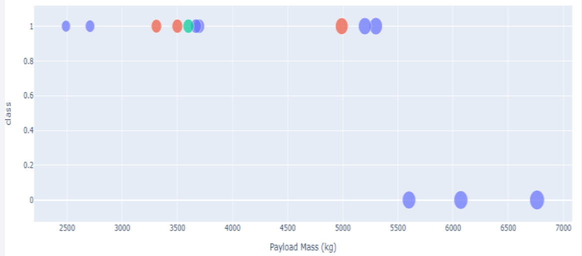
Conclusions

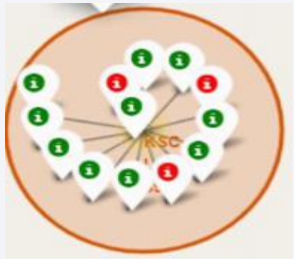
We can conclude that:

- Orbits ES-LI, GEO, HEO, SSO has highest Success rates from which SSO orbit have the most success rate; 100% and more than 1 occurrence
- 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 had the most successful launches of any sites;76.9% but increasing payload mass seems to have negative impact on success
- The Decision tree classifier is the best machine learning algorithm for this dataset.









Thank you!