IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Saeed Abderhim Ahemad

JULY 1-2022

بسم الله الرحمن الرحيم

The Contents

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

❖Summary Of Mothodoliges

- Summary of methodology
- Data collection
- Data wrangling
- EDA with SQL
- EDA with Data visualization
- Building interactive map with folium
- Belding Board with Poltly Dash
- Predictive analysis(classification)

❖Summary Of All Result

- As a result, SpaceY will be able to make more informed bids against SpaceX by using 1st stage landing predictions as a proxy for the cost of a launch.

❖Project background and context

- SpaceY is a new commercial rocket launch provider who wants to bid against SpaceX.
- SpaceX advertises launch services starting at $62 million for missions that allow some fuel to be reserved for landing the 1st stage rocket booster, so that it can be reused.
- SpaceX public statements indicate a 1st stage Falcon 9 booster to cost upwards of $15 million to build without including R&D cost recoupment or profit margin.
- Given mission parameters such as payload mass and desired orbit, the models produced in this report were able to predict the first stage rocket booster landing successfully with an accuracy level of 83.3%.

❖ Problems you want to find answers

- Determine the price of each launch. Also determining whether SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully

Section 1

Methodology

Methodology

Executive Summary

❑ **Data collection methodology**:

- **SpaceX Rest API**
- **Web Scraping from Wikipedia**

❑ **Perform data wrangling** :

- **Transforming data for machine Learning (One Hot Encoding data for machine Learning and dropping irrelevant columns**

❑ **Perform exploratory data analysis (EDA) using visualization and SQL :**

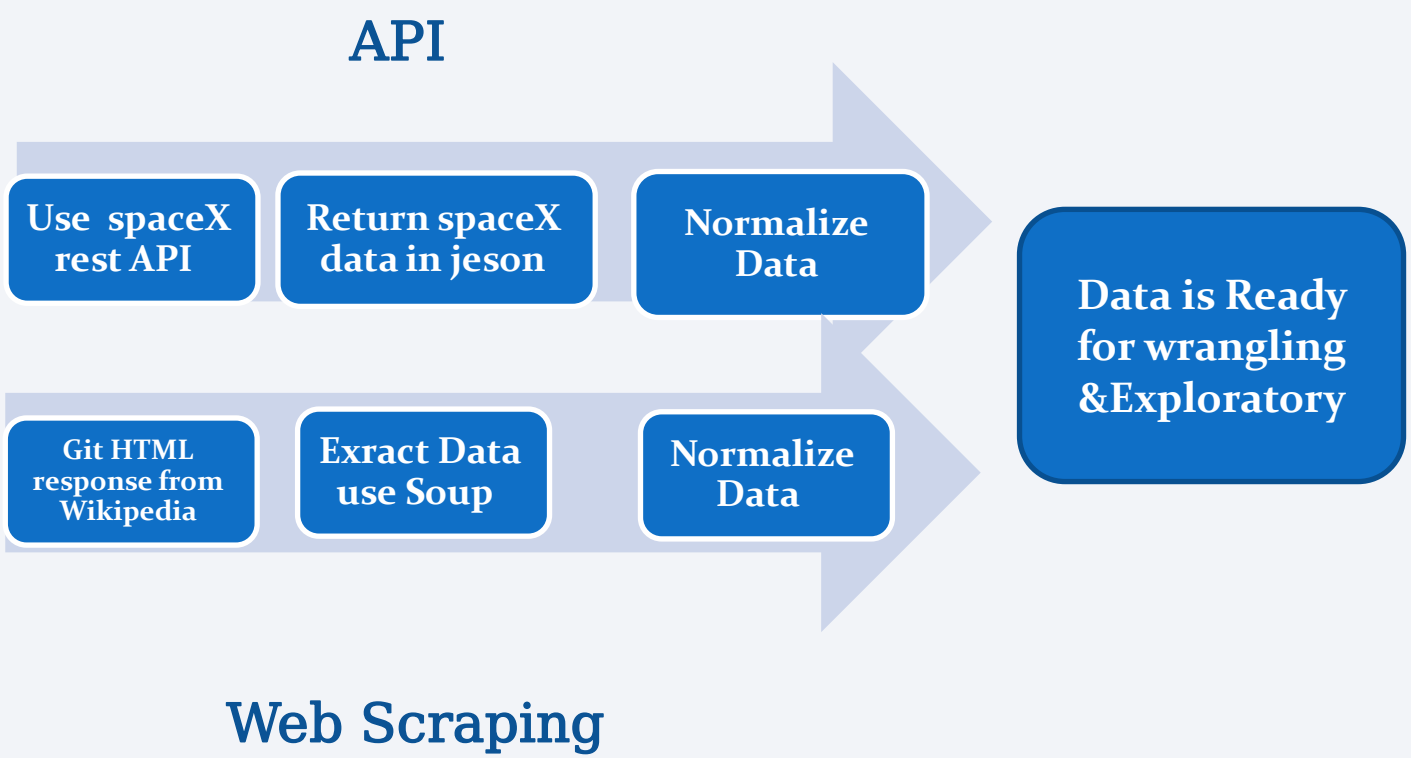
- **Plotting scatter Graph& Bar graph to show relationship between variables to show patterns of data**

❑ **Perform interactive visual analytics using Folium and Plotly Dash**

❑ **Perform predictive analysis using classification models :**

- **create a column for the class**
- **Standardize the data**
- **Split into training data and test data**
- **Find best Hyperparameter for SVM, Classification Trees and Logistic Regression**
- **Find the method performs best using test data**

Data Collection



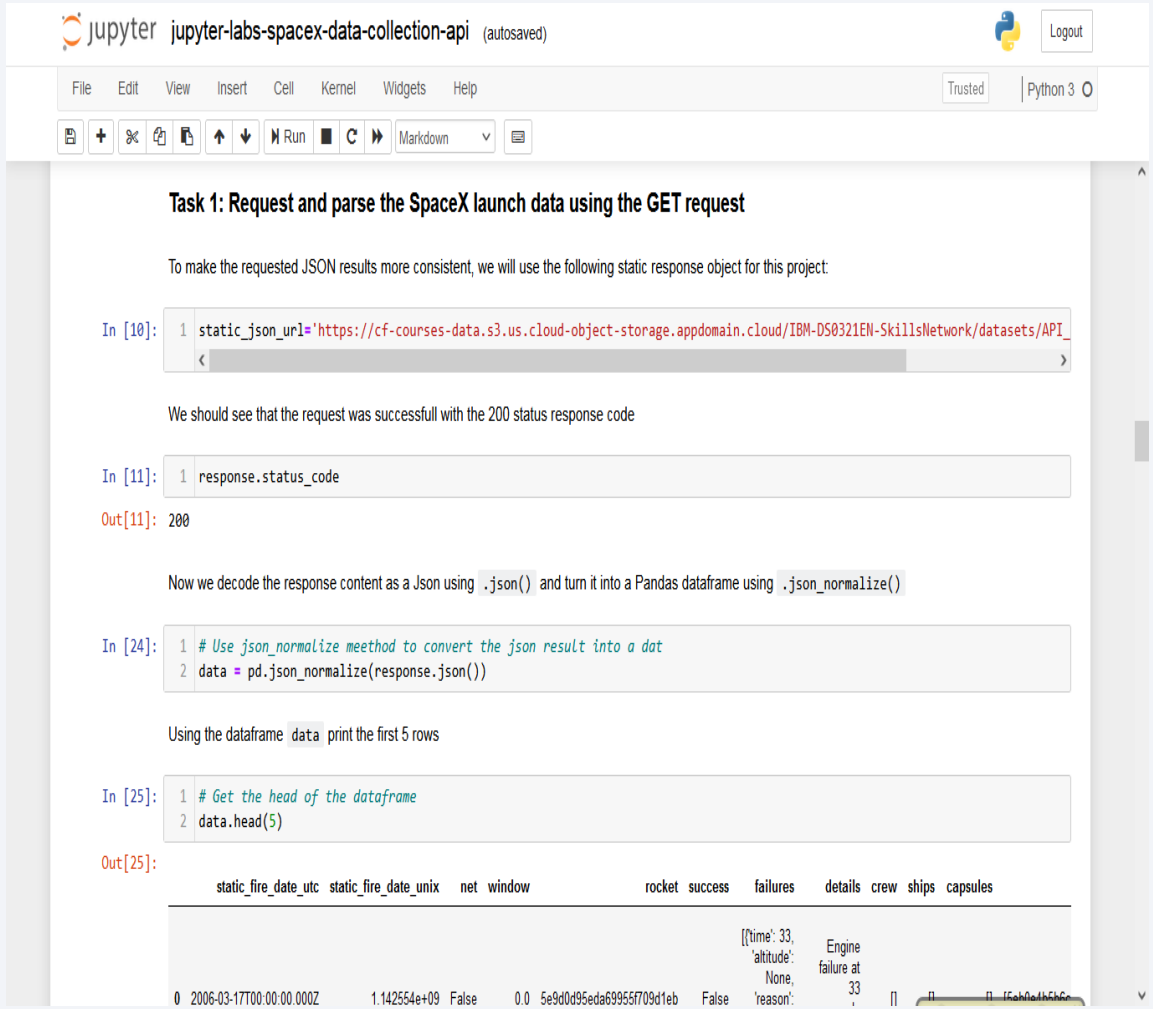
- The Following data **was collected** :
  - spaceX launch data that is gathered from the spaceX REST API
  - The spaceX REST API endpoint or URL ,start with api .spacexdata.com/v4
  - Another data source for obtaining falcon 0 launch data web scraping Wikipedia using BeautiulSoup

Data Collection – SpaceX API

- **Data Collection With SpaceX REST calls**

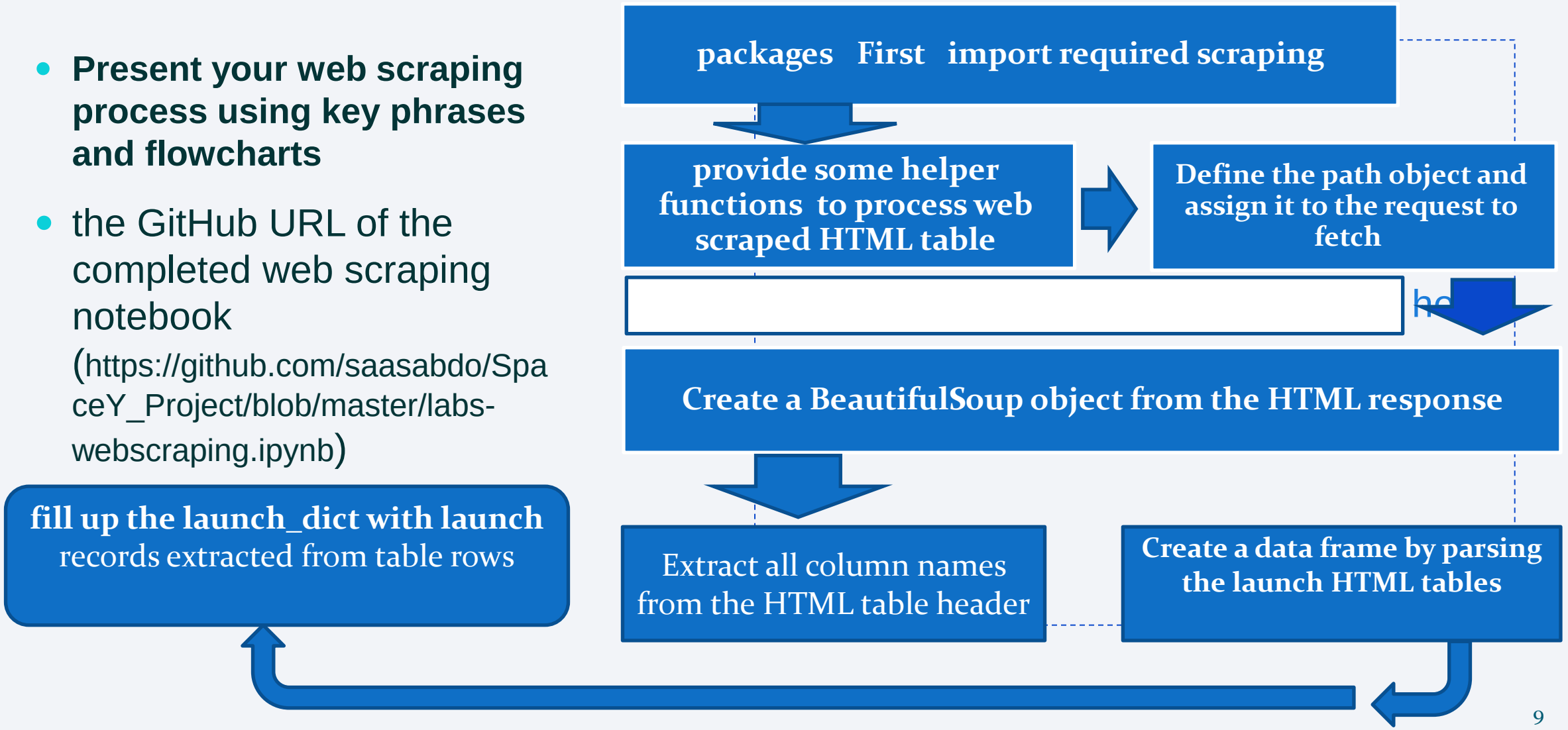
- GitHub URL of the completed SpaceX API calls notebook

https://github.com/saasabdo/SpaceY_Project/blob/master/spacex-data-collection.ipynb

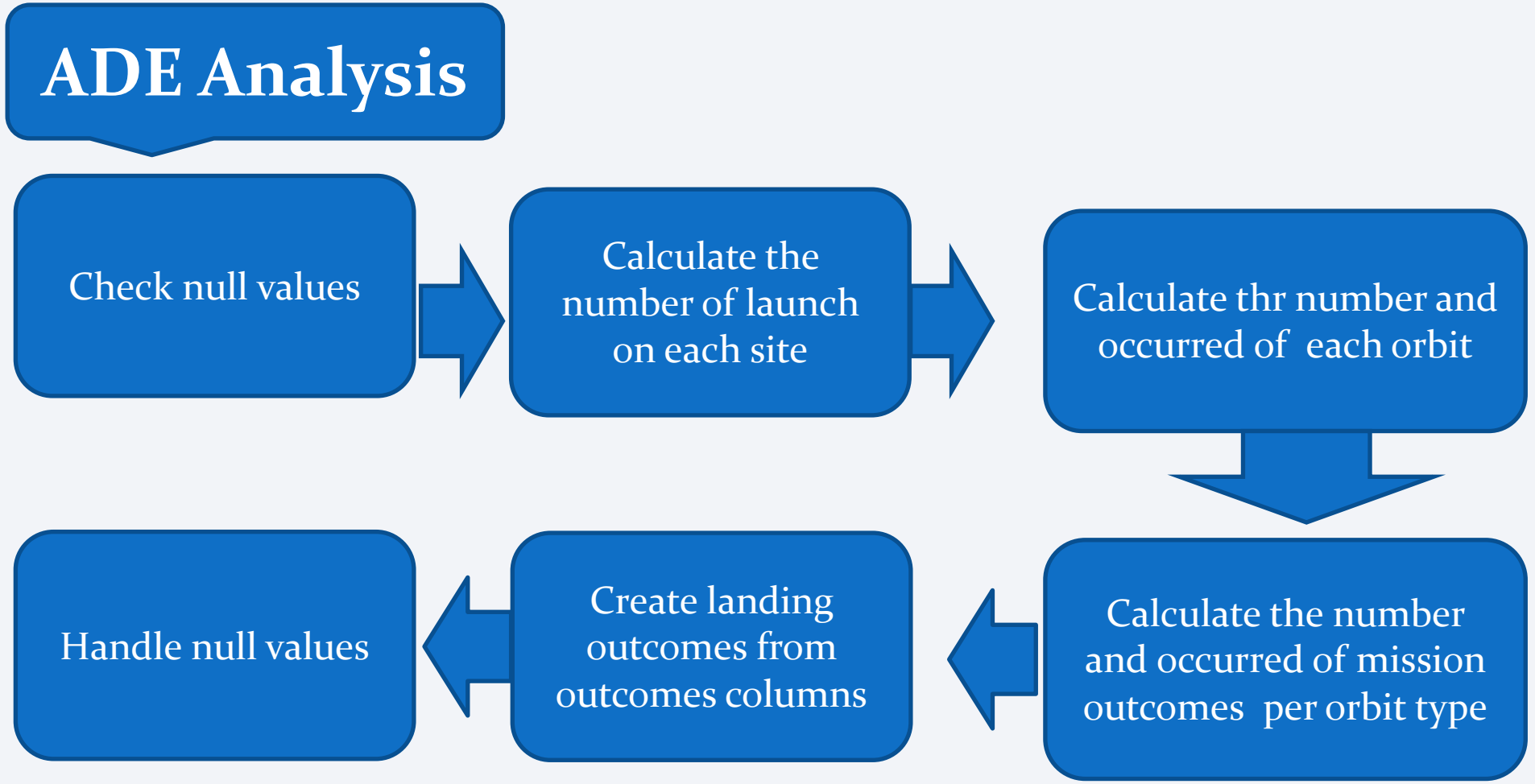


Data Collection - Scraping

- **Present your web scraping process using key phrases and flowcharts**
- the GitHub URL of the completed web scraping notebook (https://github.com/saasabdo/SpaceY_Project/blob/master/labs-webscraping.ipynb)



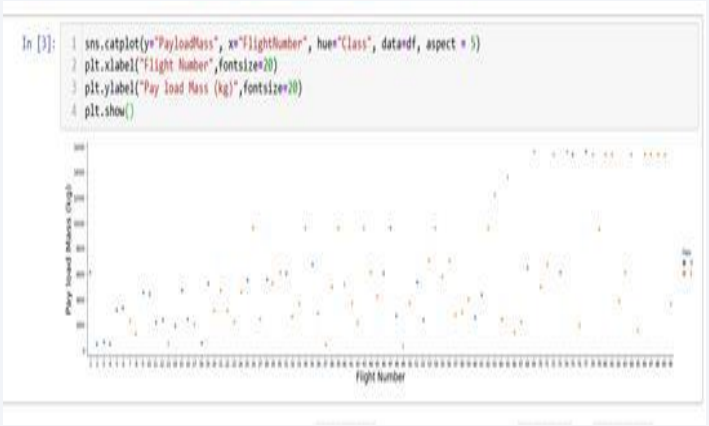
Data Wrangling

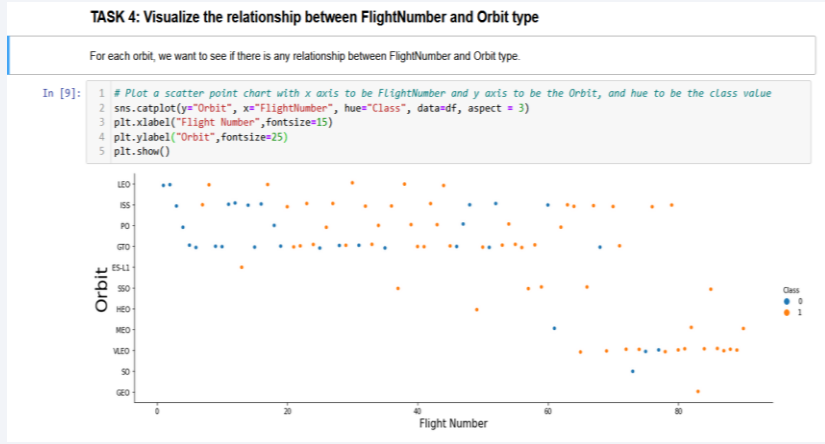


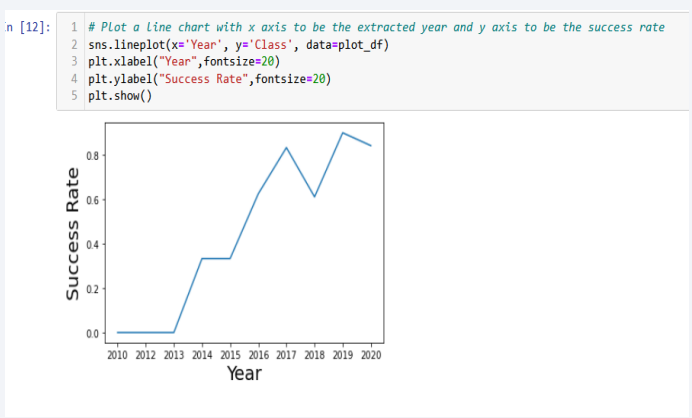
https://github.com/saasabdo/SpaceY_Project/blob/master/Space%20X%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

EDA with Data Visualization

- We will try to use the data to determine if the second stage of the Falcon 9 is automatic and use some instructions to determine if the first stage can be reused. Successfully!







https://github.com/saasabdo/SpaceY_Project/blob/master/EDA%20with%20Visualization%20.ipynb

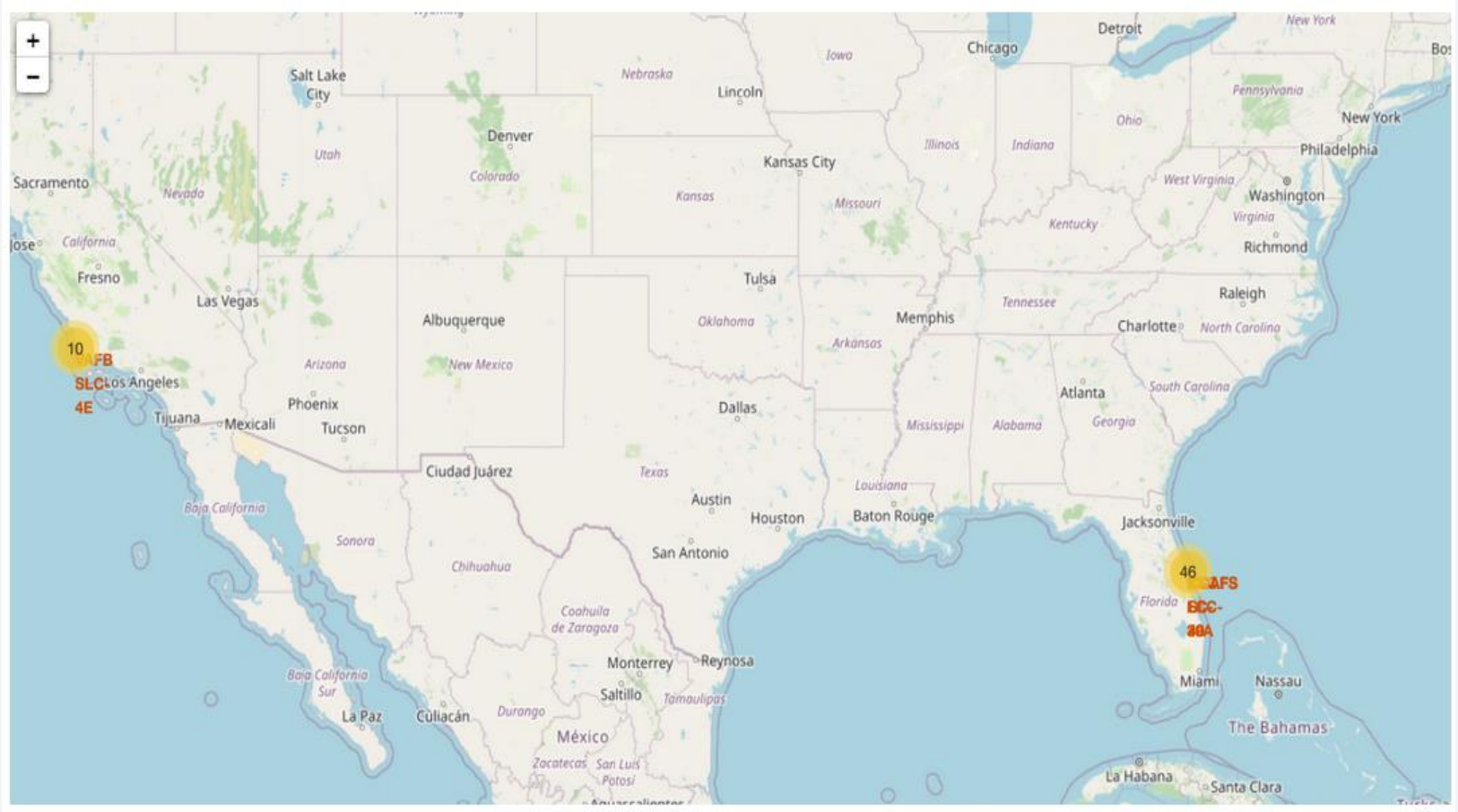
EDA with SQL

- SQL queries performed include:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'KSC'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
  - Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster
  - versions, launch_site for the months in year 2017
  - Ranking the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descendingorder.

https://github.com/saasabdo/SpaceY_Project/blob/master/Exploratory%20Data%20Analysis%20with%20SQL%2011.ipynb

Build an Interactive Map with Folium

- Map Marker have been added to the map with aim finding an optimal location for the building a launch site



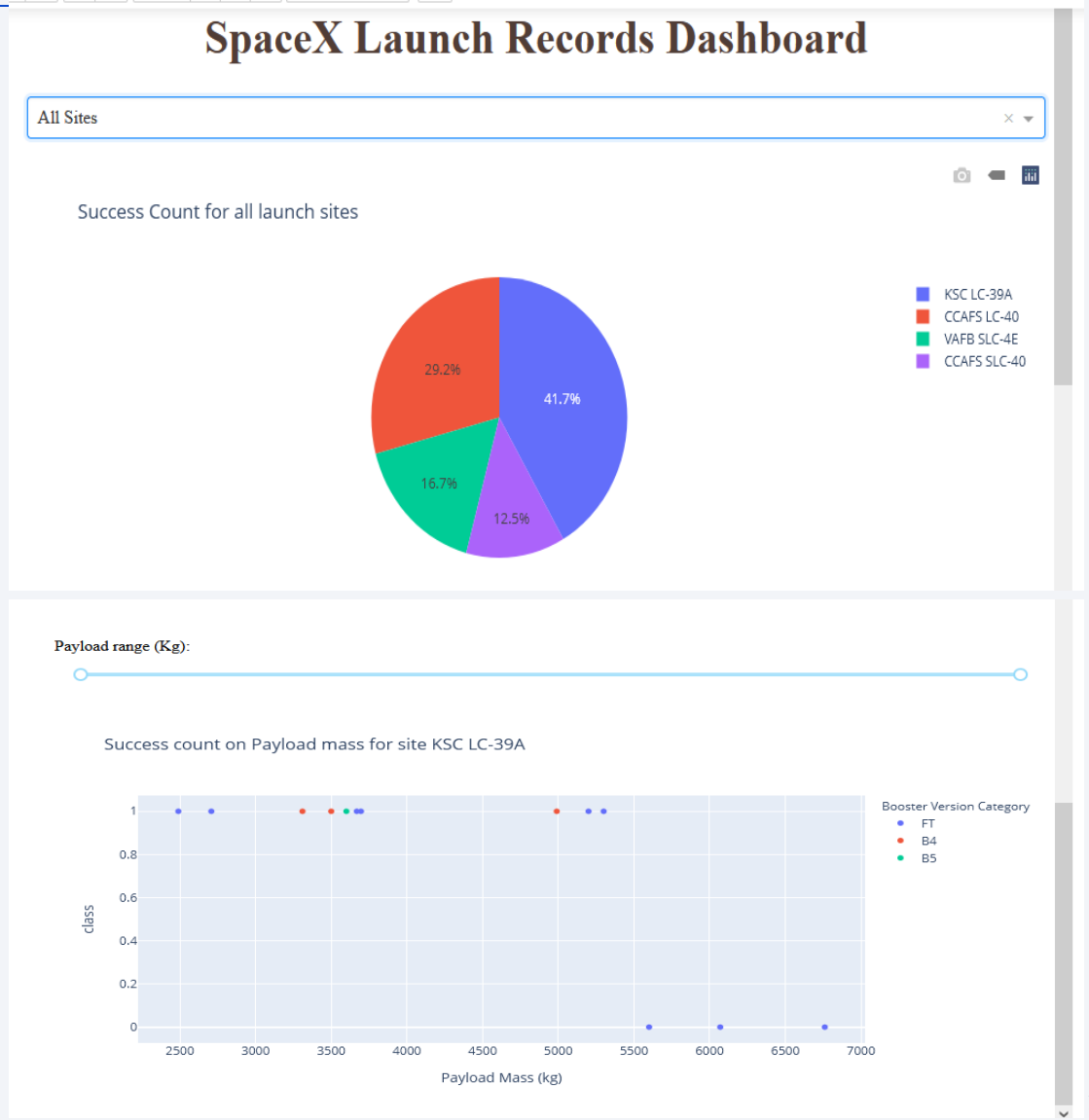
https://github.com/saasabdo/SpaceY_Project/blob/master/launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Interactive Visual Analytics enables users to explore and manipulate data in an interactive, real-time and real-time manner:
  - This can be used for common interactions including zooming in and out, panning, filtering, searching and linking.
  - Finding visual patterns faster and more effectively.
  - Add vigor rather than presenting results in static graphs,

- **GitHub URL of your completed Plotly Dash:**

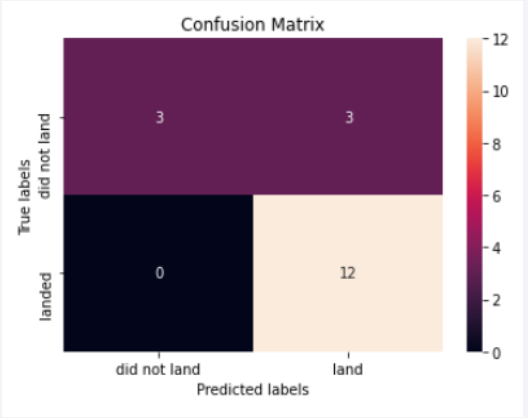
  - SpaceY_Project/SpaceX Launch Records Dashboard.ipynb at master · saasabdo/SpaceY_Project (github.com)

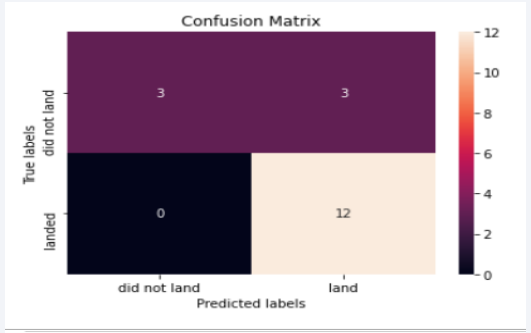


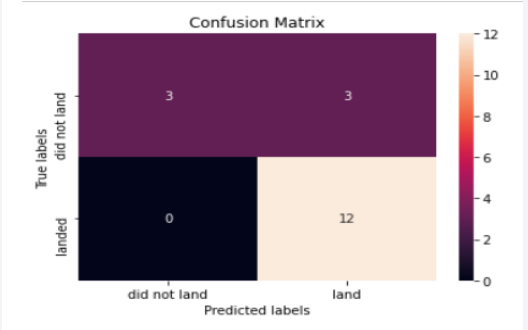
Predictive Analysis (Classification)

❖ Predictive Analysis(Model development)

- ❑ **Imported libraries and defined function to create confusion matrix**
  - •Pandas •Numpy •Matplotlib •Seaborn •Sklearn
- ❑ Loaded the dataframecreated during data collection
- ❑ Created a column for our training label 'Class' created during data wrangling
- ❑ Standardized the data
- ❑ Split the data into training data and test data
- ❑ Fit the training data to various model types
  - •Logistic Regression •Support Vector Machine •Decision Tree Classifier
  - •K Nearest Neighbors Classifier
- ❑ Used a cross-validated grid-search over a variety of hyperparameters to select the best ones for each model
- ❑ Enabled by Scikit-learn library function GridSearchCV
- ❑ Evaluated accuracy of each model using test data to select the best model







Results

- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy 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.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

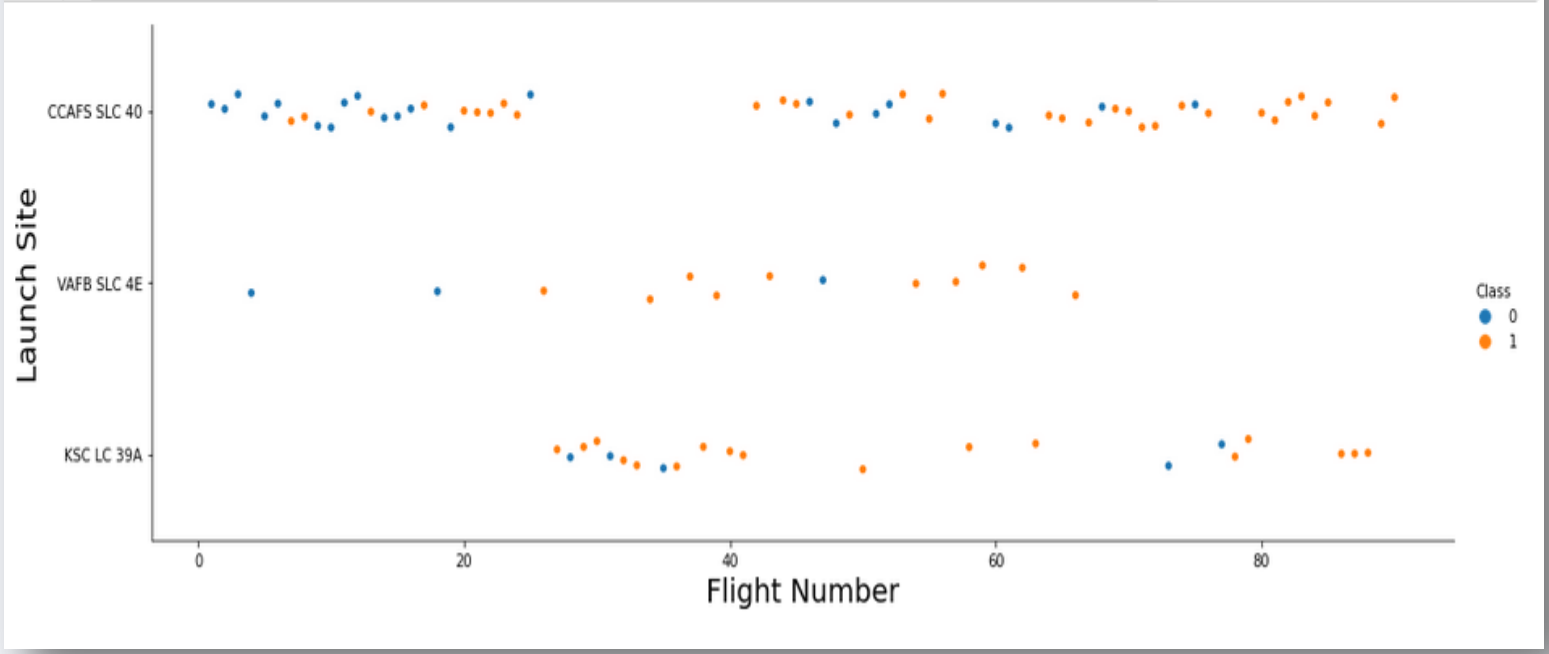
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

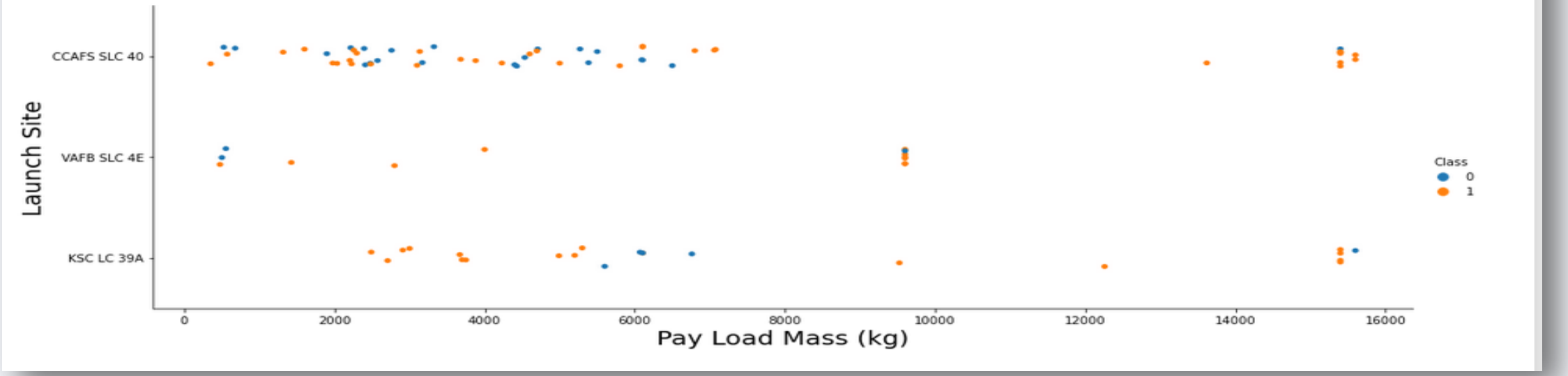
- Launches from the site of CC AFS SLC40 are significantly higher than launches from anther site

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 3)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

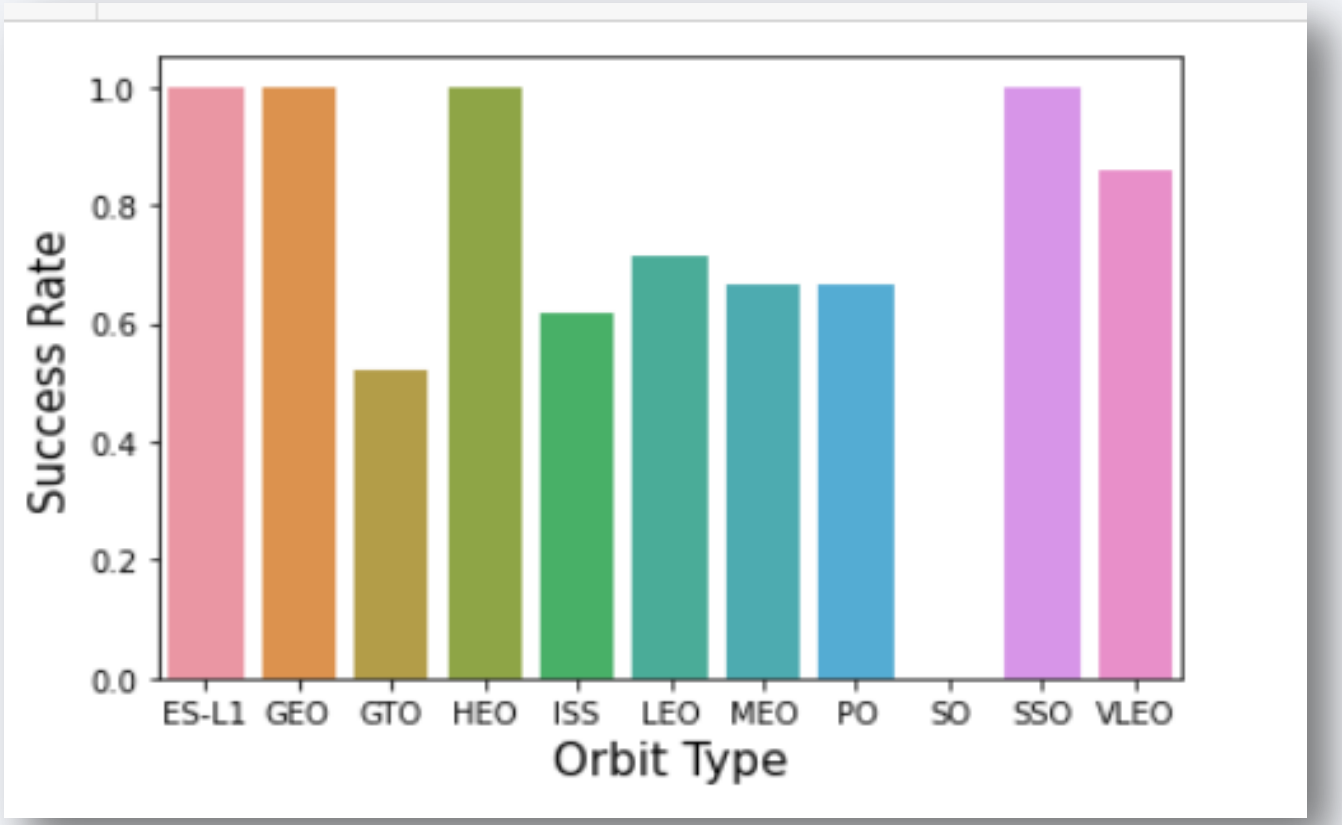
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 3)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



The majority of payloads with lower have been launched from CCAFS SLC40

Success Rate vs. Orbit Type

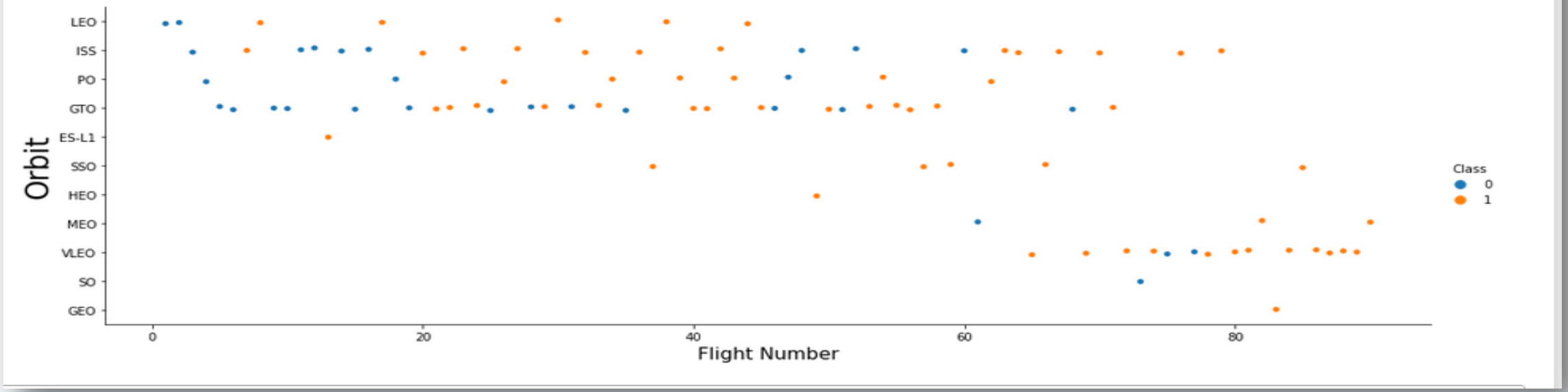
- The Orbit types of ES-L1 GEO, HEO,SSO are among the highest success



.We note that in LEO, success appears to be related to the number of flights; On the other hand, there s

Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 3)
plt.xlabel("Flight Number",fontsize=15)
plt.ylabel("Orbit",fontsize=25)
plt.show()
```

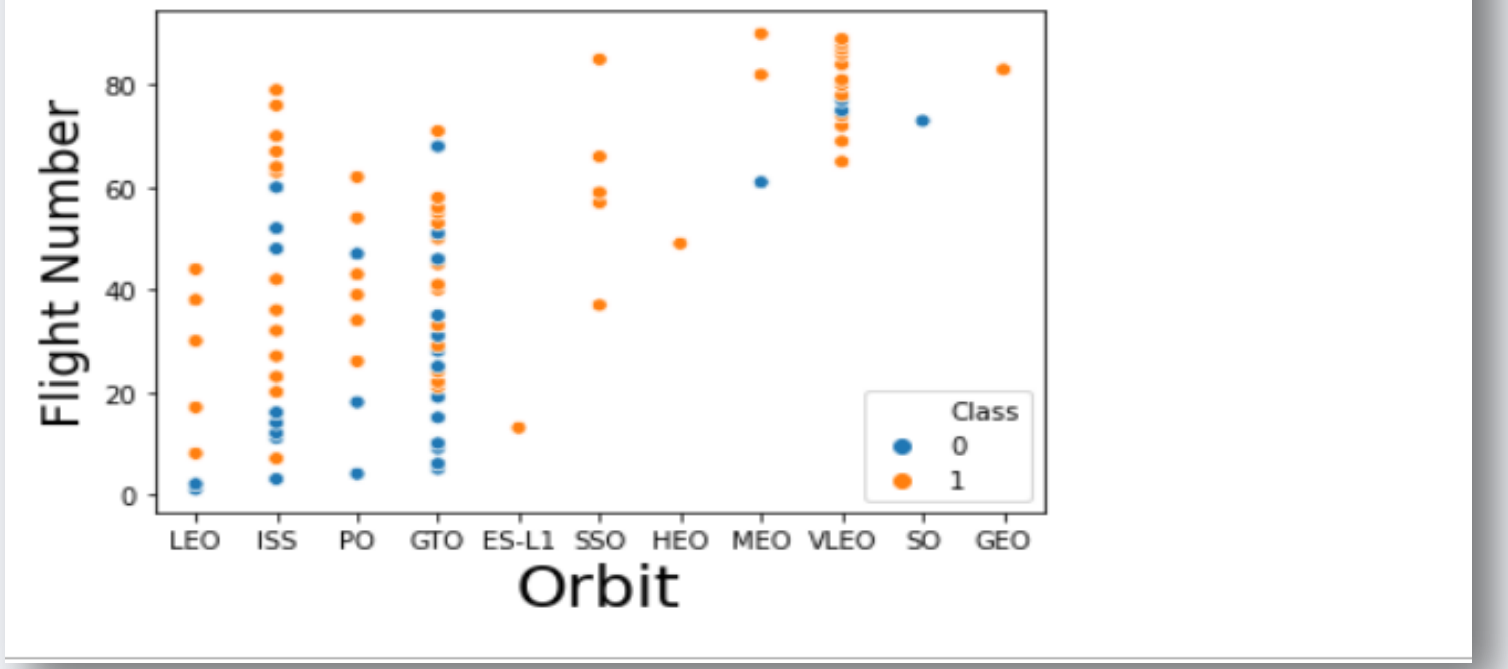


- We note that in LEO, success appears to be related to the number of flights; On the other hand, there seems to be no relationship between the flight number when you are in GTO orbit.

Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS

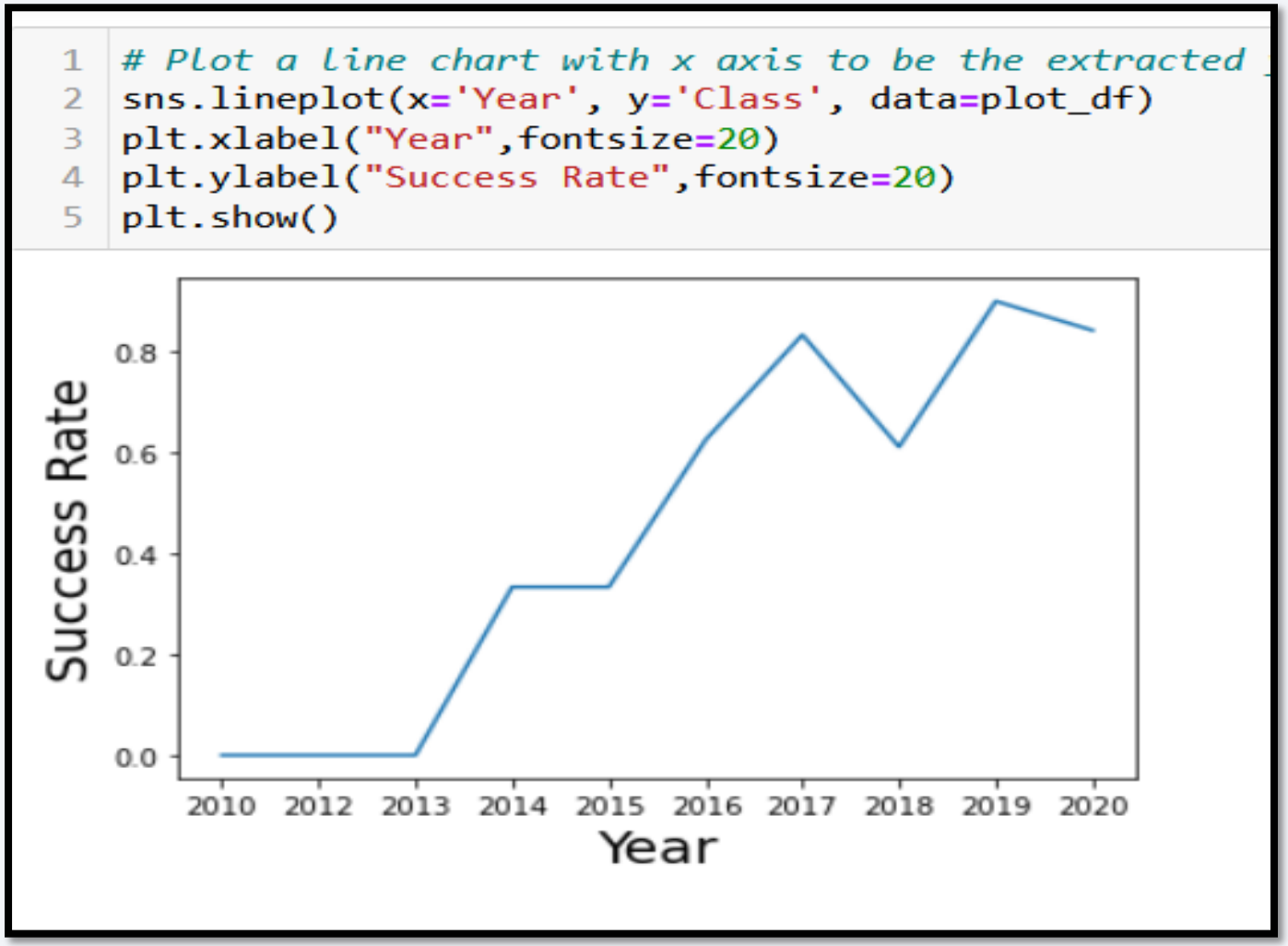
```
# Plot a scatter point chart with x axis to be Payload and y axis t
sns.scatterplot(x="Orbit",y="FlightNumber",hue="Class",data = df)
plt.xlabel("Orbit",fontsize=25)
plt.ylabel("Flight Number",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- you can observe that the sucess rate since 2013 kept increasing till 2020

```
# Plot a line chart with x axis to be the extracted
sns.lineplot(x='Year', y='Class', data=plot_df)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

- We use the key word DISTINCT to show unique launch sites

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

In [7]:
```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXDATASET
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

```
In [8]:  1  %sql SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.
```

Out[8]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	lan
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fa
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	Fa
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	

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

Total Payload Mass

```
In [9]:  1 %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE CUSTOMER='NASA (CRS)'

 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.

Out[9]:
1
45596
```

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

Average Payload Mass by F9 v1.1

```
In [10]:  1 %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE BOOSTER_VERSION='F9 v1.1'

 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.

Out[10]:
1
2928
```

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

First Successful Ground Landing Date

```
In [11]:  1 %sql SELECT min(DATE) FROM SPACEXDATASET WHERE LANDING__OUTCOME='Success (ground pad)'

 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.

Out[11]:
1
2015-12-22
```

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

Successful Drone Ship Landing with Payload between 4000 and 6000

In [13]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND L
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdo
main.cloud:31498/BLUDB
Done.
```

Out[13]:

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

- Use WHERE to 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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [14]:
```
1 %sql SELECT COUNT(*) FROM SPACEXDATASET WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIK
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdo
main.cloud:31498/BLUDB
Done.
```

Out[14]:

1
101

- List the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

- determined the booster that have carried the maximum payload using a sub query in the WHERE clause and the MAX() function

```
In [16]:  1 %sql SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS_
           * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdo
          main.cloud:31498/BLUDB
          Done.
```

Out[16]:

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

In [17]:

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.
```

Out[17]:

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

- The List records display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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

In [18]:
```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXDATASET \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

```
 * ibm_db_sa://zmt17363:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/BLUDB
Done.
```

Out[18]:

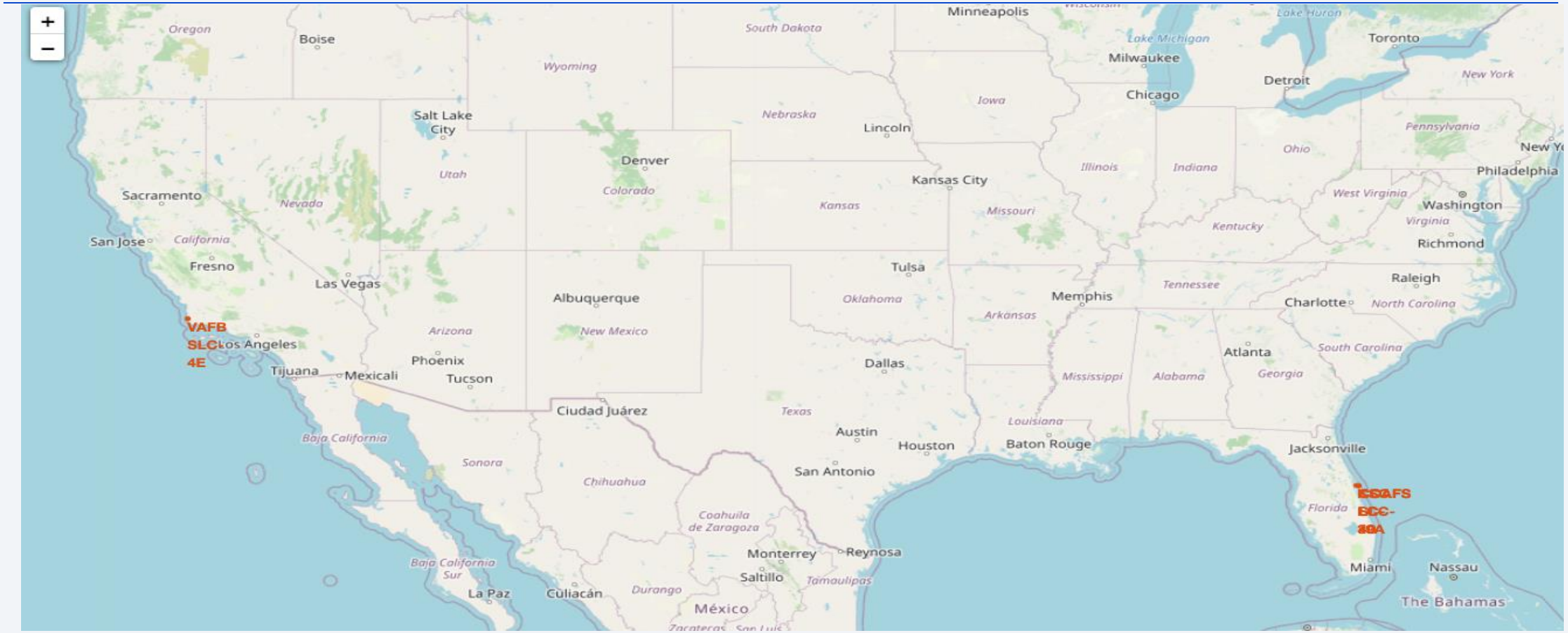
DATE	COUNT
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

- **Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.**

Section 3

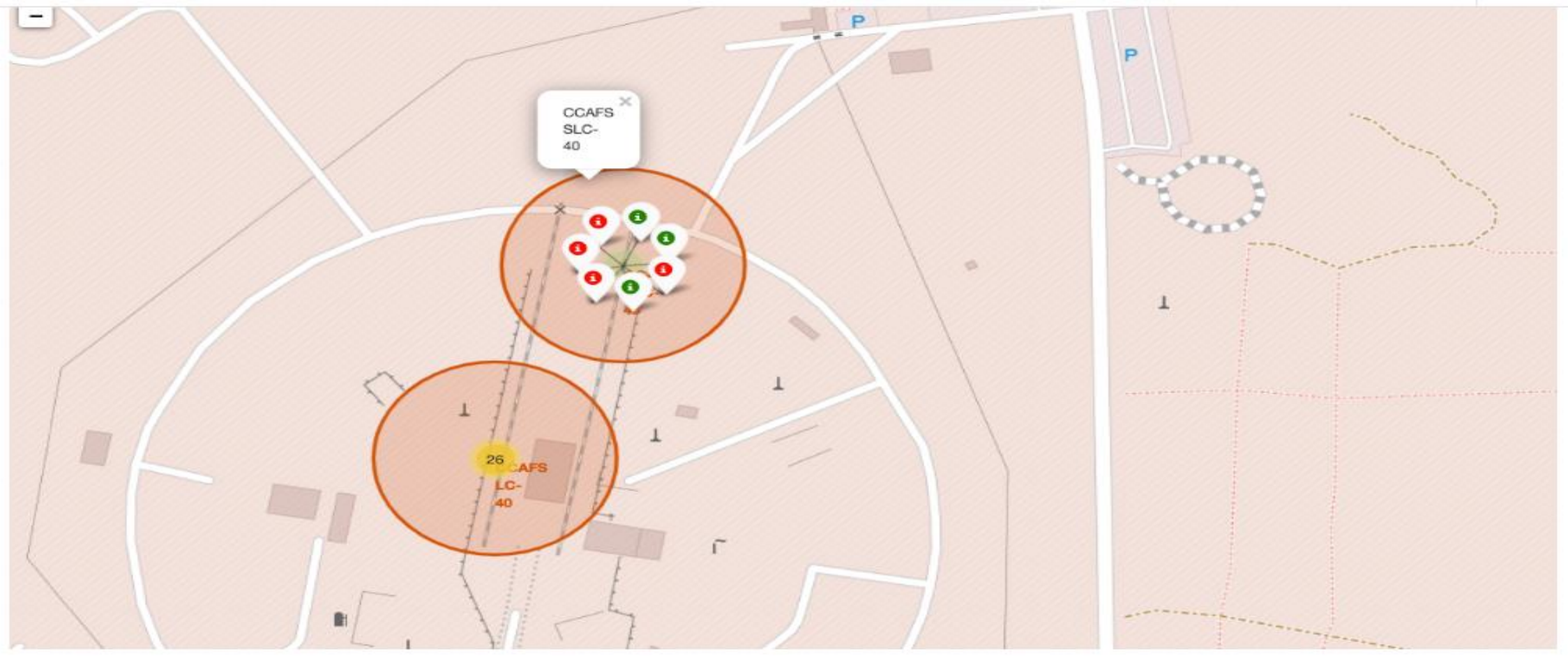
Launch Sites Proximities Analysis

Location of all the Launch Sites



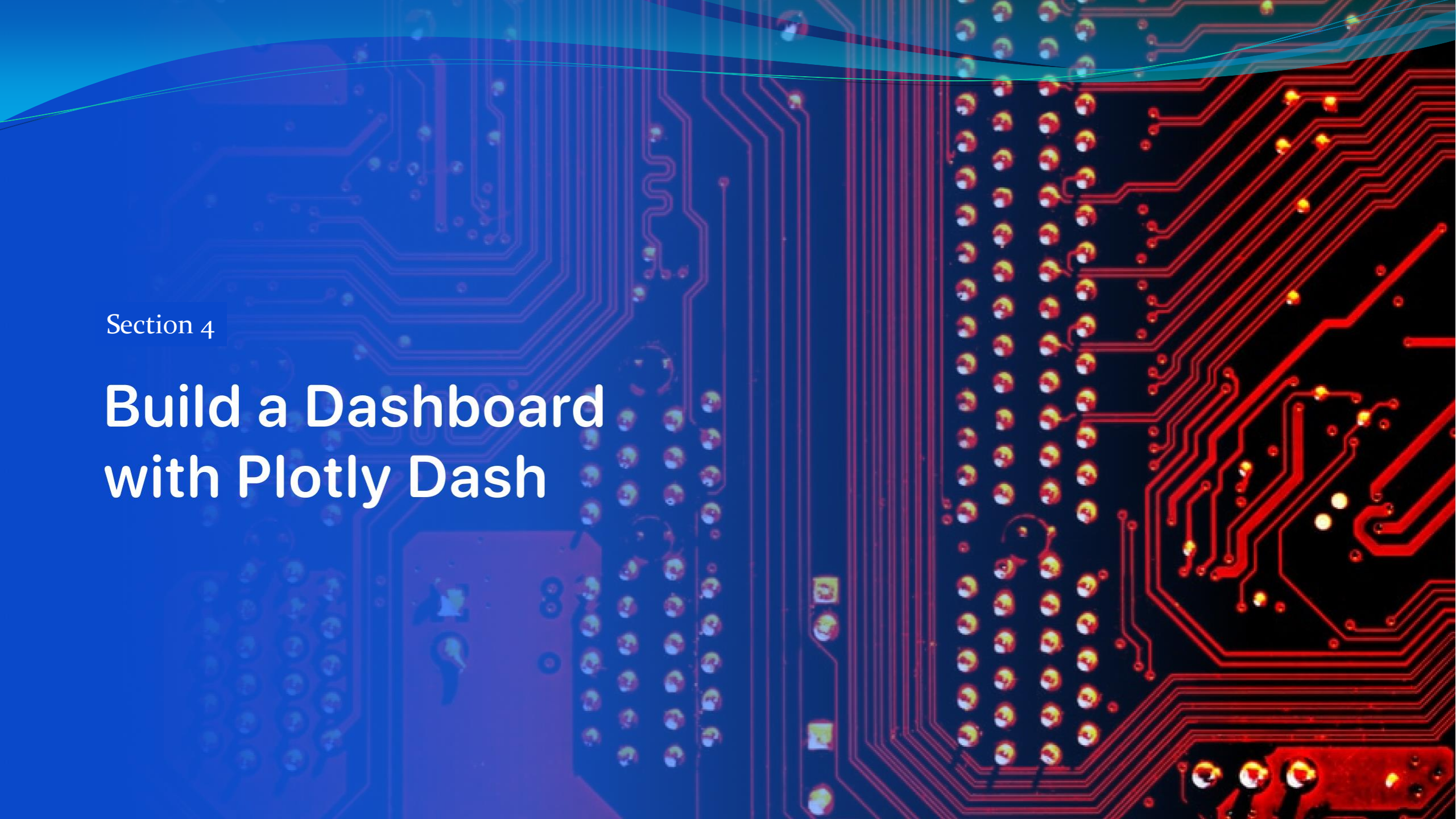
- SpaceX launch sites

Markers showing launch sites with color labels



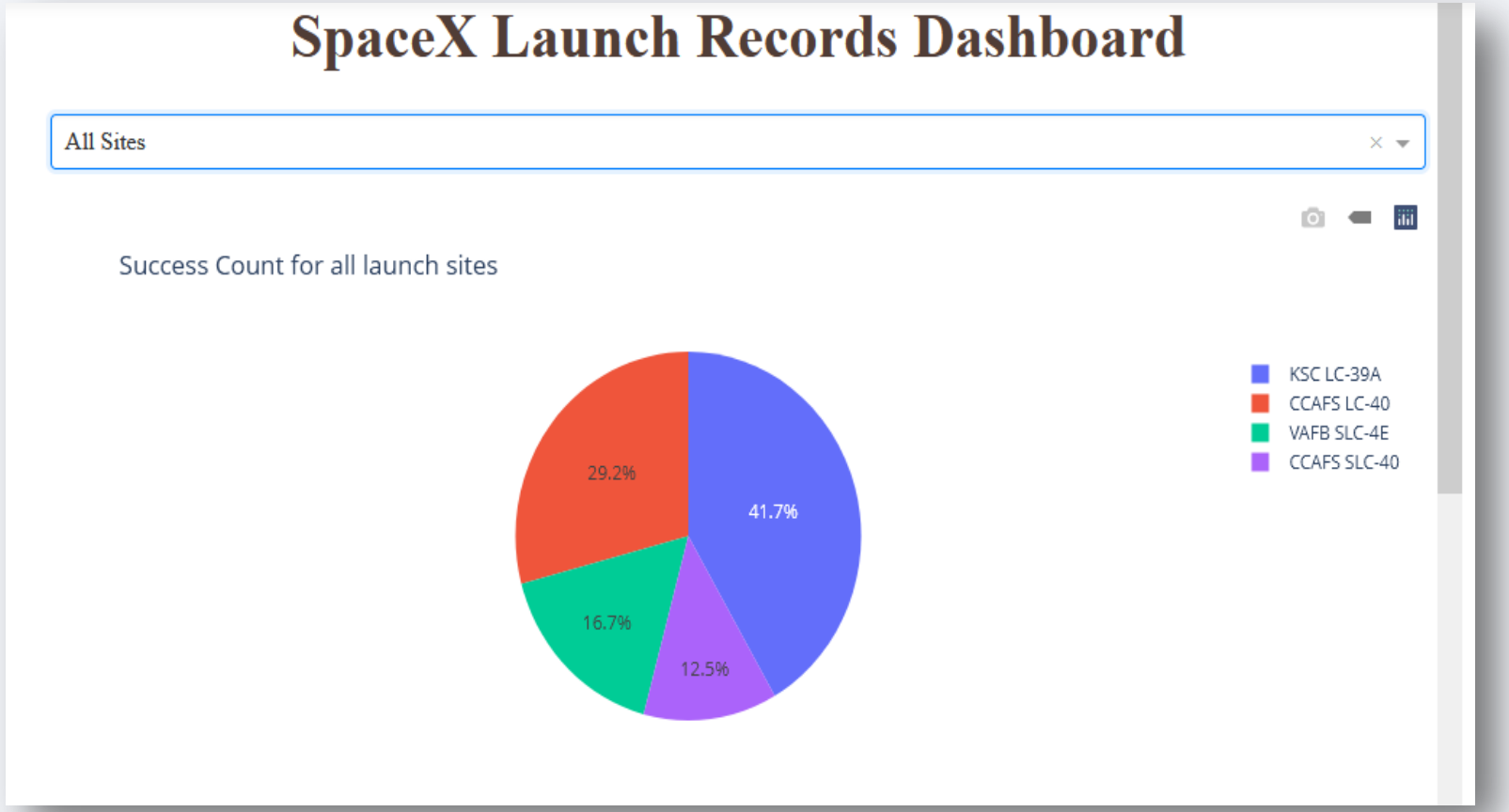
Launch Sites Distance to Landmarks

Out[20]:

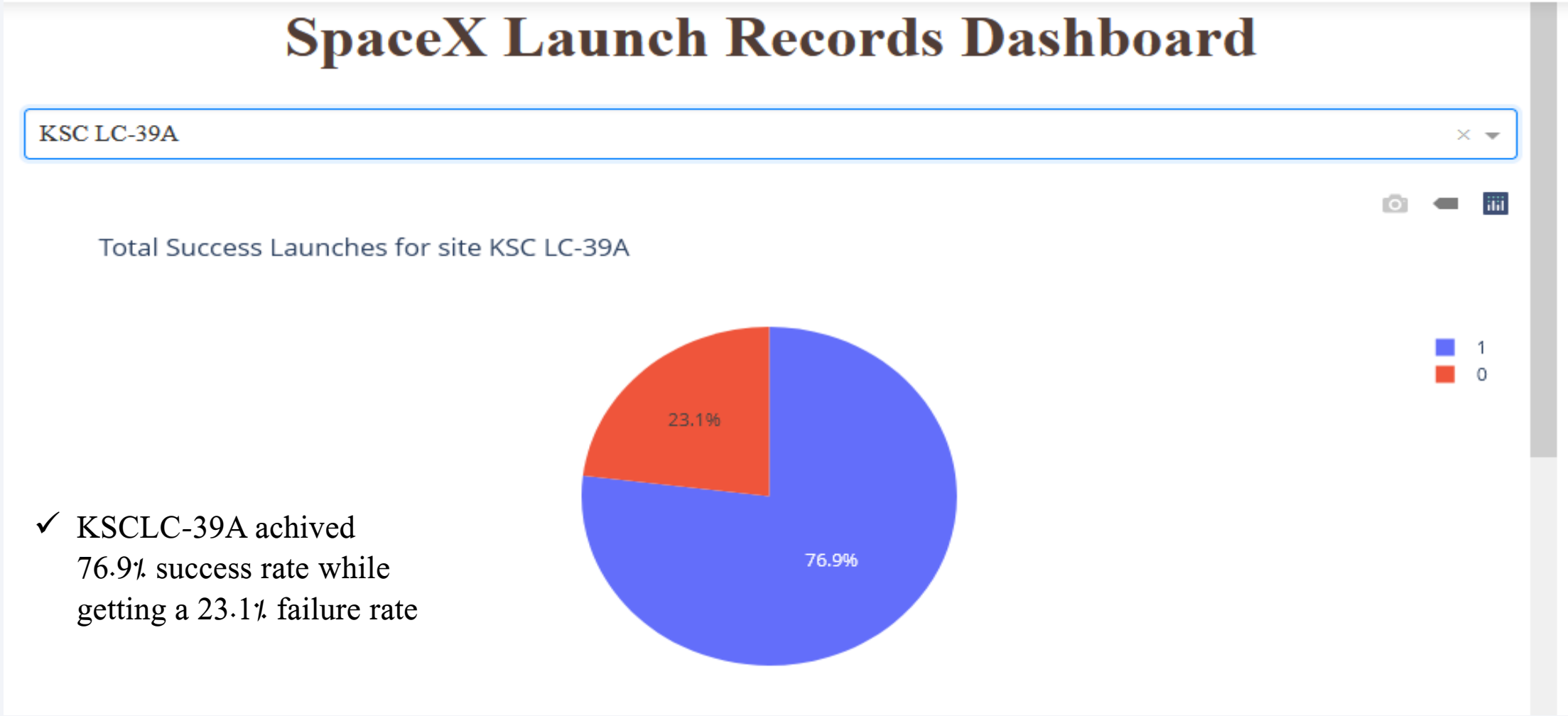

Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites

- ✓ We can see KSC LC-39A had a most successful launches from all the site and in second place CCAF LC-40

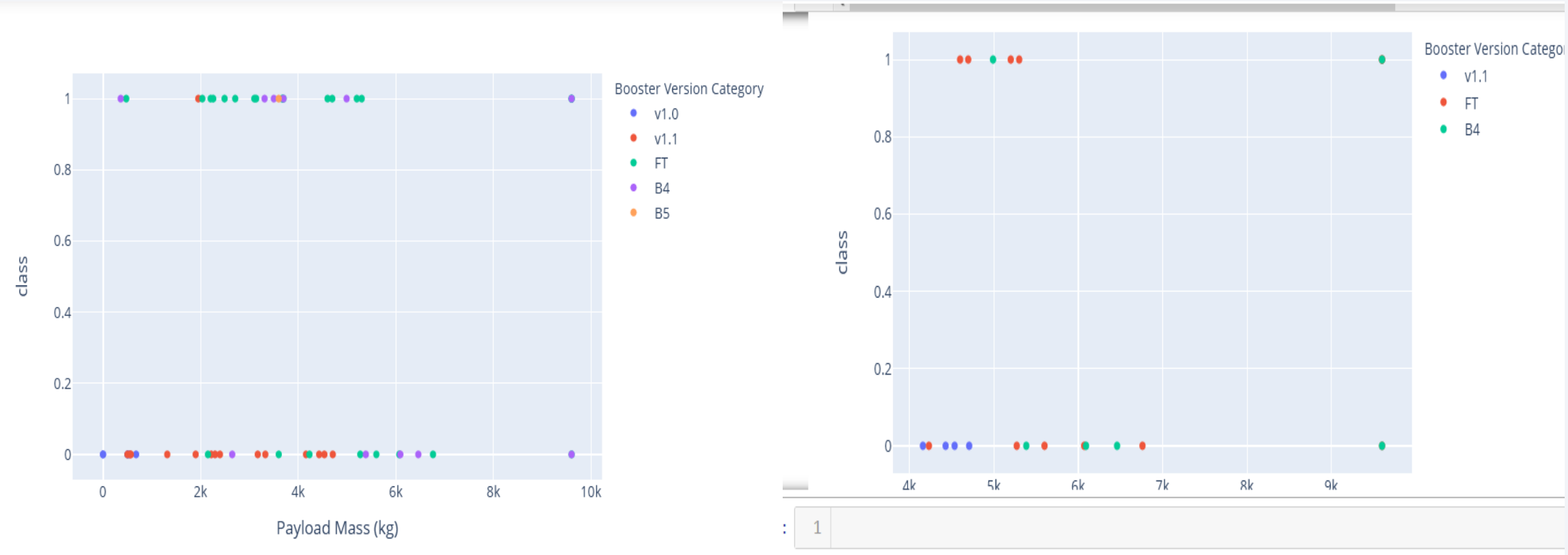


The highest launch-success ratio: KSC LC-39A



- ✓ KSCLC-39A achived 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot



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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

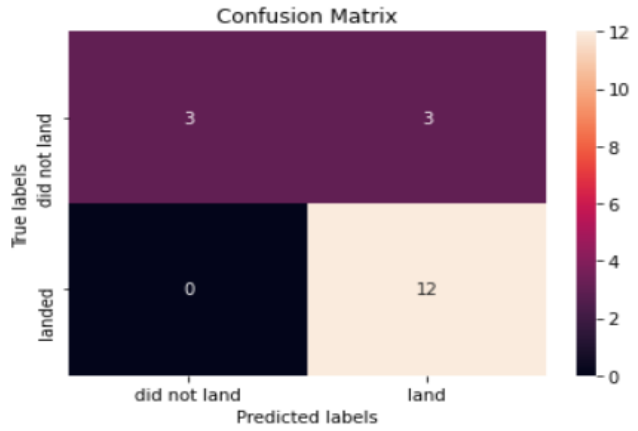
In [29]:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.b
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 0.8857142857142858
Best Params is : {'criterion': 'entropy', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf':
2, 'min_samples_split': 2, 'splitter': 'random'}
```

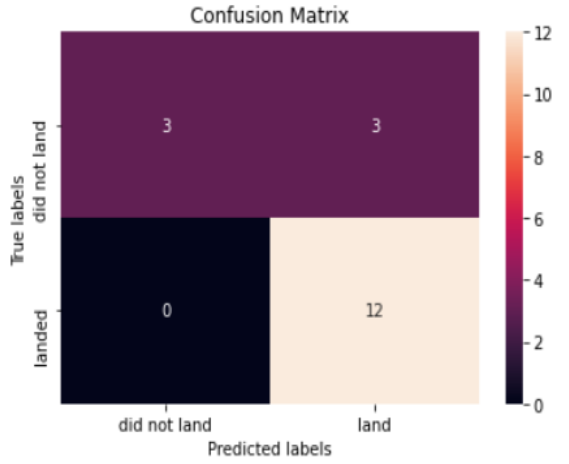
Confusion Matrix

```
1 yhat=svm_cv.predict(X_test)
2 plot_confusion_matrix(Y_test,yhat)
```

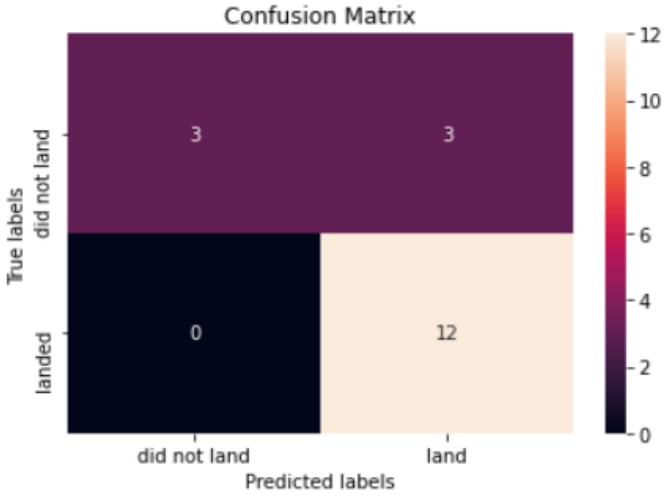


[28]:

```
1 yhat = knn_cv.predict(X_test)
2 plot_confusion_matrix(Y_test,yhat)
```

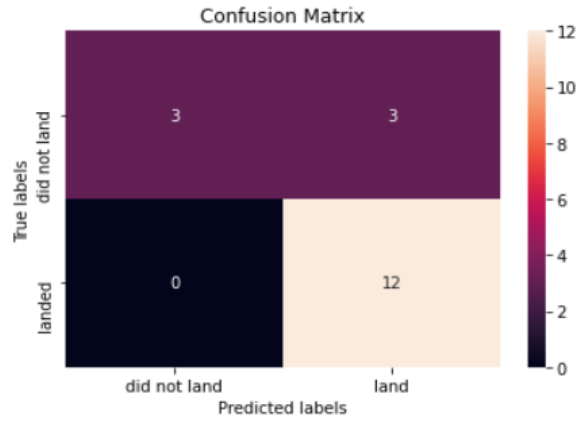



```
2 plot_confusion_matrix(Y_test,yhat)
```



In [13]:

```
1 yhat=logreg_cv.predict(X_test)
2 plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Using the models from this report SpaceY can predict when SpaceX will successfully land the 1st stage booster with 83.3% accuracy
- SpaceX public statements indicate the 1st stage booster costs upwards of $15 million to build
- This will enable SpaceY to make more informed bids against SpaceX, since they will have a good idea when to expect the SpaceX bid to include the cost of a sacrificed 1st stage booster
- With a list price of $62 million per launch, sacrificing the $15+ million 1st stage, would put the SpaceX bid at upwards of $77 million
- Biggest opportunities going forward to make even more informed bids:
  - Freeze the best performing combination of model and hyperparameters and re-fit using the whole
  - Potentially better than using only part of the data to fit the model, but you would no longer be able to measure the accuracyofthe resulting model
- Incorporate additional launch data to the dataset and model as it becomes available
- Subdivide the current model into two models
  - Predict if SpaceX will ATTEMPT to land the 1st stage
  - Predict if SpaceX will SUCCEED in their attempt
- Create a related model that predicts if SpaceX will launch using a previously-flown 1st stage booster
  - Would enable SpaceY to take into account when the SpaceX bid would likely include a discoun

Appendix

Reference Links

- Hands-on Lab : String Patterns, Sorting and Grouping
- Hands-on Lab: Built-in functions
- Hands-on Lab : Sub-queries and Nested SELECT Statements
- Hands-on Tutorial: Accessing Databases with SQL magic
- Hands-on Lab: Analyzing a real World Data Set

- https://aviationweek.com/defense-space/space/podcast-interview-spacexs-elon-musk
- Interview with Elon Musk where he discloses the 1st stage booster to cost upwards of $15 million
- https://datascience.stackexchange.com/a/33050
- Explanation of why you would rebuild your model using the full dataset
- https://www.spacex.com/vehicles/falcon-9/
- Source of SpaceX's advertised $62 million launch price

Thanks For Authors :
PhD Joseph Santarcangelo
Data Scientist at IBM Nayef Abou Tayoun
Lakshmi Holla
Yan Luo

Thank you!