IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

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

Introduction

- Project background and context
- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; 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 space X for a rocket launch. This goal of the 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:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

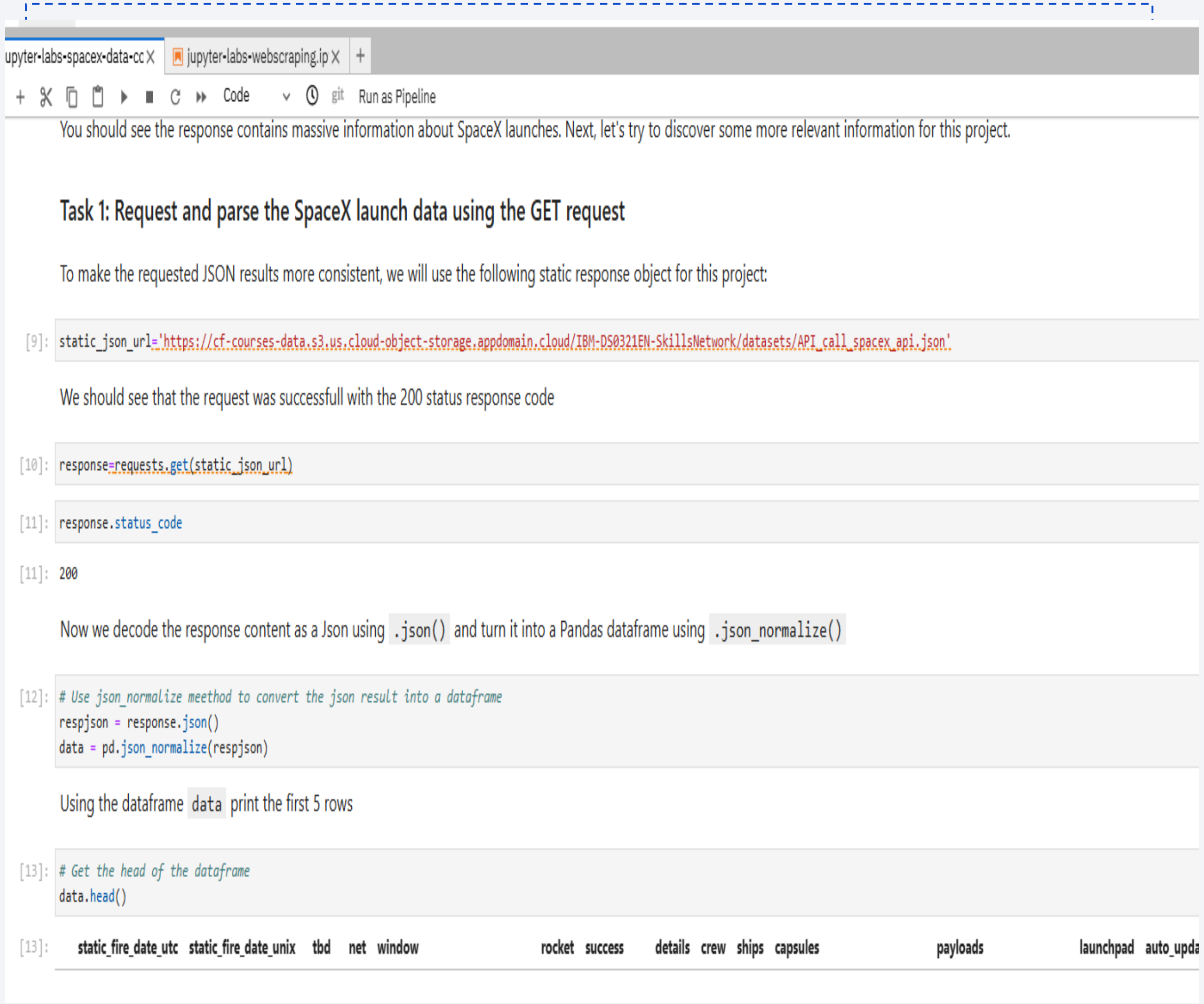
Data Collection

- Describe how data sets were collected.
- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

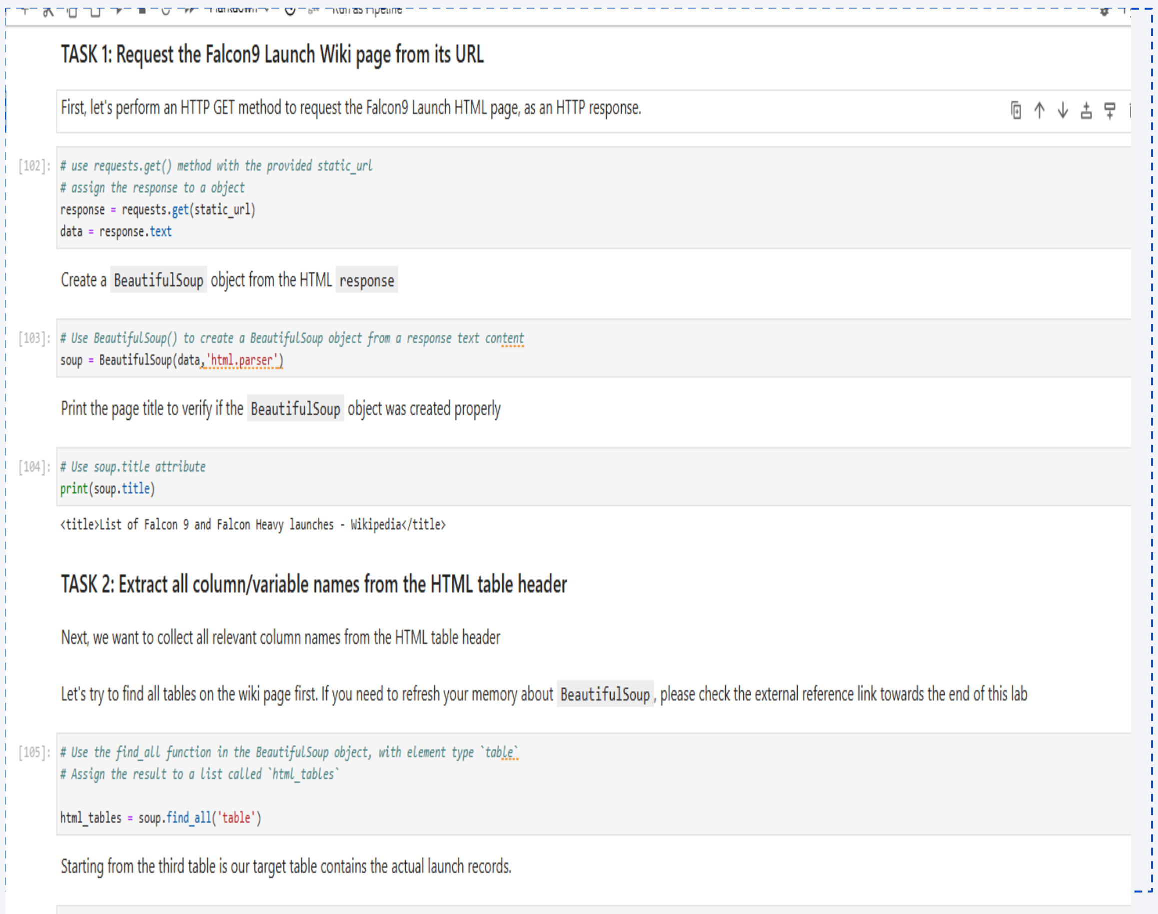
- The link to the notebook is
- https://github.com/taruna123lokesh/Data-Collection-SpaceX-API



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- The link to the notebook is
- https://github.com/taruna123lokesh/web-scrapping/blob/main/jupyter-labs-webscraping.ipynb

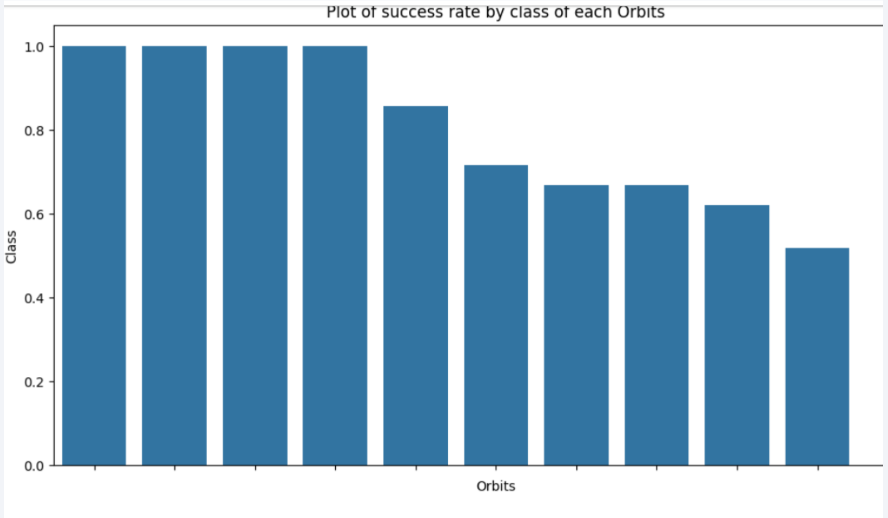


Data Wrangling

- Describe how data were processed
  - 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 and exported the results to csv.
- The link to the notebook is
- https://github.com/taruna123lokesh/web-scrapping/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.



The link to the notebook is https://github.com/taruna123lokesh/EDA-with-visualization/blob/main/edadataviz.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
- 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 total number of successful and failure mission outcomes
- The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is https://github.com/taruna123lokesh/EDA-with-sql/blob/main/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.

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is

https://github.com/taruna123lokesh/Build-a-Dashboard-with-plotly-Dash/blob/main/Dash_wildfire.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is

https://github.com/taruna123lokesh/predictive-analysis/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

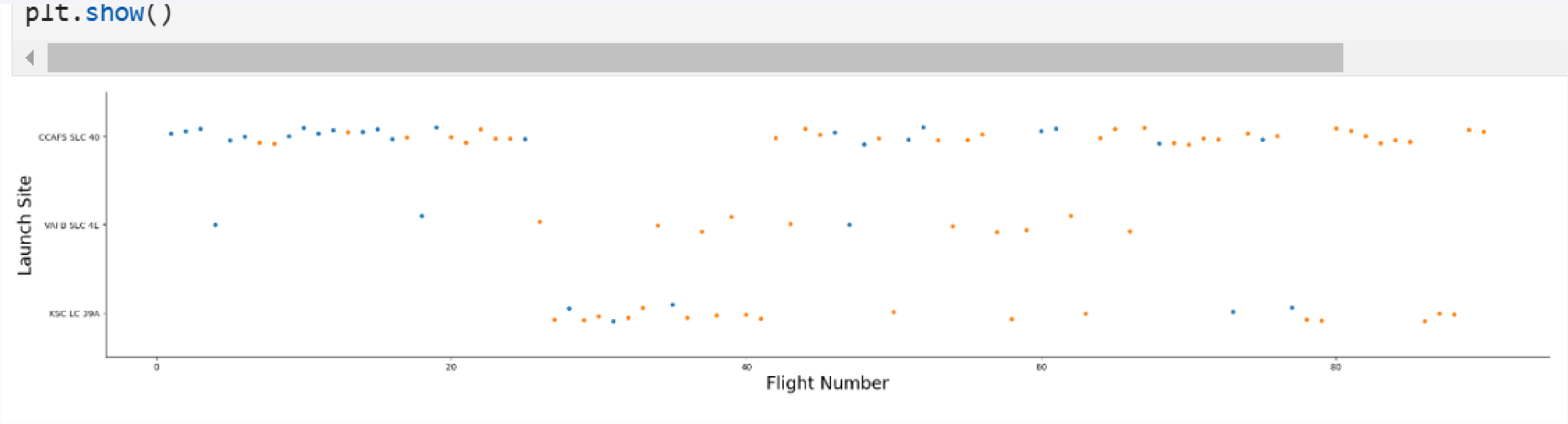
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

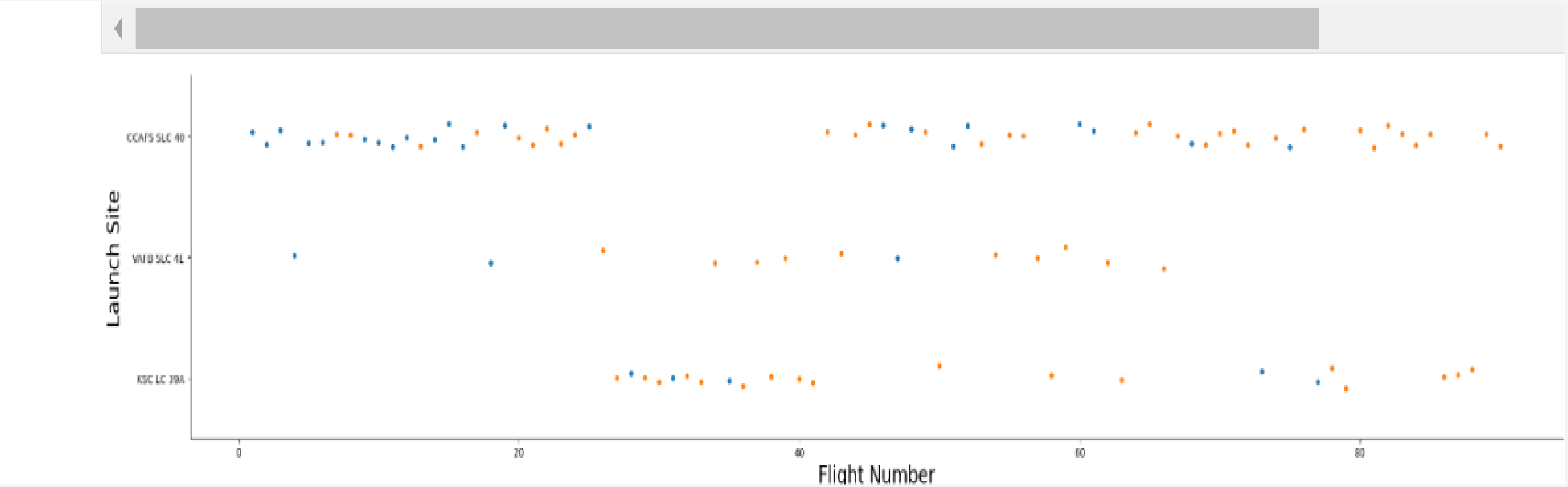
Insights drawn from EDA

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.



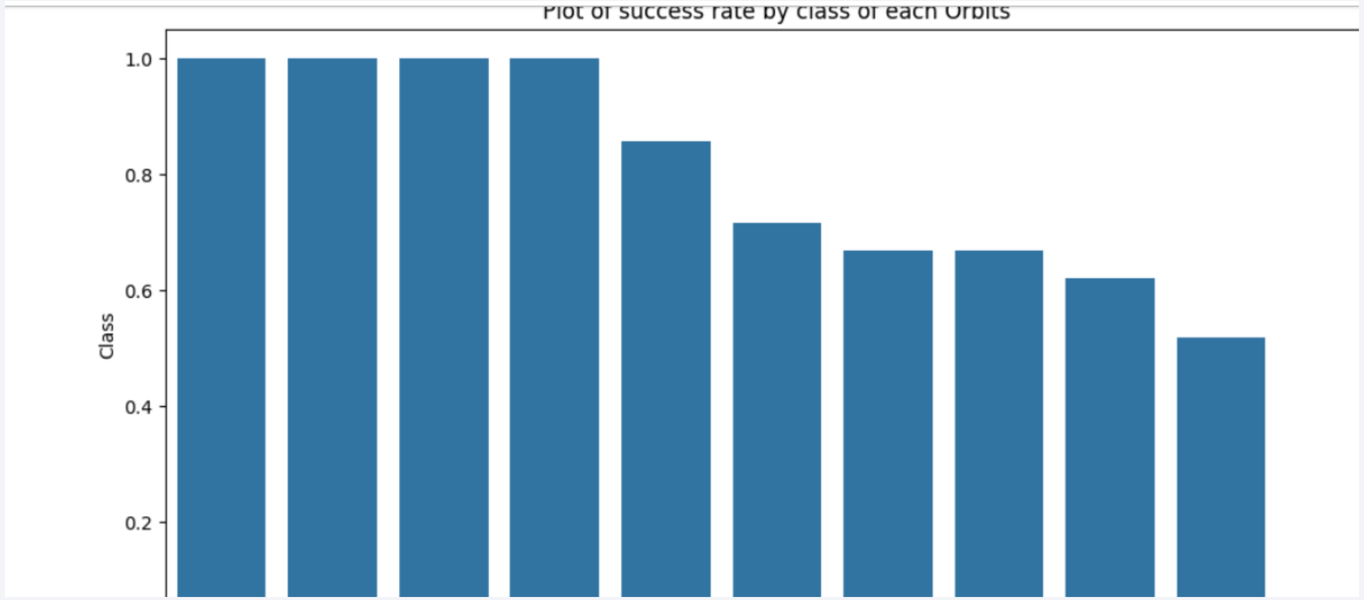
Payload vs. Launch Site



rate

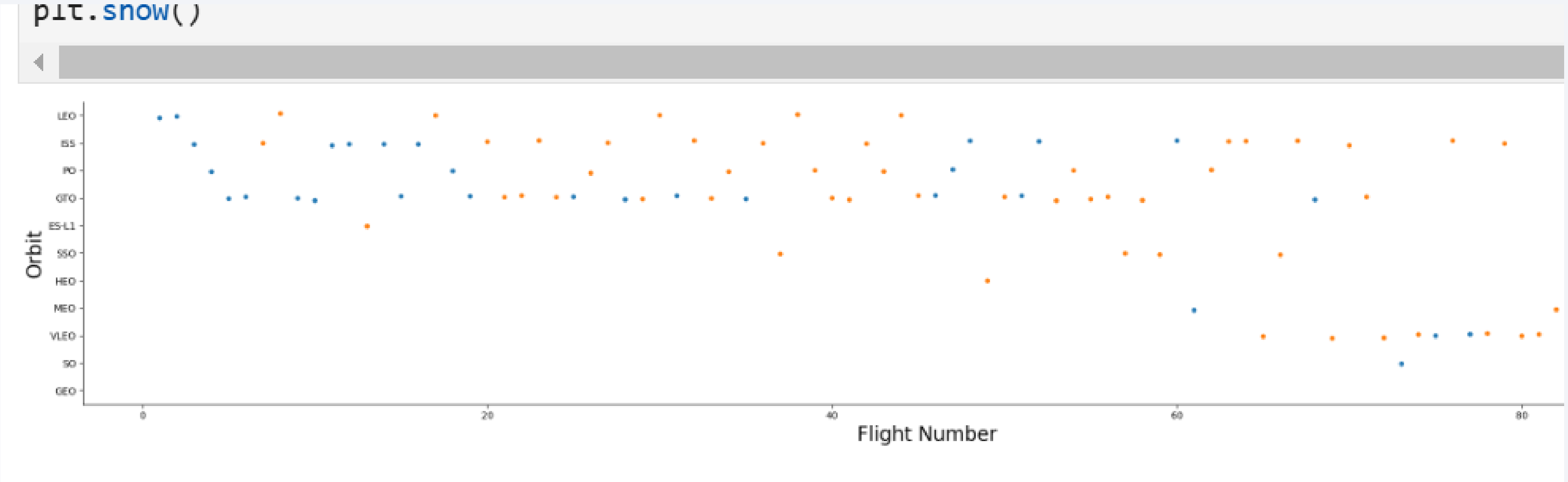
Success Rate vs. Orbit Type

From the plot we can see that ES-L1, GEO, HEO, SSO,VLEO had the most success rate .



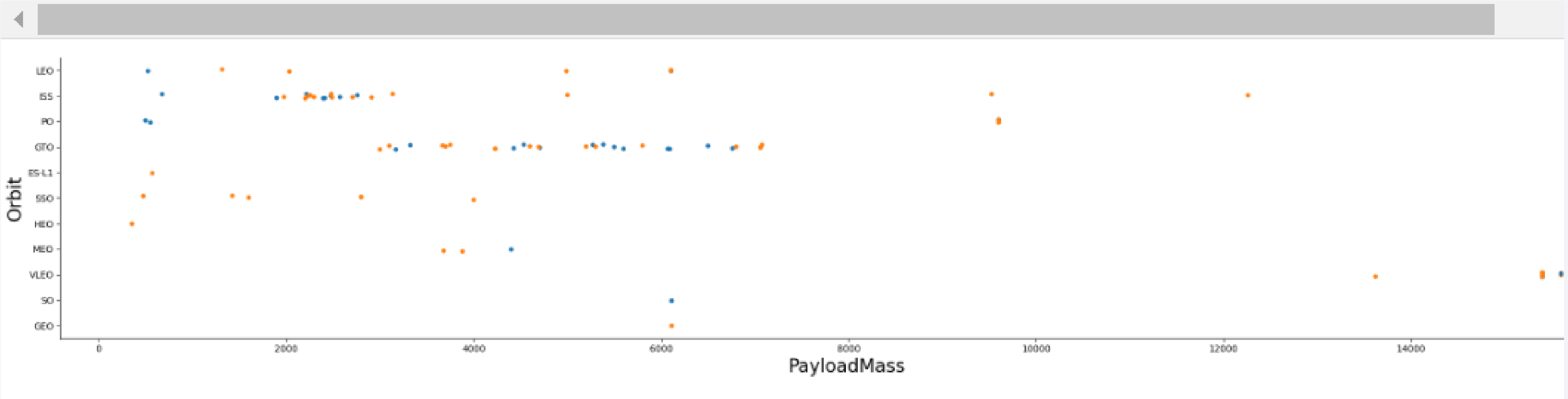
Flight Number vs. Orbit Type

- The plot below shows the 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.



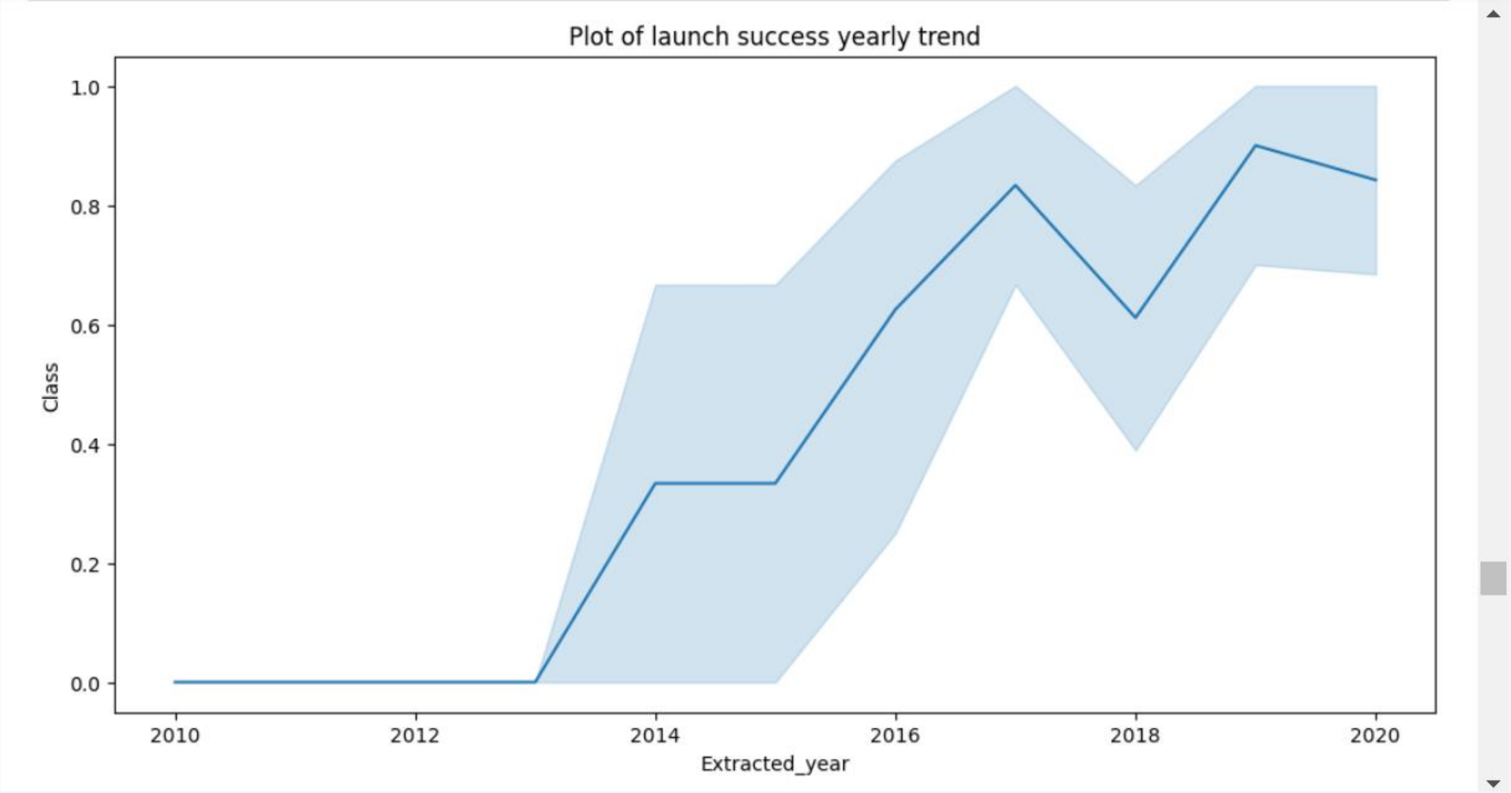
Payload vs. Orbit Type

We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

```
[12]: import pandas as pd
      import random

[14]: df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNe

[15]: unique_launch=df["Launch_Site"].unique().tolist()
      unique_launch

[15]: ['CCAFS LC-40', 'VAFB SLC-4E', 'KSC LC-39A', 'CCAFS SLC-40']
```

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

[16]:
```
count=0
for site in df["Launch_Site"]:
    if "CCA" in site and count < 5:
        print(site)
        count=count+1
```

```
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
```

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

```
[18]: total_mass=df["PAYLOAD_MASS__KG_"].sum()
      print(f"the total mas transported by NASA is {total_mass}KG")
```

```
the total mas transported by NASA is 619967KG
```

Average Payload Mass by F9 v1.1

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

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

```
19]: df["Booster_Version"].value_counts()

19]: Booster_Version
     F9 v1.1          5
     F9 v1.0  B0003   1
     F9 v1.0  B0004   1
     F9 v1.0  B0006   1
     F9 v1.0  B0005   1
                     ..
     F9 B5B1062.1     1
     F9 B5B1061.1     1
     F9 B5B1063.1     1
     F9 B5 B1049.7    1
     F9 B5 B1058.4    1
     Name: count, Length: 97, dtype: int64

20]: df_F9_1_1=df[df["Booster_Version"] == "F9 v1.1"]
     df_F9_1_1["PAYLOAD_MASS__KG_"].mean()

20]: np.float64(2928.4)
```

First Successful Ground Landing Date

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

Hint:Use min function

```
df_ground_pad=df[df['Landing _Outcome'] == 'Success (ground pad)']
df_ground_pad.iloc[0,0]
path=df_ground_pad('path', None)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
df_suc_drone_Mass4000_6000=df[(df['Landing _Outcome'] == 'Success (drone ship)') & (df['PAYLOAD_MASS__KG_'
booster=df_suc_drone_Mass4000_6000["Booster_Version"].to_list()
print(f"Booster versions with the description {booster} have masses between 4000 and 6000 and succeded in
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

[44]:
```
success=0
failure=0
for outcome in df['Mission_Outcome']:
    if "Success" in outcome:
        success=success+1
    else:
        failure=failure+1
print(f"There were {success} succesfull and {failure} unsuccesfull mission")
```

```
There were 100 succesfull and 1 unsuccesfull mission
```

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

```
[31]: max_load=df[df["PAYLOAD_MASS__KG_"] == df['PAYLOAD_MASS__KG_'].max()]
      max_load['Booster_Version'].unique()
```

```
[31]: array(['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 '], dtype=object)
```

2015 Launch Records

and substr(Date,0,5)='2015' for year.

```
5]: dates=[]
for date in df["Date"]:
    if date[6:10] == "2015":
        dates.append(date)
df_2015=df[df["Date"].isin(dates)]
df_2015=df[(df["Landing _Outcome"]=='Failure (drone ship)')]
failed_boosters=df_2015["Booster_Version"].to_list()
print(f"the boosters \n {failed_boosters} \n have failed to land with drone ships in 2015")
```

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

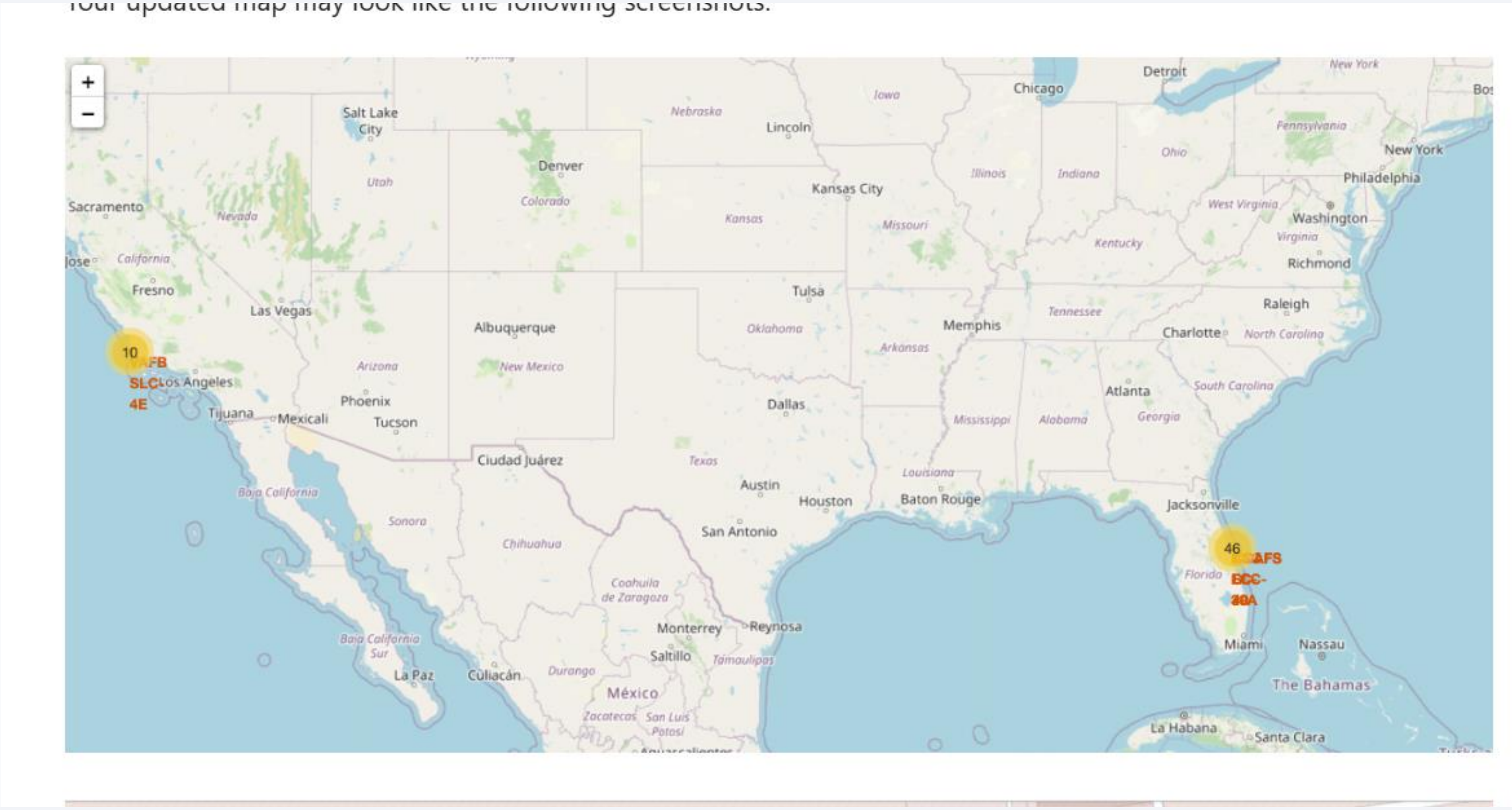
Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dat
06-04 and 2017-03-20, in descending order.

```
lower_date=df[df["Date"]=="04-06-2010"]
higher_date=df[df["Date"]=="16-03-2017"]
df_timed=df.iloc[0:31,]
df_timed['Landing _Outcome'].value_counts(ascending=False)
```

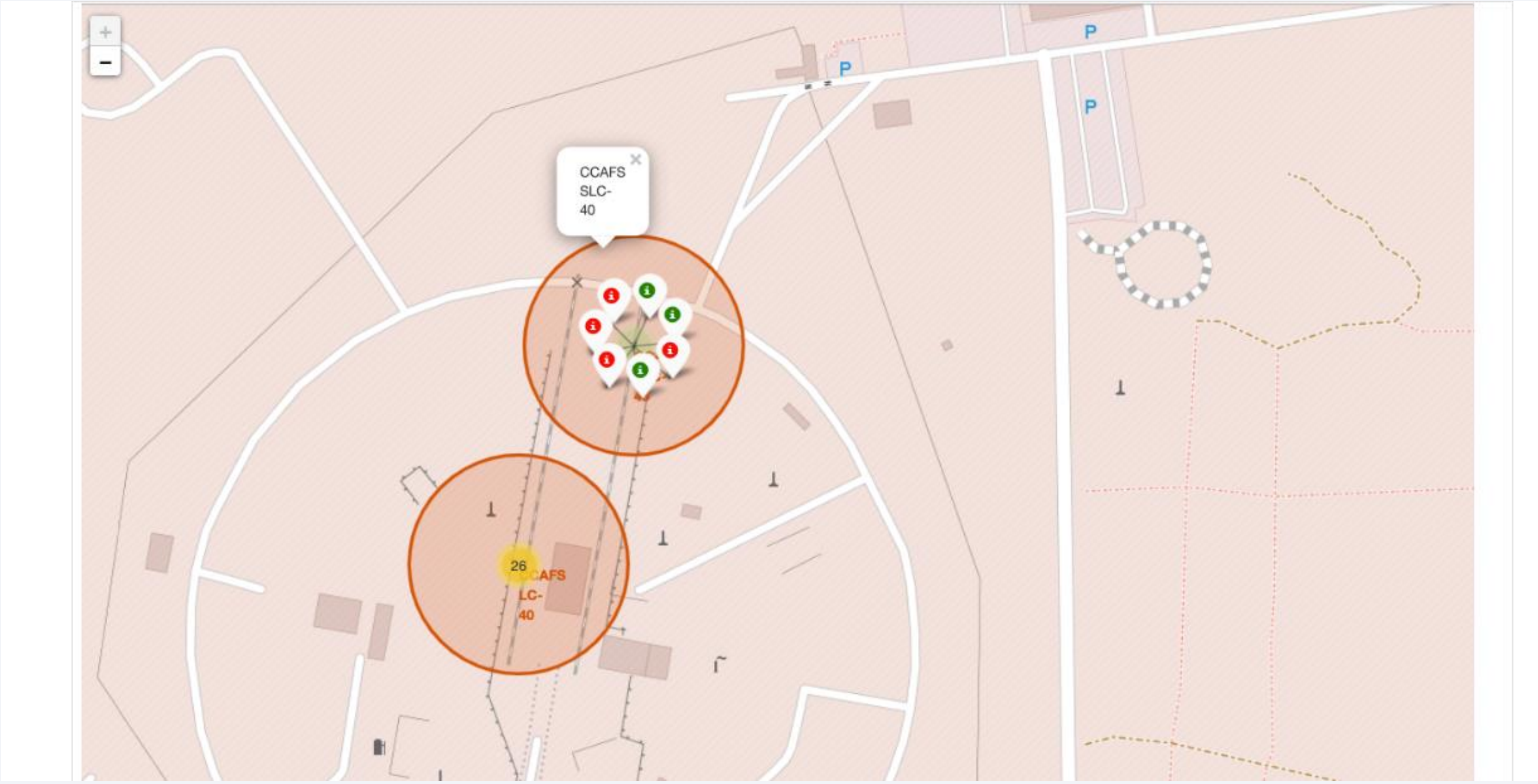
Section 3

Launch Sites Proximities Analysis

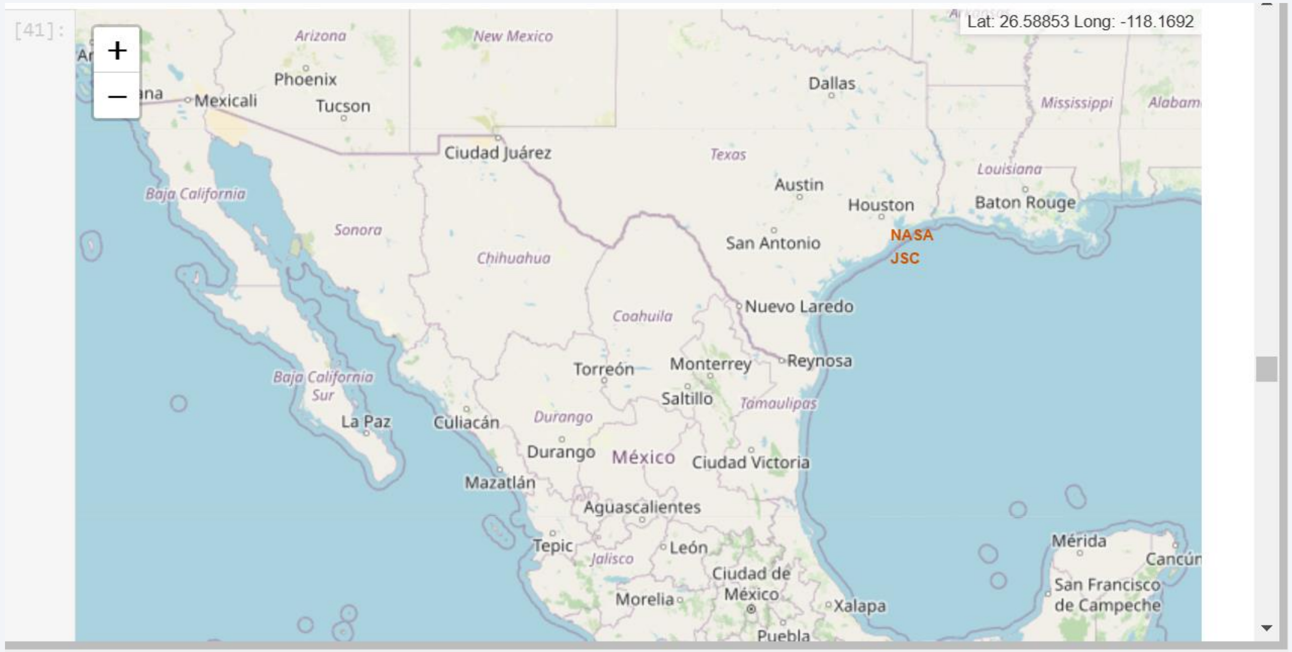
<Folium Map Screenshot 1>



<Folium Map Screenshot 2>



<Folium Map Screenshot 3>



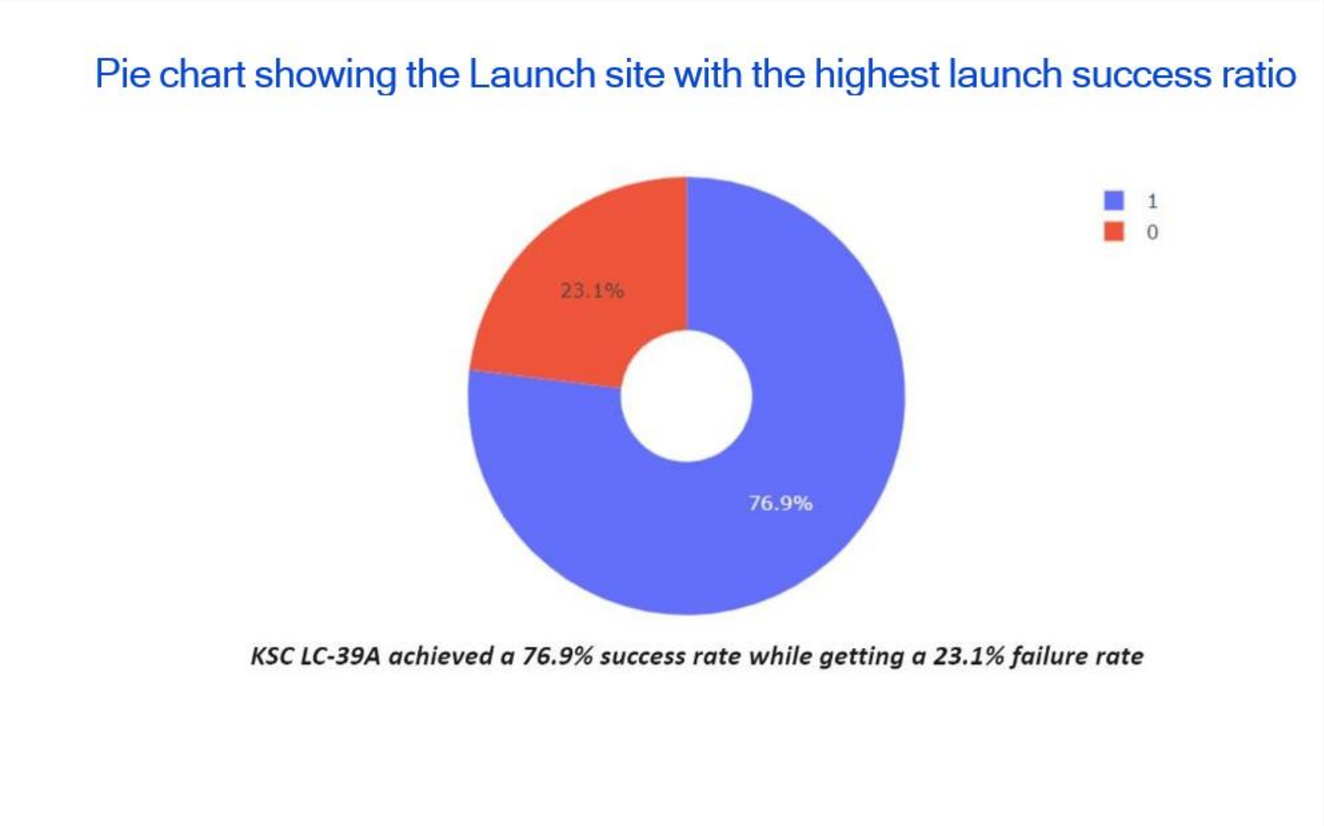
Section 4

Build a Dashboard with Plotly Dash

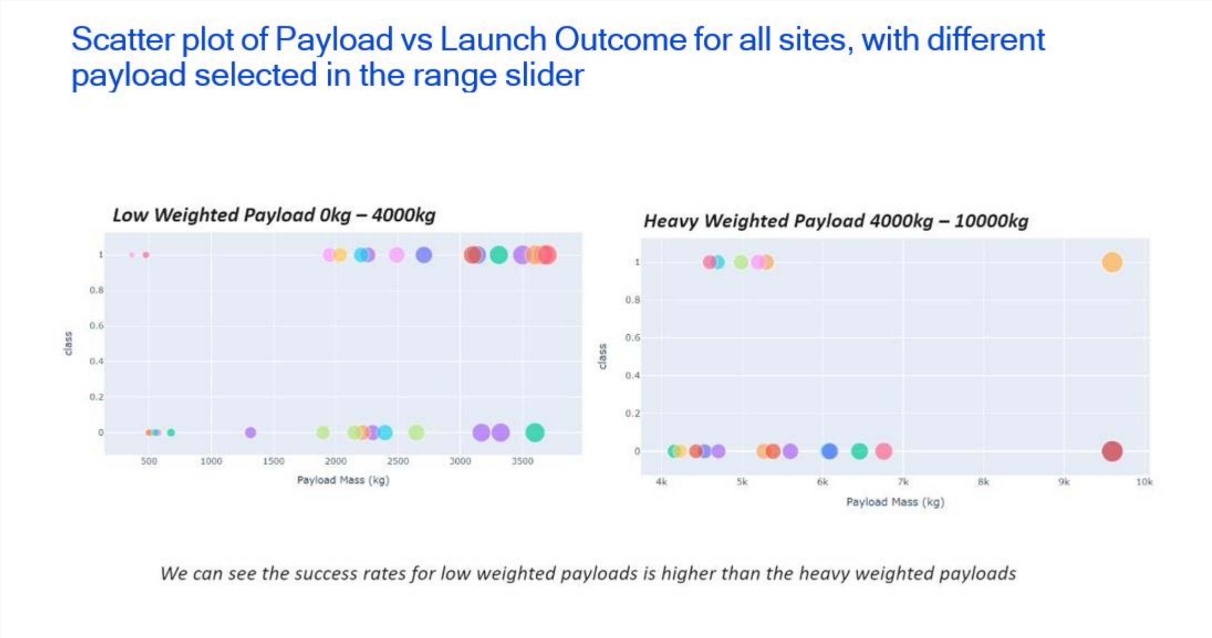
<Dashboard Screenshot 1>

Pie chart showing the success percentage achieved by each launch site

<Dashboard Screenshot 2>



<Dashboard Screenshot 3>



Section 5

Predictive Analysis (Classification)

Classification Accuracy

The decision tree classifier is the model with the highest classification accuracy

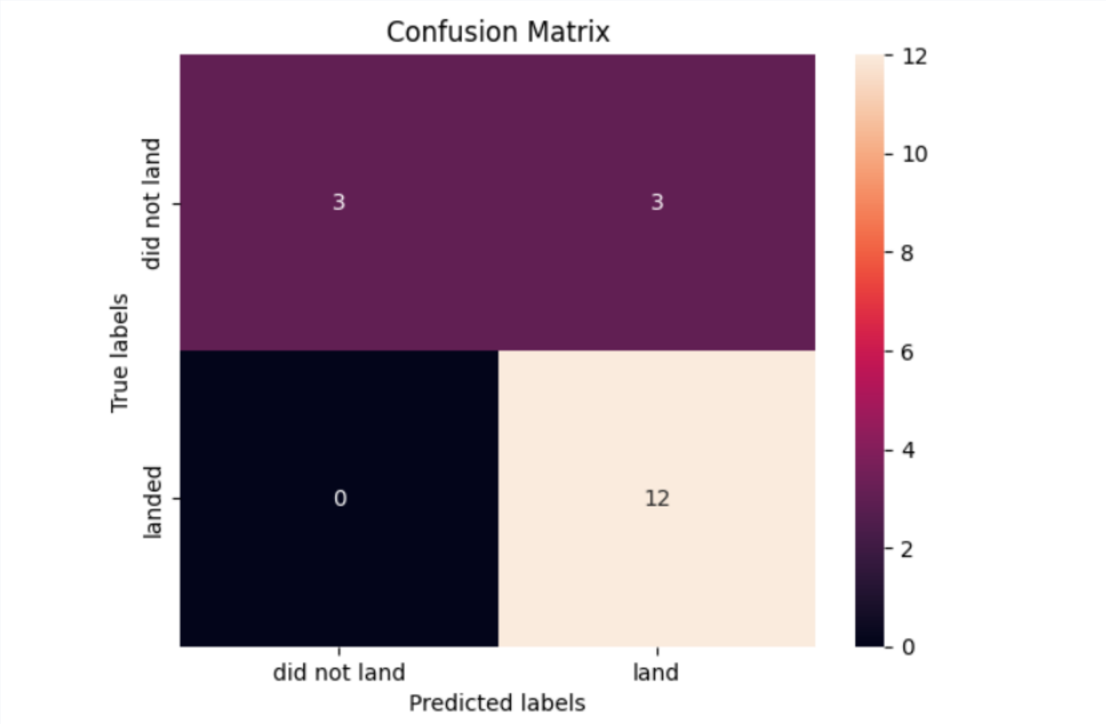
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!