IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ali Farajnia
2022 Nov

Outline

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

Executive Summary

- **Summary of methodologies**
  - EDA with SQL
  - EDA with Data Visualization
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- **Summary of all results**
  - EDA 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:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to 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

- The data was collected using various methods
  - 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

- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/spacex-data-collection-api.ipynb

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

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

```
# Get the head of the dataframe
static_json_df = res.json()
data = pd.json_normalize(static_json_df)
data.head(5)
```

```
# Calculate the mean value of PayloadMass column
PayloadMass = pd.DataFrame(data_falcon9['PayloadMass'].values.tolist()).mean(1)
print(PayloadMass)
# Replace the np.nan values with its mean value
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.
- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/webscraping.ipynb

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

```
# 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
```

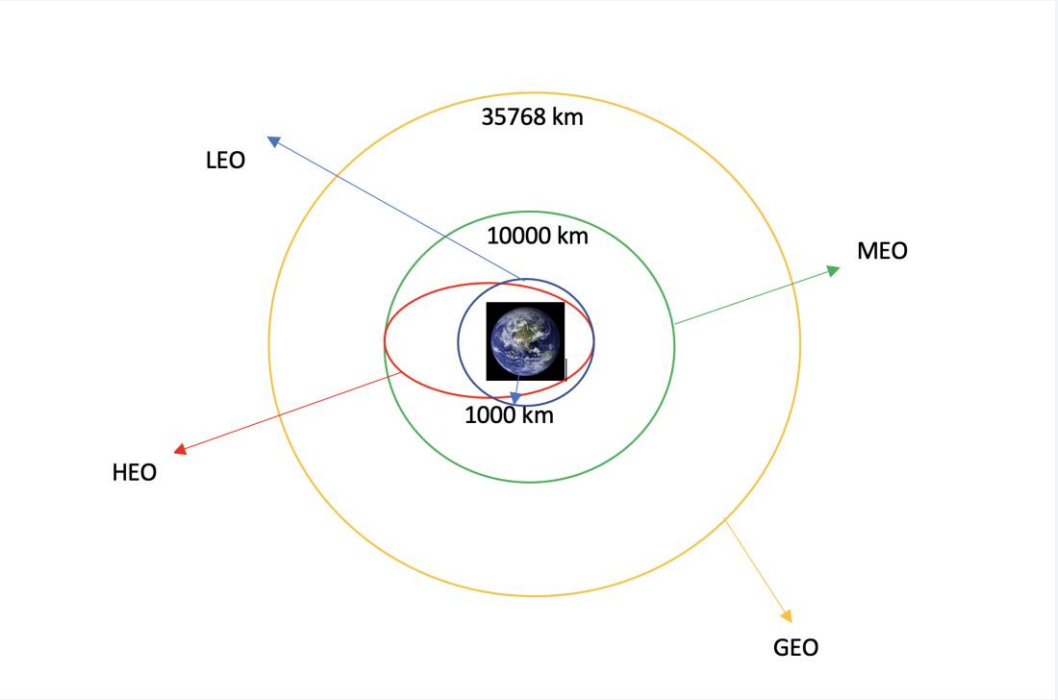
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

```
column_names = []

# Apply find_all() function with `th` element on first_laur
# Iterate each th element and apply the provided extract_cc
# Append the Non-empty column name (`if name is not None ar
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
```

Data Wrangling

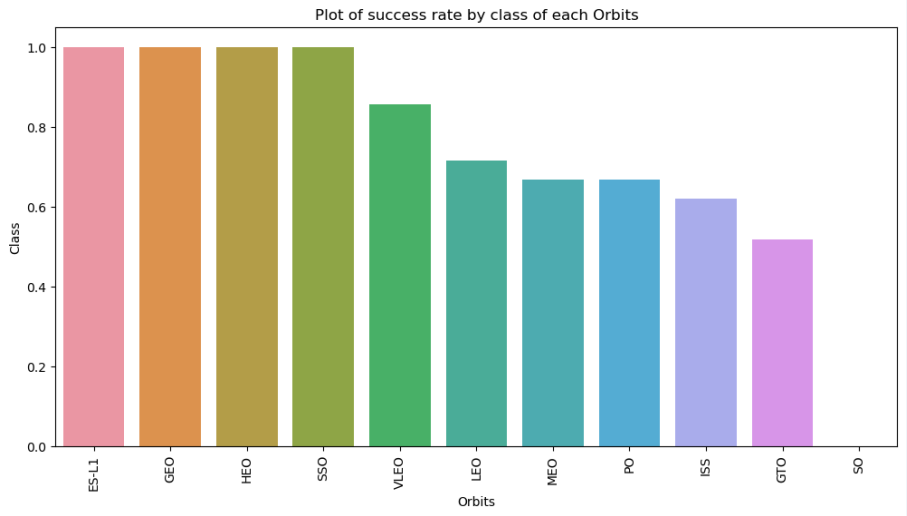
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/data_wrangling.ipynb

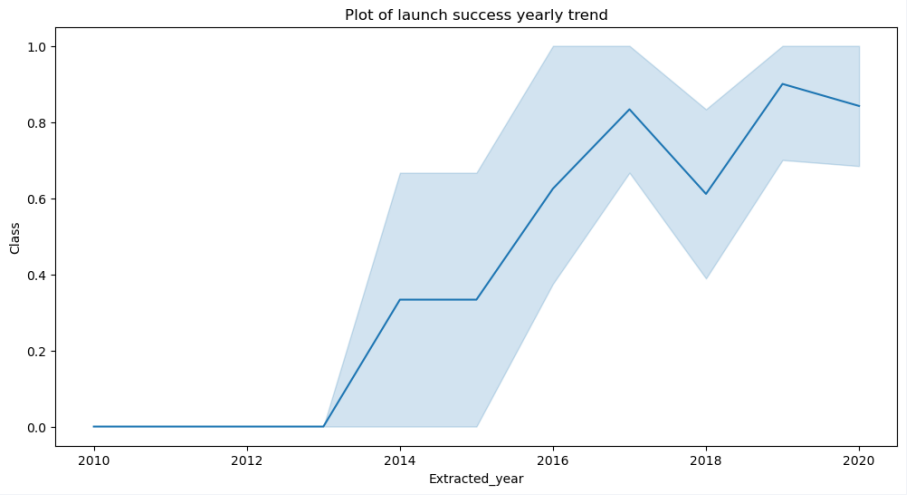


EDA with Data Visualization

- 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.

- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/eda-dataviz.ipynb





EDA with SQL

- We loaded the SpaceX dataset into a sqlite3 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.
- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/eda-sql.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

- First, we built an interactive dashboard with Plotly Dash
- We then drew pie charts showing the total launches by a particular site
- And finally, we drew a scatter plot that shows the relationship with the result and the mass of the load (kg) for different versions of the amplifier.
- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/spacex_dash_app.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.
- https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/SpaceX_Machine_Learning_Prediction.ipynb

Results

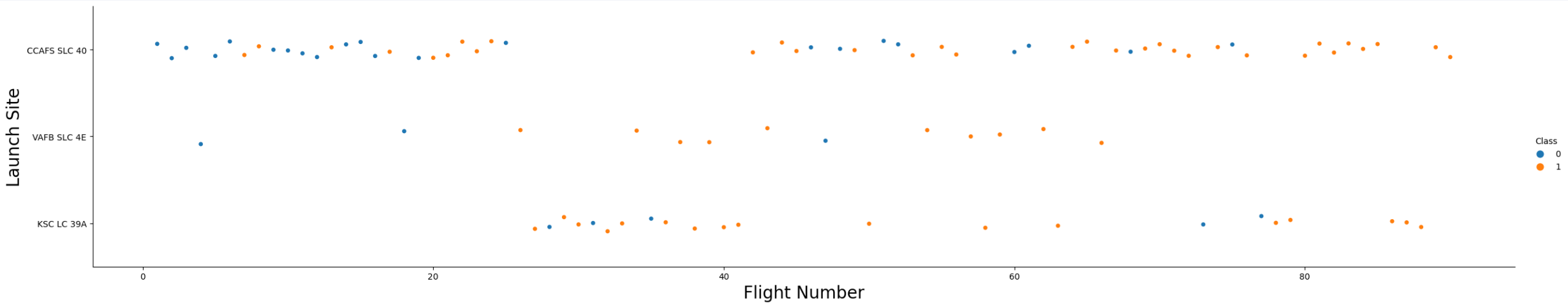
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

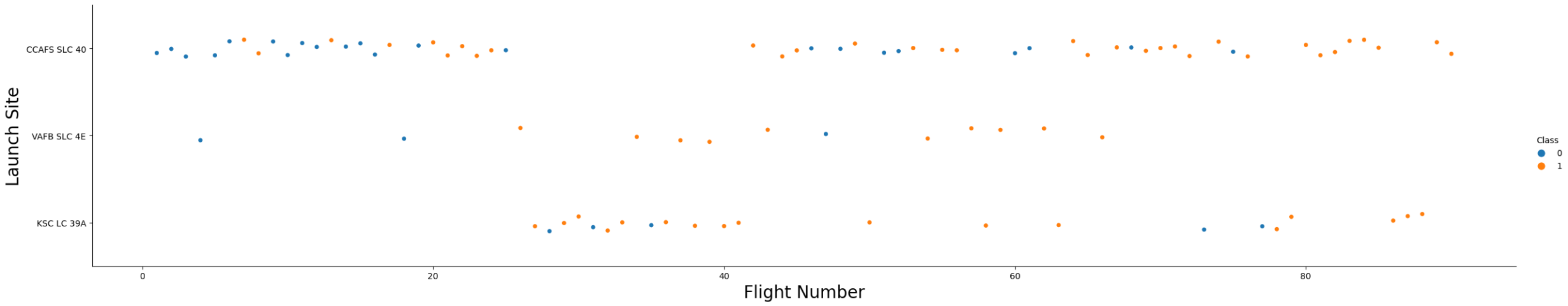
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



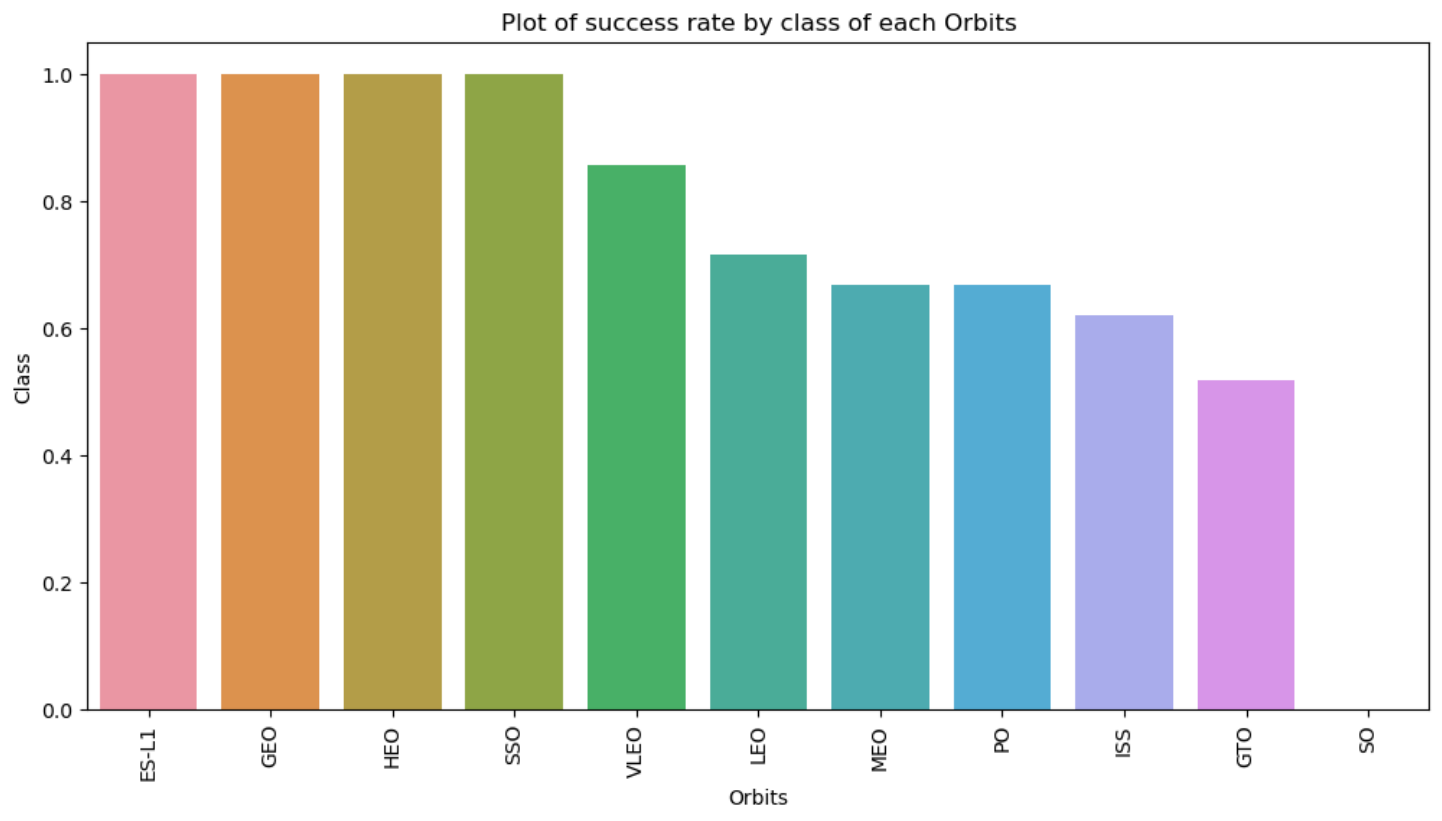
Payload vs. Launch Site

- CCAFS SLC 40 has a higher success rate.



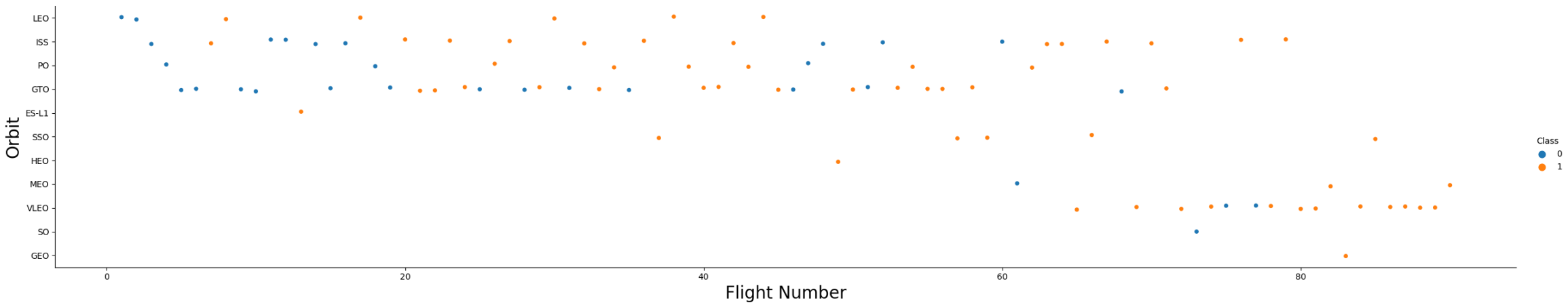
Success Rate vs. Orbit Type

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



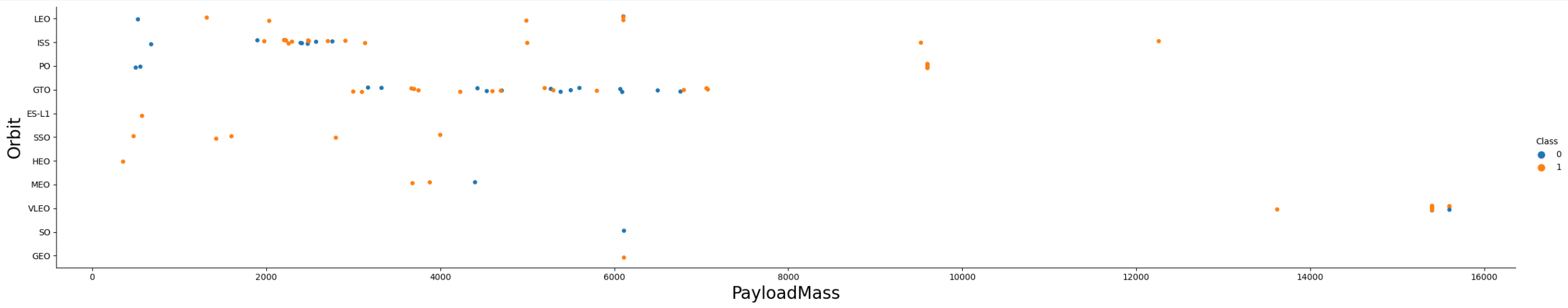
Flight Number vs. Orbit Type

- 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.



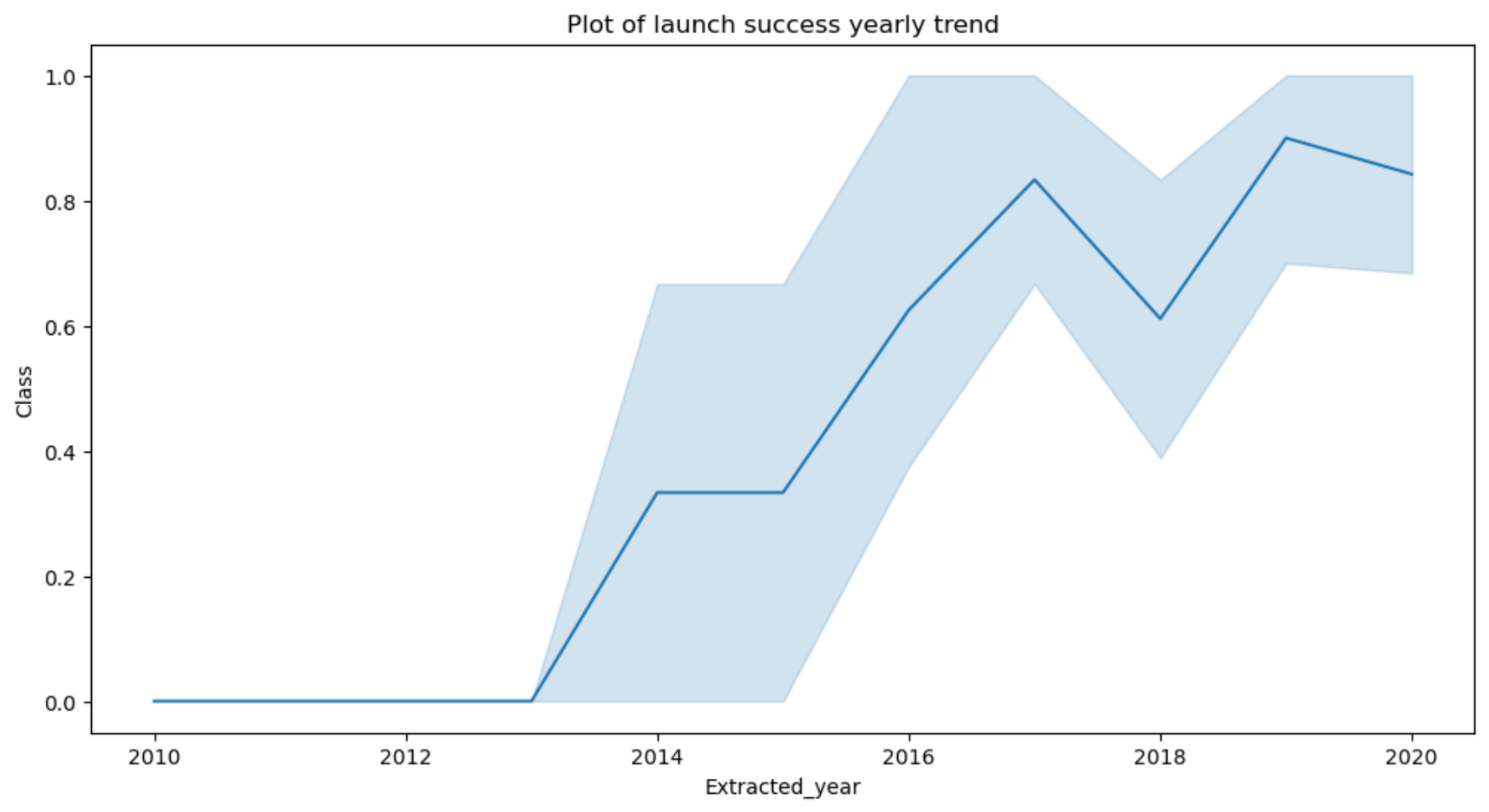
Payload vs. Orbit Type

- 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.

```
# -- %sql select Unique(LAUNCH_SITE) from SPACEXTBL;
cur.execute("select DISTINCT LAUNCH_SITE from SPACEXTBL")
cur.fetchall()
```

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with `CCA`

```
cur.execute("SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5")
cur.fetchall()
```

```
[('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',)]
```

Total Payload Mass

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

```
cur.execute("select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL")
cur.fetchall()
```

```
[(619967,)]
```

Average Payload Mass by F9 v1.1

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

```
cur.execute("select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL")
cur.fetchall()
```

```
[(6138.287128712871,)]
```

First Successful Ground Landing Date

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

Hint:Use min function

```
cur.execute("select min(DATE) from SPACEXTBL")
cur.fetchall()
```

```
[('01-03-2013',)]
```

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

```
cur.execute("select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000")
cur.fetchall()
```

[
('F9 v1.1',), ('F9 v1.1 B1011',), ('F9 v1.1 B1014',), ('F9 v1.1 B1016',), ('F9 FT B1020',),
('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT B1030',), ('F9 FT B1021.2',), ('F9 FT B1032.1’,),
('F9 B4 B1040.1',), ('F9 FT B1031.2',), ('F9 B4 B1043.1',), ('F9 FT B1032.2',), ('F9 B4 B1040.2’,),
('F9 B5 B1046.2',), ('F9 B5 B1047.2',), ('F9 B5 B1046.3',), ('F9 B5B1054',), ('F9 B5 B1048.3',),
('F9 B5 B1051.2 ',), ('F9 B5B1060.1',), ('F9 B5 B1058.2 ',), ('F9 B5B1062.1’,)
]

Total Number of Successful and Failure Mission Outcomes

```
cur.execute("select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME")
cur.fetchall()
```

```
[(1,), (98,), (1,), (1,)]
```

Boosters Carried Maximum Payload

```
cur.execute("select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)")
cur.fetchall()
```

```
[('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

```
cur.execute("SELECT MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL where DATE BETWEEN '01-01-2015' AND '31-12-2015'")
cur.fetchall()
```

```
[('Success', 'F9 v1.0  B0003', 'CCAFS LC-40'),
 ('Success', 'F9 v1.0  B0004', 'CCAFS LC-40'),
 ('Success', 'F9 v1.0  B0005', 'CCAFS LC-40'),
 ('Success', 'F9 v1.0  B0006', 'CCAFS LC-40'),
 ('Success', 'F9 v1.0  B0007', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1  B1003', 'VAFB SLC-4E'),
 ('Success', 'F9 v1.1', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1011', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1010', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1012', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1013', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1014', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1015', 'CCAFS LC-40'),
 ('Success', 'F9 v1.1 B1016', 'CCAFS LC-40'),
```

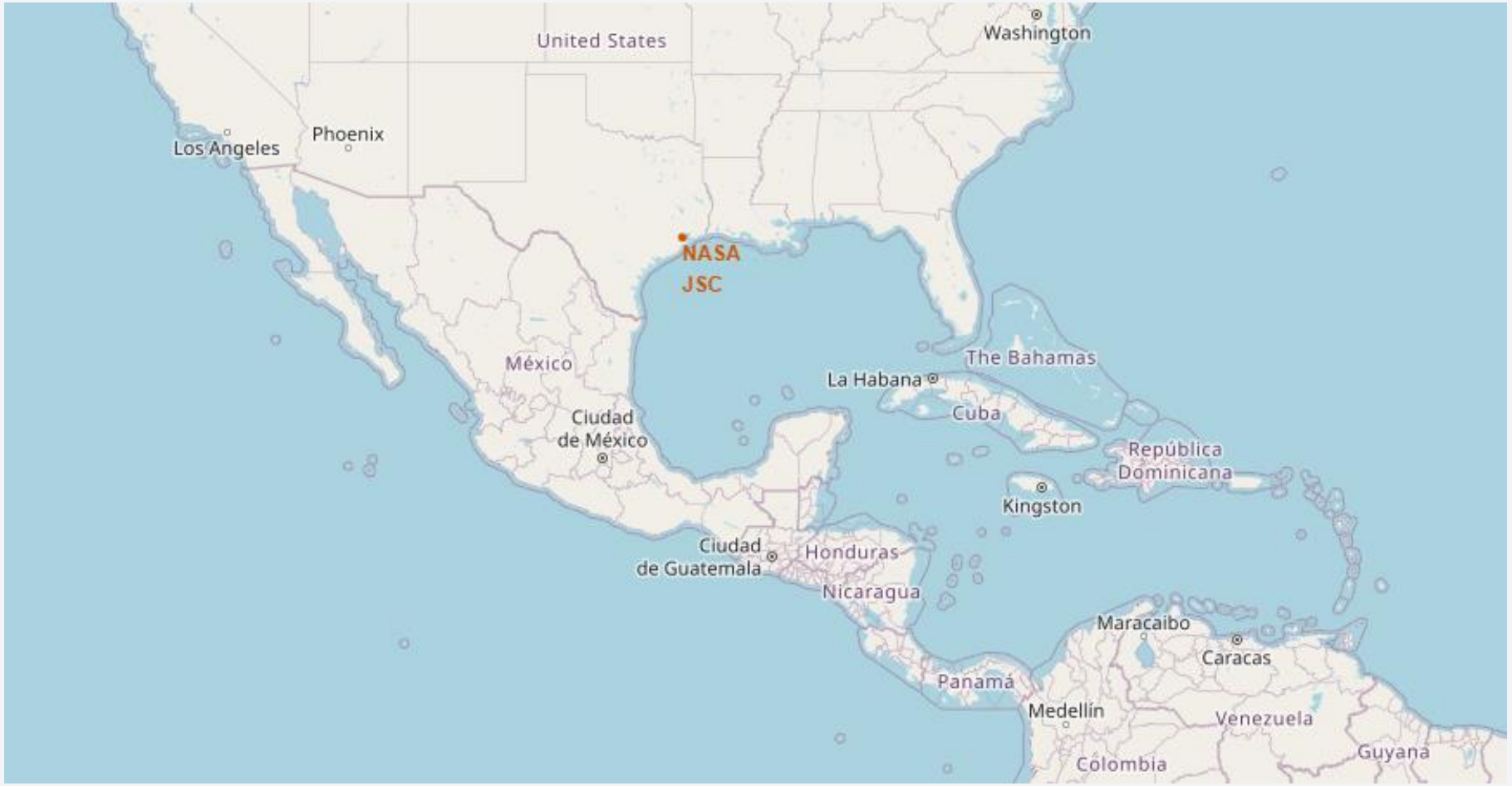
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

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

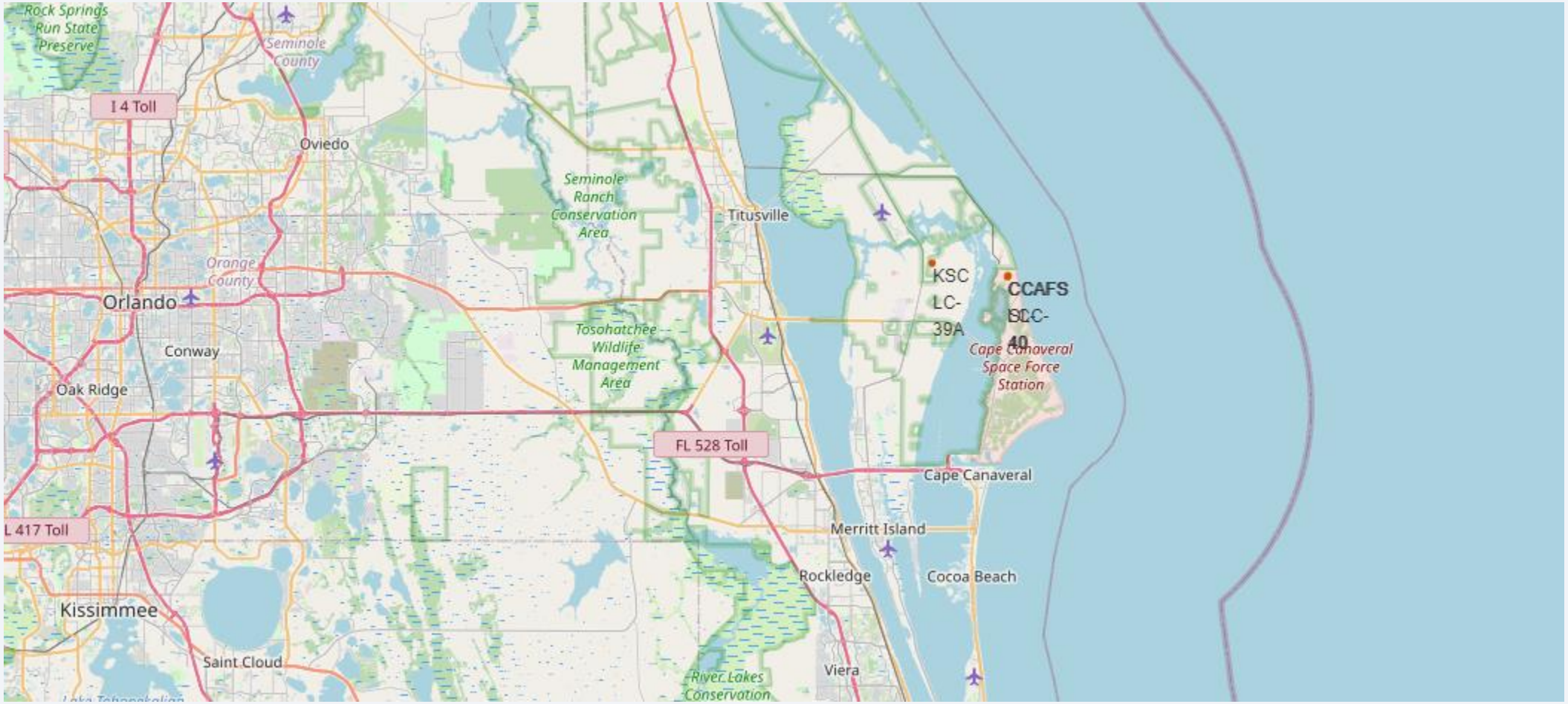
```
cur.execute("SELECT * FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' ORDER BY DATE DESC")
cur.fetchall()
```


Section 3
Launch Sites
Proximities Analysis

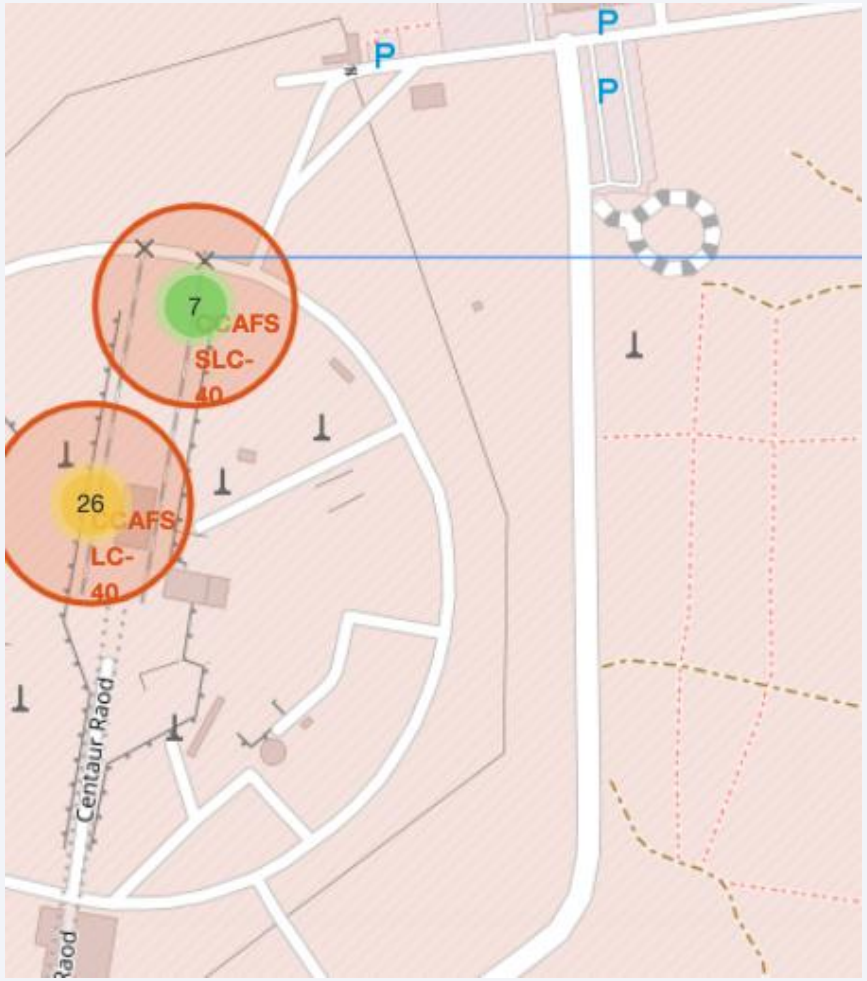
<Folium Map Screenshot 1>

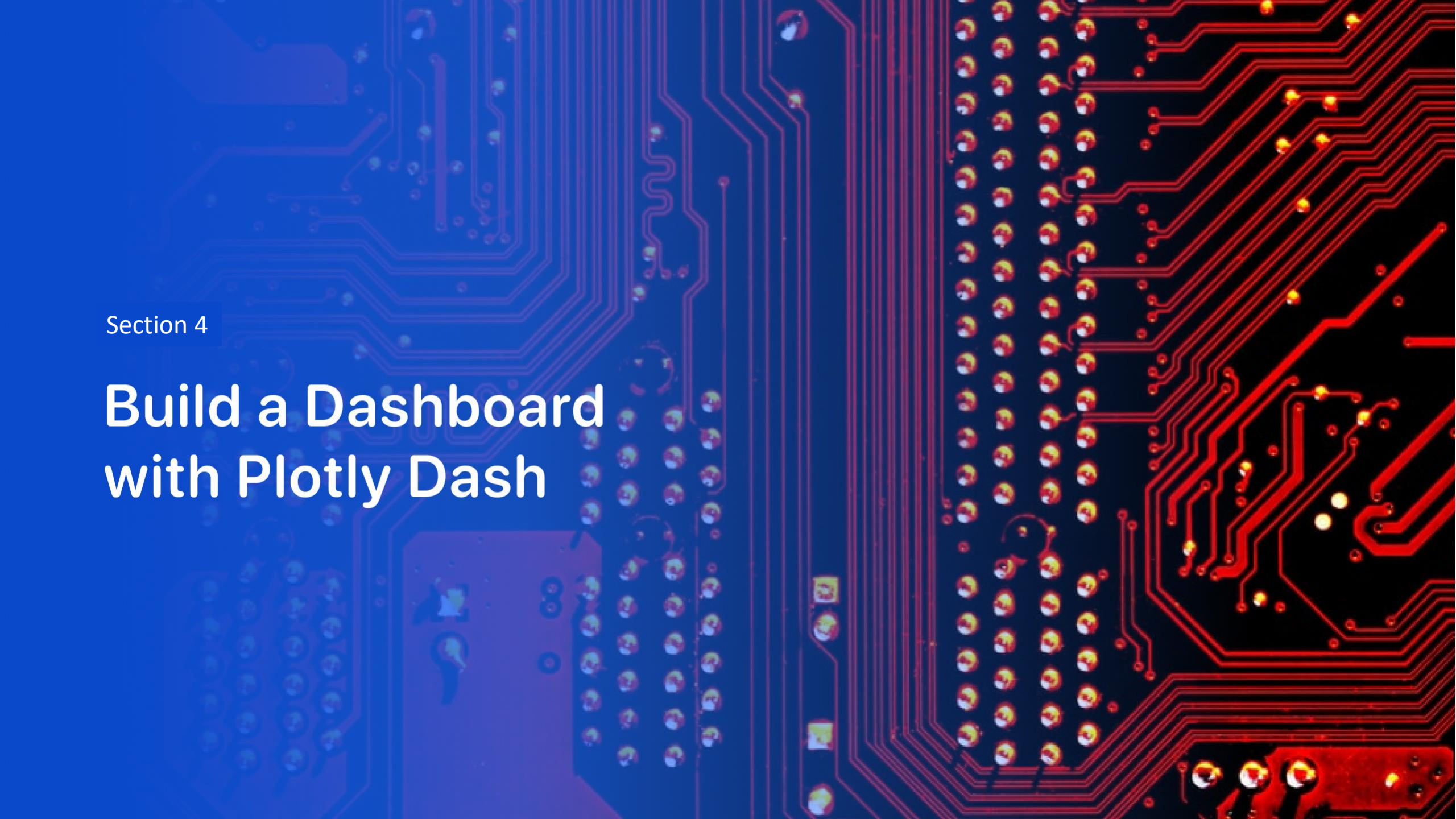


<Folium Map Screenshot 2>



<Folium Map Screenshot 3>




Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

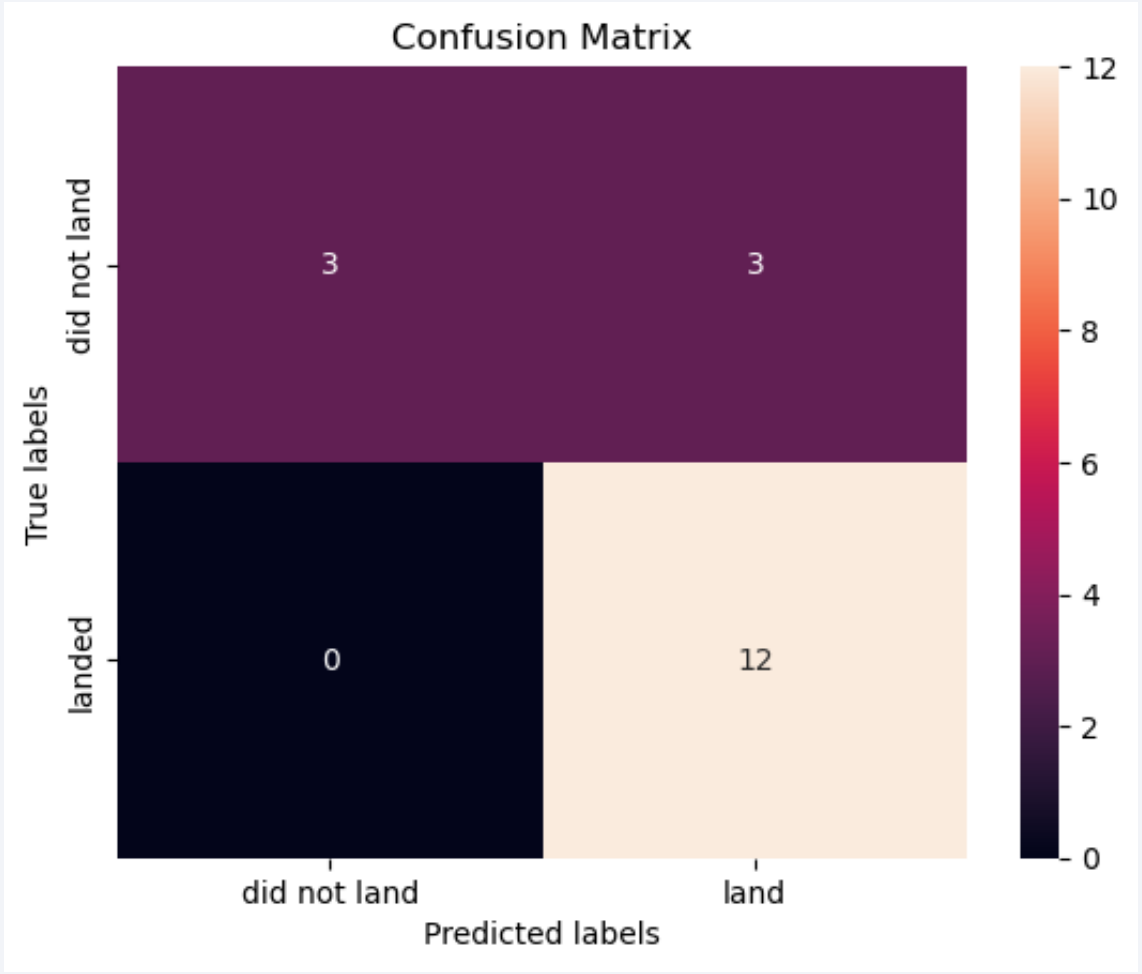
Classification Accuracy

```
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 0.8892857142857142
Best params is : {'criterion': 'gini', 'max_depth': 18, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix



Conclusions

- The higher the number of flights on the launch site, the higher the success rate on the launch site.
- Launch success began to increase from 2013 to 2020.
- ES-L1, GEO, HEO, SSO, VLEO orbits had the best performance and highest success rate.
- KSC LC-39A had the best and most successful launch among other sites.
- Decision tree classification is the best machine learning algorithm for this task.

Appendix

Machine Learning Prediction :

https://github.com/alifarajnia/IBM-DataScience-CapstoneSpaceX/blob/main/SpaceX_MachineLearning_Prediction.ipynb

Thank you!