IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Evangelos Charalampopoulos

11th October 2024

Outline

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

Executive Summary

Summary of methodologies

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K-Nearest Neighbors

Logistic Regression has the highest classification Accuracy

Summary of all results

- Launch Success rate over 60% since 2016
- First successful landing on a ground pad at 12/12/2015
- Successful missions: 60
- Failed missions: 30
- KSC LC-39A is the site with the highest success rate

Introduction

- Prediction if the Falcon 9 first stage will land successfully.
- SpaceX 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 SpaceX can reuse the first stage.
- 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.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data have been retrieved from SpaceX API
- Perform data wrangling
  - Removing of “Falcon 1 launches” keeping only the Falcon 9 launches.
  - Reset the FlightNumber column
  - Dealing with Missing Values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Model Selection, Training, Tuning, Evaluation, Interpretation, Deployment

Data Collection

- Request to the SpaceX API (https://api.spacexdata.com/v4/)

- Web scraping to collect Falcon 9 historical launch records from a Wikipedia page

Data Collection – SpaceX API

- Get request to the SpaceX API
- Clean the requested data
- Create a dataframe
- Filter the dataframe to only include Falcon 9 launches
- Dealing with Missing Values

GitHub URL: https://github.com/xaralabo/falcon/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Web scraping to collect Falcon 9 historical launch records from a Wikipedia page
- Launch records are stored in a HTML table
- Extract Falcon 9 launch records HTML table from Wikipedia using BeautifulSoup
- Extract all column/variable names from the HTML table header
- Parse the table and convert it into a Pandas data frame
- Export to a CSV file

GitHub URL: https://github.com/xaralabo/falcon/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Sequentially the following flow:

- Data Collection
- Data Cleaning
- Data Transformation
- Data Validation
- Save Cleaned Data

GitHub URL: https://github.com/xaralabo/falcon/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Charts for Visualization

- Scatter plot of Flight Number vs. Launch Site
- Scatter plot of Payload vs. Launch Site
- Bar Graph of Payload vs. Launch Site
- Scatter Plot of Flight number vs. Orbit type
- Scatter plot of payload vs. orbit type
- Linear Graph with success Yearly Trend

GitHub: https://github.com/xaralabo/falcon/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

- %sql select DISTINCT(Launch_Site) from SPACEXTABLE
- %sql select * from SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- %sql select sum() from SPACEXTABLE where Customer='NASA (CRS)'
- %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version LIKE 'F9 v1.1'
- %sql select min(Date) from SPACEXTABLE where Landing_Outcome='Success'
- %sql select Booster_Version from SPACEXTABLE where Booster_Version='Success'
- %sql select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome
- %sql select DISTINCT(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
- %sql select substr(Date, 6,2), Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where substr(Date,0,5)='2015'
- %sql select Landing_Outcome, count(*) from SPACEXTABLE where date >= '2010-06-04' and date <= '2017-03-20' group by Landing_Outcome

GitHub: https://github.com/xaralabo/falcon/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All site's location have been added on a map using site's latitude and longitude coordinates
- Highlighted circle area with a text label for each location

GitHub: https://github.com/xaralabo/falcon/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Pie chart with the total successful launches count for all sites
- Scatter chart for the correlation between payload and launch success

GitHub: https://github.com/xaralabo/falcon/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Problem Definition and Data Understanding
- Model Selection (Logistic Regression, Decision Trees, SVM, k-Nearest Neighbors)
- Data Splitting: Divided the dataset into training and testing sets
- Model Training: Trained each model on the training data and used basic evaluation metrics like accuracy to assess initial performance.
- Model Comparison: Analyzed metrics beyond accuracy and confusion matrices
- Model Testing and Deployment

GitHub: https://github.com/xaralabo/falcon/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Launch Success rate over 60% since 2016
- First successful landing on a ground pad at 12/12/2015
- Successful missions: 60
- Failed missions: 30
- KSC LC-39A is the site with the highest success rate
- Logistic Regression Model has the highest classification Accuracy

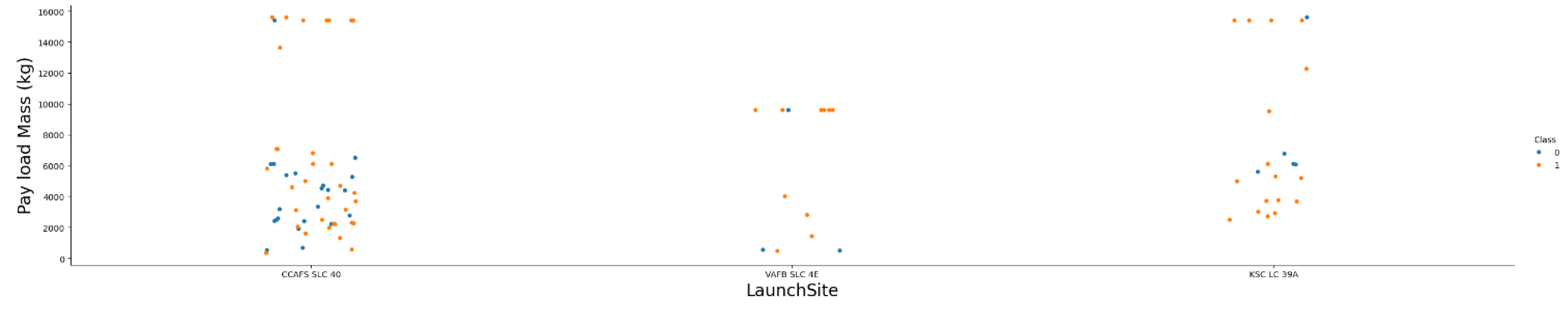
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site
- FlightNumber (indicating the continuous launch attempts.)
- Blue and Green Colors indicate the Class Hue

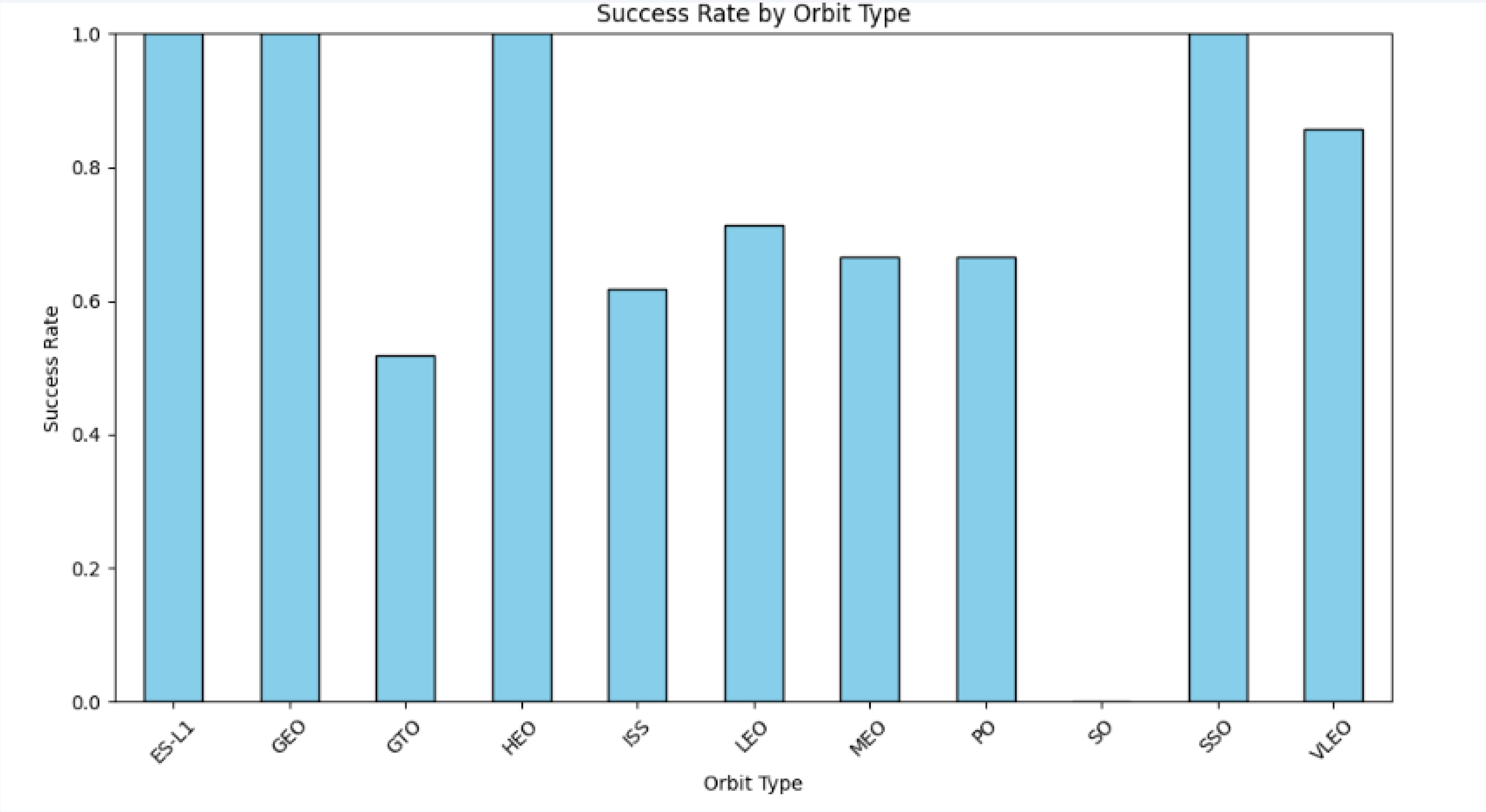
Payload vs. Launch Site



- Scatter plot of Payload vs. Launch Site
- First Launch Site has the most cases
- Blue and orange colors indicate the class

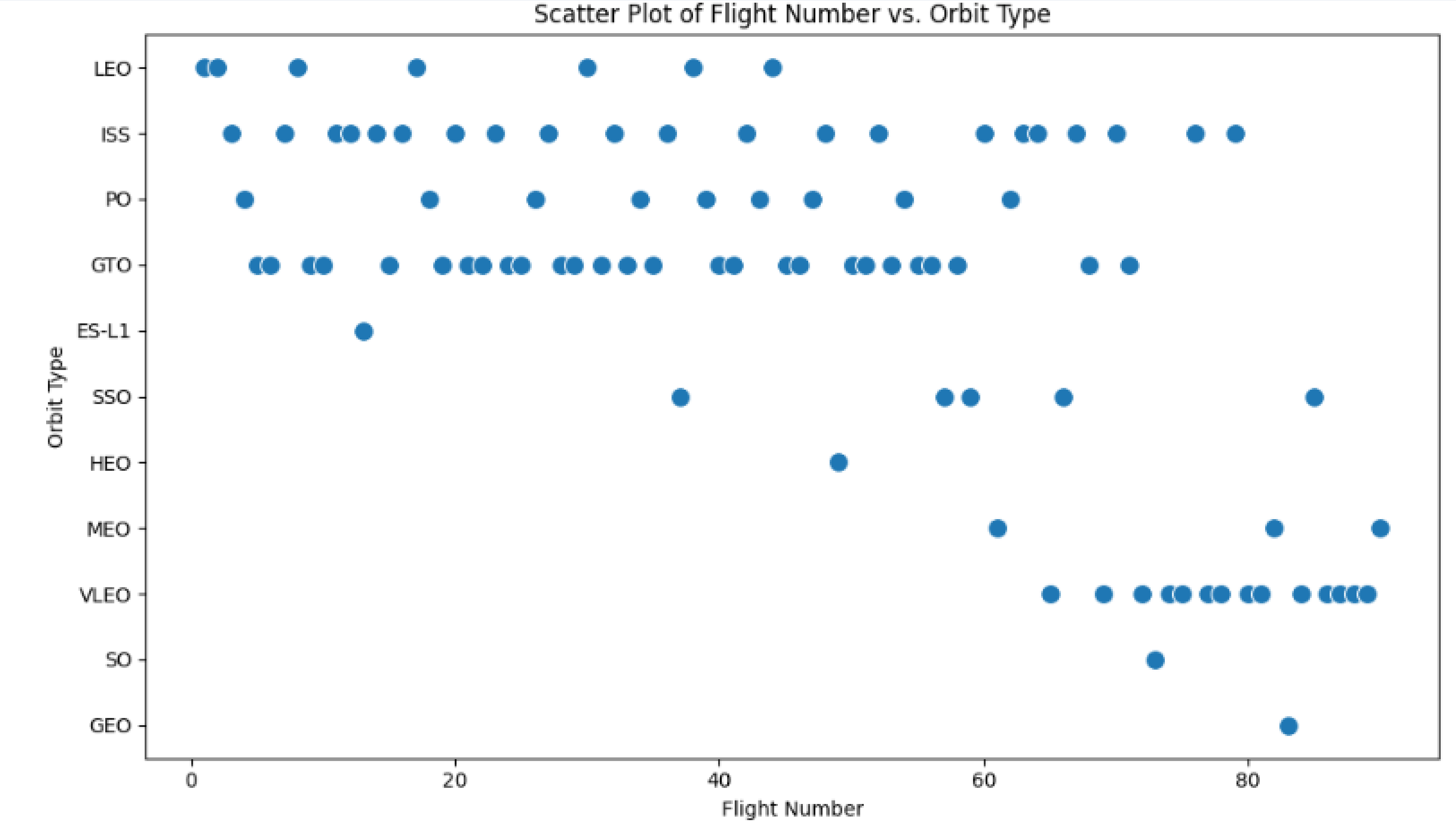
Success Rate vs. Orbit Type

- Four Orbit Types have the highest Success Rate
- SO hasn't any success rate



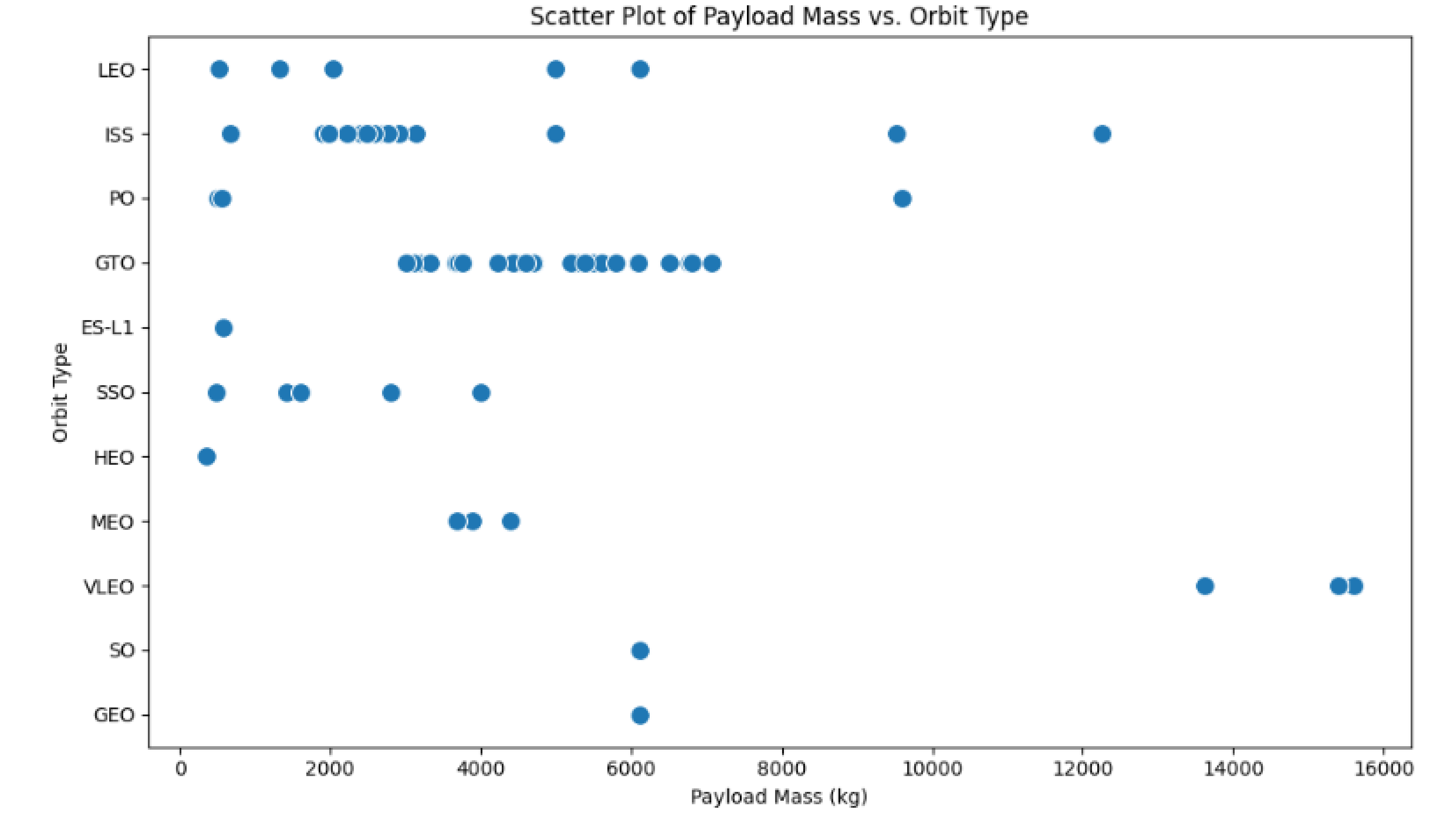
Flight Number vs. Orbit Type

- Scatter Plot of Flight number vs. Orbit type



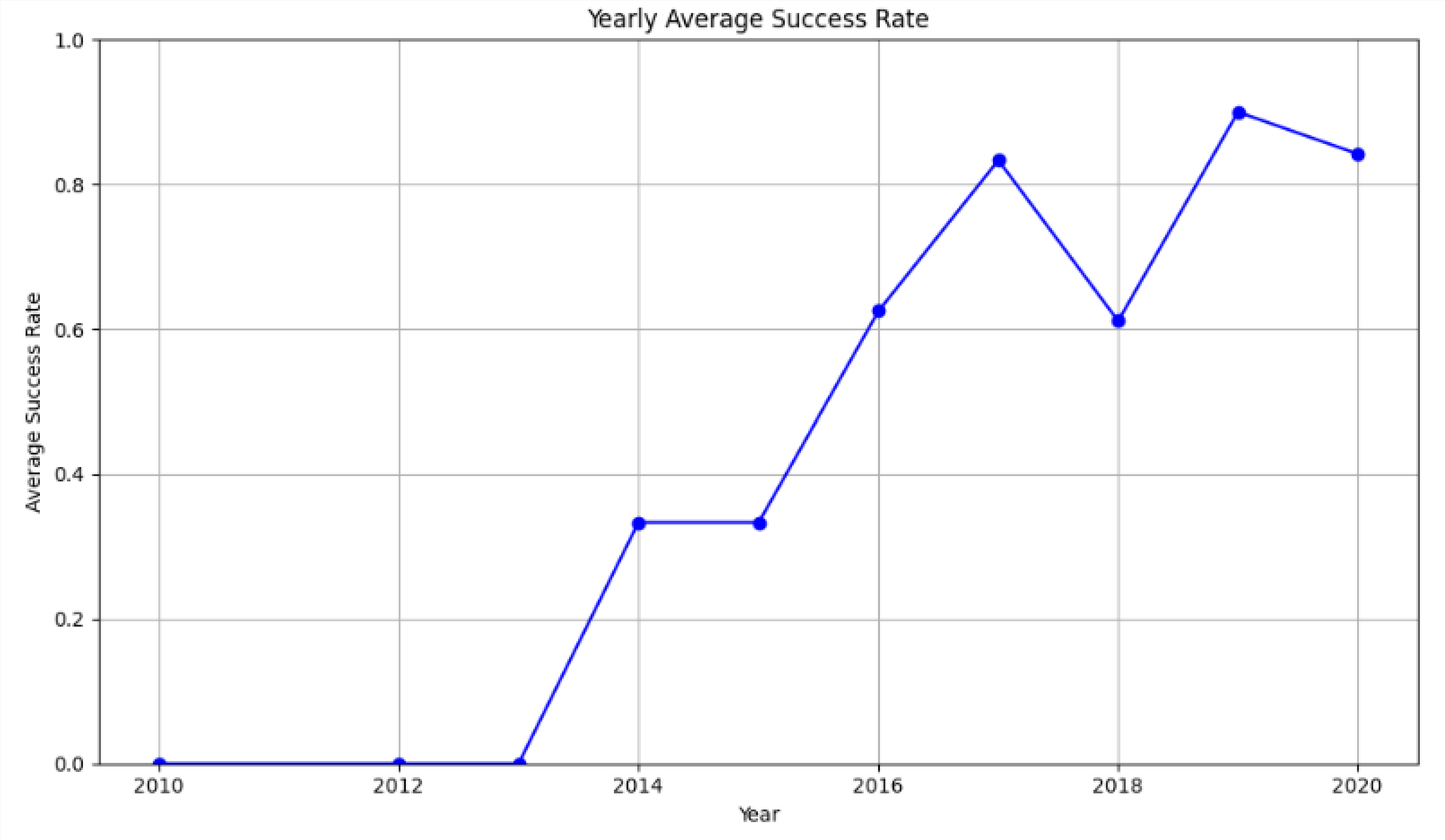
Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type
- GTO hasn't outlier values



Launch Success Yearly Trend

- Zero success rate till 2013
- Success rate over 60% since 2016



All Launch Site Names

Lanch Site Names

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

Code

```
unique_launch_sites = df['LaunchSite'].unique()
print("Unique Launch Sites:")
print(unique_launch_sites)
```

Launch Site Names Begin with 'CCA'

```
   FlightNumber        Date BoosterVersion  PayloadMass Orbit    LaunchSite  \
0             1  2010-06-04       Falcon 9  6104.959412   LEO  CCAFS SLC 40
1             2  2012-05-22       Falcon 9   525.000000   LEO  CCAFS SLC 40
2             3  2013-03-01       Falcon 9   677.000000   ISS  CCAFS SLC 40
4             5  2013-12-03       Falcon 9  3170.000000   GTO  CCAFS SLC 40
5             6  2014-01-06       Falcon 9  3325.000000   GTO  CCAFS SLC 40

     Outcome  Flights  GridFins  Reused   Legs LandingPad  Block  ReusedCount  \
0  None None        1     False   False  False        NaN    1.0            0
1  None None        1     False   False  False        NaN    1.0            0
2  None None        1     False   False  False        NaN    1.0            0
4  None None        1     False   False  False        NaN    1.0            0
5  None None        1     False   False  False        NaN    1.0            0

  Serial  Longitude   Latitude  Class
0  B0003 -80.577366  28.561857      0
1  B0005 -80.577366  28.561857      0
2  B0007 -80.577366  28.561857      0
4  B1004 -80.577366  28.561857      0
5  B1005 -80.577366  28.561857      0
```

Code

```
filtered_records = df[df['LaunchSite'].str.startswith('CCA', na=False)]

print(filtered_records.head(5))
```

Total Payload Mass

- Total payload carried by boosters from NASA

549446.3470588236 kg

Code

```
total_payload_mass = df['PayloadMass'].sum()
print(f"Total Payload Mass Carried by Boosters: {total_payload_mass} kg")
```

Average Payload Mass by F9 v1.1

- Average Payload Mass Carried by Booster Version F9 v1.1: nan kg

- Code

```
f9_v1_1_payloads = df[df['BoosterVersion'] == 'F9 v1.1’]

massaverage_payload_mass = f9_v1_1_payloads['PayloadMass'].mean()

print(f"Average Payload Mass Carried by Booster Version F9 v1.1: {average_payload_mass} kg")
```

First Successful Ground Landing Date

- Date of the first successful landing on a ground pad:

2015-12-22

- Code

```
ground_pad_success = df[df['LandingPad'].notna() & (df['Class'] == 1)]

first_success_date = ground_pad_success['Date'].min()

print(f"Date of the first successful landing on a ground pad: {first_success_date}")
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Falcon 9 has been landed successfully
- Code

```
filtered_boosters = df[
    df['LandingPad'].notna() &  # Ensures LandingPad is not NaN
    (df['Class'] == 1) &  # Successful landings (Class == 1)
    (df['PayloadMass'] > 4000) & (df['PayloadMass'] < 6000)  # Payload mass between 4000 and 6000
]
unique_boosters = filtered_boosters['BoosterVersion'].unique()
print("Boosters that successfully landed on a drone ship with specified LandingPad and payload mass between 4000 and 6000 kg:")
print(unique_boosters)
```

Total Number of Successful and Failure Mission Outcomes

- Total number of successful missions: 60
- Total number of failed missions: 30

- Code:

```
outcome_counts = df['Class'].value_counts()
print("Total number of successful missions:", outcome_counts.get(1, 0))
print("Total number of failed missions:", outcome_counts.get(0, 0))
```

Boosters Carried Maximum Payload

- Falcon 9 has carried the maximum payload mass
- Code

```
max_payload_mass = df['PayloadMass'].max()

boosters_with_max_payload = df[df['PayloadMass'] ==
max_payload_mass]['BoosterVersion'].unique()

print(f"The booster(s) that carried the maximum payload mass of {max_payload_mass} kg:")

print(boosters_with_max_payload)
```

2015 Launch Records

Failed landing in drone ship for in year 2015

```
Failed landing outcomes on drone ship in 2015, along with their booster versions and launch site names:
   BoosterVersion    LaunchSite     Outcome
11       Falcon 9  CCAFS SLC 40  False ASDS
13       Falcon 9  CCAFS SLC 40  False ASDS
14       Falcon 9  CCAFS SLC 40   None None
15       Falcon 9  CCAFS SLC 40   None ASDS
```

Code

```
df['Date'] = pd.to_datetime(df['Date'])

failed_drone_ship_landings_2015 = df[

    (df['Date'].dt.year == 2015) &  # Filter for year 2015

    (df['Class'] == 0)  # Filter for failed outcomes (Class == 0)

]

result = failed_drone_ship_landings_2015[['BoosterVersion', 'LaunchSite', 'Outcome']]

print("Failed landing outcomes on drone ship in 2015, along with their booster versions and launch site names:")

print(result)
```

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

```
Ranked count of landing outcomes between 2010-06-04 and 2017-03-20:
Outcome
None None      9
True ASDS      5
False ASDS     4
True Ocean     3
True RTLS      3
False Ocean    2
None ASDS      2
```

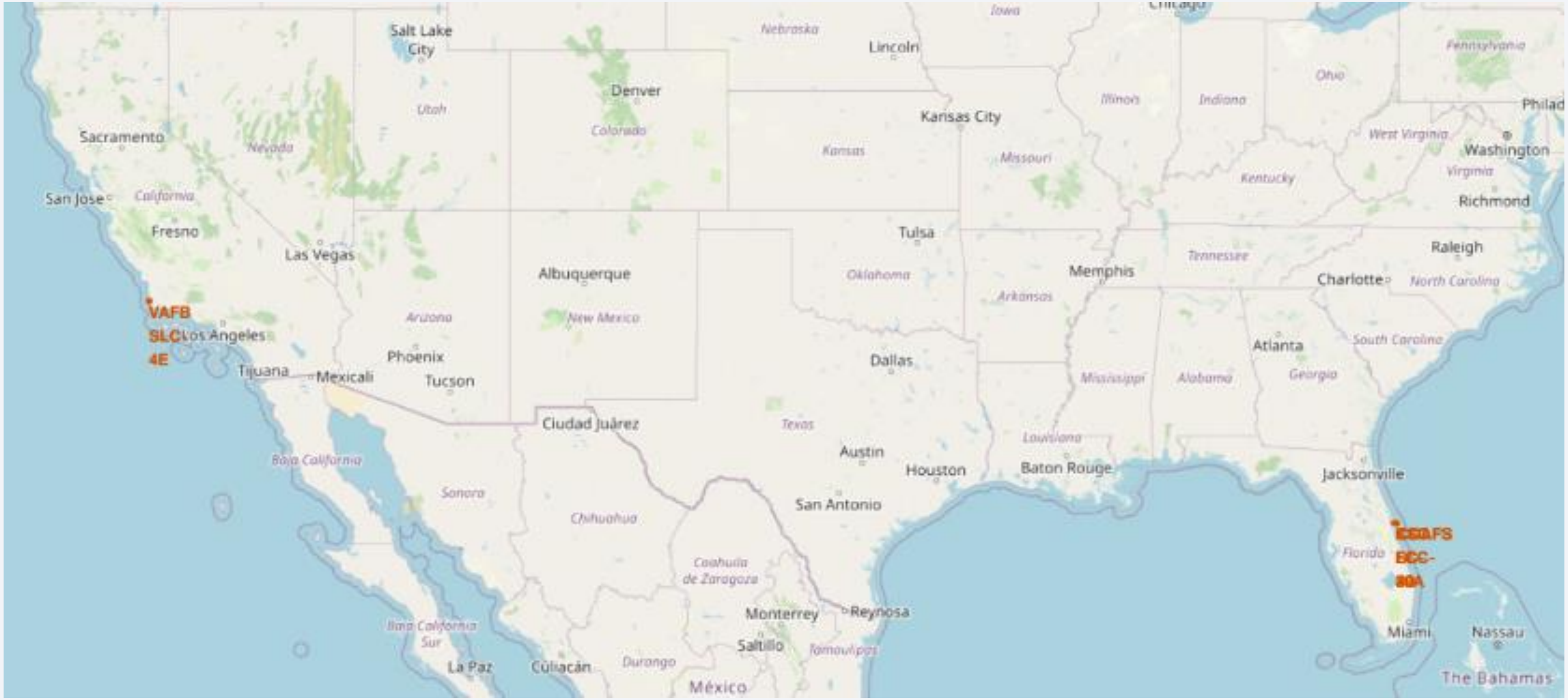
- Code

```
df['Date'] = pd.to_datetime(df['Date'])

filtered_data = df[(df['Date'] >= '2010-06-04') & (df['Date'] <= '2017-03-20')]

landing_outcome_counts = filtered_data['Outcome'].value_counts()

landing_outcome_counts_sorted = landing_outcome_counts.sort_values(ascending=False)

print("Ranked count of landing outcomes between 2010-06-04 and 2017-03-20:")

print(landing_outcome_counts_sorted)
```

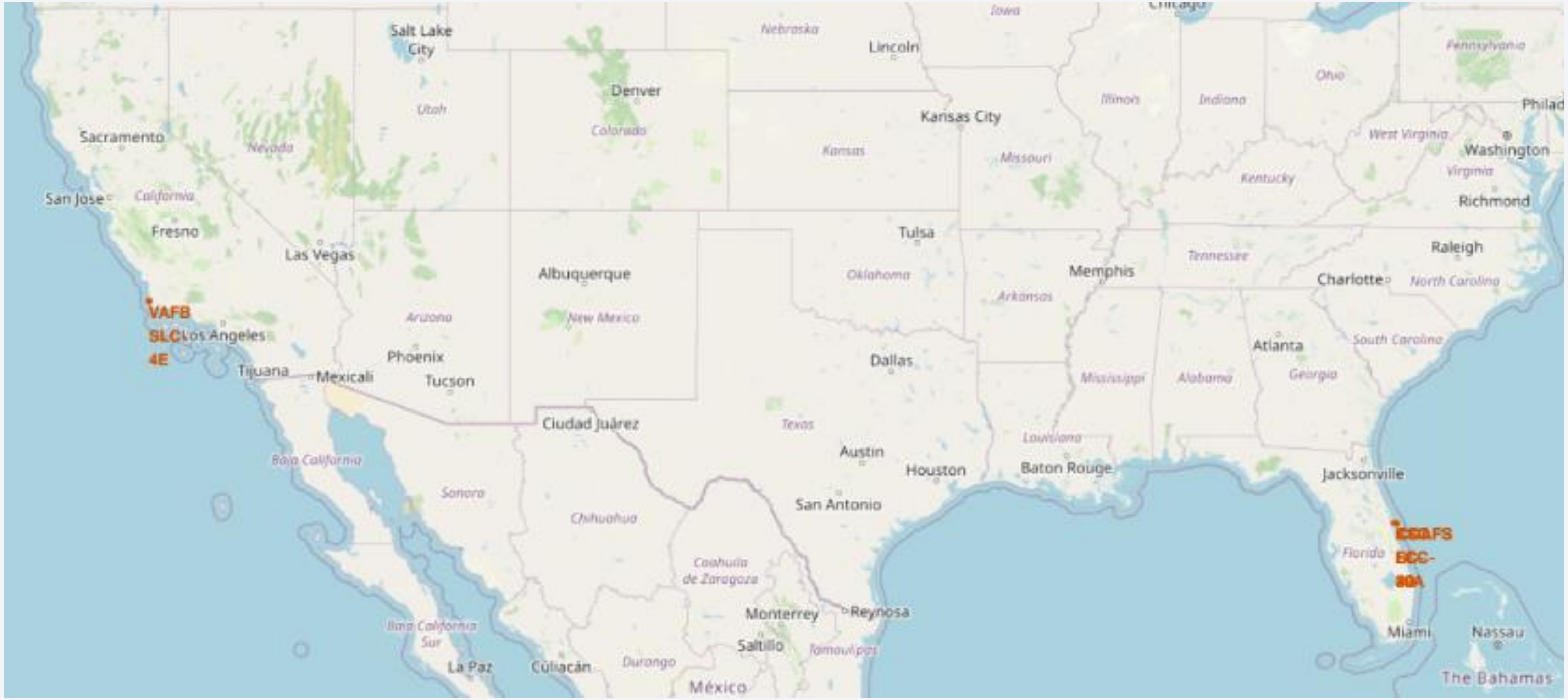
Section 3

Launch Sites Proximities Analysis

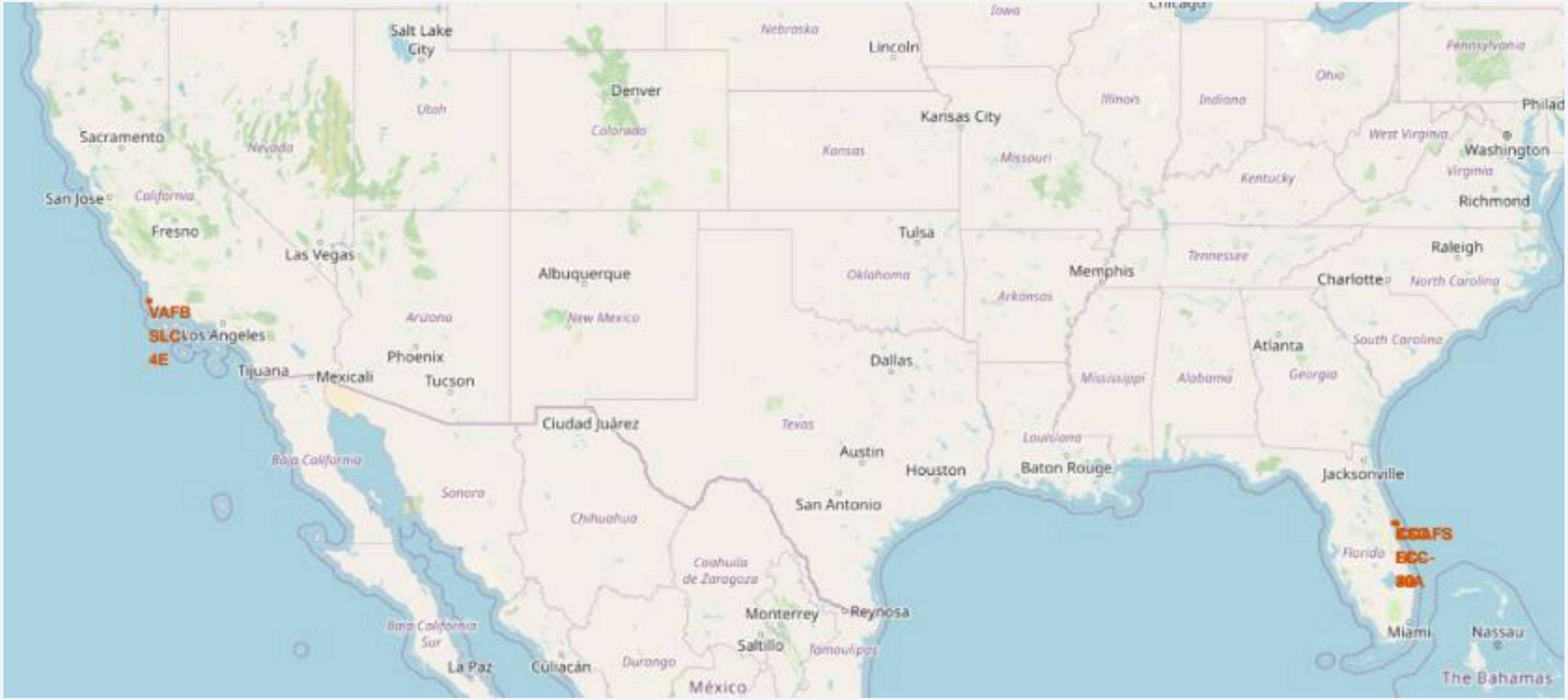
All Launch Sites



Launch outcomes on the map



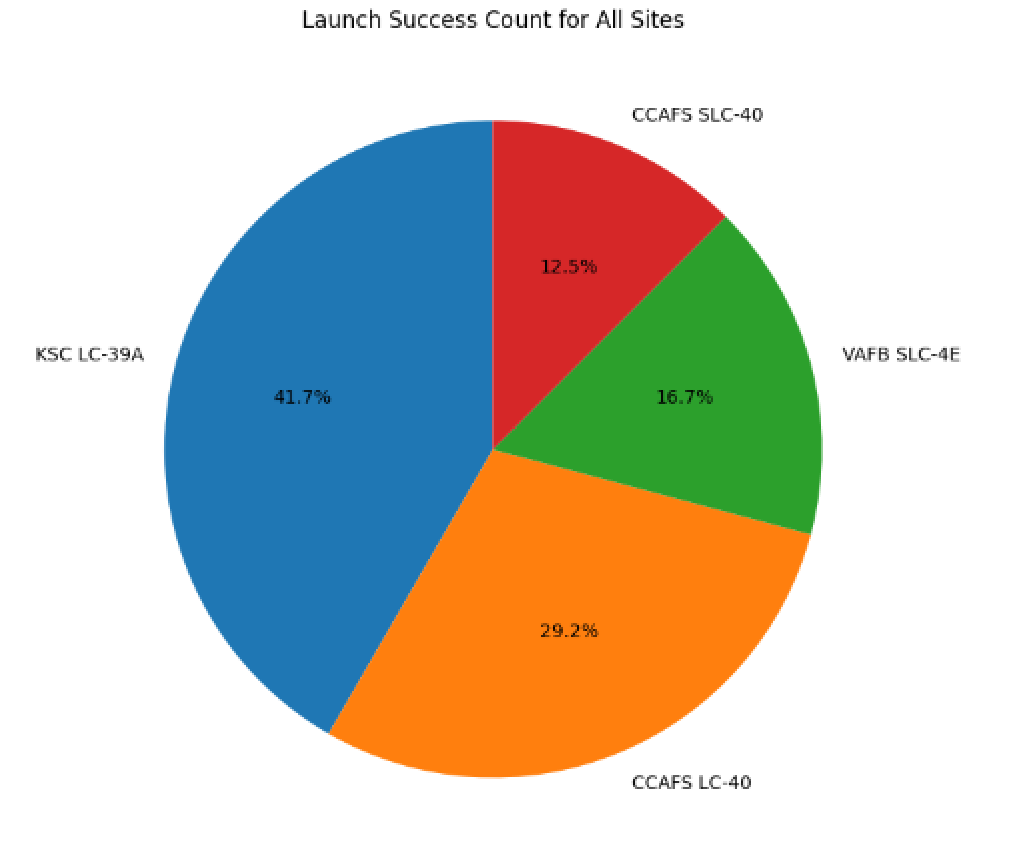
Map


Salt Lake City
Nebraska
Lincoln
Iowa
Pennsylvania
Ohio
Denver
Utah
Colorado
Kansas City
Illinois
Indiana
Philad
Sacramento
Nevada
Kansas
Missouri
West Virginia
Washington
Kentucky
Virginia
San Jose
California
Richmond
Fresno
Tulsa
Las Vegas
Raleigh
Albuquerque
Oklahoma
Memphis
Tennessee
Charlotte
North Carolina
Arkansas
VAFB
Arizona
New Mexico
SLC
Los Angeles
South Carolina
Atlanta
Phoenix
Dallas
4E
Tijuana
Mexicali
Tucson
Mississippi
Alabama
Georgia
Ciudad Juárez
Texas
Louisiana
Baja California
Austin
Houston
Baton Rouge
Jacksonville
Sonora
San Antonio
Chihuahua
Florida
Coahuila de Zaragoza
Monterrey
Reynosa
Baja California Sur
Saltillo
Tamaulipas
Miami
Nassau
La Paz
Culiacán
Durango
México
The Bahamas

Section 4

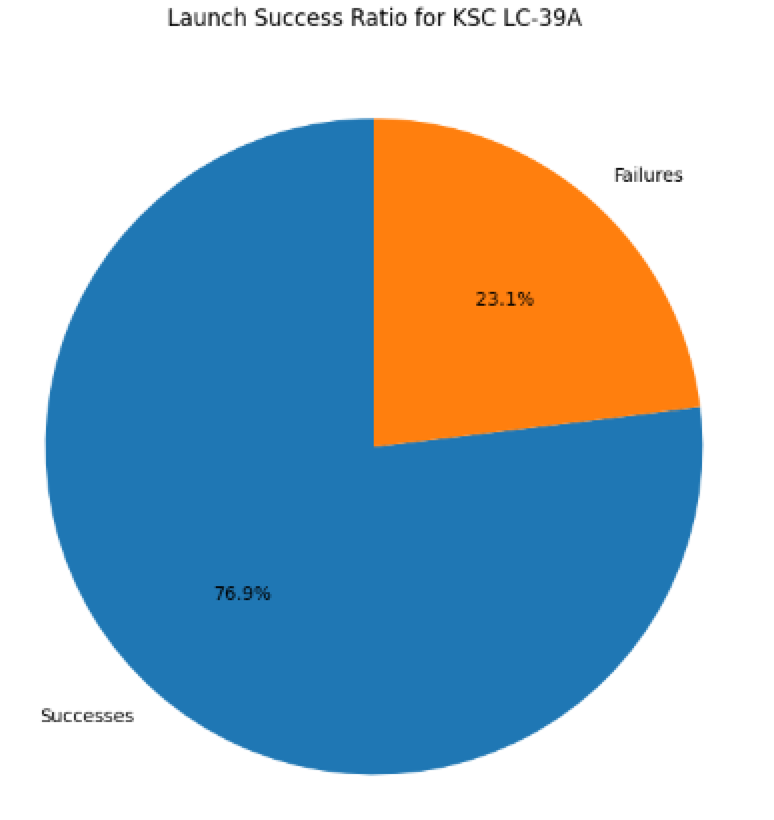
Build a Dashboard with Plotly Dash

Launch Success Count for all sites



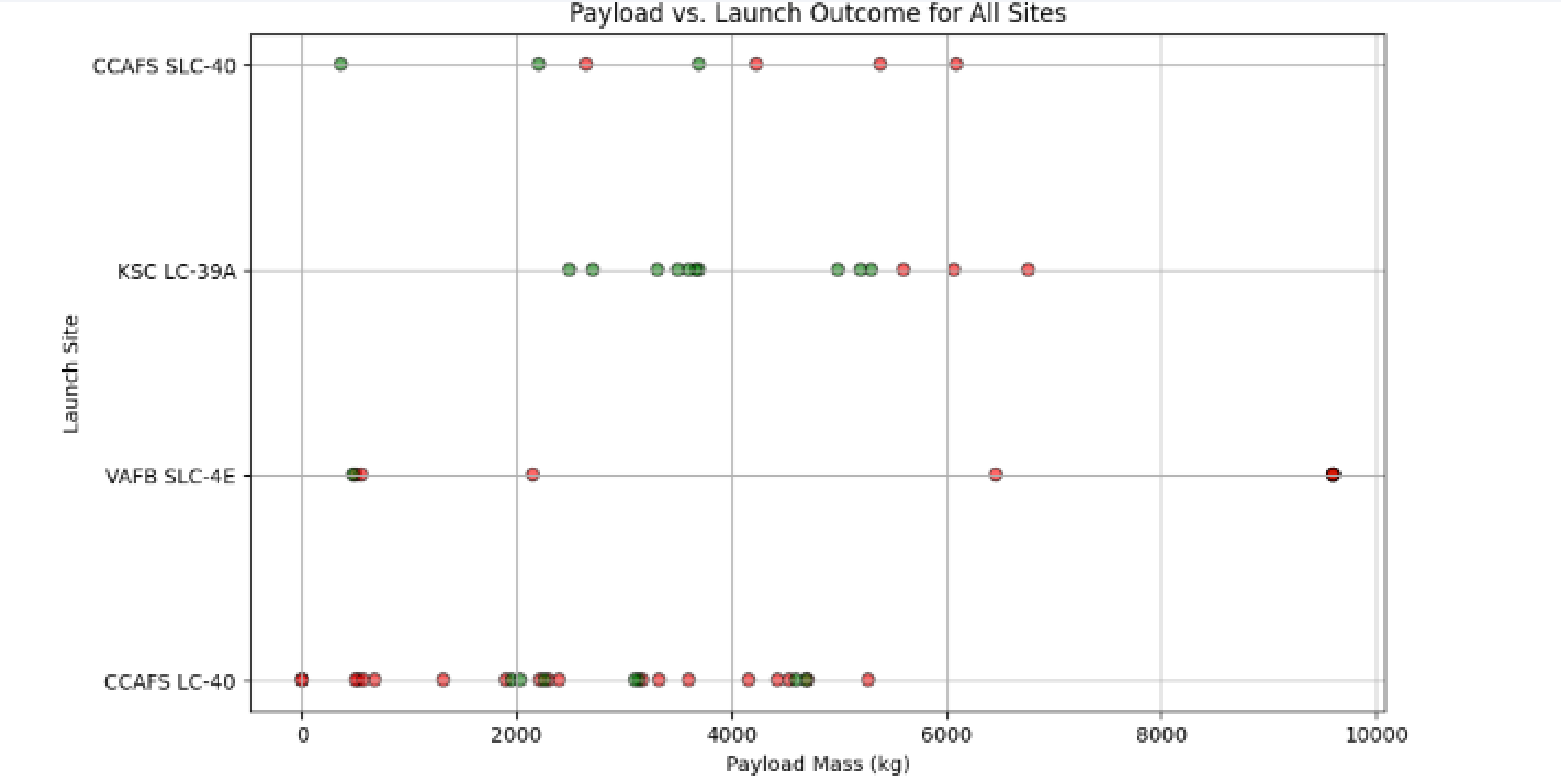
KSC LC-39A has the highest success rate
CCAFS SLC-40 has the lowest success rate

Statistics for KSC LC-39A



KSC LC-39A is the site with the highest success rate

Payload vs Launch Outcome

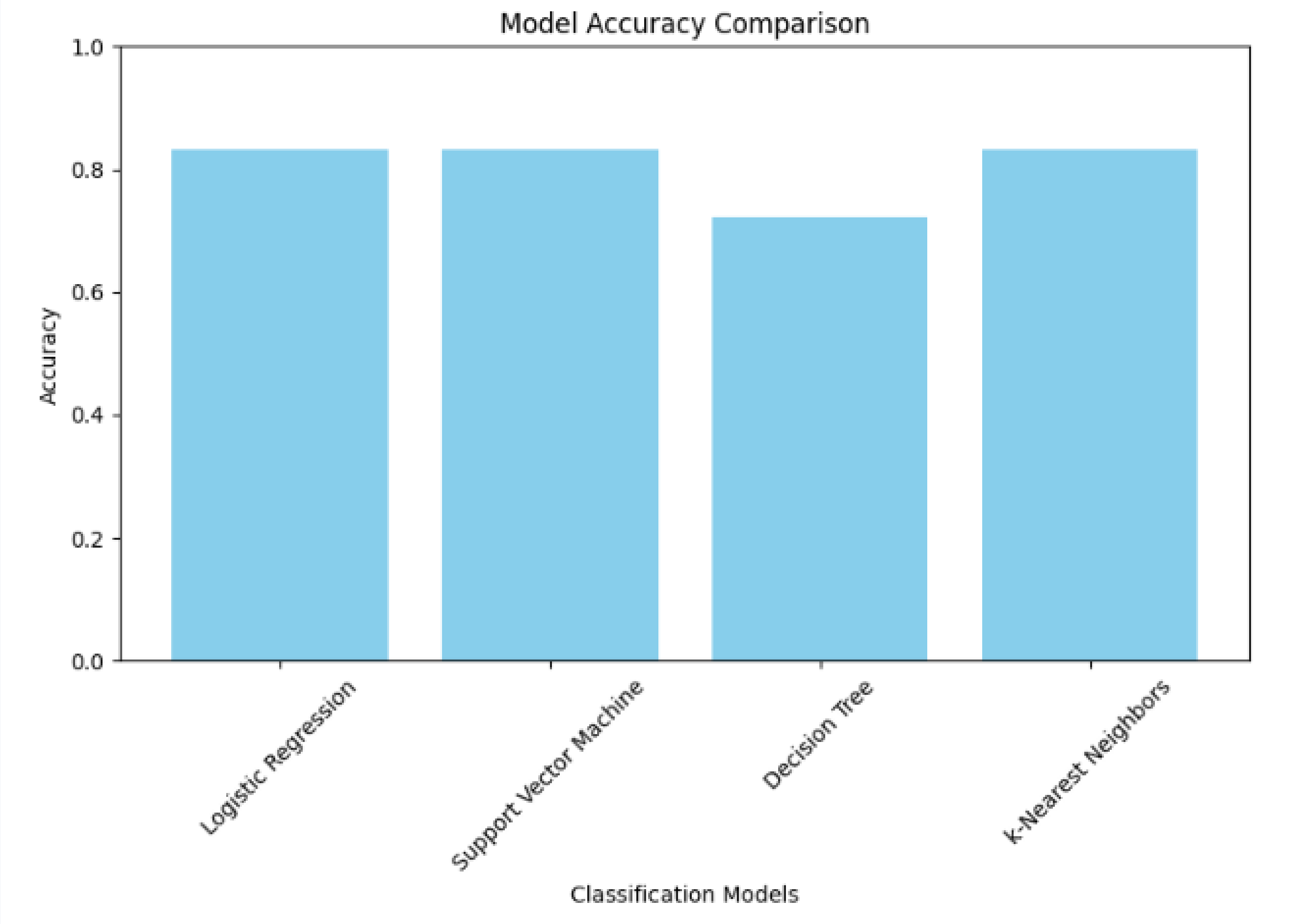


Section 5

Predictive Analysis (Classification)

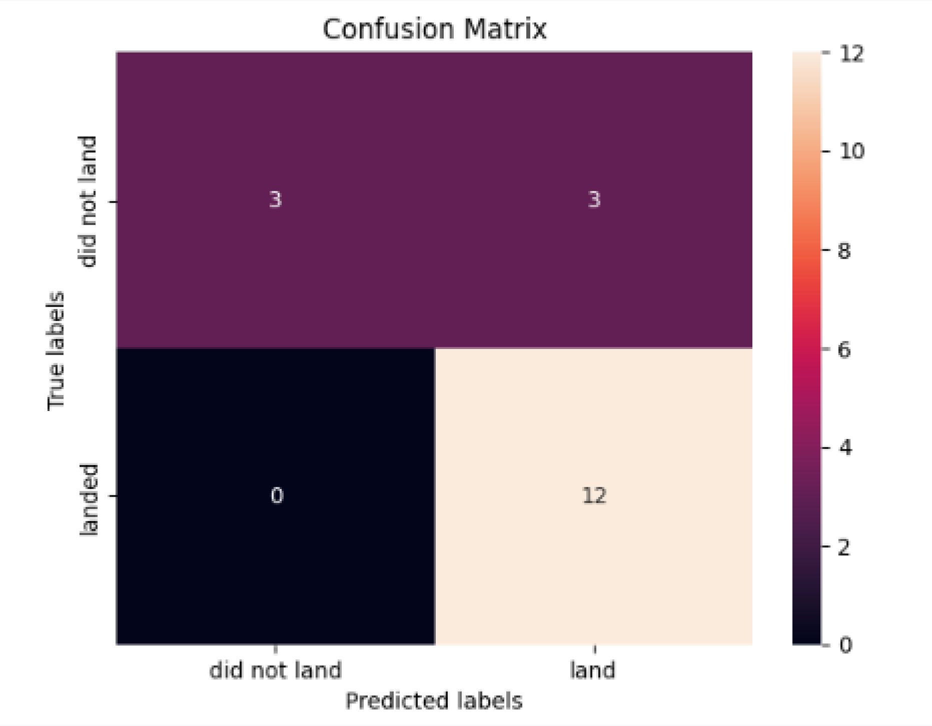
Classification Accuracy

- Logistic Regression Model has the highest classification Accuracy



Confusion Matrix

- Confusion matrix of the best performing model (Logistic Regression)
- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the problem is false positives.
- True Positive - 12 (True label is landed, Predicted label is also landed)
- False Positive - 3 (True label is not landed, Predicted label is landed)



Conclusions

- Launch Success rate over 60% since 2016
- First successful landing on a ground pad at 12/12/2015
- Successful missions: 60
- Failed missions: 30
- KSC LC-39A is the site with the highest success rate

Appendix

Python Libraries/Packages that were used:

- numpy
- pandas
- seaborn
- matplotlib
- sklearn
- requests
- piplite

Thank you!