Winning Space Race with Data Science

Puiching Lui
07/31/2023

Outline

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

Executive Summary

- We use SpaceX existing launch data to predict whether SpaceX will attempt to land a rocket next.
  - Previous results of Falcon 9 rockets are used
- There are multiple factors that can affect if a rocket will land or not. We will go through them and see which factors are important to determine if the next rocket will land.

Introduction

1. Gather information from previous rocket launches at SpaceX
2. Wrangle data for Falcon 9
3. Analyze data using visualization with Python and SQL
4. Provide interactive visual analytics with Folium and Plotly Dash
5. Predict if SpaceX will land the next rocket successfully

Section 1

Methodology

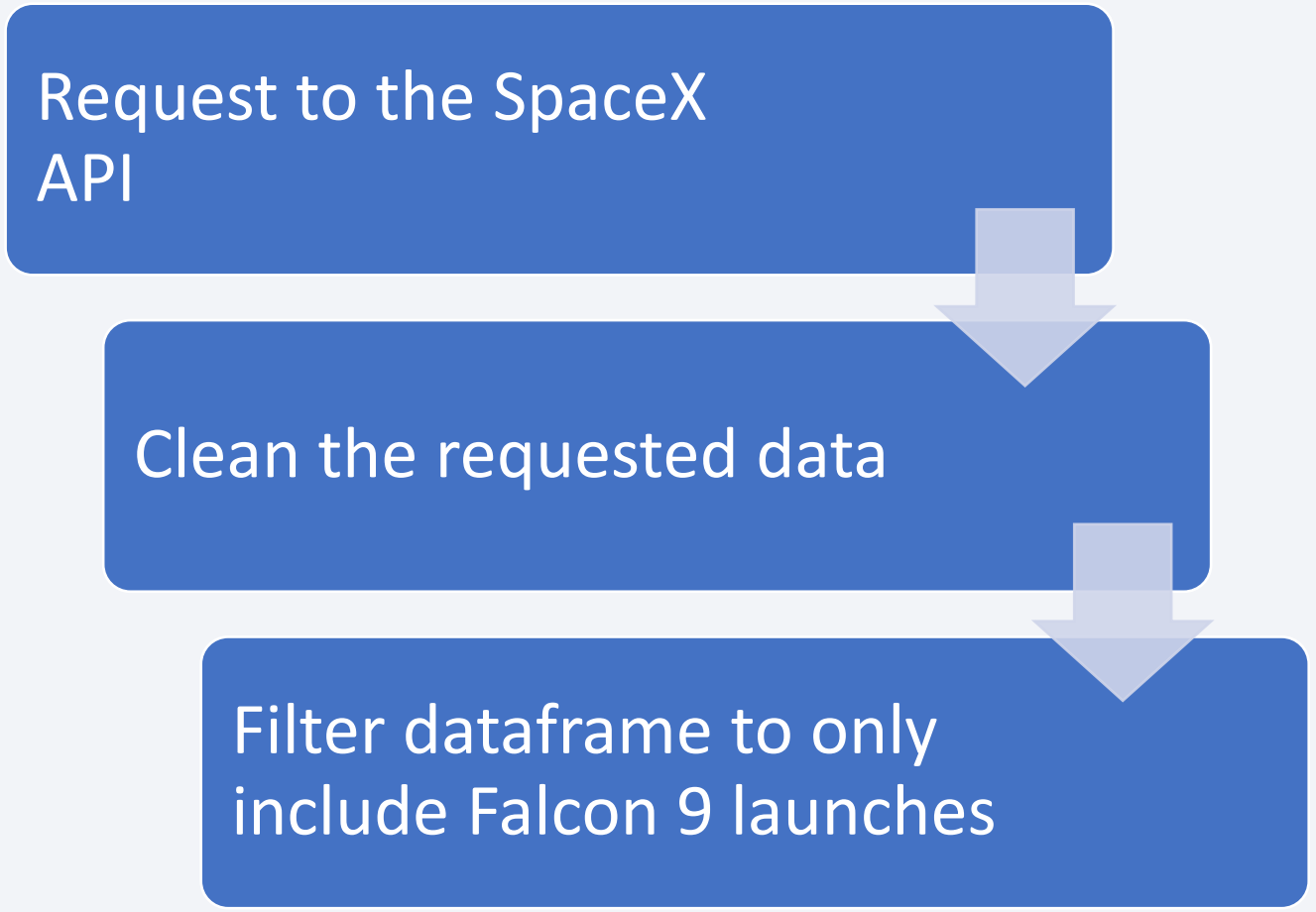
Methodology

Executive Summary

- SpaceX launch data collected in Wikipedia for Falcon 9, which is gathered from the SpaceX REST API.
- Perform data wrangling in Python to find the success rate of previous launches
- Perform exploratory data analysis (EDA) using visualization in Python and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - We use this data to predict whether SpaceX will attempt to land a rocket or not.

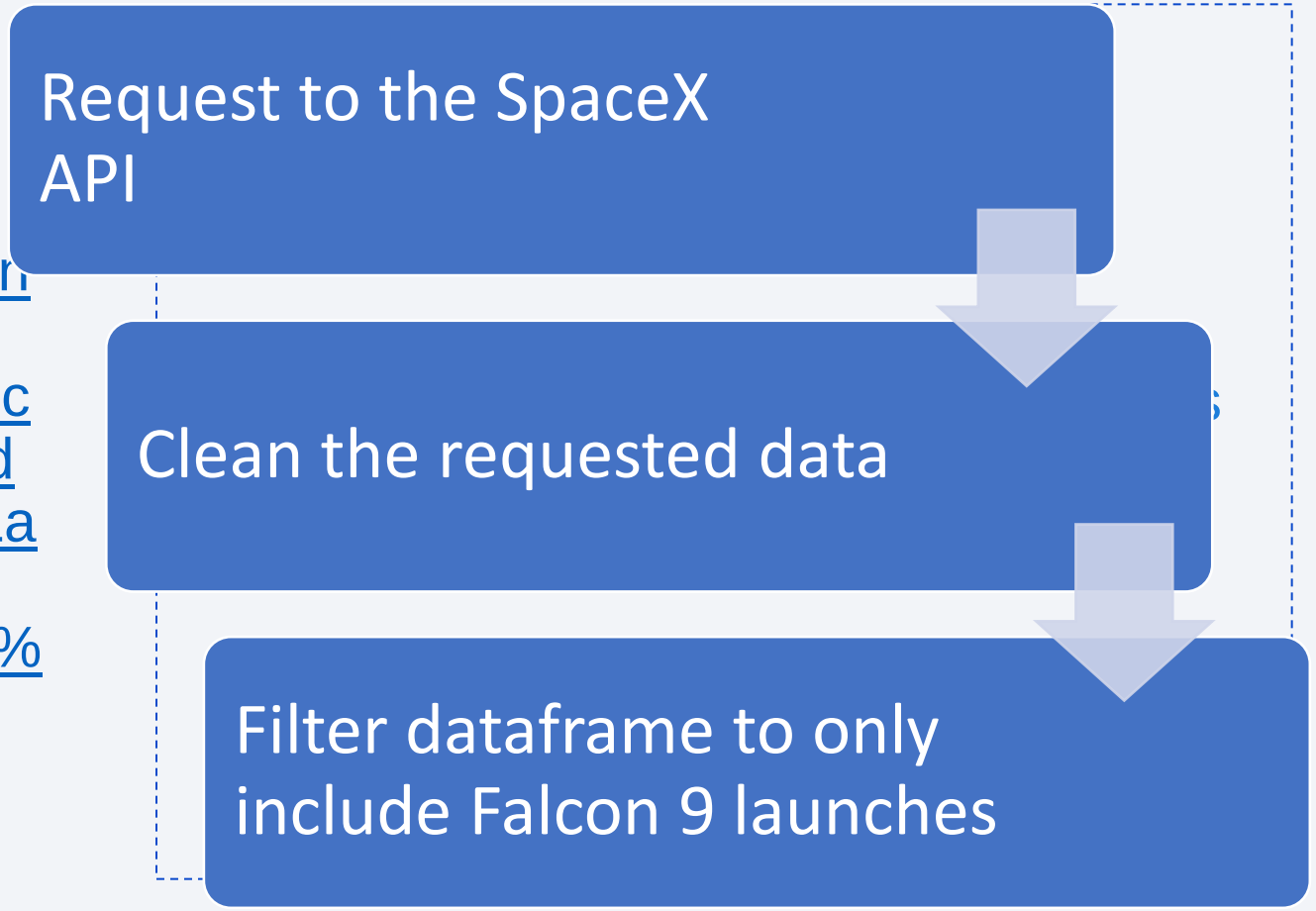
Data Collection

- Data is collected from Wikipedia
- Filtered out Falcon 1 Booster Version



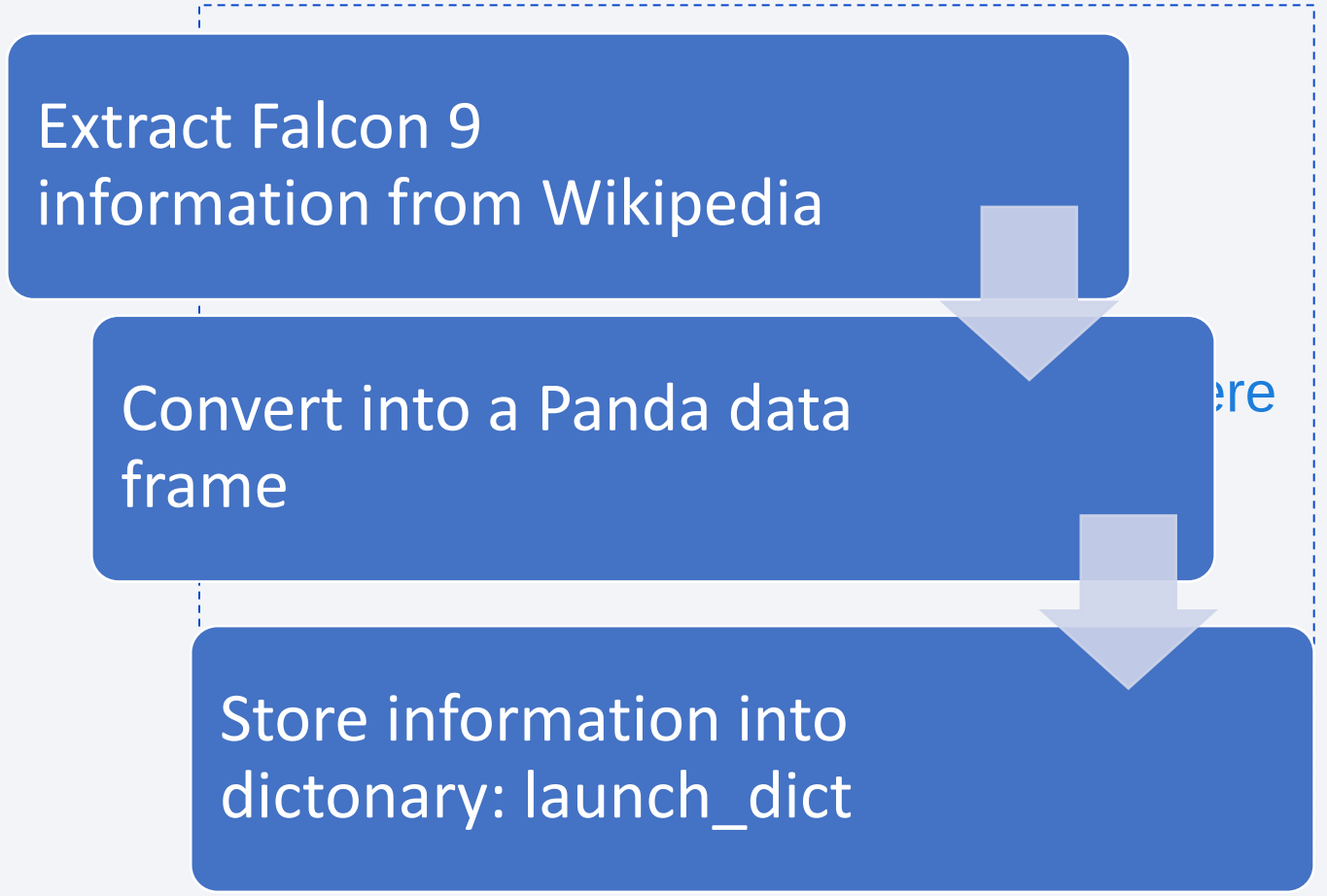
Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%201%20Hands%20on%20Lab%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab%20jupyter-labs-webscraping.ipynb



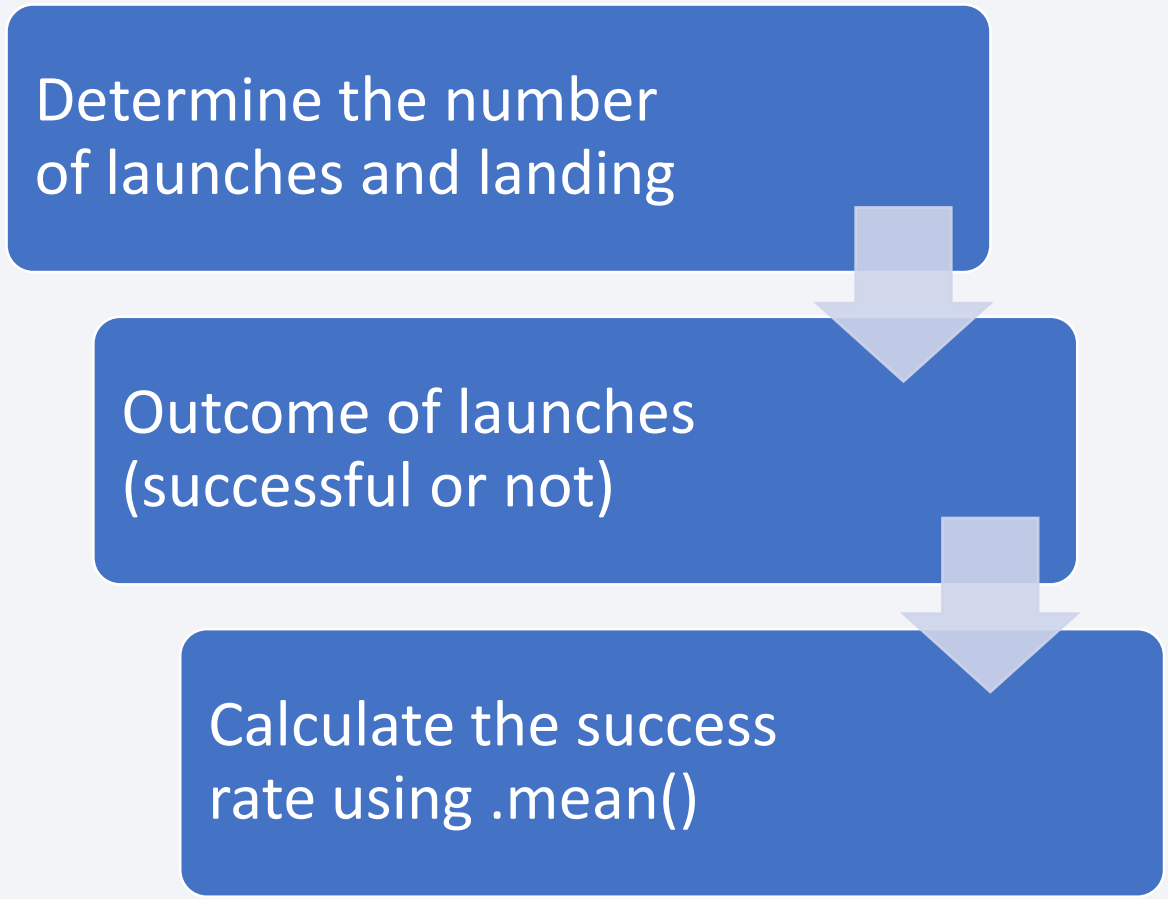
Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%201%20Hands%20on%20Lab%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab%20jupyter-labs-webscraping.ipynb



Data Wrangling

- Describe how data were processed
  - We use data from previous launches to find success rate, in order to predict for the next launch.
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%201%20Hands%20on%20Lab%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab%20jupyter-labs-webscraping.ipynb



EDA with Data Visualization

- First, we've used scatter point charts for relationship of different variables affecting the launch outcome to show large quantities of data. Then, we've used bar charts to visualize success rate by orbits, because it can summarize large complex data into an easy visual format to understand. Finally, we've used line chart to visualize the progress over years because it can show changes and trends.
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/Module%202%20Hands%20on%20Lab%20Complete%20the%20EDA%20with%20Visualization%20jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- There are 4 launch sites:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
- Total payload mass carried by NASA (CRS) boosters: 45596.0 kg
- Avg payload mass carried by F9 v1.1 boosters: 2928.4 kg
- The date where the successful landing outcome in drone ship was achieved: 04/08/2016.

EDA with SQL

- The names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000:

- 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 FT B1032.2
- F9 B4 B1040.2
- F9 B5 B1046.2
- F9 B5 B1047.2
- F9 B5 B1048.3
- F9 B5 B1051.2
- F9 B5B1060.1
- F9 B5 B1058.2
- F9 B5B1062.1

EDA with SQL

- Successful mission outcomes: 98
- Failure mission outcomes: 1
- Booster_versions with maximum payload mass:
  - '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 '.

EDA with SQL

- The records display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
  - ('February', 'Success (ground pad)', 'F9 FT B1031.1', 'KSC LC-39A')
  - ('January', 'Success (ground pad)', 'F9 FT B1032.1', 'KSC LC-39A')
  - ('March', 'Success (ground pad)', 'F9 FT B1035.1', 'KSC LC-39A')
  - ('August', 'Success (ground pad)', 'F9 B4 B1039.1', 'KSC LC-39A')
  - ('July', 'Success (ground pad)', 'F9 B4 B1040.1', 'KSC LC-39A')
  - ('December', 'Success (ground pad)', 'F9 FT B1035.2', 'CCAFS SLC-40')

EDA with SQL

- 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:
  - ('Success', 20)
  - ('No attempt', 10)
  - ('Success (drone ship)', 8)
  - ('Success (ground pad)', 7)
  - ('Failure (drone ship)', 3)
  - ('Failure', 3)
  - ('Failure (parachute)', 2)
  - ('Controlled (ocean)', 2) ('No attempt ', 1)
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%202%20Hands%20on%20Lab%20Complete%20the%20EDA%20with%20SQL%20jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

- I created markers and circles for the four launch sites to show their location on the map. The color green and red are added to show if the launch happened was successful or not. I also added lines to a folium map to show their proximity to shore and to local railway.
- We noticed that all launch sites are in proximity to the Equator line. Also, they are in very close proximity to the coast. Finally, the sites keep certain distance away from cities, due to noise and safety reasons.
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%203%20Hands%20on%20Lab%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- We built pie charts to show the success rate for all 4 sites to see which site has the highest success rate.
- We included a range slider to select Payload and also built scatter plots in reference to payload mass, because we've seen in previous models that Payload between 4k and 6k can increase success rate.
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/module%203%20Hands%20on%20Lab%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.jupyterlite.ipynb

Predictive Analysis (Classification)

- In this project, we built, evaluated, improved, and found the best performing classification model
- https://github.com/puijann/Machine-Learning-Capstone-Project/blob/250c8b403b3dd0a9c268c4e67f9532cbea82407b/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

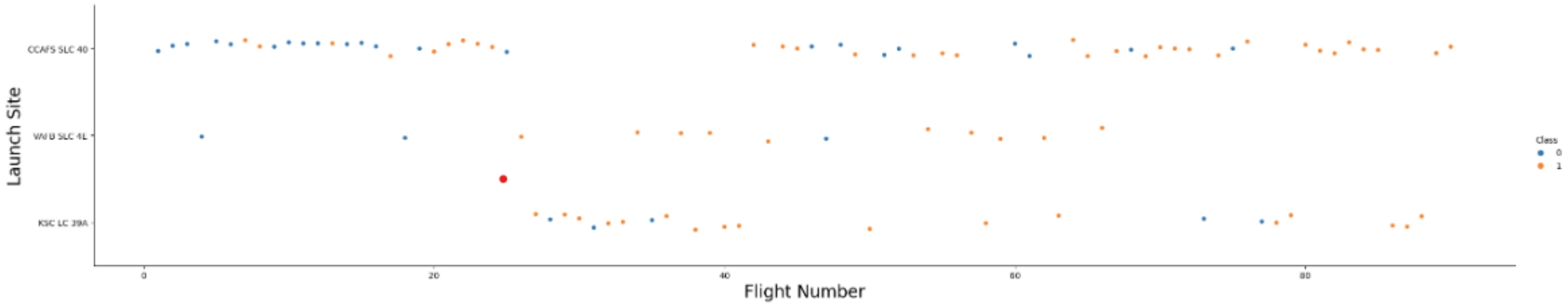
- The overall success rate kept increasing since 2013 till 2020.
- The rocket will be more likely to land if:
  - Payload mass between 4k and 6k.
  - Launch at site KSC
  - Orbit type of ES-L1, GEO, HEO, and SSO
  - No direct relationship observed with flight number, and booster

Section 2

Insights drawn from EDA

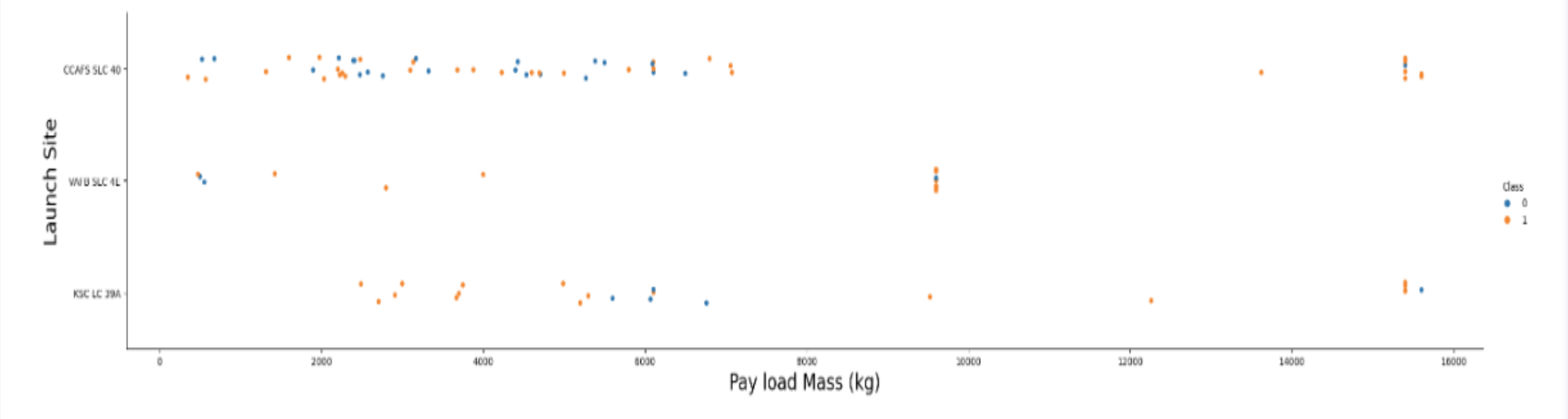
Flight Number vs. Launch Site

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- The success rate increases with larger flight number. For example, a success rate of 100% is observed with flight number of 80 or higher at all sites.



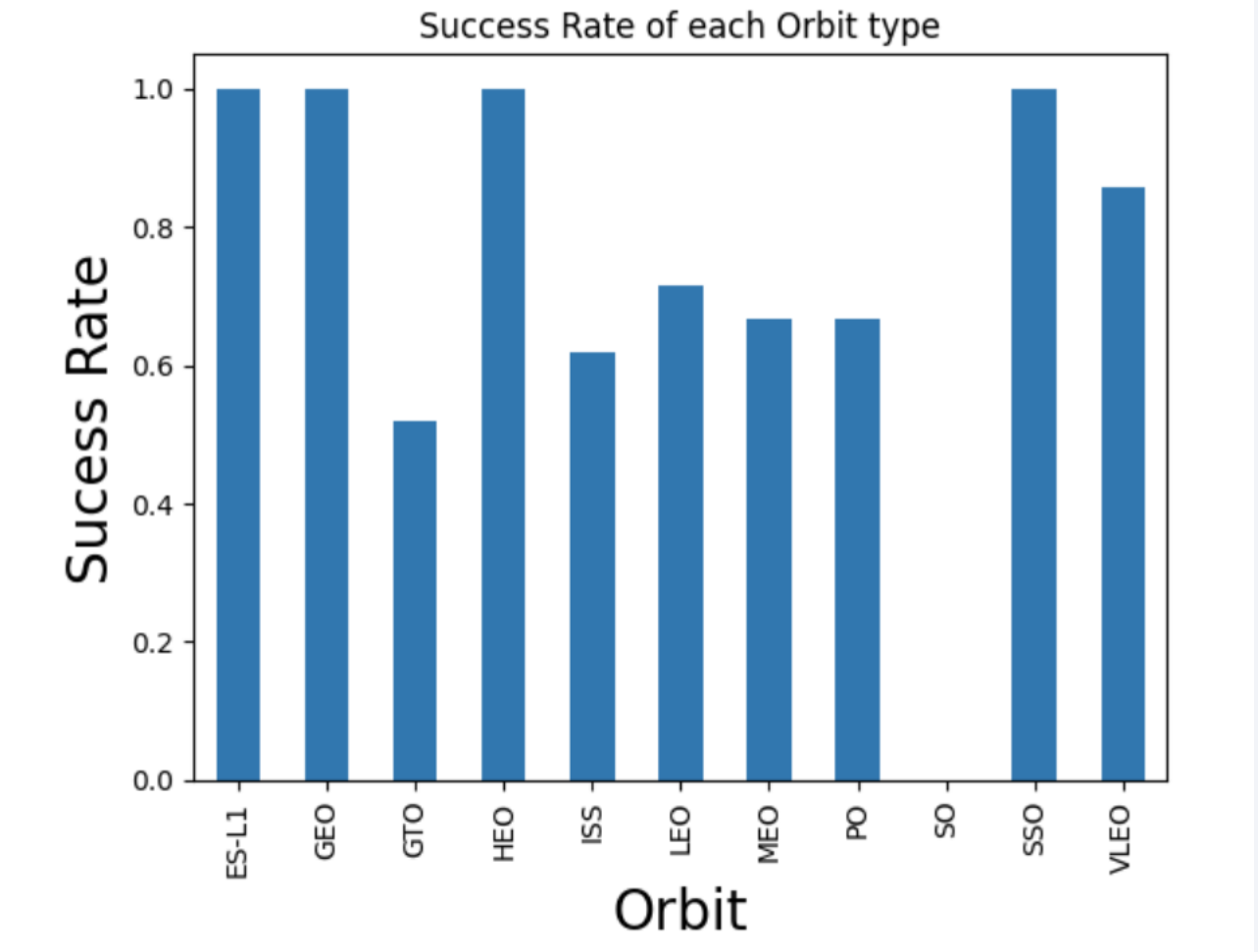
Payload vs. Launch Site

- Payload Vs. Launch Site scatter point chart shows you the VAFB-SLC launch site did not launch rockets for heavypayload mass(greater than 10000).



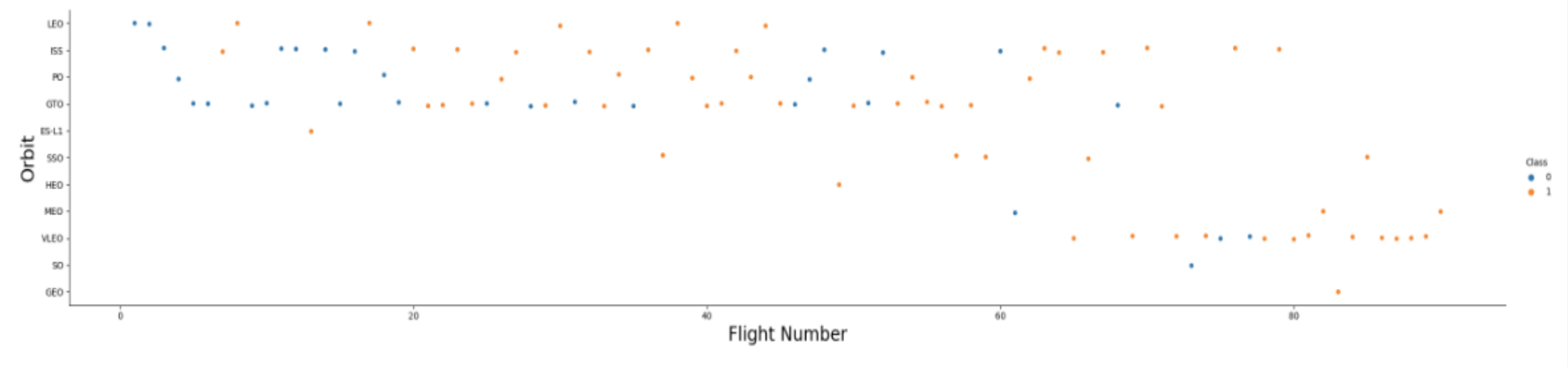
Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO have high success rate of 100% where as orbit SO has low success rate of 0%.



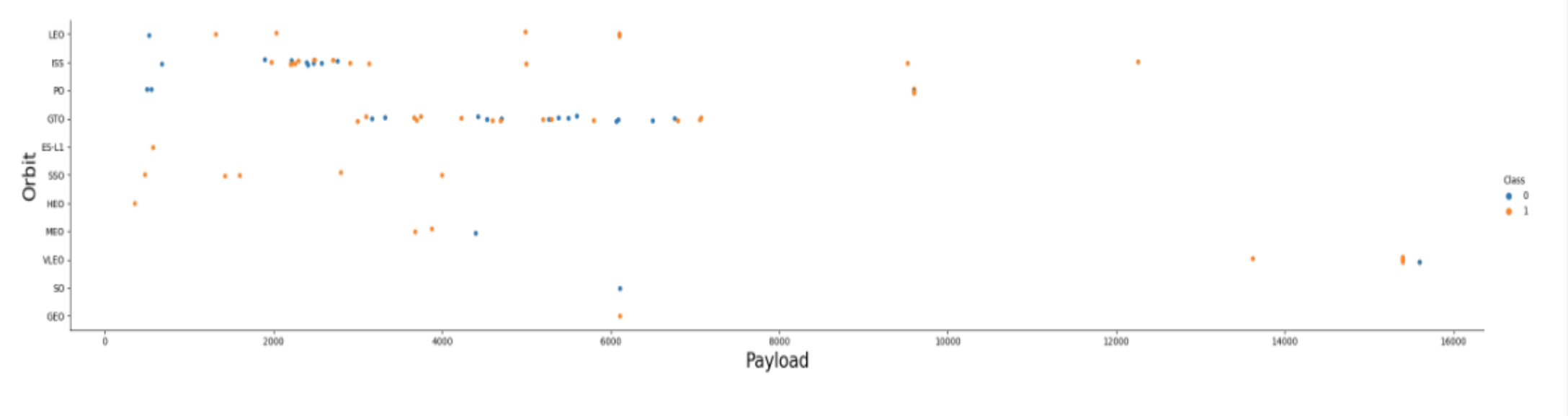
Flight Number vs. Orbit Type

- In the LEO orbit, the success appears related to the number of flights
- However, there seems to be no relationship between flight number when in GTO orbit.



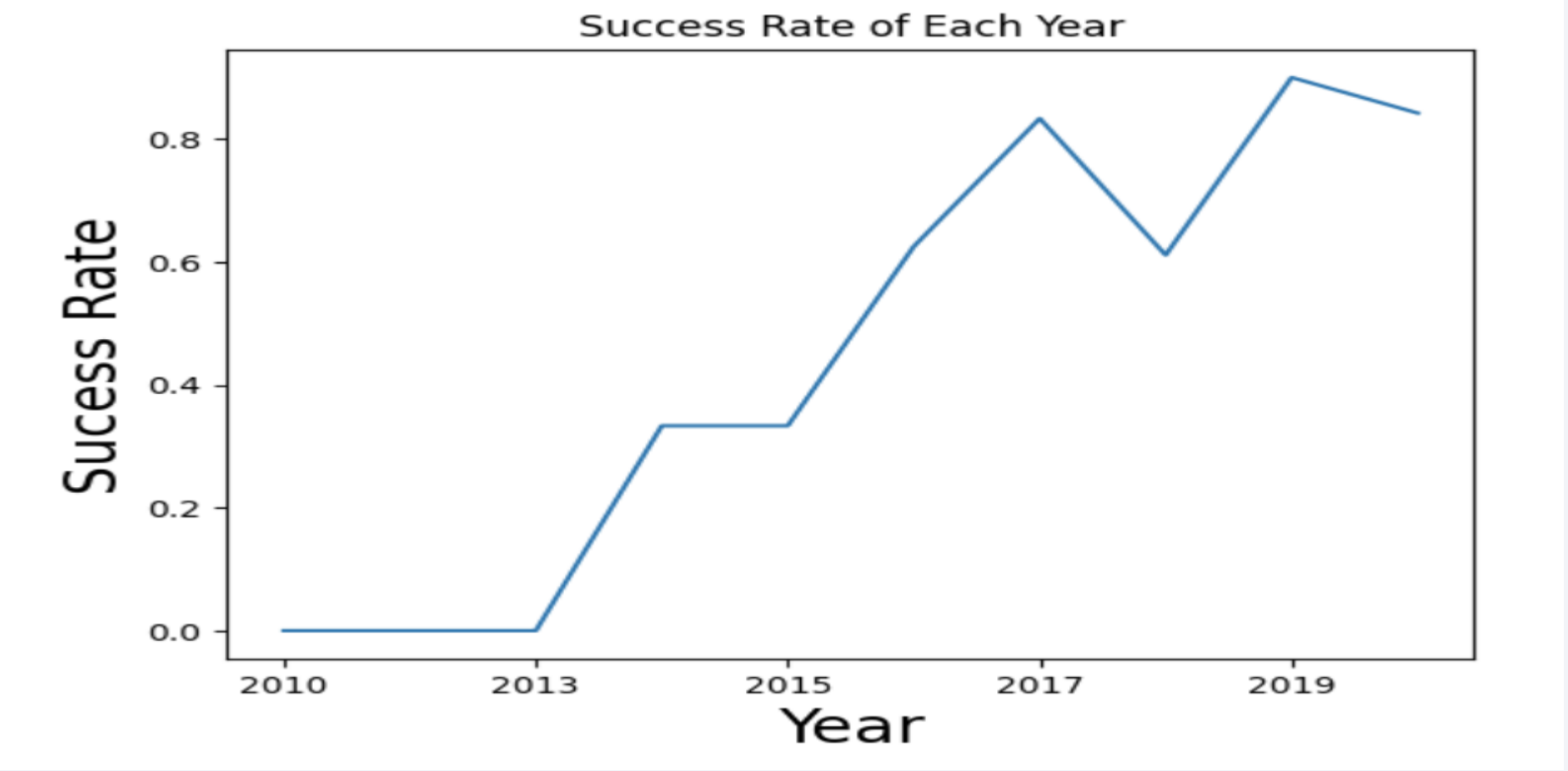
Payload vs. Orbit Type

- With heavy payloads the successful landing rate are higher for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there.



Launch Success Yearly Trend

- The success rate kept increasing from 2013 to 2020



All Launch Site Names

- Using SQL, we can find the names of the unique launch sites:

```
# Execute the SQL query
    cur.execute("SELECT DISTINCT launch_site FROM SPACEXTBL

# Fetch all the unique launch site names
    launch_sites = cur.fetchall()

# Print the results
    for site in launch_sites:
        print(site[0])
```

```
CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

None
```

Launch Site Names Begin with 'KSC'

```
# Execute the SQL query
cur.execute("SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'KSC%' LIMIT 5")
# Fetch the 5 records where launch sites begin with 'KSC'
records = cur.fetchall()
# Print the results
for record in records:
    print(record)
```

```
('19/02/2017', '14:39:00', 'F9 FT B1031.1', 'KSC LC-39A', 'SpaceX CRS-10', 2490.0, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'Success (ground pad)')
('16/03/2017', '6:00:00', 'F9 FT B1030', 'KSC LC-39A', 'EchoStar 23', 5600.0, 'GTO', 'EchoStar', 'Success', 'No attempt')
('30/03/2017', '22:27:00', 'F9 FT  B1021.2', 'KSC LC-39A', 'SES-10', 5300.0, 'GTO', 'SES', 'Success', 'Success (drone ship)')
('05/01/2017', '11:15:00', 'F9 FT B1032.1', 'KSC LC-39A', 'NROL-76', 5300.0, 'LEO', 'NRO', 'Success', 'Success (ground pad)')
('15/05/2017', '23:21:00', 'F9 FT B1034', 'KSC LC-39A', 'Inmarsat-5 F4', 6070.0, 'GTO', 'Inmarsat', 'Success', 'No attempt')
```

Total Payload Mass

```
# Execute the SQL query
cur.execute("SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM
    SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'")

# Fetch the total payload mass
total_payload_mass = cur.fetchone()[0]

# Print the result
print("Total payload mass carried by NASA (CRS) boosters:", total_payload_mass, "kg")
```

Result:

```
Total payload mass carried by NASA (CRS) boosters: 45596.0 kg
```

Average Payload Mass by F9 v1.1

```
# Execute the SQL query
cur.execute("SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM
    SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'")

# Fetch the AVG payload mass
AVG_PAYLOAD_MASS = cur.fetchone()[0]

# Print the result
print("Avg payload mass carried by F9 v1.1 boosters:", AVG_PAYLOAD_MASS, "kg")
```

Result: Avg payload mass carried by F9 v1.1 boosters: 2928.4 kg

First Successful Ground Landing Date

```
# Execute the SQL query
cur.execute("SELECT MIN(date) AS EARLIEST_LANDING_DATE FROM SPACEXTBL
    WHERE LANDING_OUTCOME = 'Success (drone ship)'")

# Fetch the total payload mass
EARLIEST_LANDING_DATE = cur.fetchone()[0]

# Print the result
print("DATE:", EARLIEST_LANDING_DATE)
```

Result: DATE: 04/08/2016

Successful Drone Ship Landing with Payload 4000 to 6000

```
#Execute the SQL query
cur.execute("SELECT DISTINCT BOOSTER_VERSION FROM
    SPACEXTBL WHERE mission_outcome = 'Success' AND
    payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000")

# Fetch all the rows (records) from the query result
booster_names = cur.fetchall()

# Print the names of the boosters
for name in booster_names:
    print(name[0])
```

```
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 FT  B1032.2
F9 B4  B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
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

```
#Calculate the sum of successful and failure missions
success_count = df[df["Mission_Outcome"] == "Success"].shape[0]
failure_count = df[df["Mission_Outcome"] == "Failure (in flight)"].shape[0]

print("Successful mission outcomes:",success_count)
print("Failure mission outcomes:",failure_count)
```

Result:

```
Successful mission outcomes: 98

Failure mission outcomes: 1
```

Boosters Carried Maximum Payload

```
# Find the maximum payload mass

max_payload_mass = df["PAYLOAD_MASS__KG_"].max()


# Filter the DataFrame to get the booster_versions with the maximum payload mass

max_payload_booster_versions = df[df["PAYLOAD_MASS__KG_"] == max_payload_mass]["Booster_Version"].tolist()


print("Booster_versions with maximum payload mass:", max_payload_booster_versions)
```

Result:

```
Booster_versions with maximum payload mass: ['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 ']
```

2017 Launch Records

 # Execute the query to list the records which will display the month names, succesful landing_outcomes in ground pad, booster versions, launch_site for the months in year 2017

```
query = """
    SELECT
        CASE
            WHEN substr(Date, 4, 2) = '01' THEN 'January'
            WHEN substr(Date, 4, 2) = '02' THEN 'February'
            WHEN substr(Date, 4, 2) = '03' THEN 'March'
            WHEN substr(Date, 4, 2) = '04' THEN 'April'
            WHEN substr(Date, 4, 2) = '05' THEN 'May'
            WHEN substr(Date, 4, 2) = '06' THEN 'June'
            WHEN substr(Date, 4, 2) = '07' THEN 'July’
```

2017 Launch Records (continue)

```
        WHEN substr(Date, 4, 2) = '08' THEN 'August'
        WHEN substr(Date, 4, 2) = '09' THEN 'September'
        WHEN substr(Date, 4, 2) = '10' THEN 'October'
        WHEN substr(Date, 4, 2) = '11' THEN 'November'
        WHEN substr(Date, 4, 2) = '12' THEN 'December'
    END AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTBL
WHERE substr(Date, 7, 4) = '2017'
AND Landing_Outcome = 'Success (ground pad)'
"""
```

2017 Launch Records (continue)

```
# Fetch and display the result
cursor = con.execute(query)
result = cursor.fetchall()
for row in result:
    print(row)
```

Result:

```
('February', 'Success (ground pad)', 'F9 FT B1031.1', 'KSC LC-39A')
('January', 'Success (ground pad)', 'F9 FT B1032.1', 'KSC LC-39A')
('March', 'Success (ground pad)', 'F9 FT B1035.1', 'KSC LC-39A')
('August', 'Success (ground pad)', 'F9 B4 B1039.1', 'KSC LC-39A')
('July', 'Success (ground pad)', 'F9 B4 B1040.1', 'KSC LC-39A')
('December', 'Success (ground pad)', 'F9 FT  B1035.2', 'CCAFS SLC-40')
```

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

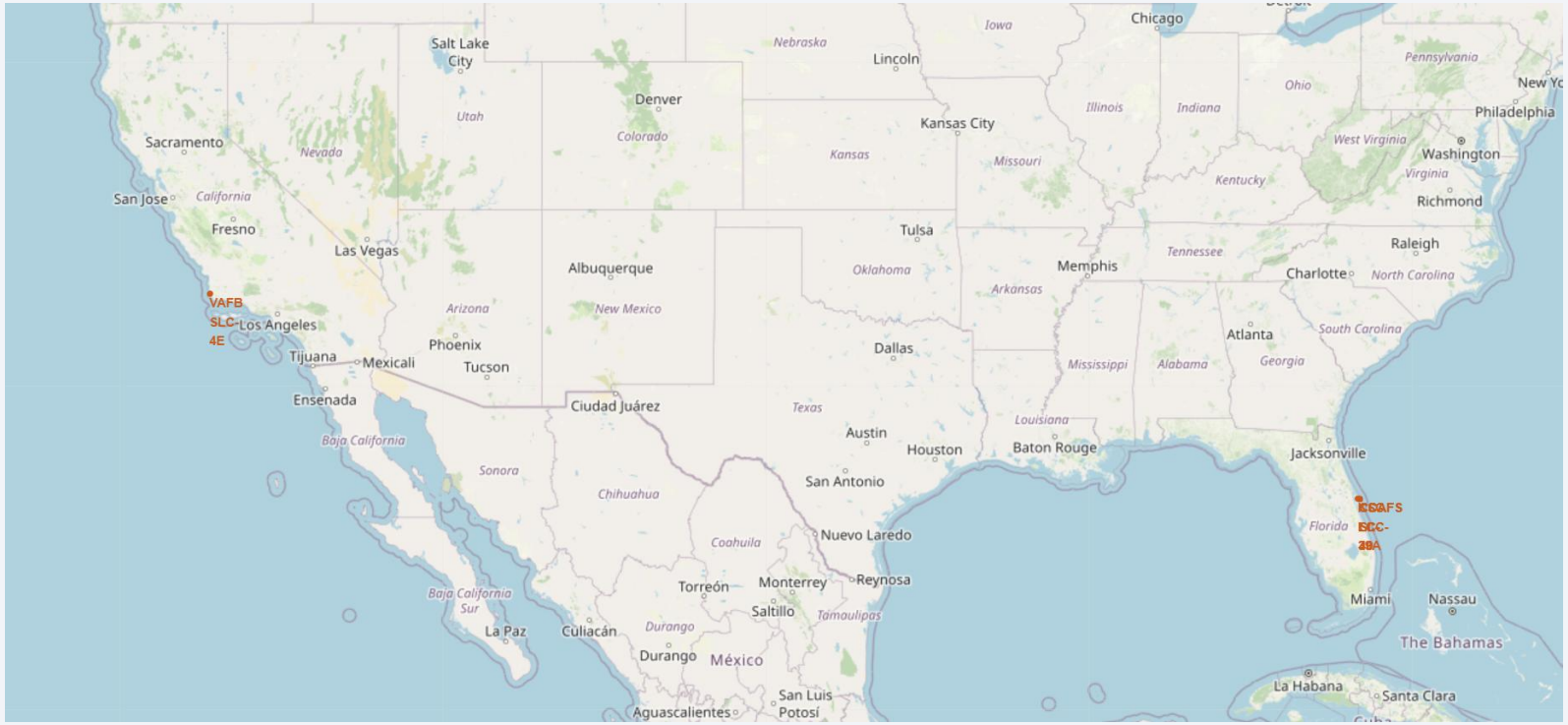
```
# Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))
query = """
    SELECT Landing_Outcome, COUNT(*) AS Outcome_Count
    FROM SPACEXTBL
    WHERE Date BETWEEN '04-06-2010' AND '20-03-2017'
    GROUP BY Landing_Outcome
    ORDER BY Outcome_Count DESC
"""“
cursor = con.execute(query)
result = cursor.fetchall()
for row2 in result:
    print(row2)
```

```
('Success', 20)
('No attempt', 10)
('Success (drone ship)', 8)
('Success (ground pad)', 7)
('Failure (drone ship)', 3)
('Failure', 3)
('Failure (parachute)', 2)
('Controlled (ocean)', 2)
('No attempt ', 1)
```


Section 3
Launch Sites
Proximities Analysis

Folium map with all launch sites' location

- There are 1 site on west coast and 3 sites on east coast

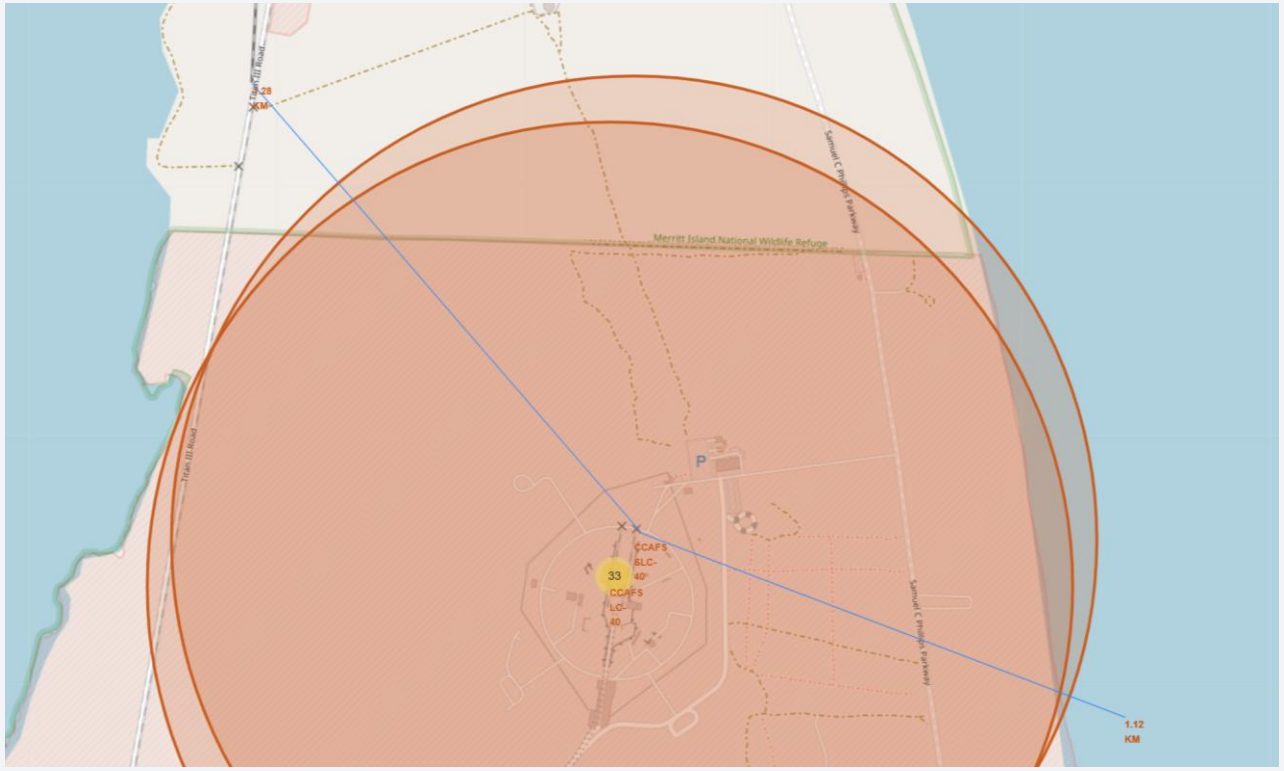


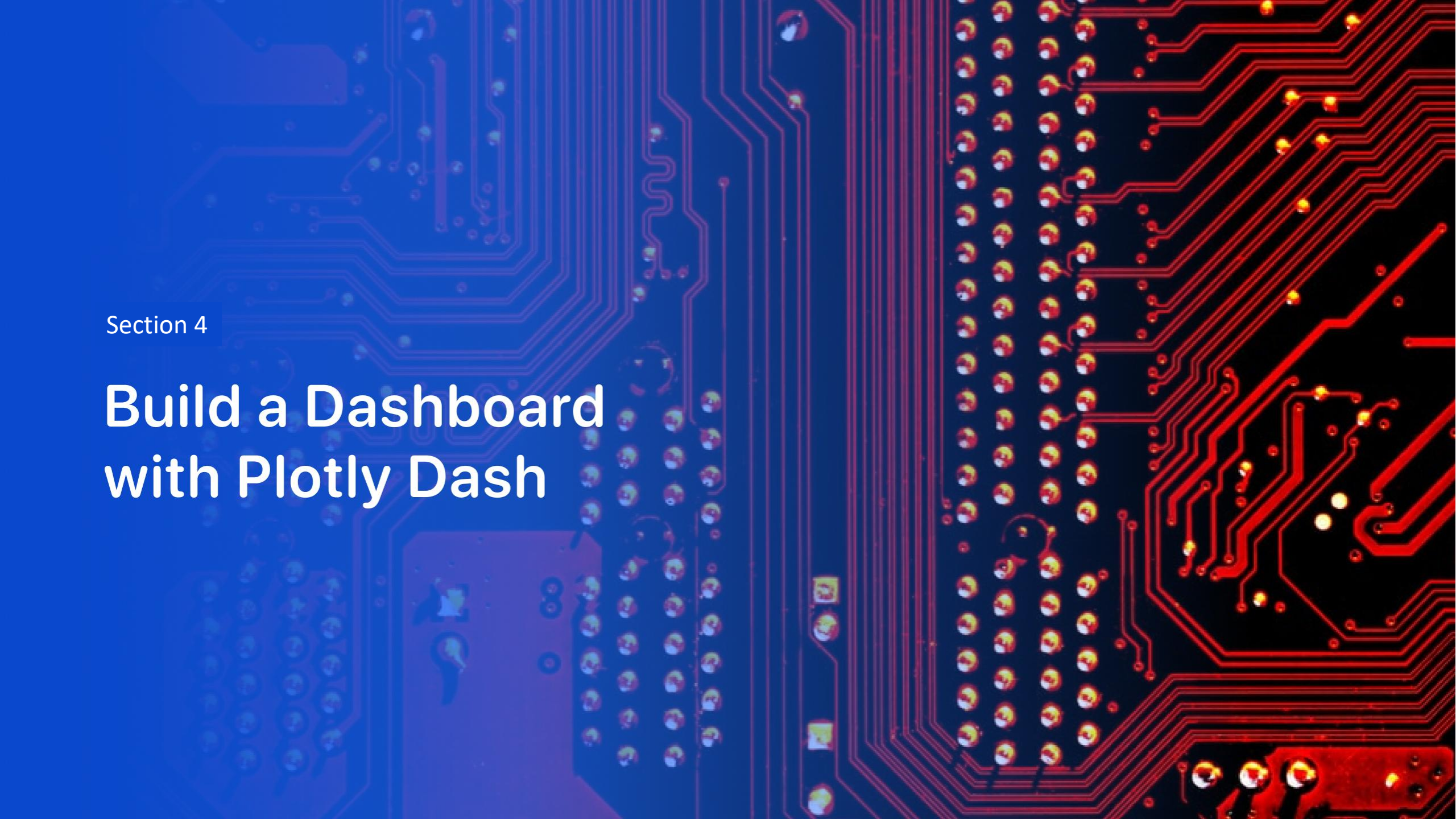
Folium map with color-labeled launch outcomes

- We can see the success rate here at site CCAFS SLC-40 is 43%

Distance from shore and railway

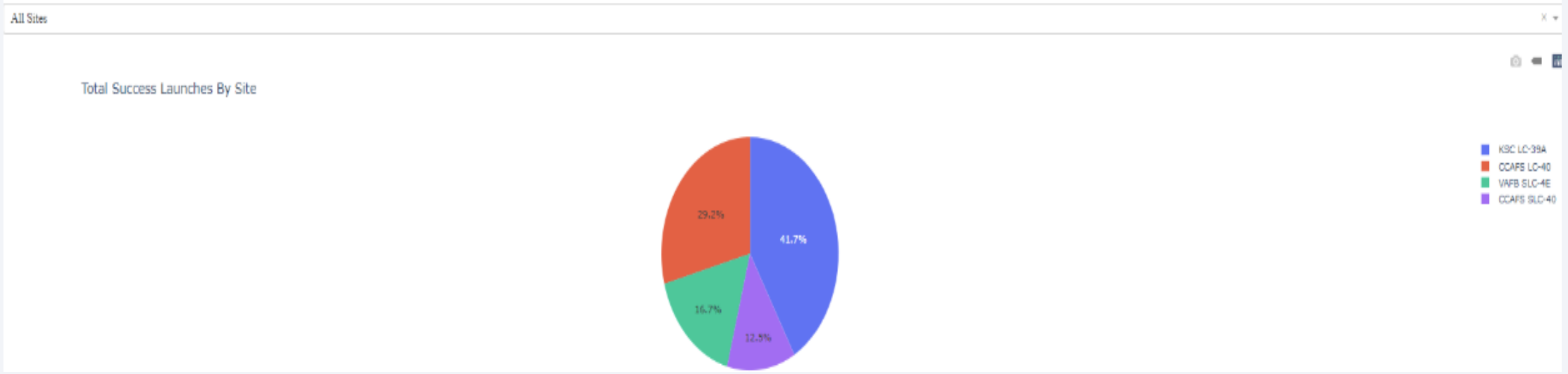
- Launch site is less than 1.12km from the shore
- Launch site is 1.28km from the closest railway
- Sites are relatively close to the shore for safe rocket testing



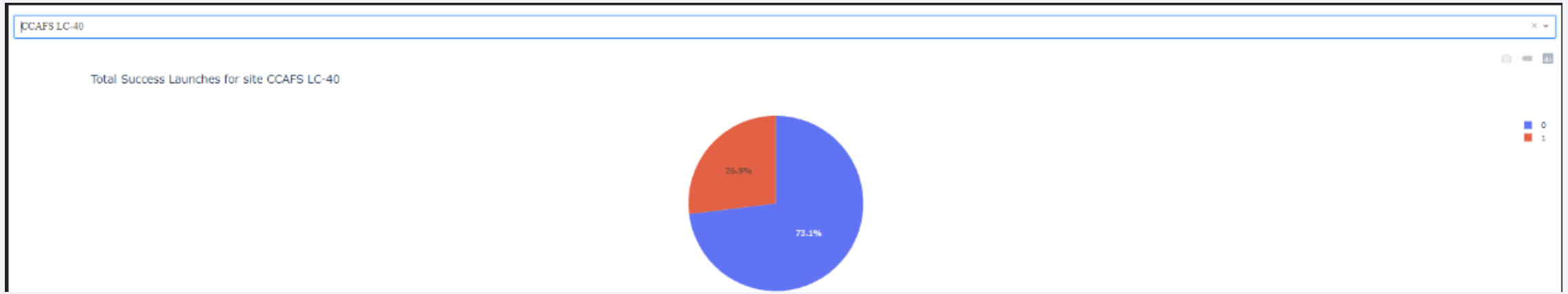

Section 4
Build a Dashboard with Plotly Dash

Launch success count for all sites

- Highest success rate: KSC LC-39A
- Lowest success rate: CCAFS SLC-40



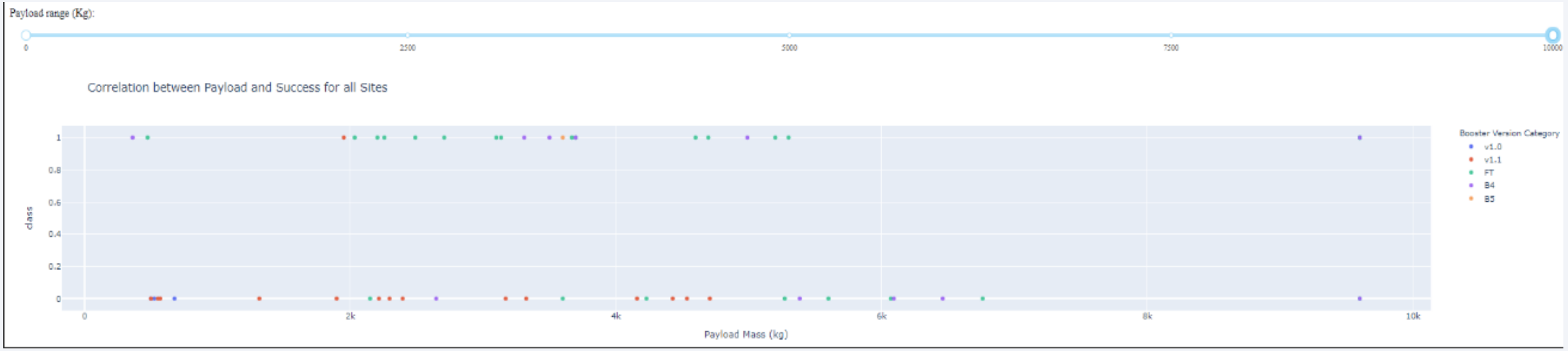
Launch site with highest launch success ratio



- The highest launch success ration is 73.1%

Payload vs. Launch Outcome scatter plots

- The payload range between 2k and 4k has the largest success rate.



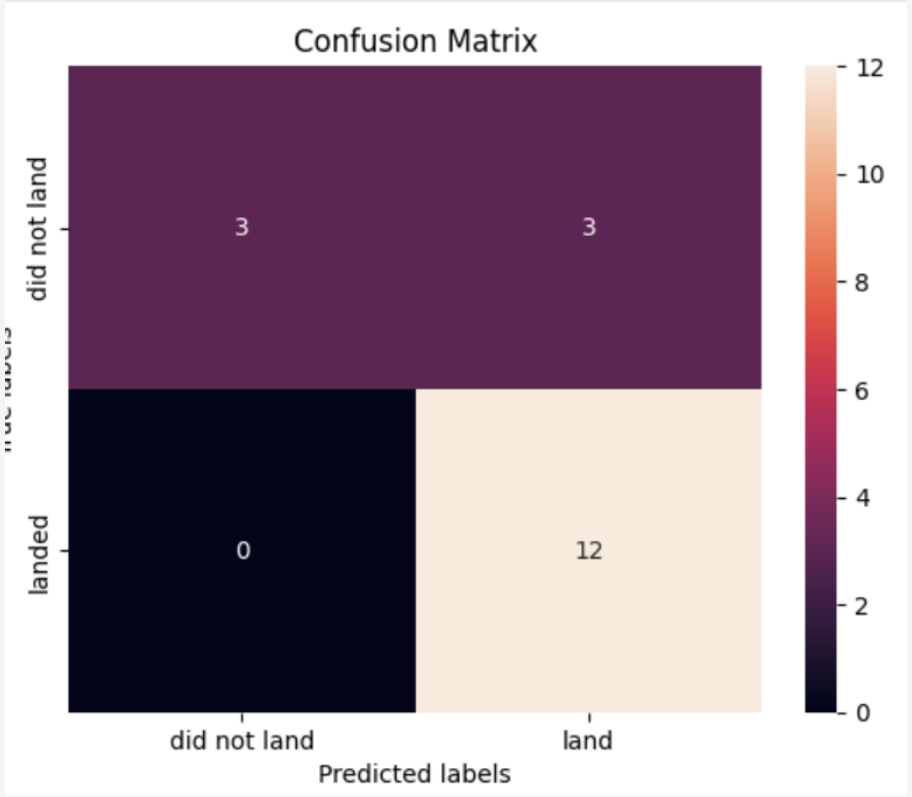
Section 5

Predictive Analysis (Classification)

Classification Accuracy

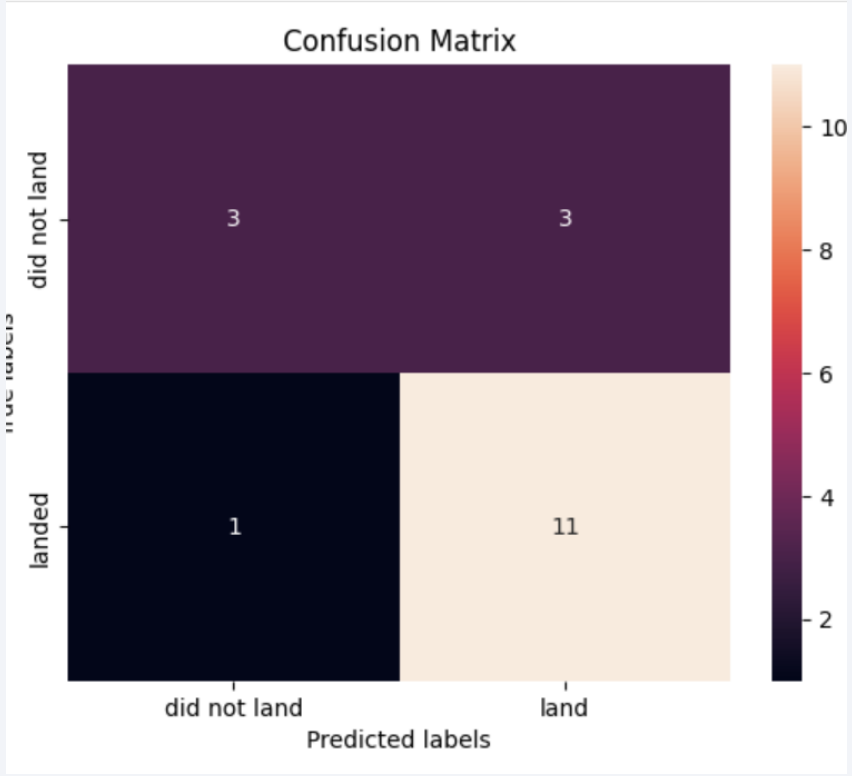
- The model with highest accuracy:

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



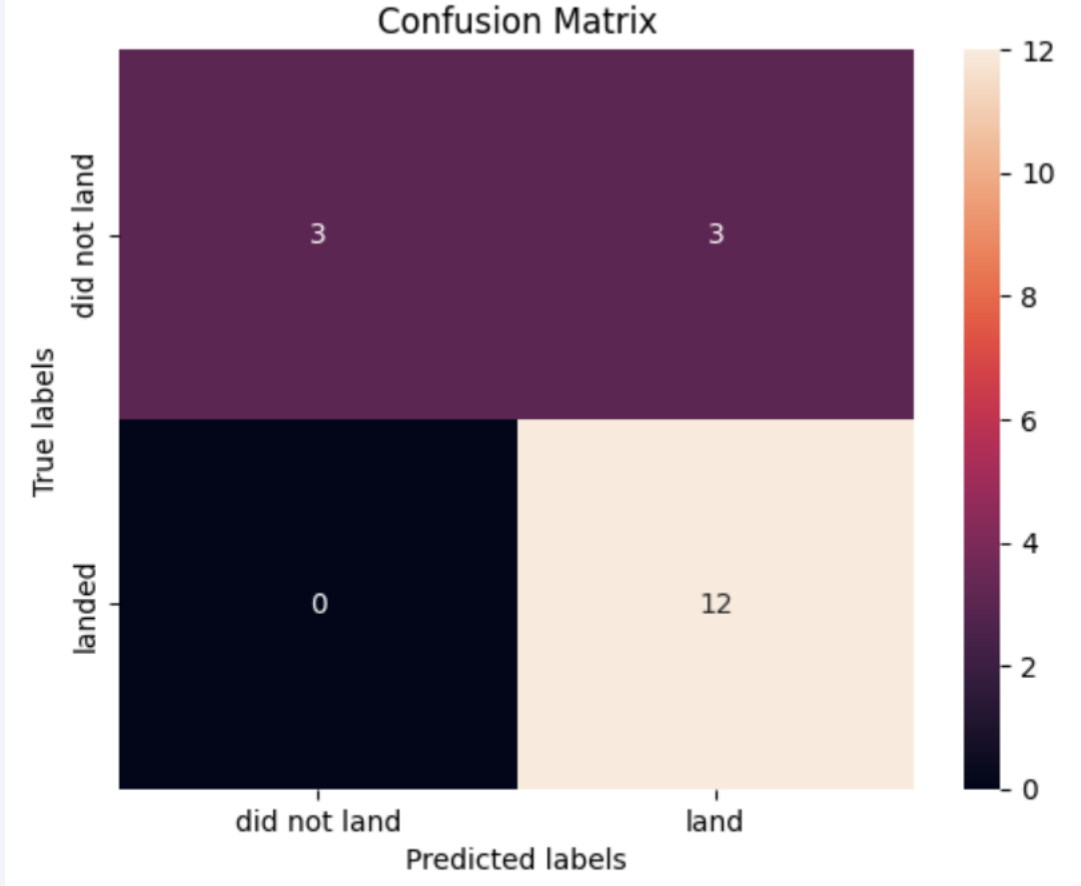
The Model with lowest accuracy:

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Confusion Matrix

- The best performing model is by using LR, SVM or KNN, where the accuracy is 83.3%
- The main problem is false positive: predicting the rocket to land but it failed in reality.



Conclusions

- Using confusion matrix,
  - 12 of the 18 were predicted to land and actually landed
  - 3 of the 18 were predicted to fail and actually fail.
  - 3 of the 18 were predicted to land but did not actually land
- Base on our result, the next rocket will most likely land.

Thank you!