IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michael S
7/14/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through SpaceX API and web scraping
  - Data wrangling to prepare for analysis
  - Exploratory data analysis with SQL & visualizations
  - Interactive data analysis with Folium maps & Plotly dashboard
  - Machine learning algorithm for predictions
- Summary of all results
  - Launch success rate has increased over time
  - Most launch sites are located in the southern US, on coasts, and away from cities
  - All predictive models performed similarly, with an 83% accuracy rate

Introduction

- Background
  - 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
- Explore
  - How do features, such as number of flights, launch site, payload mass, orbits, and boosters affect landing success
  - Find trend of successful landings over time
  - Identify best predictive model for future launches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and web scraping
- Perform data wrangling
  - Filtered data set as needed, addressed missing values, and applied one hot encoding for machine learning
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used GridSearchCV to find the optimum parameters, then compared models using accuracy score

Data Collection

- Data was collected from Space X API
- Additional data was scraped from Wikipedia
- Data was then filtered and wrangled to create a data set of Falcon 9 launches

Data Collection – SpaceX API

- GET request was used to retrieve launch data
- Returned JSON data was converted to Pandas data frame
- Key features were identified and compiled into a new data frame, then filtered to only include Falcon 9
- Lastly, missing payload mass values were filled in using the mean value method
- GitHub URL https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

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

[16]: response = requests.get(spacex_url)

[31]: # Use json_normalize meethod to convert the json result into a dataframe
      data = response.json()
      data = pd.json_normalize(data)

[68]: # Hint data['BoosterVersion']!='Falcon 1'
      data_falcon9 = launch_df[launch_df['BoosterVersion'] != 'Falcon 1'].copy()

[77]: # Calculate the mean value of PayloadMass column
      payload_mass_mean = data_falcon9['PayloadMass'].mean()

      # Replace the np.nan values with its mean value
      data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean, inplace=True)

      data_falcon9.isnull().sum()
```

Data Collection - Scraping

- Used GET request retrieve Falcon 9 launch data from Wikipedia
- Created BeautifulSoup object from the response content
- Retrieved column names from HTML table
- Lastly, the HTML tables were parsed and placed into a data frame
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-webscraping.ipynb

[11]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

[13]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

[27]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

first_launch_table = html_tables[2]  # Assuming we are interested in the first table on the page
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Check the extracted column names

[30]:
```
print(column_names)
```

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

Data Wrangling

- Number of launches per site was calculated
- Number and types of orbits were calculated
- Mission outcomes were determined
- Lastly, a "class" was created to indicate if the landing was successful (1) or not successful (0). This was added to the data frame for additional analysis
- The average success rate was determined to be 66.6%
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

```
[12]: # Apply value_counts() on column LaunchSite
      launch_site_counts = df['LaunchSite'].value_counts()
      print(launch_site_counts)

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
[14]: # Apply value_counts on Orbit column
      orbit_counts = df['Orbit'].value_counts()
      print(orbit_counts)

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
[16]: # landing_outcomes = values on Outcome column
      landing_outcomes = df['Outcome'].value_counts()
      print(landing_outcomes)

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

```
[37]: df["Class"].mean()

[37]: 0.6666666666666666
```

EDA with Data Visualization

- Various charts were utilized to explore the data
  - Scatterplot to explore flight number and payload on outcome
  - Scatterplot to explore flight number and launch site on outcome
  - Scatterplot to explore payload and launch site on outcome
  - Bar chart to visualize success rate of each orbit
  - Scatterplot to explore flight number and orbit type on outcome
  - Scatterplot to explore payload and orbit type on outcome
  - Line plot to visualize success rate over time
- By exploring the data visually, we can gain preliminary insights to how important each variable is to the success rate, allowing us to pick features for our prediction model
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Retrieved the distinct names of each launch site
- Retrieved top five records where one site begins with CCA
- Calculated the total payload mass carried by boosters launched by NASA (CRS)
- Calculated the average payload mass carried by booster version F9 V1.1
- Retrieved the first successful ground pad landing date
- Retrieved the boosters which have success in drone ship and have a payload mass between 4000 and 6000
- Calculated the total number of successful and failure mission outcomes
- Retrieved the booster versions that have carried the maximum payload mass
- Retrieved the failure records for drone ship in the year 2015
- Ranked the landing outcomes by count between specified dates
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites were marked on the map to visualize locations
- Marker clusters were used on each launch site to visualize success and failure outcomes
- Lines and distances were added to understand the proximity to railways, highways, and cities
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Graphed total successes by site, to see which launch sites had the highest successful launches, and which had the highest success rate
- Graphed payload success rates, to visualize the impact of payload and booster version on success
- The dashboard also has filters to allow flexibility in data exploration
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data was preprocessed and split into training and test data sets
- The data was then trained on multiple models, using GridSearchCV to find the optimized parameters for each model
- The models used were logistic regression, SVM, decision tree, and K nearest neighbors
- Accuracy was scored against the test data set to find the best model
- While the decision tree had the highest accuracy on the training data, all models performed the same on the test data. Consequently, no model had any advantage over the others
- GitHub URL: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

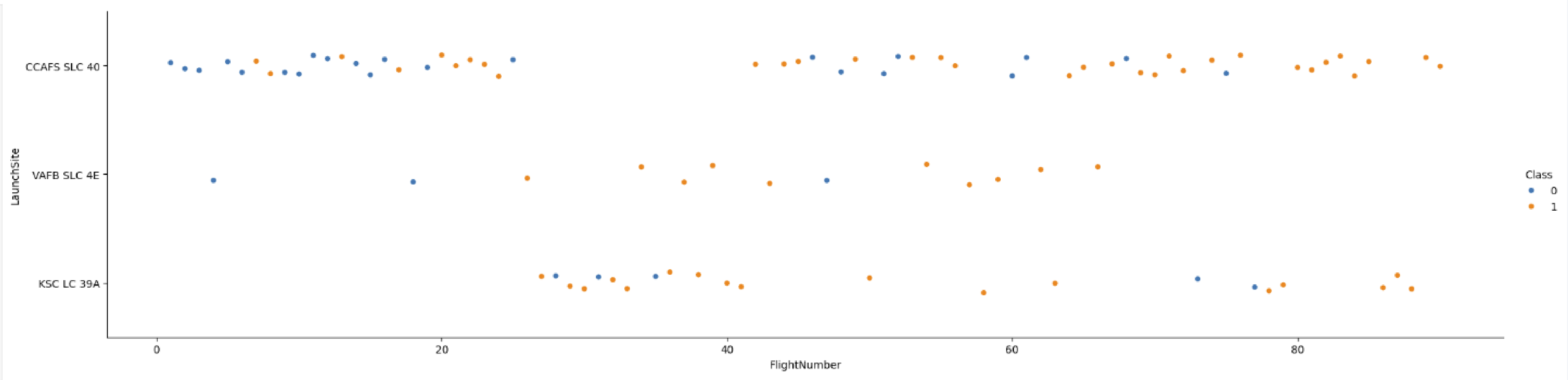
- Exploratory data analysis results
  - The success rate has improved over time
  - KSC LC-39A has had the greatest number of successful launches
  - CCAFS SLC-40 has had the least number of successful launches, but the highest success rate
  - Success rates vary by orbit as well, with ES-L1, GEO, HEO, SSO, and VLEO being the highest
- Interactive analytics
  - Launch sites are typically located in the southern US on the coast
  - They are far from cities, but still accessible by highways
- Predictive analysis results
  - All models performed the same, with 83% accuracy

Section 2

Insights drawn from EDA

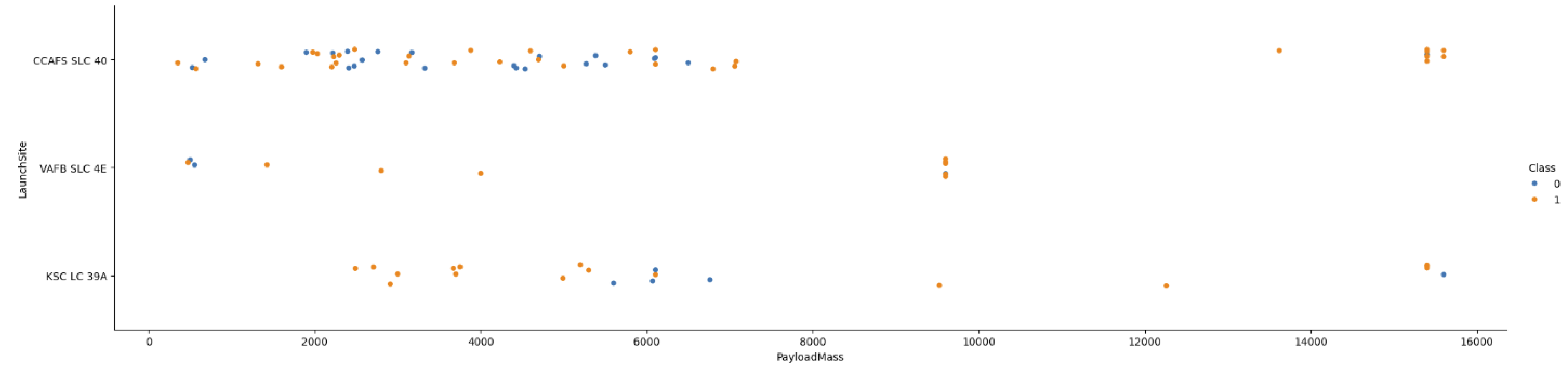
Flight Number vs. Launch Site

- There was an increase in the success rate over time from the CCAFS SLC 40 launch site



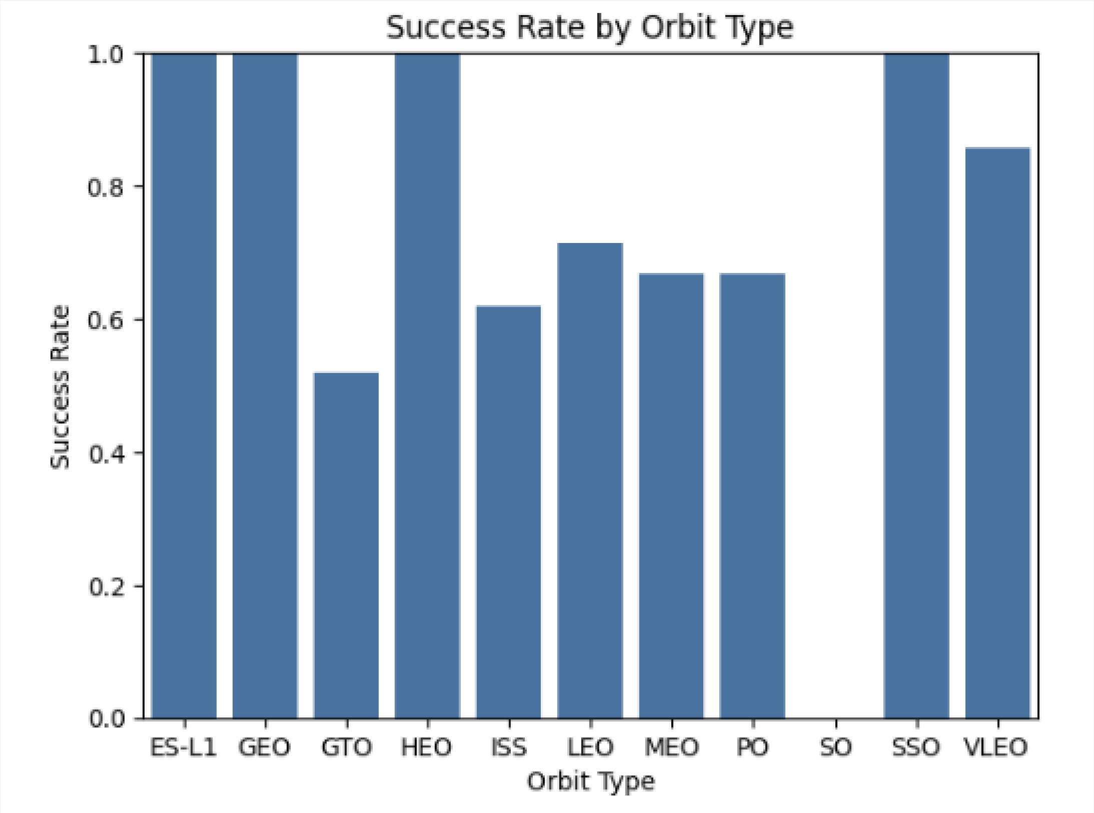
Payload vs. Launch Site

- Higher payloads tend to have higher success rates at the CCAFS SLC 40 launch site
- VAFB SLC 4E did not have any heavy payload masses greater than 10,000



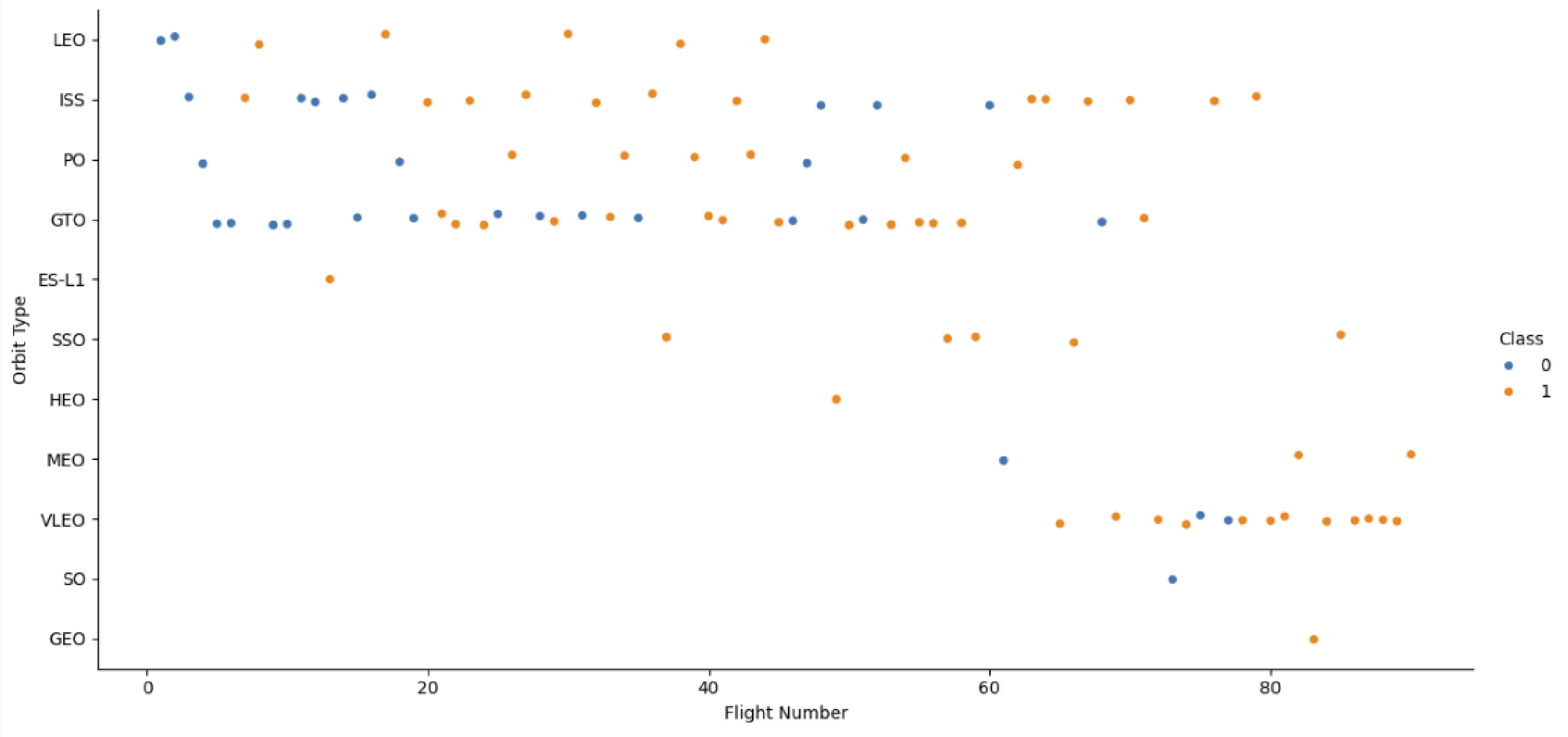
Success Rate vs. Orbit Type

- Charting the orbit success rate: ES-L1, GEO, HEO, SSO, VLEO had the highest rates



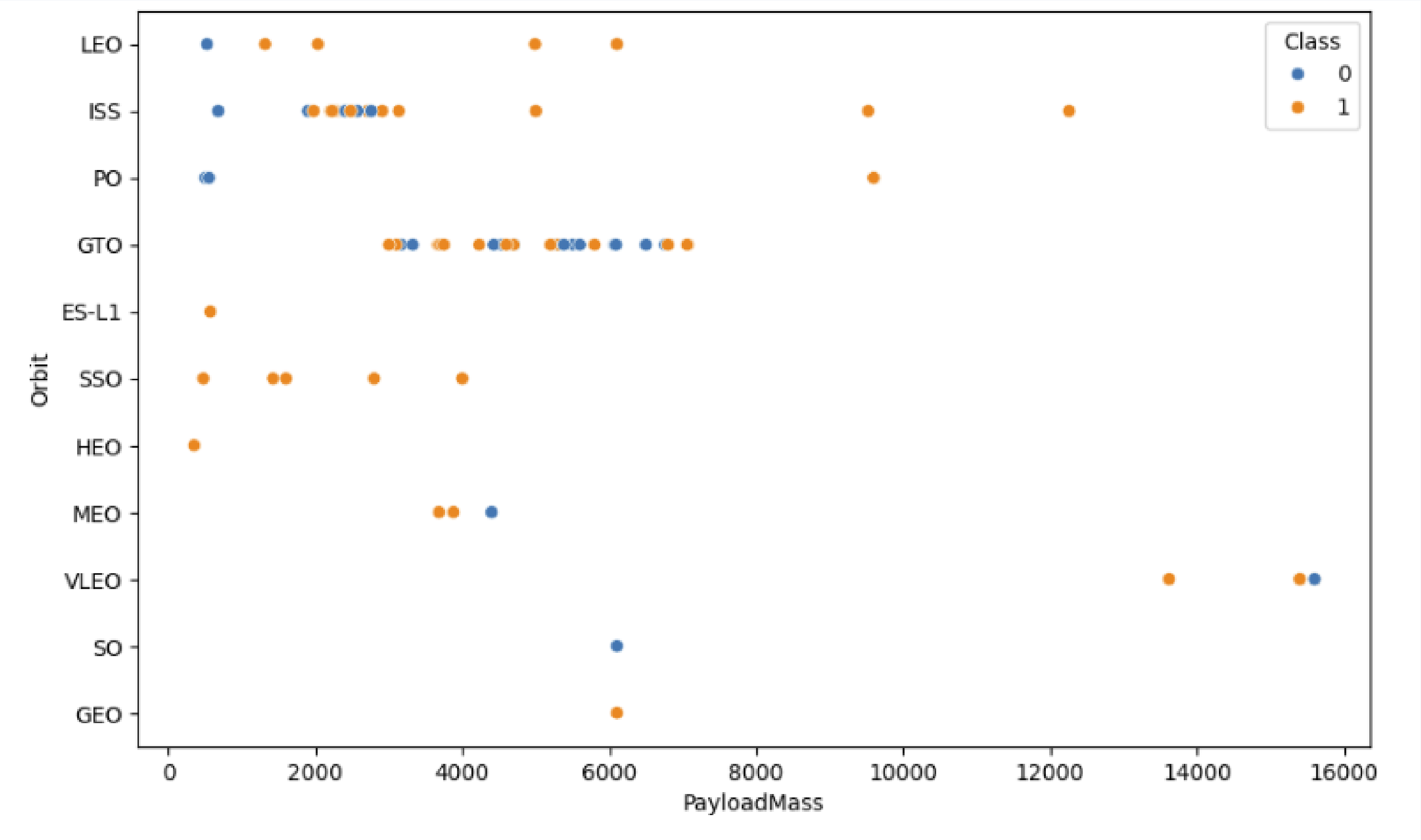
Flight Number vs. Orbit Type

- LEO demonstrates increased success related to flight number
- GTO does not show a relationship between success and flight number



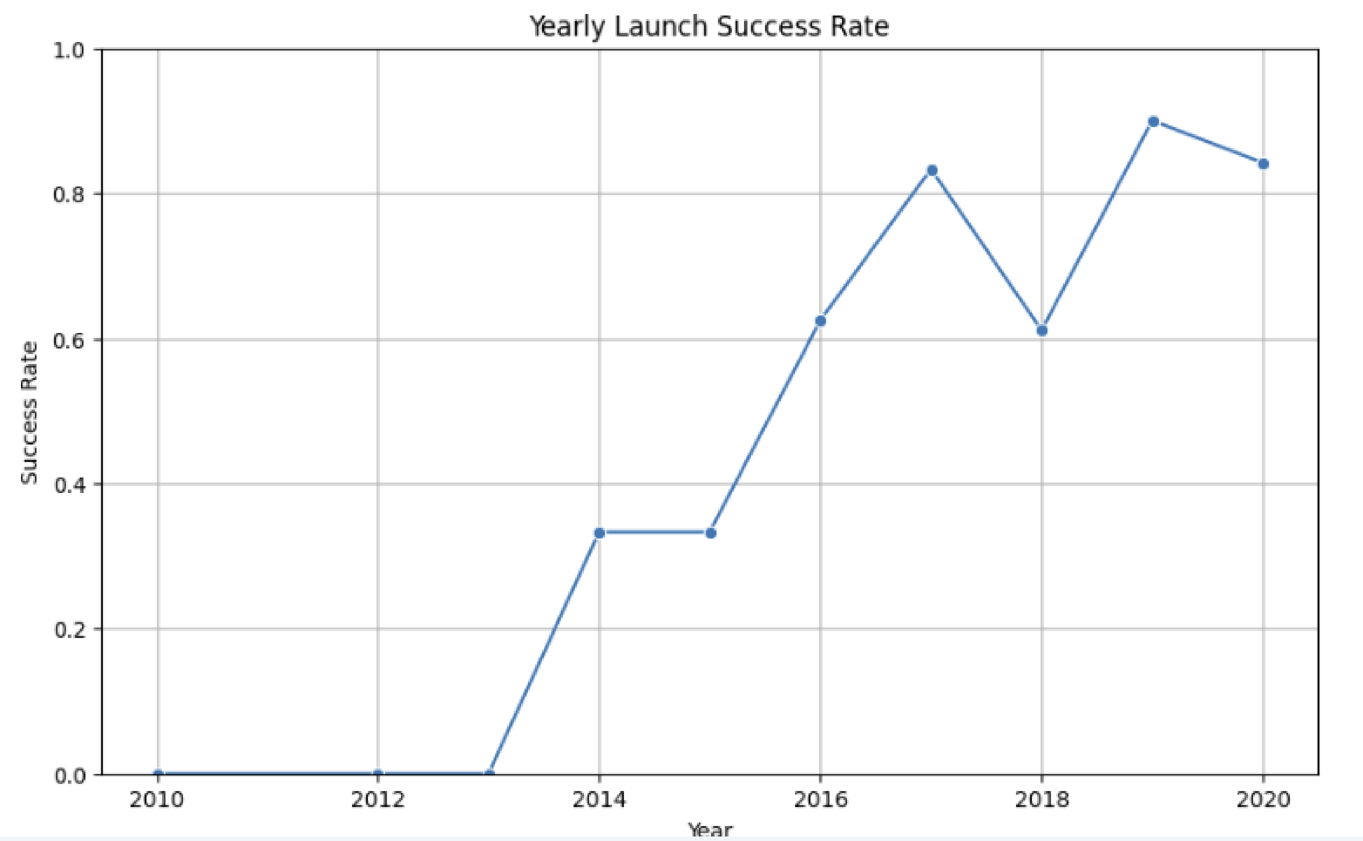
Payload vs. Orbit Type

- With heavy payloads the successful landing rate is more for Polar, LEO and ISS
- For GTO we cannot distinguish this well as



Launch Success Yearly Trend

- Over time, the yearly launch success rate has continually increased
- There was a dip in 2018 & 2020, but the trend is upward



All Launch Site Names

- There were four distinct launch sites found

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

```
[21]: qt1 = 'SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE'
t1 = pd.read_sql_query(query, con)
t1
```

[21]:

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

```
[23]: qt2 = '''
SELECT *
FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
'''
t2 = pd.read_sql_query(qt2, con)
t2
```

[23]:

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Total payload carried by boosters from NASA : 45,596 kg

[25]:
```
qt3 = '''
SELECT SUM("PAYLOAD_MASS__KG_") as Total_Payload_Mass
FROM SPACEXTABLE
WHERE "Customer" = 'NASA (CRS)';
'''
t3 = pd.read_sql_query(qt3, con)
t3
```

[25]:

	Total_Payload_Mass
0	45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 : 2928.4 kg

[27]:
```
qt4 = '''
SELECT AVG("PAYLOAD_MASS__KG_") as Average_Payload_Mass
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
'''
t4 = pd.read_sql_query(qt4, con)
t4
```

[27]:

	Average_Payload_Mass
0	2928.4

First Successful Ground Landing Date

- First successful landing outcome on ground pad : 12/22/2015

```
[29]: qt5 = '''
SELECT MIN("Date") as First_Successful_Landing_Date
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
'''
t5 = pd.read_sql_query(qt5, con)
t5
```

[29]:

	First_Successful_Landing_Date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Of note: they are all FT boosters

[31]:
```
qt6 = '''
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
AND "PAYLOAD_MASS__KG_" > 4000
AND "PAYLOAD_MASS__KG_" < 6000;
'''
t6 = pd.read_sql_query(qt6, con)
t6
```

[31]:

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

```
[33]: qt7 = '''
SELECT "Mission_Outcome", COUNT(*) as Outcome_Count
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
'''
t7 = pd.read_sql_query(qt7, con)
t7
```

[33]:

	Mission_Outcome	Outcome_Count
0	Failure (in flight)	1
1	Success	98
2	Success	1
3	Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass
- Of note: they are all B5 boosters

```
[35]: qt8 = '''
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
'''
t8 = pd.read_sql_query(qt8, con)
t8
```

[35]:

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
[43]: qt9 = '''
SELECT substr("Date", 6, 2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND substr("Date", 0, 5) = '2015';
'''
t9 = pd.read_sql_query(qt9, con)
t9
```

[43]:

	Month	Landing_Outcome	Booster_Version	Launch_Site
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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

[39]:
```
qt10 = '''
SELECT "Landing_Outcome", COUNT(*) as Outcome_Count
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
'''
t10 = pd.read_sql_query(qt10, con)
t10
```

[39]:

	Landing_Outcome	Outcome_Count
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Failure (parachute)	2
7	Precluded (drone ship)	1

Section 3

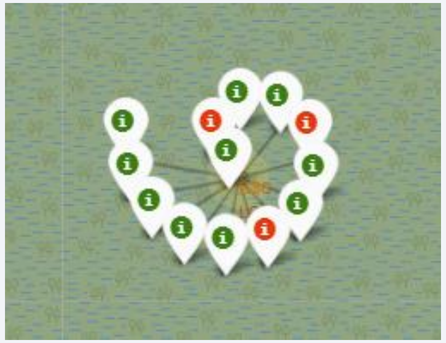
Launch Sites Proximities Analysis

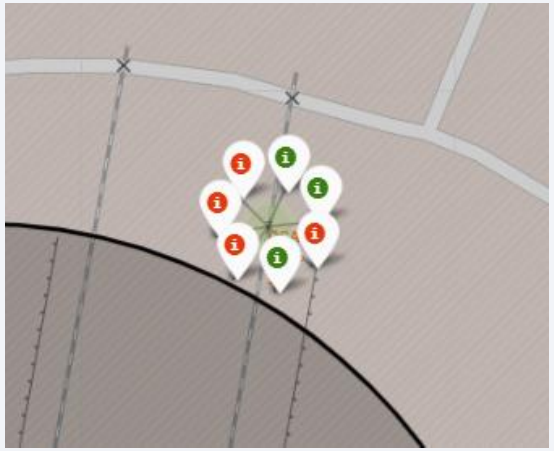
Launch Sites

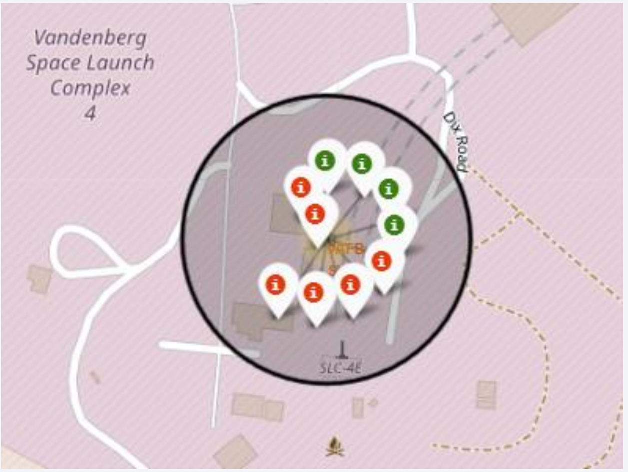
- With the launch sites mapped, we see that they are located in the southern US and on the West and East coasts

Launch Success Rate by Site

- By mapping the launches and coloring by success or failure, we can visualize both the quantity of launches at each site and the number of successes versus failures

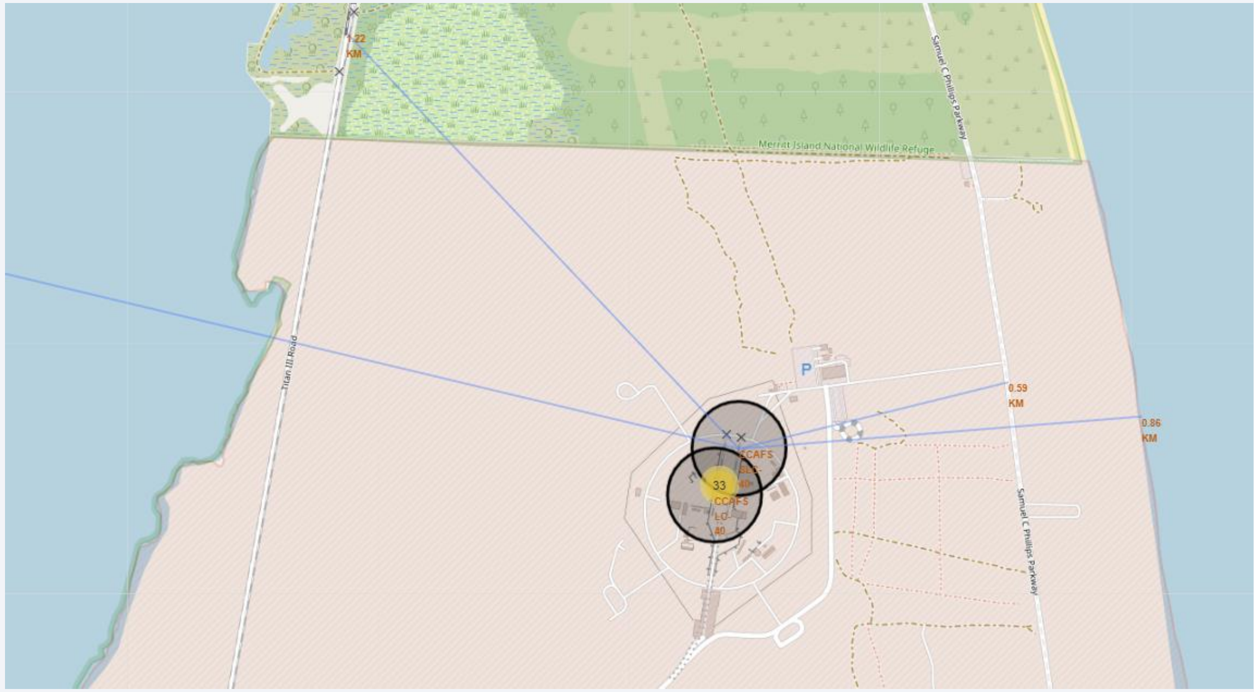






Launch Site Distance to Other Map Objects

- Launch sites may be located near railways, highways, and coastlines
- Launch sites are typically located away from cities

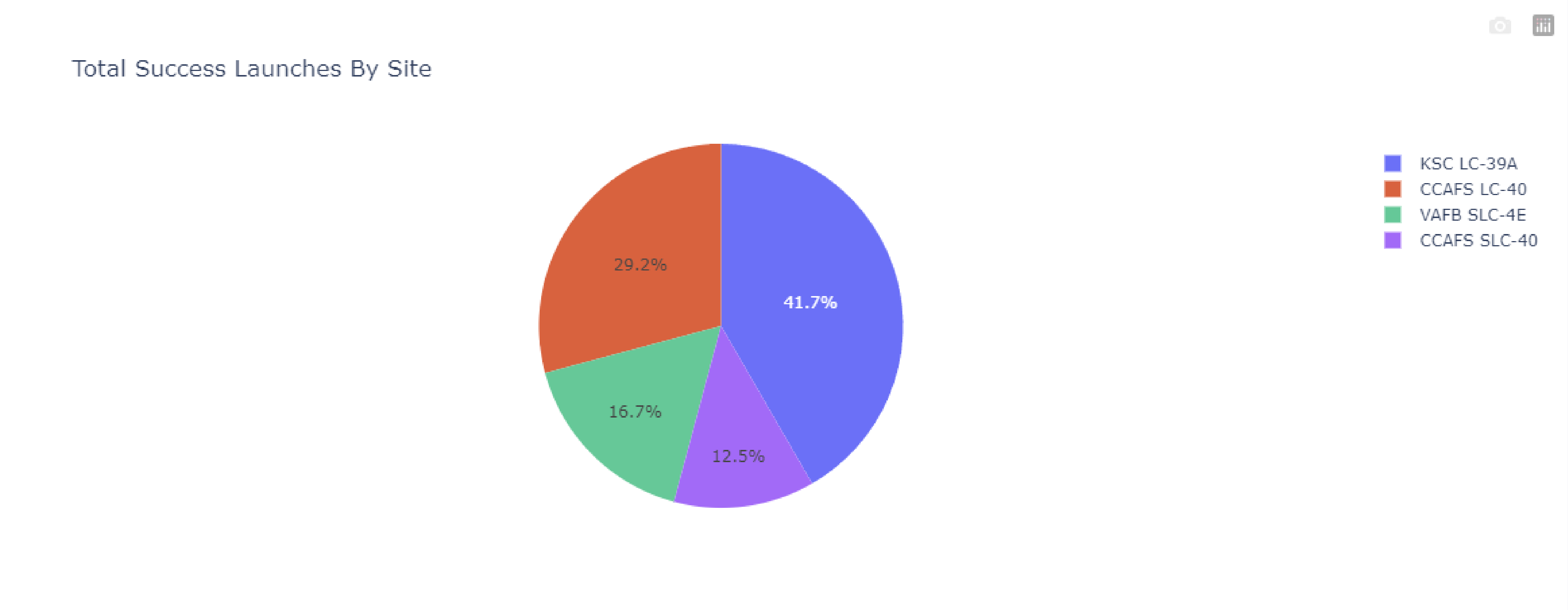


Section 4

Build a Dashboard with Plotly Dash

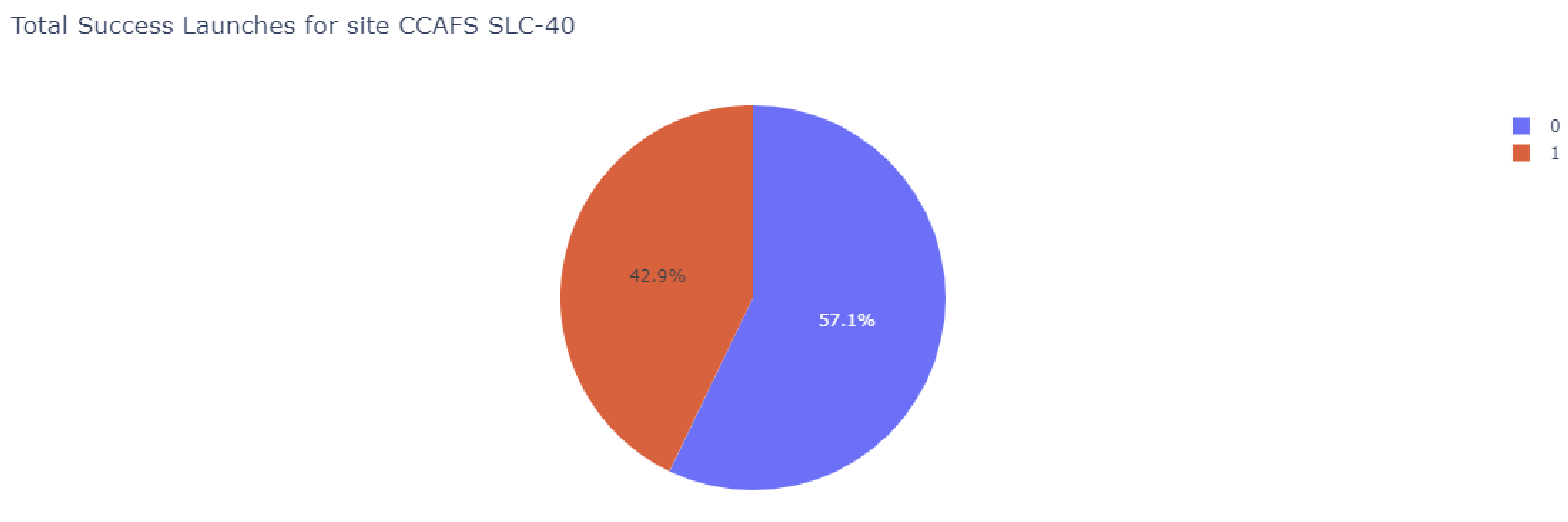
Successful Launches by Site

- KSC LC-39A has had the greatest number of successful launches
- CCAFS SLC-40 has had the least



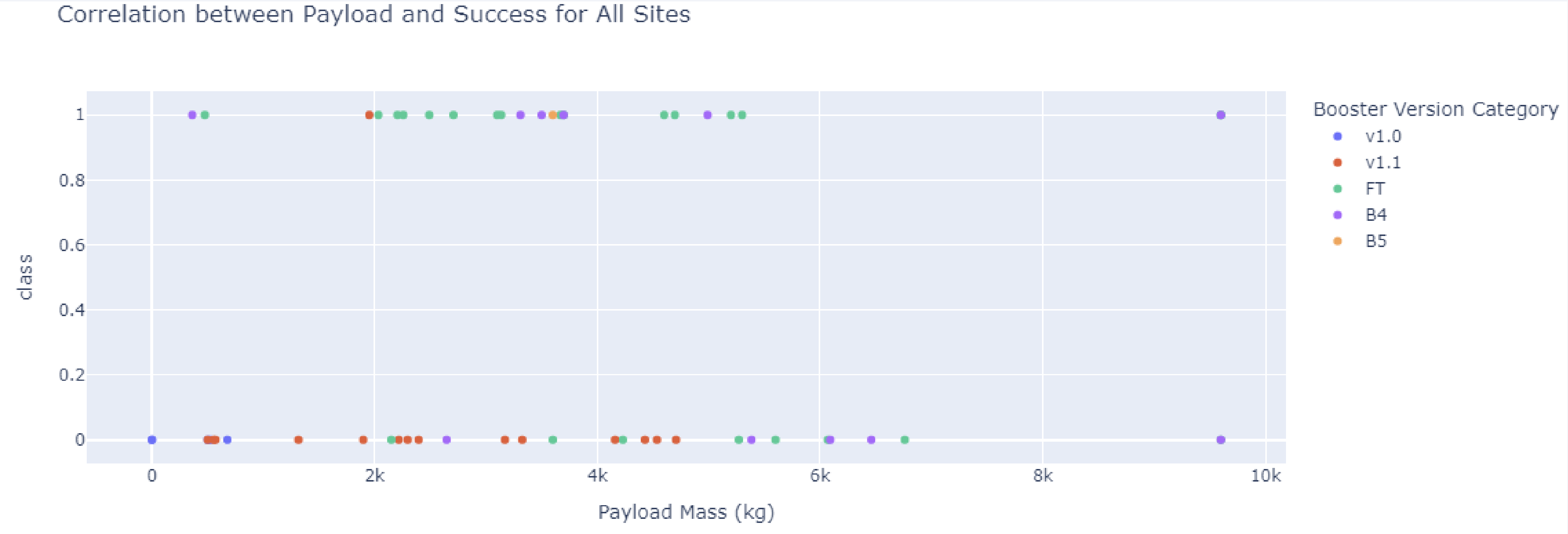
Launch Site with Highest Success Ratio

- CCAFS SLC-40 had the highest success rate for launches at approximately 43%



Success Rate Based on Payload and Booster Version

- The FT booster has had the highest number of successful launches
- v1.1 has had the most unsuccessful launches



Section 5

Predictive Analysis (Classification)

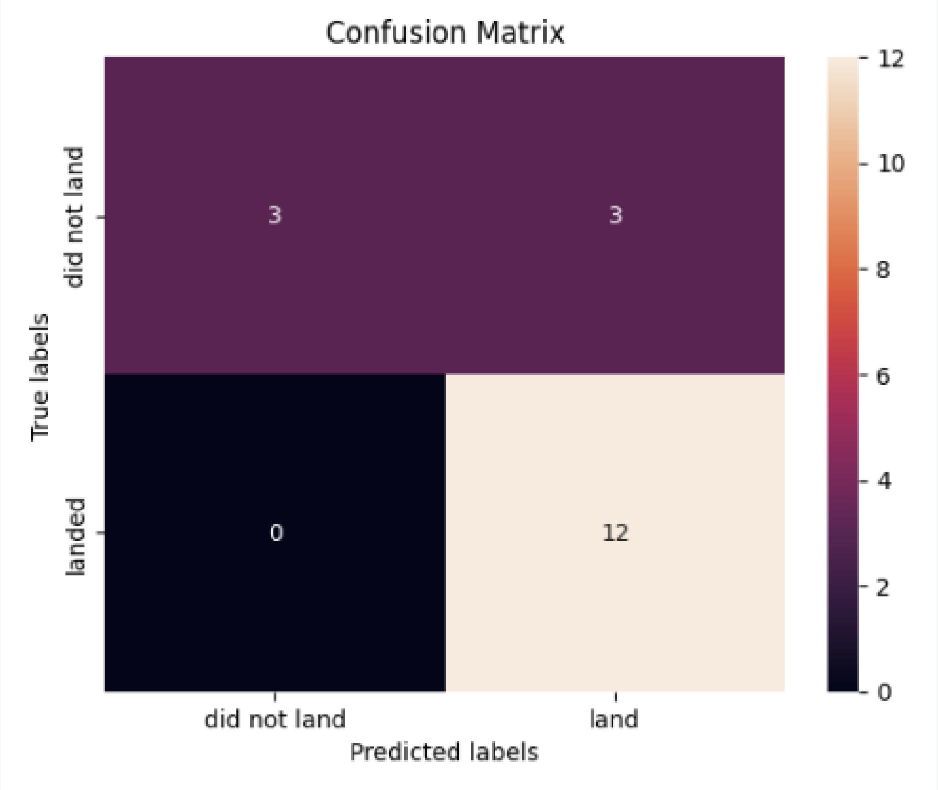
Classification Accuracy

- The accuracy of classification on the test data was identical among all the models, 83.3%

```
[24]: accuracy = logreg_cv.score(X_test, Y_test)
      accuracy

[24]: 0.8333333333333334

[27]: accuracy_svm = svm_cv.score(X_test, Y_test)
      accuracy_svm

[27]: 0.8333333333333334

[42]: accuracy_tree = tree_cv.score(X_test, Y_test)
      accuracy_tree

[42]: 0.8333333333333334

[37]: accuracy_knn = knn_cv.score(X_test, Y_test)
      accuracy_knn

[37]: 0.8333333333333334
```

Confusion Matrix

- All the models performed the same and generated the same confusion matrix
- All landings were predicted correctly, but there were some failures that were predicted incorrectly (false positives)



Conclusions

- Launch success rate has improved over time
- KSC LC-39A has had the greatest number of successful launches
- CCAFS SLC-40 has had the least number of successful launches, but the highest success rate
- Launch sites are located in the southern US (closer to the equator), close to coastlines, and away from cities
- All predictive models performed the same, with approximately 83% classification accuracy

Appendix

- GitHub project link: https://github.com/msalte2006/IBM-Data-Science-SpaceX-Capstone


Thank you!