IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

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

- Summary of methodologies
  - Data Collection via SpaceX API
  - Data Collection via Web Scraping
  - Data Wrangling
  - SQL Exploratory Data Analysis (EDA)
  - EDA with Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Finalized Exploratory Data Analysis
  - Visual analytics summarized with screenshots
  - Data predictions

- Project background and context
  - In this project, we predicted whether or not the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62 million, while other providers cost upward of $165 million. Much of these savings are due SpaceX's ability to reuse the first stage. Therefore, if we can determine if the first stage will land successfully, we can determine the cost of a launch. This information can then be used as leverage if an alternative company wants to bid against SpaceX for a rocket launch.
- Questions to answer:
  - How can we determine if the rocket's first stage will land successfully?
  - What relevant data can be extracted to determine these insights, and how can this data be used and visualized?
  - Are there certain factors, such as location or payload mass, that are more determinant of the success or failure of the launch?

Section 1

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX’s API and scraped from Wikipedia tables.
- Perform data wrangling
  - Methods such as “one-hot encoding” and training label determination from categories.
- 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 was collected via the following methods:
  - API requests to SpaceX’s API
  - Web Scraping from Wikipedia tables regarding SpaceX launches
  - Data cleaning (removing missing values & replacing relevant values)
  - Exploratory analysis with SQL from CSV files

- https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_1_Data_Collection_API_Lab.ipynb

**1) Get response from API**

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

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

**2) Convert response to .json file**

```
# Use json_normalize meethod to convert the json result int
response = requests.get(static_json_url)
json_data = response.json()
df = pd.json_normalize(json_data)
print(df)
```

**3) Apply data cleaning functions**

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

**4) Assign list to dictionary and dataframe**

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

**5) Filter dataframe and export to CSV**

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion']!='Falcon 1']
```

```
payload_mass_mean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean)
data_falcon9.to_csv('dataset_part_1.csv', index=False)
data_falcon9.isnull().sum()
```

- **https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_1_Web_Scraping_Lab.ipynb**

1) Get response from HTML

```
response = requests.get(static_url).text
```

2) Create BeautifulSoup object

```
soup = BeautifulSoup(response, "html.parser")
```

3) Find tables

```
html_tables = soup.find_all("table")
```

4) Get column names

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

5) Create dictionary

```
launch_dict= dict.fromkeys(column_names)
del launch_dict['Date and time ( )']
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6) Append data to keys (too much info here to include entire screenshot)

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row=rows.find_all('td')
        if flag:
            extracted_row += 1
            launch_dict['Flight No.'].append(flight_number)
            datatimelist=date_time(row[0])
```

7) Convert dictionary to dataframe and CSV

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df.to_csv('spacex_web_scraped.csv', index=False)
```

- https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_1_Data_Wrangling_Lab.ipynb

1) Check for Null Values

```
df.isnull().sum()/len(df)*100
```

2) Calculate number of launches per site

```
launch_site_counts = df['LaunchSite'].value_counts()
```

3) Calculate the number and occurrence of each orbit

```
orbit_counts = df['Orbit'].value_counts()
```

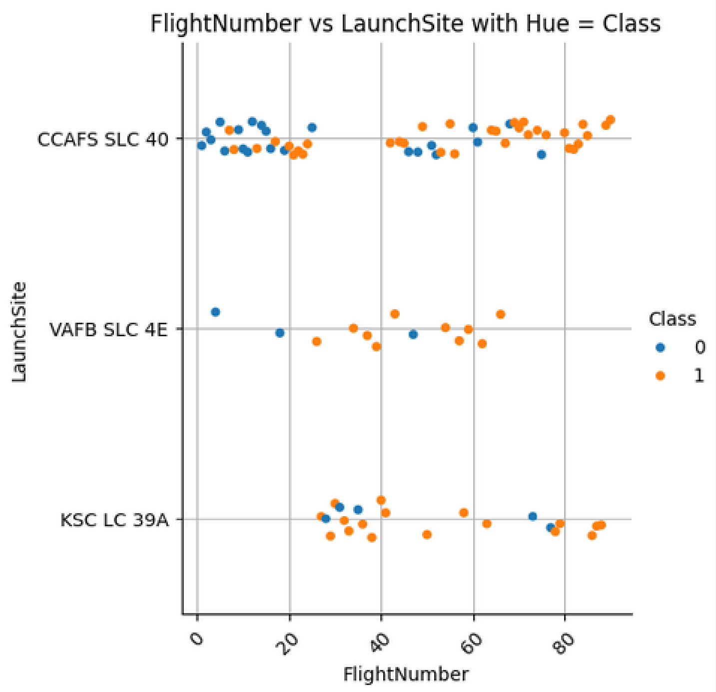
4) Calculate number and occurrence of mission outcomes

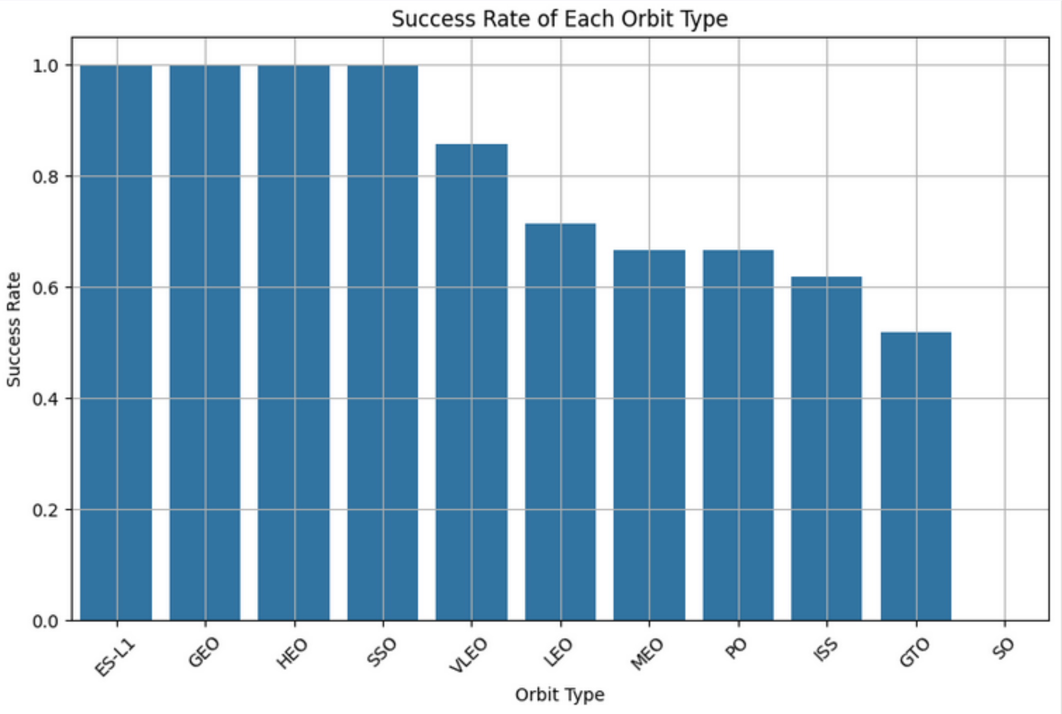
```
landing_outcomes = df['Outcome'].value_counts()
```

5) Create landing outcome labels

```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```

- Visualizations were made to compare factors such as Launch Site vs Payload Mass, Class vs Orbit, Flight Number vs Orbit, Payload Mass vs Orbit, Flight Number vs Launch Site, and Success Rate per Year.
- https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_2_Visualization_EDA_Lab.ipynb



- SQL queries performed include:
  - Names of unique launch sites
  - 5 records where launch sites begin with ‘CCA’
  - Total payload mass of boosters launched by NASA(CRS)
  - Average payload mass by booster
  - Date of first successful landing outcome in ground pad
  - Names of boosters with success in drone ship, and mass between 4000-6000 KG
  - Total number of successful and failure mission outcomes
  - Names of booster versions that have carried the max payload
  - Month names, failure landing outcomes in drone ships ,booster versions, and launch sites for the months in year 2015
  - The count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

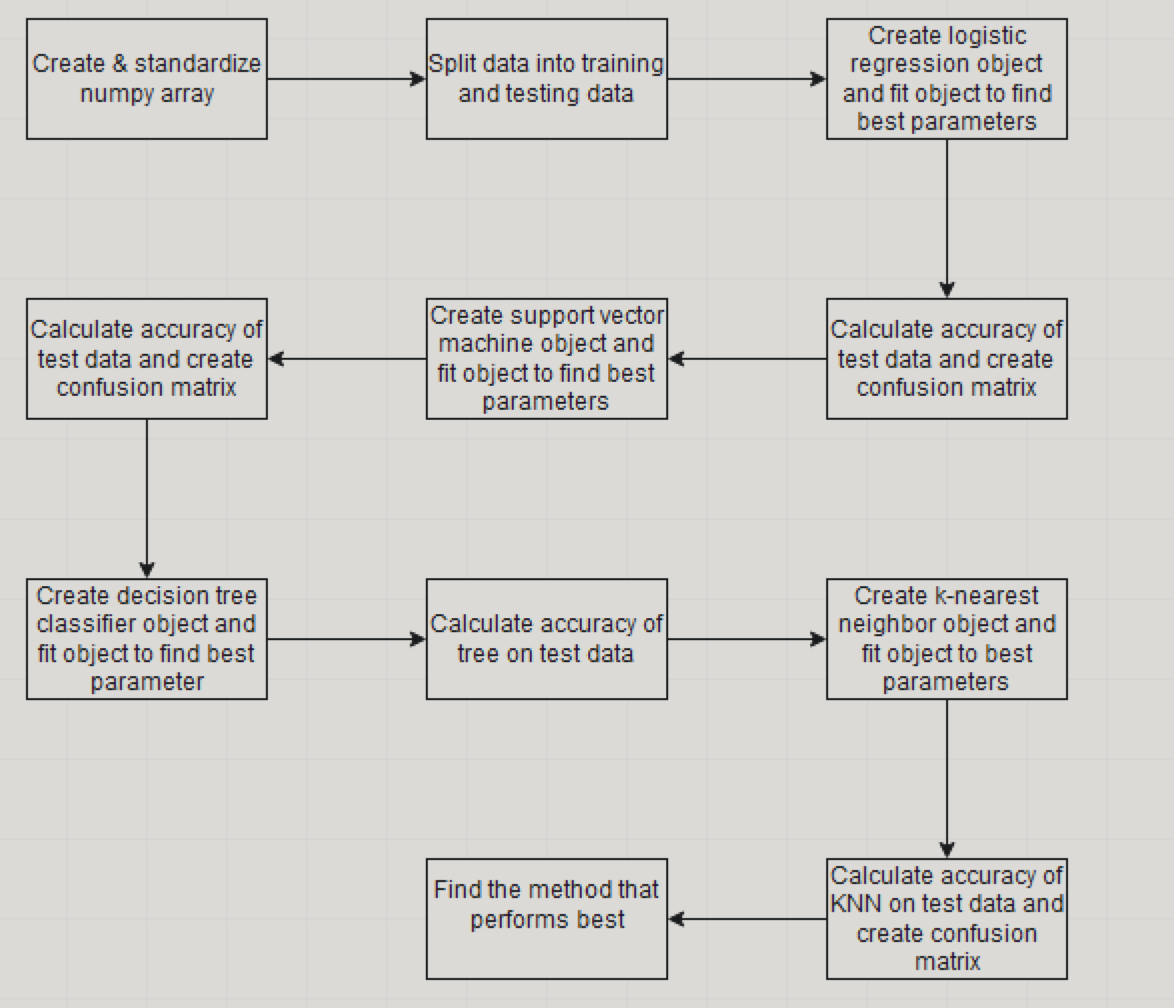
https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_2_SQL_EDA_Lab.ipynb

- Launch sites were marked on the map with markers, circles, and lines to mark the success or failure of the launches for each site on a folium map.
- Launch outcomes were also assigned.
- These markers enabled identification of launch sites with high and low success rates, thus finding the optimal launch sites.

https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_3_Folium_Interactive_Visual_Analytics_Lab.ipynb

- We used Plotly Dash to build an interactive dashboard displaying statistics such as total successes per site, total launches, payload mass vs. outcome, and different booster versions.
- This interactive dashboard illustrates these statistics in a user-friendly manner for anyone to observe easily.
- https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_3_Plotly_Interactive_Dash.py

- Data was loaded with numpy & pandas to be transformed and split.
- GridSearchCV was used to tune hyperparameters, and accuracy was used to improve the model in order to find best performing classification model.
- https://github.com/puk82/IBM-Applied-Data-Science-Capstone-Project/blob/main/Week_4_Machine_Learning_Prediction_Lab.ipynb

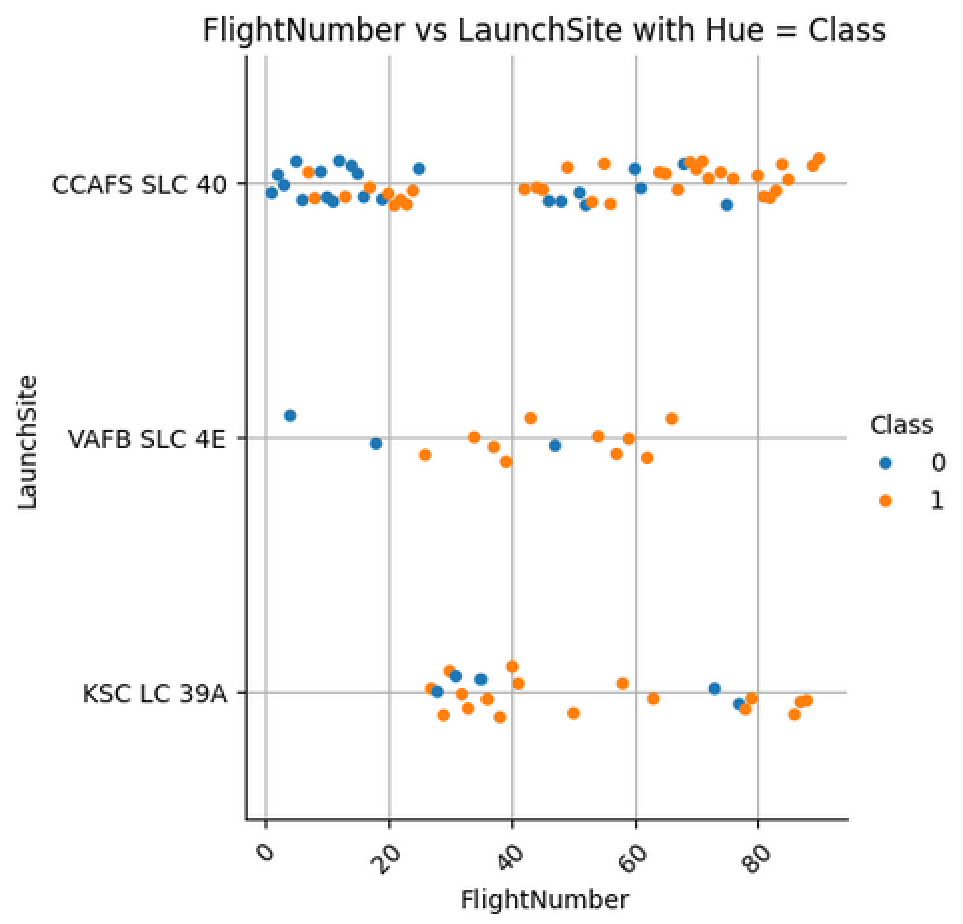


- For best results in prediction accuracy, SVM, KNN and logistic regression models should be used.
- Heavier payloads fail more often than lighter payloads.
- According to yearly analysis, success rates are regularly improving over time.
- The most successful Launch Site is KSC LC 39A.

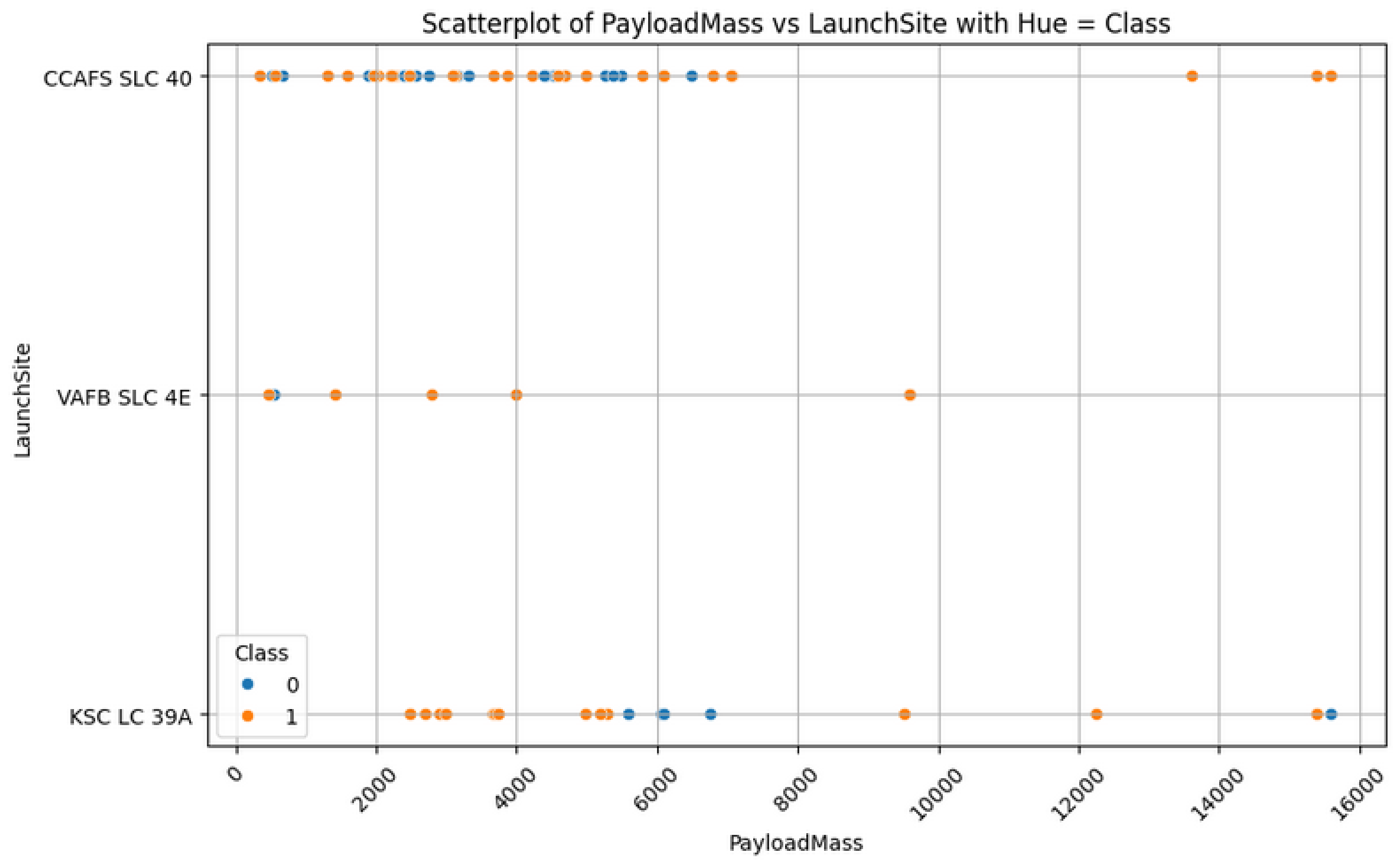
Section 2

# Insights drawn from EDA

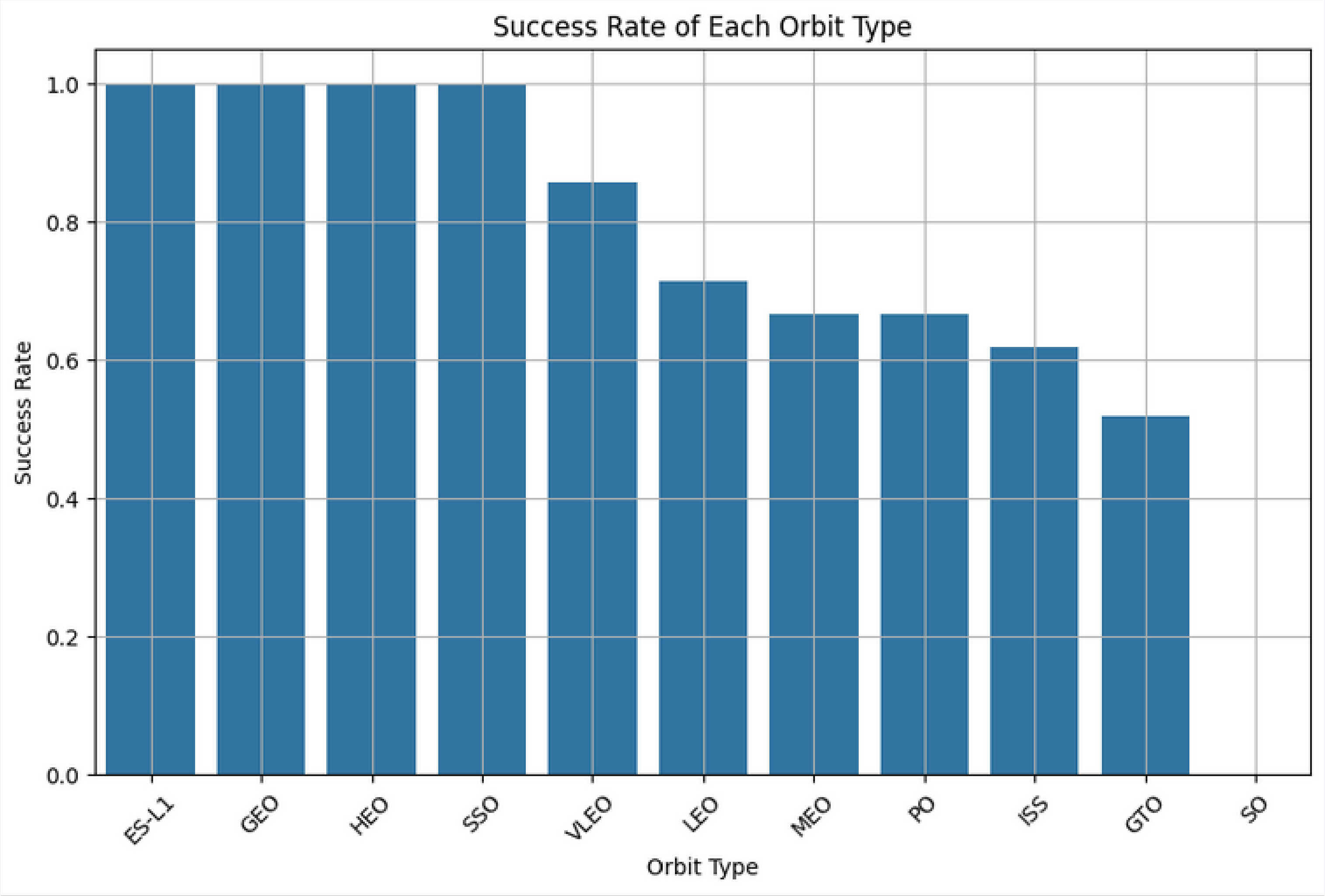
- There are far more launches from CCAFS SLC 40 than the other two sites.



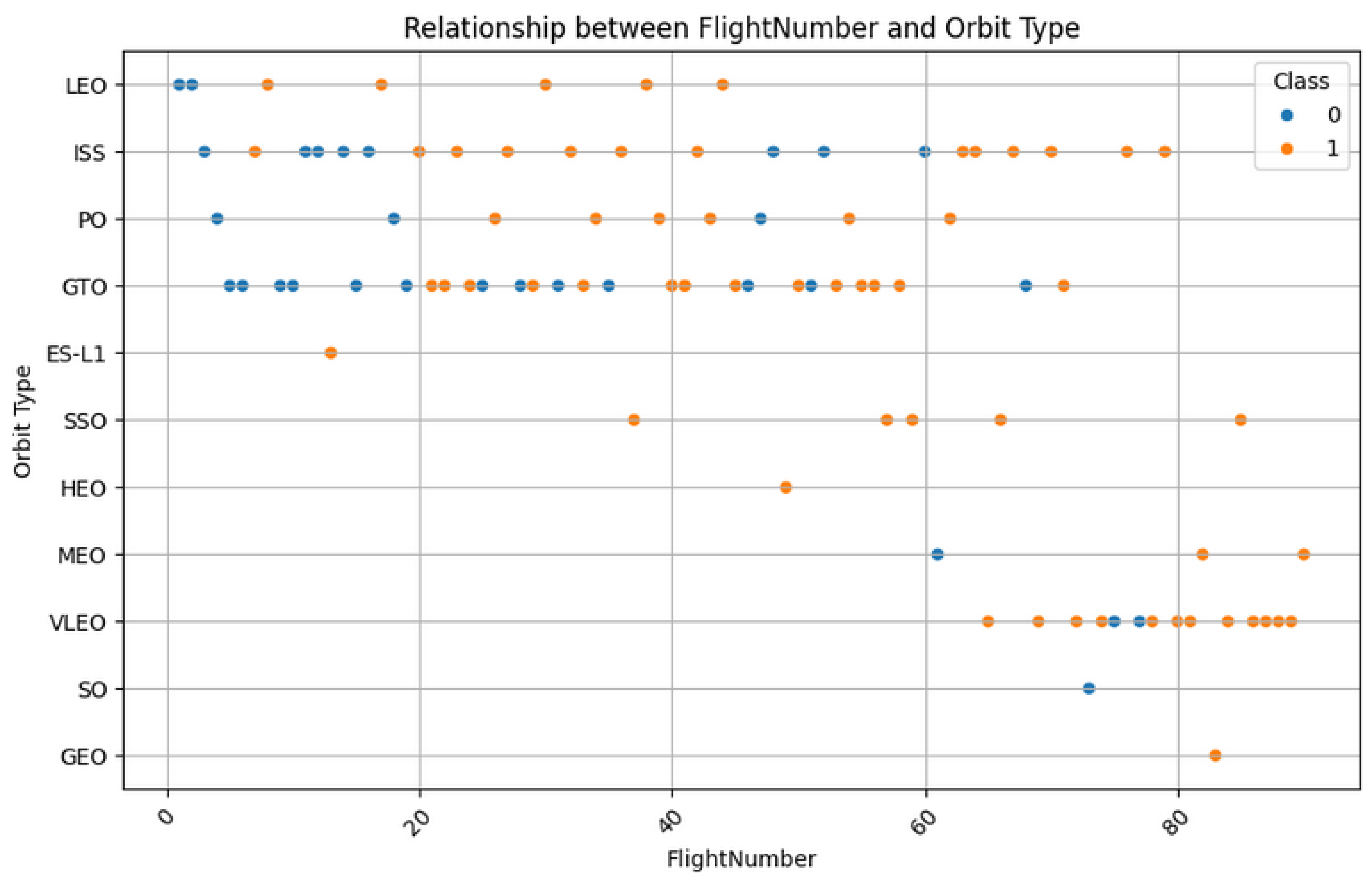
- Lighter payloads tend to succeed more often.



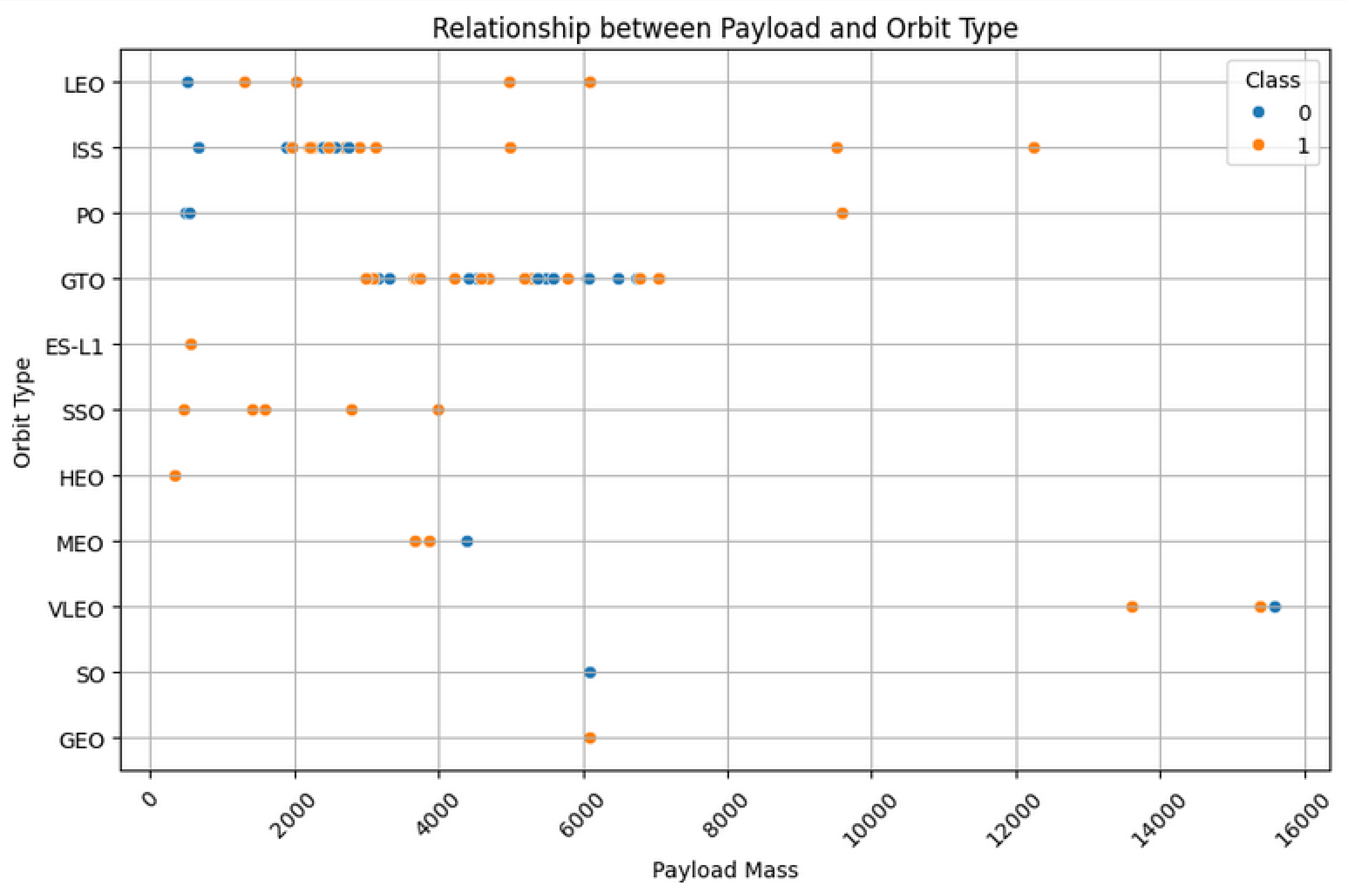
- Highest success rates include ES-L1, GEO, HEO, and SSO.



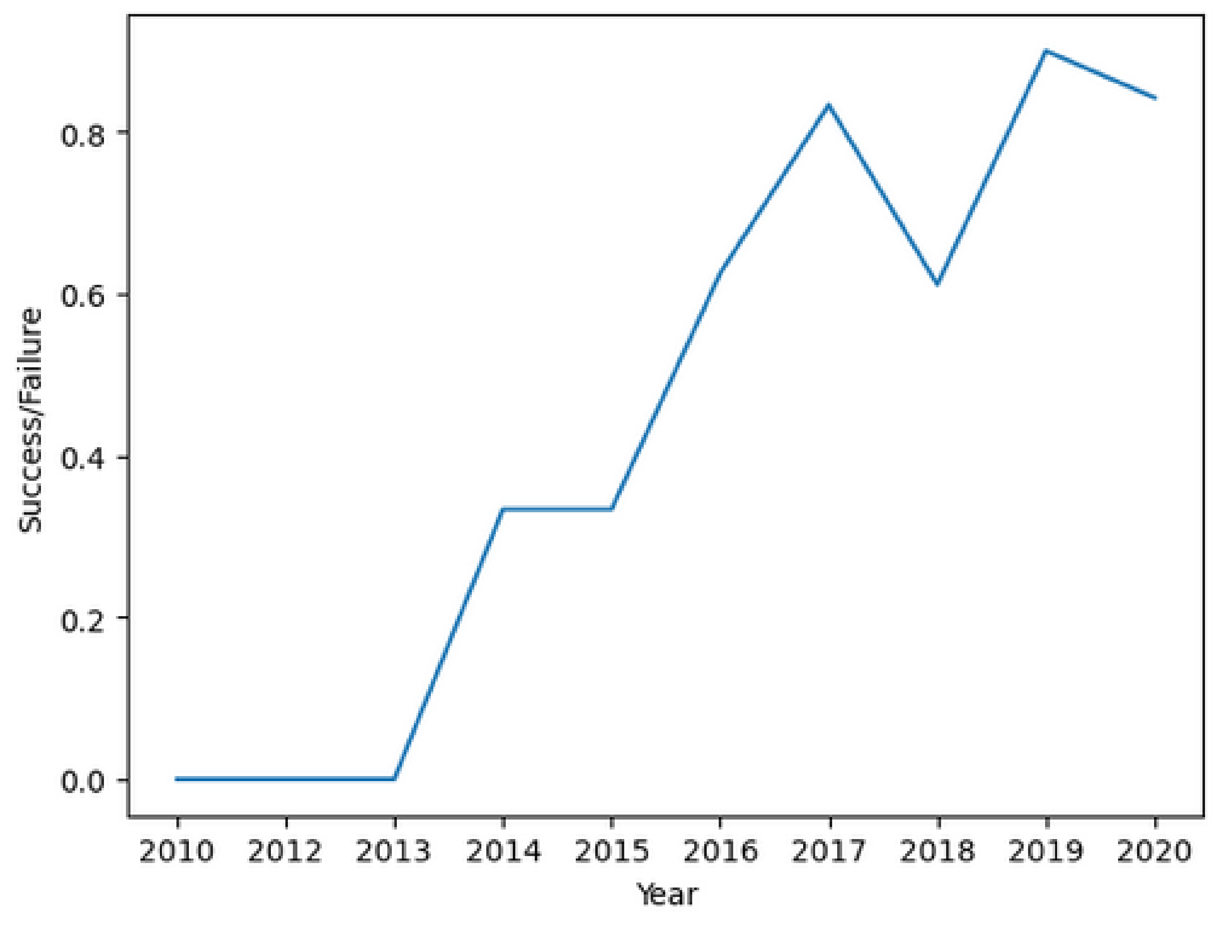
- LEO, ISS, PO and GTO orbit types correspond to older launches, whereas the more recent launches tend to be VLEO orbit types.



- Certain orbit types correspond strongly with certain payload mass ranges, such as ISS between 2000-4000 and GTO between 2000-8000.



- Success rates seem to be largely increasing per year with some exceptions in 2018 and 2020.



- Four different unique launch sites.

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

- Find 5 records between 2010 and 2013where launch sites begin with `CCA`

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

- Calculate the total payload carried by boosters from NASA

| SUM(PAYLOAD_MASS__KG_) |
|---:|
| 45596 |

- Calculate the average payload mass carried by booster version F9 v1.1

| AVG(PAYLOAD_MASS__KG_) |
| ---: |
| 2534.6666666666665 |

- Find the dates of the first successful landing outcome on ground pad

| MIN(Date) |
| --- |
| 2015-12-22 |

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

| Booster_Version |
| --- |
| F9 FT B1021.1 |
| F9 FT B1022 |
| F9 FT B1023.1 |
| F9 FT B1026 |
| F9 FT B1029.1 |
| F9 FT B1021.2 |
| F9 FT B1029.2 |
| F9 FT B1036.1 |
| F9 FT B1038.1 |
| F9 B4 B1041.1 |
| F9 FT B1031.2 |
| F9 B4 B1042.1 |
| F9 B4 B1045.1 |
| F9 B5 B1046.1 |

- Calculate the total number of successful and failure mission outcomes
- Number of outcomes as distinct (unique)

| COUNT(DISTINCT(Mission_Outcome)) |
| ---: |
| 4 |

- List the names of the booster which have carried the maximum payload mass

| Booster_Version |
| --- |
| 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 |

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

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

- 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

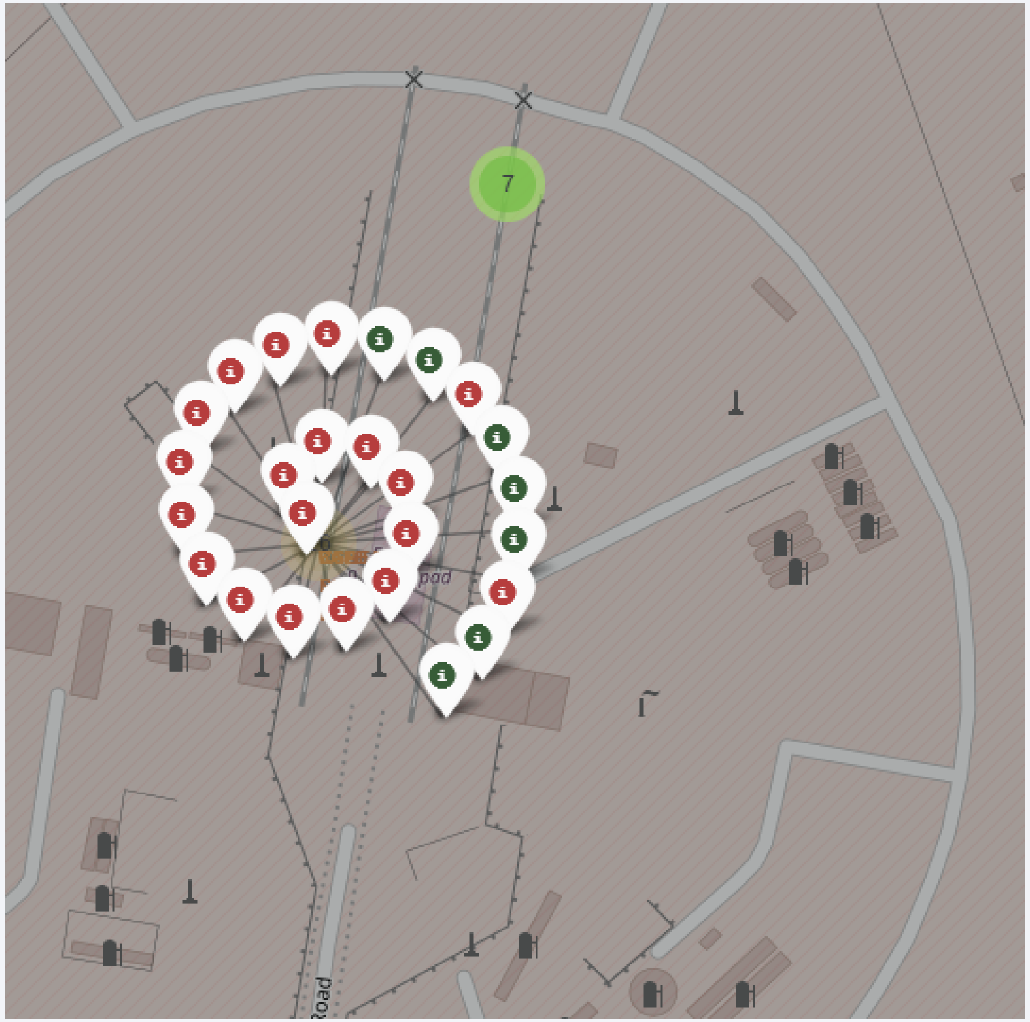
| | landingoutcome | count |
|---|---|---|
| 0 | No attempt | 10 |
| 1 | Success (drone ship) | 6 |
| 2 | Failure (drone ship) | 5 |
| 3 | Success (ground pad) | 5 |
| 4 | Controlled (ocean) | 3 |
| 5 | Uncontrolled (ocean) | 2 |
| 6 | Precluded (drone ship) | 1 |
| 7 | Failure (parachute) | 1 |

Section 3

# Launch Sites Proximities Analysis

- SpaceX launch sites on East and West coasts of USA

- One example of launch site outcomes. Green is success and red is failure.

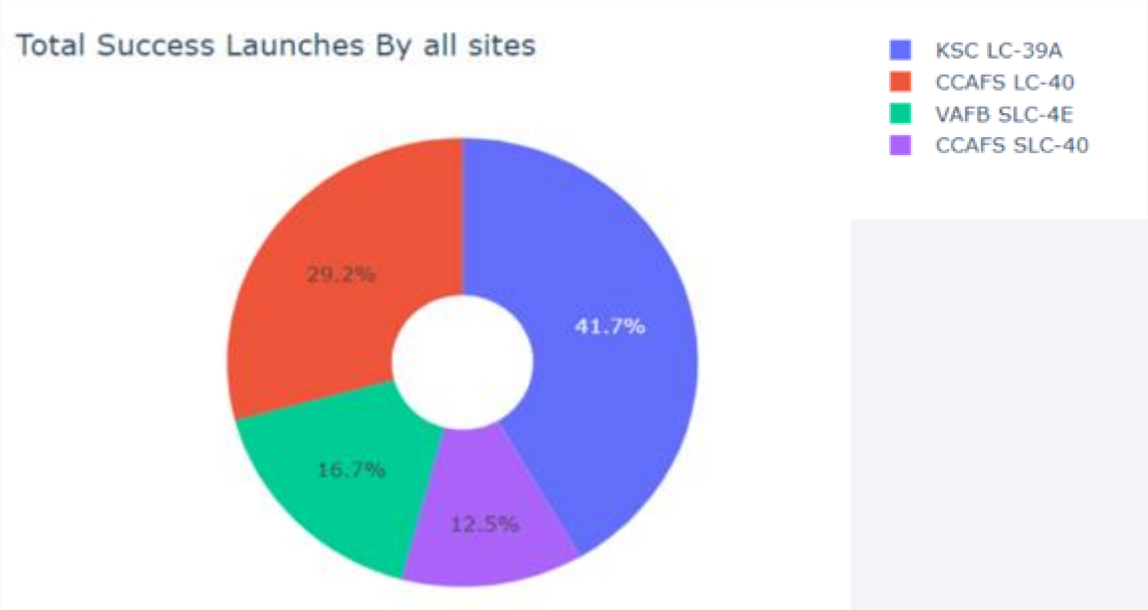


- Proximity to coastline point illustrated with blue line.

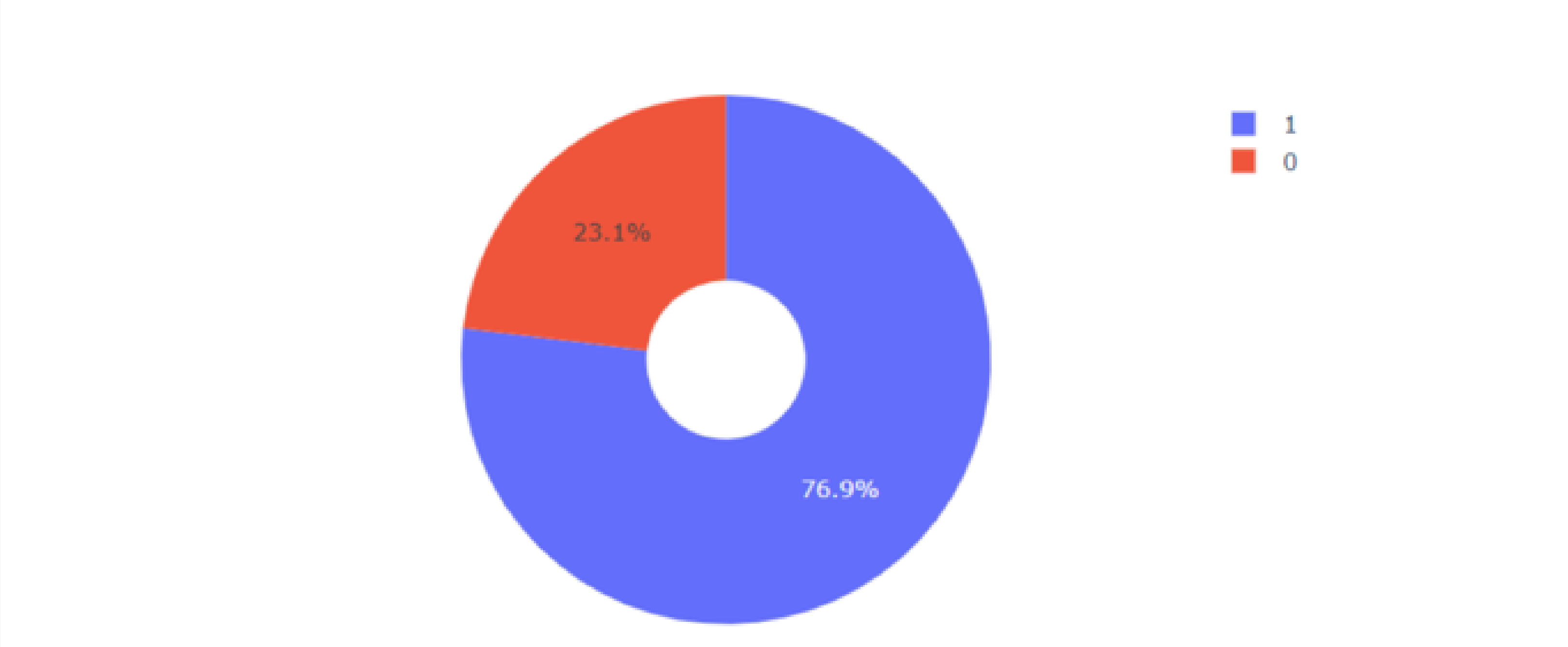
Section 4

# Build a Dashboard with Plotly Dash

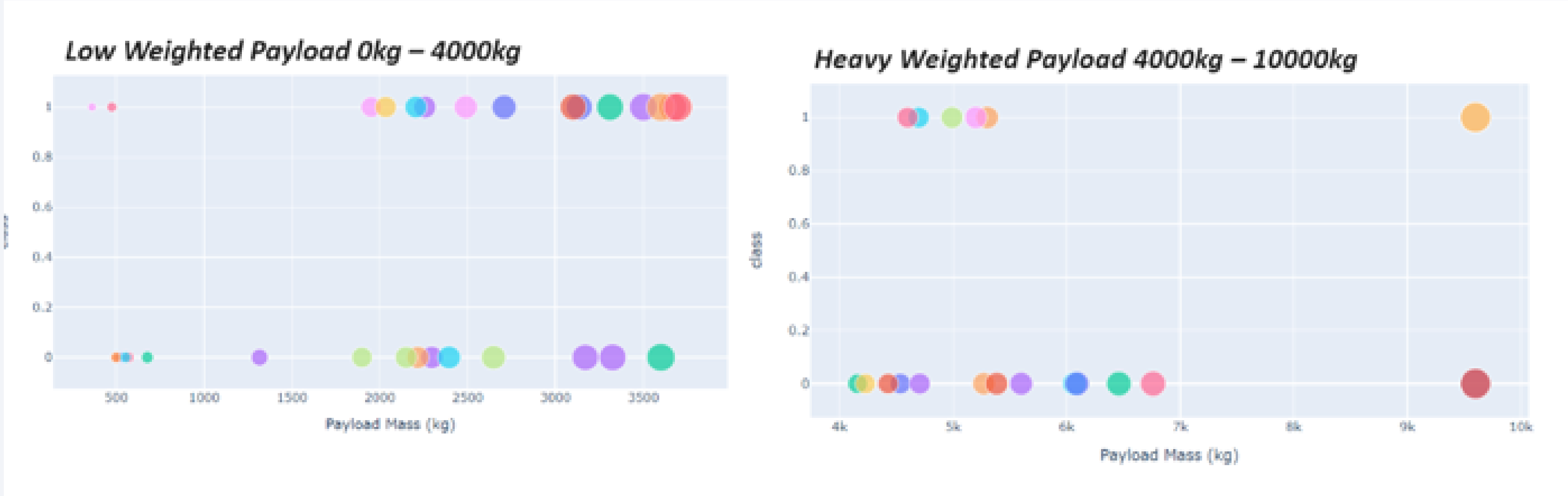
- A pie chart explaining different success rates of launches from launch sites.



- KSC LC-39A has a 76.9% successful launch rate, where 1 is success and 0 is failure.



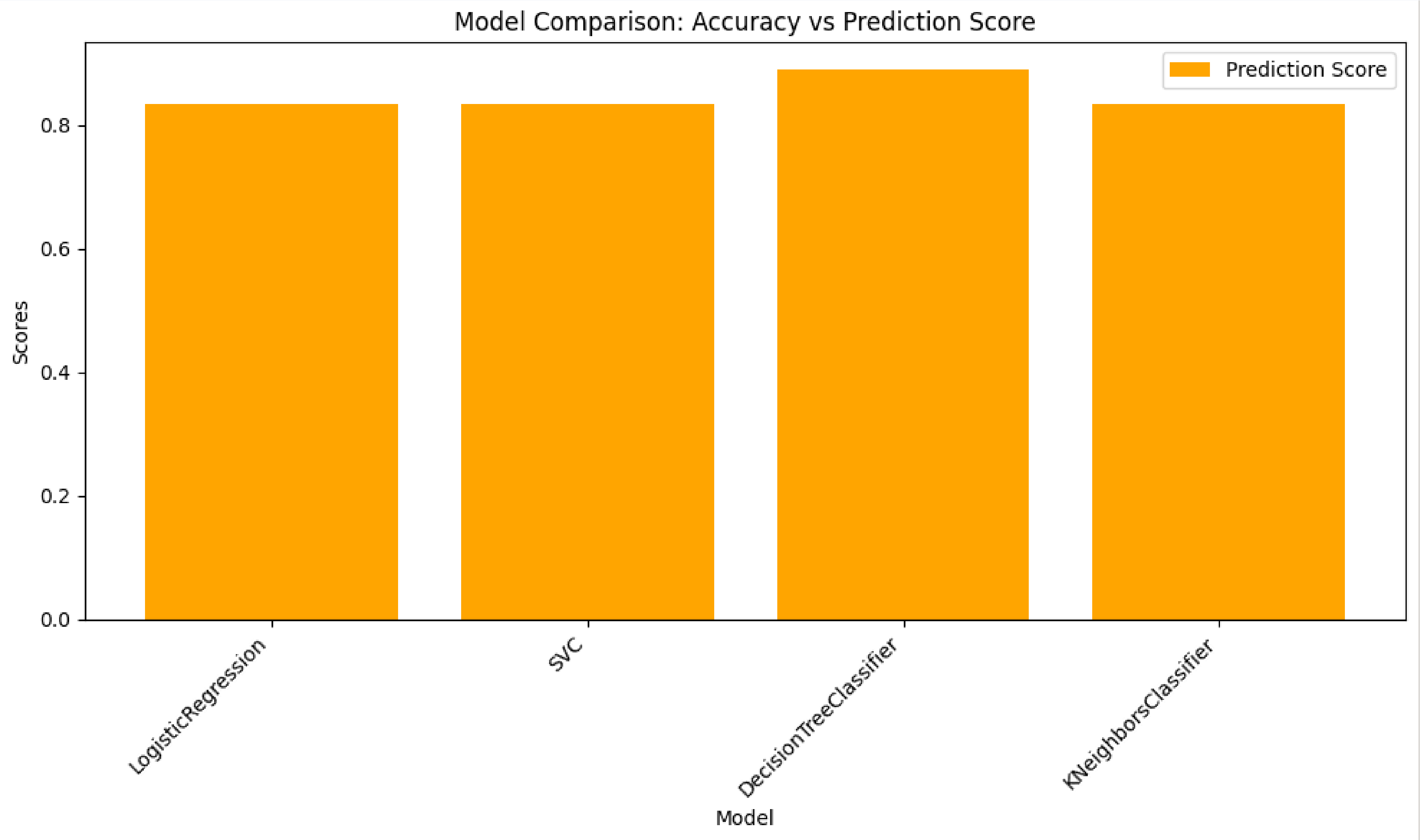
- Payloads with lower weight succeed more often than heavier payloads.



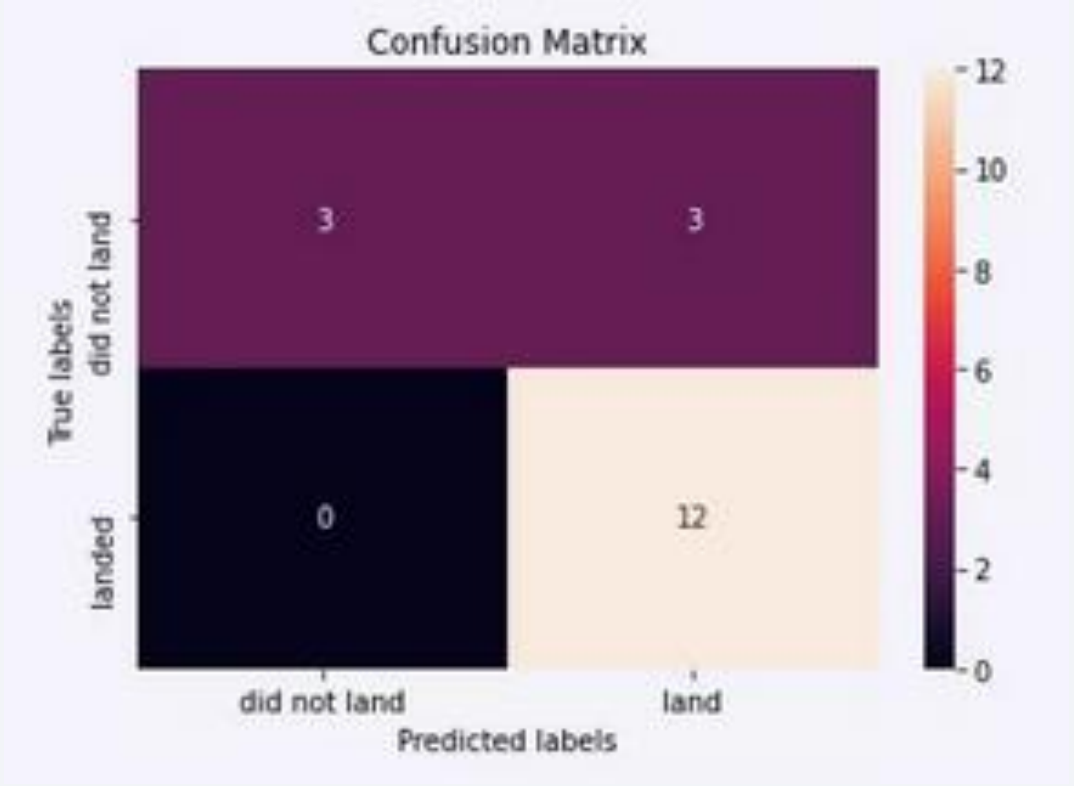
Section 5

# Predictive Analysis (Classification)

- The DecisionTreeClassifier has the highest prediction score.



- The false positives of the Decision Tree Classifier confusion matrix are the largest category, meaning unsuccessful landings were marked as successful.



- In conclusion, low weighted payloads perform better than heavier payloads overall.
- Launch success rates are showing an increasing trend per year.
- Orbits ES-L1, GEO, HEO, SSO and VLEO show the highest success rates.
- KSC LC 39A was the most successful launch across all sites.
- We can predict via Decision Tree Classifier that these launches will only get more successful with time.

# Thank you!