IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Bruno Moreno
02/10/2024

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis 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:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

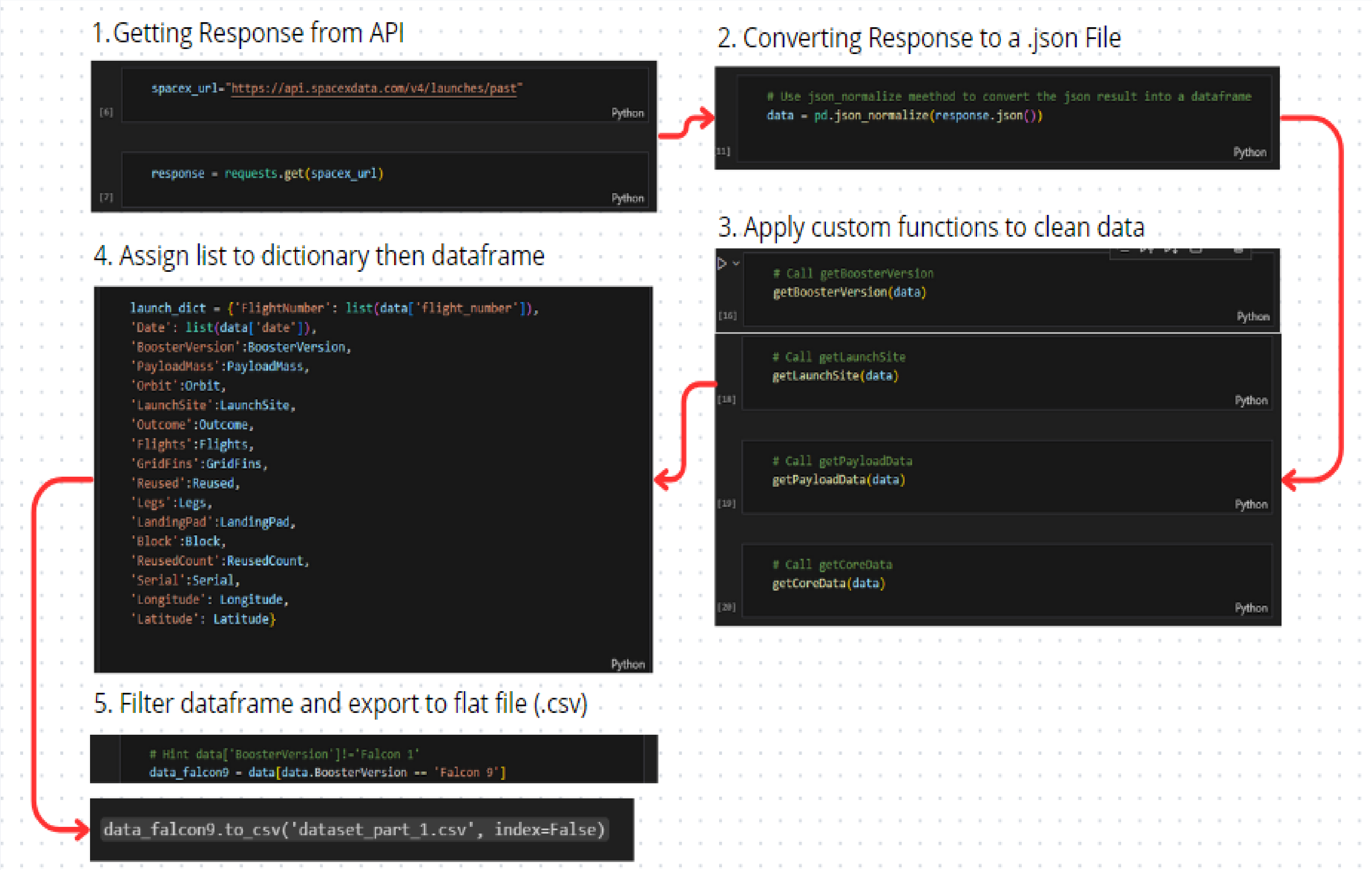
# Data Collection

- The following datasets was collected:
    - SpaceX launch data that is gathered from the SpaceX REST API.
    - This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
    - The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
    - Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.

# Data Collection – SpaceX API

- Data collection with SpaceX REST calls

https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/01_jupyter-labs-spacex-data-collection-api.ipynb



1.Getting Response from API

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

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

2. Converting Response to a .json File

```python
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3. Apply custom functions to clean data

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

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

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

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

4. Assign list to dictionary then dataframe

```python
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 flat file (.csv)

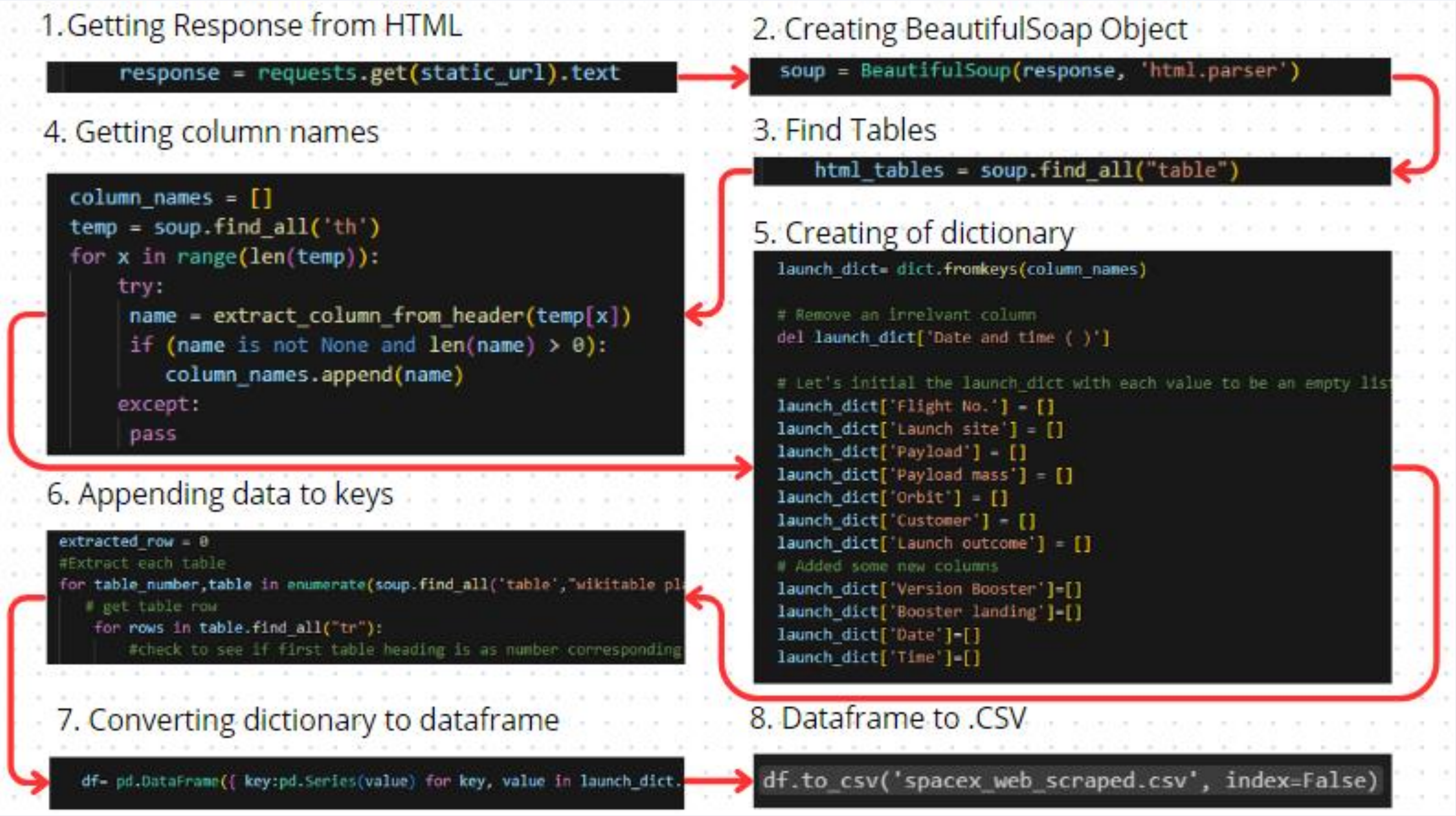
```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
```

```python
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

# Data Collection - Scraping

- Web Scrapping from Wikipedia

- https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/02_jupyter-labs-webscraping.ipynb



1. Getting Response from HTML

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

2. Creating BeautifulSoap Object

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

3. Find Tables

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

4. Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Creating of dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty lis
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable pl
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding
```

7. Converting dictionary to dataframe

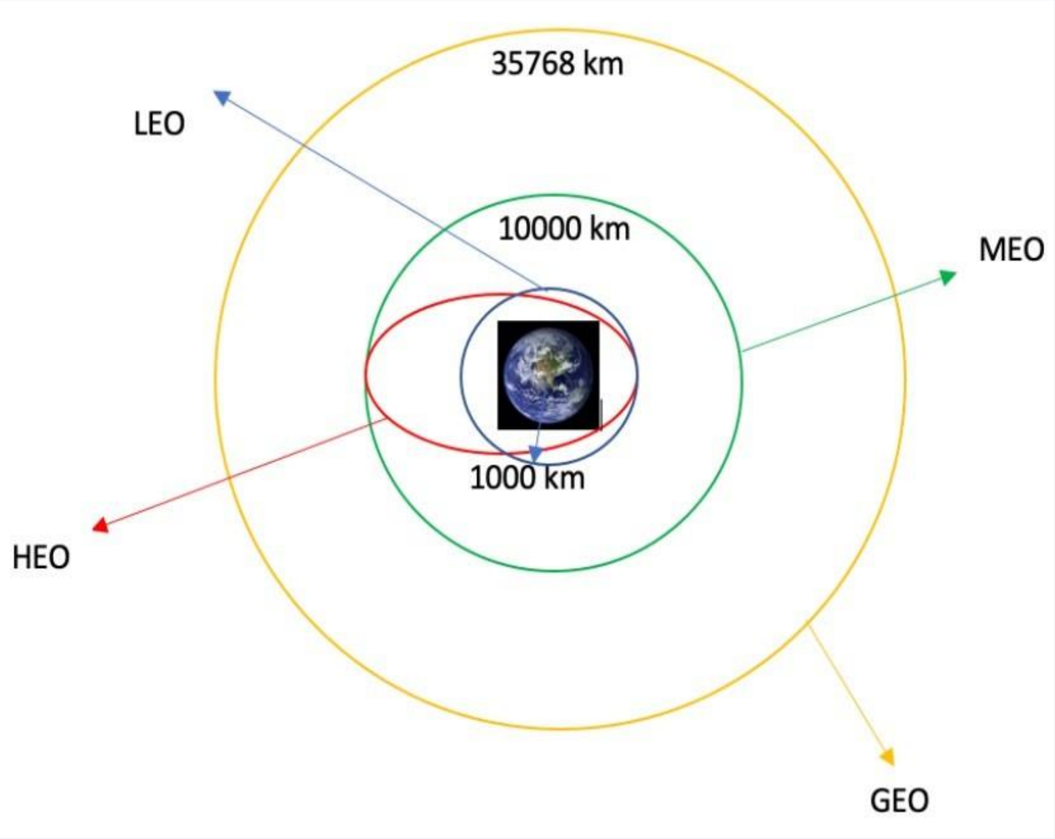
```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.
```

8. Dataframe to .CSV

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

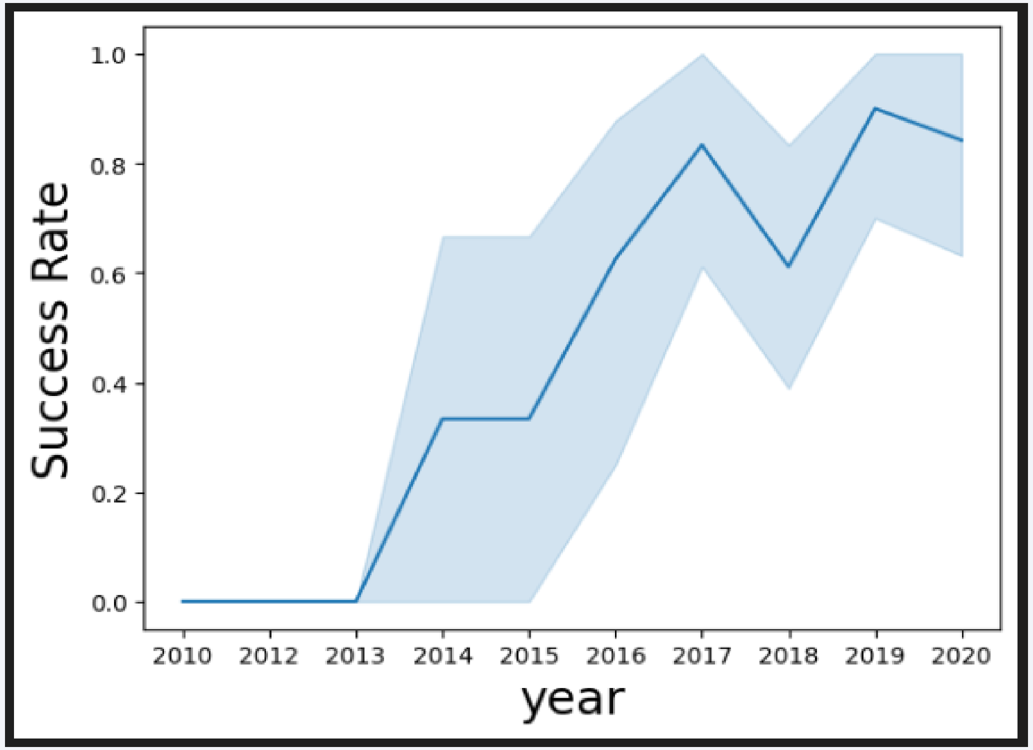
# Data Wrangling

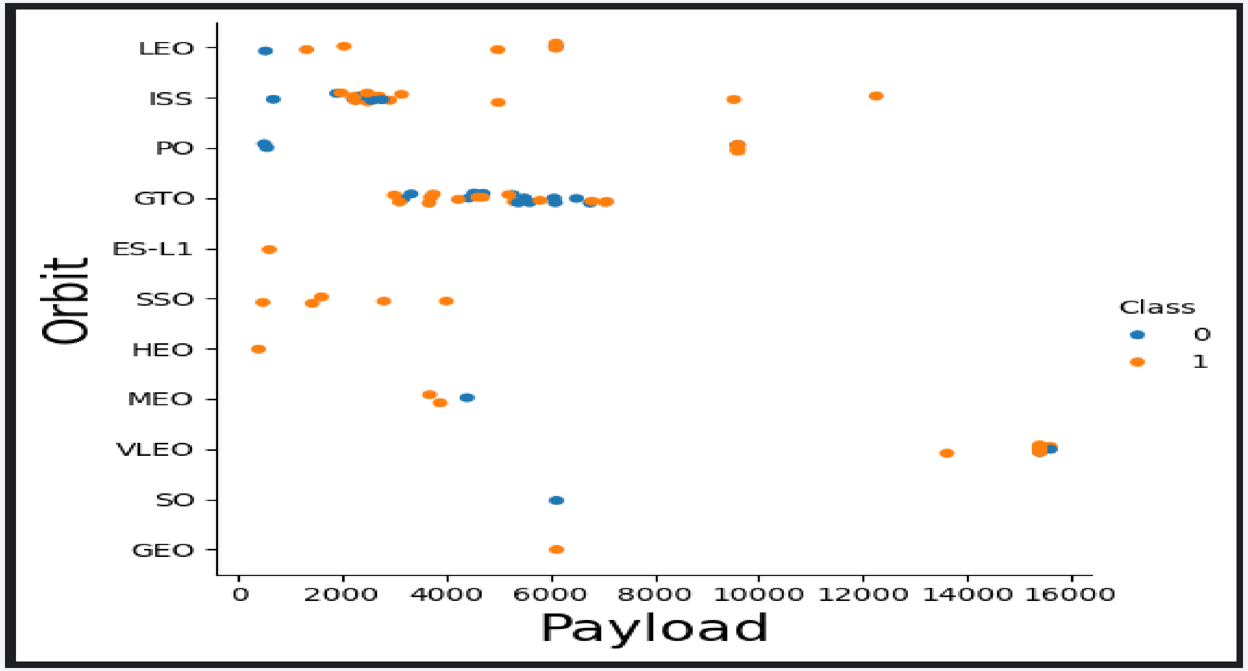
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at  each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/03_labs-jupyter-spacex-Data%20wrangling.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.





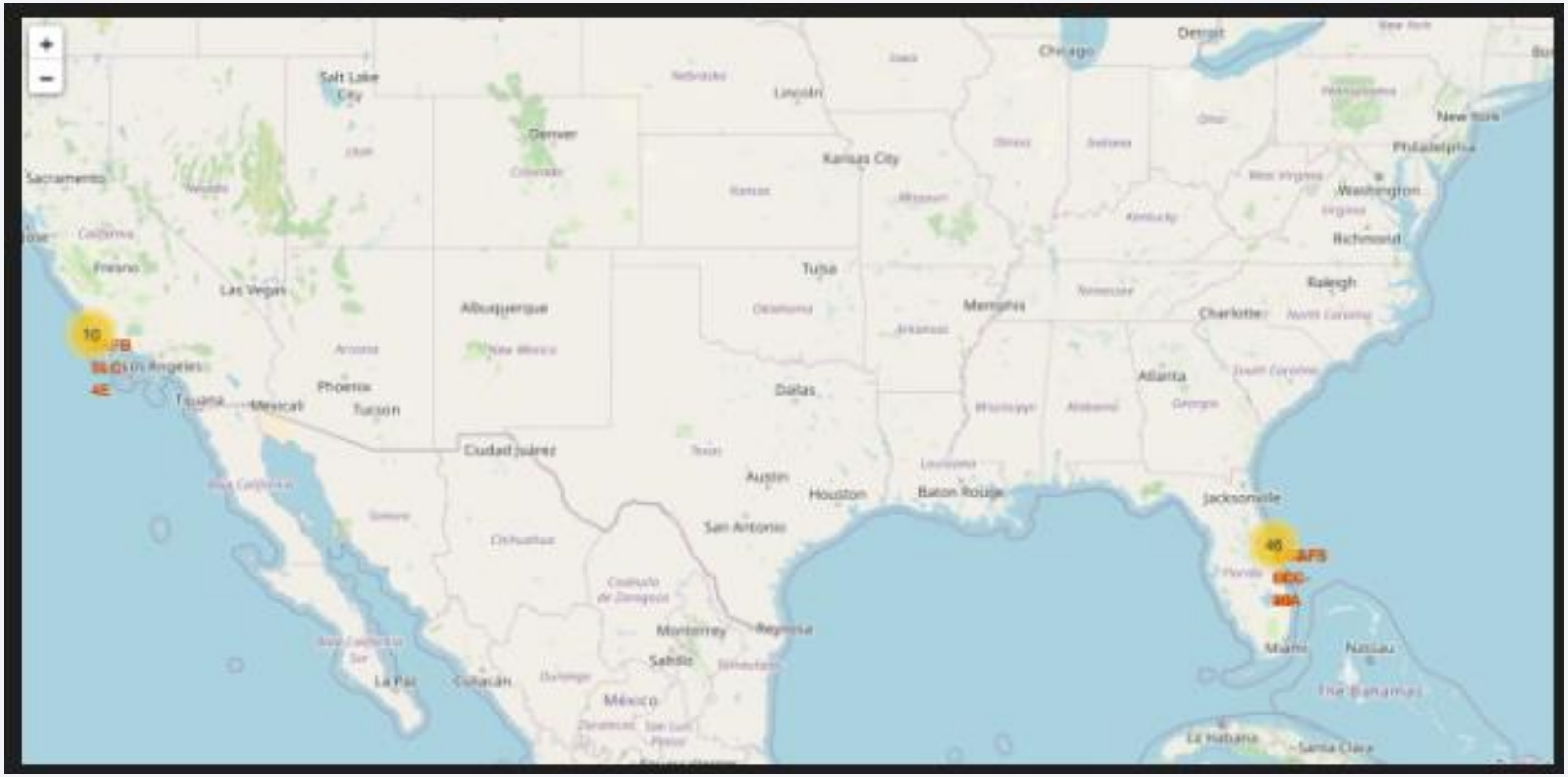
- The link of the notebook:

https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/05_edadataviz.ipynb

# EDA with SQL

- SQL queries performed include:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'KSC'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster
  - versions, launch_site for the months in year 2017
  - Ranking the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descendingorder.
  - https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/05_edadataviz.ipynb

# Build an Interactive Map with Folium



Map markers have been added to the map with aim to finding an optimal location for building a launch site

https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/06_lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash

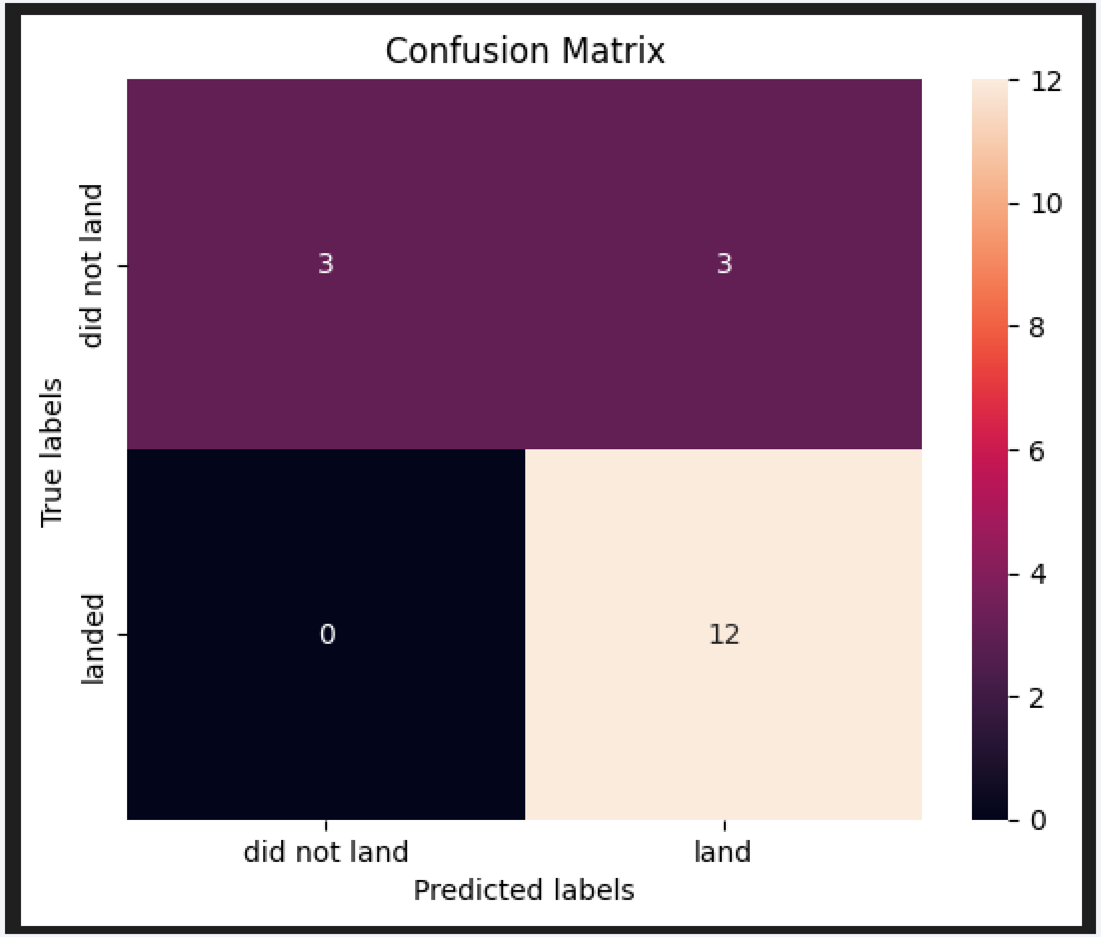
We plotted pie charts showing the total launches by a certain sites

We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

# Predictive Analysis (Classification)

- The SVM, KNN, and Logistic Regression model achieved the highest  accuracy at 83.3%, while the SVM performs the best in terms of Area Under the Curve at 0.958.

- https://github.com/brunom268/Final_Project_IBM_DataScience_SpaceX/blob/master/07_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



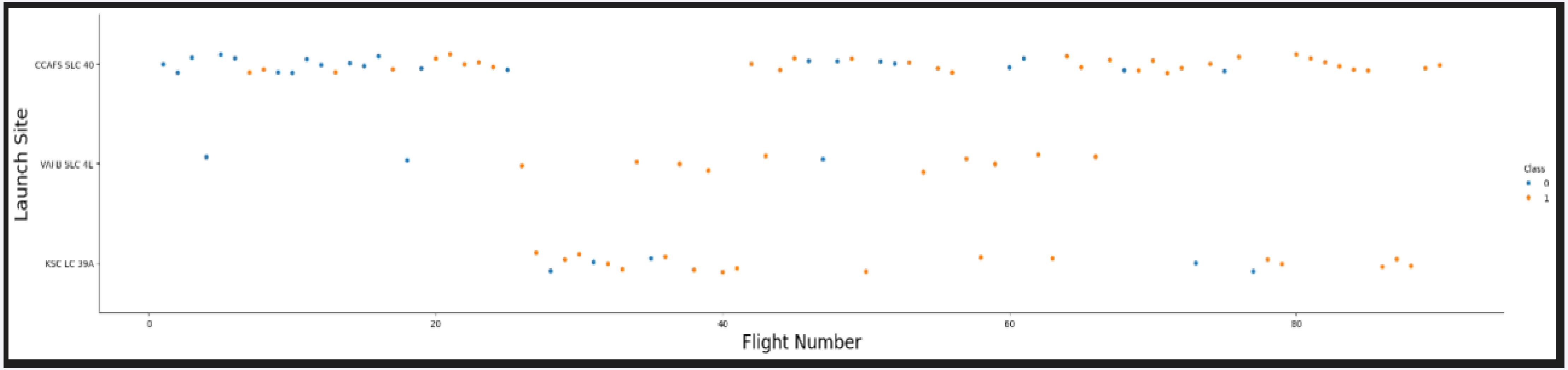
# Results

- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they  will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

Section 2

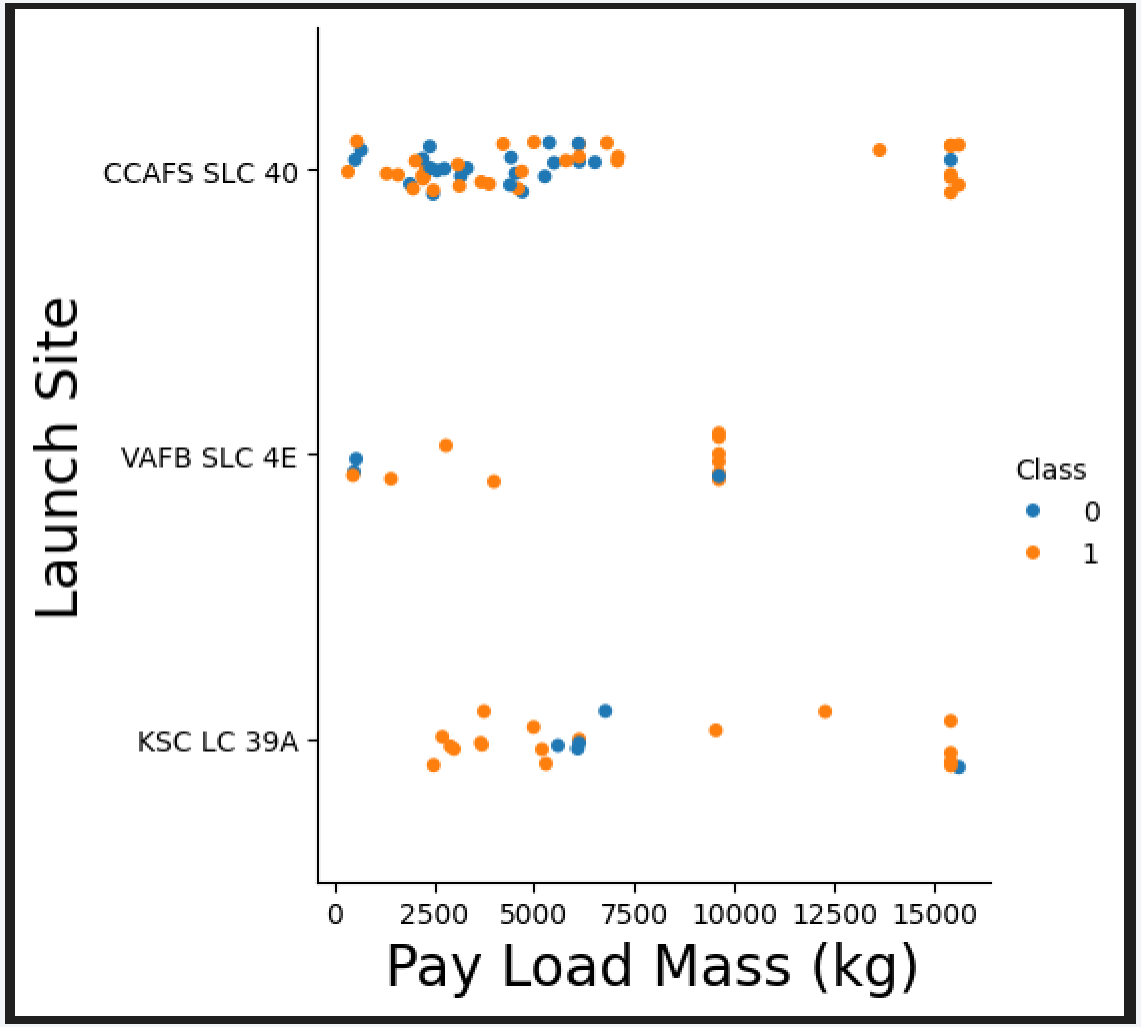
# Insights drawn from EDA

# Flight Number vs. Launch Site



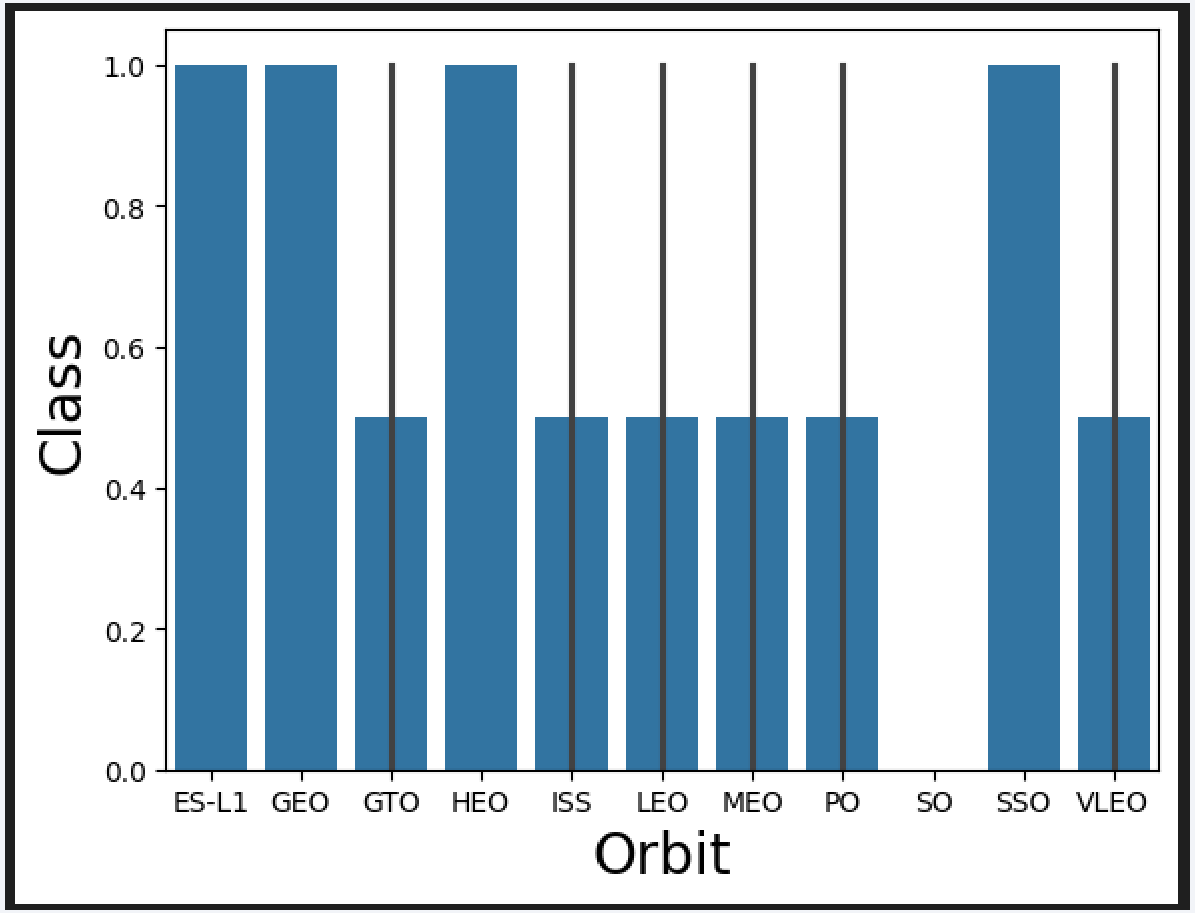
- Launches from the site of CCAFS SLC 40 are significantly higher than launches form other sites.

# Payload vs. Launch Site



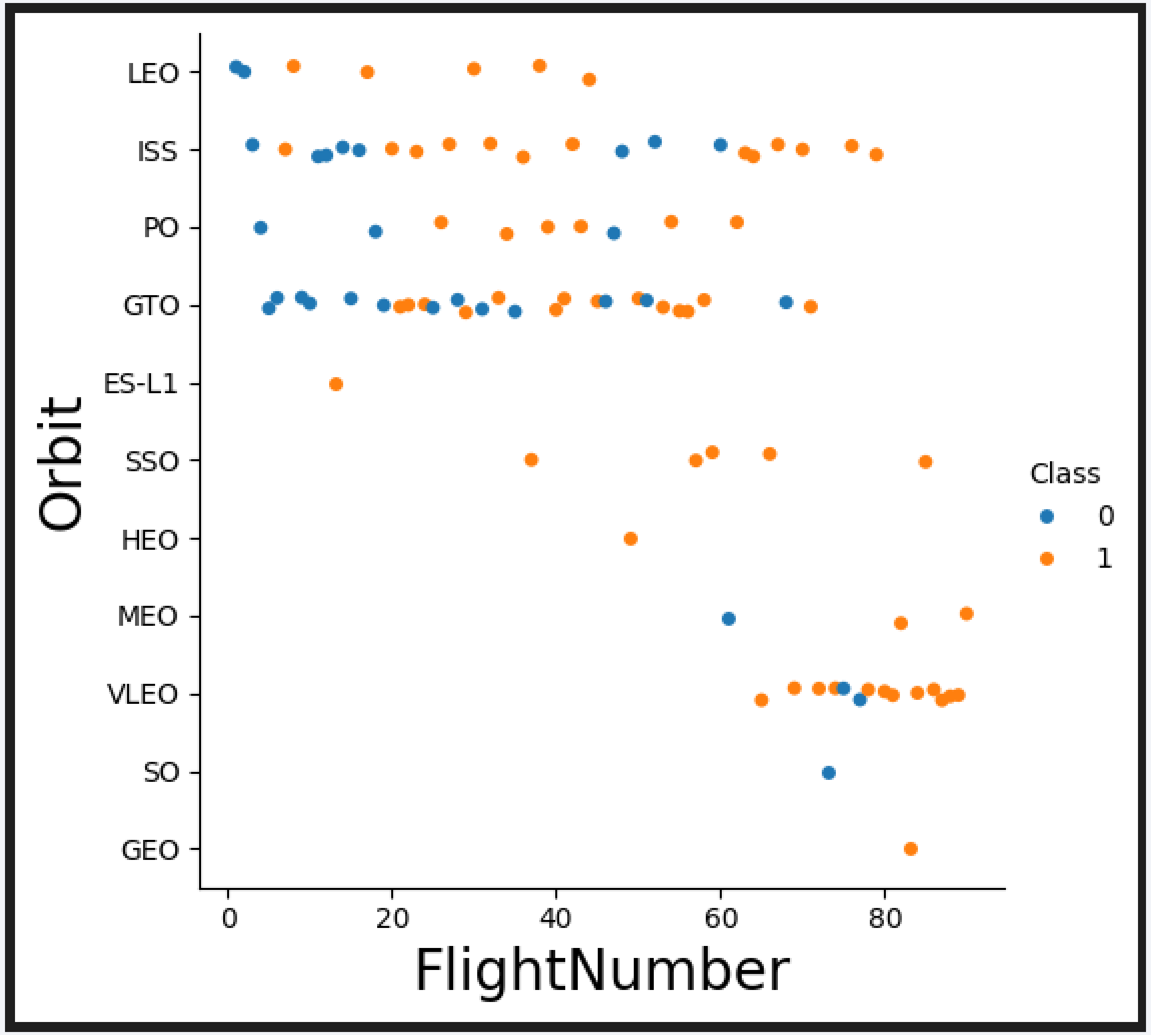
- The majority of lPay Loads with lower  Mass have been launched from CCAFS SLC 40.

# Success Rate vs. Orbit Type



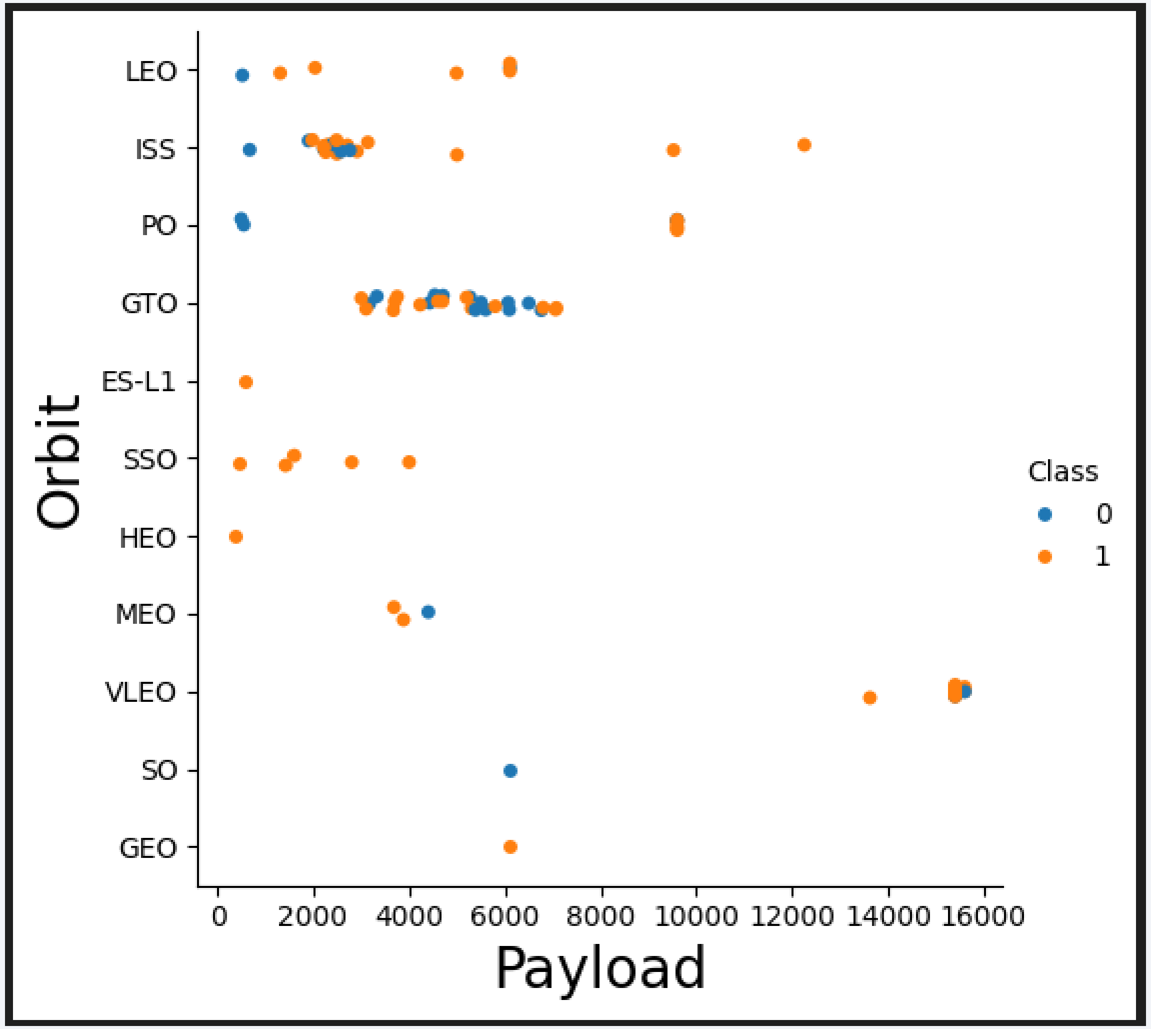
- The orbit types of ES-L1, GEO, HEO, SSO are among the highest success rate.

# Flight Number vs. Orbit Type



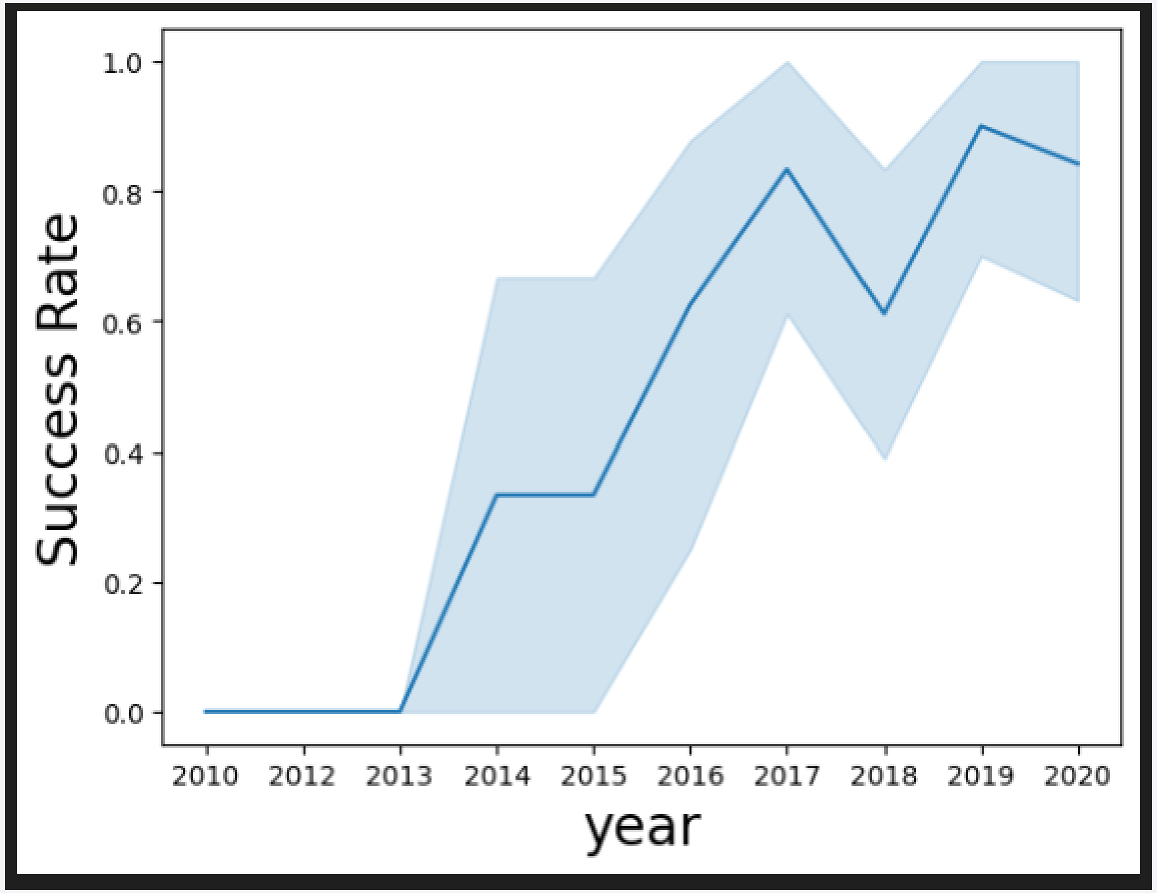
- A trend can be observed of shifting to VLEO launches in recent years.

# Payload vs. Orbit Type



- There are strong correlation between  ISS and Payload at the range around  2000, as well as between GTO and the range of 4000-8000.

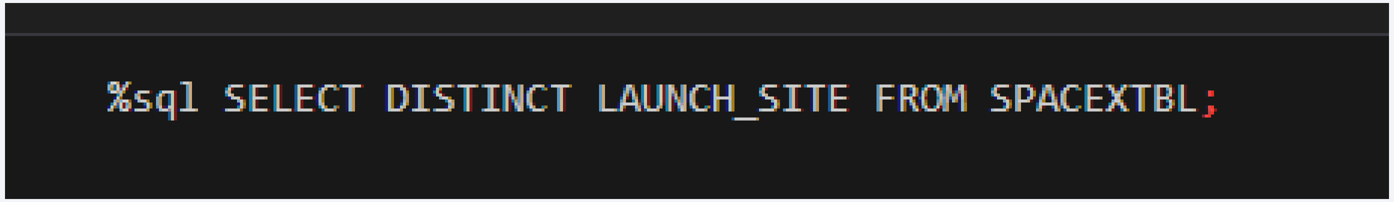
# Launch Success Yearly Trend



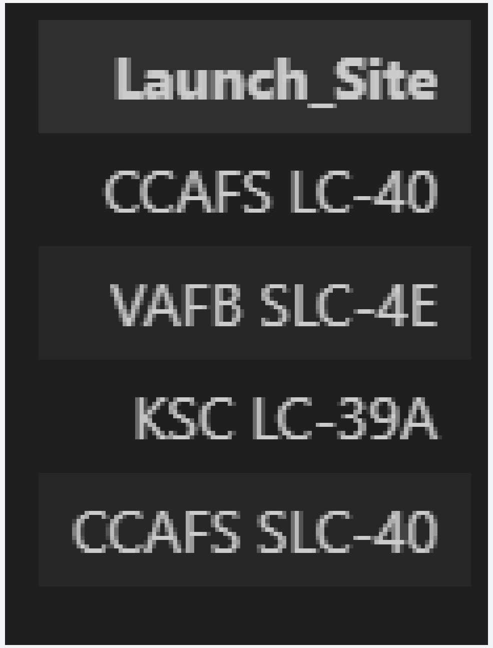
- Launch success rate has increased significantly since 2013 and has stablised since 2019, potentially due to advance in technology and lessons  learned.

# All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the  SpaceX data.

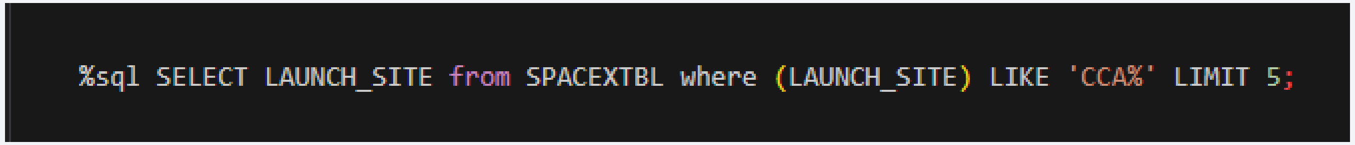


```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

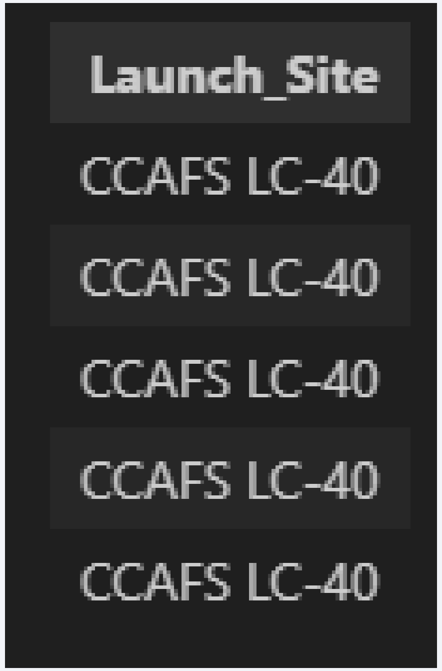


| Launch_Site |
|---|
| CCAFS LC-40 |
| VAFB SLC-4E |
| KSC LC-39A |
| CCAFS SLC-40 |

# Launch Site Names Begin with 'CCA'

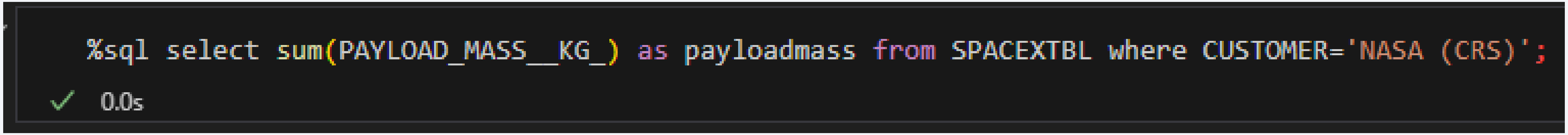


```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```



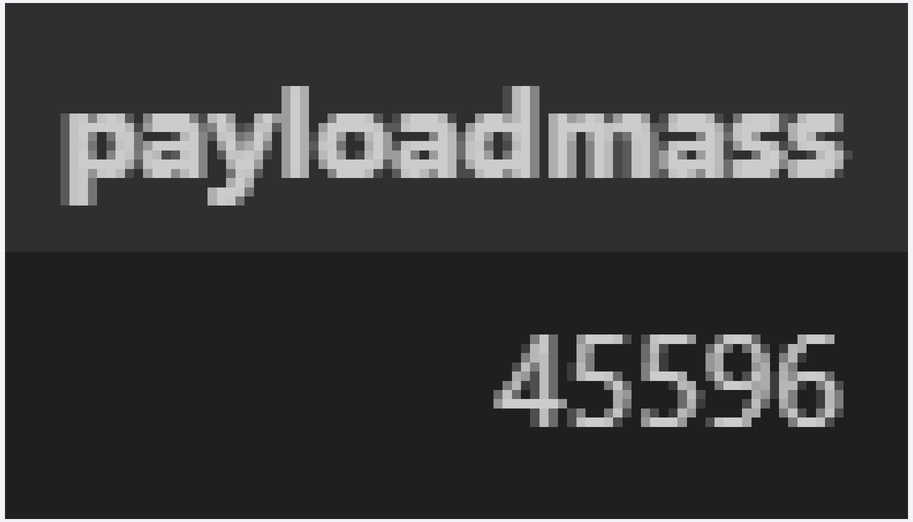
| Launch_Site |
|---|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

# Total Payload Mass



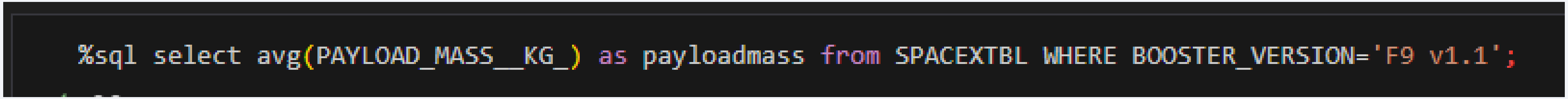
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where CUSTOMER='NASA (CRS)';
```

✓ 0.0s

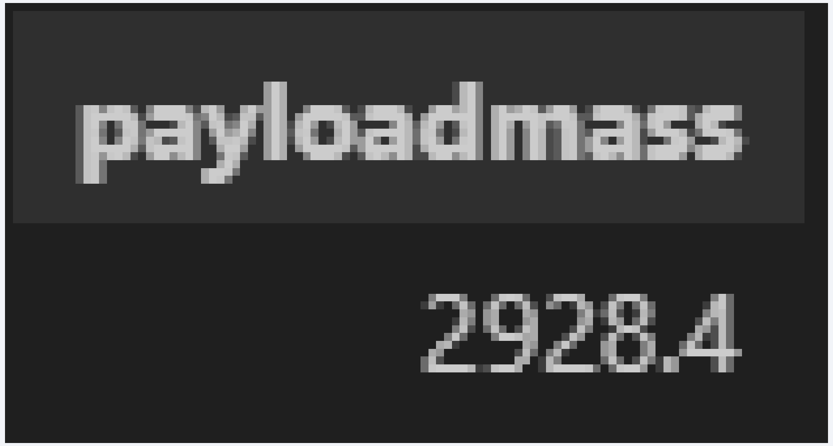


| payloadmass |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

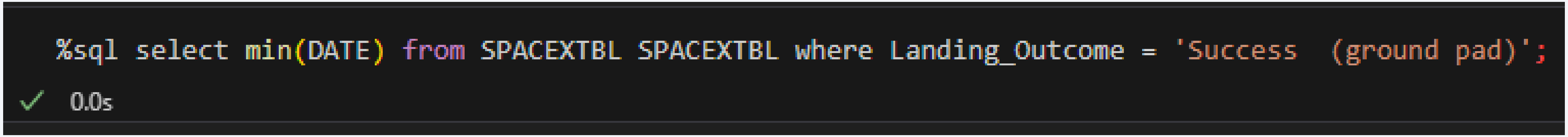


```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1';
```

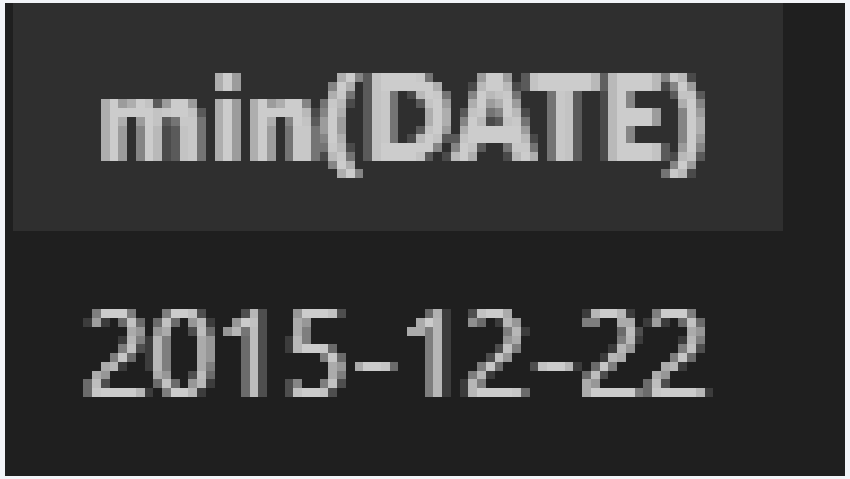


```
payloadmass
     2928.4
```

# First Successful Ground Landing Date

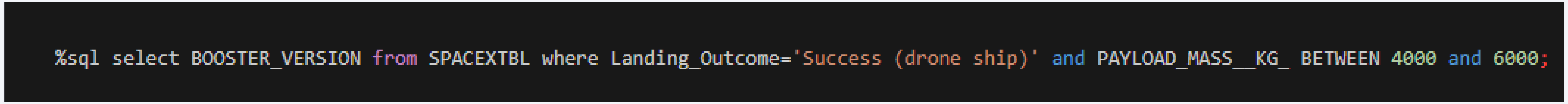


```
%sql select min(DATE) from SPACEXTBL SPACEXTBL where Landing_Outcome = 'Success  (ground pad)';
✓ 0.0s
```

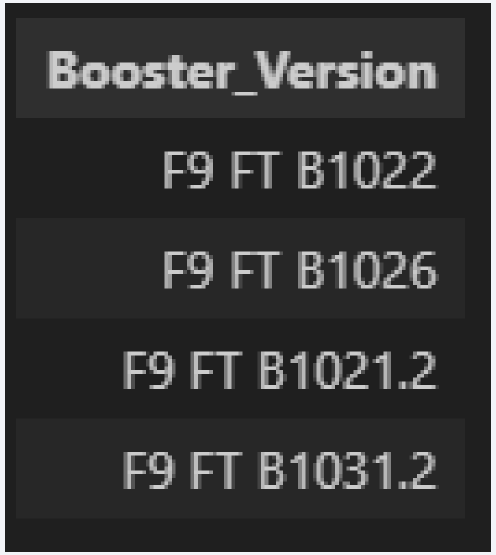


```
min(DATE)
2015-12-22
```

# Successful Drone Ship Landing with Payload between 4000 and 6000

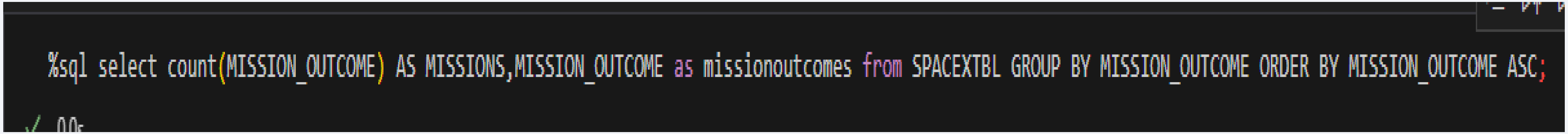


```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

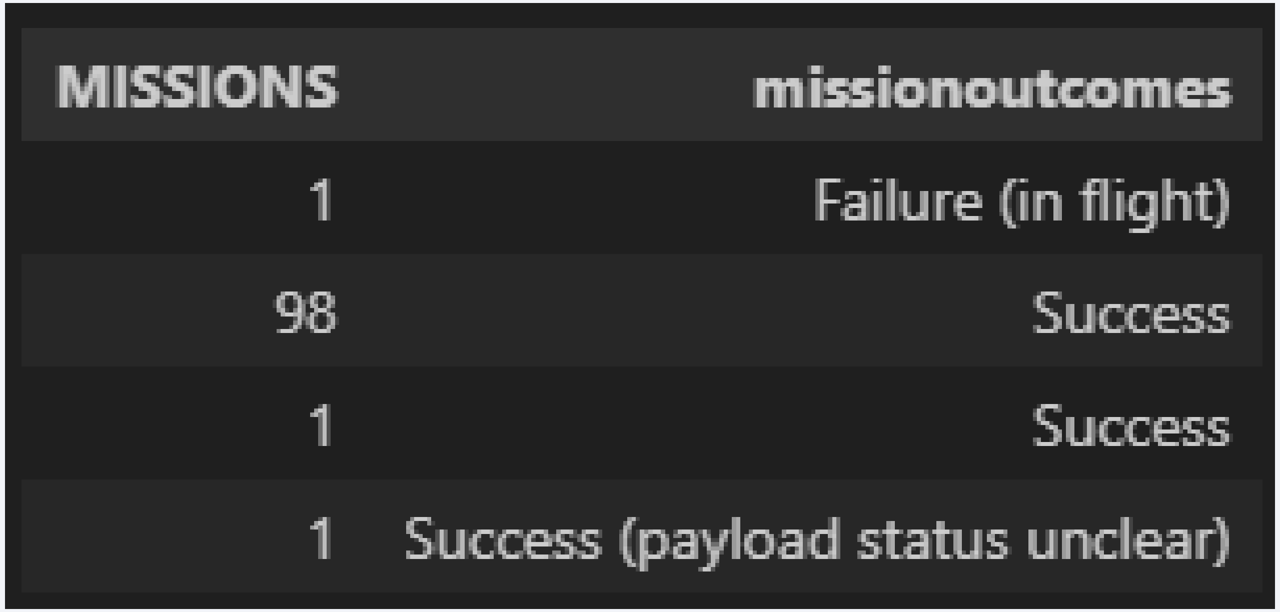


| Booster_Version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

# Total Number of Successful and Failure Mission Outcomes

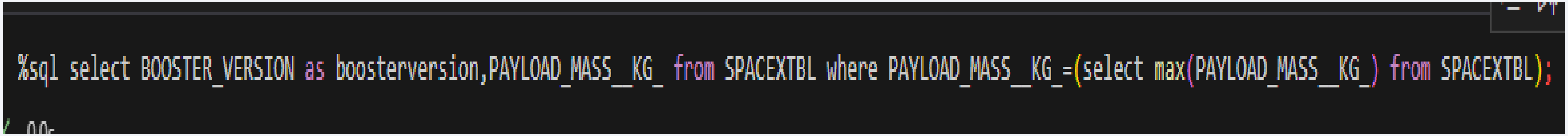


```
%sql select count(MISSION_OUTCOME) AS MISSIONS,MISSION_OUTCOME as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME ASC;
✓ 0.0s
```



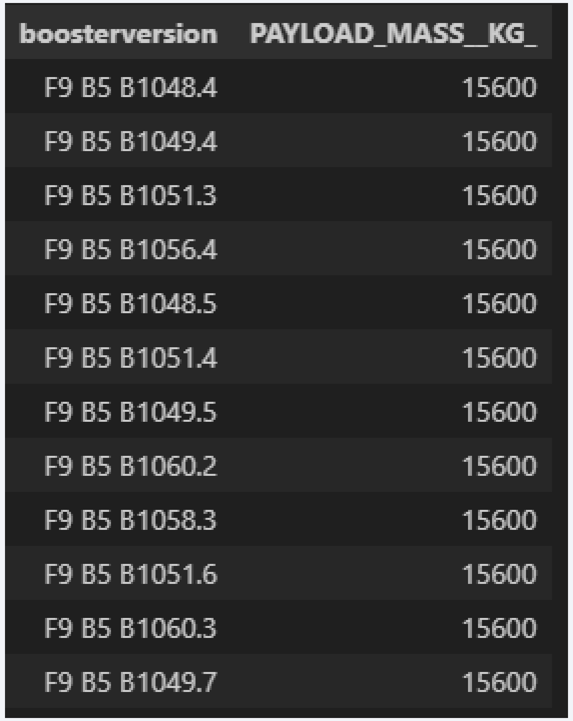
| MISSIONS | missionoutcomes |
|---:|---:|
| 1 | Failure (in flight) |
| 98 | Success |
| 1 | Success |
| 1 | Success (payload status unclear) |

# Boosters Carried Maximum Payload



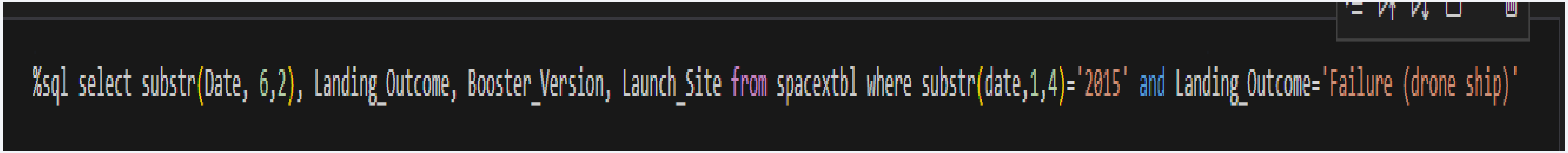
```
%sql select BOOSTER_VERSION as boosterversion,PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

✓ 0.0s

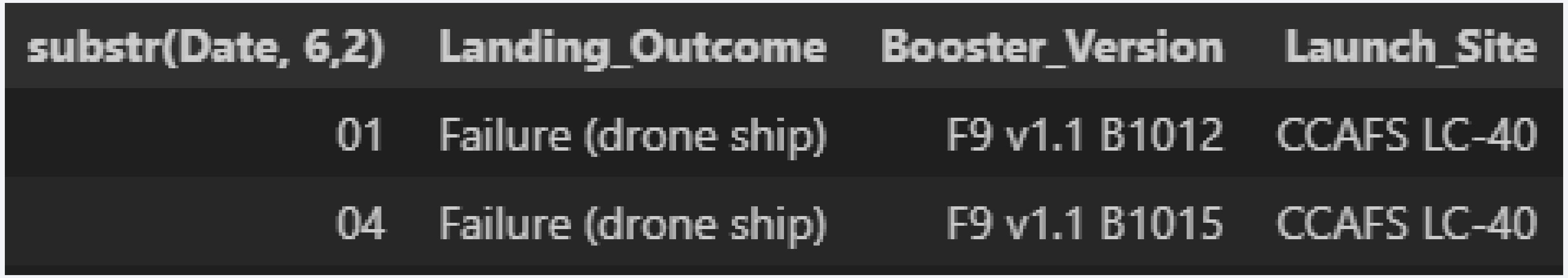


| boosterversion | PAYLOAD_MASS__KG_ |
|---|---:|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

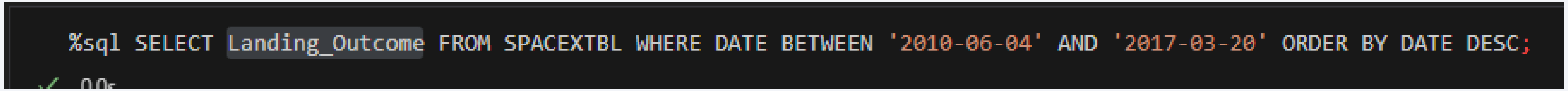


```
%sql select substr(Date, 6,2), Landing_Outcome, Booster_Version, Launch_Site from spacextbl where substr(date,1,4)='2015' and Landing_Outcome='Failure (drone ship)'
```

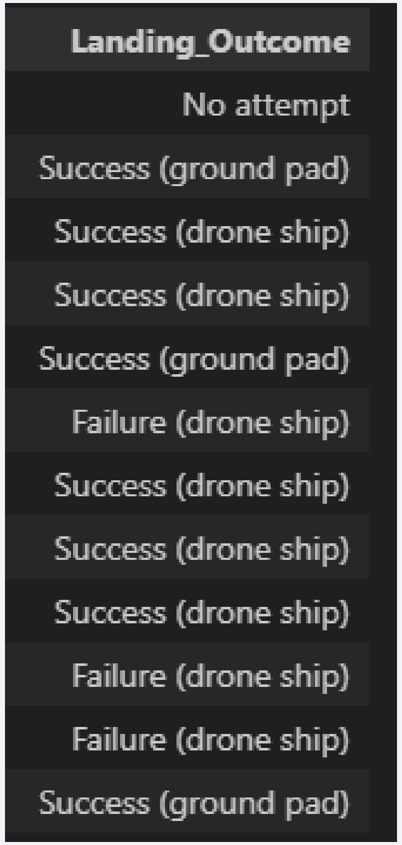


| substr(Date, 6,2) | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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



```
%sql SELECT Landing_Outcome FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
✓ 0.0s
```

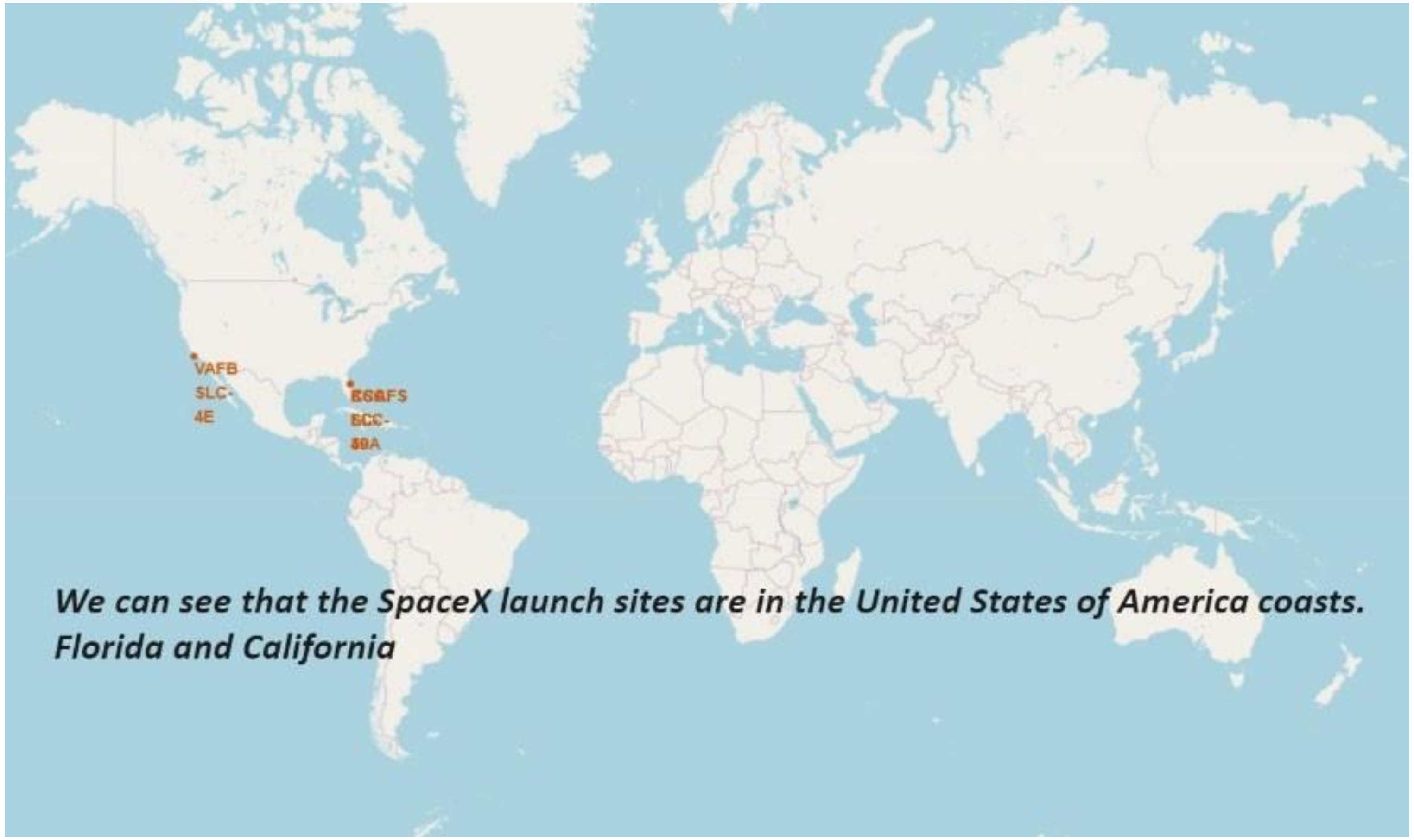


| Landing_Outcome |
|---:|
| No attempt |
| Success (ground pad) |
| Success (drone ship) |
| Success (drone ship) |
| Success (ground pad) |
| Failure (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Failure (drone ship) |
| Failure (drone ship) |
| Success (ground pad) |

Section 3

# Launch Sites Proximities Analysis

# All launch sites global map markers



***We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California***

# <Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

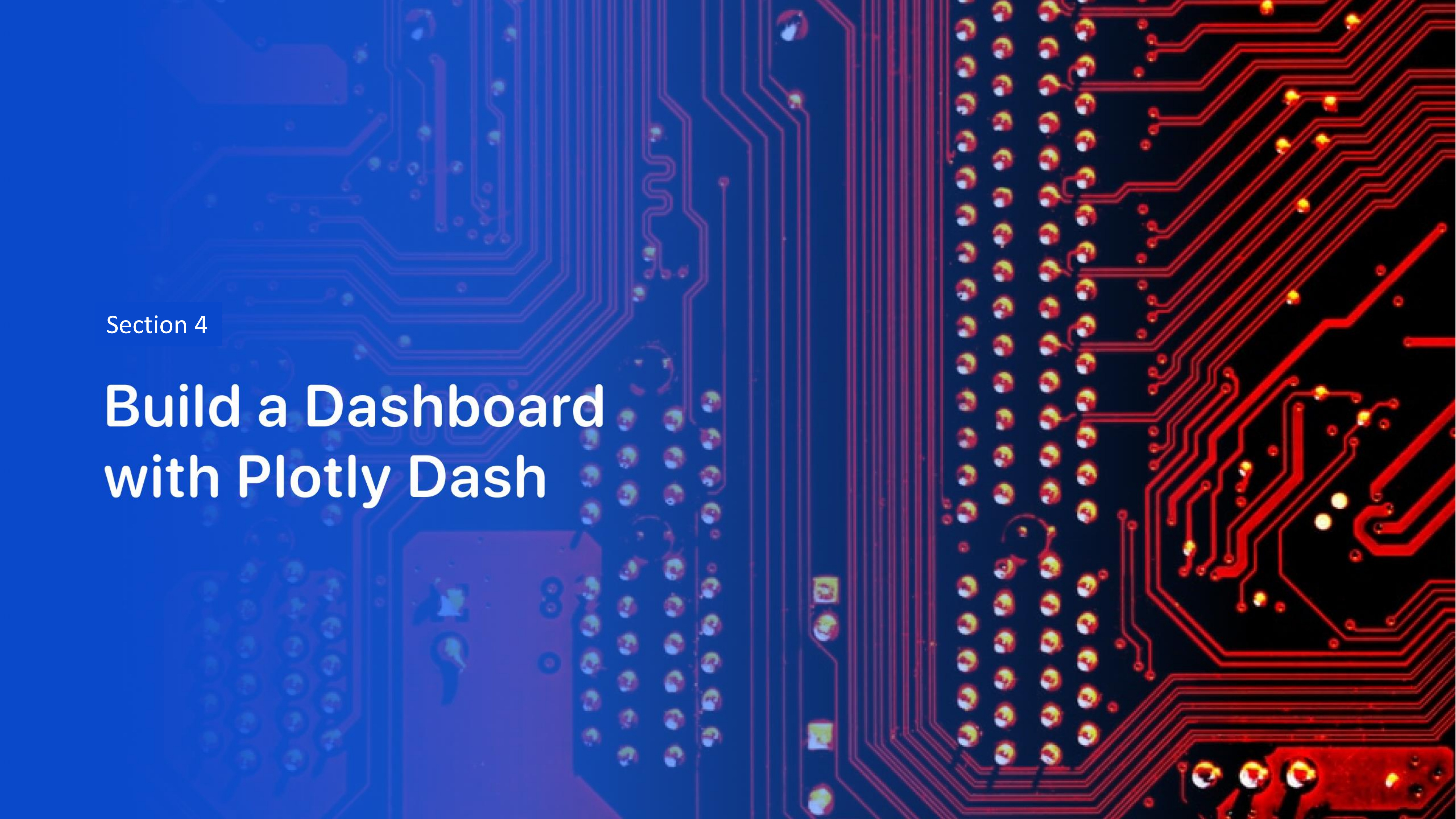
- Explain the important elements and findings on the screenshot

# <Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot



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

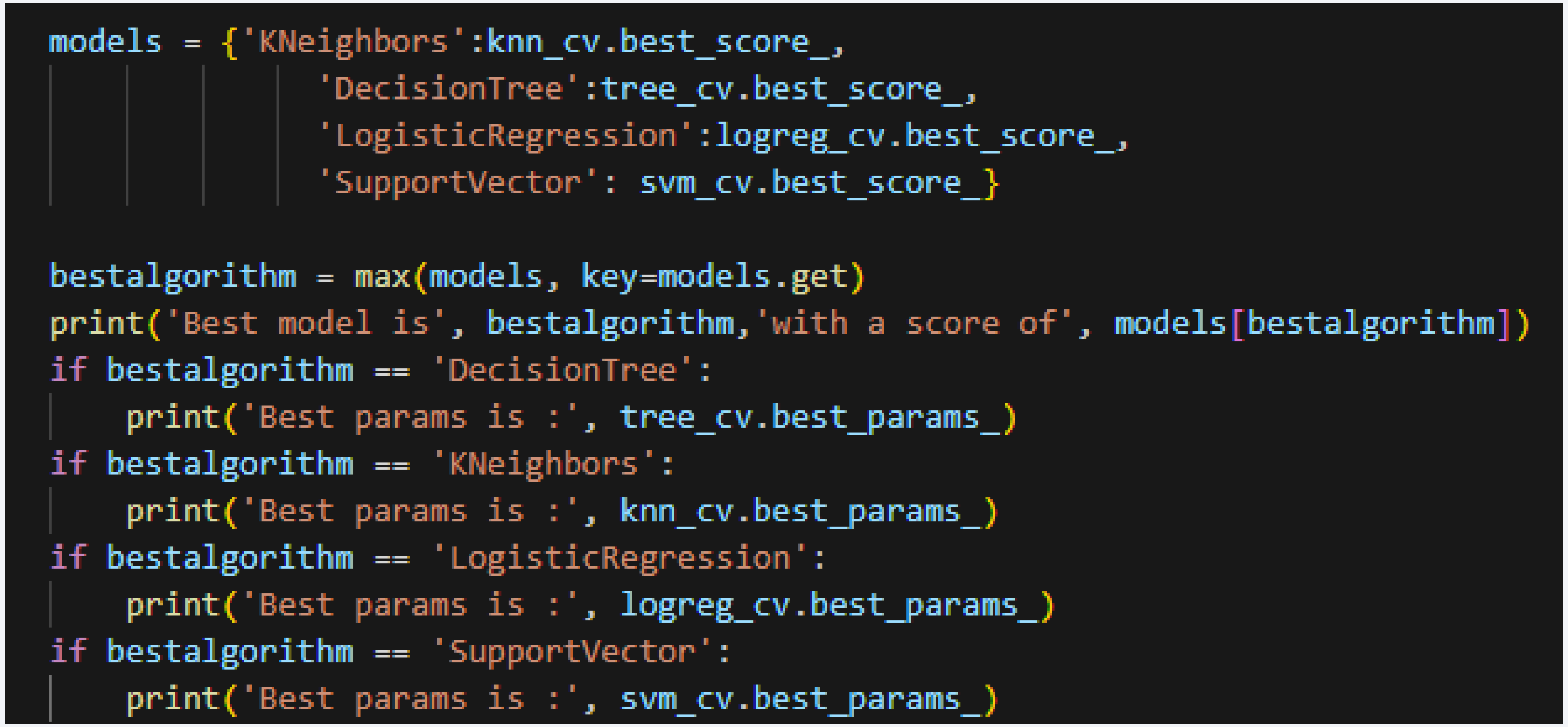
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- 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



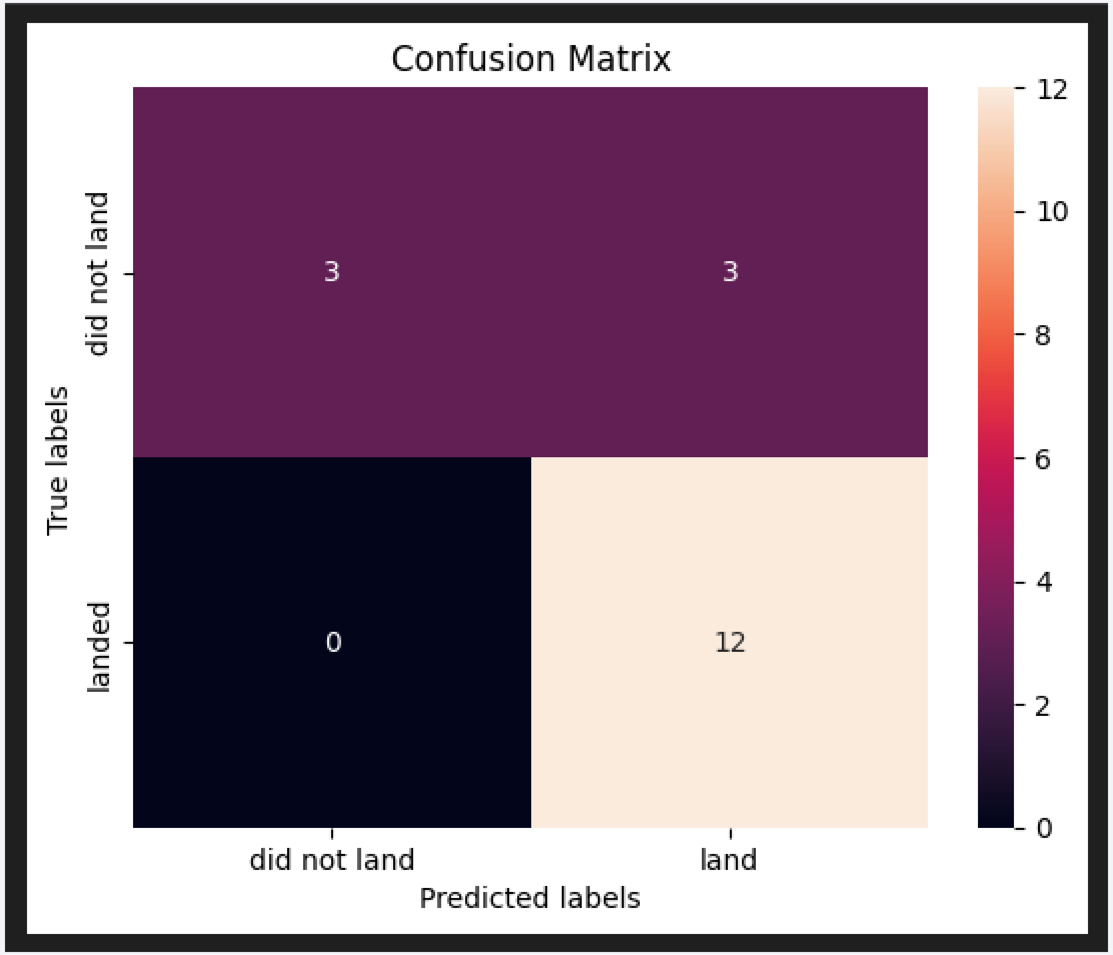
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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_)
```

The decision tree classifier is the model with the highest classification accuracy

# Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives. So, unsuccessful landing marked as  successful landing by the classifier.



# Conclusions

- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Launch success rate started to increase in 2013 till 2020.
- The success rates for SpaceX launches is directly proportional time in  years they will eventually perfect the launches.
- KSC LC-39A had the most successful launches of any sites.

# Thank you!