# IBM Data Science Capstone Project


by Aleksej Talstou

27.12.2024

# OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

# EXECUTIVE SUMMARY

- Data collection via API
- Data collection with Web Scrapping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analysis with Folium
- Data Visualization with Dash
- Machine Learning Prediction

# INTRODUCTION

**Project 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. This information can be used if an alternate company wants to bid against Space X for a rocket launch.

**Key questions to answer:**

What factors influence the successful landing of the first stage?

What analytical approaches and machine learning models can be used to predict the outcome of the landing better?

# METHODOLOGY

# METHODOLOGY

1. Data Collection:
   - Data Collection from SpaceX API.
   - Data Collection via Web Scrapping from Wikipedia.
2. Data Wrangling:
   - Compiling and Cleaning the Data.
   - Preparing the Data for further classification with ML modes with One Hot Encoding.
3. Exploratory Data Analysis with Data Visualization and SQL.
4. Interactive Visual Analysis with Folium and Dash.
5. Predictive Data Analysis with Various Classification Models.

# Data Collection from SpaceX API

The first source of data collection was SpaceX API. The json file was obtained with the get request and normalized.

The obtained data was organized into a dictionary and later a dataframe.

The dataframe was filtered to contain information regarding Falcon 9 launches and the missing values of the Payload Mass column were filled with mean values.

[34]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response.json()
data = pd.json_normalize(response.json())
```

[16]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[18]:
```
response = requests.get(spacex_url)
```

Check the content of the response

[20]:
```
print(response.content)
```

b'[{"fairings":{"reused":false,"recovery_attempt":false,"rec
all":"https://images2.imgbox.com/94/f2/NN6Ph45r_o.png","larg
o.png"},"reddit":{"campaign":null,"launch":null,"media":null

# Data Collection via Web Scrapping

From the Wikipedia page, the records on Falcon 9 launches were extracted using BeautifulSoup.

They were parsed into the dataframe.

First, the names of the columns were organized into the dictionary as keys, second, the dictionary was filled with values from the table. The dataframe was saved as a csv file.

```
[5]: # 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`

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

```
[21]: extracted_row = 0
      #Extract each table
      for table_number,table in enumerate(soup.find_all('table',"wikita
         # get table row
          for rows in table.find_all("tr"):
              #check to see if first table heading is as number corresp
              if rows.th:
                  if rows.th.string:
                      flight_number=rows.th.string.strip()
                      flag=flight_number.isdigit()
              else:
```

# Data Wrangling

```
[55]: # Apply value_counts() on column LaunchSite
      df['LaunchSite'].value_counts()

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

```
[77]: df['Class']=landing_class
      df[['Class']].head(8)
```

[77]:

| | Class |
|---|---|
| **0** | 0 |
| **1** | 0 |
| **2** | 0 |

```
[57]: # Apply value_counts on Orbit column
      df['Orbit'].value_counts()

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

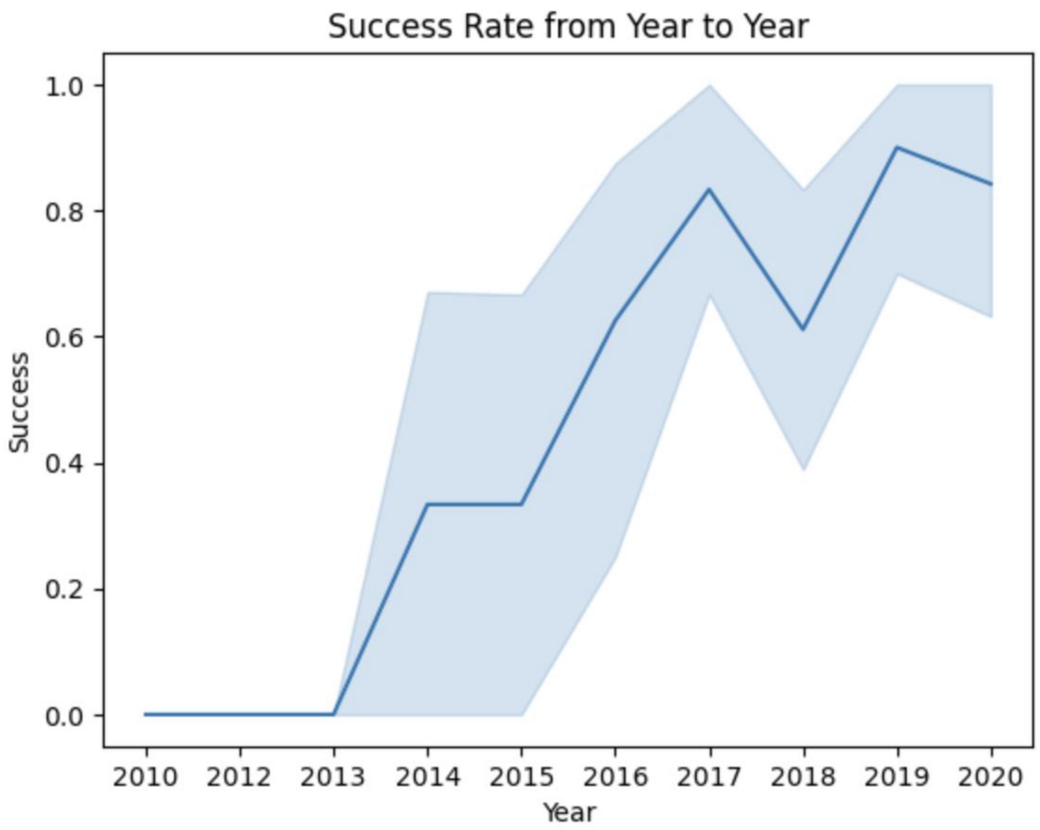
Further, the dataframe was explored by calculating the number of launches from particular sites as well as the occurances of certain orbits.

The feature Class was calculated as a result of the outcome that allowed us to indicate if the launch was successful or not.

# EDA. Data Visualization. Seaborn

In the next stage, the data was explored using visualization tools of the Seaborn library, like scatter plots, line plots, and bar charts.

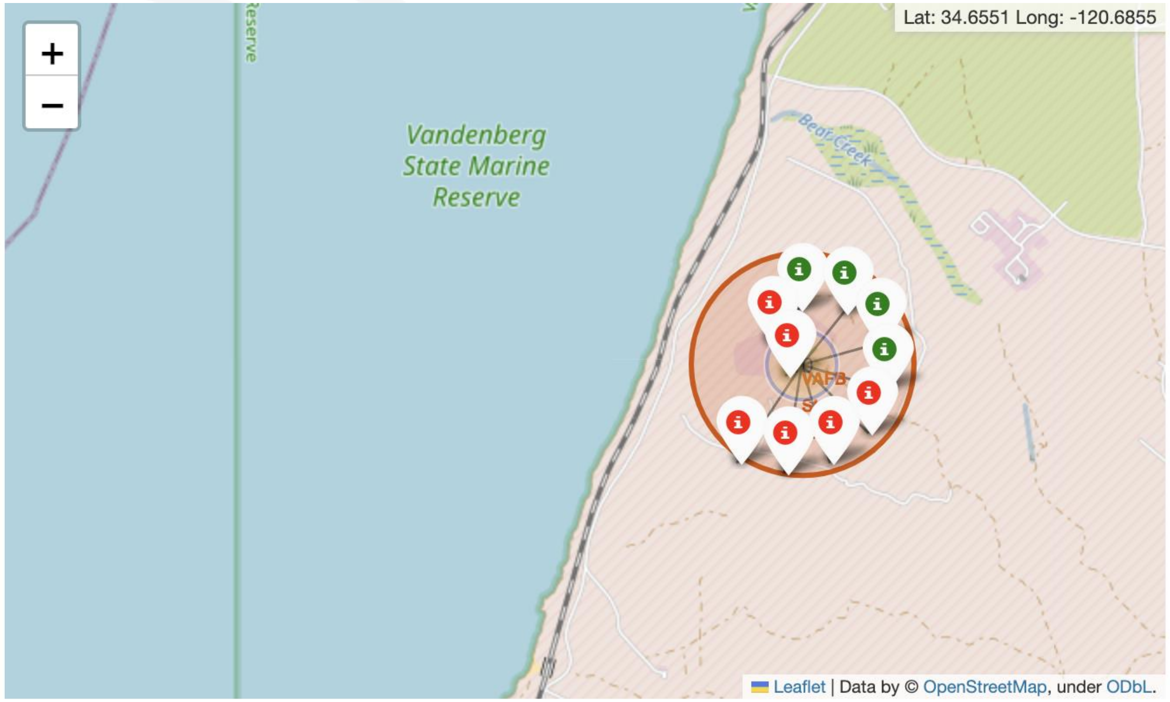
The goal was to investigate relations between different features, like the number of flights, payload mass, launch site, orbit, and others, to better understand which factors influence the success of the launches.



# EDA. Data Visualization. Folium

The Folium library allowed us to create an interactive map of all launch sites, mark the number of successful and unsuccessful launches on it as well as measure the distance from the launch site to other important locations and infrastructure such as highways, railroads, sea coast, and the nearest city. With an interactive map, one can understand the context better and this can lead to additional insights.

[66]:

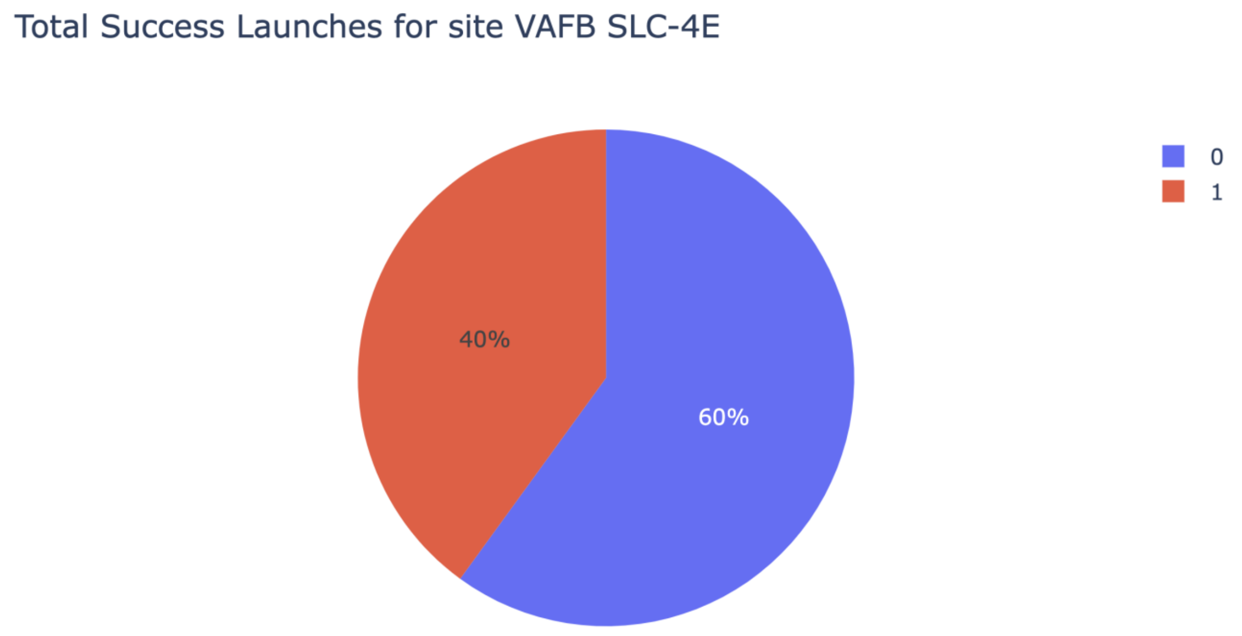


# EDA. Data Visualization. Plotly Dash

The creation of an interactive dashboard with Plotly gave us access to data visualization that could be sorted by various features.

The pie chart was showing the number of successful launches per lauch site.

The scatterplot described the relationship between the payload mass and the outcome of the flight.



# EDA. SQL

[26]:
```
%%sql

select Booster_Version
from SPACEXTABLE
where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
and Landing_Outcome like 'Success%drone ship%';
```

```
 * sqlite:///my_data1.db
Done.
```

[26]:

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

[22]:
```
%%sql

select avg(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where Booster_Version = 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

[22]:

| avg(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

Exploring the data with SQL allowed us to answer several questions and make various calculations. We were able to list Booster versions, calculate the average and total payload mass, display failed missions according to the particular landing conditions, etc.

# Predictive analysis

The target feature Class was transformed into Numpy array, normalized with StandardScaler, and split into train/test sets.

Logistic regression, SVM, Decision tree, and KNN models were fit while creating  GreedSearchCV object to calculate the optimal parameters.

Finally, accuracy was tested and a confusion matrix was created for each model.

```
[369]: Y = data['Class'].to_numpy()
       Y

[369]: array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
              1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
              1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
              1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
              1, 1])
```

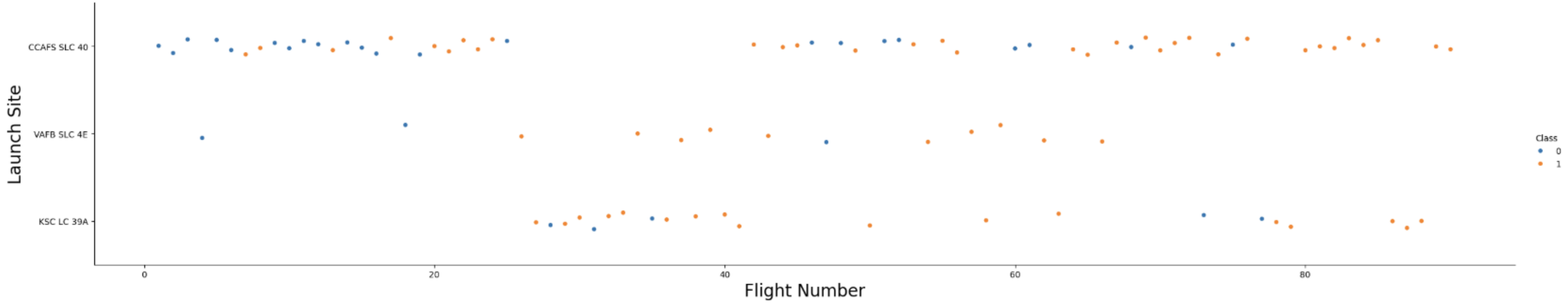
```
[379]: parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
       lr=LogisticRegression()
       logreg_cv = GridSearchCV(estimator=lr, cv=10, param_grid=parameters).fit(X_train, Y_train)
```

```
[383]: logreg_score = logreg_cv.score(X_test, Y_test)
       print("score :", logreg_score)

       score : 0.8333333333333334
```

# RESULTS

# EDA. Visualization with Seaborn

## Flight Number Vs. Launch Site

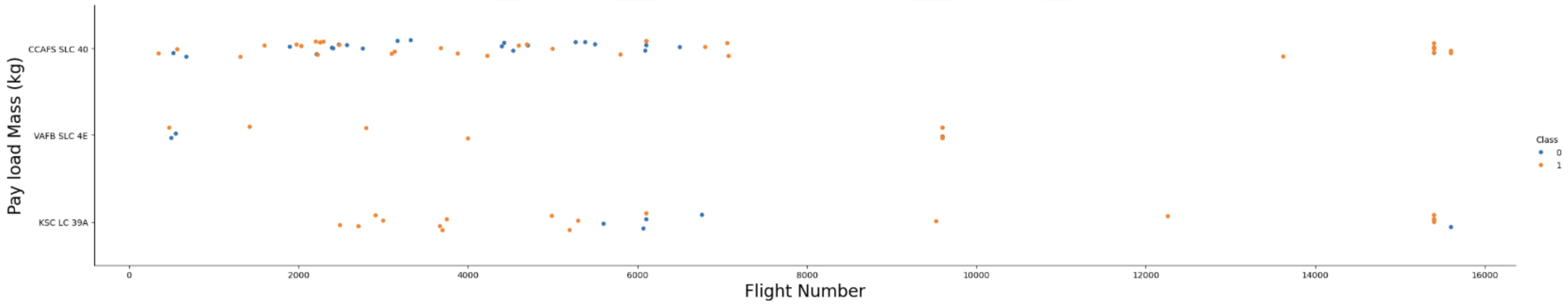



Overall, **the rates of successful missions** increase with time and the number of flights from a particular launch site.

The **largest number of flights** was launched from the CCAFS SLC 40 launch site.

# EDA. Visualization with Seaborn

## Payload Mass Vs. Launch Site




Most of the flights have a **Payload Mass below 7000**.

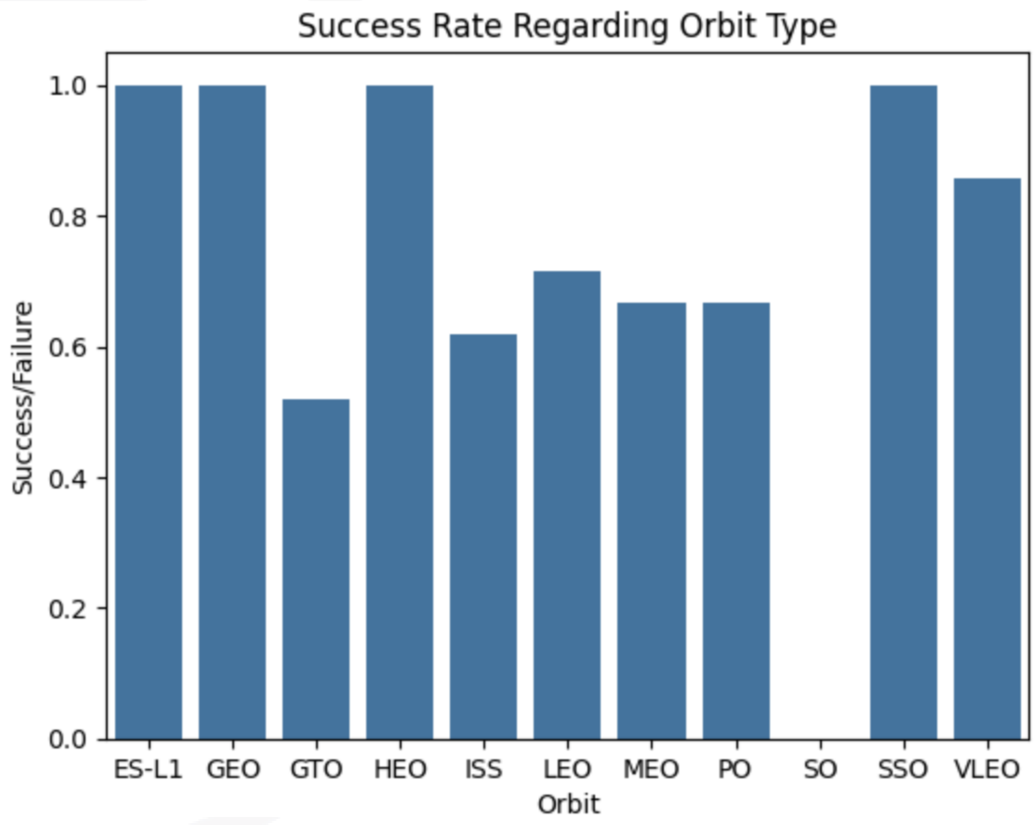
There were no **heavy payload mass flights from VAFB-SLC** with a mass higher than 10000.

# EDA. Visualization with Seaborn

## Success Rate Regarding Orbit Type

The flights to the orbits **ES-L1, GEO, HEO, SSO, and VLEO** have a **100%** success rate.

The flights to **ISS, LEO, MEO, and PO** have a similar success rate.

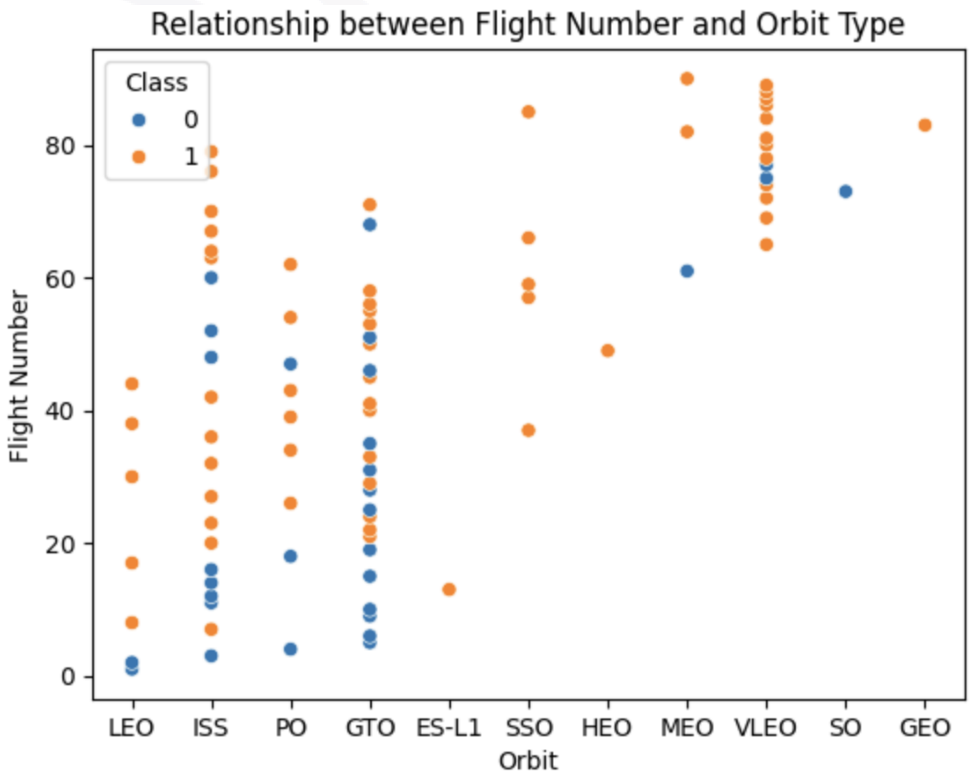


# EDA. Visualization with Seaborn

## Relationship between Flight Number and Orbit Type

In case of **LEO** success is related to the number of flights.

The success of the flights to **GTO** doesn't depend on the number of flights.

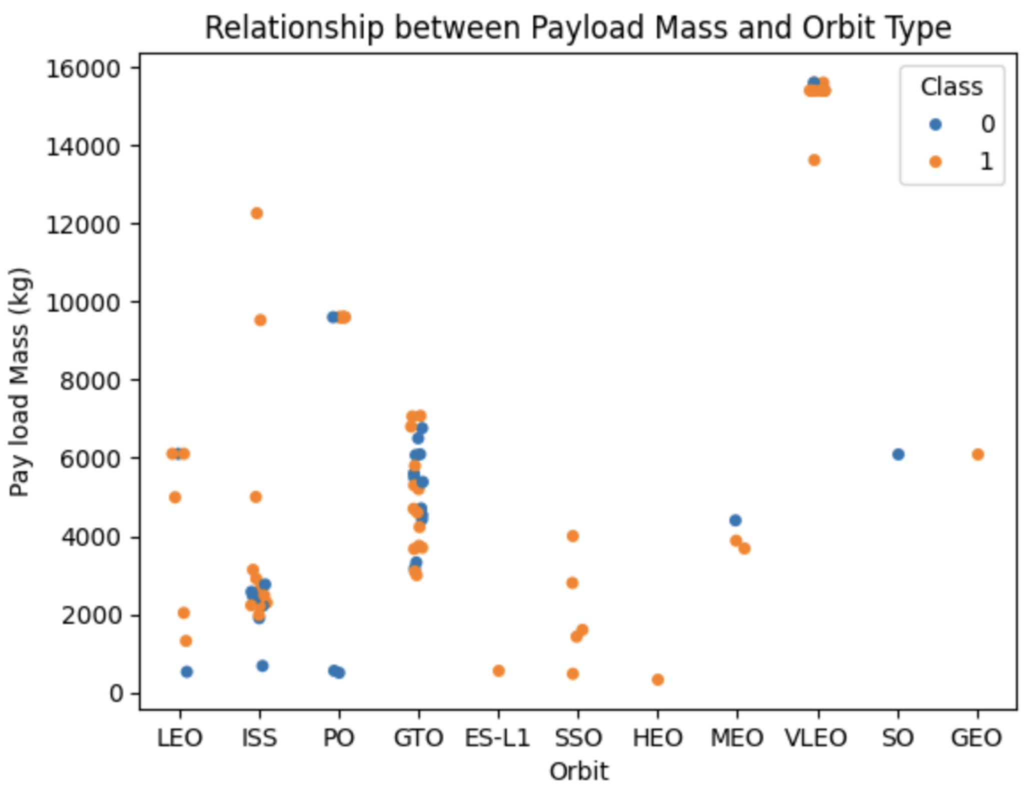


# EDA. Visualization with Seaborn

## Relationship between Payload Mass and Orbit Type

For the flights with heavy payloads, the successful or positive landing rate is higher for **Polar, LEO, and ISS**.

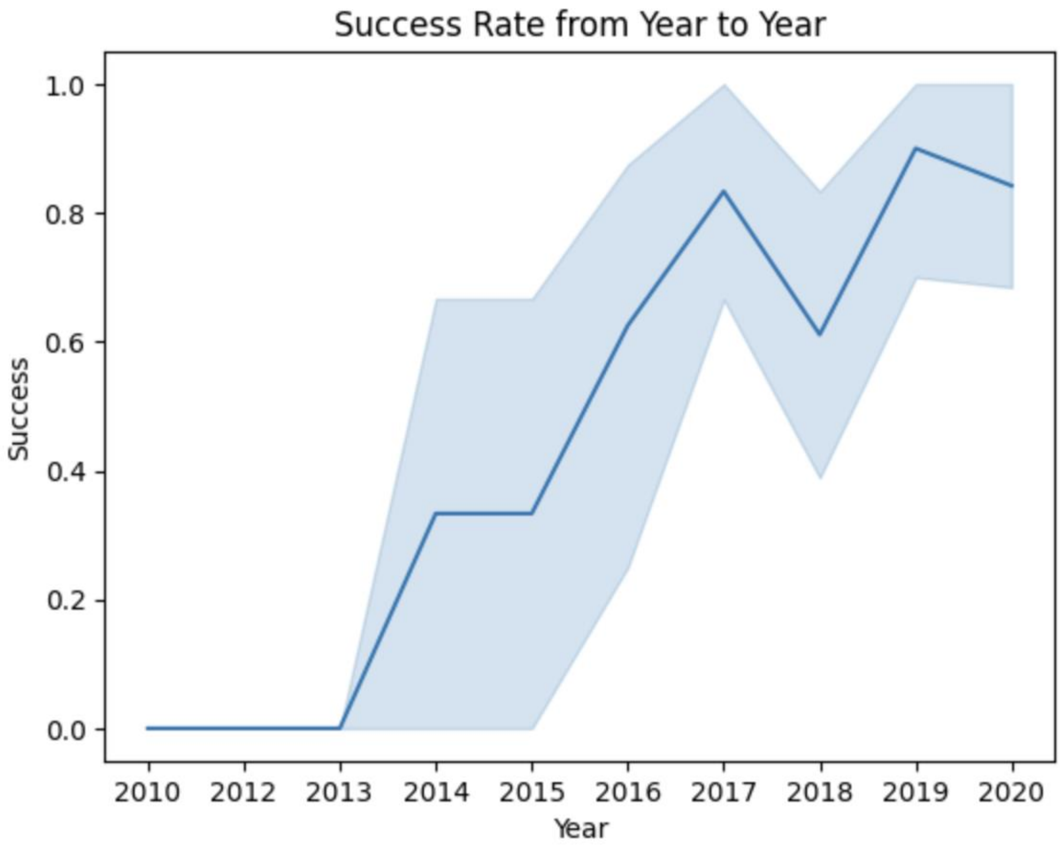
It is hard to track a dependency of the payload mass and success rate for the flights for **GTO**.



# EDA. Visualization with Seaborn

**Relationship between Payload Mass and Orbit Type**

The overall **flight success rate keeps increasing** from 2013 to 2020.



# EDA. SQL

## Launch Sites

A query to display the unique launch sites.

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

```
[16]: %%sql

SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

```
 * sqlite:///my_data1.db
Done.
```

[16]:

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

# EDA. SQL

## CCA- Launch Sites Records

[18]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 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 (parachute) |
| 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 |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

A query to display the records for the launch sites that begin with "CCA".

# EDA. SQL

## Payload Mass of NASA Boosters

A query to display the total payload mass of all the boosters launched by NASA.

Display the total payload mass carried by boosters launched by NASA (CRS)

[20]:
```
%%sql

select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE
where Customer = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

[20]:

| sum(PAYLOAD_MASS__KG_) |
| --- |
| 45596 |

# EDA. SQL

## Average Mass F9 v.1.1 Boosters

A query to display the average payload mass of the boosters version F9 v.1.1.

Display average payload mass carried by booster version F9 v1.1

```
[22]: %%sql

select avg(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where Booster_Version = 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

[22]:

| avg(PAYLOAD_MASS__KG_) |
| --- |
| 2928.4 |

# EDA. SQL

## Date of the First Successful Landing on the Ground Pad

A query to display the first successful landing on the ground pad.

```
[24]: %%sql

select min(Date)
from SPACEXTABLE
where Landing_Outcome like 'Success%ground pad%';
```

```
 * sqlite:///my_data1.db
Done.
```

[24]:

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

# EDA. SQL

## Querying on Multiple Conditions

This query displays the list of the names of the boosters that have success in landing on a drone ship and have a payload mass greater than 4000 but less than 6000.

[26]:
```
%%sql

select Booster_Version
from SPACEXTABLE
where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
and Landing_Outcome like 'Success%drone ship%';
```

```
 * sqlite:///my_data1.db
Done.
```

[26]:

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

# EDA. SQL

## Successful and Failed Mission Outcomes

Listing the total number of successful and failed mission outcomes.

List the total number of successful and failure mission outcomes

[63]:
```
%%sql

select Mission_Outcome, count(*)
as total_number
from SPACEXTABLE
group by Mission_Outcome;
```

```
 * sqlite:///my_data1.db
Done.
```

[63]:

| Mission_Outcome | total_number |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# EDA. SQL

## Booster Versions that Carried the Maximum Payload Mass

Listing the all the booster versions that were carrying the maximum payload mass.

[30]:

| 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 |

# EDA. SQL

## Querying on Multiple Conditions with Introduction of Month Names

[32]:

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

Displaying the month names, failure landing outcomes on a drone ship, the booster versions, and the launch site for the months in the year 2015.

# EDA. SQL

## Ranking the Count Landing Outcomes by Number for a Particular Period

Displaying the ranked count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.

[34]:

| Landing_Outcome | counts_of_outcomes |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

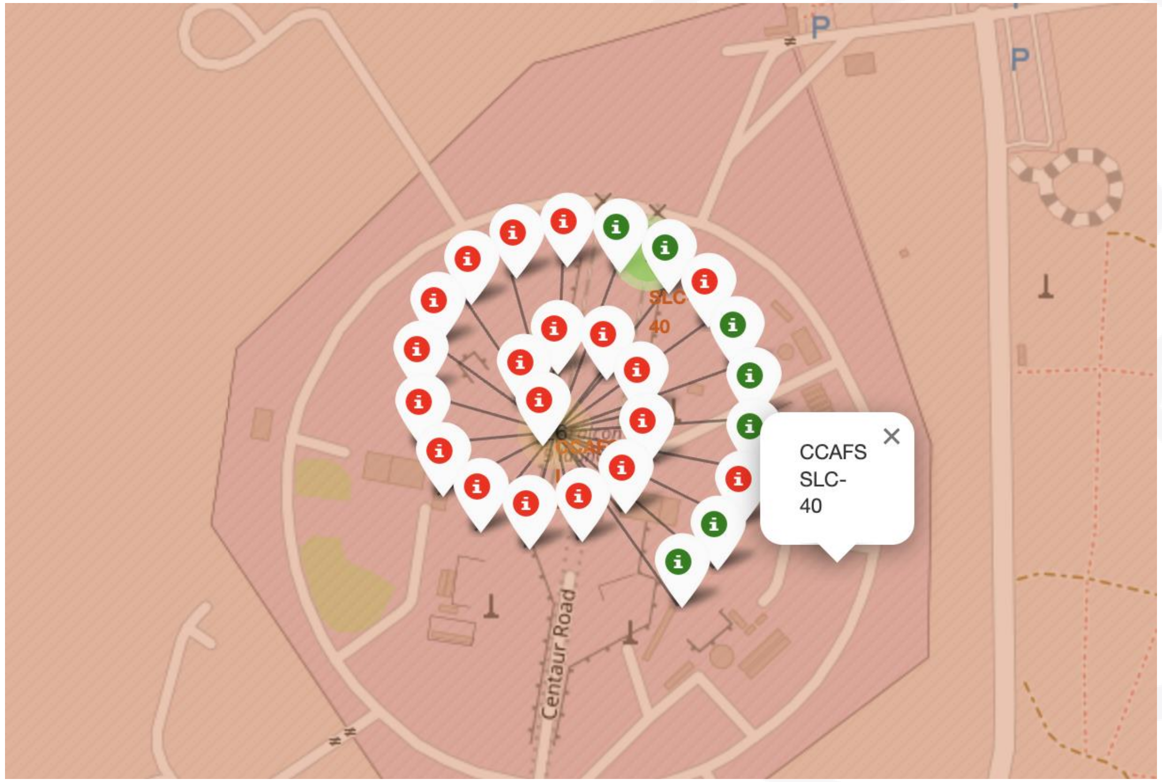
# Interactive Visual Analytics with Folium

## Locations of SpaceX Launch Sites in Florida and California


The coordinates of the launch sites return locations at the coast in the Southern states of Florida and California in a proximity of the coast line.

# Interactive Visual Analytics with Folium

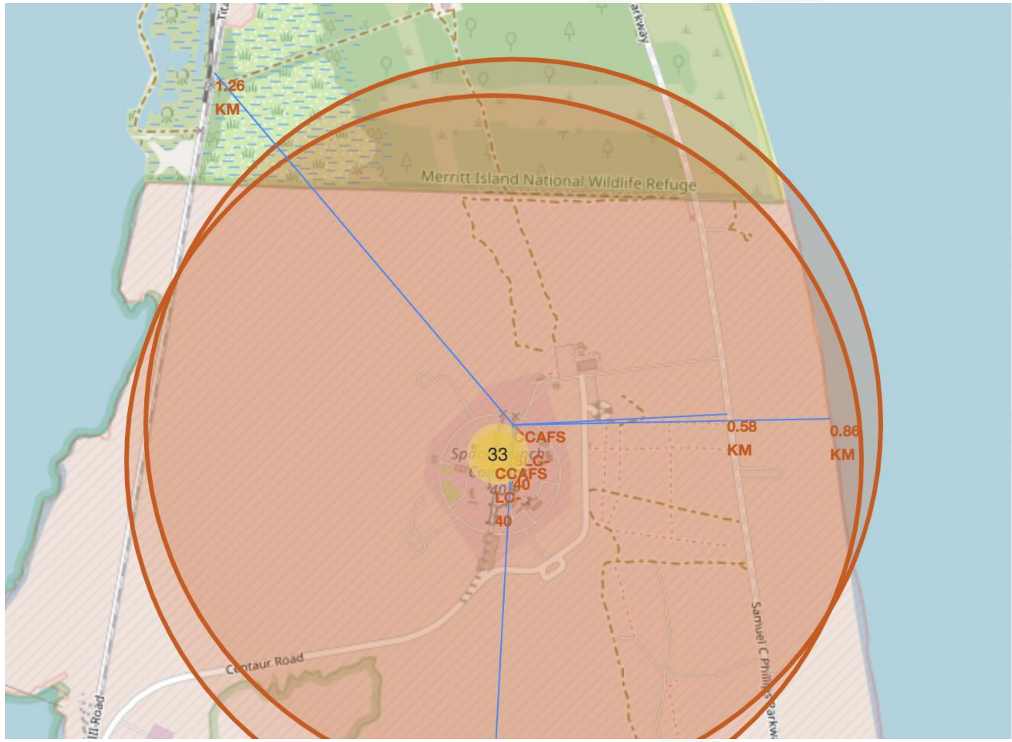
## Adding Markers and Marker Clusters




To enrich the map with additional information, markers and marker clusters were added to show the number of successful (green) and unsuccessful (red) launch outcomes for each launching site.

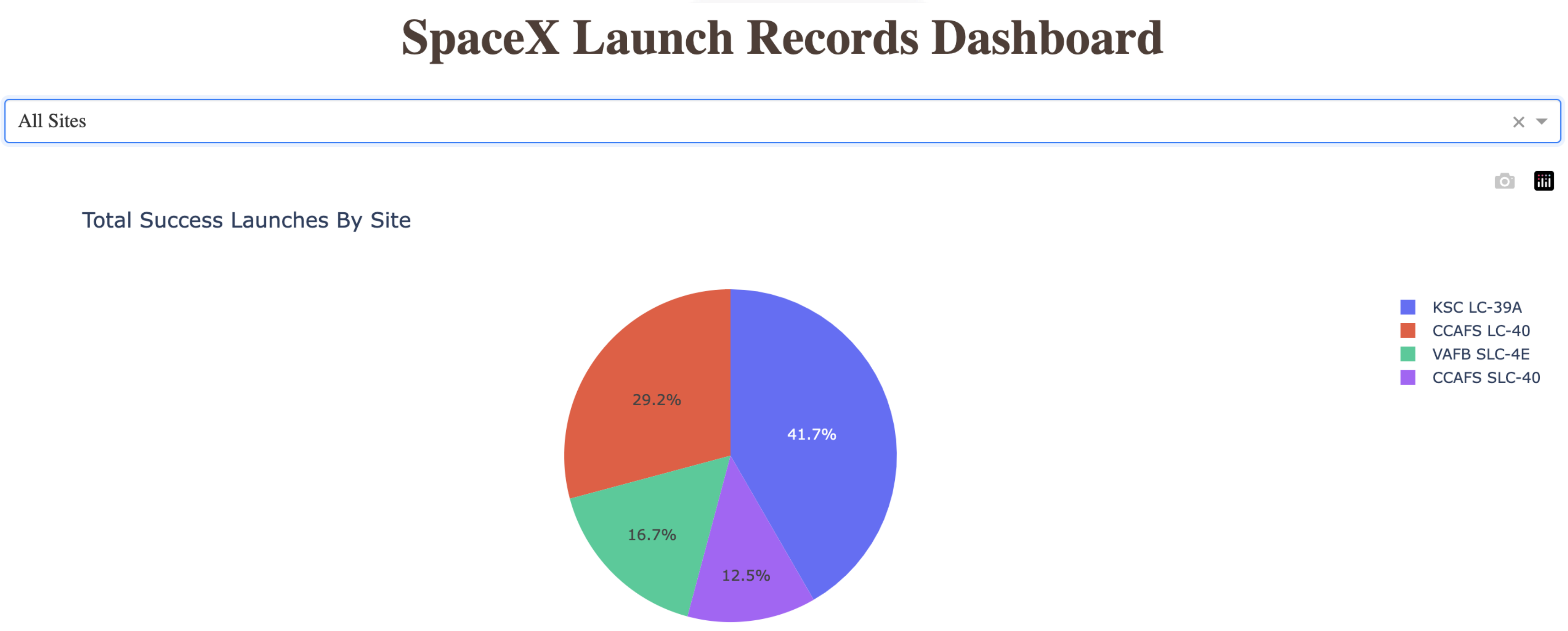
# Interactive Visual Analytics with Folium

## Adding Distances to the Coastline, Infrastructure and Cities



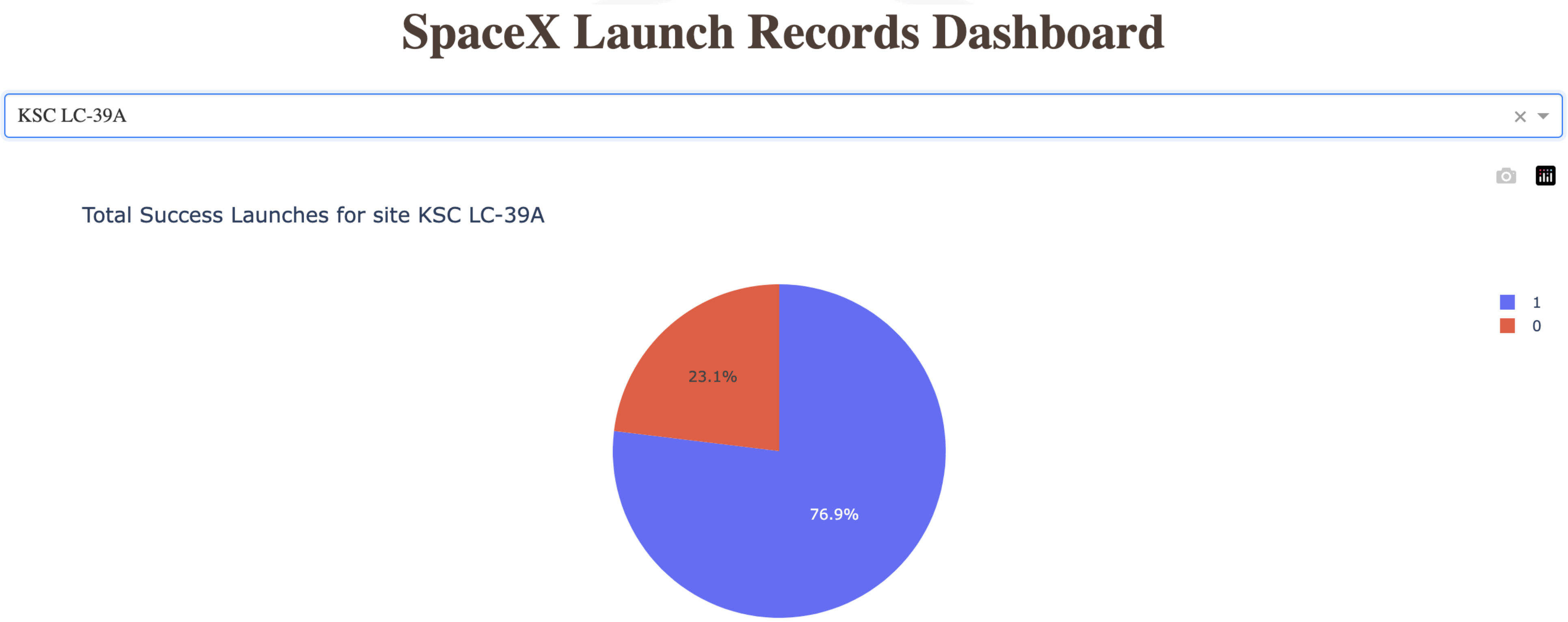

The distances to the nearest point at the coast, the nearest highway and railway, as well as to the nearest city were calculated and added to the map.

# Dashboards with Plotly Dash



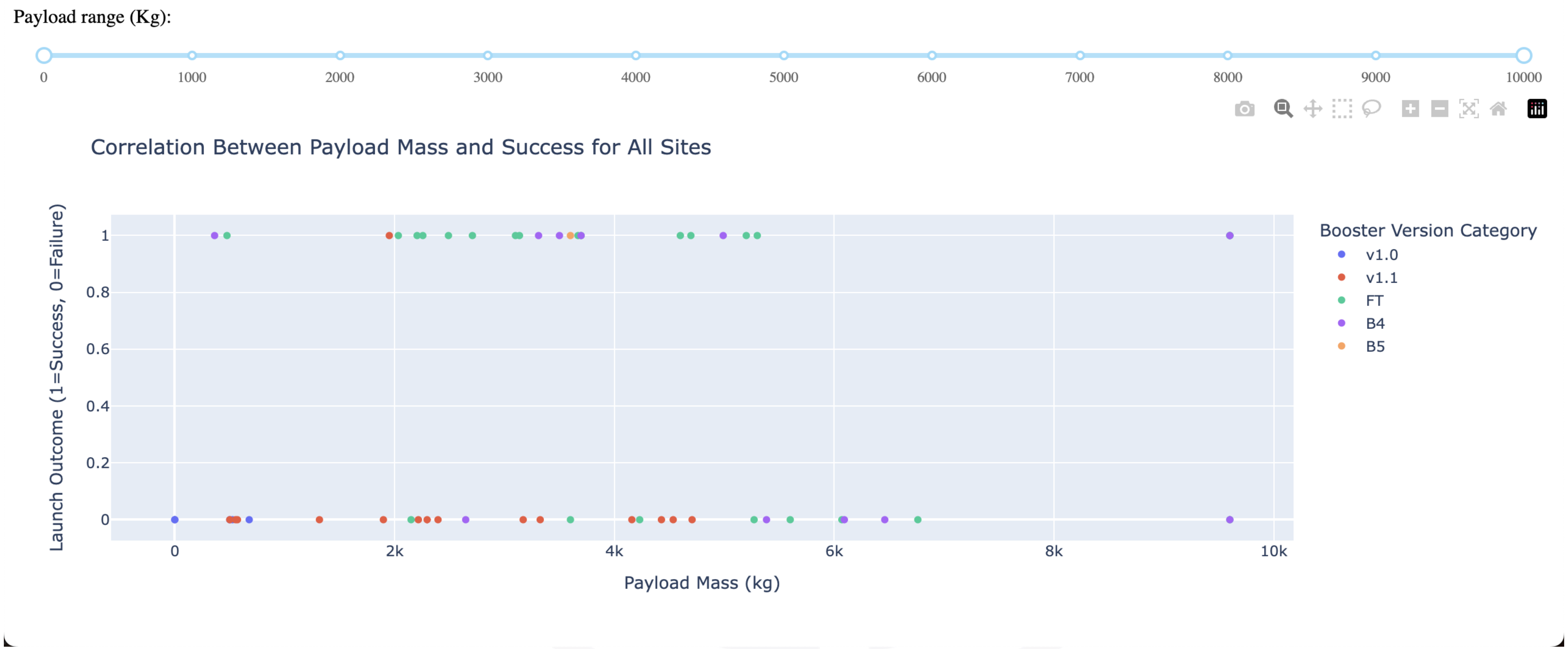

A pie chart for the total number of successful launches from all launch sites with KSC LC-39A shows a significantly higher success rate.

# Dashboards with Plotly Dash




Percentage of successful and unsuccessful launch outcomes for KSC LC-39A launching site.

# Dashboards with Plotly Dash




Here we can see the influence of the payload mass on the flight's outcome.

# Predictive Analysis

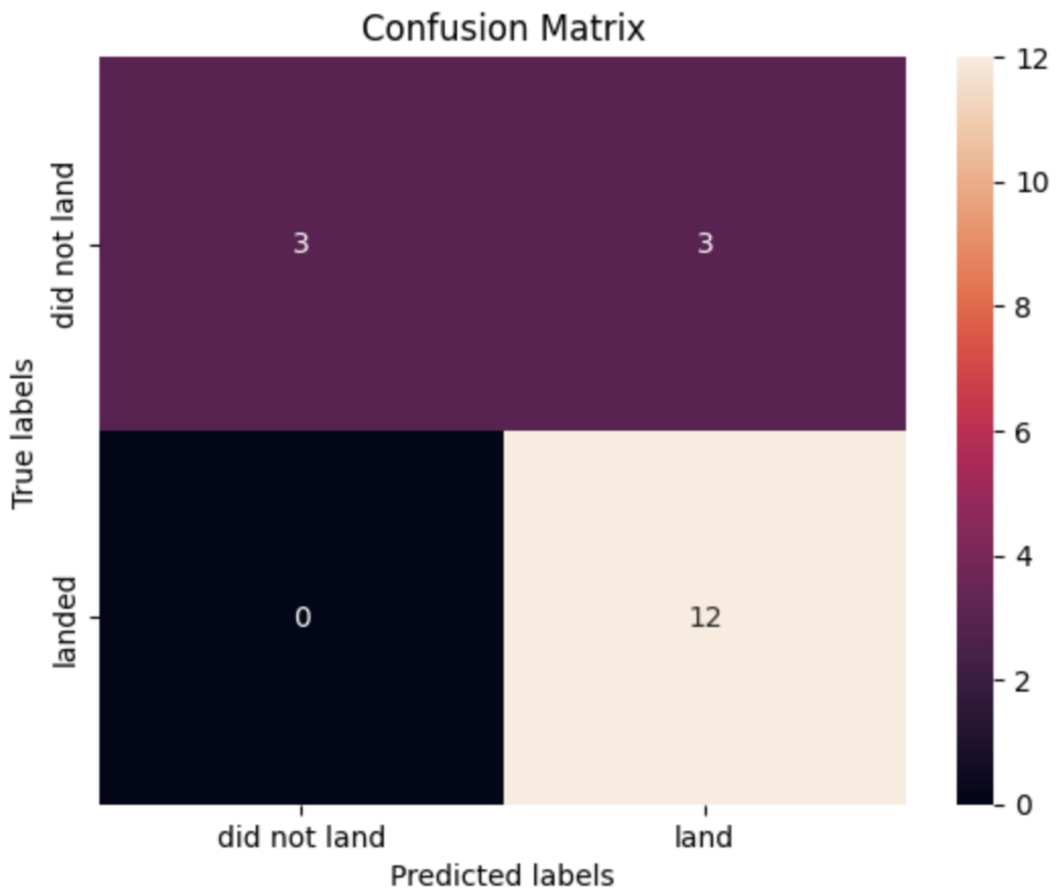
Because of the scarcity of the data, the received results are in most cases very similar. Additional testing or training on the larger dataset can increase the accuracy of the models.

```
[55]: data = {'model':['Logistic Regression', 'SVM', 'Decision Tree', 'KNN'],
              'Accuracy': [logreg_score, svm_score, tree_score, knn_score]}
      Report = pd.DataFrame(data)

      print(Report)

      best_model_idx = Report['Accuracy'].idxmax()  # Get the index of the highest accuracy
      best_model = Report.loc[best_model_idx, 'model']
      best_accuracy = Report.loc[best_model_idx, 'Accuracy']

      # Print the result
      print(f'The best performing method: {best_model}. Accuracy score: {best_accuracy:.4f}')
```

```
                 model  Accuracy
0  Logistic Regression  0.833333
1                  SVM  0.833333
2        Decision Tree  0.833333
3                  KNN  0.833333
The best performing method: Logistic Regression. Accuracy score: 0.8333
```

# Predictive Analysis

Confusion matrices of all models show that regardless of the model the classifier can distinguish between different classes. Also, the main issue is false positive predictions.



# Conclusion

From the described research we can conclude that:

There is an overall trend for **increasing success rates** for the flights from 2013 to 2020.

Regarding the particular launch sites **the higher the number of flights the higher the success rate.**

**KSC LC-39A** has the highest success rate of launches among the other sites.

The success rate of the **orbits ES-L1, GEO, HEO, and SSO is 100%**.