IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michael Shahbaz
17th December 2024

OUTLINE

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

EXECUTIVE SUMMARY

- **Summary of Methodologies**
    - Data Collection through API
    - Data Collection with Web Scraping
    - Data Wrangling
    - Exploratory Data Analysis with Data Visualization and Visual Analytics
    - Exploratory Data Analysis with SQL
    - Interactive Map with Folium

- **Summary of Results**
    - Plotly Dashboard Results
    - Predictive Analysis results
    - Exploratory Data Analysis results
    - Interactive analytics in screenshots

INTRODUCTION

- **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.

  The goal of the project is to create a machine learning pipeline to predict whether or not the first stage will land successfully.

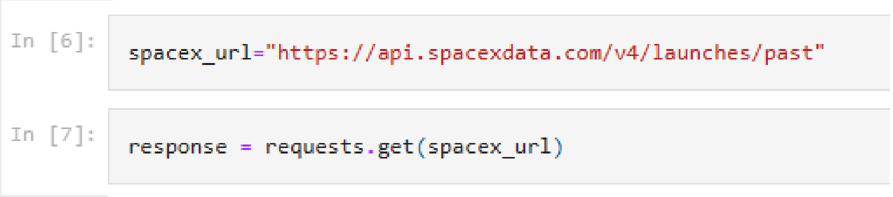
- **Finding Answers to the Following Issues**
  - Which factors will lead to a successful landing?
  - What are the operating conditions that need to be in place to ensure success?

SUMMARY OF METHODOLOGIES

- Data Collection via API
- Data Wrangling
- Exploratory Data Analysis(EDA) through:
  - Visualization
  - SQL
- Interactive Visual Analysis
  - Folium (Powerful Python library to create several types of maps)
  - Plotly (free, open-source, browser-based graphing library that provides tools for: Data visualization, Statistical analysis, Online graphing, and Scientific graphing)
- Predictive Analysis using Classification Models
  - Build, tune, and evaluate

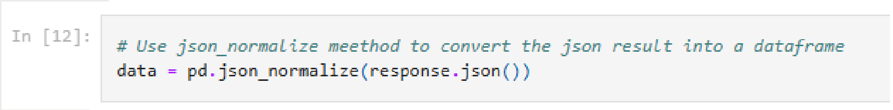
DATA COLLECTION

- Data was collected by sending a GET request to the SpaceX API
- When the response was received, the data collected was in JSON format
  - We used a built-in function - json_normalize() - to convert the JSON data and drop it into a Pandas dataframe
- After the data was put into a dataframe, the data was cleaned by checking for missing values and fill in said missing values, when necessary
- Additionally, to obtain Falcon 9 launch records, a web scraping tool - called BeautifulSoup - was used to scrape their Wikipedia page
  - The objective was to extract the launch records as an HTML table, parse the table, and then convert it to a Pandas dataframe for future analysis

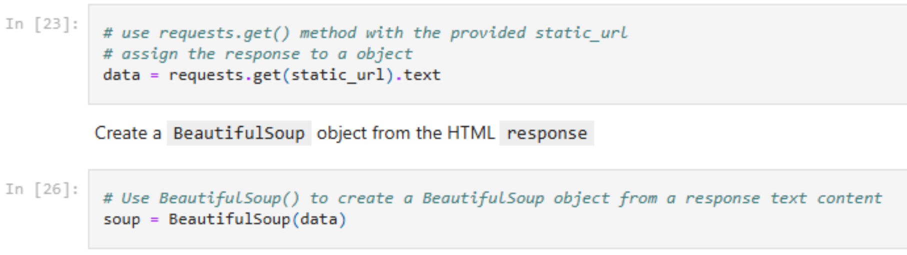


In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```



In [12]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```



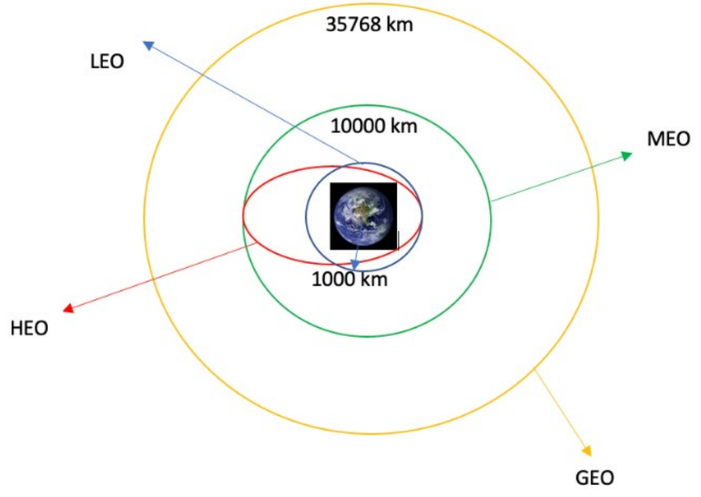
In [23]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

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

In [26]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data)
```

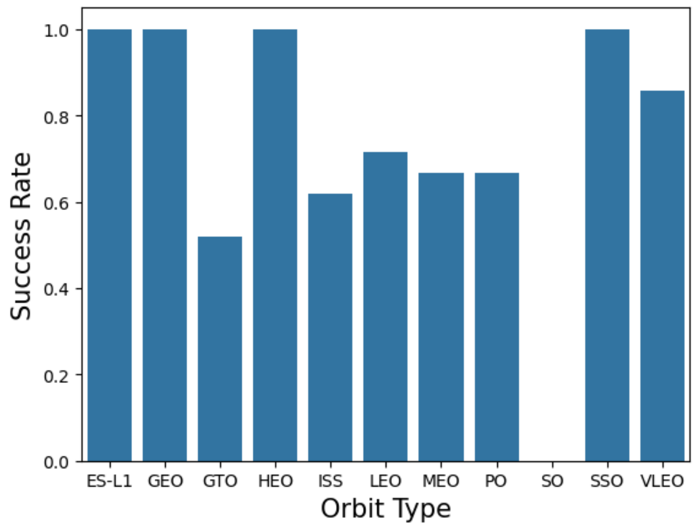
DATA WRANGLING

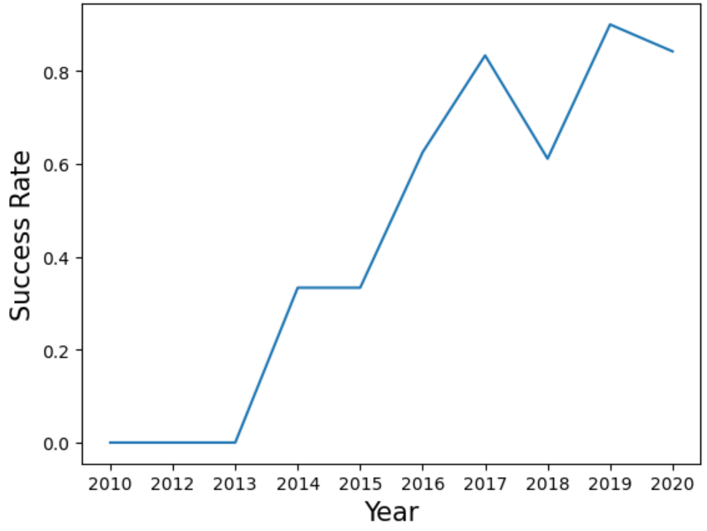
- Objectives
  - Perform exploratory Data Analysis to find some patterns in the data
  - Determine what would be the label for training supervised models
- Calculated number of launches at each site and occurrence of each orbit
- Created landing outcome labels from outcome column and exported the results to csv



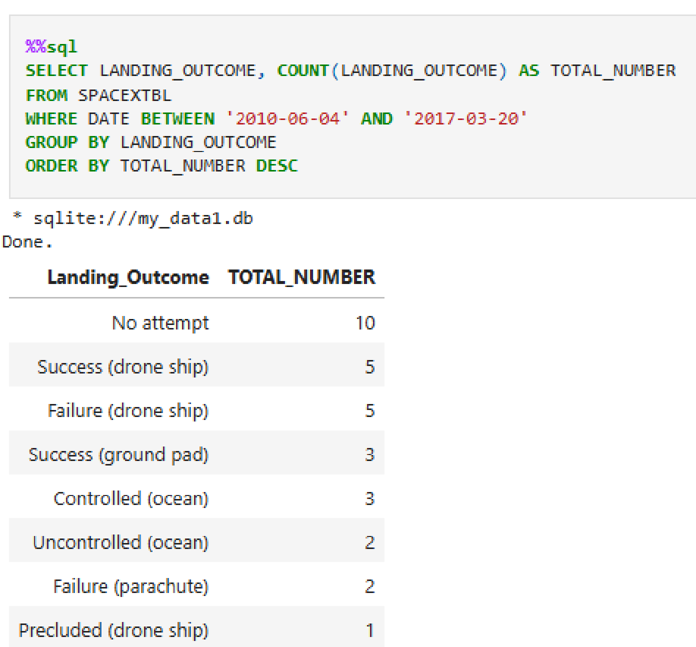
EXPLORATORY DATA ANALYSIS (EDA) with DATA VISUALIZATION

- Explored the data by visualizing the relationship between:
  - Flight Number and Launch Site
  - Payload Mass and Launch Site
  - Success rate of each Orbit type
  - Flight Number and Orbit type
  - Payload Mass and Orbit type
  - Launch success yearly trend





EXPLORATORY DATA ANALYSIS (EDA) with SQL



```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

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

Landing_Outcome	TOTAL_NUMBER
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

- Loaded the SpaceX dataset into a PostgreSQL database
- Applied EDA with SQL to get insight from the data by writing queries to find out:
  - The names of unique launch sites in the space mission
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names

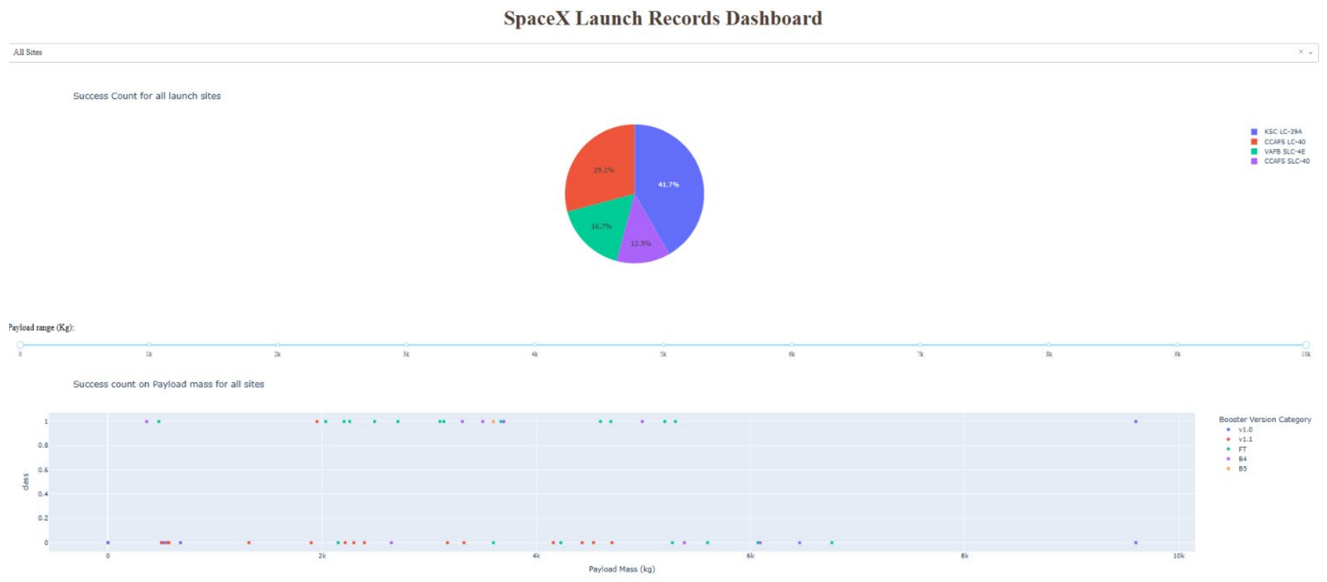
INTERACTIVE MAP with FOLIUM

- Objectives
  - Utilize Folium to generate visual maps to find any geographical patterns about launch sites
- Marked all launch sites, added map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the Folium map
- Assigned the feature launch outcomes (failure or success) to class 0 and 1 where 0 is for failure, and 1 is for success
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rates
- Calculated the distances between a launch site to its proximities
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?

CCAFS SLC-40
P
26
CAFS
LC-
40

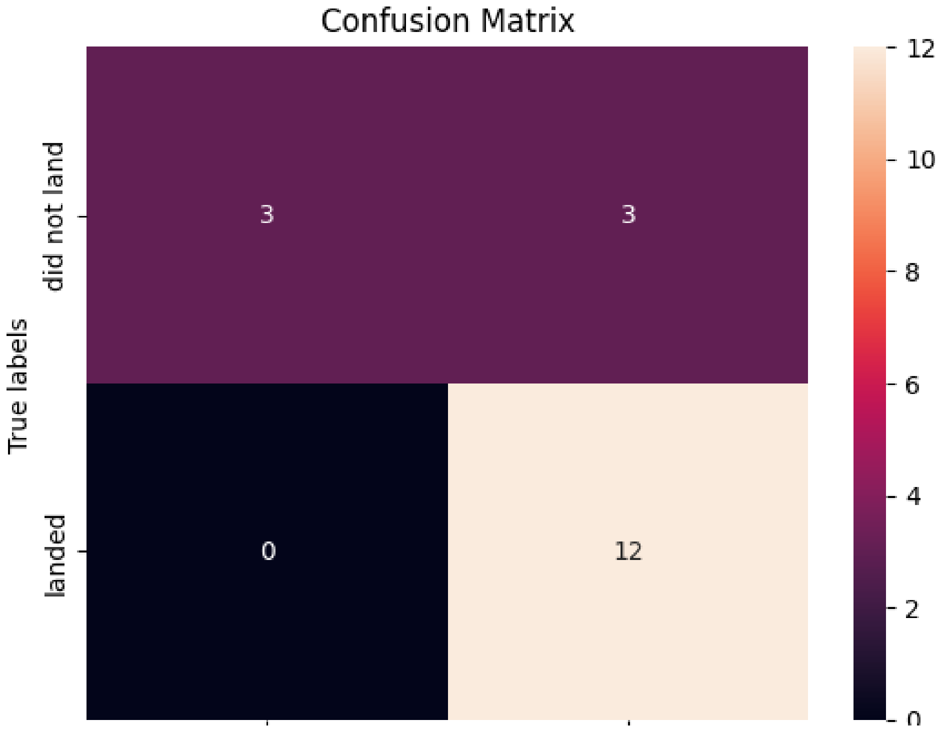
PLOTLY DASHBOARD

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version



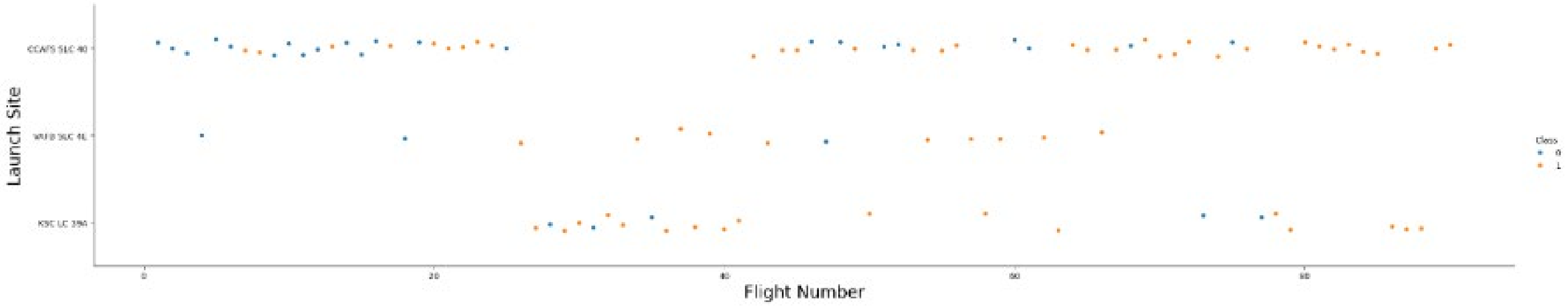
PREDICTIVE ANALYSIS (CLASSIFICATION)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing
- Built different machine learning models and tuned different hyperparameters using GridSearchCV
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning
- Using accuracy, we also found the best performing classification model out of K Nearest Neighbors, Decision Tree, and Logistical Regression



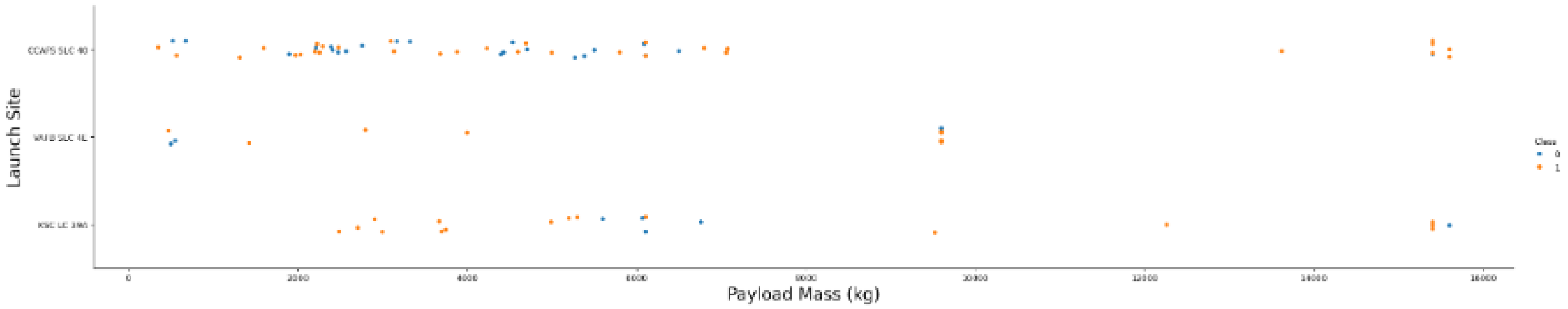
FINDINGS & IMPLICATIONS

- In the next set of slides, we’ll go over the some of the key insights and findings from the SpaceX data we reviewed:
  - EDA with Data Visualization Analysis
  - EDA with SQL
  - Folium Map results
  - Plotly dashboard results
  - Classification method results



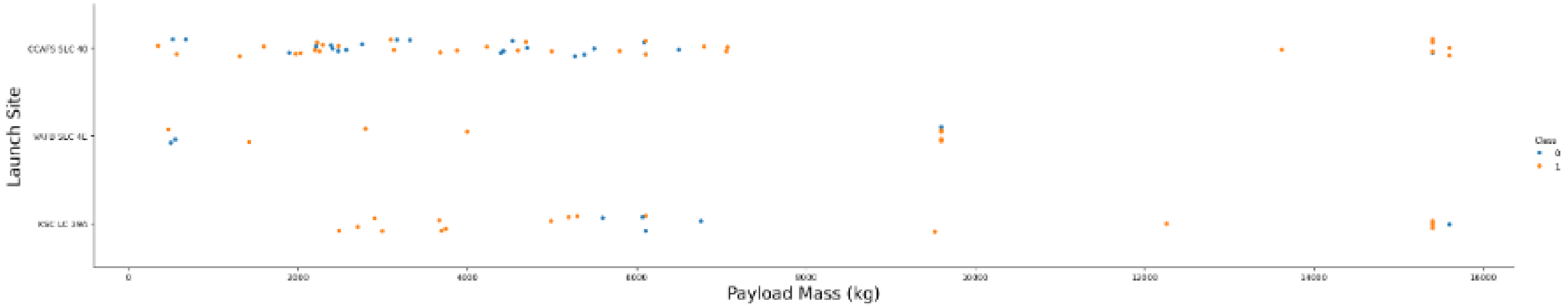
EDA with Data Visualization Analysis

- **Flight Number vs. Launch Site**
  - From the plot, we found that more frequent flights at a launch site, the greater the success rate



EDA with Data Visualization Analysis

- **Payload Mass vs. Launch Site**
  - The greater the payload mass at site CCAFS SLC 40, the higher the success rate

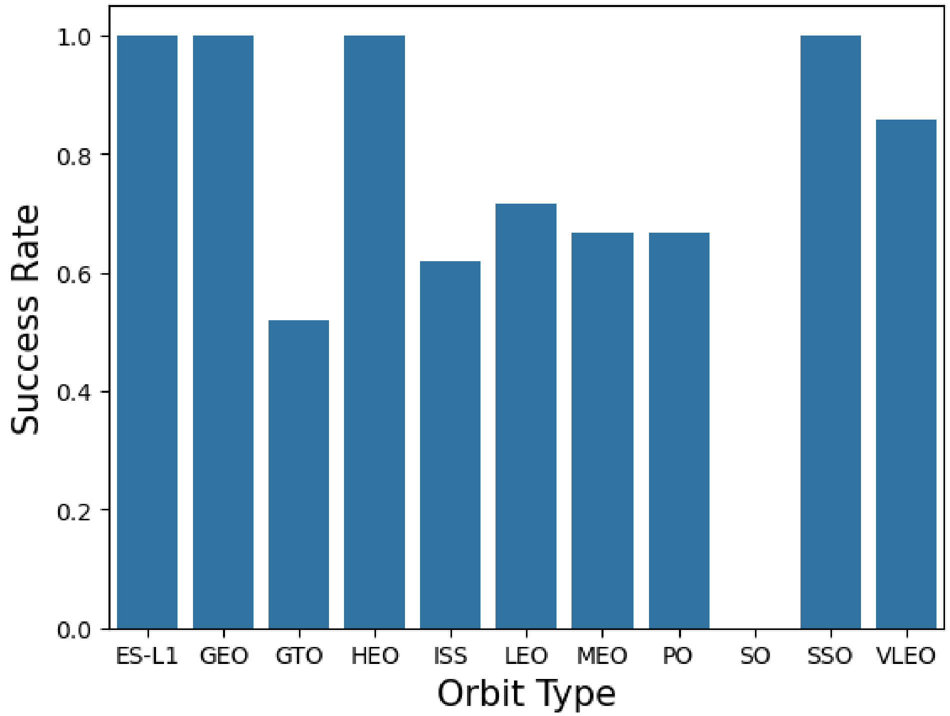


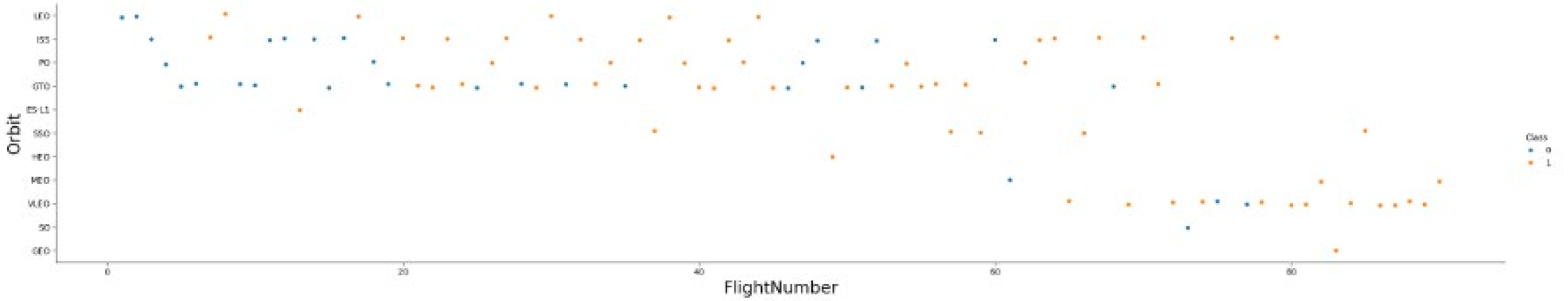
EDA with Data Visualization Analysis

- **Payload Mass vs. Launch Site**
  - The greater the payload mass at site CCAFS SLC 40, the higher the success rate

EDA with Data Visualization Analysis

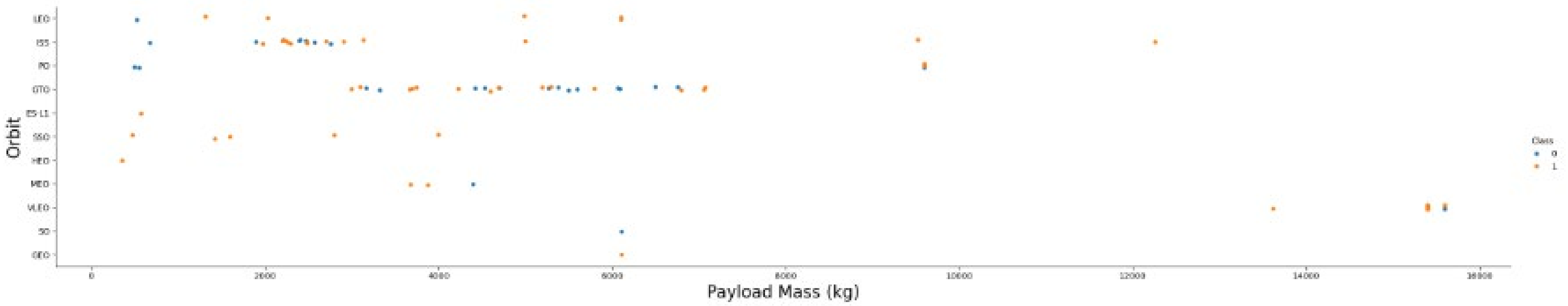
- **Success Rate vs. Orbit Type**
  - The bar graph shows that orbit types: ES-L1, GEO, HEO, SSO, VLEO had the highest success rates





EDA with Data Visualization Analysis

- **Flight Number vs. Orbit Type**
  - From the plot above, it appears that in the LEO orbit, success is related to the number of flights
  - Conversely, in the GTO orbit, there appears to be no relationship between flight number and orbit type

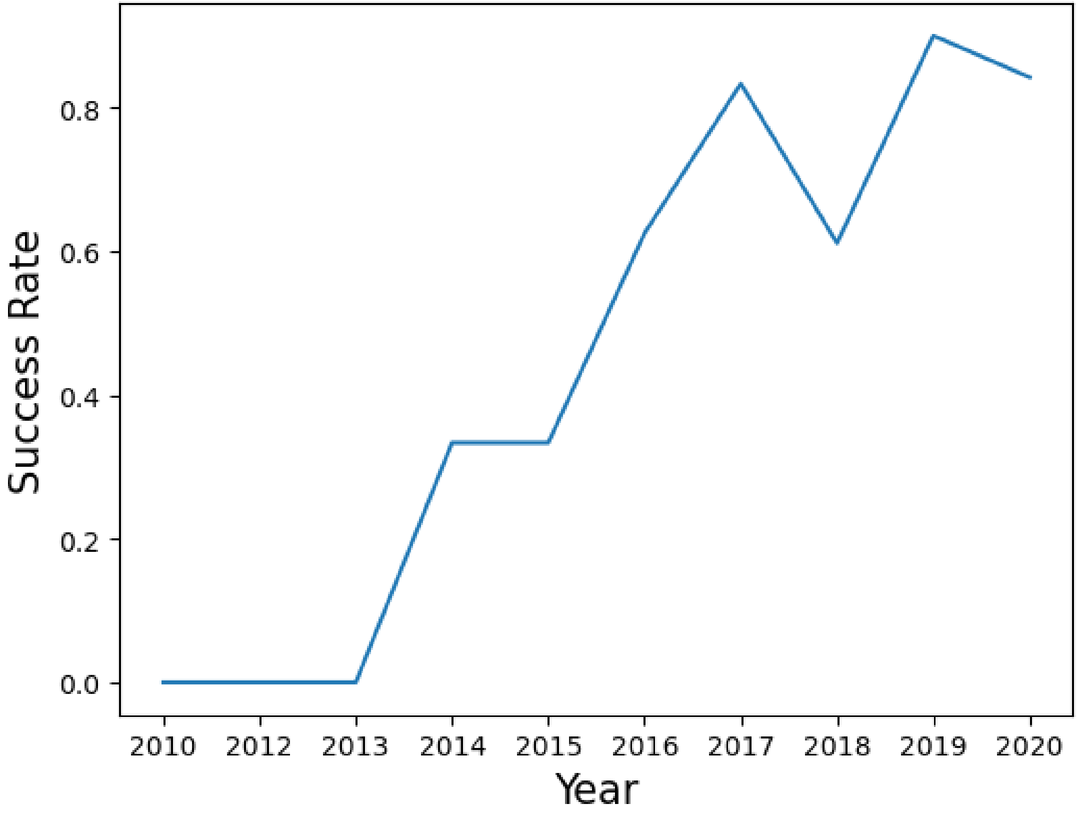


EDA with Data Visualization Analysis

- **Payload Mass vs. Orbit Type**
  - From the plot above, it appears that with heave payloads the successful landing rates are more for Polar, LEO, and ISS
  - Conversely, in the GTO orbit, it’s difficult to distinguish between successful and unsuccessful landings – as both outcomes are present

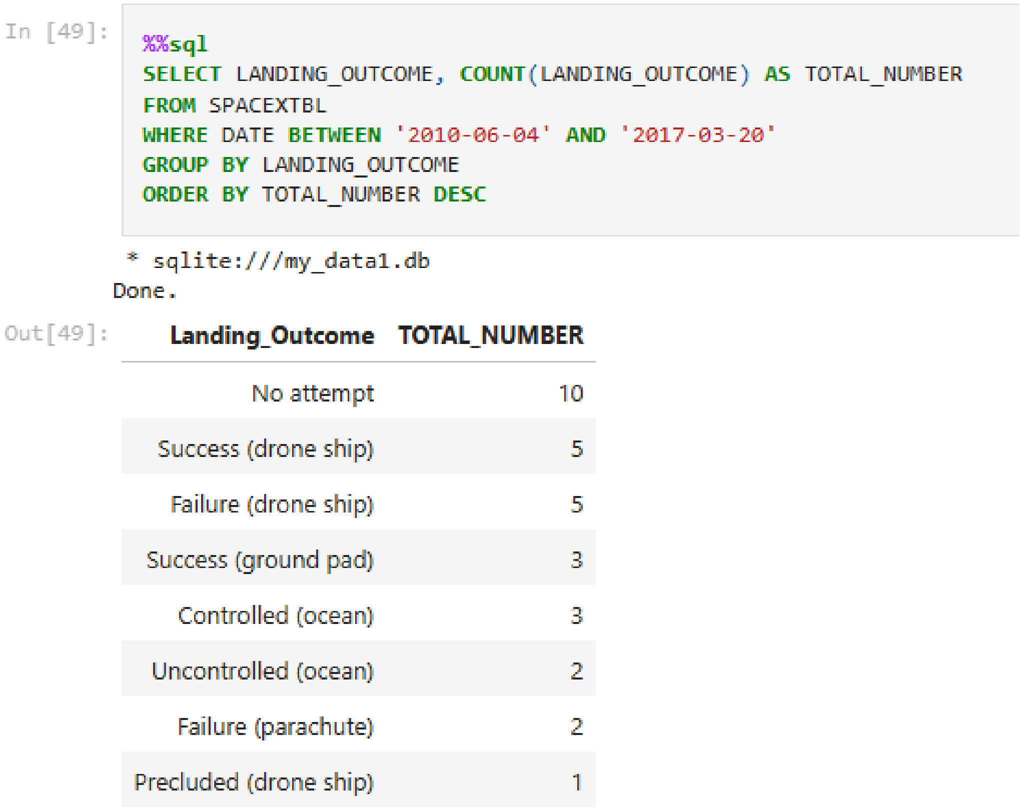
EDA with Data Visualization Analysis

- **Yearly Success Trend**
  - The line graph shows that success rate has increased from 2013 until 2020



EDA with SQL Analysis

- Using an SQL query on the SpaceX data table, we ranked 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
  - We can see that the number of successful landings and unsuccessful landings were the same for drone ships for the 7 year period at 5 each
  - There 10 landings that weren’t even attempted in that same time span



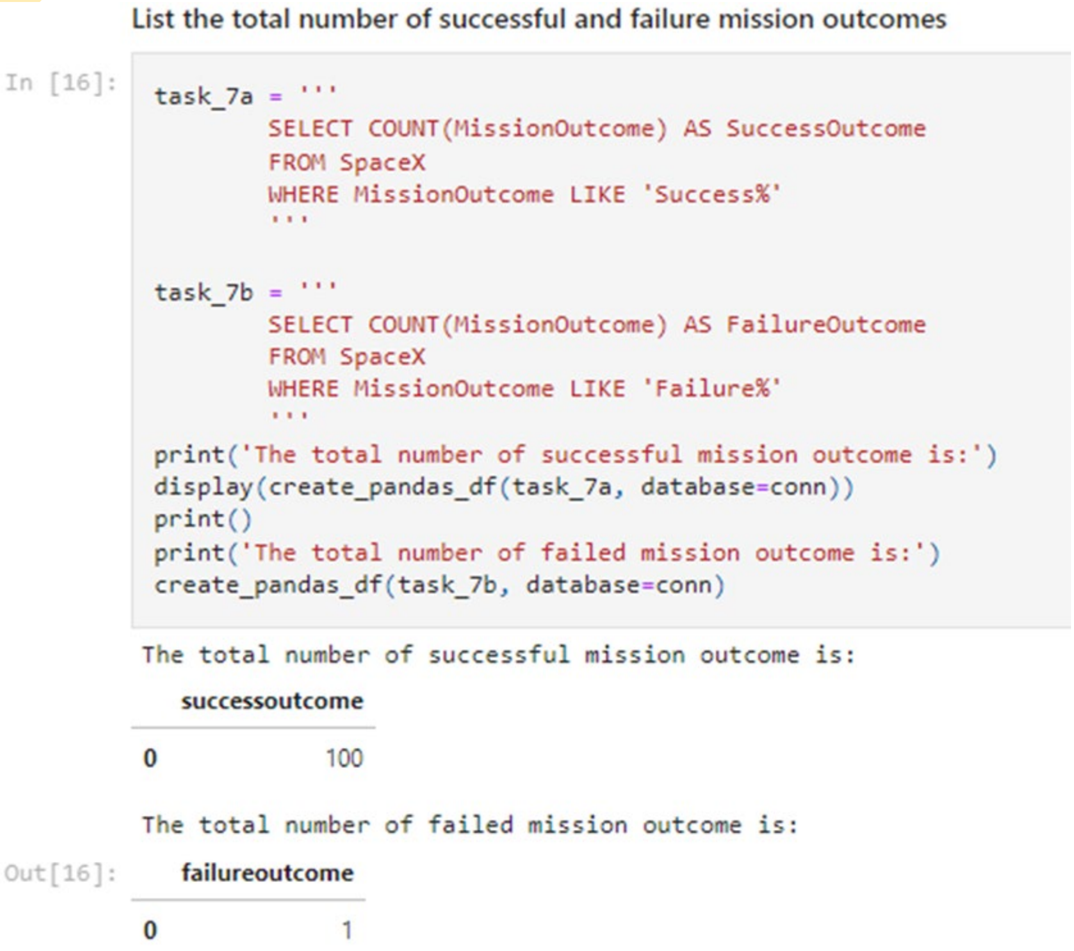
```
In [49]: %%sql
         SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
         FROM SPACEXTBL
         WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
         GROUP BY LANDING_OUTCOME
         ORDER BY TOTAL_NUMBER DESC

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

Out[49]:

Landing_Outcome	TOTAL_NUMBER
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

EDA with SQL Analysis



List the total number of successful and failure mission outcomes

```
In [16]:
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

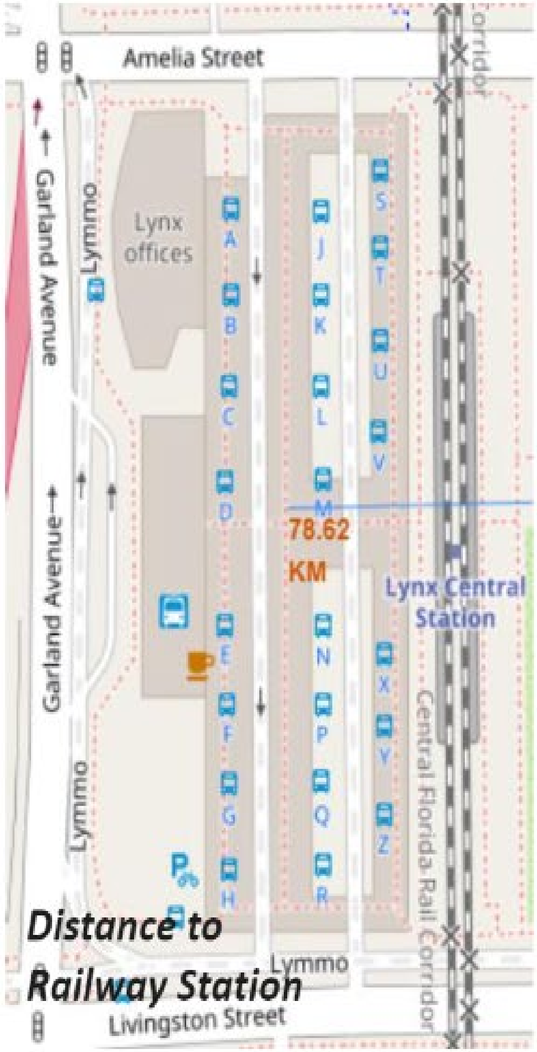
```
The total number of failed mission outcome is:
```

Out[16]:

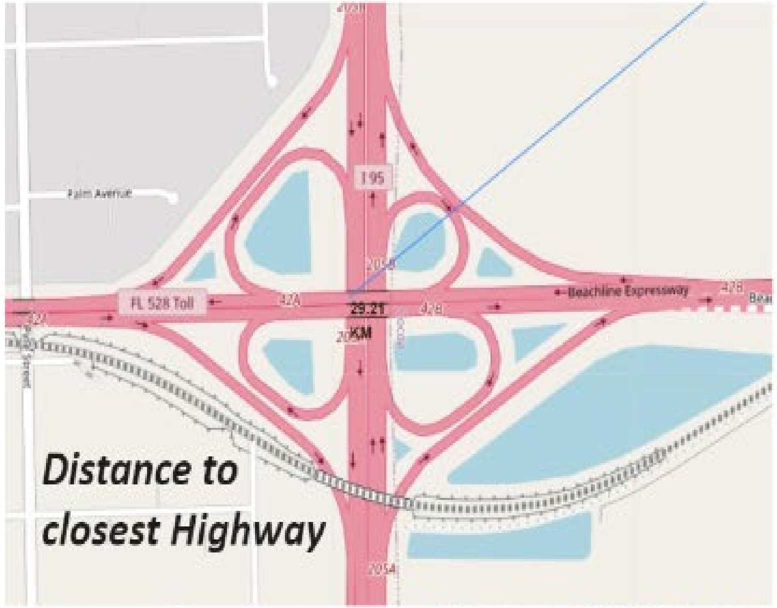
	failureoutcome
0	1

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.
  - We can see that there was only 1 failed mission in the entire data table, so that tells us the rate of successful missions is very high

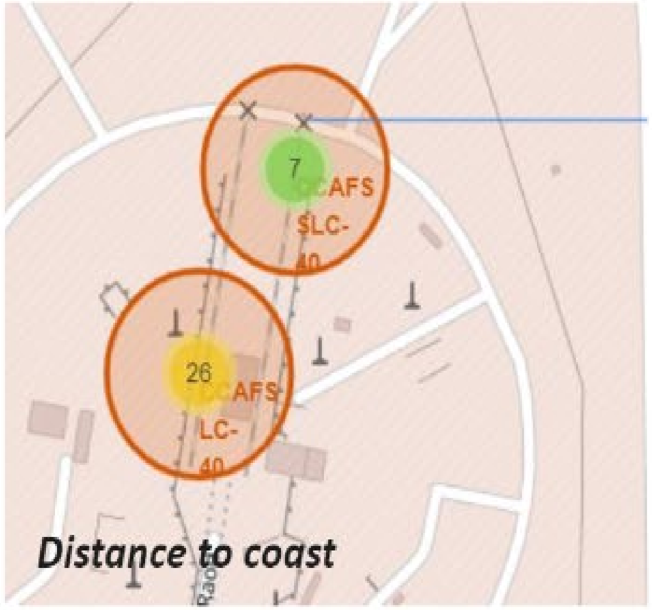
FOLIUM MAP RESULTS



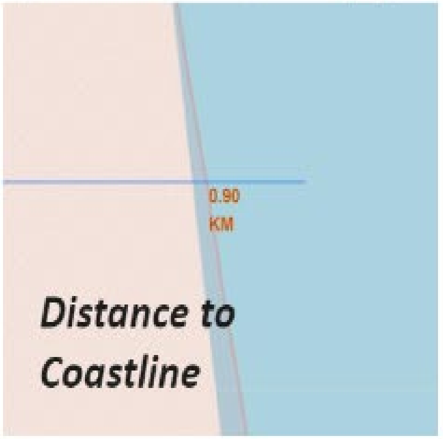
Distance to Railway Station



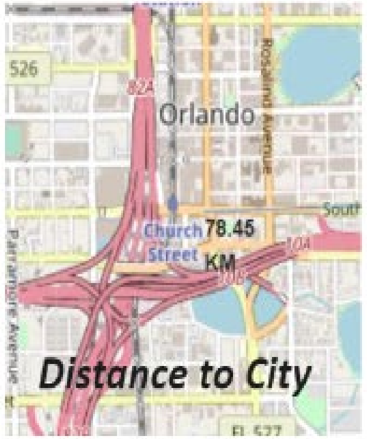
Distance to closest Highway



Distance to coast



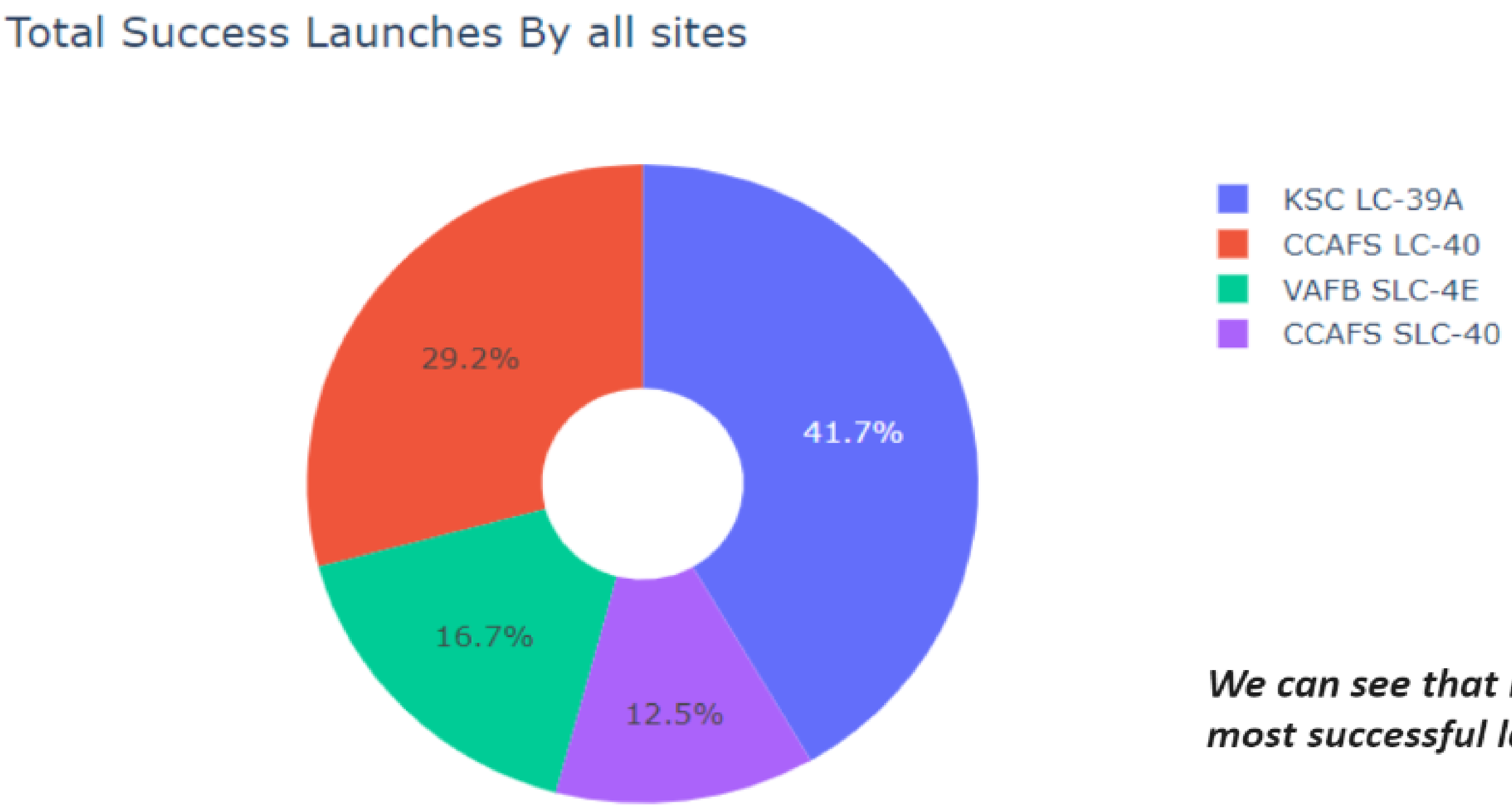
Distance to Coastline



Distance to City

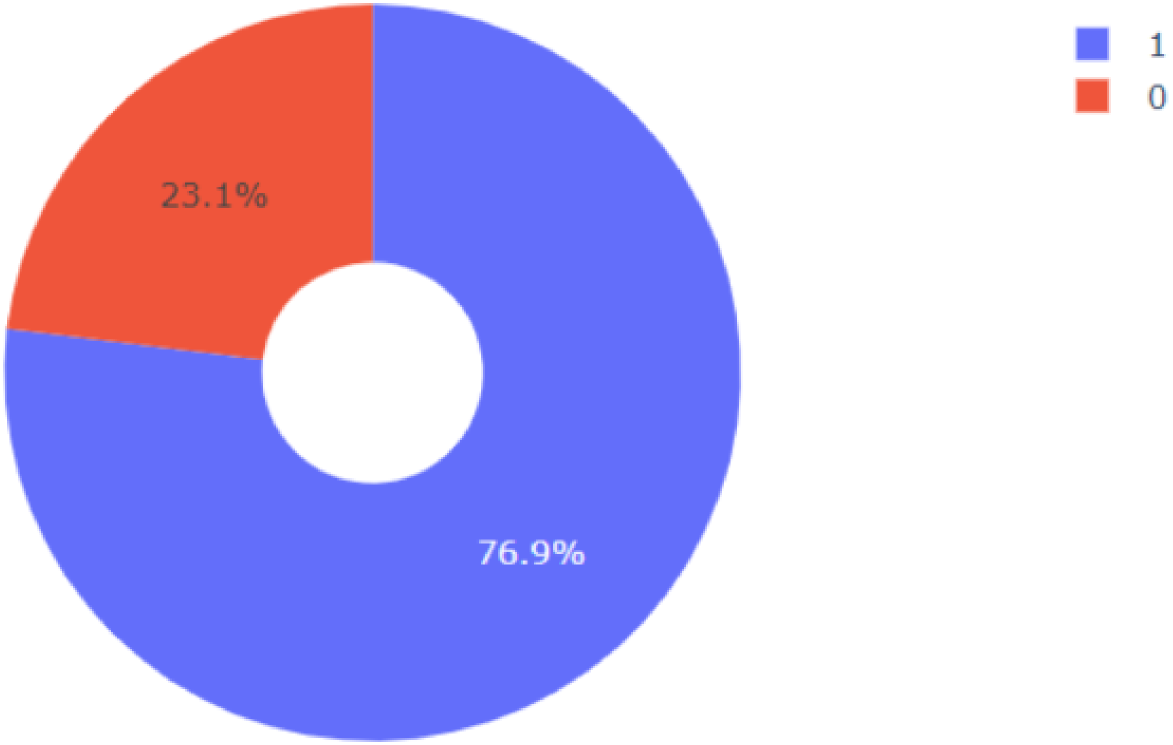
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

PLOTLY DASHBOARD RESULTS



We can see that KSC LC-39A had the most successful launches from all the sites

PLOTLY DASHBOARD RESULTS

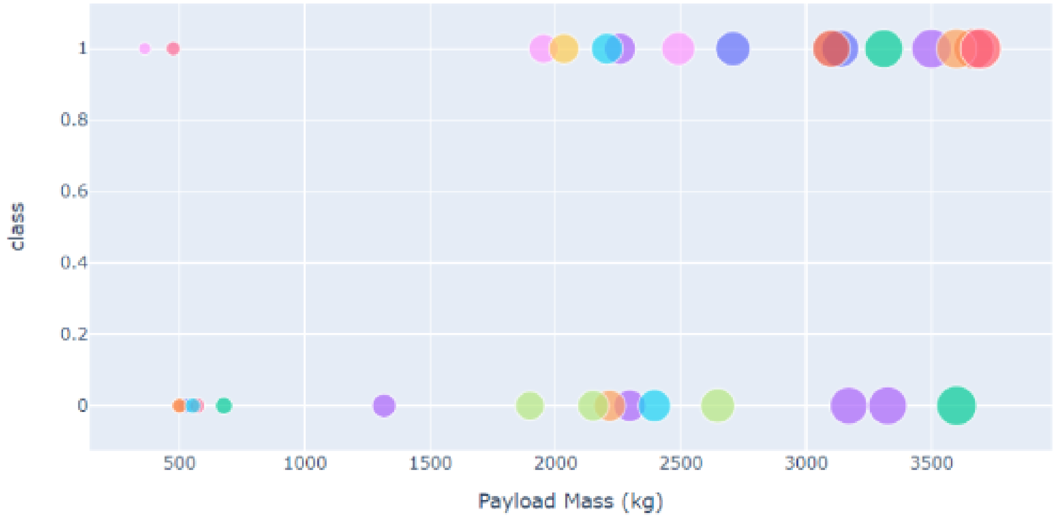


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

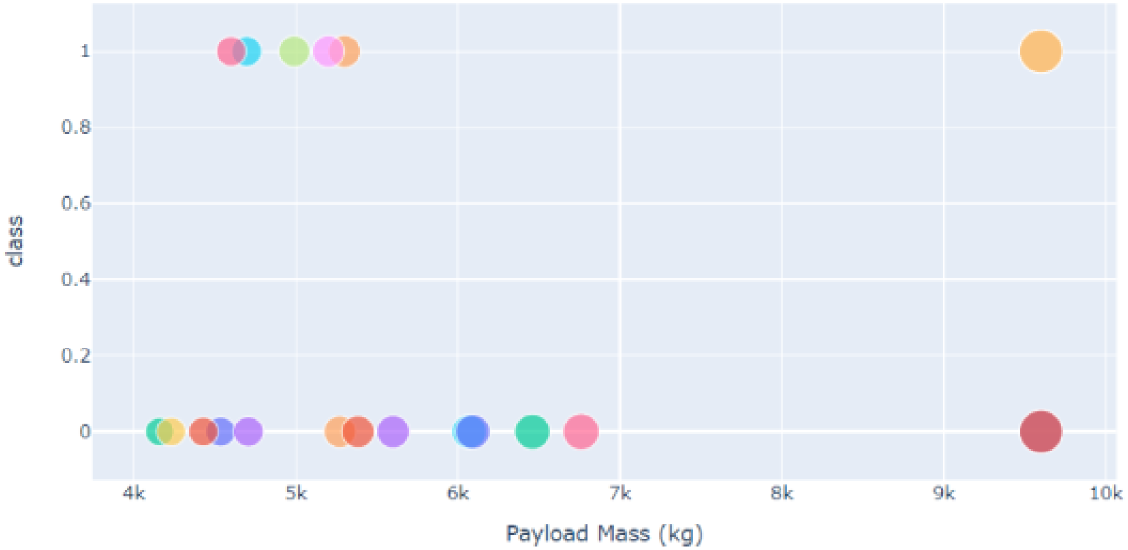
PLOTLY DASHBOARD RESULTS

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

Low Weighted Payload 0kg – 4000kg



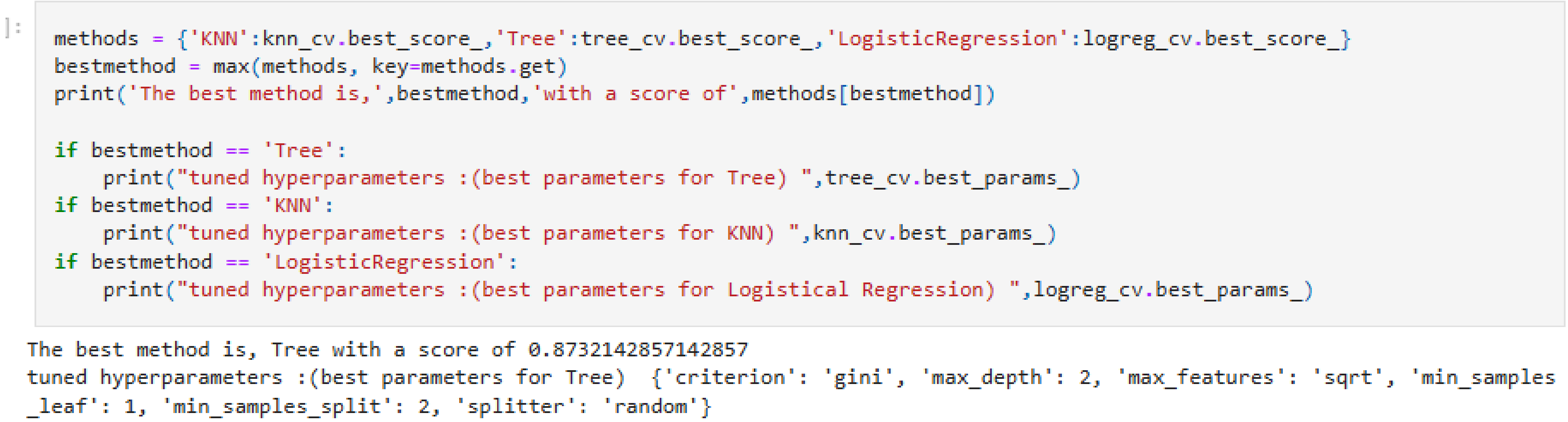
Heavy Weighted Payload 4000kg – 10000kg



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Classification Accuracy

- We can see below that the decision tree classifier is the model with the highest classification accuracy



```
methods = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestmethod = max(methods, key=methods.get)
print('The best method is,',bestmethod,'with a score of',methods[bestmethod])

if bestmethod == 'Tree':
    print("tuned hyperparameters :(best parameters for Tree) ",tree_cv.best_params_)
if bestmethod == 'KNN':
    print("tuned hyperparameters :(best parameters for KNN) ",knn_cv.best_params_)
if bestmethod == 'LogisticRegression':
    print("tuned hyperparameters :(best parameters for Logistical Regression) ",logreg_cv.best_params_)
```

```
The best method is, Tree with a score of 0.8732142857142857
tuned hyperparameters :(best parameters for Tree)  {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples
_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
```

CONCLUSION

- In conclusion, based on the overarching objective to determine if the first stage of the SpaceX launch will land, we can define the cost of a launch.
- The goal of the project was to create a machine learning pipeline to predict whether or not the first stage will land successfully – which we did and we found the following insights based on our machine learning and data analysis methodologies:
  - The more frequent flights at a launch site, the greater the success rate
  - Launch success rate started to increase from 2013 till 2020
  - Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate
  - KSC LC-39A had the most successful launches of any sites.
  - The Decision tree classifier is the best machine learning algorithm for this task

Thank you!