IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rafael Muniz
January 1st 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with REST API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL and Data Visualization
  - Interactive visual analytics using Folium
  - Machine Learning models
- Summary of all results
  - This presentation shows the insights from the Exploratory Data Analysis, showing that orbits ES-L1, GEO, HEO, and SSO have the best success rates and that the success rate by year in an upward trend. From the interactive visual analytics, it shows that KSC LC-39A launch site has the most successful number of launches, and Booster version FT has the most successful launch outcomes. At the Machine Learning section, it shows that after fine tuning the models, they all have the same accuracy when predicting the desired outcome, 83.3%.

Introduction

- Project background and context
  - Commercial space age emergence
  - Different players: Virgin Galactic, Blue Origin, Space X – working towards affordable space travel
  - Creation of Space Y company

- Problems you want to find answers
  - Predicting if SpaceX will reuse the 1st stage of the Falcon 9 rocket, so Space Y can use that information to know the real cost of the rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and Web Scraping techniques.
- Perform data wrangling
  - Identification of missing values, numerical calculations, and one-hot encoding.
- 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 Collection

1) Data collected from a public data source with two methods:
    1) SpaceX REST API
        1) Link: https://api.spacexdata.com/v4/launches/past - the data in this API gives information about the rocket, payload, launchpad, landing specifications, and landing outcomes.
        2) Decode the API response in JSON format
        3) Transform it into a Pandas dataframe
    2) Web Scraping from Wikipedia
        1) Link: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
        2) Extracting all the information of the table in Wikipedia using BeautifulSoup
        3) Transform it into a Pandas dataframe

Data Collection – SpaceX API

- SpaceX REST API calls:

GitHub URL of the SpaceX API calls notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w1.1%20-%20jupyter-labs-spacex-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [12]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [13]: response = requests.get(spacex_url)
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [19]: # Use json_normalize meethod to convert the json result into a dataframe
static_response = requests.get(static_json_url)
static_json_data = static_response.json()

spacex_data = response.json()
data = pd.json_normalize(spacex_data)
```

we can apply the rest of the functions here:

```
In [26]: # Call getLaunchSite
getLaunchSite(data)
```

```
In [27]: # Call getPayloadData
getPayloadData(data)
```

```
In [28]: # Call getCoreData
getCoreData(data)
```

Task 2: Filter the dataframe to only include `Falcon 9` launches

```
In [32]: # Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion'] == 'Falcon 9']
```

Data Collection - Scraping

- Web scraping from Wikipedia process:

GitHub URL of the web scraping notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w1.2%20-%20jupyter-labs-webscraping.ipynb

1) Get the Wikipedia link

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1
```

2) Request the SpaceX Launch Wiki page

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```

3) Using BeautifulSoup get and parse the data

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

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

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

In [10]:
```
# Use the find_all function in the Bea
# Assign the result to a list called `

# Find all tables on the page
html_tables = soup.find_all('table')
```

Data Wrangling

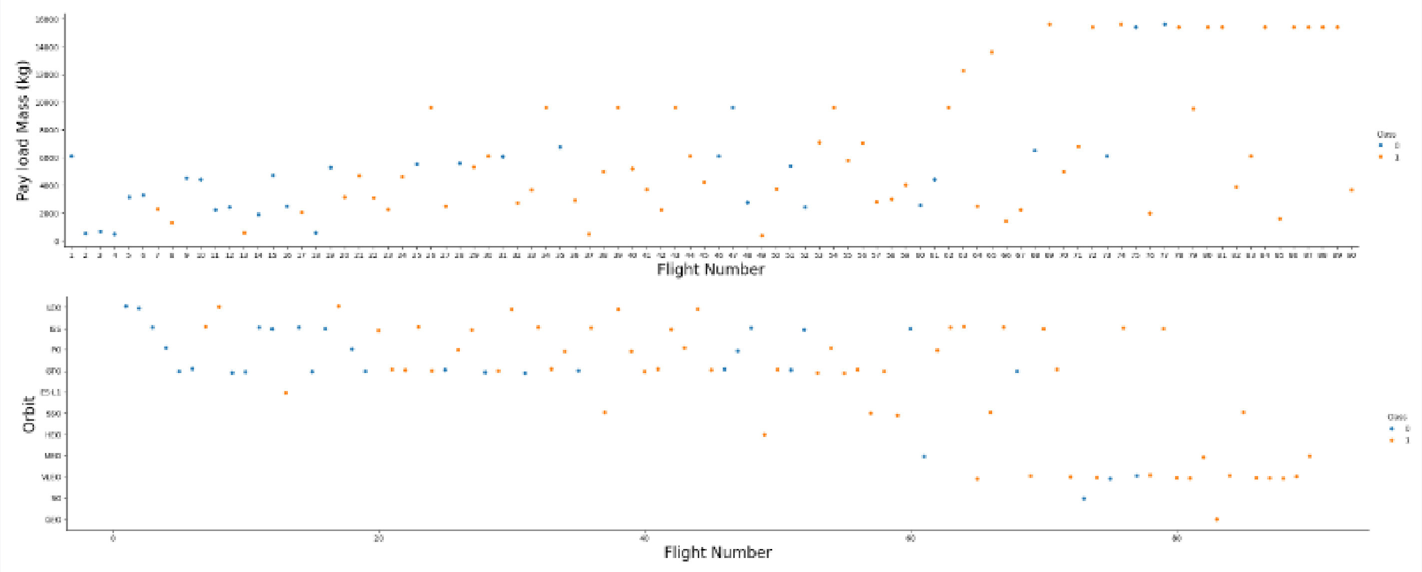
Steps of Data Wrangling process:

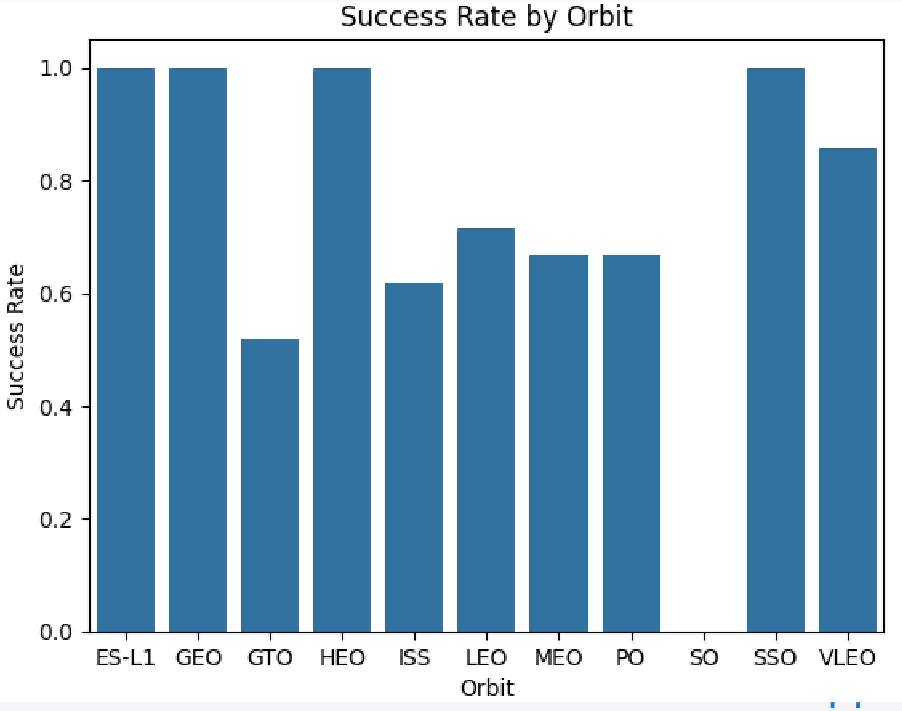
1) Identify the missing values
2) Calculate the number of launches on each site
3) Calculate the number and occurrence of each orbit
4) Calculate the number and occurence of mission outcome of the orbits
5) Create a landing outcome label from Outcome column

GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w1.3%20-%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Several charts were plotted: Flight number vs. PayloadMass, Flight number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload Mass vs. Orbit, Year vs. Success Rate. A sample of them are shown below:





GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w2.2%20-%20jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- SQL queries performed:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - 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.

GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w2.1%20-%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- The following was marked in the Folium map:
  - All launch sites – to see where all he launch sites were in the US territory.
  - The success/failed launched for each site – assigning with green (success) and red (failure) the launches in each launch site, to give a better understanding of how the launches performed in each site.
  - Distances between one of the launch sites to its proximities (railway, highway, coastline, city), with the use of a line – to know the distance to some key points of interest, avoiding risks and leveraging the location.

GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w3.1%20-%20lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Using Plotly Dash a web based dashboard was built, containing the following features:
  - A dropdown menu in which the used could choose the launch sites.
  - A pie chart that showed the success launches by site or the success/failure percentage according the launch site chosen.
  - A slider filter in which the user could choose the payload range in Kg.
  - A scatter plot of the Payload Mass vs. Outcome (success/failure) for different booster versions.

GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w3.2%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

- The predictive analysis followed the process shown below:
  - The data was loaded, transformed, and split into training and testing data, using Pandas, Numpy and Scikit Learn.
  - Different machine learning models were built: Logistic Regression, Support Vector Machine, Decision Tree, and K Nearest Neighbors
  - Those machine learning models were fine tuned with the best hyperparameters, improving their performance
  - Once they were fine tuned, the accuracy of each one of them was calculated, using the test data.

GitHub URL of the notebook: https://github.com/rafael-muniz/dsproject01/blob/c6480e4e716b5f580bc31aa9a977bbc4cdf9f8ce/w4.1%20-%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

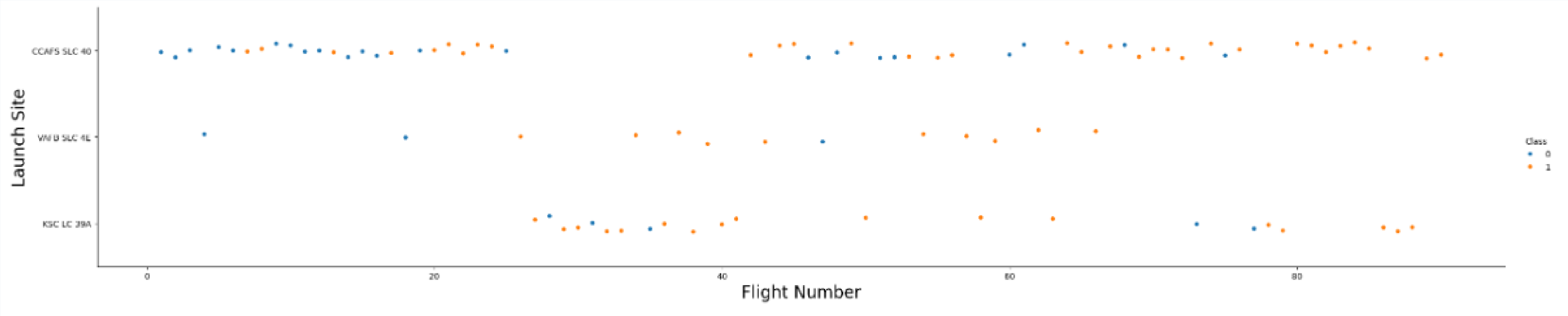
Results

- ES-L1, GEO, HEO, and SSO are the orbits with the best success rates.
- Success rate by Year is in upward trend, because of the increased number of launches SpaceX performs each year.
- KSC LC-39A is the launch site with the most successful number of launches.
- Booster version FT has the most successful launch outcomes
- After fine tuning all the machine learning models, they all had the same accuracy: 83.3%

Section 2

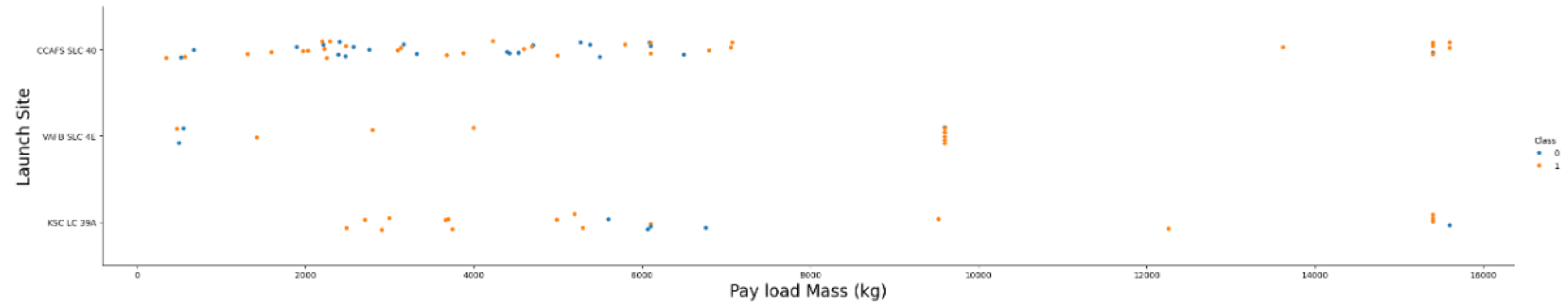
Insights drawn from EDA

Flight Number vs. Launch Site



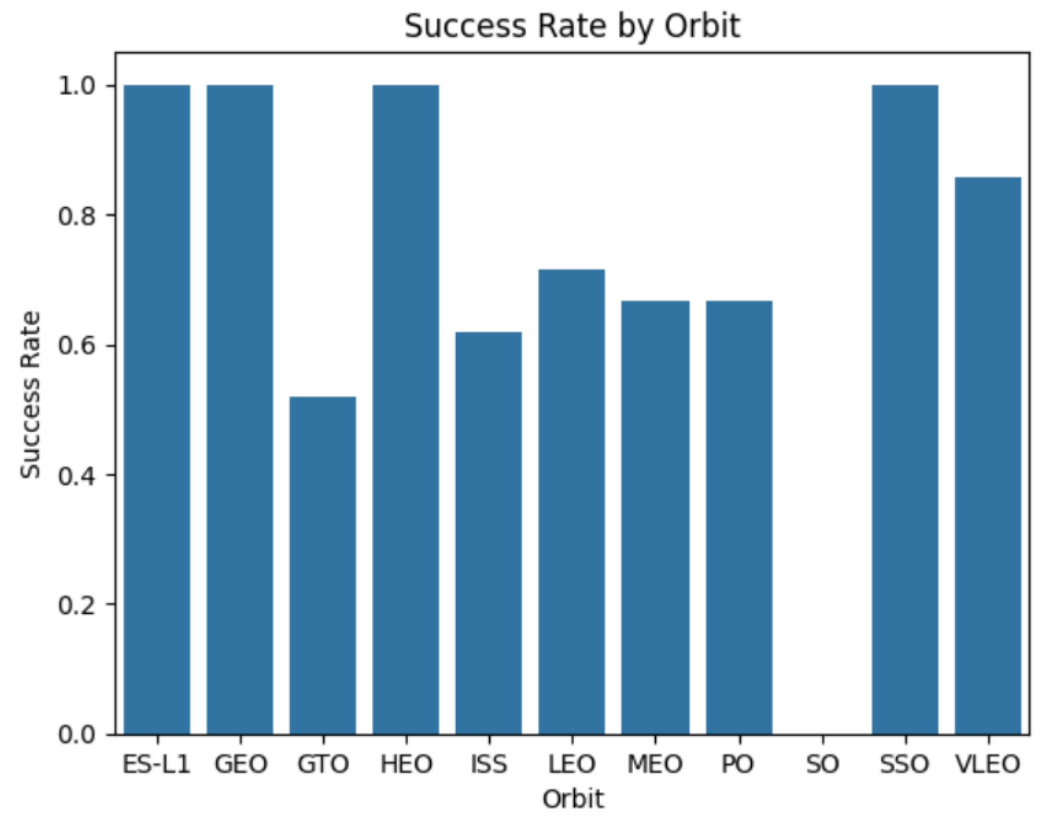
- As the number of flights increase, the success rate (class = 1) tend to increase, no matter where the launch took place. However, we can see that, overall, the launch site KSC LC 39A has the best success ration among them.

Payload vs. Launch Site



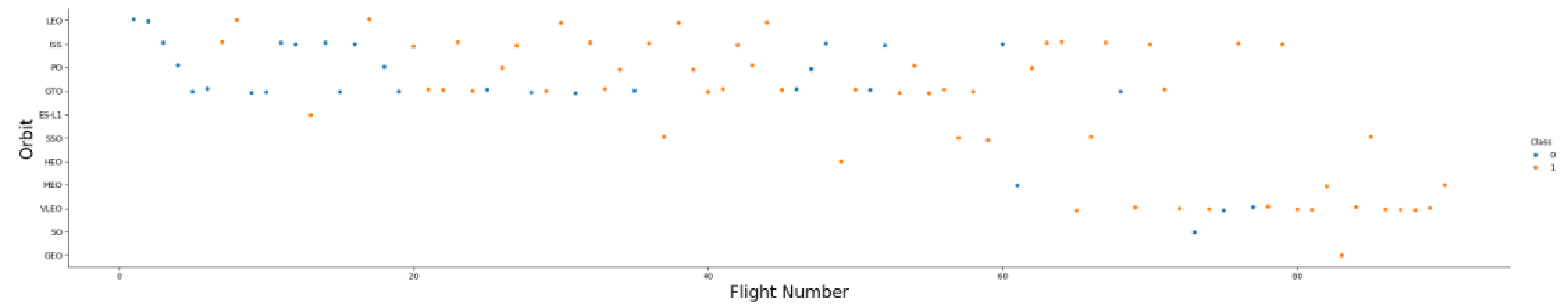
- We can see in this graph that for the VAFB SLC 4E Launch Site there were no rockets launched for a Payload mass greater than 10,000 Kg.

Success Rate vs. Orbit Type



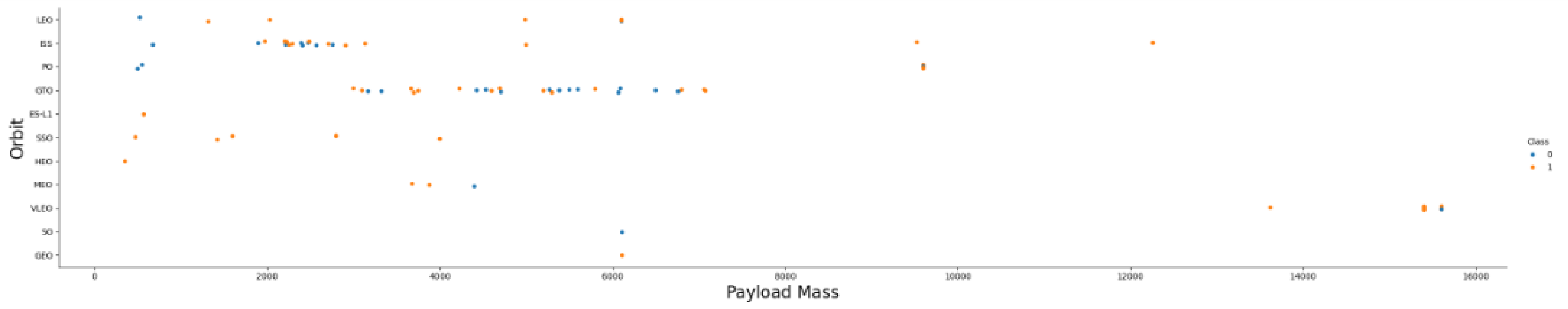
- EL-L1, GEO, HEO, and SSO are the orbits with 100% of landing success rate.

Flight Number vs. Orbit Type



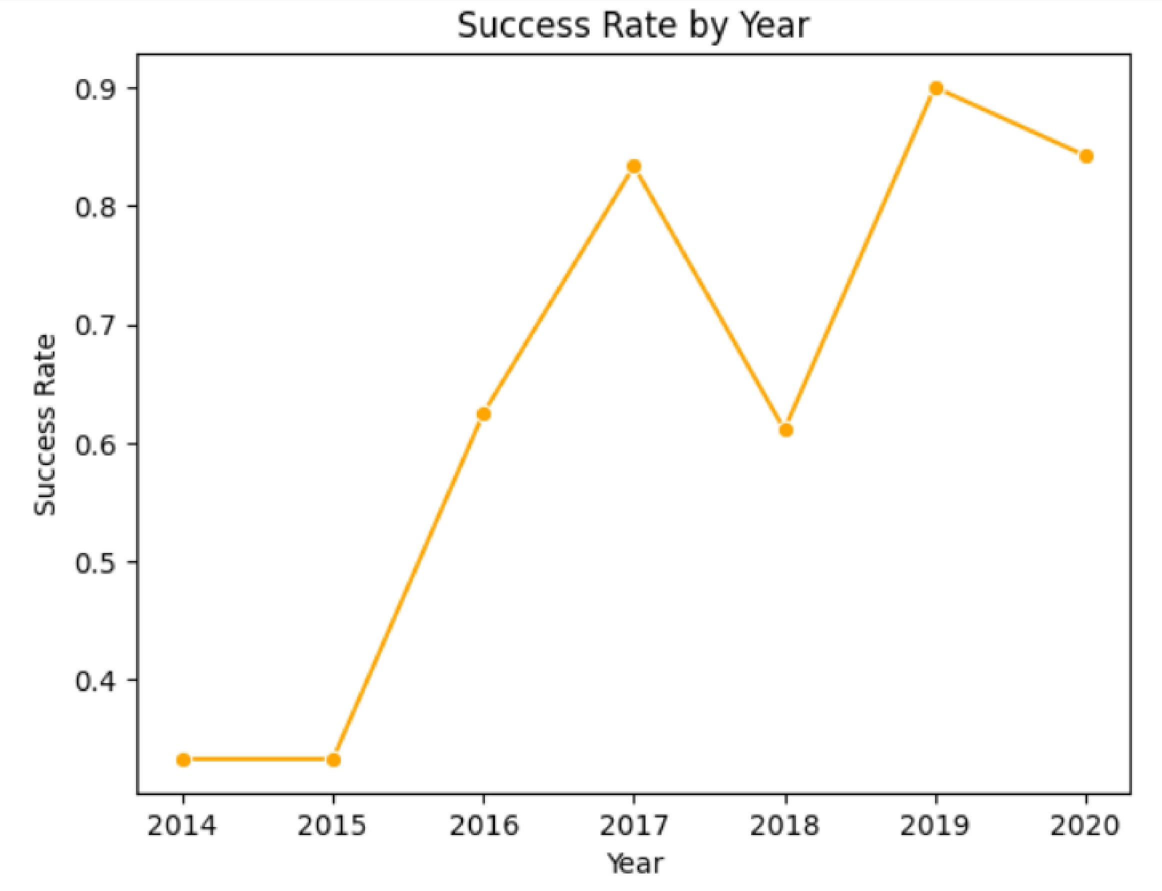
- In the LEO orbit, the success appears related to the number of flights.
- However, in the GTO orbit, there is no relationship between the success rate and the number of flights.

Payload vs. Orbit Type



- We can see that for LEO and ISS orbits there is successful landing rate for the heavier payloads.

Launch Success Yearly Trend



- The success rate is increasing from 2014 until 2020 – upward trend.

All Launch Site Names

- The SELECT DISTINCT statement was used to filter all launch site names from the SpaceX data.

```
In [11]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;

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

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

Launch Site Names Begin with 'CCA'

In [12]:

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

- The LIKE operator was used to filter the launch sites beginning with ‘CCA’ and the operator LIMIT was used to show just 5 entries of the result.

Total Payload Mass

- The total payload mass carried by boosters from NASA was 45596, using the query below (with the statements WHERE and GROUP BY, and with the function SUM):

In [29]:
```
%sql SELECT Customer, SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)' GROUP BY Cu
```

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

Out[29]:

Customer	TotalPayloadMass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,534.67, as shown in the query below (using ROUND, WHERE, and LIKE):

```
In [33]: %sql SELECT Booster_Version, ROUND(AVG(PAYLOAD_MASS__KG_), 2) AS AveragePayloadMass FROM SPACEXTABLE WHERE Booster_Version L

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

Out[33]:

Booster_Version	AveragePayloadMass
F9 v1.1 B1003	2534.67

First Successful Ground Landing Date

- December 12th 2015 was the date of the first successful landing outcome on ground pad was achieved, as observed in the result below:

In [36]:
```
%sql SELECT Landing_Outcome, MIN(Date) AS FirstSuccessfulLandingDate FROM SPACEXTABLE WHERE Landing_Outcome
```

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

```
WHERE Landing_Outcome = 'Success (ground pad)';
```

Out[36]:

Landing_Outcome	FirstSuccessfulLandingDate
Success (ground pad)	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Below we can observe the list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000. The query used had the WHERE clause and AND condition to determine the result.

```
In [37]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MAS

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

Out[37]:

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

%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

Total Number of Successful and Failure Mission Outcomes

- The total number of successful (total = 100) and failure (total = 1) missions outcome can be observed below. For this task the COUNT function, WHERE clause, LIKE and OR operators, and GROUP BY statement, were used.

In [39]:

```
%sql SELECT Mission_Outcome, COUNT(*) AS TotalCount FROM SPACEXTABLE WHERE Mission_Outcome LIKE '%Success%'
```

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

Out[39]:

Saved to this PC

Mission_Outcome	TotalCount
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

```
%sql SELECT Mission_Outcome, COUNT(*) AS TotalCount FROM SPACEXTABLE WHERE Mission_Outcome LIKE '%Success%' OR
Mission_Outcome LIKE '%Failure%' GROUP BY Mission_Outcome;
```

Boosters Carried Maximum Payload

- Below we can observe the names of the booster which have carried the maximum payload mass, using a subquery in the WHERE clause:

```
In [40]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLO

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

Out[40]:

Booster_Version	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

%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

2015 Launch Records

- Combining the usage of the WHERE and AND condition, we can observe the list of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```
In [42]:  %sql SELECT substr(Date, 6, 2) AS Month, substr(Date, 0, 5) AS Year , Booster_Version, Launch_Site, Landing

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

Out[42]:

Month	Year	Booster_Version	Launch_Site	Landing_Outcome
01	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

%sql SELECT substr(Date, 6, 2) AS Month, substr(Date, 0, 5) AS Year , Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome = 'Failure (drone ship)';

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

- Below is the ranking 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, using the WHERE clause, BETWEEN, GROUP BY, and ORDER BY conditions:

In [44]:
```
%sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND
```

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

Out[44]:

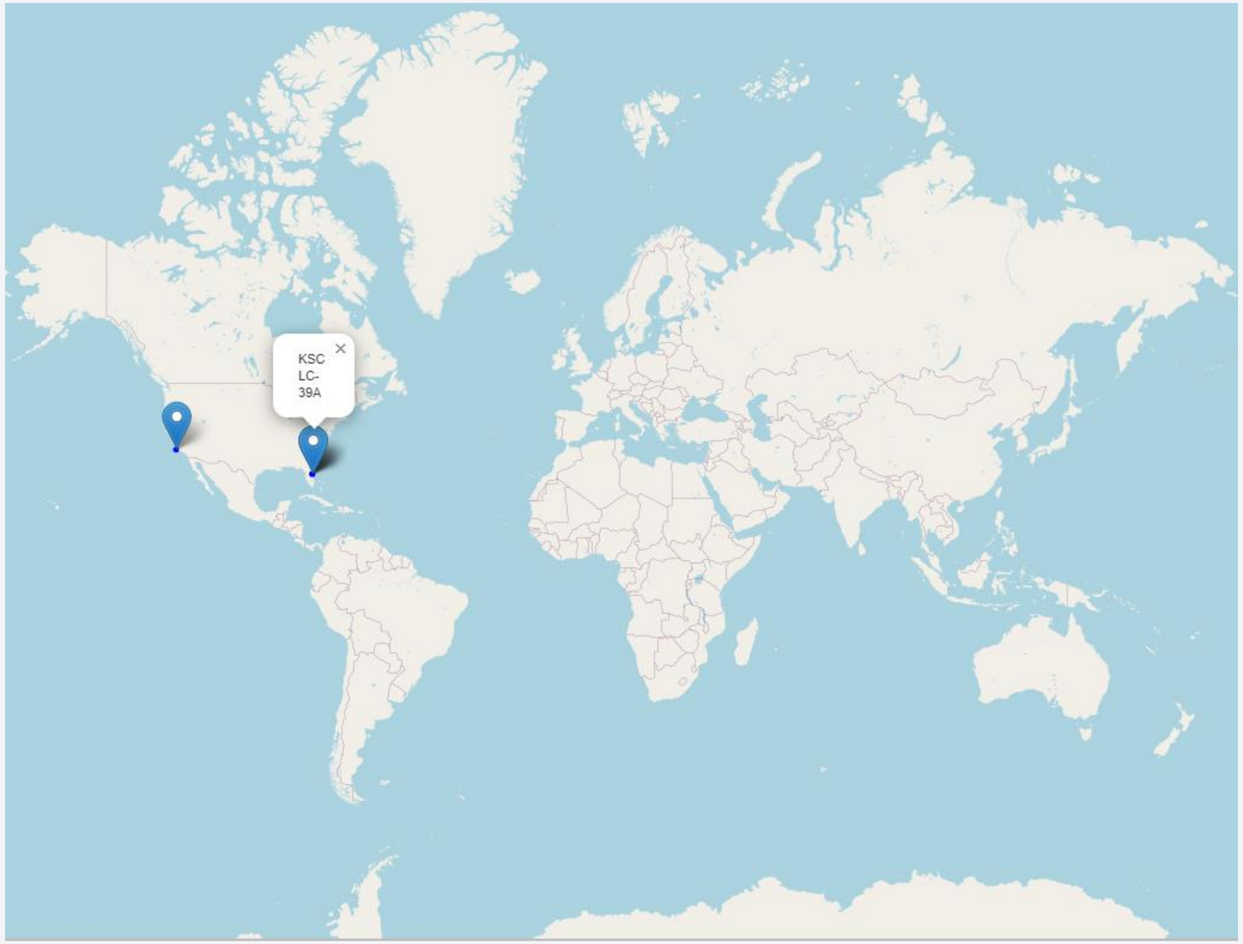
Landing_Outcome	OutcomeCount
Failure (drone ship)	5
Success (ground pad)	3

%sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outcome IN ('Failure (drone ship)', 'Success (ground pad)') GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC;

Section 3

Launch Sites Proximities Analysis

SpaceX’s launch sites

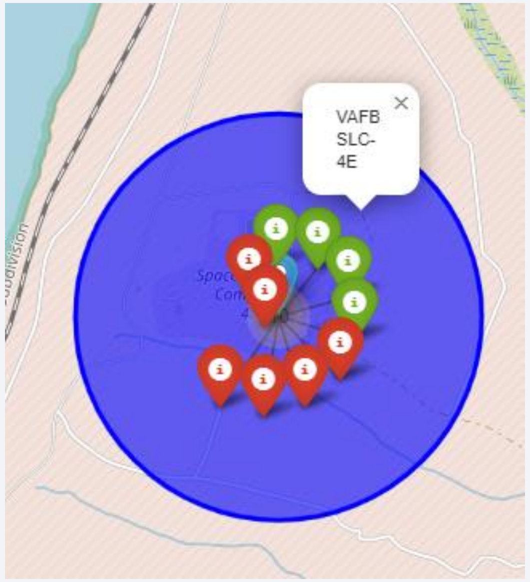


- From this map we can observe that all SpaceX’s launch sites are inside of the US territory.
- This is shown by the blue markers on the map, and if you click in each market it will show the name of the launch site.

Success/failed launches for each site

- Below we can see all of the launch sites, one in California and three in Florida. In each of them, there are markers for successful launches (green) and for failures (red)

<u>California</u>



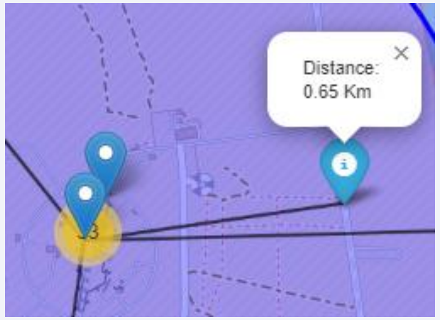
<u>Florida</u>

Launch site and its proximities

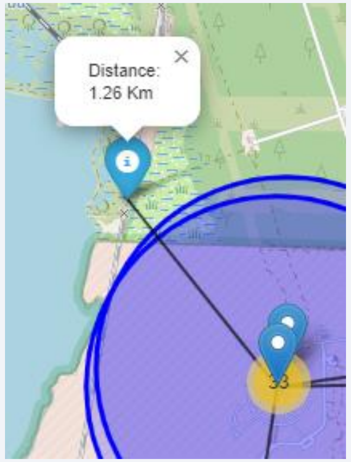
- Below we can observe the distances to key points from one of the launch sites:

Distance to the coastline:

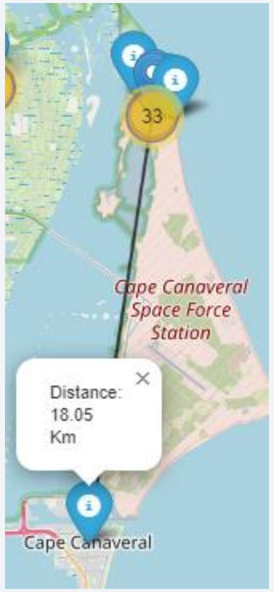
Distance to the highway:



Distance to railway:



Distance to the nearest city:

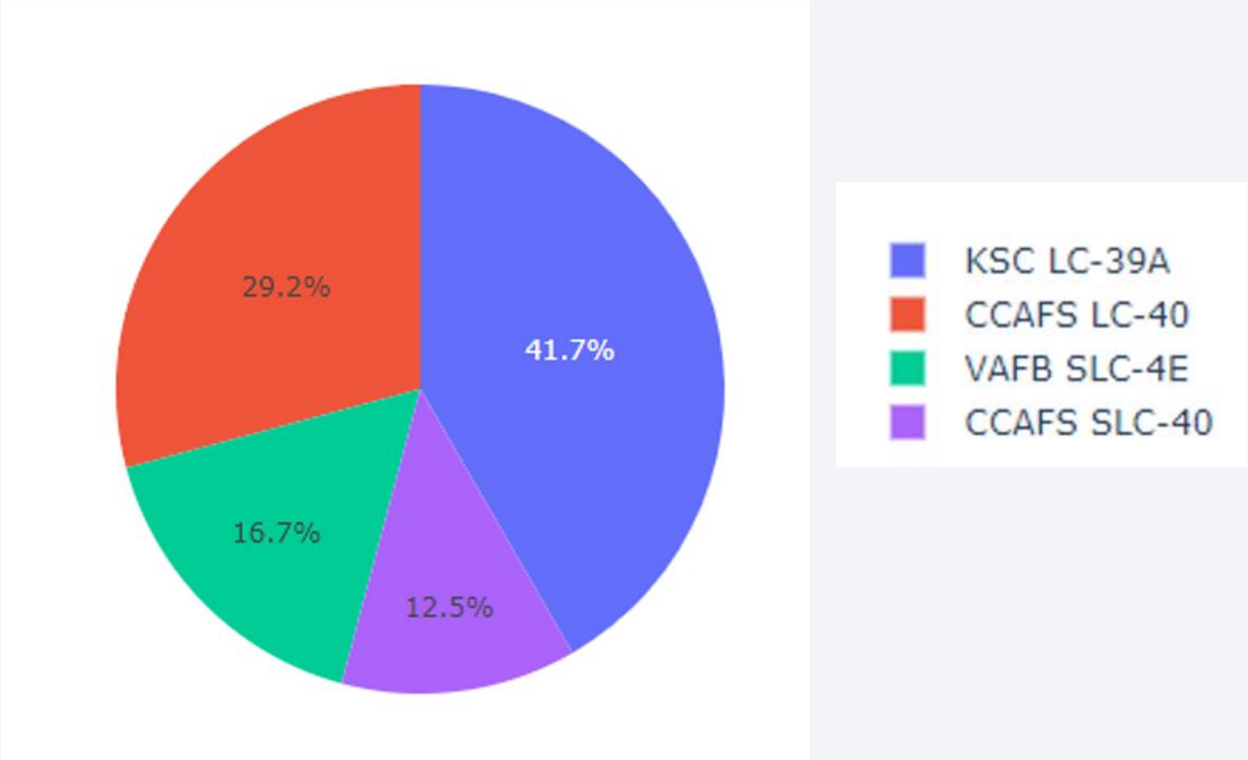


Section 4

Build a Dashboard with Plotly Dash

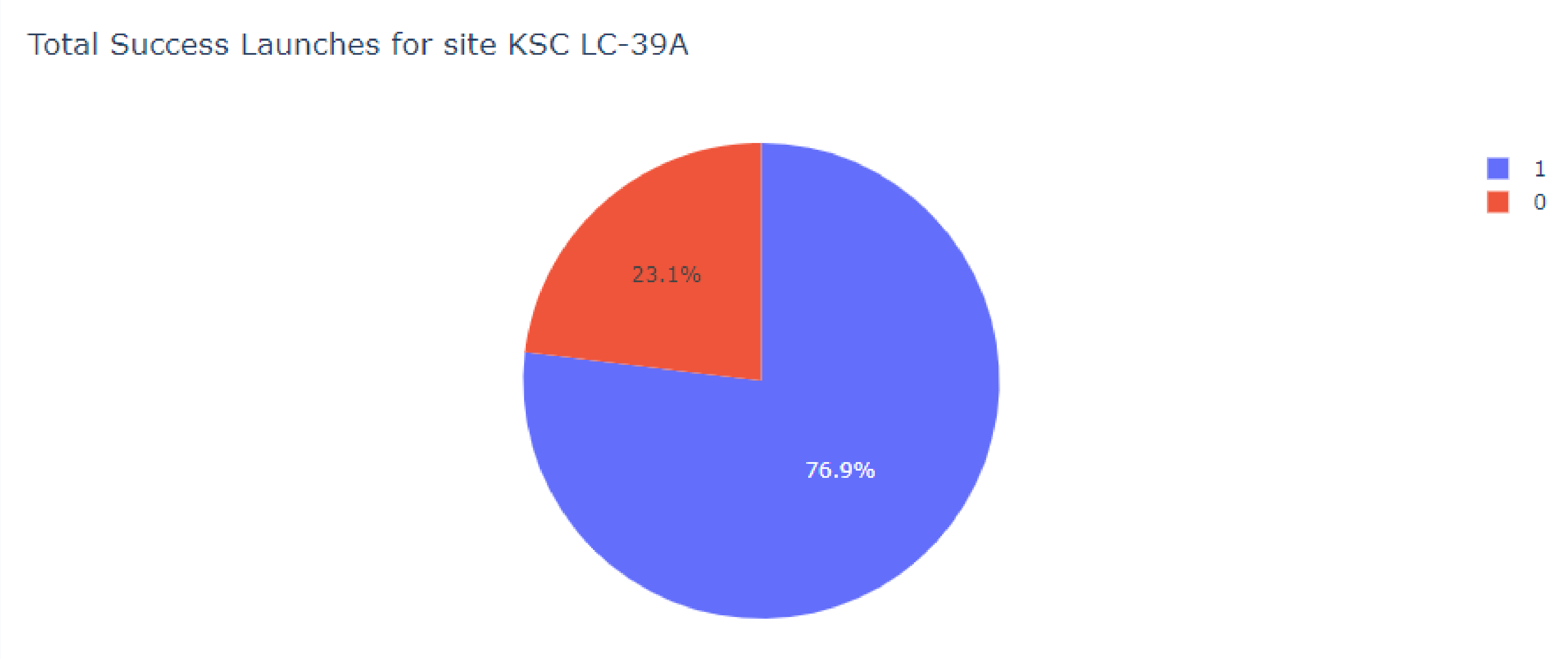
Total success launches by site

- We can observe in the pie chart below that KSC LC-39A is the place that had the most significant successful launches among the launch sites.



Total launches for site KSC LC-39A

- We can see in the graph below that 76.9% of the launches performed in site KSC LC-39A were successful and 23.1% failed.



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

- We can see that in both cases of payload (low and mid range) the Booster version FT is the most successful.

Payload: 0kg – 5000kg

Payload: 2500kg – 7500kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Below, the accuracy of the classification models can by visualized:

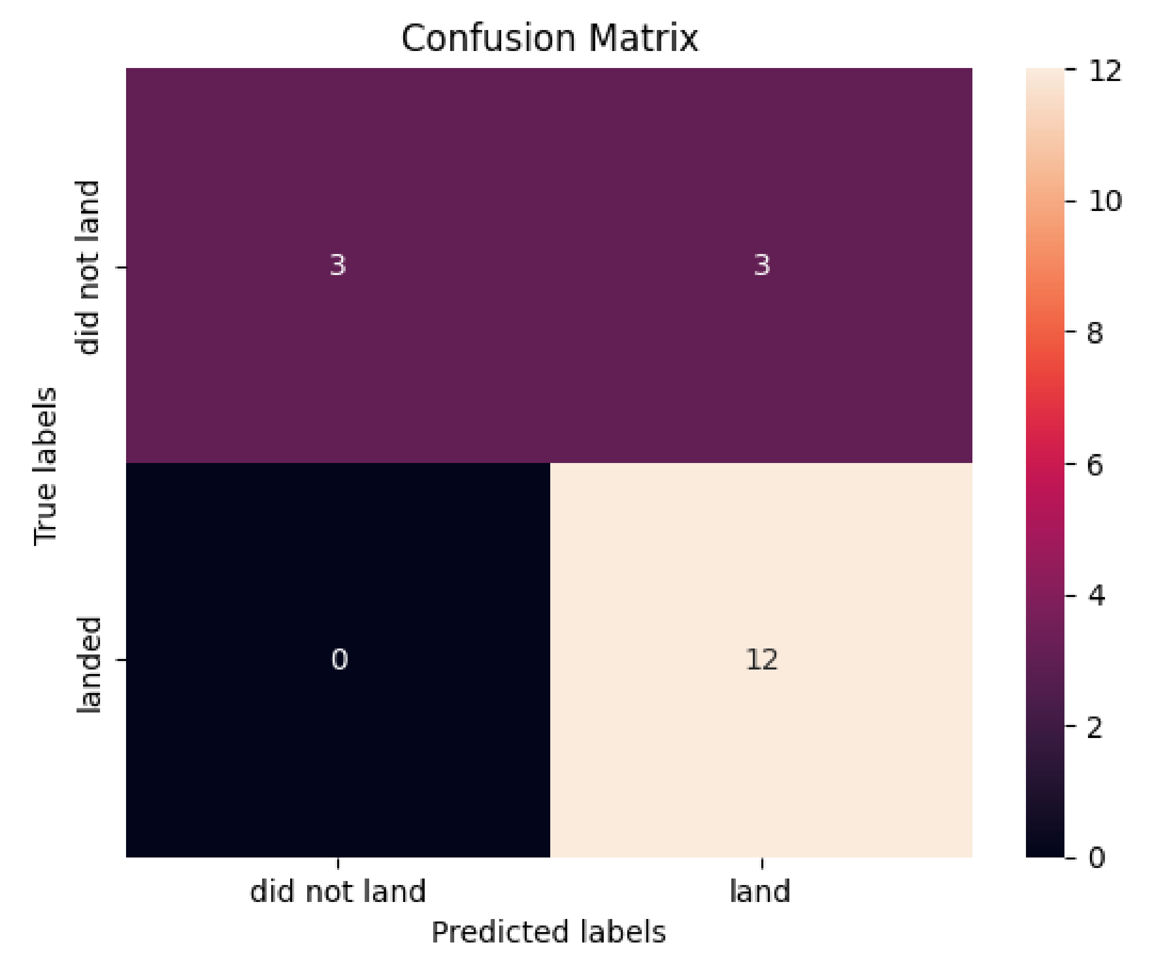
In [40]:
```
logreg_accuracy = logreg_cv.best_estimator_.score(X_test, Y_test)
svm_accuracy = svm_cv.best_estimator_.score(X_test, Y_test)
tree_accuracy = tree_cv.best_estimator_.score(X_test, Y_test)
knn_accuracy = knn_cv.best_estimator_.score(X_test, Y_test)

print("Logistic Regression Accuracy:", logreg_accuracy)
print("Support Vector Machine Accuracy:", svm_accuracy)
print("Decision Tree Accuracy:", tree_accuracy)
print("k nearest neighbors Accuracy:", knn_accuracy)
```

```
Logistic Regression Accuracy: 0.8333333333333334
Support Vector Machine Accuracy: 0.8333333333333334
Decision Tree Accuracy: 0.8333333333333334
k nearest neighbors Accuracy: 0.8333333333333334
```

From the results, we can conclude that all the models have the same accuracy: 83.34%.

Confusion Matrix



- The confusion matrix for the decision tree classifier shows that the model can predict successfully the landed cases, but it has issues in the false positive case (when the classifier marks as successful landing and it was unsuccessful).

Conclusions

- ES-L1, GEO, HEO, and SSO are the orbits with the best success rates.
- Success rate by Year is in upward trend, because of the increased number of launches SpaceX performs each year.
- KSC LC-39A is the launch site with the most successful number of launches.
- Booster version FT has the most successful launch outcomes
- After fine tuning all the machine learning models, they all had the same accuracy: 83.3%
- Better and more precise results we can get if more data is available, which will happen in the future. This is the key element about data science.

Appendix

Libraries used in this project:

1) Pandas
2) Numpy
3) BeautifulSoup
4) Sqlite3
5) Matplotlib
6) Seaborn
7) Folium
8) Dash
9) Plotly
10) Sklearn

Thank you!