IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Miguel Humberto D’Alessandro
Octubre 2022

Outline

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

Executive Summary

- **Summary of methodologies**

1. Data collection
2. Data wrangling
3. EDA with data visualization
4. EDA with SQL
5. Building an interactive map with Folium
6. Building a Dashboard with PlotlyDash
7. Predictive analysis - Classification

- **Summary of all results**

1. Exploratory data analysis results
2. Interactive analytics demo in screenshots
3. Predictive analysis results

Introduction

- **Project background and context**

SpaceX advertises Falcon 9 launches on its website, with a cost of 62 million dollars; other provider cost 165 million dollars each, much of the savings is because SpaceX can reuse the first stage rocket. Therefore, if we can determine if the first stage rocket will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **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

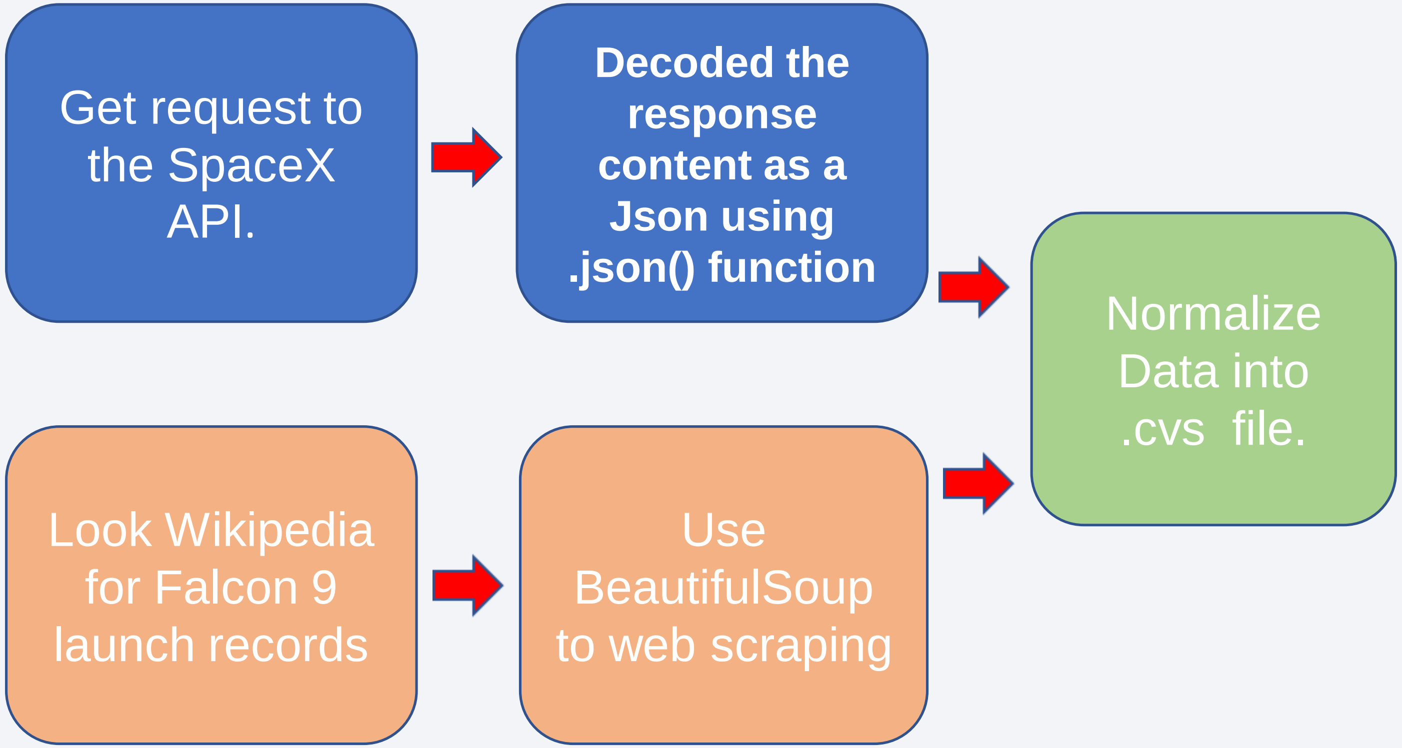
Methodology

Executive Summary

- **Data collection methodology:**
  - SpaceX Rest API
  - Web scrapping from Wikipedia
- **Perform data wrangling**
  - One-hot encoding was applied to categorical features
- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Plotting: Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - How to build, tune, evaluate classification models

Data Collection

- **Data collection was done using:**



The data is collected from the SpaceX REST API, and the API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. Another data is collected from Wikipedia using BeautifulSoup.

Data Collection – SpaceX API

- Used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is: https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/01.1%20-%20Lab%201-%20Collecting%20the%20data-API.ipynb

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

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

```
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()`

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

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

```
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}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.
- The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/01.3%20-%20Lab%201%20-%20jupyter-labs-webscraping.ipynb

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# 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`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, "html.parser")
```

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

```
# Use soup.title attribute
soup.title
```

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

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/01.4%20-%20Lab%201%20-%20jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

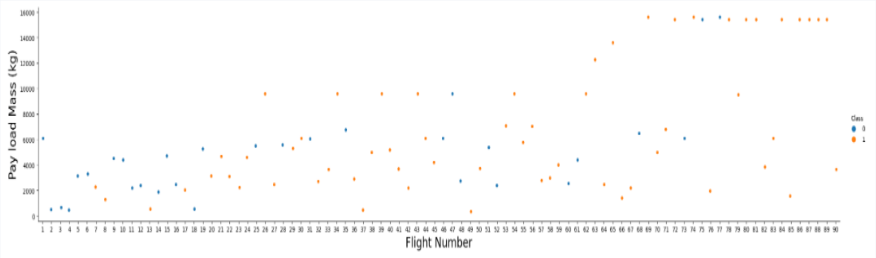
• Scatter Graphs show how much one variables is affected by another. The relationship between two variables is called their correlation. Scatter plots usually consist of a larger body of data. Fight Number VS. Payload Mass, Fight Number VS. Launch Site, Payload VS. Launch Site, Orbit VS. Fight Number, Payload VS. Orbit Type, Orbit VS. Payload Mass

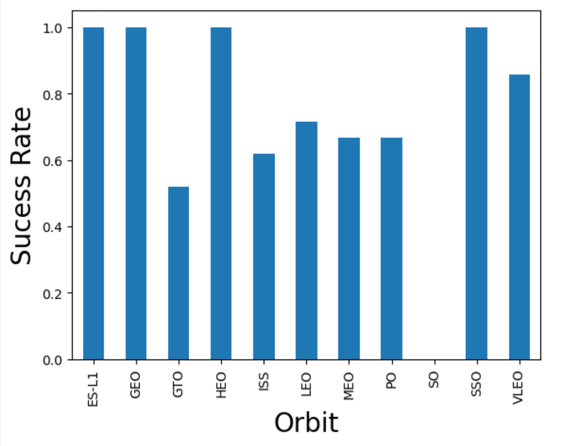
• Bar Graph makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time. Mean VS. Orbit

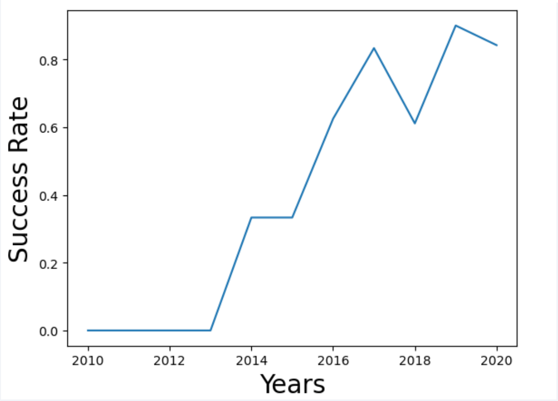
• Line Graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.

• The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/03.1%20-%20jupyter-labs-eda-dataviz.ipynb







EDA with SQL

SQL queries

•Connect to the database

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

•Display 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 successful landing outcome in ground pad was achieved.

•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. Use a subquery

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

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/02%20-%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map objects added to a folium map

- Mark all launch sites on a map: we took the latitude and longitude coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site
- Mark the success/failed launches for each site on the map: we assigned the dataframe launch_outcomes (failures, success) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster
- Calculate the distances between a launch site to its proximities: using Haversine's formula we calculated the distance from the Launch Site to various landmakers to find various trends about what is around the Launch Site to measure patterns. Lines are dran on the map to measure distance to lanmarks
- The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/04%20-%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- 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.
- The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/5e1d4dfc15072536e9ba6e6f55ce975f11dc7426/05%20-%20spacex_dash_app.py

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 tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Found the best performing classification model.
- The link to the notebook is:

https://github.com/Miguelhdg/Applied_Data_Science_Capstone/blob/173800c0a9685ca732d58b534ccaf575a846e5e3/06%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

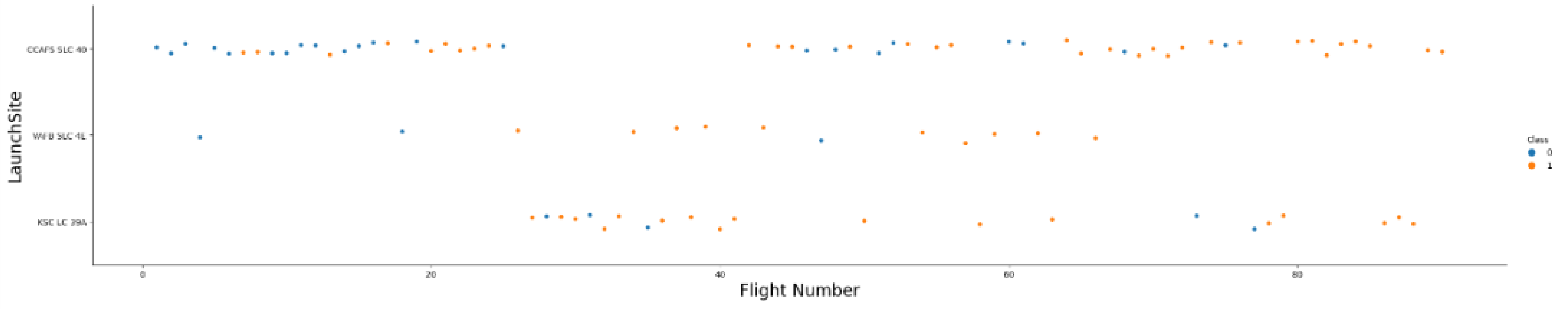
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

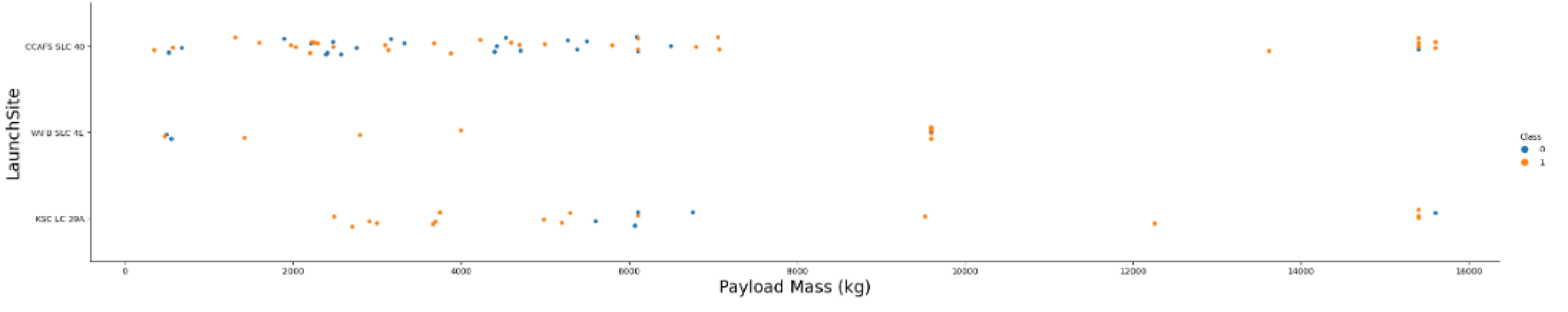
Scatter plot of Flight Number vs. Launch Site



The more amount flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site

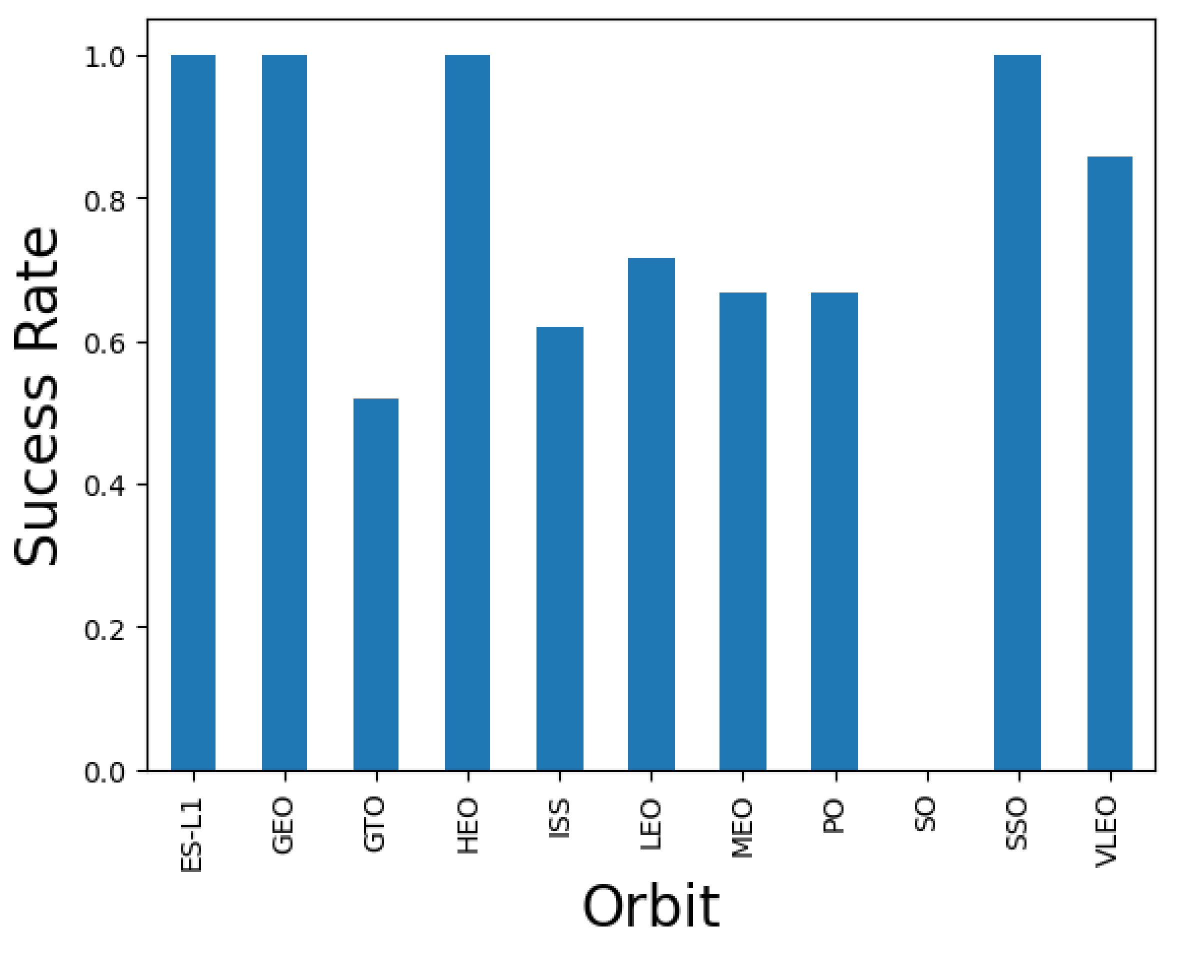
Scatter plot of Payload vs. Launch Site



The greater the payload mass for launch stie CCAFS SLC 40 the higher the success rate for the rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the launch site is dependant on pay load mass for a success launch.

Success Rate vs. Orbit Type

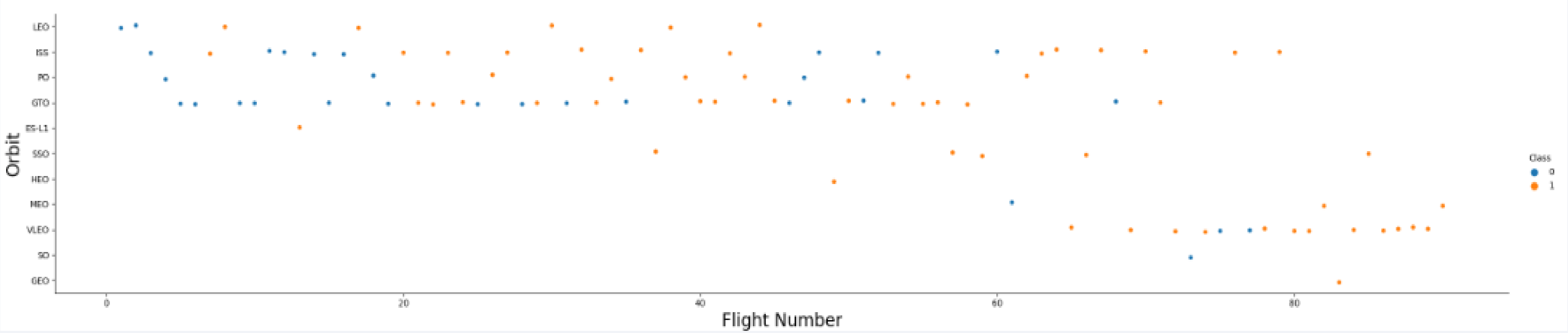
Bar chart for the success rate of each orbit type



From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type

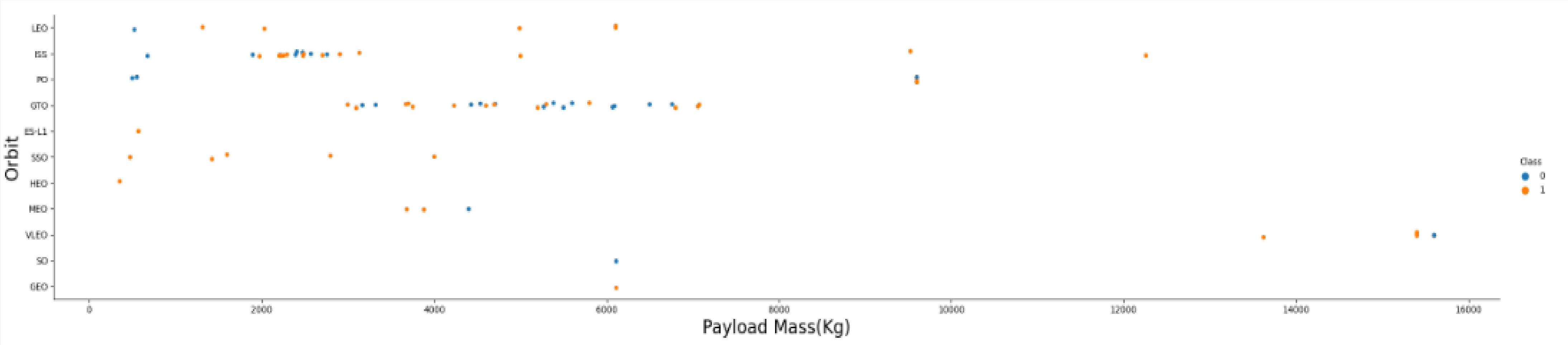
Scatter point of Flight number vs. Orbit type



The plot shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type

Scatter point of payload vs. orbit type

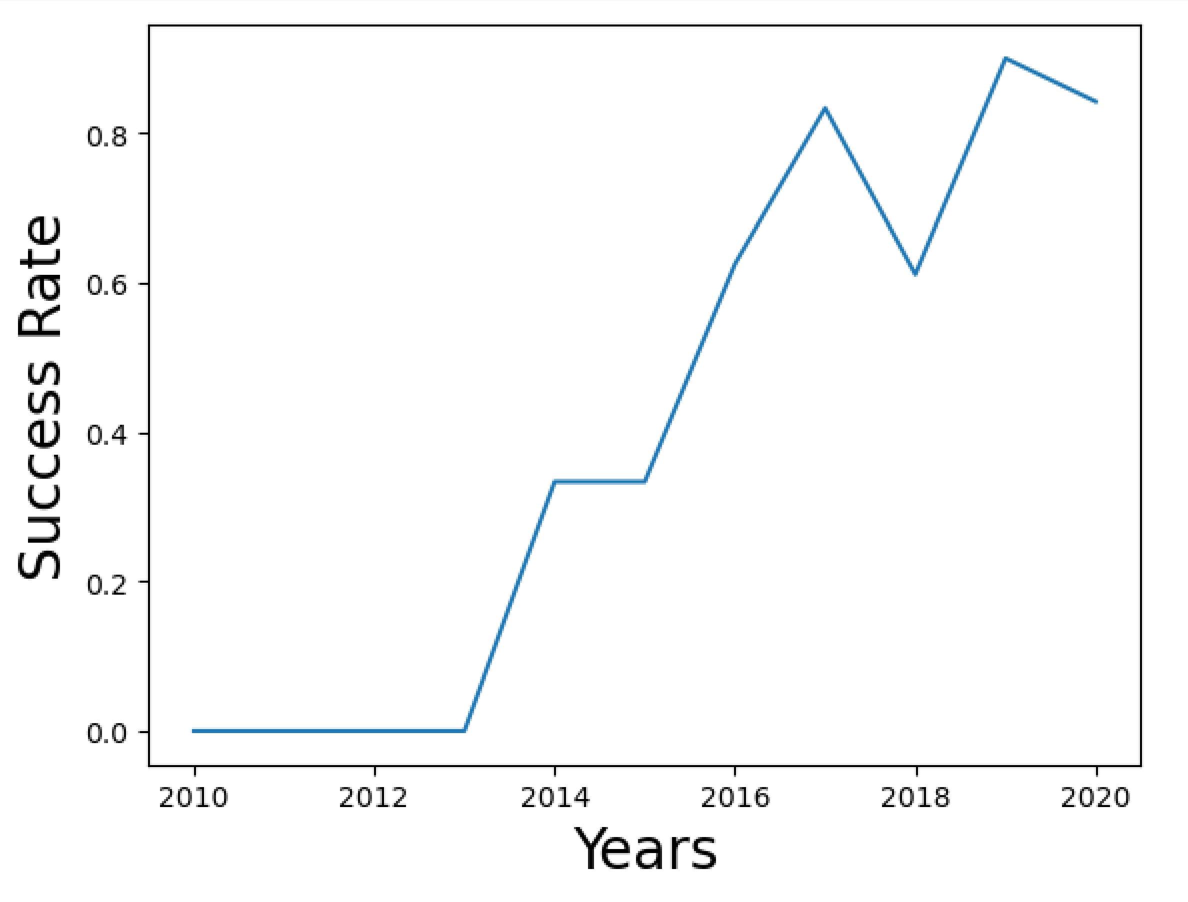


We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

Launch Success Yearly Trend

Line chart of yearly average success rate

We can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

Using the word DISTINCT in the query means that it will only show unique in the LAUNCH_SITE column from SPACEXDATASET

%sql select Unique(LAUNCH_SITE) from SPACEXDATASET;

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

Launch Site Names Begin with 'CCA'

Using the word LIMIT 5 in the query means that it will only show 5 records from SPACEXDATASET and LIKE keyword has a wild card with the words ‘CCA%’ the percentage in the end suggests that the launch_site name must start with CCA

sql SELECT * from SPACEXDATASET 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

Using the function SUM summates the total in the column payload_mass_kg _. The where clause filters the dataset to only perform calculations on customer NASA (CRS)

sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXDATASET WHERE customer = 'NASA (CRS)';

Payloadmass = 45596

Average Payload Mass by F9 v1.1

Using the function AVG works out the average in the column PAYLOAD_MASS__KG_. The where clause filters the dataset to only perform calculate on Booster_Version = 'F9

sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXDATASET where Booster_Version = 'F9

Payloadmass = 2928

First Successful Ground Landing Date

Using the function MIN works out the minimum data in the column DATE, the where clause filters the dataset to only perform calculates on landing__outcome = 'Success (drone ship)’.

sql select min(DATE) from SPACEXDATASET where landing__outcome = 'Success (drone ship)’

2016 - 04 - 08

Successful Drone Ship Landing with Payload between 4000 and 6000

Selecting only Booster_Version the where filters the dateset to landing__outcome = 'Success (drone ship)', the AND clause specifies additional filter conditions payload_mass__kg _ > 4000 AND payload_mass__kg _ < 6000.

sql select Booster_Version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg _ > 4000 AND payload_mass__kg _ < 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

Count shows the total number, group by shows different kinds.

sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXDATASET GROUP BY MISSION_OUTCOME;

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

Subquery put the maximum payload mass, select BOOSTER_VERSION from SPACEXDATASET, and where evaluates PAYLOAD_MASS__KG_= max value.

sql select BOOSTER_VERSION as boosterversion from SPACEXDATASET where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXDATASET);

boosterversion
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

2015 Launch Records

Where filters landing_outcome = 'Failure (drone ship)' and EXTRACT(YEAR FROM

DATE)='2015’, and select shows information

sql SELECT MONTH(DATE), MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXDATASET where landing__outcome = 'Failure (drone ship)' and EXTRACT(YEAR FROM DATE)='2015';

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40

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

Select LANDING__OUTCOME and DATE columns, from SPACEXDATASET, Where filters date between 2010 06 04 and 2017 03 20 and landing_outcome is Failure (drone ship) or Success (ground pad).

%sql SELECT LANDING__OUTCOME, DATE FROM SPACEXDATASET WHERE (landing_outcomeIN ('Failure (drone ship)', 'Success (ground pad)')) AND (DATE BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY DATE DESC;

landing__outcome	DATE
Success (ground pad)	2017-02-19
Success (ground pad)	2016-07-18
Failure (drone ship)	2016-06-15
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Success (ground pad)	2015-12-22
Failure (drone ship)	2015-04-14
Failure (drone ship)	2015-01-10

Section 3

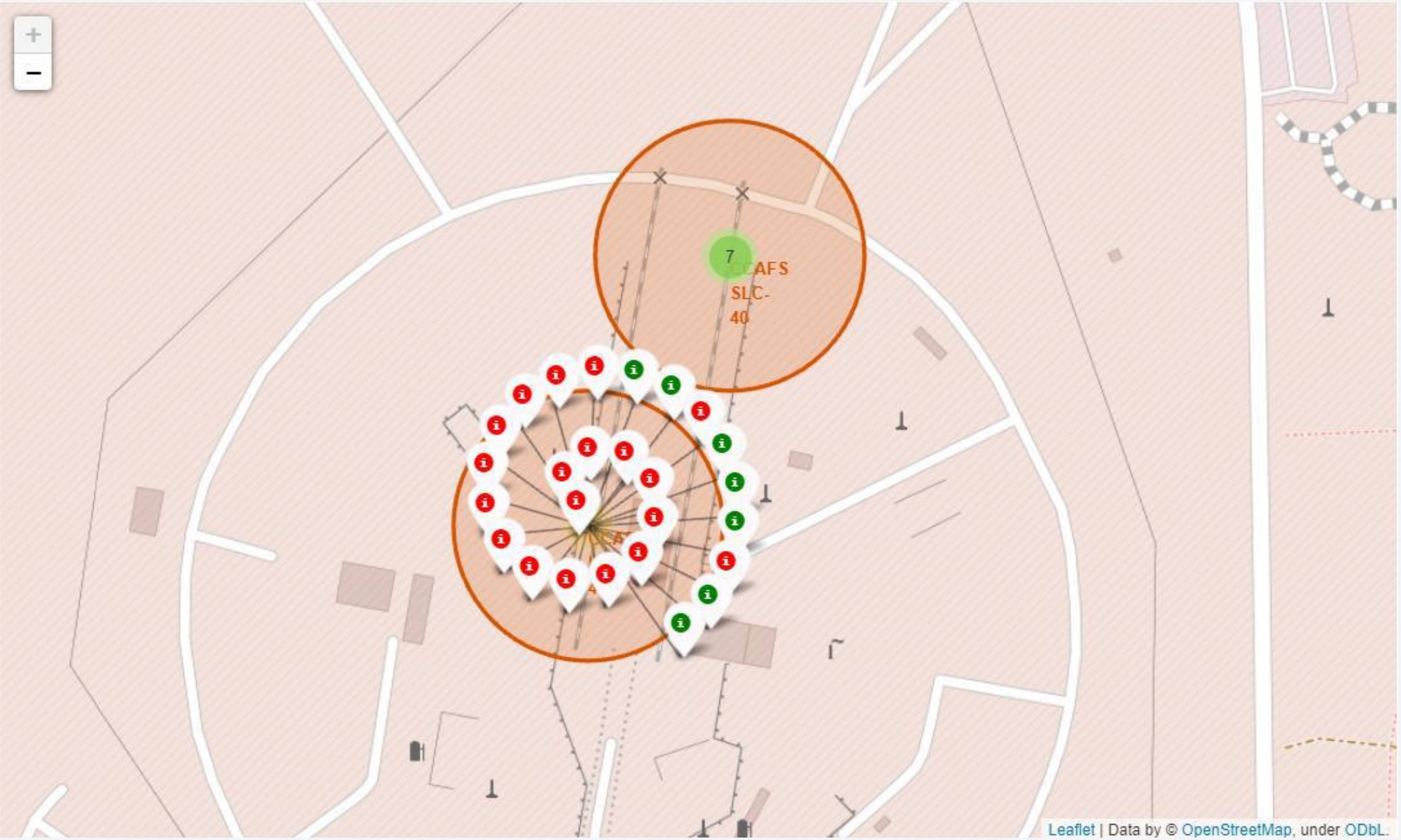
Launch Sites Proximities Analysis

All launch sites global map markers

SpaceX launch sites are in America coasts, Florida, and California.

Markers showing launch sites

Green shows successful and red shows failures.



Launch Site distance to landmarks

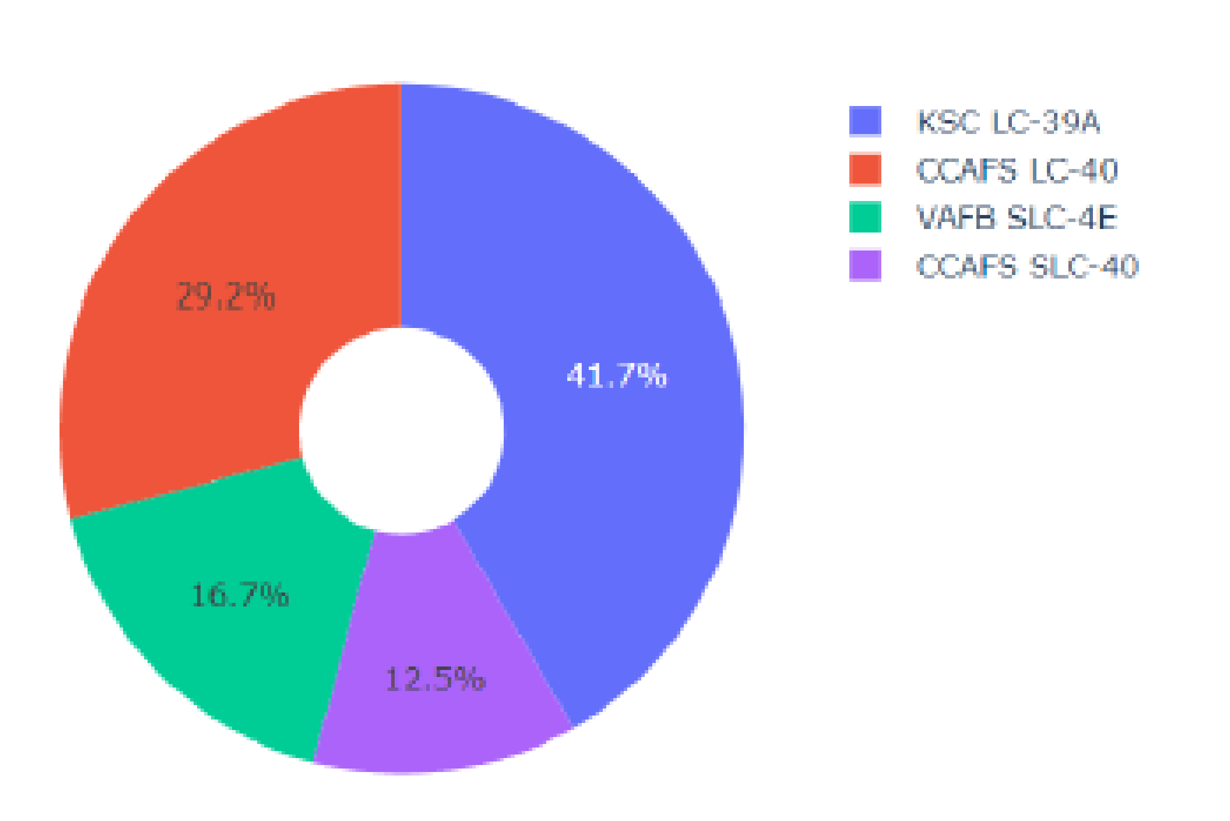
Showing the distance to coasts

Section 4

Build a Dashboard with Plotly Dash

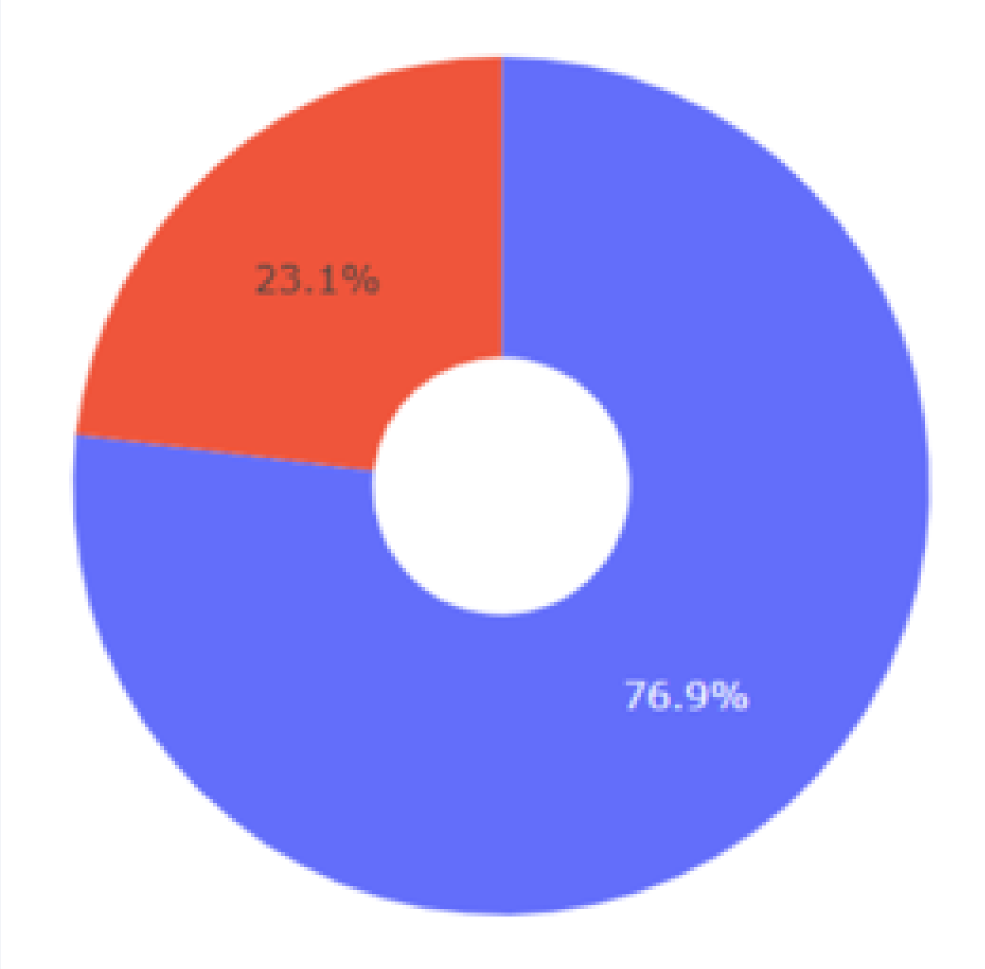
Total Success Launches By all sites

KSC LC-39A had the most successful launches.



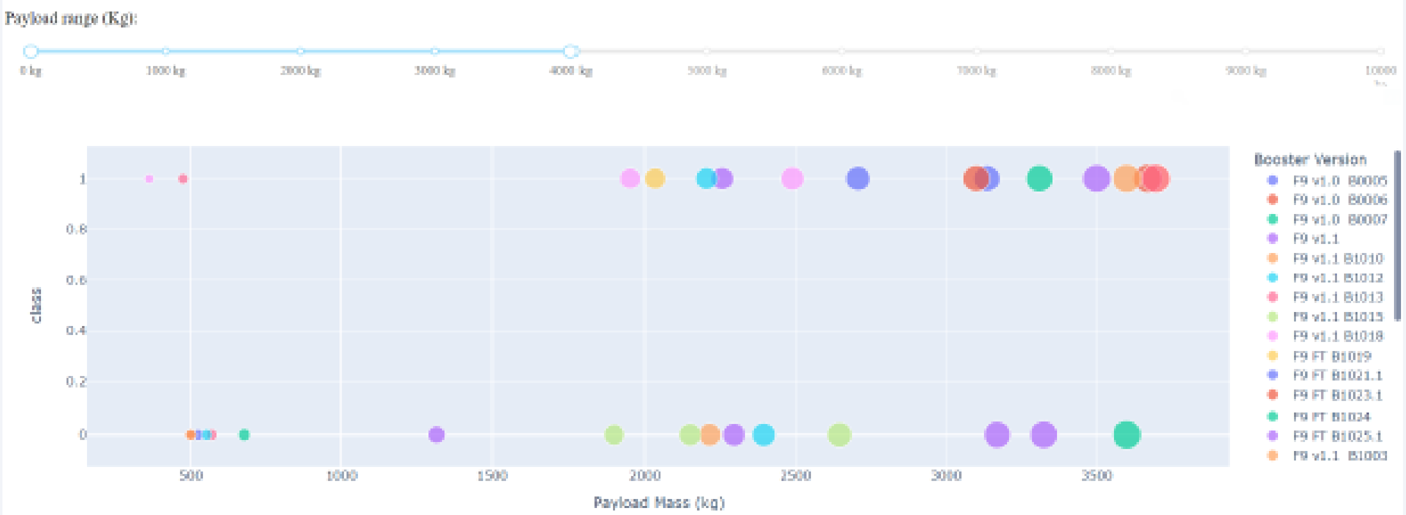
Pie chart showing the Launch site with the highest launch success ratio

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



Payload VS. Mass with different paylaod

The success rates for low weighted payloads is higher than the heavy.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

The Tree is the best for machine learning for this dataset

Find the method performs best:

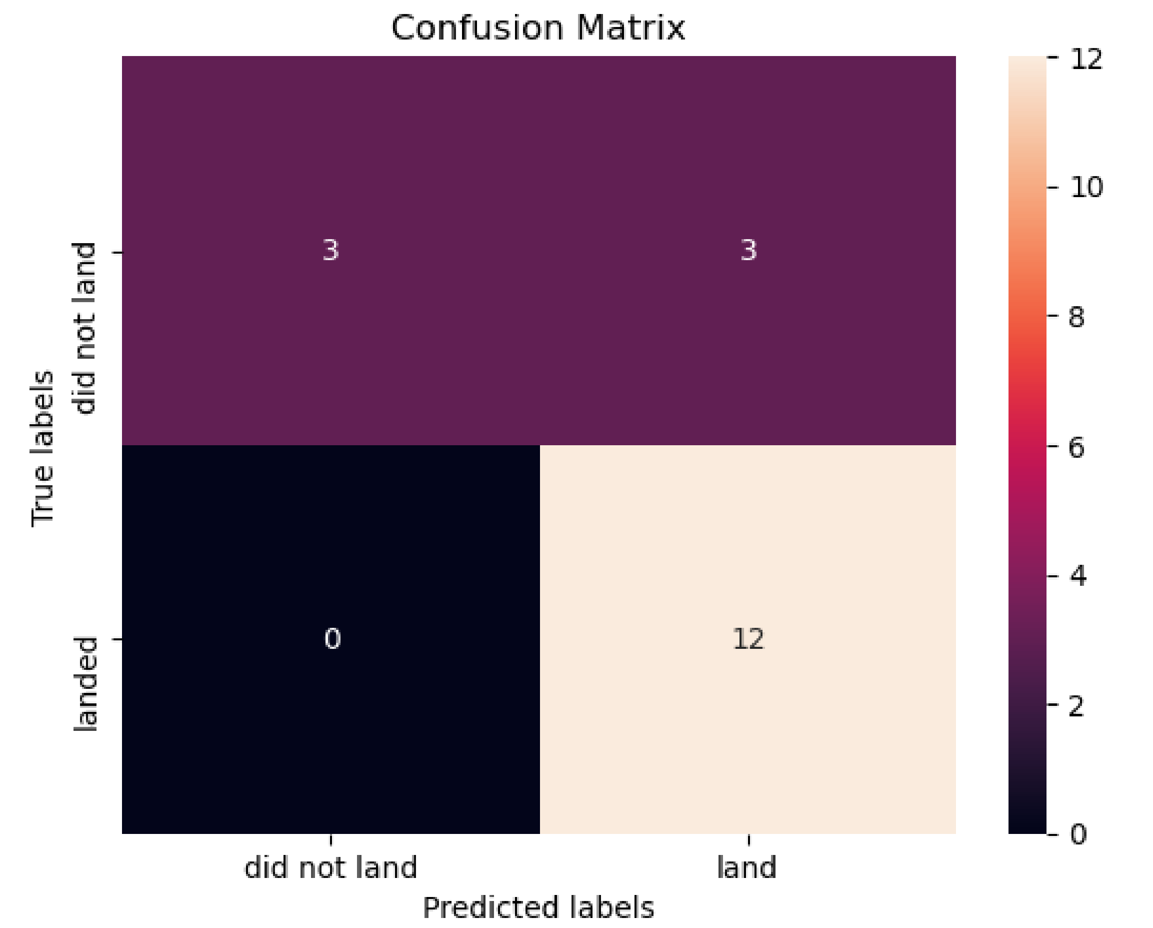
```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

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_)
```

```
Best model is DecisionTree with a score of 0.875
Best params is : {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

Tree can distinguish between the different classes. The major problem is false positives.



Conclusions

1. The Tree is the best for machine learning for this dataset
2. Low weighted payloads perform better than the heavier payloads
3. The success rates for SpaceX launches is proportional time in years they will eventually perfect the launches
4. Launch success rate started to increase in 2013 till 2020.
5. KSC LC-39A had the most successful launches from all the sites
6. Orbit GEO, HEO, SSO, ES-L1 has the best Success Rate.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!