IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rui Alberto Munguambe Júnior
11/20/2022

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
  - 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?
    - Which interaction amongst various features that determine the success rate of a successful landing?
    - What is the impact of each feature to the landing outcome?
    - What operating conditions needs to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Describe how data was collected
- Perform data wrangling
    - Describe how data was processed
- 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

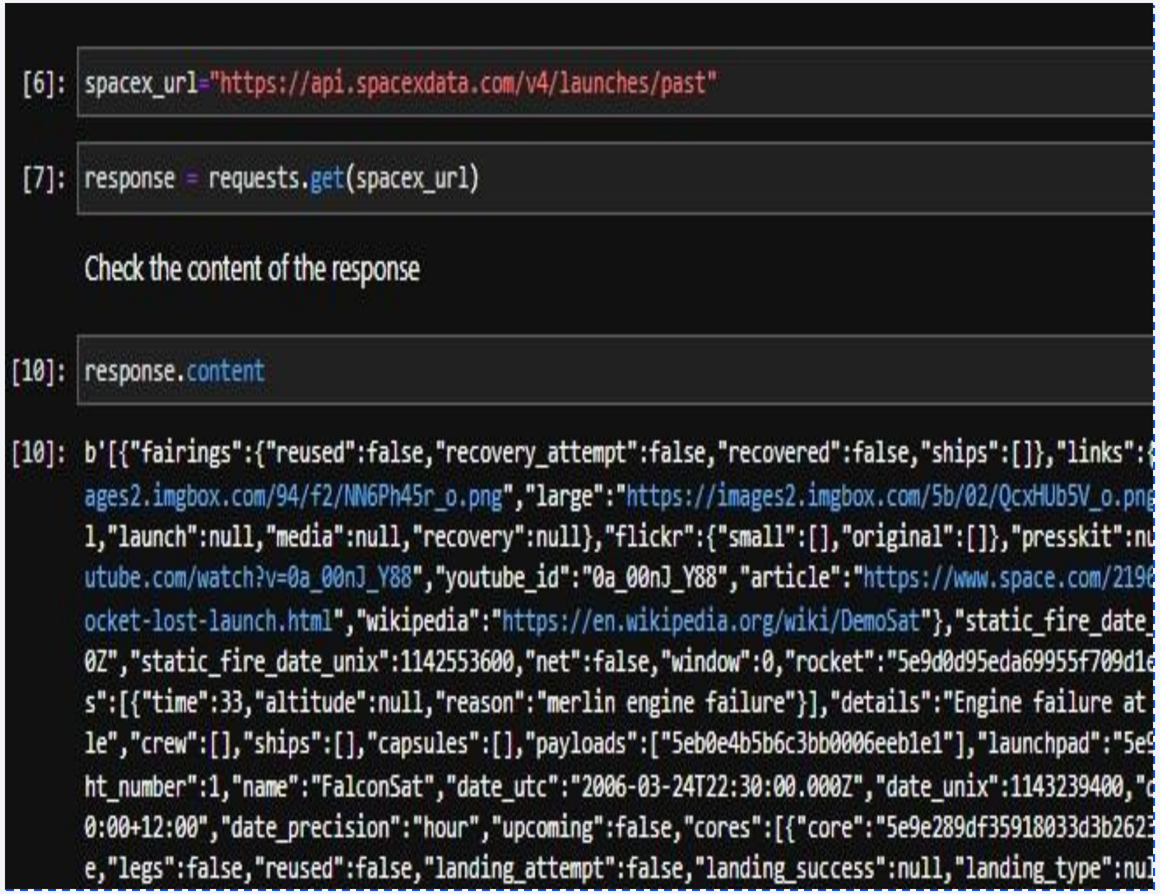
Data Collection

- Describe how data sets were collected.
  - The data was collected using various methods
    - Data collected through the SpaceX API, was collected using the get request function. Then the response content was conveted into a Json using .json() function call and turn it into a pandas dataframe using .json_normalize(). Next, I dealt with the missing values by replacing them with the mean.
    - Web scraping from Wikipedia for Falcon 9 launch records using BeautifulSoup. The goal was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- To collect data, I used the SpaceX API, to which I made a request and extracted the required data from the response. After collecting the data, I dealt with the missing values and then stored the data into a csv file.

- The GitHub URL of the completed SpaceX API calls notebook is https://github.com/RAM-Jr/My-Notebooks/blob/d9e15eb64c362aedc65f7fce1cc065ffb64ade54/DS%20applied%20capstone/assets/notebook/notebook_Data_Collection_bVkK_ACm7.ipynb



Data Collection - Scraping

- I applied web scrapping to scrape Falcon 9 launch records in tables with BeautifulSoup.
- After extracting the data from the tables on the webpage, I stored the data into a pandas dataframe and then saved the data into a csv file.
- The GitHub URL for the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_Data_Collection_Webscraping_hBtds8vUd.ipynb

```
[6]: # 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`

```
[7]: # 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

```
[8]: # Use soup.title attribute
     soup.title
```

```
[8]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

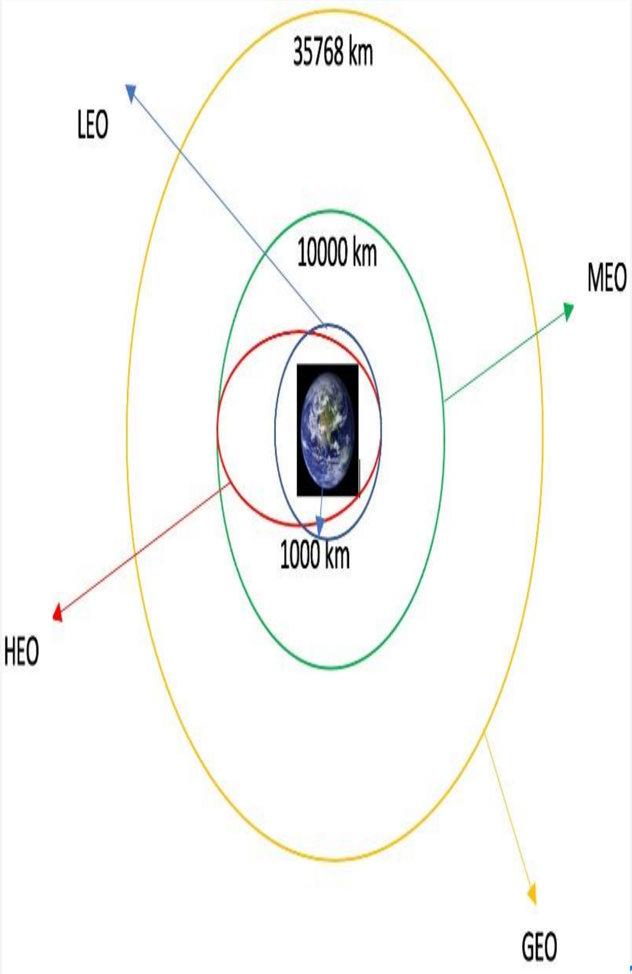
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please reference link towards the end of this lab

```
[9]: # 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.

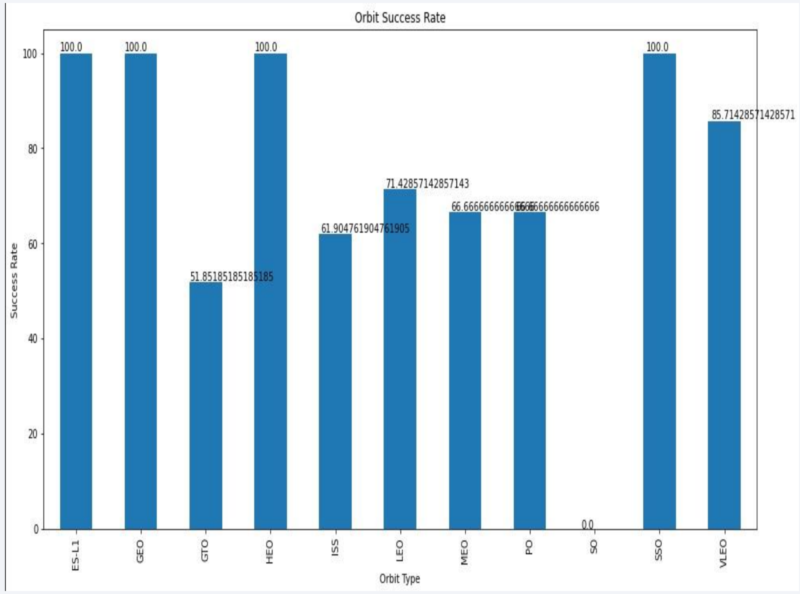
Data Wrangling

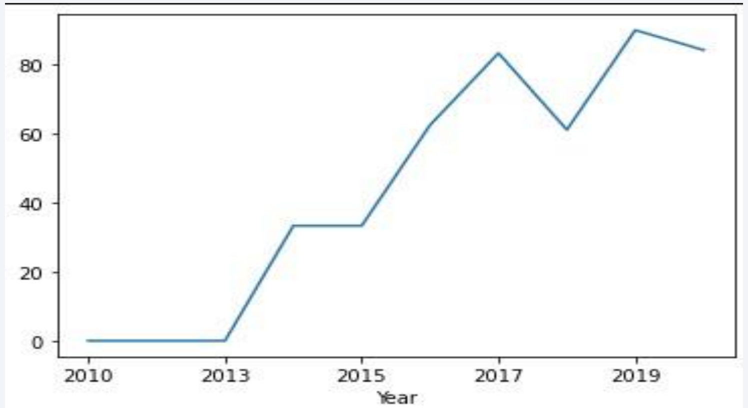
- I performed exploratory Data Analysis and determine Training Labels.
- During this process, I completed the following tasks:
    - Calculated the number of launches on each site;
    - Calculated the number and occurrence of each orbit;
    - Calculated the number and occurence of mission outcome per orbit type; and
    - Created a landing outcome label from Outcome column, and saved the updated data into a csv file.
- The link to the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_EDA_wK_OsUb7i.ipynb



EDA with Data Visualization

- The exploratory data analysis was made based on the visualization of different features such as flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, and the launch success yearly trend. Through the plots I understood the relationship of the independent features and the dependent feature.
- The link to the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_EDA_with_Data_Visualization_LfDszlXlV.ipynb





EDA with SQL

- The SpaceX dataset was stored into a table on the db2 cloud database and then accessed from a jupyter notebook.
- I made EDA with SQL through queries to find out the following:
  - 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.
- The link to the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_EDA_with_SQL_vXoIpTQR7.ipynb

Build an Interactive Map with Folium

- I marked all launch sites using map objects such as markers, circles, lines with different proprieties to indicate the success or failure of launches for each site on the folium map. • I assigned mapped each value of the feature launch outcomes to the feature class with values 0 and 1, where, 0 is for failure, and 1 for success.
- I created color-labeled marker clusters, to identify which launch sites have relatively high success rate.
- And calculated the distances between a launch site to its proximities. Through the results, I was able to answer the following questions:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?
- The link to the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_lab_jupyter_launch_site_location_iI6YpCCl6.ipynb

Build a Dashboard with Plotly Dash

- I built an interactive dashboard with Plotly dash, on which I plotted:
  - Pie charts showing the total successful launches for each launch site, and the total successful and failed launches for each of the launch sites;
  - 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/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/spacex_dash_app.py

Predictive Analysis (Classification)

- I imported the libraries that would be used, loaded and transformed the data, split it into train and test data;
- After spliting the data, I used the train data to train the logistic, knn, svm, and decision tree machine learning models, and to find the best parameters for each of these models, I used the GridSearchCV to tune the hyperparameters.
- After training the model with the best parameters, I used the test data to get the out of sample accuracy.
- The link to the notebook is https://github.com/RAM-Jr/My-Notebooks/blob/main/DS%20applied%20capstone/assets/notebook/notebook_SpaceX_Machine_Learning_Prediction_Part_5_Z6mWfAmr8.ipynb

Results

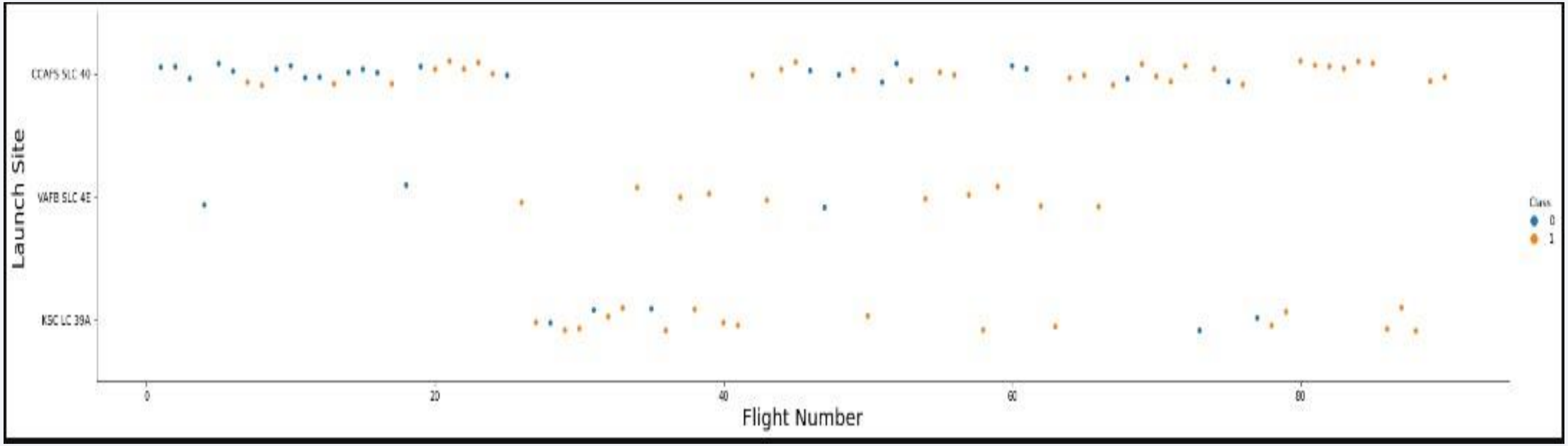
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

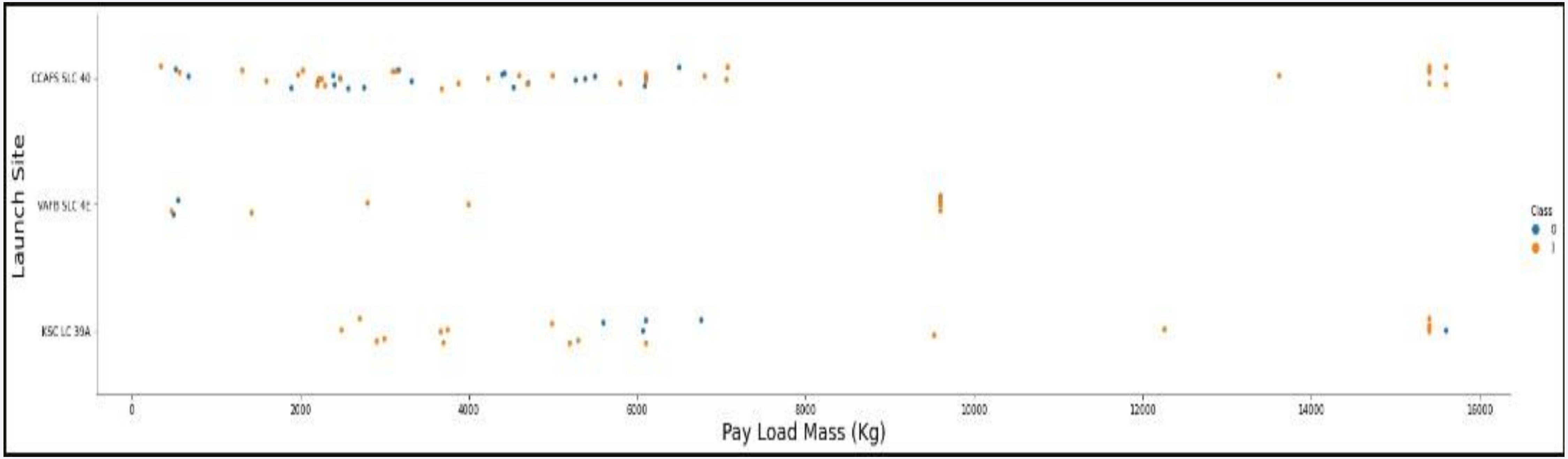
Flight Number vs. Launch Site

- It can be seen thar for `CCAFS LC-40`, there's a high chance of landing successfuly if the `Flight Number` is higher than 80. For `VAFB SLC 4E` there are no `Flight Number's` higher than 70, and there's higher chance of success for `Flight Number` with values from 20 to 45 and values higher the 55. Like `CCAFS LC-40`, there's higher success rate for `KSC LC-39A` in with a high value for `Flight Number` and values between 35 and 65.



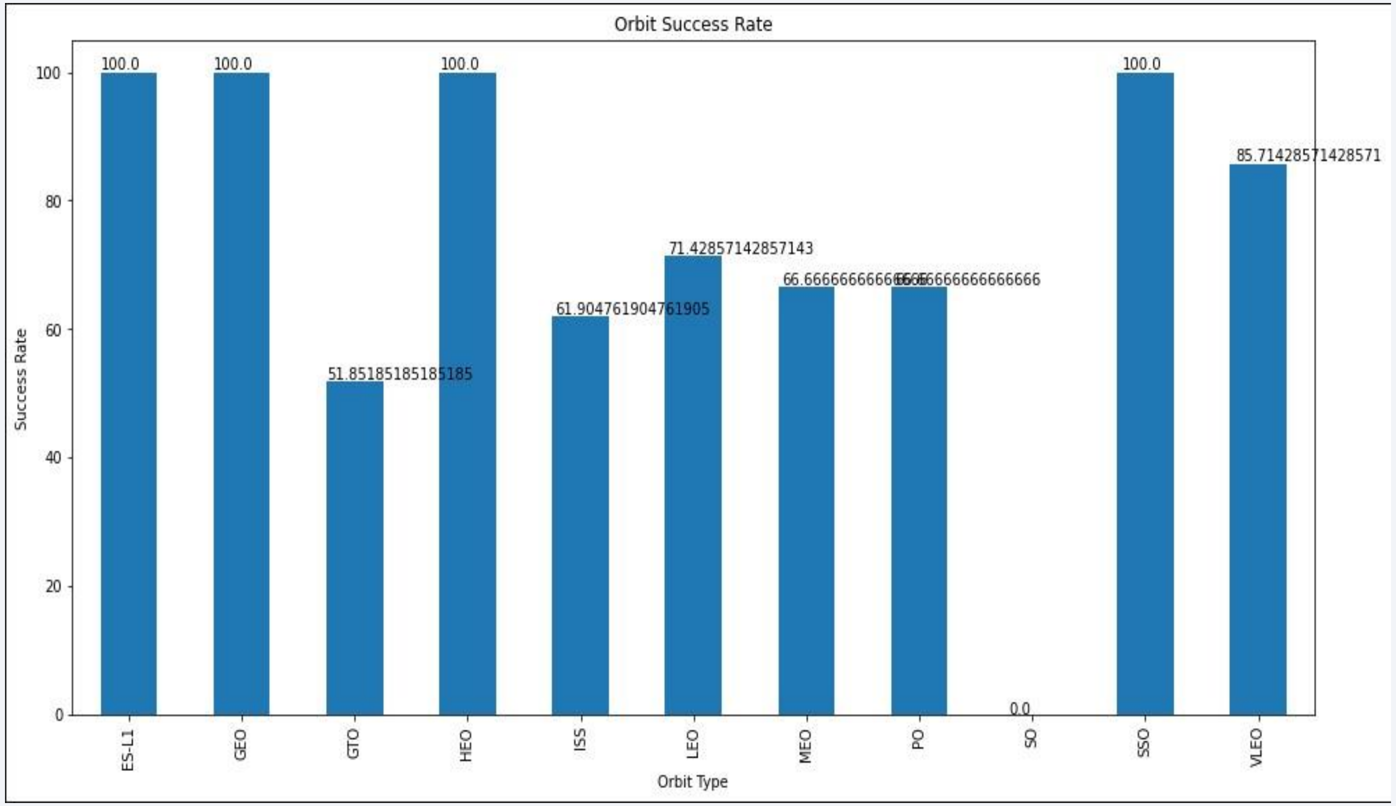
Payload vs. Launch Site

- Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



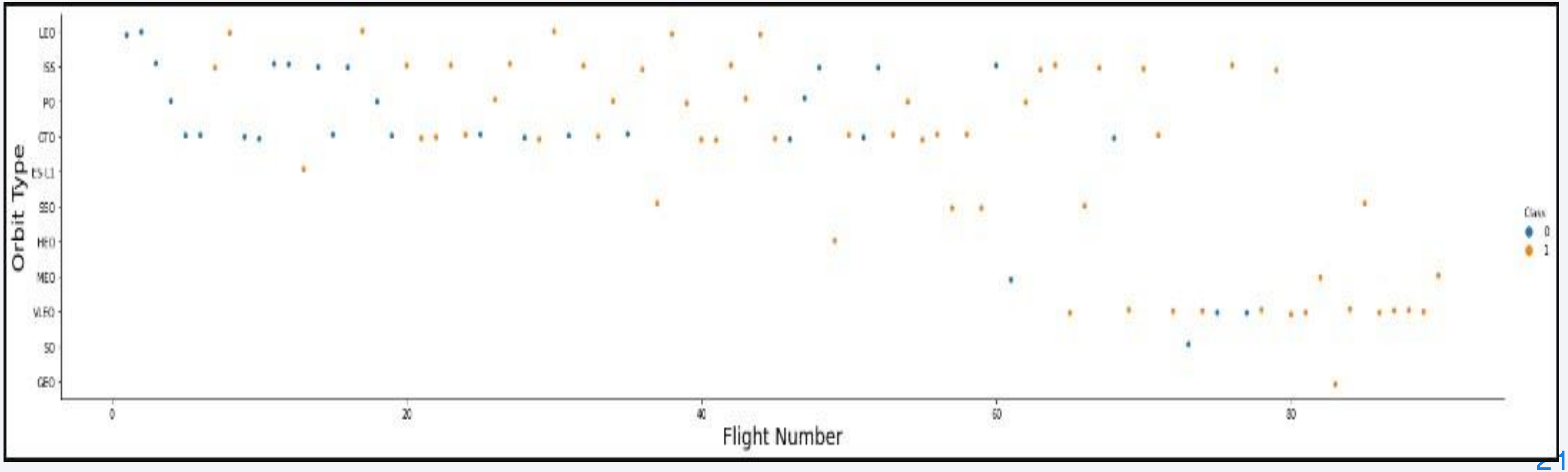
Success Rate vs. Orbit Type

- It can be seen that there's a 100% success rate for the following orbit types: ES-L1, GEO, HEO and SSO. The orbit type with the worst success rate is GTO, with a success rate of 51.85%



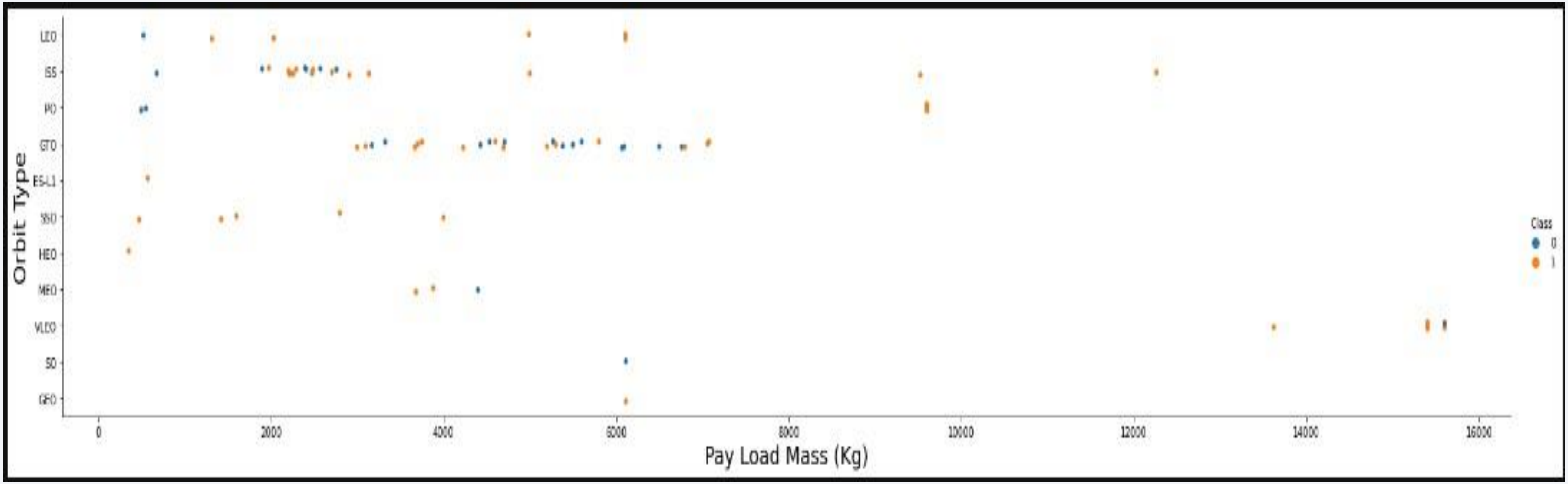
Flight Number vs. Orbit Type

- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



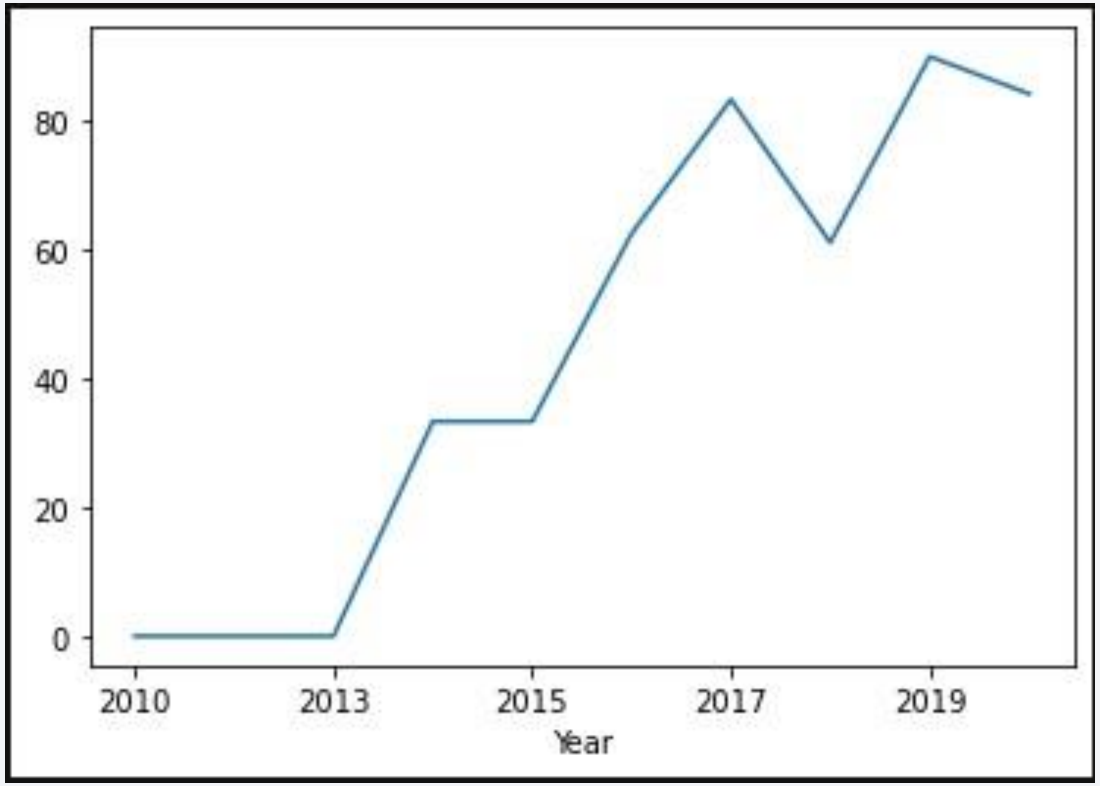
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- you can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- This is the query results for the names of the unique launch sites

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

Launch Site Names Begin with 'CCA'

- The first 5 records where launch sites begin with `CCA`

Out[16]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Total Payload Mass

- The total payload carried by boosters from NASA

Out[19]:

1
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad

Out[38]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Out[36]:

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

Out[61]:

landing_outcome	total
Failure	3
Success	38

Boosters Carried Maximum Payload

- The names of the booster_versions which have carried the maximum payload mass

Out[66]:

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

2015 Launch Records

- List of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Out[67]:

booster_version	launch_site	landing_outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Rank of 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

Out[81]:

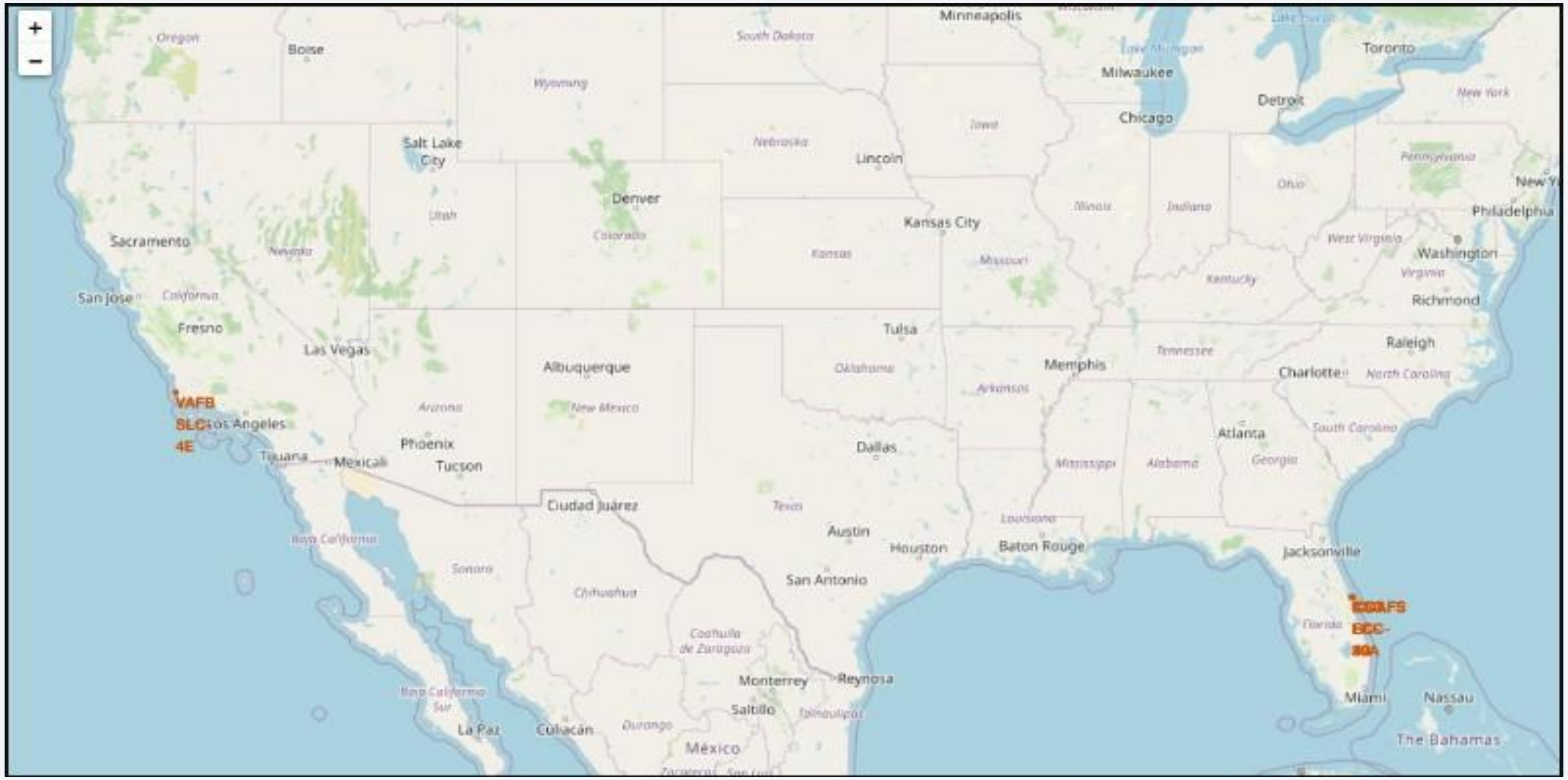
landing_outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

- All launch sites are very close proximity to the coast.



Launch sites with color labels

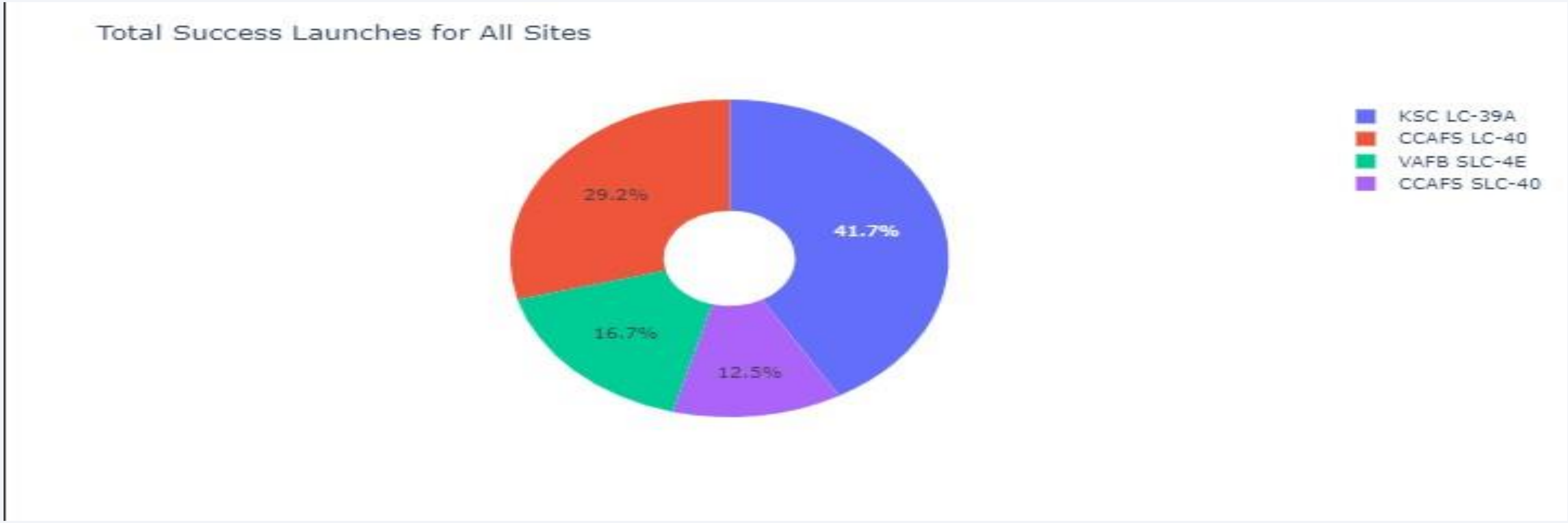
- Green shows success, and red shows failure.

Section 4

Build a Dashboard with Plotly Dash

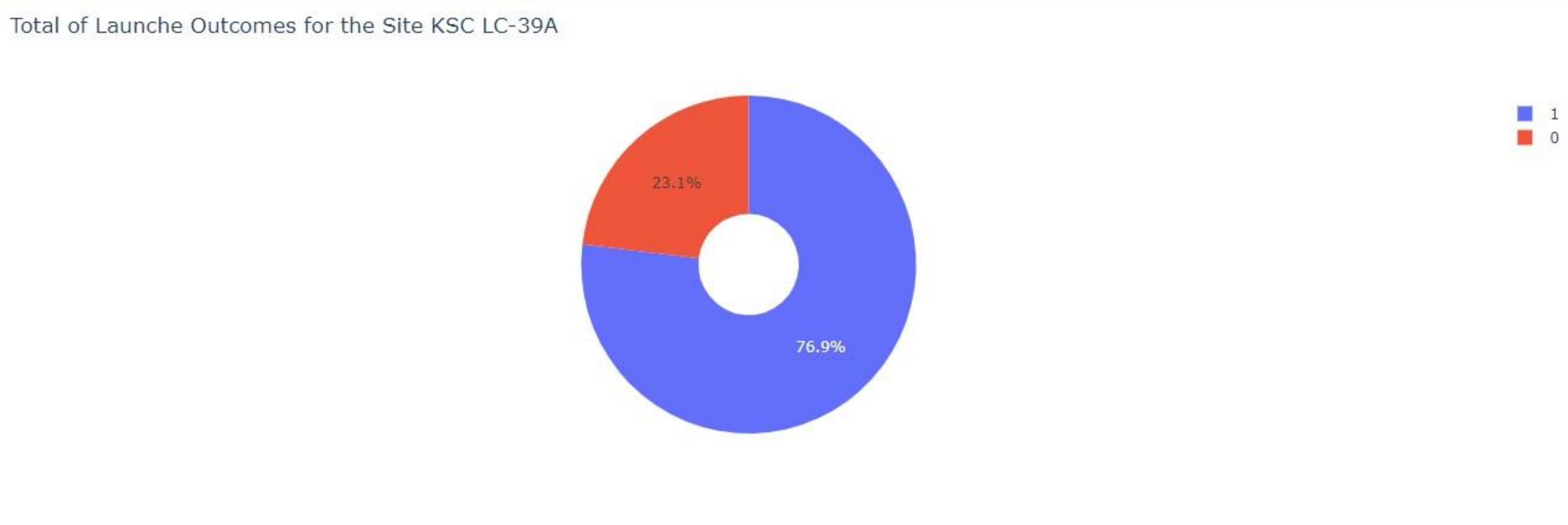
Pie chart for the launch success for all sites

- From the pie chart it can be seen that KSC LC-39A has the highest number of launch success.



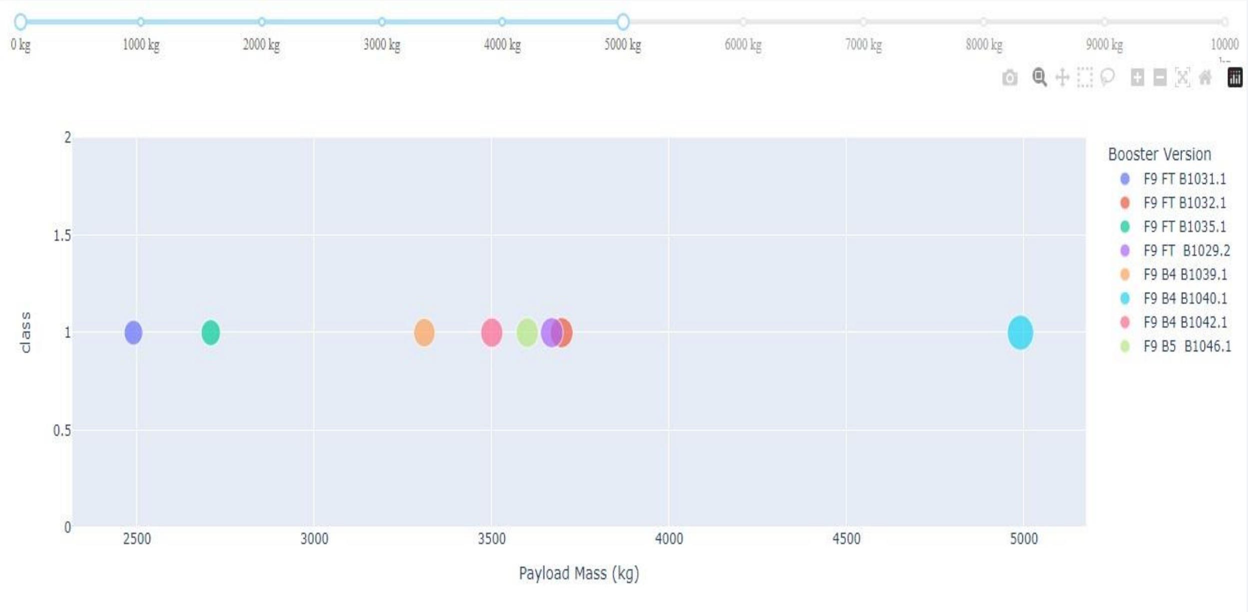
Piechart for the launch site with highest launch success ratio

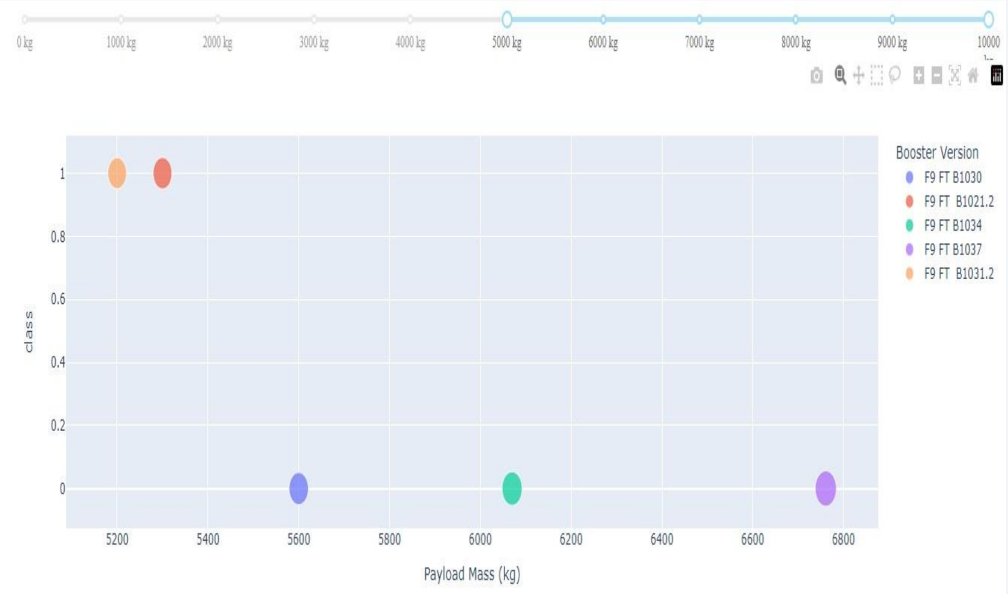
- For this site, there is a 76.9% of success rate



• Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- For Launches with a payload between 0 and 5000, the launch outcome was a success for all sites, while for payload mass higher than 5000 kg, the outcomes were successful only for booster versions F9 FT B1021.2 and B1031.2





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- While all the models have the same test accuracy, the decision tree classifier is the model with the highest test accuracy

```
In [82]:  tree_cv = GridSearchCV(tree,param_grid=parameters)
          tree_cv.fit(X_train, Y_train)

Out[82]: GridSearchCV(estimator=DecisionTreeClassifier(),
                      param_grid={'criterion': ['gini', 'entropy'],
                                  'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                                  'max_features': ['auto', 'sqrt'],
                                  'min_samples_leaf': [1, 2, 4],
                                  'min_samples_split': [2, 5, 10],
                                  'splitter': ['best', 'random']})

In [83]:  print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
          print("accuracy :",tree_cv.best_score_)

         tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2,
         'splitter': 'random'}
         accuracy : 0.8885714285714286
```

TASK 9

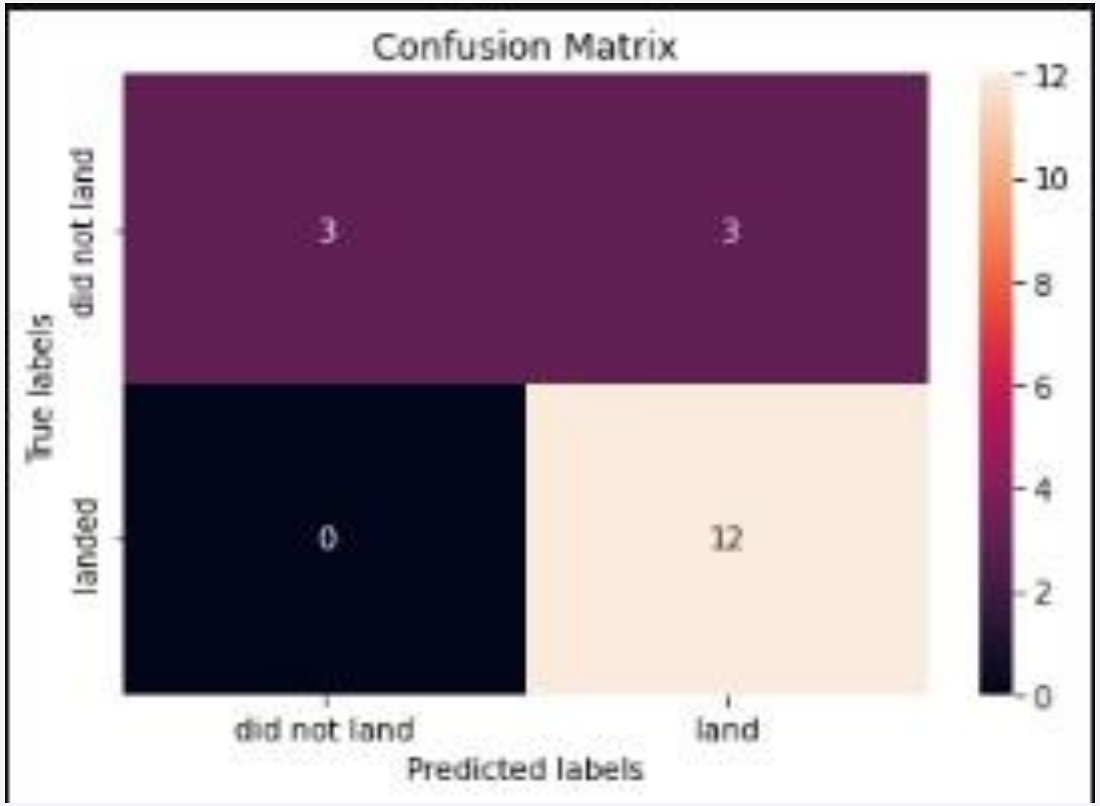
Calculate the accuracy of tree_cv on the test data using the method `score`:

```
In [84]:  scores.append(tree_cv.score(X_test,Y_test))
          tree_cv.score(X_test,Y_test)

Out[84]: 0.8333333333333334
```

Confusion Matrix

- From the confusion matrix it can be seen that the decision tree classifier can perfectly classify the data for the outcomes that landed successfully, while for the negative outcomes, there are false positives.



Conclusions

- From this project I conclude that:
  - The lower the payload, the higher the chance of success for the launch;
  - The higher the flight number at a launch site, the higher the success rate at a launch site;
  - There was no launch success for launches made before 2013;
  - The launches for the orbits ES-L1, GEO, HEO, SSO, and VLEO had the most success rate;
  - The site KSC LC-39A had the most successful launches;
  - All the models have the same out of sample accuracy, yet the Decision tree classifier is the one with the best perfomance in training.

Thank you!