IBM – Applied Data Science Capstone

Final Project – SpaceX

Outline

- Executive summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive summary

Summary of methodolody

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis - Classification

Summary of all results

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

Introduction

- Project background and context

The aim was to predict whether the first stage of Falcon 9 would land successfully. The SpaceX announces the launch of the Falcon 9 rocket on its website at a cost of 62 million dollars. Other providers cost more than 165 million dollars each, much of the savings come because SpaceX can reuse the first stage. So, if you can determine whether the first stage will land, you can determine the launch cost. This information can be used by another company that wants to compete with Space X for rocket launches.

- Problems that need answers

What influences the rocket to land successfully?

The effect of certain variables for a successful landing

What should Space X do to achieve the best results?

Methodology

- Data collection methodology:

  SpaceX Rest API

  Web Scrapping) from Wikipedia

- Perform Data wrangling (for Machine Learning)
  Transforming data for Machine Learning

- Perform Exploratory data analysis (EDA) using visualization and SQL
- Plotting : Scatter Graphs, bar graphs to show relationships between variables and to find patterns on data.

- Perform Interactive visual analytics by using Folium and Plotly Dash

- Perform Predictive analysis by using classification models

  How to build, tune, and evaluate classification models

Data collection

- The data was collected from the Space X REST API (api.spacexdata.com/v4/.) The API provides us with complete and detailed data about previous Space X releases. right. Another way to get the data is through web scraping with Falcon 9 data that can be obtained from wikipedia with BeautifulSoup.

Data collection – SpaceX API

1- Getting response from api

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

```
response = requests.get(spacex_url)
```

2- Converting response to a .json file

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

3- Applying custom funcions to clean the data

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4- Assign list to dictionary then DataFrame

```
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}
```

5- Filter dataframe and export to .csv

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
data_falcon9
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Link: [myproject/Applied Data Science Capstone.ipynb at master · marhenr/myproject (github.com)]

Data collection – web scraping

1-Getting Response from HTML

```
page = requests.get(static_url)
page.status_code
```

2-Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3-Finding tables

```
html_tables = soup.find_all('table')
```

4-Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5-Creating dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6-Appending data do keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7-Converting dictionary to dataframe and saving it to .cvs

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Link: [myproject/Applied Data Science Capstone - Week 1.ipynb at master · marhenr/myproject (github.com)]

Data wrangling

- In the database there are several cases where landing was not successful. sometimes a landing was attempted but failed due to an accident; for example, “True Ocean” means the mission result was landed successfully in a specific region of the ocean, while “False Ocean” means the mission result was landed unsuccessfully in a specific region of the ocean. “True RTLS” means the mission result landed successfully on a ground platform “False RTLS” means the mission result was landed unsuccessfully on a ground pad. “True ASDS” means that the mission result was successfully landed on a drone ship. “False ASDS” means that the mission result was unsuccessfully landed on a drone ship. The results were converted into training labels with 1 meaning successful landing and 0 meaning unsuccessful landing.

Describing the process

- Perform Exploratory Data Analysis EDA on dataset

  1- Calculate the number of launches at each site:

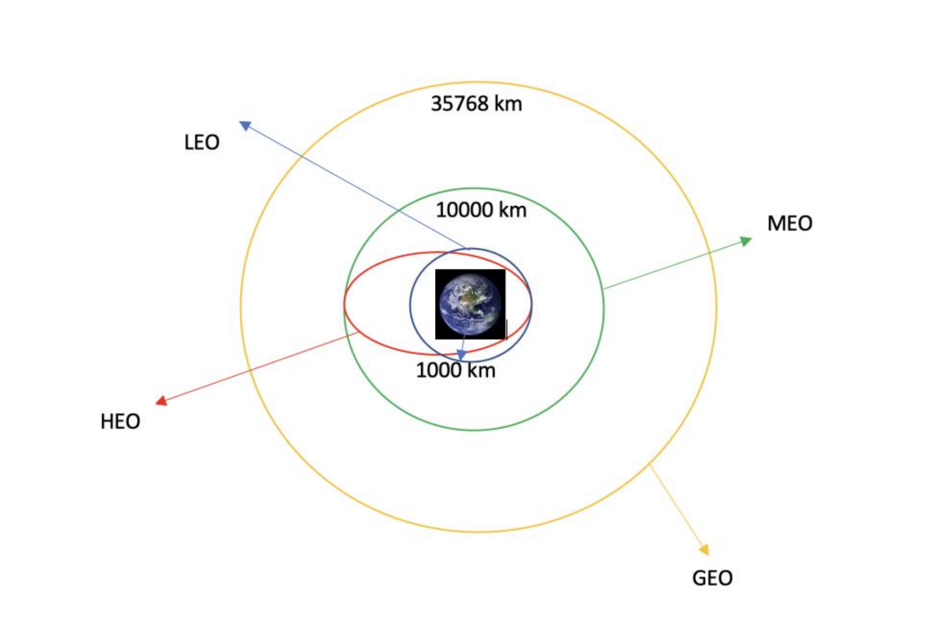
  Calculate the number and occurrence of mission outcome per orbit type

  Export dataset as .CSV

  2 - Calculate the number and occurrence of each orbit:

  Create a landing outcome label from Outcome column

  Work out success rate for every landing in dataset



Orbits used by SpaceX

Link: [myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)]

EDA with data visualization

- Scatter graphs

Flight Number VS. Payload Mass

Flight Number VS. Launch Site

Payload VS. Launch Site

Orbit VS. Flight Number

Payload VS. Orbit Type

Orbit VS. Payload Mass

Scatter graphs shows how much one variable is affected by another. This relationship is called correlation.

- Bar graphs

Mean VS. Orbit

A bar chart makes it straightforward to compare sets of data across multiple groups at a glance. On one pivot, the chart refers to categories, while on the other, it speaks to a distinct esteem. The goal is to demonstrate the connection between the two tomahawks. Bar charts can also show significant changes in data over time.

- Line graphs

Success Rate VS. Year

Line graphs are valuable because they clearly display data variables and patterns, and they can aid in making predictions about the outcomes of data that has not yet been recorded.

Link: [myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)]

EDA with SQL

Performed SQL queries to gather information about the dataset

-Displaying the names of the unique launch sites in the space mission

-Displaying 5 records where launch sites begin with the string 'KSC’

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

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

-Listing the date where the successful landing outcome in drone ship was achieved.

-Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

-Listing the total number of successful and failure mission outcomes

-Listing the names of the booster_versions which have carried the maximum payload mass.

-Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Link:myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)

Building an interactive map with Folium

To make an interactive map out of the Launch Data. We used the Latitude and Longitude Coordinates for each launch site to create a Circle Marker with the name of the launch site labeled around it.

Was used the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster() Used Haversine’s formula to calculate the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks.

Link:myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)

Building an interactive map with Folium

Example of some trends in which the Launch Site is situated in:

- 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

Link:myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)

Built an interactive dashboard with Flask and Dash

Graphs

Pie Chart showing the total launches by a certain site/all sites

-display relative proportions of multiple classes of data.

- size of the circle can be made proportional to the total quantity it represents.

Link:myproject/Applied Data Science Capstone - Data Wrangling.ipynb at master · marhenr/myproject (github.com)

Predictive analysis (Classification)

Building a model

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

Evaluating the model

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

Link:myproject/Applied Data Science Capstone - Week4.ipynb at master · marhenr/myproject (github.com)

Predictive analysis (Classification)

Evaluating the model

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

Finding the best performing model

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook

Link:myproject/Applied Data Science Capstone - Week4.ipynb at master · marhenr/myproject (github.com)

Predictive analysis (Classification)

Evaluating the model

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

Finding the best performing model

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook

Link:myproject/Applied Data Science Capstone - Week4.ipynb at master · marhenr/myproject (github.com)

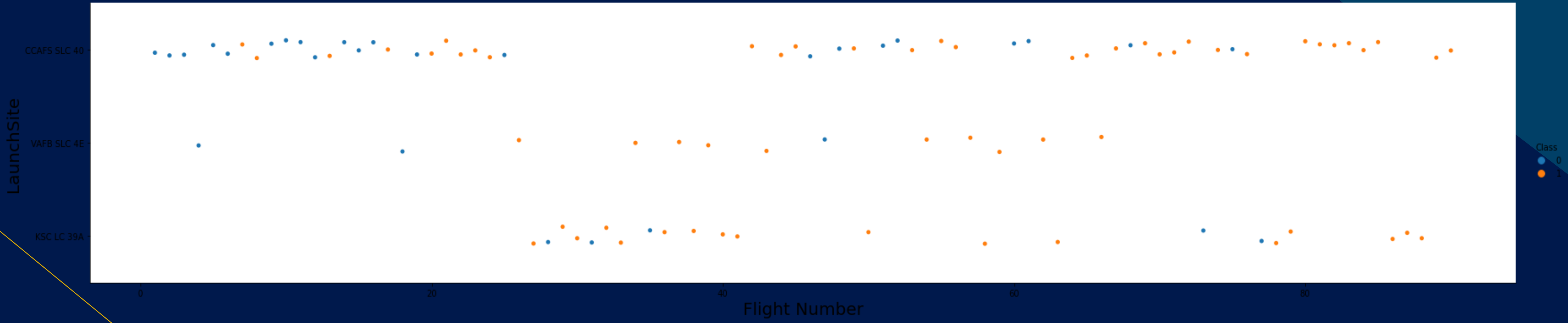
Results

Evaluating the model

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

EDA with Visualization

Launch site vs. Flight number

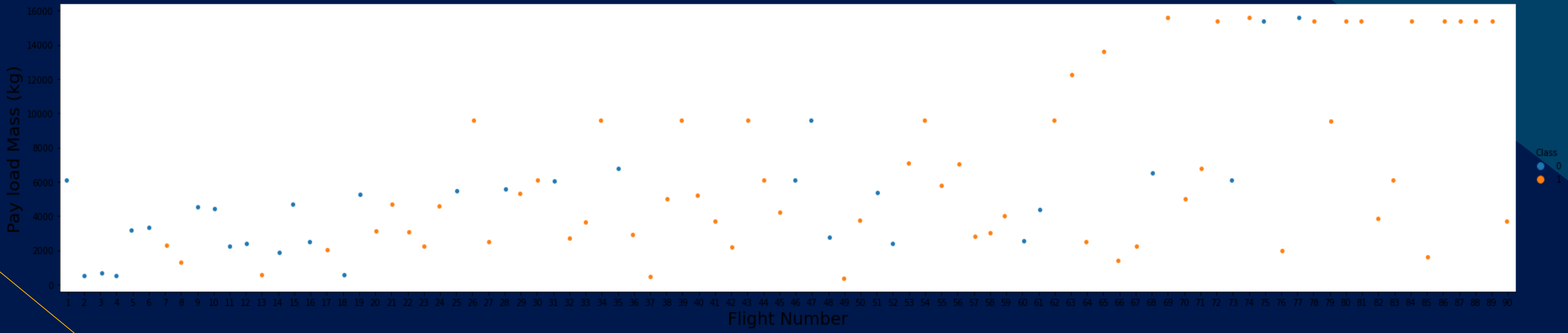


The more amount of flights at a launch site the greater the success rate at a launch site.

Link:[myproject/Applied Data Science Capstone - Week2-2.ipynb at master · marhenr/myproject (github.com)]

EDA with Visualization

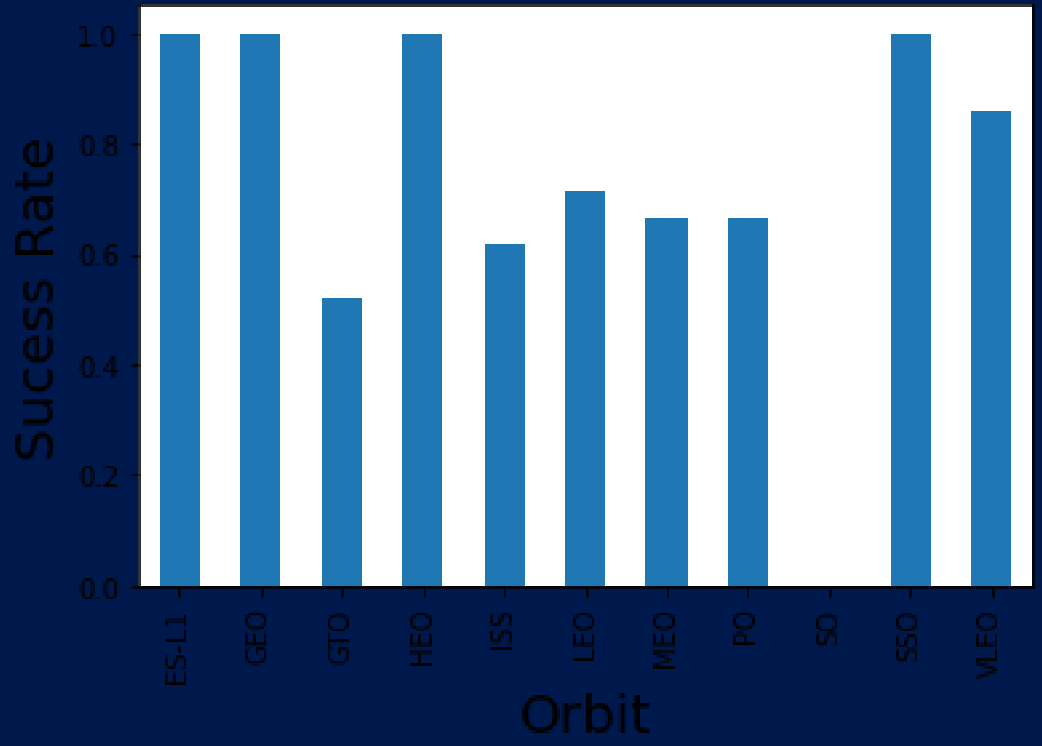
PayloadMass vs. Flight number



Link:myproject/Applied Data Science Capstone - Week2-2.ipynb at master · marhenr/myproject (github.com)

EDA with Visualization

Sucess Rate vs. Orbit Type

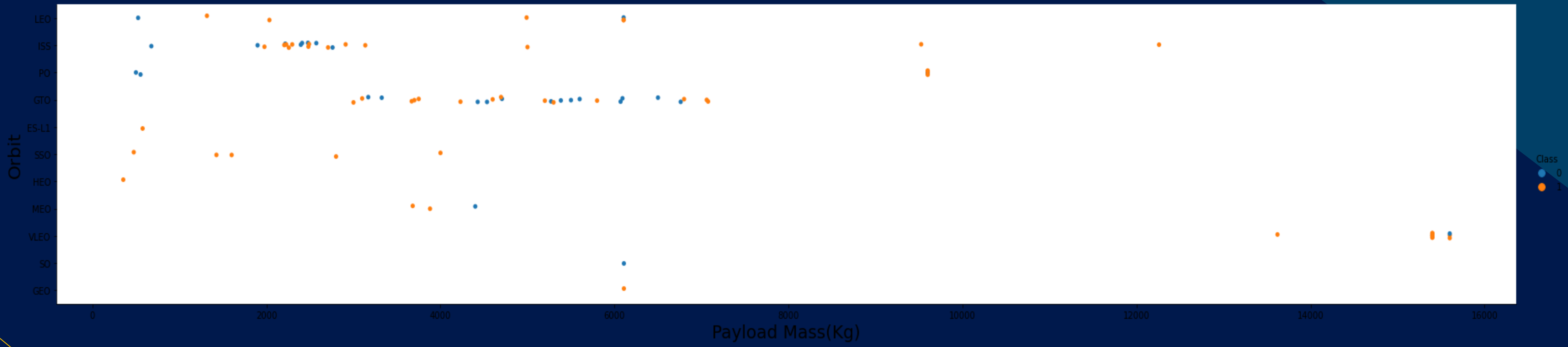


GEO, HEO, SSO and ES-L1 orbits has the best success rate

Link:[myproject/Applied Data Science Capstone - Week2-2.ipynb at master · marhenr/myproject (github.com)]

EDA with Visualization

PayloadMass vs. Orbit type

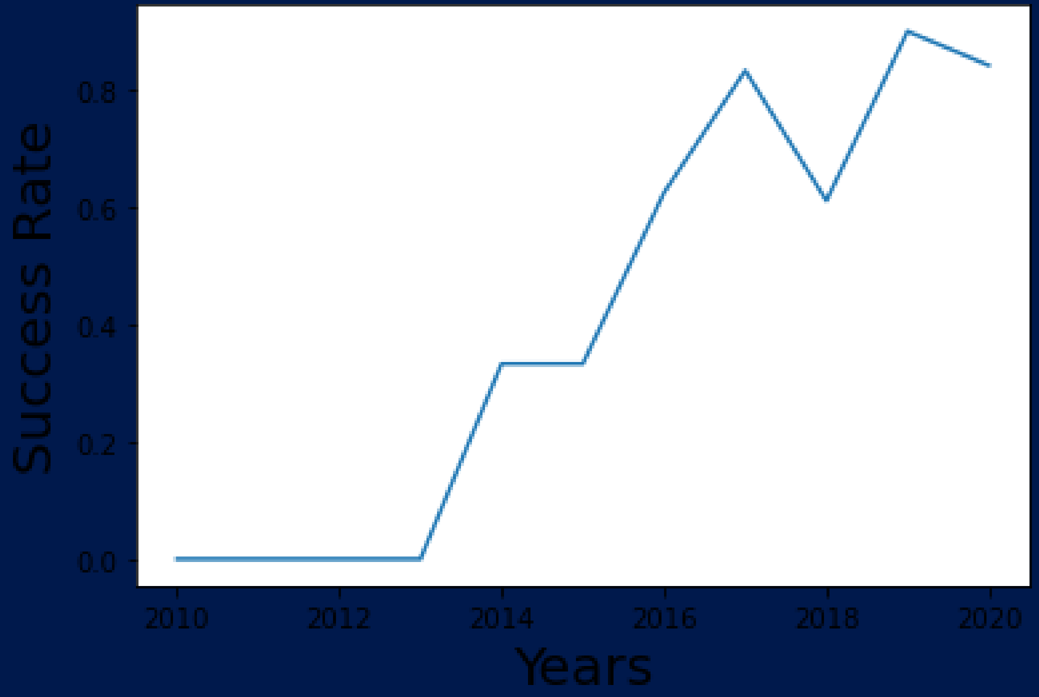


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Link:myproject/Applied Data Science Capstone - Week2-2.ipynb at master · marhenr/myproject (github.com)

EDA with Visualization

Launch sucess: yearly trend



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

Link:[myproject/Applied Data Science Capstone - Week2-2.ipynb at master · marhenr/myproject (github.com)]

EDA with SQL

Unique launch sites

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

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

Using the word Unique in the query means that it will only show Unique values in the Launch_Site column from tblSpaceX

Link:myproject/Applied Data Science Capstone - Week 2.ipynb at master · marhenr/myproject (github.com)

here you can see the names of sites that start with CCA

Launch site names begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

In [7]:
```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[7]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Here you can see the names of sites that start with “CCA”

Link:myproject/Applied Data Science Capstone - Week 2.ipynb at master · marhenr/myproject (github.com)

Total Payload Mass by Customer NASA (CRS)

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

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

payloadmass
619967

The total PayloadMass was 619967. 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).

Link:myproject/Applied Data Science Capstone - Week 2.ipynb at master · marhenr/myproject (github.com)

Average Payload Mass carried by booster version F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/b
ludb
Done.
```

payloadmass
6138

Average Payload Mass carried by booster version F9 v1.1 was 6138. Using the function AVG works out the average in the column PAYLOAD_MASS_KG_ The WHERE clause filters the dataset to only perform calculations on Booster version F9 v1.1

Link:myproject/Applied Data Science Capstone - Week 2.ipynb at master · marhenr/myproject (github.com)

The the date when the first succesful landing outcome in ground pad was acheived.

```
%sql select min(DATE) from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

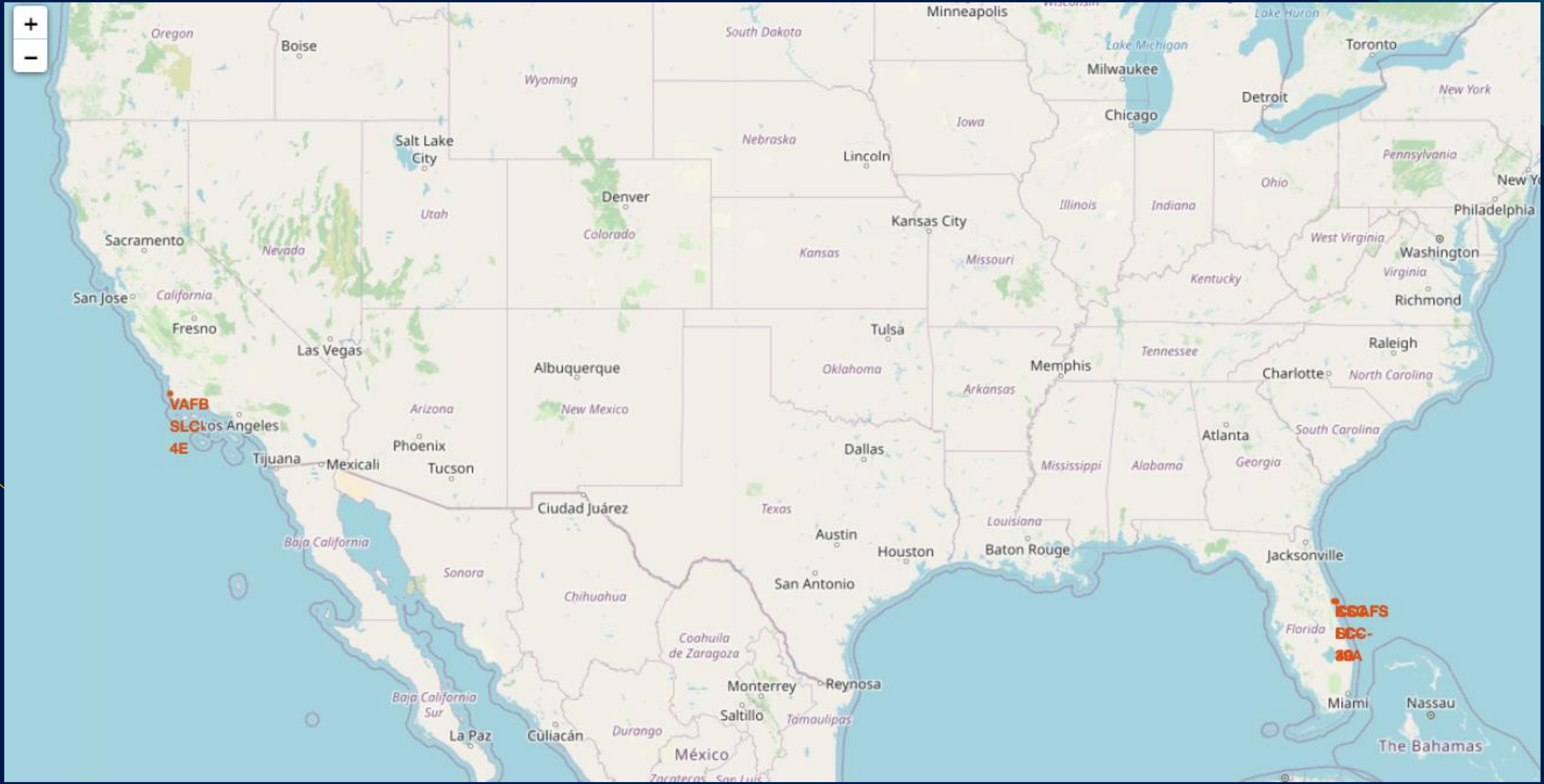
1
2010-06-04

2010-06-04 was the date where the successful landing outcome in drone ship was achieved. Using the function MIN works out the minimum date in the column Date.

Link:myproject/Applied Data Science Capstone - Week 2.ipynb at master · marhenr/myproject (github.com)

Interactive map with Folium

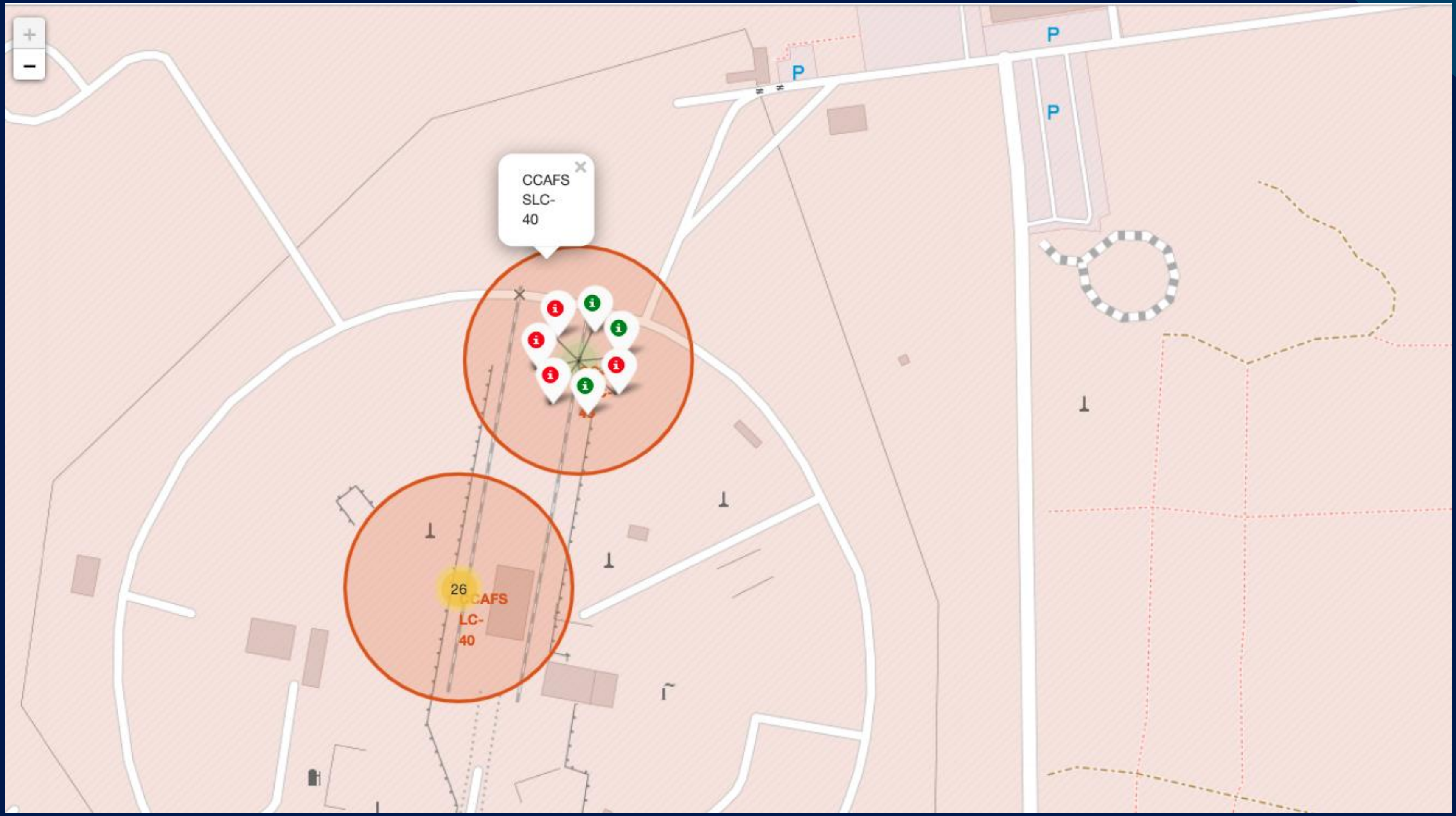
All launch sites global map markers



The SpaceX launch sites are in the United States of America coasts. Florida and California

Link:myproject/Applied Data Science Capstone - Week3.ipynb at master · marhenr/myproject (github.com)

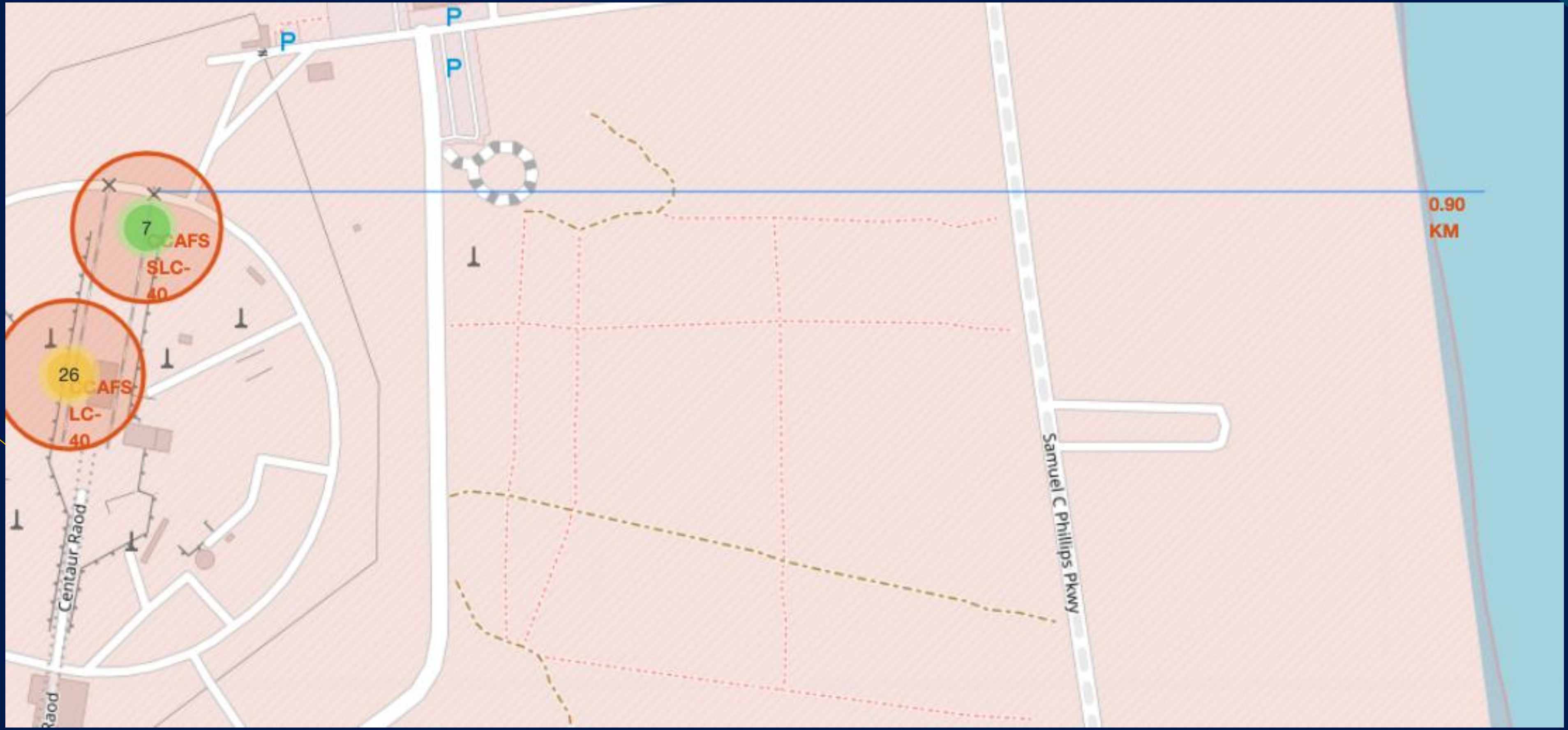
Colour Labelled Markers



Green Marker shows successful Launches and Red Marker shows Failures

Link:myproject/Applied Data Science Capstone - Week3.ipynb at master · marhenr/myproject (github.com)

Distance from coastline

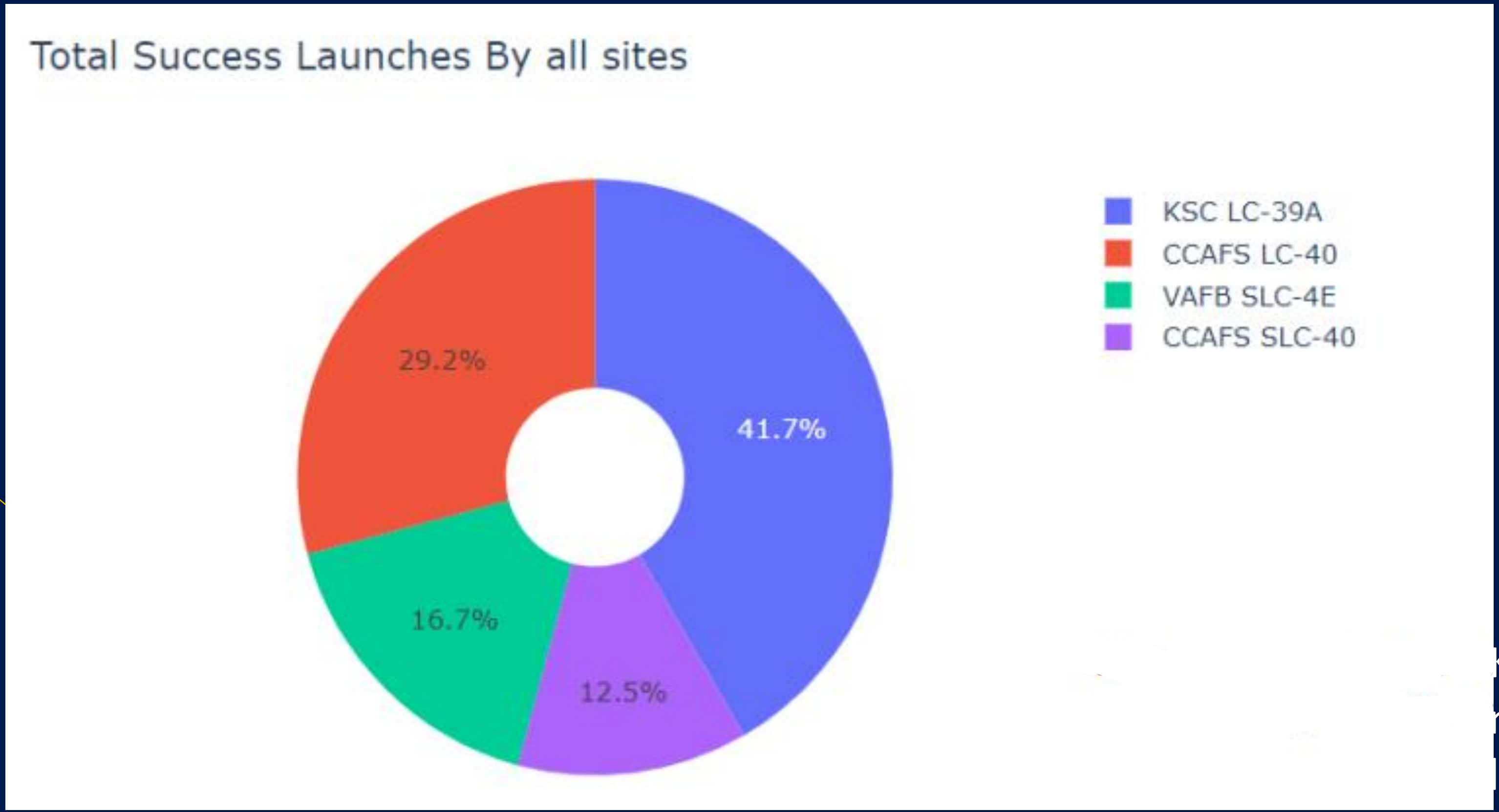


We can see that the distance is 0.90 KM from coastiline

Link:[myproject/Applied Data Science Capstone - Week3.ipynb at master · marhenr/myproject (github.com)]

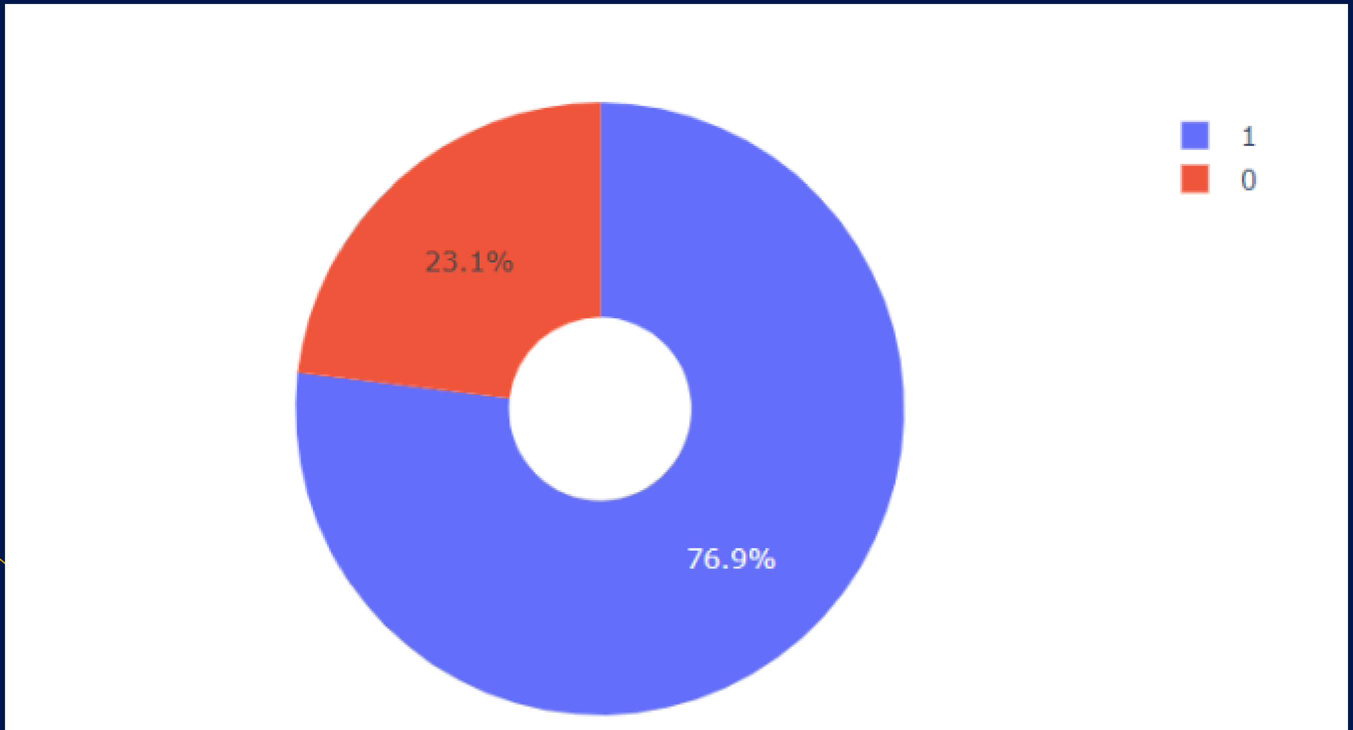
Dashboard with Plotly Dash

Dashboard – Pie chart – success percentage by each launch



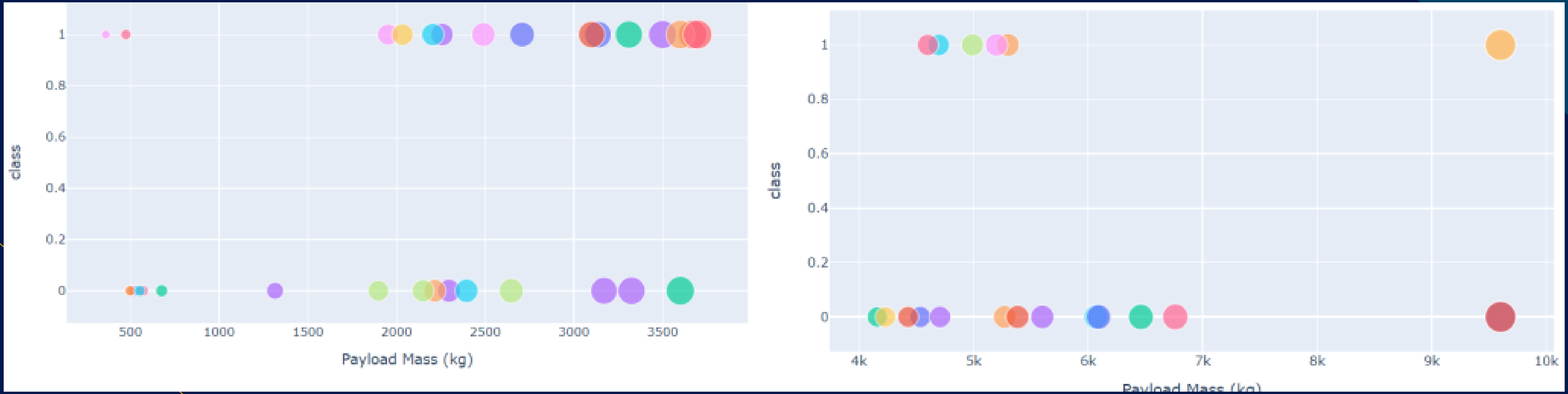
KSC LC-39A had the most successful launches from all the sites

Dashboard – Pie chart – launch site with highest launch success ratio



KSC LC-39A achieved 76.9% success rate and 23.1% failure rate

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



Success rates for low weighted payloads is higher than the heavy weighted payload

Predictive analysis - Classification

Classification Accuracy using training data

```
scores = [lr_score,svm_score,tree_score,knn_score]
print(scores)
print(scores.index(max(scores)))
```

```
[0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334]
0
```

Interestingly, the accuracy was the same for all models.

Link: myproject/Applied Data Science Capstone - Week4.ipynb at master · marhenr/myproject (github.com)

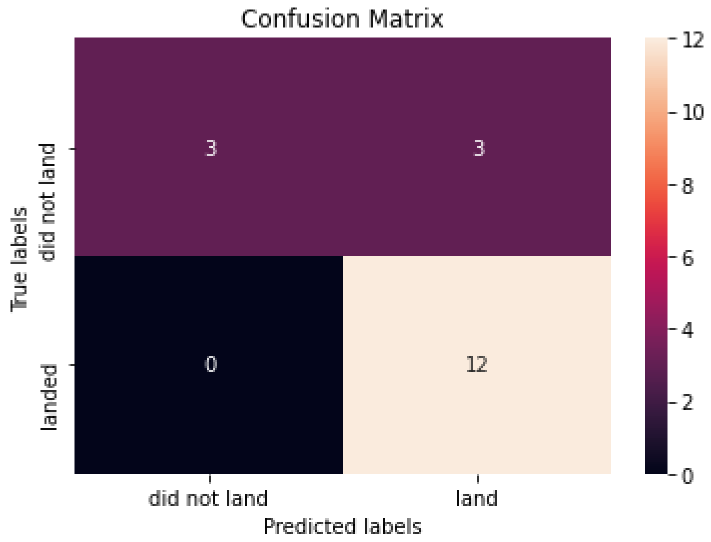
Confusion matrix

```
lr_score=logreg_cv.score(X_test,Y_test)
lr_score
```

```
0.8333333333333334
```

Lets look at the confusion matrix:

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



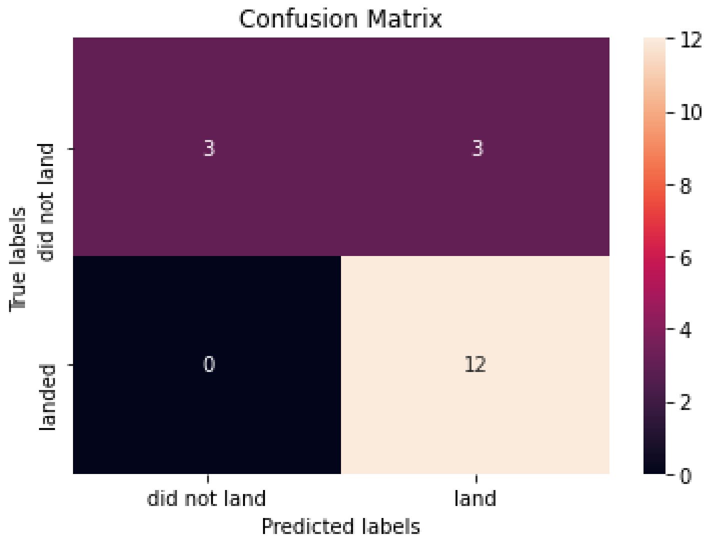
Linear regression

```
svm_score = svm_cv.score(X_test,Y_test)
svm_score
```

```
0.8333333333333334
```

We can plot the confusion matrix

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Vector Machine

Link: myproject/Applied Data Science Capstone - Week4.ipynb at master · marhenr/myproject (github.com)

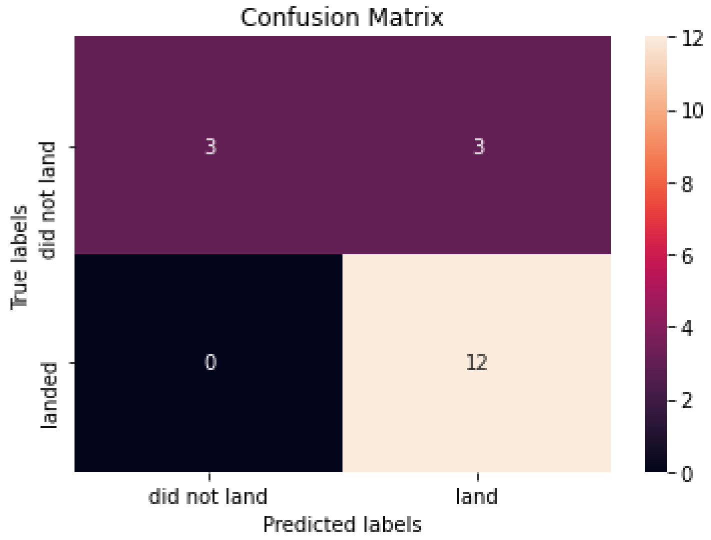
Confusion matrix

```
tree_score = tree_cv.score(X_test,Y_test)
tree_score
```

0.8333333333333334

We can plot the confusion matrix

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



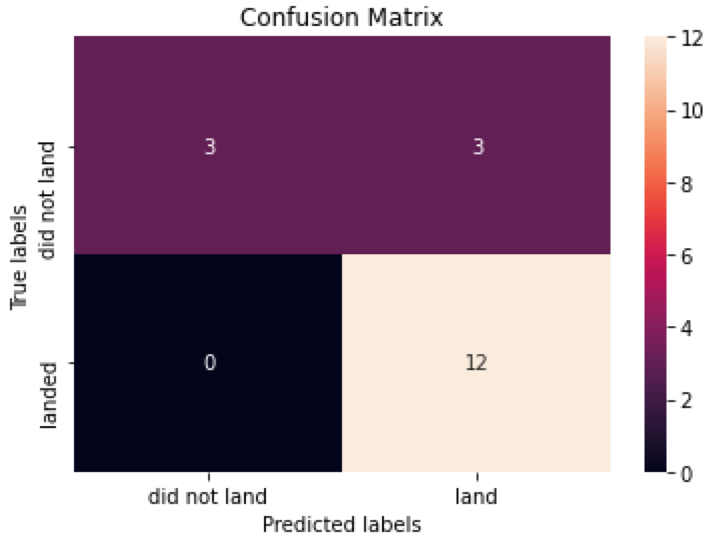
Decision Tree

```
knn_score = knn_cv.score(X_test,Y_test)
knn_score
```

0.8333333333333334

We can plot the confusion matrix

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Knn model

Conclusion

- For this dataset, all machine learning models analyzed had the same accuracy.
- Payloads with a low weight perform better than payloads with a higher weight.
- The success rate of SpaceX launches is proportional to the number of years it takes them to perfect the launch.
- KSC LC-39A from all the sites, had the most successful launching
- Orbit GEO, HEO, SSO, ES-L1 has the highest rate of success

Appendix

In this project, Google Colab was also used to help with the tasks.

Thank you.

Marcelo Henrique

[marhenr/myproject at master (github.com)]