IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Arunima Dey>
<November 4, 2023>

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Executive Summary

- Summary of methodologies
  - Collecting Space X launch data through API
  - Webscraping Falcon 9 launch data
  - Wrangling the data by dealing with null values and using sampling data
  - EDA with SQL
  - EDA with Data Visualization
  - Visual Analytics with Folium
  - Prediction by using Machine Learning

- Summary of the results
- - Screenshots of SQL queries with results
- - Prediction of results
- - Screenshots of Proximity Analysis

Introduction

- Project background and context
- The project deals with the advertisement of SpaceX for the Falcon 9 rocket launch on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. So instead of using rocket science, we will use machine learning to determine if the first stage is successful and it SpaceX will reuse it.
- Problems you want to find answers
- - What type of conditions are important to keep in mind for a successful landing program.
- - Which are the important aspects for a successful landing?

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

Collected the Space X launch data through API

Perform data wrangling

Used one hot encoding to handle null values and used sample data

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

The data was collected as follows:

- Collected data by using SpaceX API.

- Used the .json() function to view the launch data and the .json_normalize() function to convert json to dataframe.

- Webscraped Falcon 9 records by using python BeautifulSoup.

- Wrangling the data by dealing with null values and using sampling data.

Data Collection – SpaceX API

Used requests.get() to collect the data for SpaceX, cleaned data to include only Falcon 9 and used data wrangling to deal with null values

The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/jupyter_labs_spacex_data_collection_api.ipynb

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
response = requests.get(static_json_url)
response.json()
data = pd.json_normalize(response.json())
```

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

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

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

```
# Calculate the mean value of PayloadMass column
Mean = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean)
```

Data Collection - Scraping

- Used web scrapping to scrap records of falcon 9 launch with BeautifulSoup

- The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/jupyter_labs_webscraping_(1).ipynb

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

```
[10] # use requests.get() method with the provided static_url
     # assign the response to a object
     dataset  = requests.get(static_url).text
```

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

```
[11] # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     soup = BeautifulSoup(dataset, 'html5lib')
```

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

```
# Use soup.title attribute
print(soup.title)
```

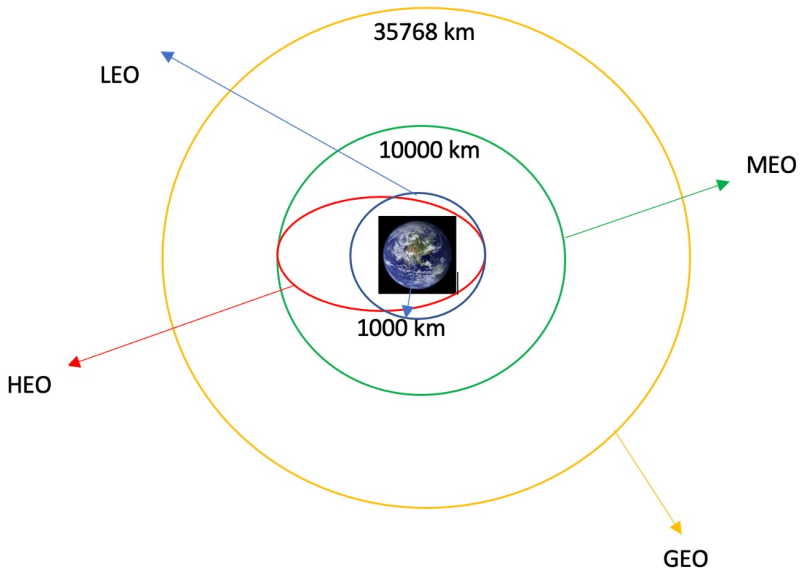
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
[14] # 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.

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

Data Wrangling



- Calculated the number of launches at each site and the number and occurrence if each orbit.
- Created a landing outcome label from Outcome column

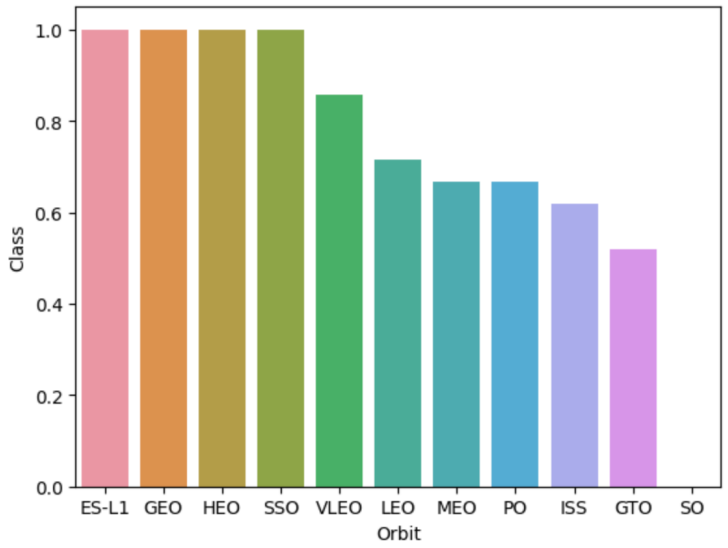
- The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/labs_jupyter_spacex_Data_wrangling.ipynb

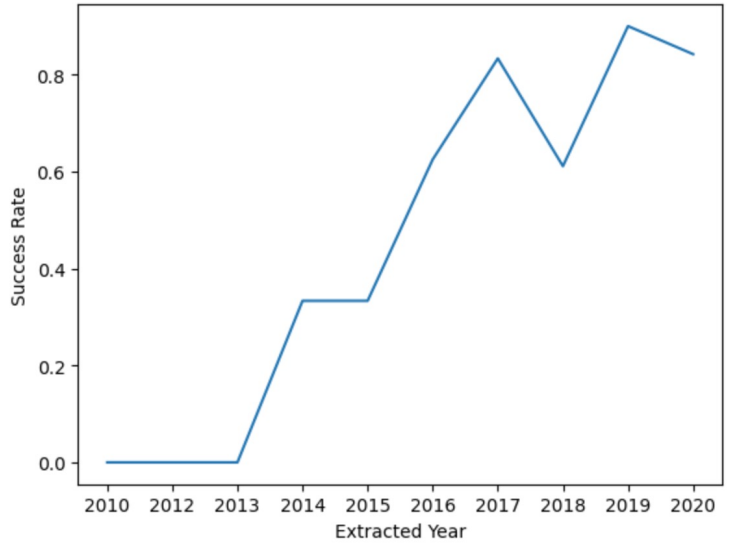
EDA with Data Visualization

- Through the graphs it cane be inferred that the success rate tends to increase right after the year 2013 after remaining constant for two years. It kept increasing up until 2017 and after that it comes down a bit in the year 2018. It increases again in the year 2019 and reaches the highest value of success rate.

- Moreover, Through the plot it can be inferred that the Orbit Types ES-L1, GEO, HEO and SSO have experienced the most success rate. The Orbit Types VLEO, LEO, MEO, PO, ISS and GTO have experienced a gradual decrease in the success rate in the order mention. Lastly the Orbit Type SO does not have any success rate.

- The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/jupyter_labs_eda_dataviz_(1).ipynb





EDA with SQL

- The SpaceX dataset was loaded and then queries were performed on it.
- The queries used to get a better understanding of the dataset:

  - Displayed the names of the unique launch sites in the space mission.

  - Listed the date when the first successful landing took place.

  - Listed the total number of successful and failure mission outcomes.

  - Calculated total payload mass by boosters launched by NASA(CRS).

  - Listed the names of the booster versions that carried maximum payload mass.

- The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/jupyter-labs-eda-sql-coursera_sqllite%20(3).ipynb

Build an Interactive Map with Folium

- Marked all the launch sites and it can be inferred that the launch sites are located in the United States of America in North America in the states California and Florida.
- In order to interpret the success and failure of the launches, it can be observed that the red markers show launches that failed and the green markers show launches that succeeded.
- Calculated the distances of the launch sites from cities, railways, highways and coastline and answered few questions regarding the proximity of these features with the launch sites.
- The link: https://github.com/yedra25/FINAL-PROJECT/blob/e1c7eebb2f90cc8d5887ac37aecc35d693ff529f/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Predictive Analysis (Classification)

- In order to load the data, used numpy and divided the data into training and testing to be able to analyze better.
- Calculated the accuracy of the test data using the score method and acquired a confusion matrix which helps in getting a better understanding of false positives.
- The link: https://github.com/yedra25/FINAL-PROJECT/blob/62f1acf694872ea3ebef2d176343d694f42d50ca/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(3).ipynb

Results

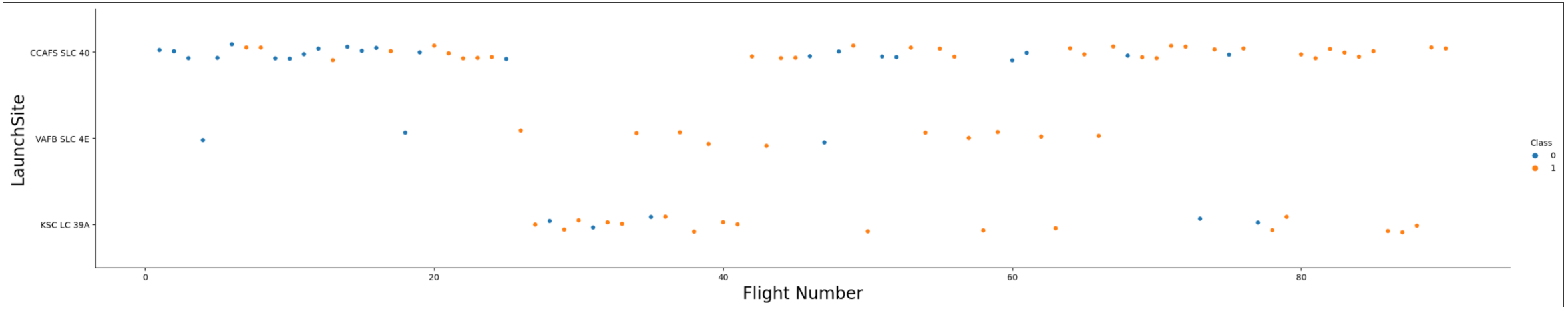
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

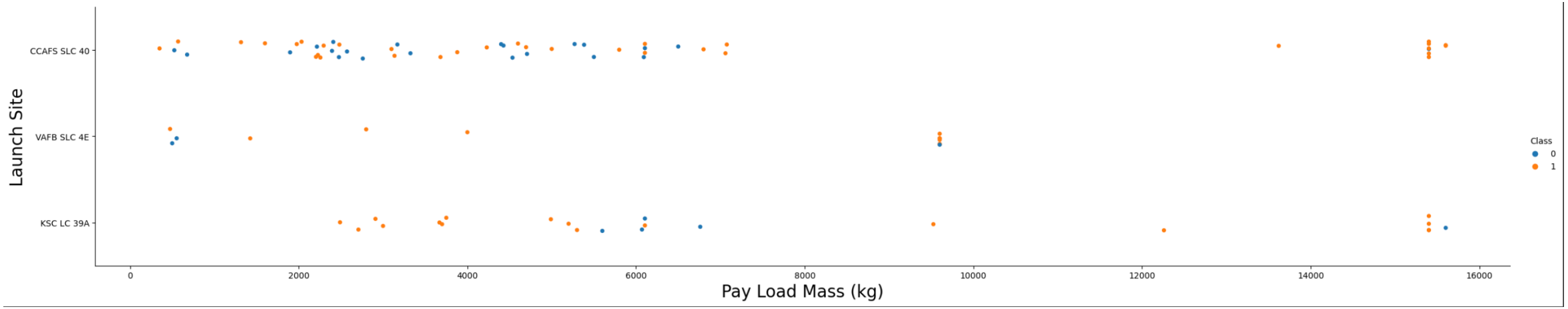
Flight Number vs. Launch Site

- Through the pot it can be inferred that with the increase in the number of flights, the success rate of the launch sites also increase.



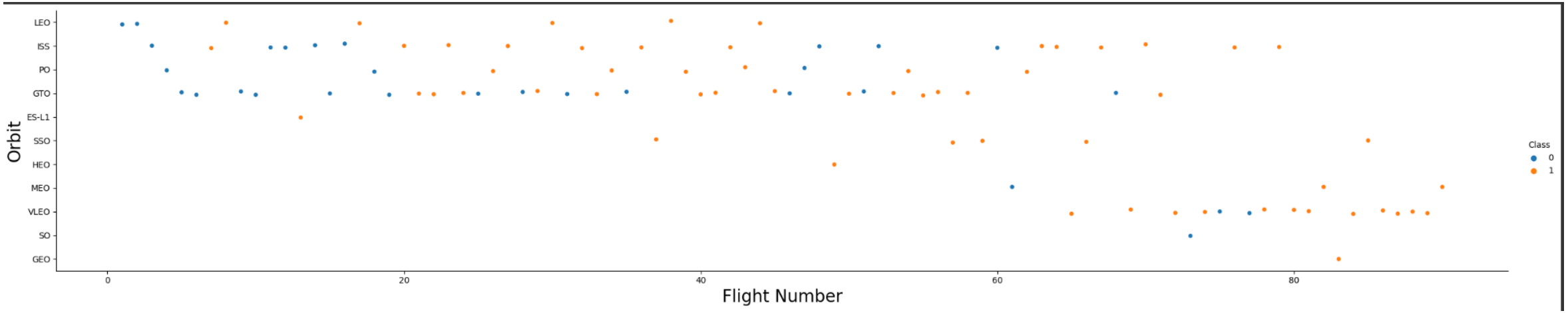
Payload vs. Launch Site

- Through the plot it can be inferred that with decrease in the Pay Load Mass, there is an increase in the success rate of the Launch Sites.



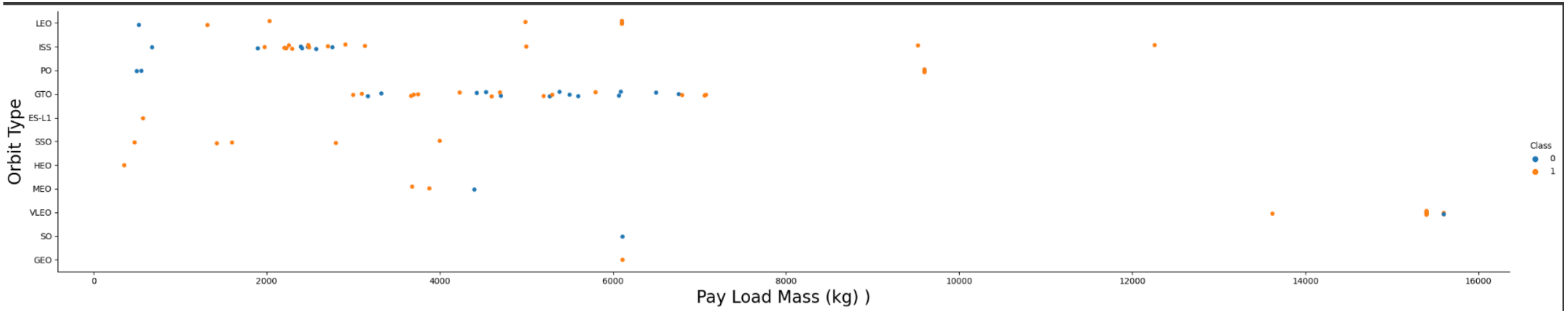
Flight Number vs. Orbit Type

- Through the plot it can be inferred, that with the decrease in the number of flights, the success rate of the the Orbit Types LEO, ISS, PO, GTO increases.
- On the other hand, with the increase inn the number of flight the success rate of the Orbit Types VLEO, SO and GEO decreases.



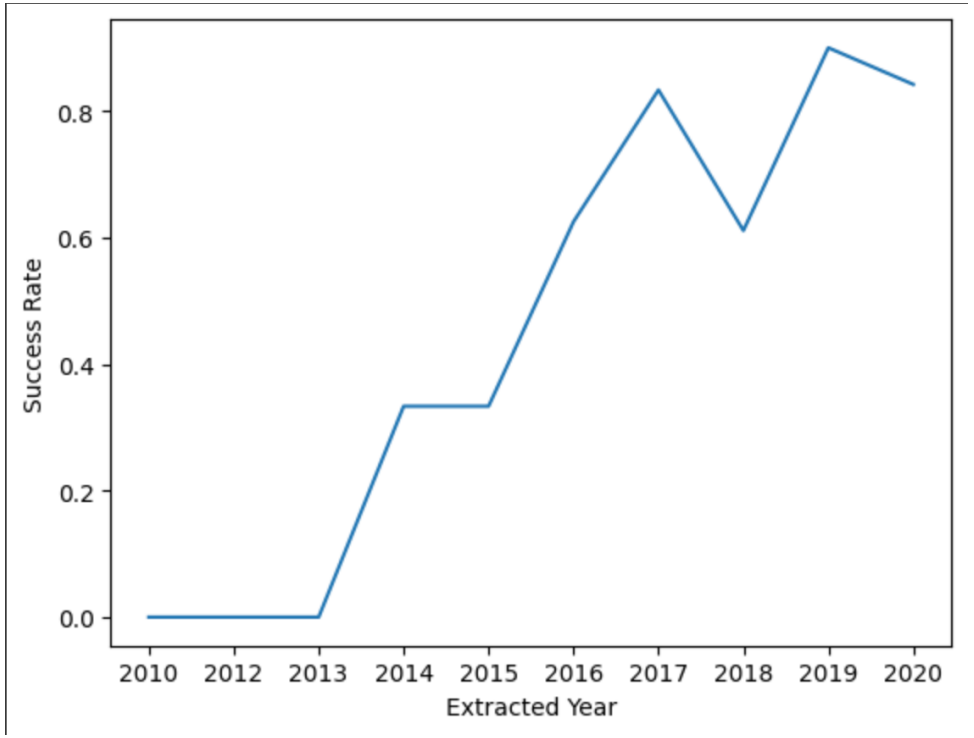
Payload vs. Orbit Type

- Through the plot it can be inferred, that lesser the Pay Load Mass(kg) the more is the success rate of the Orbit Types specifically LEO, ISS, PO and GTO.



Launch Success Yearly Trend

- Through the plot it can be observed that the success rate tends to increase right after the year 2013 after remaining constant for two years.
- It kept increasing up until 2017 and after that it comes down a bit in the year 2018.
- It increases again in the year 2019 and reaches the highest value of success rate.



All Launch Site Names

- By using the keyword DISTINT in the SELECT stamen the unique launch sites in the space mission were obtained.

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

[10]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

[10]:

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

Launch Site Names Begin with 'CCA'

- By using LIKE operator a condition was put using the WHERE condition where the launch sites should start with CCA.

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

[12]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

[12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcom
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute
12/8/2010	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
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attemp
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attemp
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attemp

Total Payload Mass

By using the SUM function, the total payload mass was acquired.

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

```
[20]: %sql SELECT sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTBL;
```

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

[20]:

total_payload_mass
619967

Average Payload Mass by F9 v1.1

- By using the average function, the average payload mass was obtained where the Booster Version was F9 v1.1

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

```
%sql SELECT avg(PAYLOAD_MASS__KG_) as avg_payload_mass FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

avg_payload_mass
2928.4

First Successful Ground Landing Date

By using the MIN function, the first date was acquired when the ground landing was successful.

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

Hint:Use min function

```
%sql SELECT min(DATE) as firs_succesful_landing_date from SPACEXTBL;
```

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

firs_succesful_landing_date
1/10/2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- By using a condition with WHERE function, acquired the Drone Ships that had a successful landing with a payload mass between 4000 and 6000.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

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

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

Total Number of Successful and Failure Mission Outcomes

By using the COUNT function acquired the number of mission outcomes.

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS number FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

By using a subquery in the WHERE condition, acquired the maximum payload mass

.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

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

- By using substr and WHERE condition, acquired the list of records including booster version, month names, launch site that experienced landing failure as an outcome in the year 2015.

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date,4,2) AS month, Date, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE\
[Landing_Outcome] = 'Failure (drone ship)' AND substr(Date,7,4)='2015';
```

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

Notebook is in Command mode

month	Date	Booster_Version	Launch_Site
04	14/04/2015	F9 v1.1 B1015	CCAFS LC-40

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

- By using COUNT, GROUP BY , ORDER BY and WHERE condition, acquired the rank of the landing outcomes between 2010-06-04 and 2017-03-20.

Task 10

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.

```
%sql SELECT [Landing_Outcome], count(*) as count_outcome FROM SPACEXTBL WHERE DATE between '04-06-2010' and '20-03-2017'\
group by [Landing_Outcome] order by count_outcome DESC;
```

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

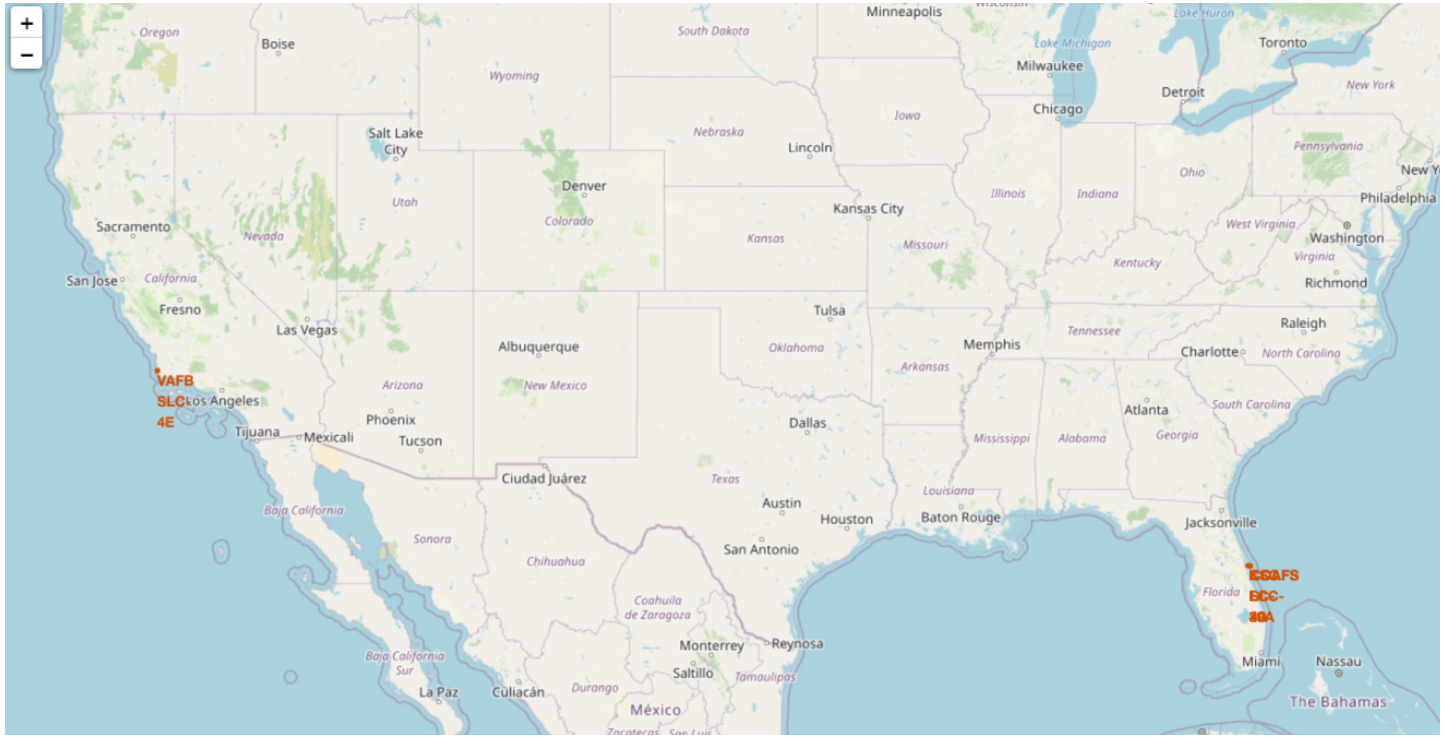
Landing_Outcome	count_outcome
Success	15
No attempt	6
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

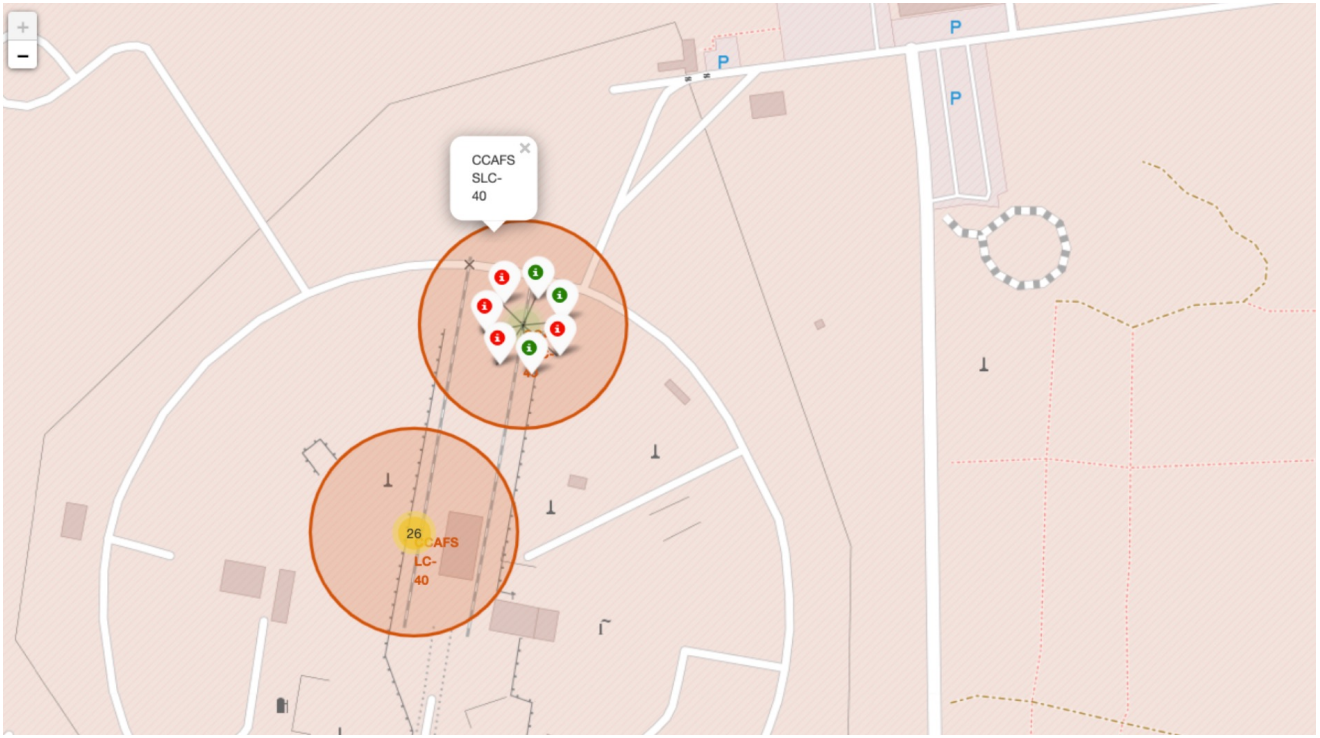
Launch Sites in United States of America

- Through the image, it can be observed that the Space X launch sites are located in the United States of America in North America in the states California and Florida.



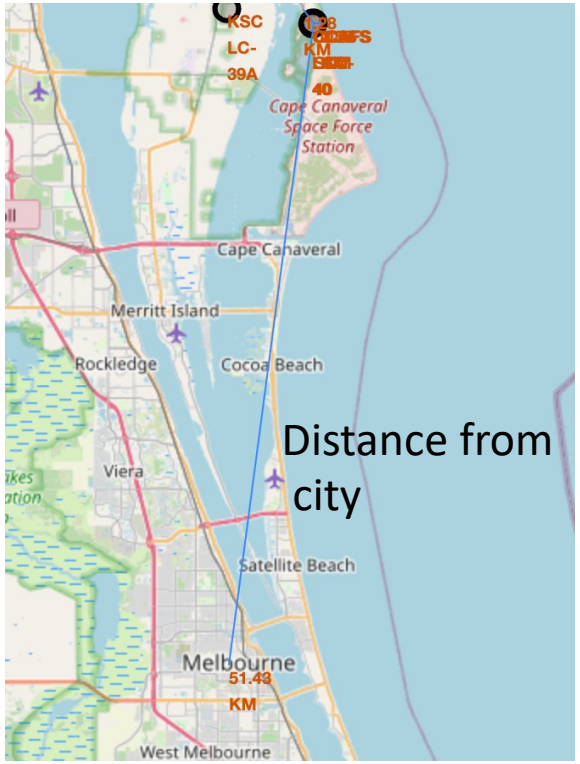
Launch sites outcomes using colored markers

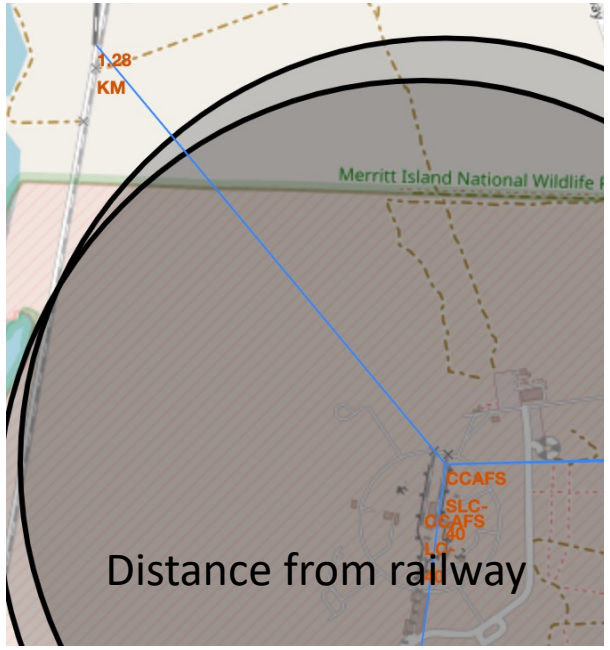
- According to the image it can be observed that the red markers show launches that failed and the green markers show launches that succeeded.

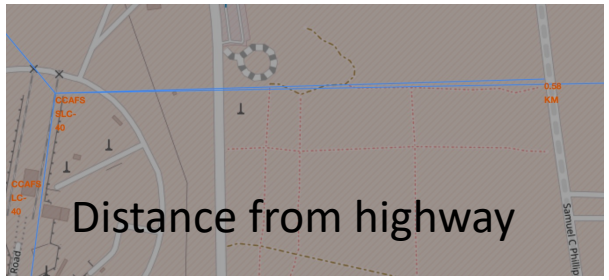


Launch Sites Distances from different landmarks/features

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







Section 5

Predictive Analysis (Classification)

Classification Accuracy

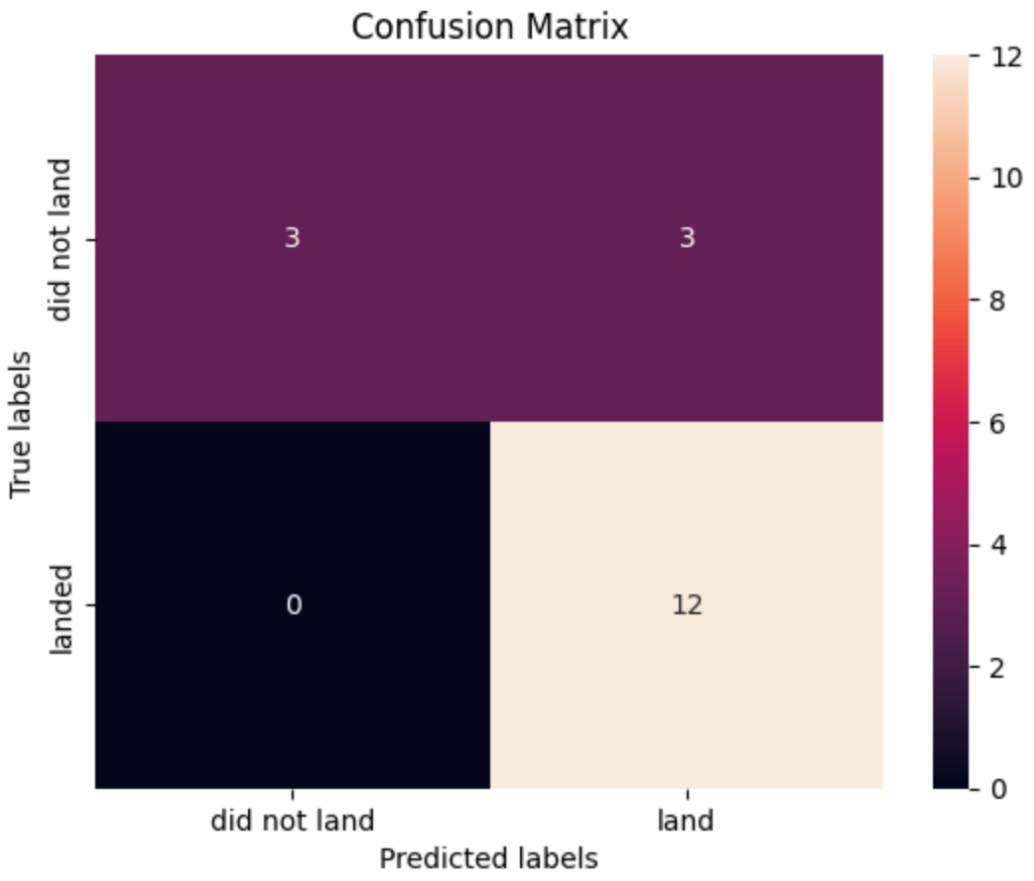
Through the following code the model with the highest classification accuracy can be acquired:

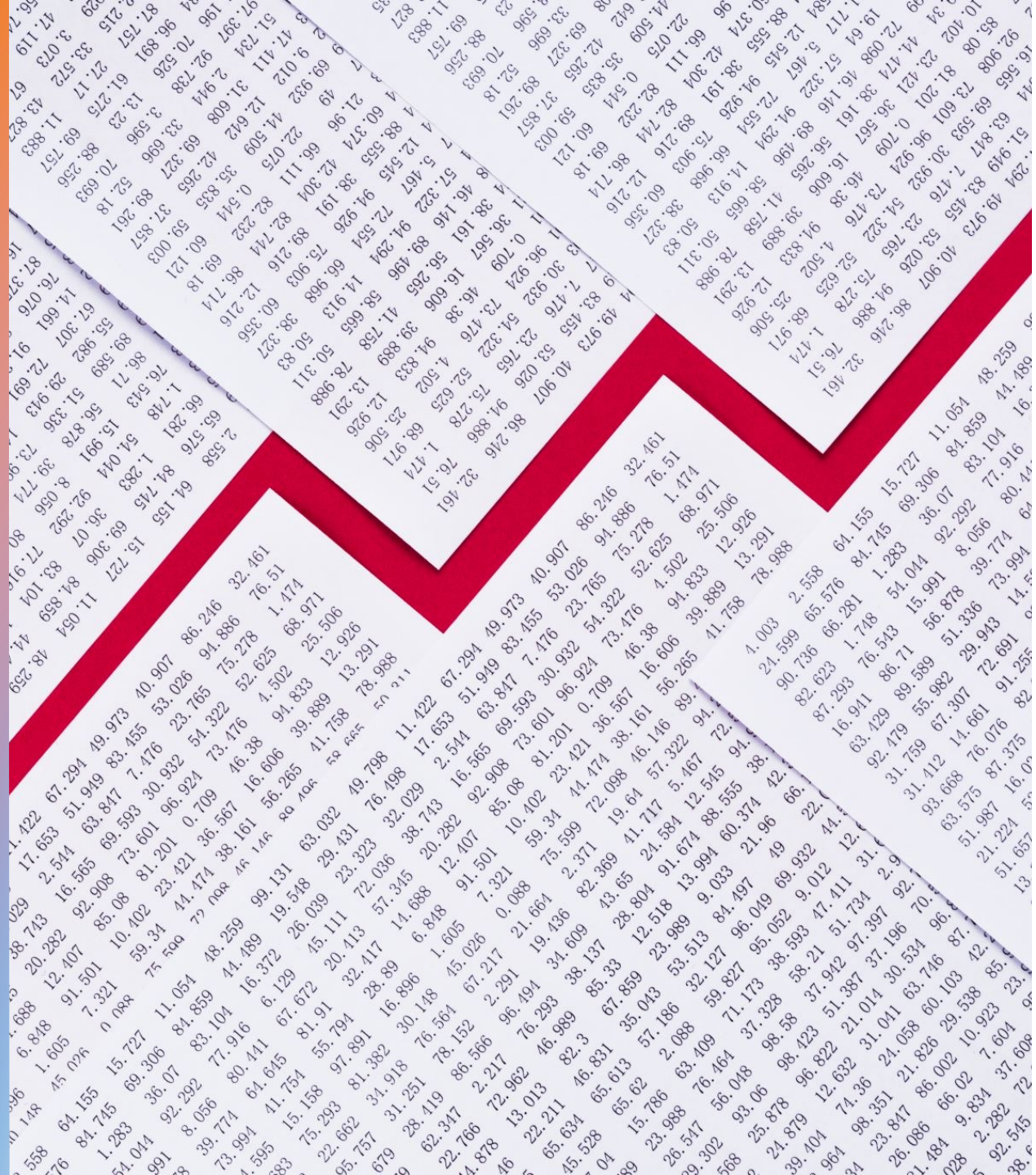
```
models = {'SupportVector': svm_cv.best_score_,
              'KNeighbors':knn_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,}

alg = max(models, key=models.get)
print('The model with highest accuracy is', alg)
if alg == 'SupportVector':
    print('Parameters :', svm_cv.best_params_)
if alg == 'KNeighbors':
    print('Parameters :', knn_cv.best_params_)
if alg == 'LogisticRegression':
    print('Parameters :', logreg_cv.best_params_)
if alg == 'DecisionTree':
    print('Parameters :', tree_cv.best_params_)
```

Confusion Matrix

The confusion matrix basically helps in understanding the disparity between different types of classes. Moreover, it deals with landing that are unsuccessful and have been marked successful.





Conclusions

- It can be concluded that:
- The success of the launches tends to increase right after the year 2013 after remaining constant for two years. It kept increasing up until 2017 and after that it comes down a bit in the year 2018. It increases again in the year 2019 and reaches the highest value of success rate.
- The Orbit Types ES-L1, GEO, HEO and SSO have experienced the most success rate. The Orbit Types VLEO, LEO, MEO, PO, ISS and GTO have experienced a gradual decrease in the success rate in the order mention. Lastly the Orbit Type SO does not have any success rate.
- The decrease in the number of flights, the success rate of the the Orbit Types LEO, ISS, PO, GTO increases. On the other hand, with the increase inn the number of flight the success rate of the Orbit Types VLEO, SO and GEO decreases.

Thank you!