IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Indre White
1/6/2024

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection via API, web scraping
  - Exploratory Data Analysis (EDA) including data wrangling, data visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis (Classification)
- **Summary of all results**
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive analysis results

Introduction

Project background and context

- SpaceX has been the leader in the space industry providing Falcon 9 launches with a cost of 62 million dollars (when the first stage of rockets can be reused.
- Other providers that were not able to reuse the first stage cost up to 165 million per launch.
- The purpose of this project is to predict if the Falcon 9 first stage will successfully land. By determining if the stage will land, we can determine the cost of a launch.
- In order to determine the price of the launch we can use public data and machine learning models.

Questions to be answered

- How variables like payload mass, launch site, number of flights, orbits affect first stage landing success.
- Best places and predictive model for successful launches.
- The rate of successful landings over years (Does it increase?).

Section 1

Methodology

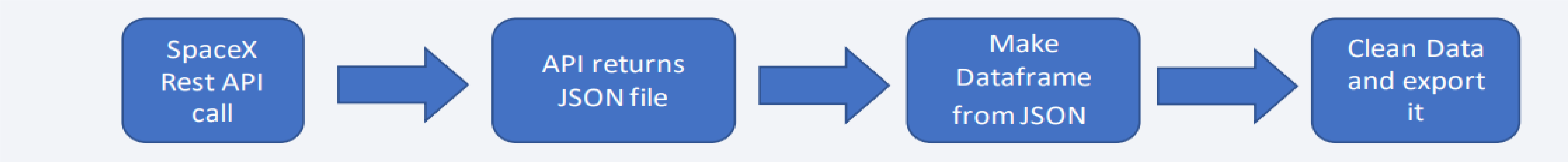
Methodology

Executive Summary

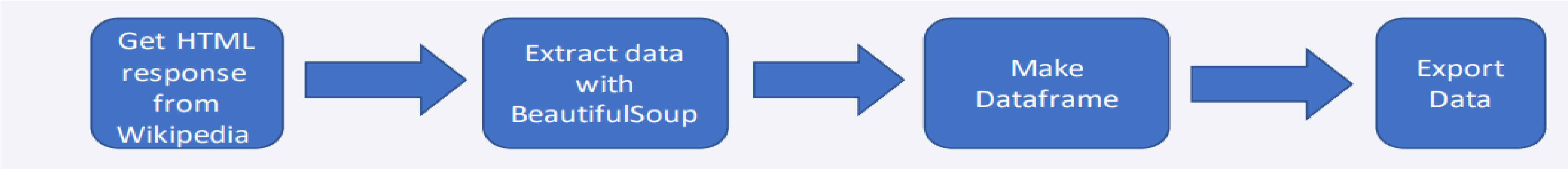
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Data was filtered, missing values handled, one hot encoding was used
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating of classification models

Data Collection

- **Datasets are collected from Rest SpaceX API (rocket launch date) and web scrapping Wikipedia (get info from API - URL is api.spacexdata.com/v4/)**



URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

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

3. Transform data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

[link to code]

4. Create dictionary with data

```
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. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

1. Getting Response from HTML

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

2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables

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

4. Get column names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create 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. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

See notebook for the rest of code

7. Create dataframe from dictionary

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

8. Export to file

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

[link to code]

Data Wrangling

- **In the dataset, there are several cases where the booster did not land successfully. Could be a failure because of accident. Examples below:**
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- **We need to convert the outcomes to training labels where 1 means the booster landed successfully, and 0 means the mission failed.**

1. Calculate launches number for each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Calculate the number and occurence of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from Outcome column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

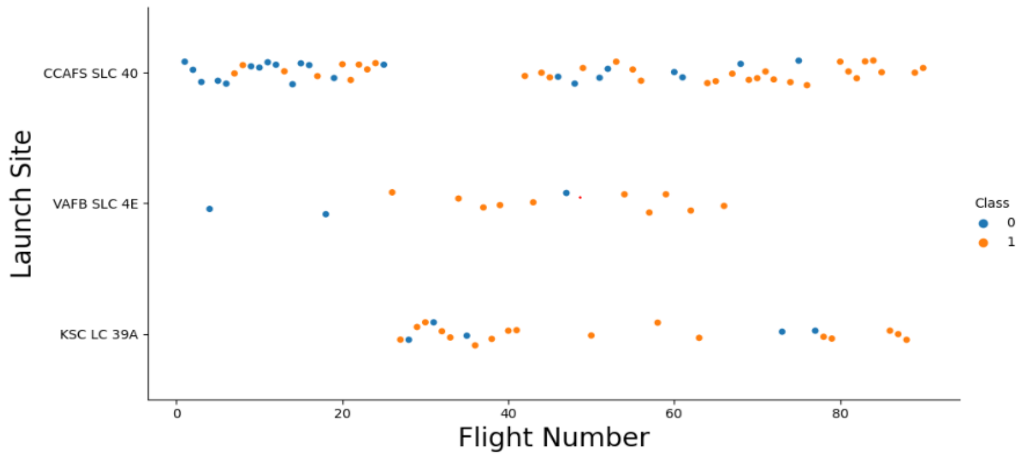
5. Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

[link to code]

EDA with Data Visualization

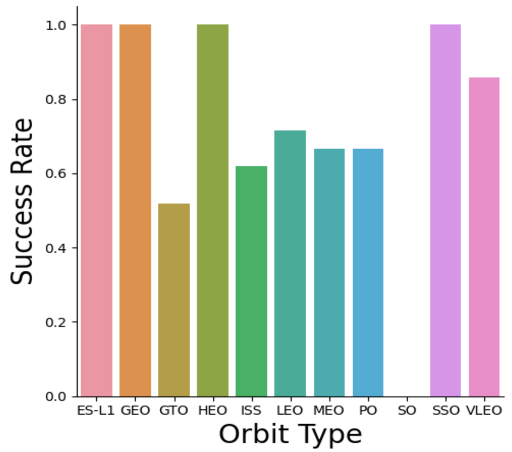
- **Scatter Graphs (shows correlation)**
- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload Mass (kg) vs. Launch Site
- Payload Mass (kg) vs. Orbit Type
- Orbit vs. Flight Number
- Orbit vs. Payload Mass



[Link to code]

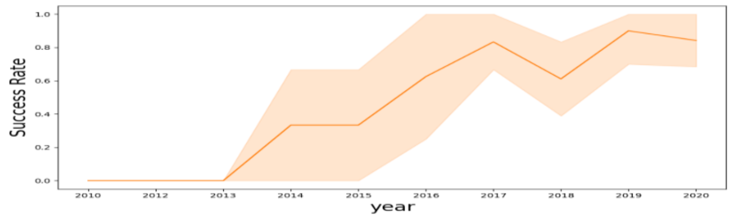
- **Bar Graph (Success Rate vs Orbit)**

Comparison of discrete measures



- **Line Graph (Success Rate vs Year)**

Shoes trends in data over time



EDA with SQL

- Below are the SQL queries that were used:
  - The names of the unique launch sites (space mission).
  - Top 5 records where launch sites begin with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Listing the date when the first successful landing outcome in ground pad was achieved.
  - Listing the names of the boosters which have success in drone ship 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 showing failed landing outcomes in drone ship, booster versions, launch sites for the months in year 2015.
  - Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

[Link to code]

Build an Interactive Map with Folium

- **Folium map with markers centered on NASA Johnson Space Center in Houston, Texas**
  - Circles at NASA Johnson Space Center's coordinate with label showing its name
  - Circles at each launch site coordinates with label showing launch site
  - The grouping of points in a cluster to display multiple and different information for the same coordinates
  - Markers to show successful (green) and unsuccessful landings (red). Green for successful landing and Red for unsuccessful landing.
  - Markers to show distance between launch site to proximities to key locations (railway, highway, coastline, city) and plot a line between

[Link to code]

Build a Dashboard with Plotly Dash

- **Dashboard has dropdown list with Launch Sites**

  Dropdown allows a user to choose the launch site or all launch sites

- **Pie chart shows successful launches**

  User can see successful and unsuccessful launches as % of total

- **Slider of payload mass range**

  Allows user to select payload mass in a fix range

- **Scatter chart shows payload mass vs. success rate by booster version**

  Allows user to see the correlation between Payload and Mass success

link to code

Predictive Analysis (Classification)

- **Data preparation**
  - Load dataset, create NumPy array from Class column
  - Standardize the data (StandardScaler)
  - Splitting the data into training and testing sets (train_test_split_function)
- **Model preparation**
  - Selection of machine learning algorithms
  - Set parameters for each algorithm with GridSearchCV for optimization
  - Training GridSearchModel models with training dataset
- **Model evaluation**
  - Get best hyper parameters for each type of model
  - Compute accuracy for each model using .score()
  - Plot Confusion Matrix for all models
- **Model comparison**
  - Comparison of models according to their accuracy
  - Identify best model using Jaccard_Score, F1_Score and Accuracy

Link to code

Results

- Exploratory data analysis results

    Shows that launch success improved over time

    Site KSC LC-39A has the highest success rate

- Interactive analytics demo in screenshots

    Launch sites are distant enough from major highways/railways/cities in case of launch failure

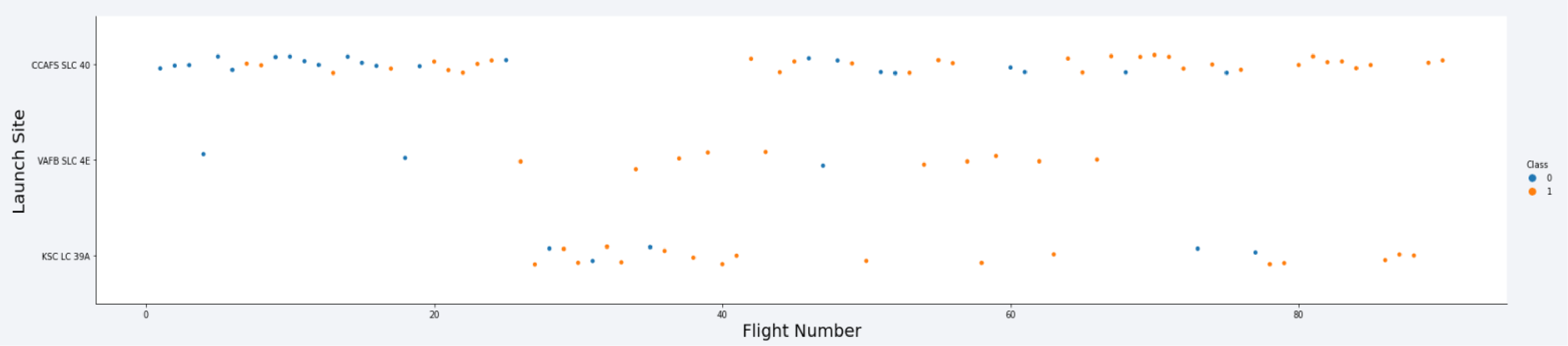
- Predictive analysis results

    Decision tree model is the best predictive model

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



According to the chart above the most successful launch site is CCAF5 SLC 40

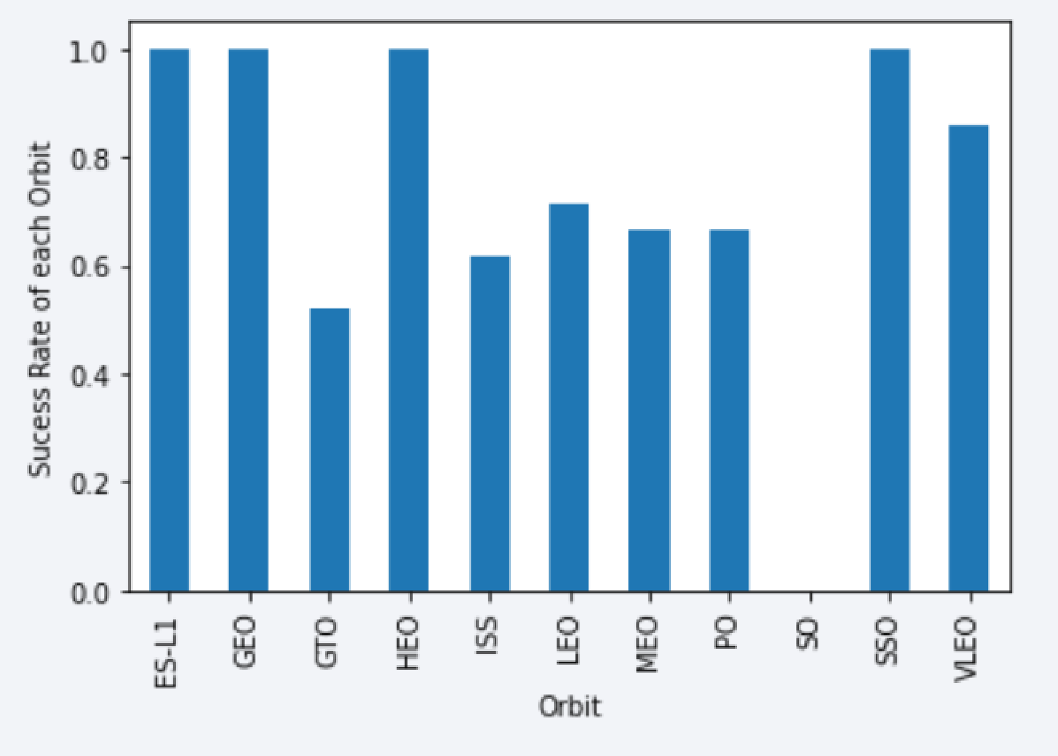
General success rate improved over time.

Payload vs. Launch Site

Depending on the launch site, heavier payloads had greater success rates, most of launches over 7000 kg were successful

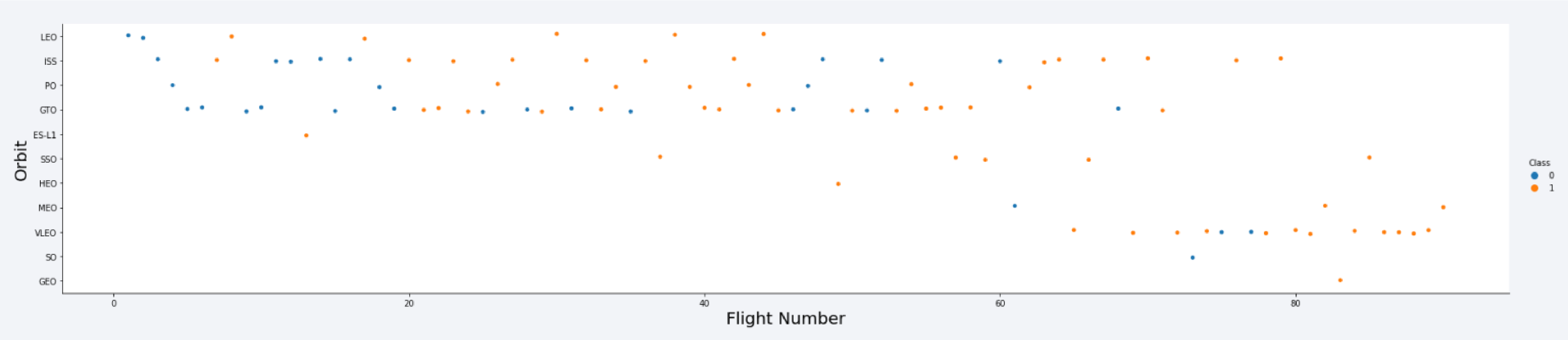
Payloads over 12,000 kg (heavy payload) can only be possible only on CCAFS SLC 40 and KSC LC 39A launch sites

Success Rate vs. Orbit Type



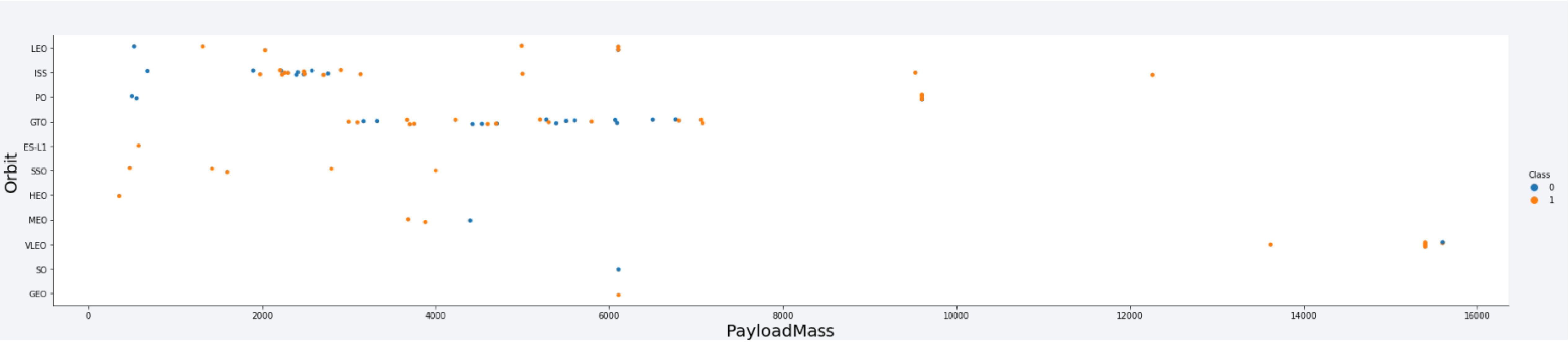
The graph shows success rates for different orbit types. According to this result orbits ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



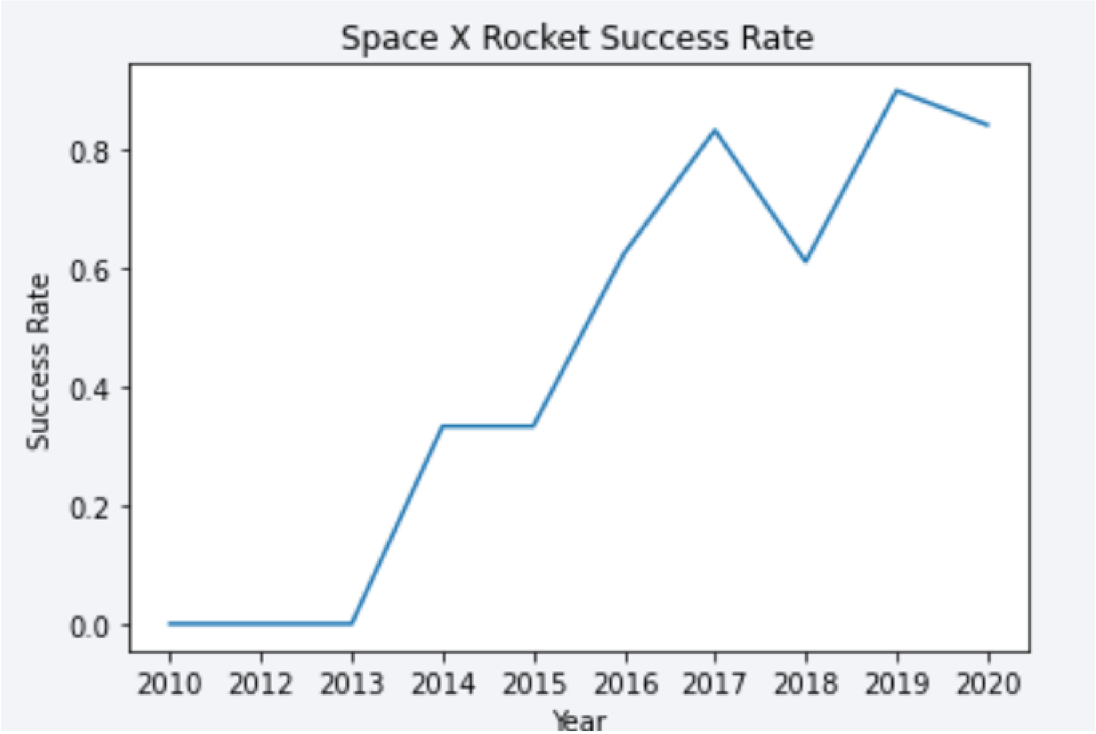
According to this chart - the success rate increased with the number of flights over time, especially for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights.

Payload vs. Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads have a negative influence on GTO orbits, but positive influence on the LEO orbit. There are only few launches for the orbits SO and GEO.

Launch Success Yearly Trend



Since 2013, we can see an increase in the Space X Rocket success rate all the way thru 2020.
First 3 years seem to be a period of adjustment.

All Launch Site Names

SQL Query

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Explanation

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE and we have 4 unique results.

Results

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

Launch Site Names Begin with 'CCA'

SQL Query

```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Results

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

The result filtered top 5 records and shows launch sites that being with name like 'CCA'

Total Payload Mass

SQL Query

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

The query returned the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL Query

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

This query returned the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL Query

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results

MIN("DATE")
01-05-2017

The query shows oldest successful landing (minimum).
The WHERE clause filters dataset in order to keep only records where landing was successful.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

Results

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

The query returned the booster version where landing was successful and payload mass is between 4000 and 6000 kg. Used WHERE and AND clauses to filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Results

SUCCESS	FAILURE
100	1

In this case we use subquery, first one counts successful mission, second one counts failures.

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Subquery was used to filter only heaviest payload mass using max function, then we selected only unique (distinct) values.

Results

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

SQL Query

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Results

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

This query returned month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function was used to get a from the year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

SQL Query

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Results

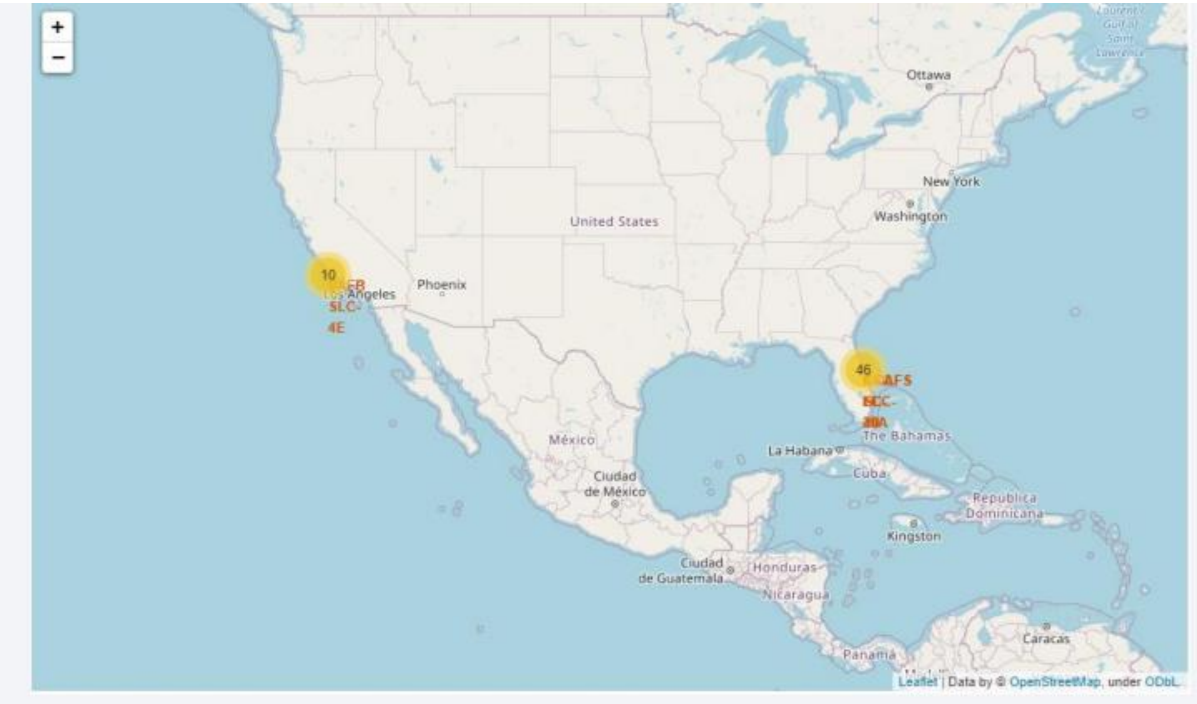
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

This query returned # of landing outcomes where mission was successful and date is between 2010/06/04 and 2017/03/20. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

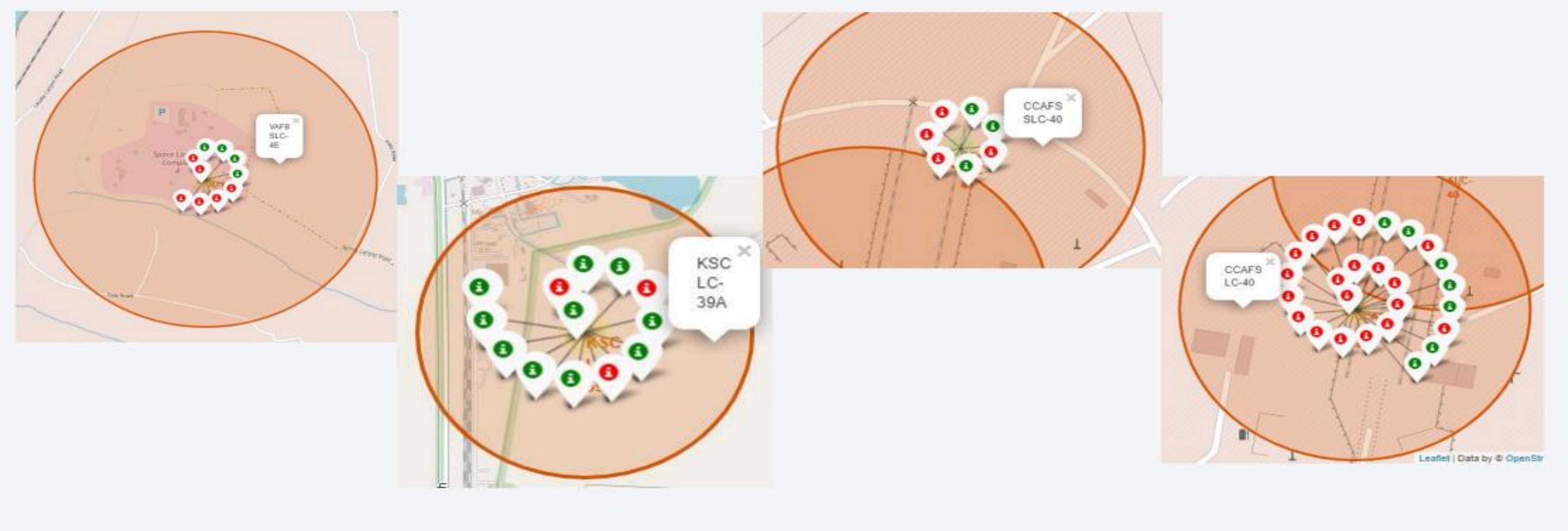
Launch Sites Proximities Analysis

Folium map – Ground stations



According to the map the Space X launch sites are located on the coast of the United States

Folium map – Color Labeled Markers



Green marker represents successful launches. Red marker represents unsuccessful launches. According to the map the KSC LC-39A has a higher launch success rate.

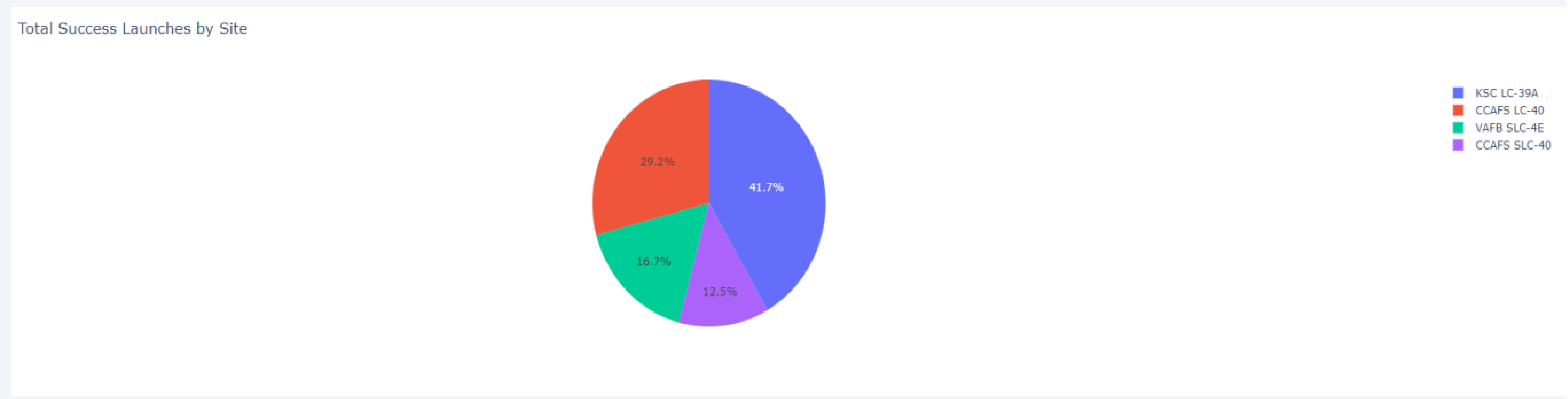
Folium Map – Distances between CCAFS SLC-40 and its proximities

Questions to answer: Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

Section 4

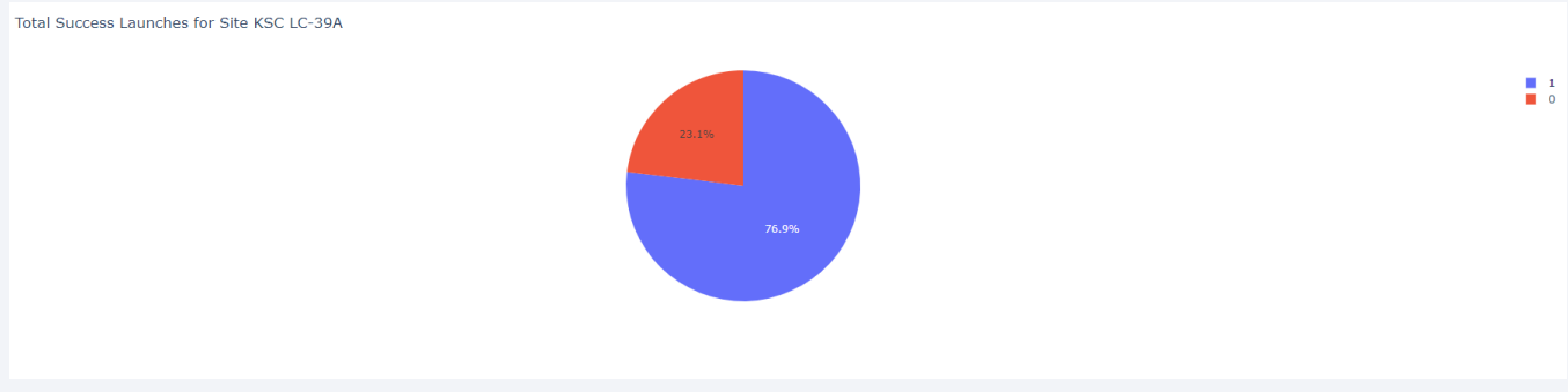
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



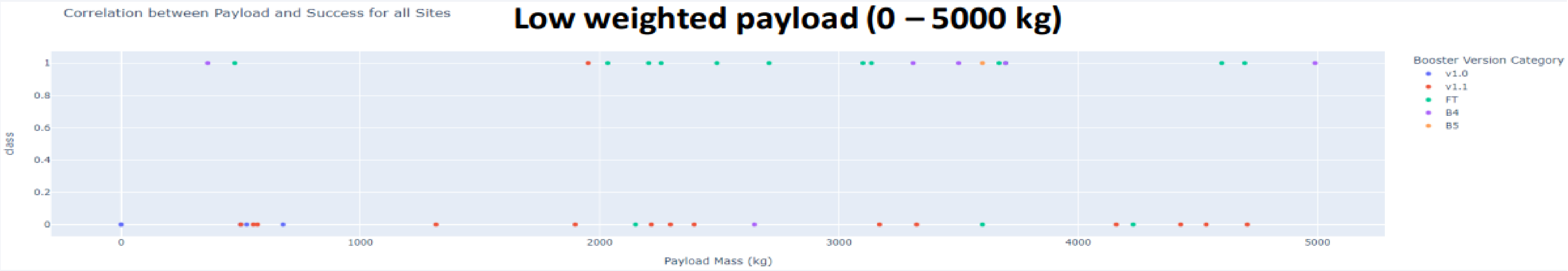
The launch place seems to be a very important factor to the success of the missions, KSC LC-39A has the best success rate of all launches.

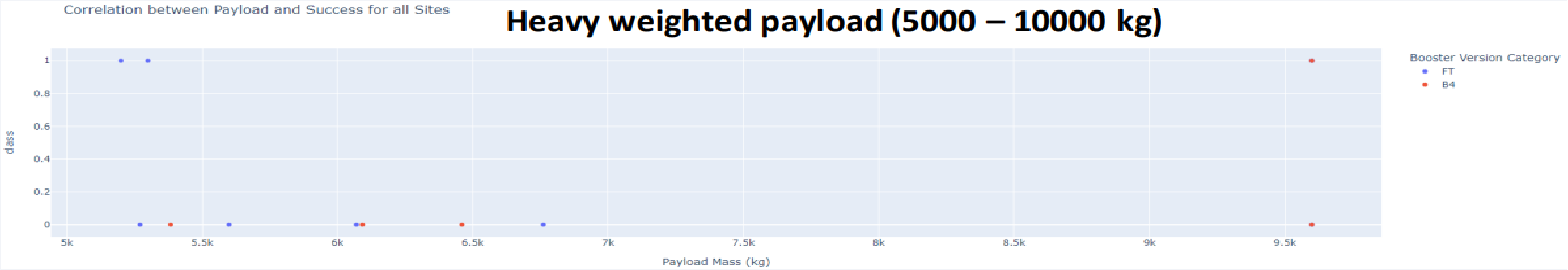
Dashboard – Total success launches for Site KSC LC-39A



The result indicates that KSC LC-39A has achieved a 76.9% success rate.

Dashboard – Payload mass vs. Outcome for all sites with different payload mass selected





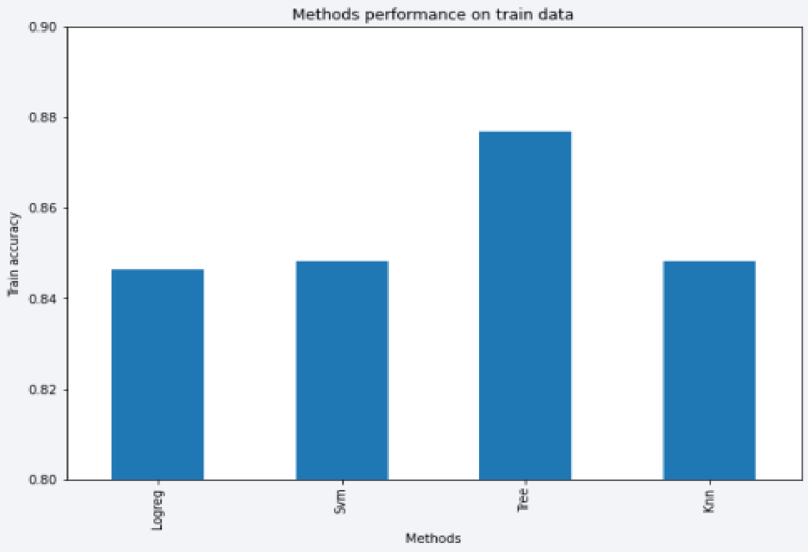
Low weighted payloads and Ft boosters have a better success rate than the heavy weighted payloads.

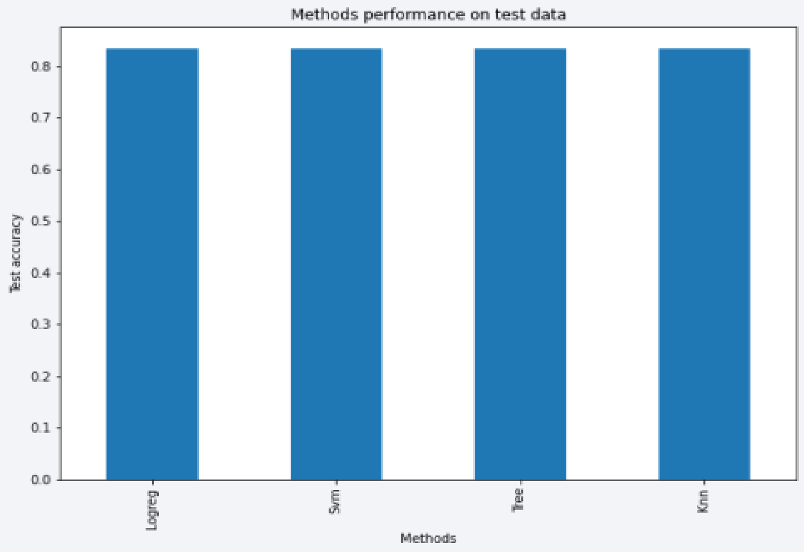
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333





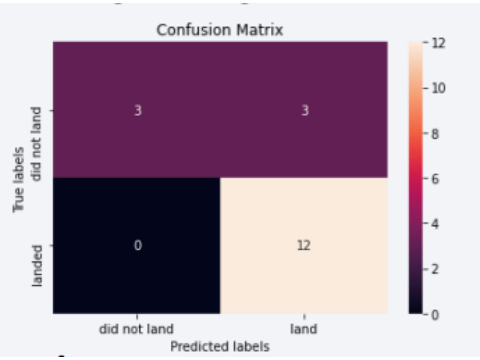
For accuracy test, all methods performed similar. Highest accuracy resulted from Decision Tree.

Decision tree best parameters

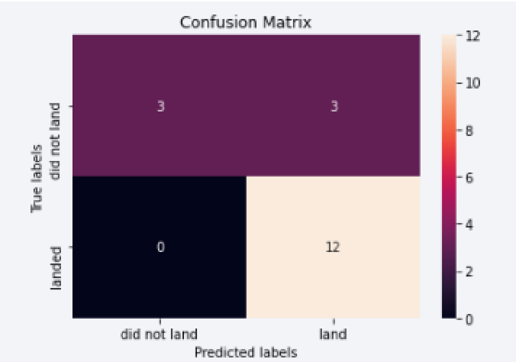
```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix

Logistic regression

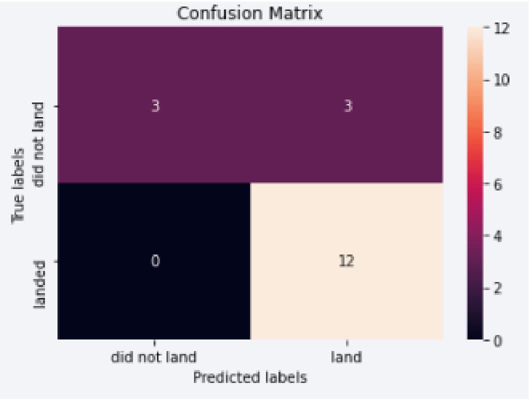


Decision Tree

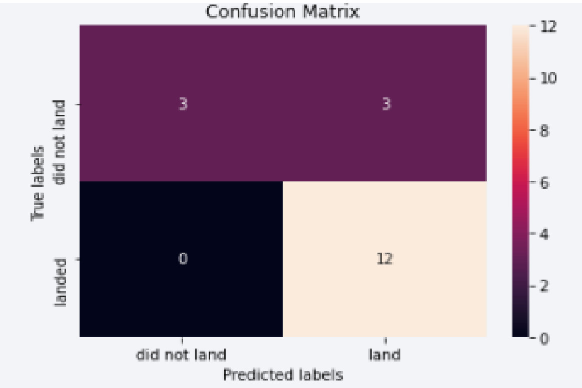


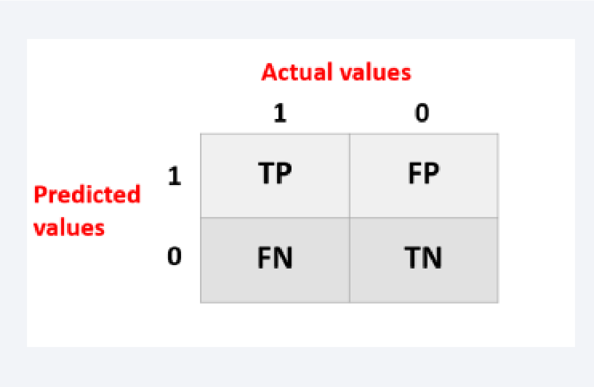
Accuracy tests are all equal, the confusion matrices are also identical. The main problem here in these models are high number of true positives and true negatives.

kNN



SVM





Conclusions

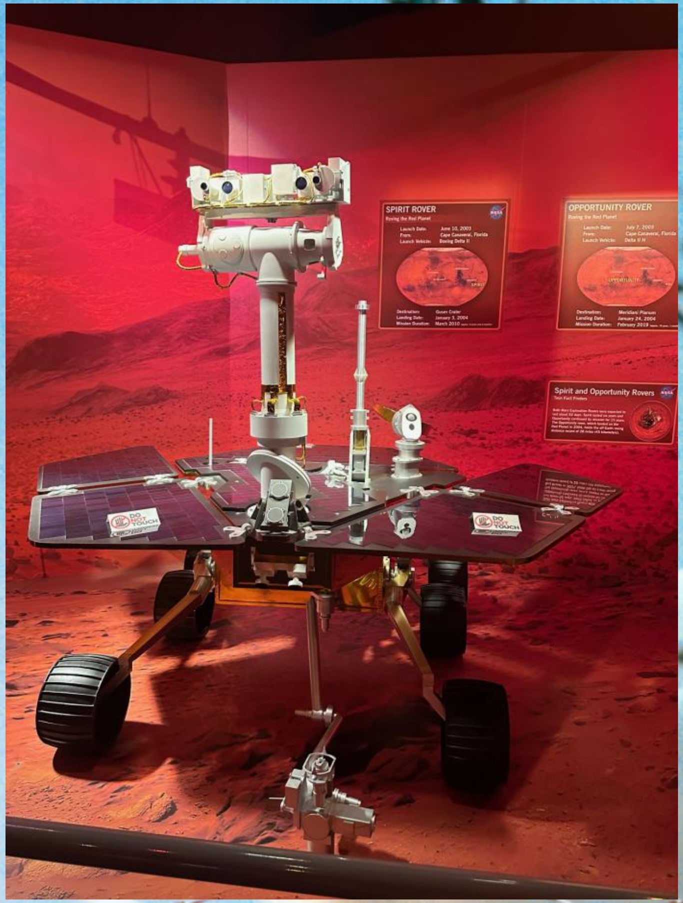
- The success of a mission can be expected with 83.3% accuracy.
- KSC LC-39A has the highest success rate of the launches from all sites.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- The success rate of launches increases over the years.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. In general, payloads with low mass perform better than the heavy weighted payloads.
- For this dataset, the Decision Tree Algorithm was the best model considering that the test accuracy between all the models used is identical. Decision Tree Algorithm had better accuracy using training model.

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

- Links to Github were added in the presentation

Thank you!


SPIRIT ROVER
OPPORTUNITY ROVER
Spirit and Opportunity Rovers