IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicolás Eugenio H.
September, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API and Web Scraping
  - Exploratory Data Analysis and Data Visualization
  - Exploratory Data Analysis using SQL
  - Visualization maps using Folium
  - Interactive Dashboards with Plotly
  - Predictive Analysis methods
- Summary of all results
  - Overview of predictive analysis methods
  - Accuracy of each method
  - Best performance method

Introduction

- Project background and context
  - The aim of the project was to predict if in a new launch of the SpaceX Falcon 9, the first stage will successfully land. SpaceX saves million dollars per launch recovering the first stage so it's crucial to understand the ideal conditions for achieving this. For other companies interested in competing with SpaceX this information is of enormous help.
- Problems you want to find answers
  - Main characteristics of a successful or failed landing.
  - Relationship of the rocket variables on the outcome of the landing.
  - Conditions on which SpaceX have achieved the best success rates for landings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API & Web Scraping from Wikipedia
- Perform data wrangling
  - Cleaning the created dataframes, keeping only relevant columns. One Hot Encoding for modeling manipulation
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, adjust and evaluate classification models

Data Collection

- The data collection stage consisted in two separated process:
  - SpaceX REST API:

  The first method to retrieve data was using the SpaceX REST API, that gave data about launches, including useful details about the launching conditions and landing outcomes.

  - Webscraping from Wikipedia:

  Using the library BeautifulSoup the launch records were extracted from a Wikipedia webpage and parsed it into a pandas dataframe.

Data Collection – SpaceX API

notebook link

- SpaceX REST API:

The calls and method is illustrated as follows:

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

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

```
# Call getBoosterVersion
getBoosterVersion(data)
```

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

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

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

The data was retrieved, cleaned and exported:

```
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}
```

```
data2 = pd.DataFrame(launch_dict)
```

```
data_falcon9 = data2[data2['BoosterVersion']!='Falcon 1']
```

```
payload_mean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mean)
```

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

Data Collection - Scraping

notebook link

- Webscraping:

In this stage the data was retrieved from a Wikipedia table using BeautifulSoup, and parsed into a pandas dataframe

```
response = requests.get(static_url)
soup = BeautifulSoup(response.text,'html')
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
tc = first_launch_table.find_all('th')
for th in tc:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
            column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each
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']=[]
```

```
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()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #
            launch_dict['Flight No.'].append(flight_number)
            print(flight_number)
```

```
for k in launch_dict.keys():
    print("Key {} => Len {}".format(k, len(k)))
```

```
df = pd.DataFrame(dict([ (k,pd.Series(v)) for k,v in launch_dict.items() ]))
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

notebook link

- In the data wrangling process some Exploratory Data Analysis (EDA) was performed, as follows:

```
df.isnull().sum()/df.shape[0]*100
df.dtypes
df['LaunchSite'].value_counts()
df['Orbit'].value_counts()
landing_outcomes = df['Outcome'].value_counts()
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
landing_class = df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)
df["Class"].mean()
```

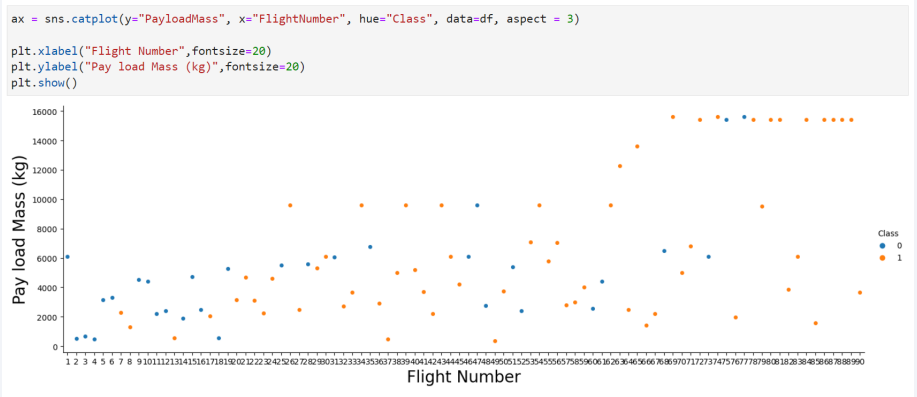
EDA with Data Visualization

notebook link

The following types of charts were plotted:

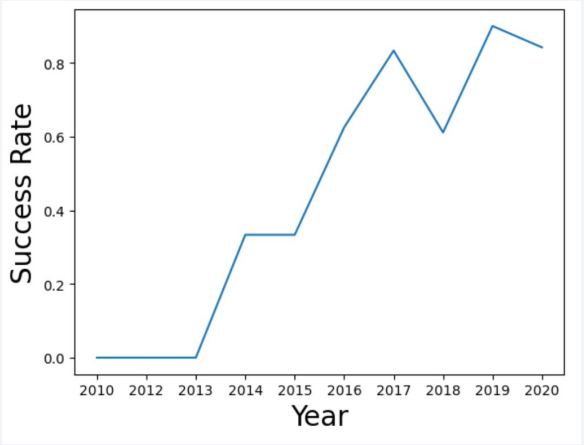
- Scatter plot
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Orbit vs Flight Number
  - Payload vs Orbit Type
  - Orbit vs Payload Mass

These plots were used to explore correlation between variables



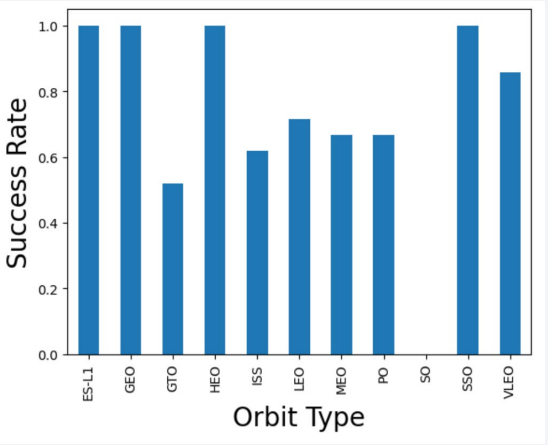
- Line plot
  - Success Rate vs Year

Line plot was used for exploring the trend over time



- Bar plot
  - Success rate vs Orbit

Bar plot was used for show the relationship for categorical variables



EDA with SQL

notebook link

The following SQL queries were performed:

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

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

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

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome='Success (ground pad)'
```

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND Landing_Outcome='Success (drone ship)'
```

```
%%sql SELECT
MISSION_OUTCOME
, COUNT(*) AS Total_number
FROM SPACEXTBL
WHERE
MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
GROUP BY(MISSION_OUTCOME)
```

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

```
%%sql
SELECT
  substr(Date,6,2) as month
, Landing_Outcome
, BOOSTER_VERSION
, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
AND substr(Date,1,4) = '2015'
```

```
%%sql
SELECT
Landing_Outcome
, COUNT(Landing_Outcome) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY TOTAL_NUMBER DESC
```

Interactive Map with Folium

notebook link

- The following map Folium objects were created:
  - Red circle at NASA Ground stations as: name(folium.Circle, folium.map.Marker).
  - Red circles at launching sites coordinates as: (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - Markers cluster for grouping launching sites as: (folium.plugins.MarkerCluster).
  - Color labeled markers to show successful and unsuccessful landings as: (folium.map.Marker, folium.Icon).
  - Lines betweens launching sites to key locations and distance labels as: (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- These objects were shown to visualize the situation easily and gain insights on the trends and behavior of the geographical conditions.

Dashboard with Plotly Dash

notebook link

- The created dashboard has a dropdown for condition selection, a pie chart, a rangeslider and a scatter plot.

Each of these interactive features were added to have either an specific data visualization, or a better visualization in the selected data:

- The dropdown allows the user to choose the launch site.
- The pie chart shows the success ratio of the selected launch site.
- The rangeslider allows the user to select a payload mass range to have a better visualization
- The scatter chart shows the relationship between the Success ratio against the Payload Mass

Predictive Analysis (Classification) notebook link

Several steps were used to found the best performing classification model, such as:

Data preparation

- Load dataset
- Normalize data
- Split data into training and test sets

Model preparation

Selection of ML algorithms
Set parameters for each algorithm to GridSearchCV
Training GridSearchModel models with training dataset

Model evaluation

Get best hyperparameters for each type of model
Compute accuracy for each model with test dataset
Plot Confusion Matrix

Model comparison

Comparison of models according to their accuracy
Choose the best model

Results

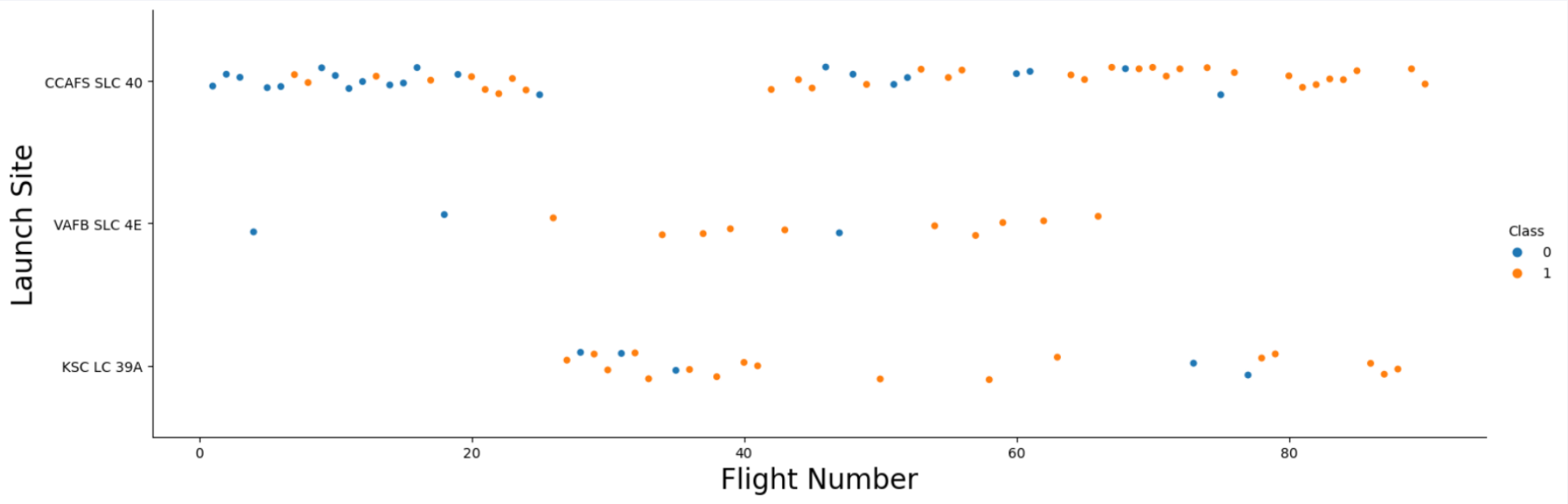
The obtained results can be summarized as follows and are shown in the next section:

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

Section 2

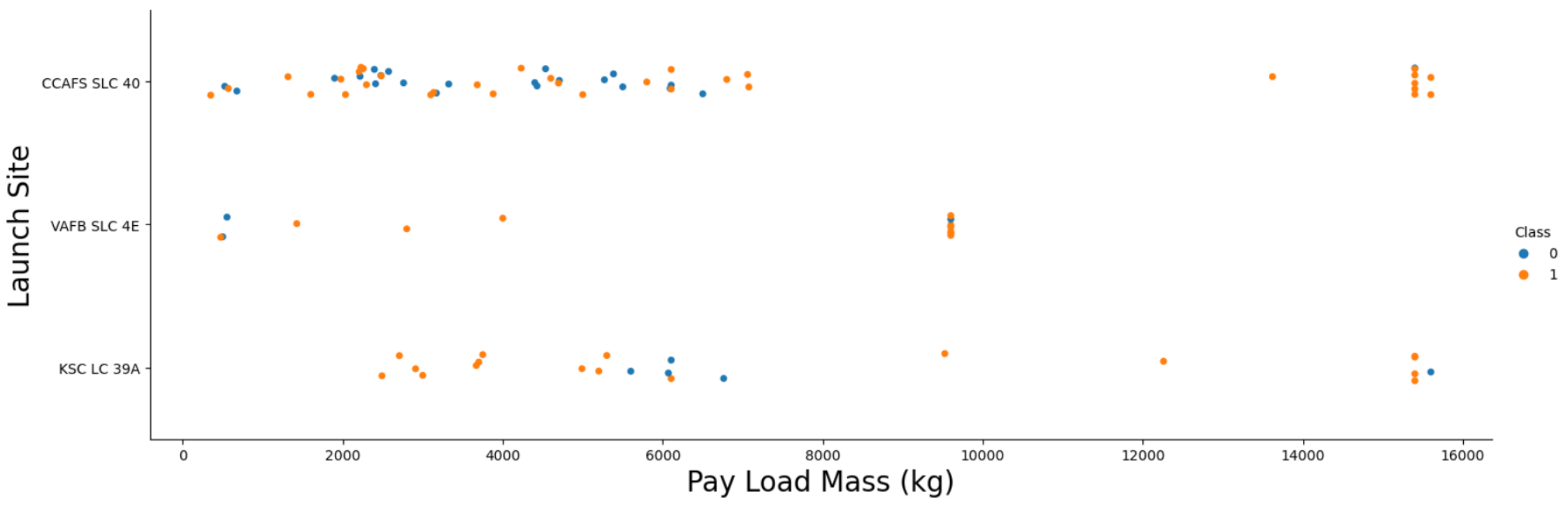
Insights drawn from EDA

Flight Number vs. Launch Site



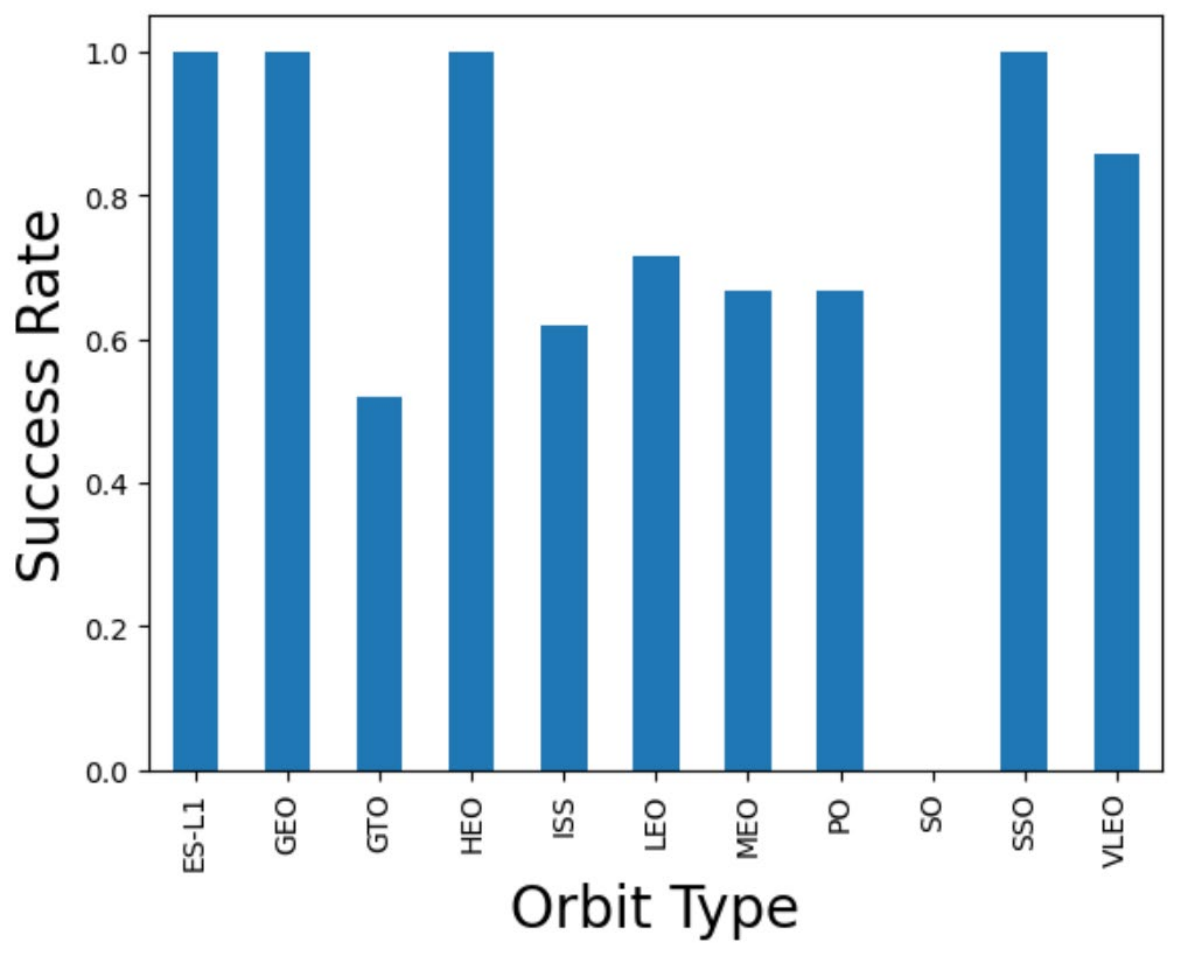
- We see that different launch sites have different success rates. Being CCAFS LC-40 the one with the lowest success rate, and KSC LC-39A and VAFB SLC 4E have the highest ones.

Payload vs. Launch Site



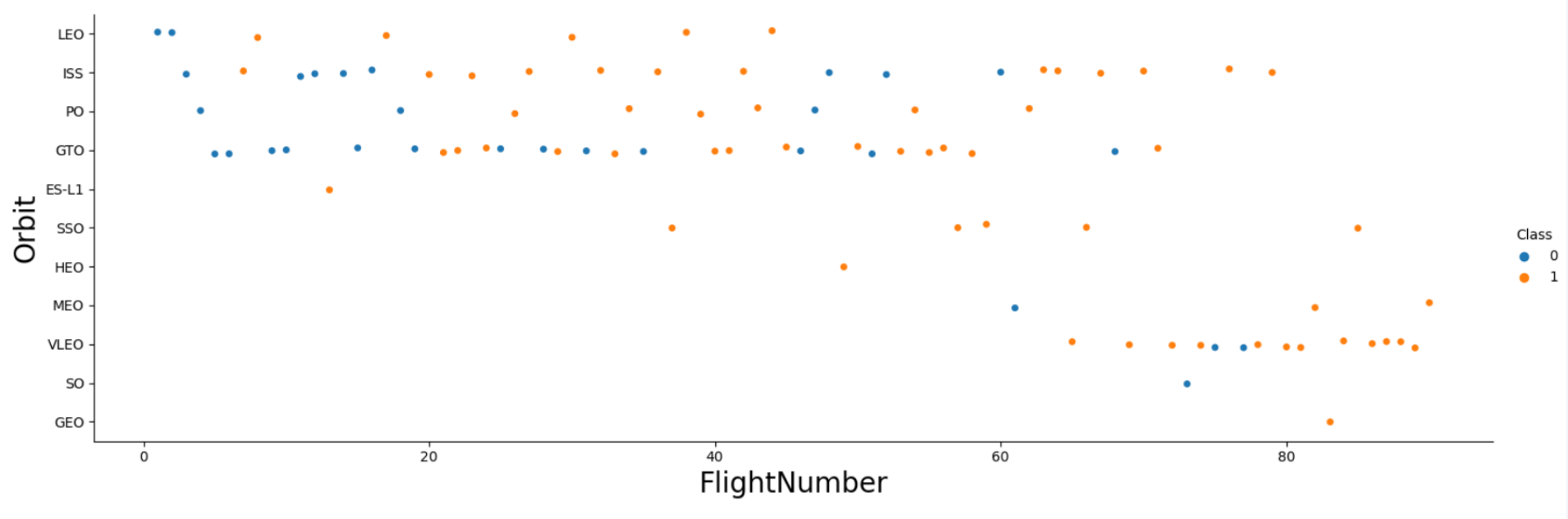
- For the VAFB-SLC launchsite there are no rockets launched with heavy payload mass (greater than 10000). Also above 10000 kg there are fewer unsuccessful landings.

Success Rate vs. Orbit Type



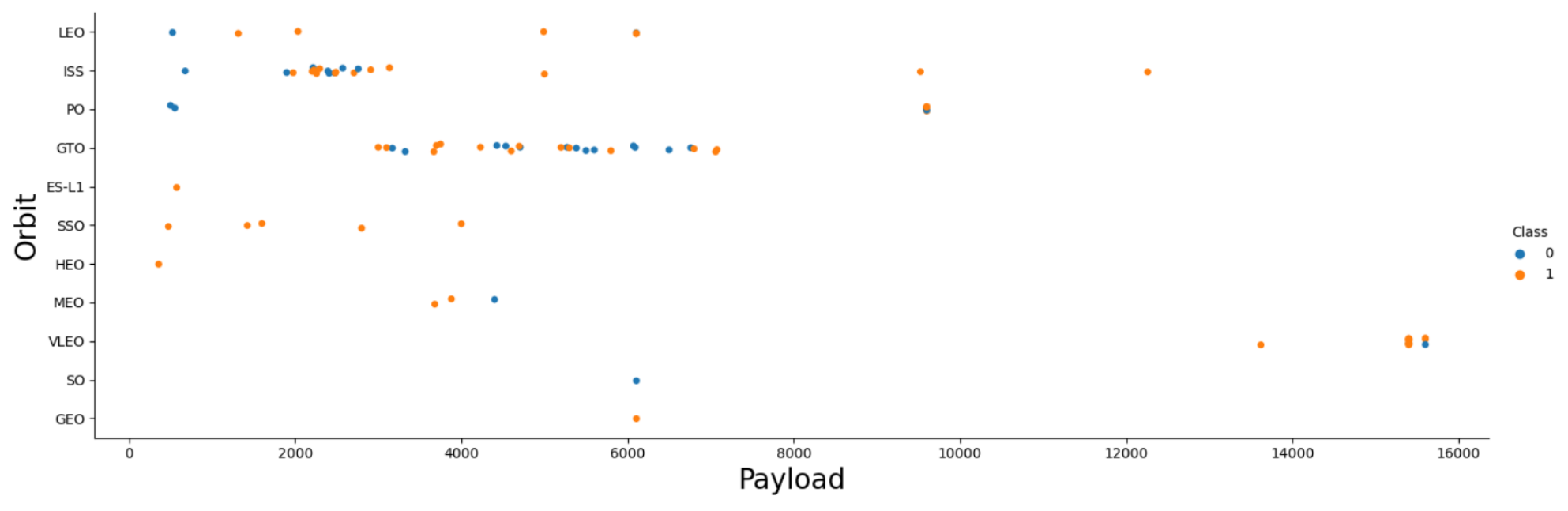
- We can observe that the ES-L1, GEO, HEO and SSO there is a success rate of 1.
- The GTO orbit is the one with the lowest success rate

Flight Number vs. Orbit Type



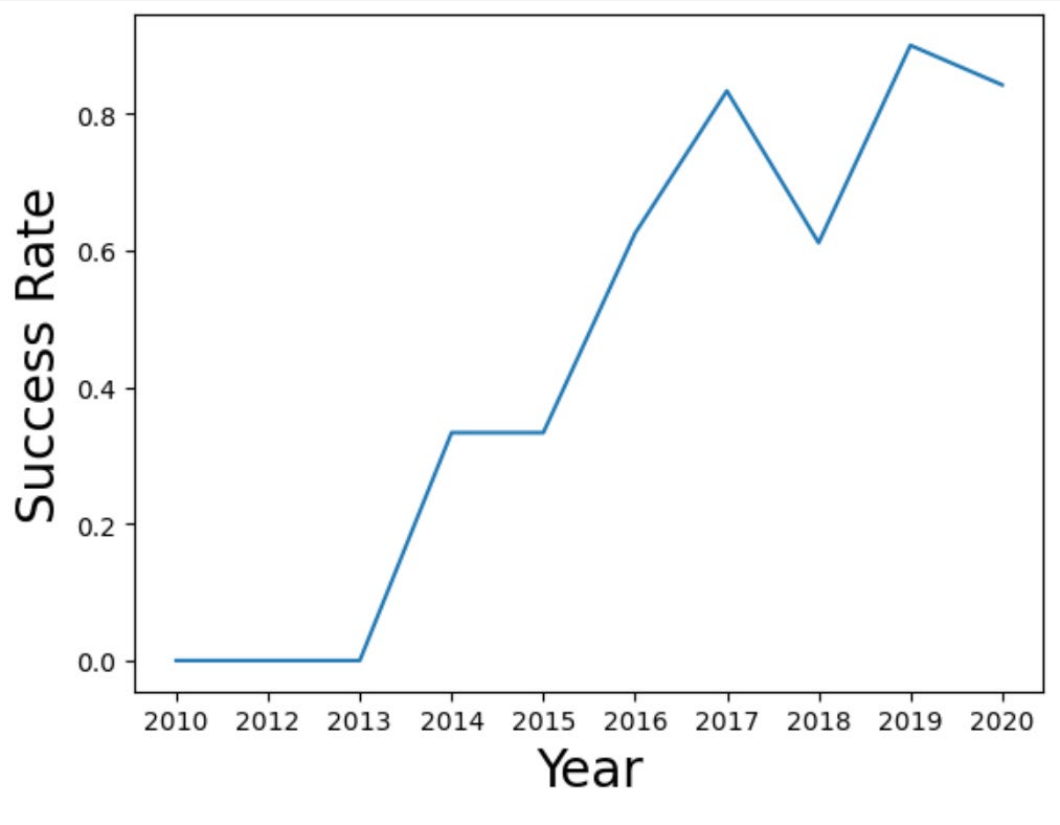
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- In the LEO orbit the Success appears related to the 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 are both there.

Launch Success Yearly Trend



- The success rate since 2013 had an increasing trend till 2020, having a drawback in 2018.

All Launch Site Names

- For finding the name of distinct launches the “DISTINCT” clause was used when calling the column “LAUNCH_SITE”.

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

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

To find 5 records beginning with ‘CAA’, the LIKE clause was used adding the wildcard character % at the end of the prefix, and a LIMIT clause to display only 5 records.

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- To calculate the total payload carried by boosters from NASA, the entries of the column PAYLOAD_MASS__KG_ was summed, using the WHERE clause to choose the CUSTOMER column entries that matches ‘NASA (CRS)’

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1, it was took the AVERAGE of the PAYLOAD_MAS__KG column of the entries that matches the ‘F9 v1.1’ in the BOOSTER_VERSION using a WHERE clause

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- To find the dates of the first successful landing outcome on ground pad, the MIN entry was selected from the column DATE, for the landing outcomes that matches 'Success (ground pad)' in the Landing_Outcome column using a WHERE clause

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome='Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, the BOOSTER_VERSION column was selected, filtering the entries of the PAYLOAD_MASS__KG_ that was in a range of 4000 and 6000 using a BETWEEN clause, and that were also matching the 'Success (drone ship)' entry on the Landing_Outcome column using an AND clause in the WHERE filtering

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 PAYLOAD_MASS__KG_ between 4000 and 6000 AND Landing_Outcome='Success (drone ship)'
```

```
 * 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

- To calculate the total number of successful and failure mission outcomes, it was counted the number of all columns using the clause COUNT, along with selecting the column MISSION OUTCOME, and grouping the results by the latter column. In this way the counting is performed on each group of outcome. Also this was filter to count only the outcomes of Success and Failures entries using the LIKE clause.

```
%%sql SELECT
MISSION_OUTCOME
, COUNT(*) AS Total_number
FROM SPACEXTBL
WHERE
MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
GROUP BY(MISSION_OUTCOME)
```

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

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

Boosters Carried Maximum Payload

- To list the names of the booster which have carried the maximum payload mass, the Booster_Version and PAYLOAD_MASS__KG_ were selected, and using a WHERE clause, only the entries that matches the maximum PAYLOAD_MASS__KG_ was selected, getting that number from a subquery performed applying the MAX function to the PAYLOAD_MAS__KG_ column.

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- To list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, the entries that matches the ‘Failure (drone ship)’ string were selected from the Landing_Outcome column using a WHERE clause. Using the substr function the first four characters of the Date column were extracted to get the year of the entries, and using the AND clause this was compared to ‘2015’ to filter the selected year within the WHERE section.

```
%%sql
SELECT
  substr(Date,6,2) as month
, Landing_Outcome
, BOOSTER_VERSION
, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
AND substr(Date,1,4) = '2015'
```

month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

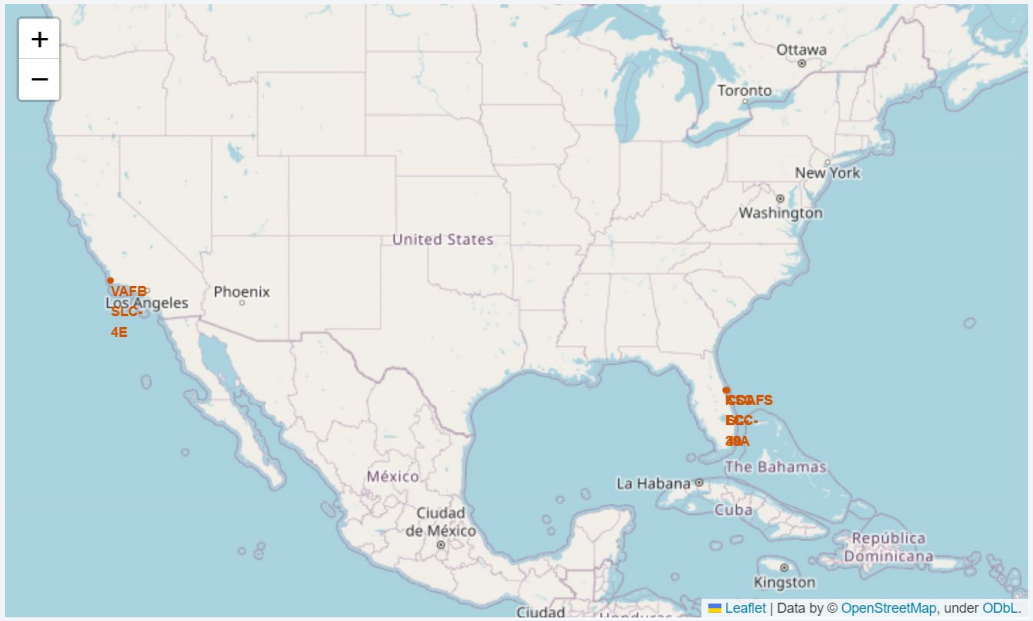
- To rank the count of landing outcomes or between the date 2010-06-04 and 2017-03-20, in descending order, the procedure was as same as previous, grouping by the Landing_Outcome column and counting the column in each group as TOTAL_NUMBER. Here also is added a filter using a WHERE clause to match the dates using a BETWEEN clause, specifying the desired date range. Then it was sorted in a descendent order using the clauses ORDER BY and DESC.

```
%%sql
SELECT
Landing_Outcome
, COUNT(Landing_Outcome) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY TOTAL_NUMBER DESC
```

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

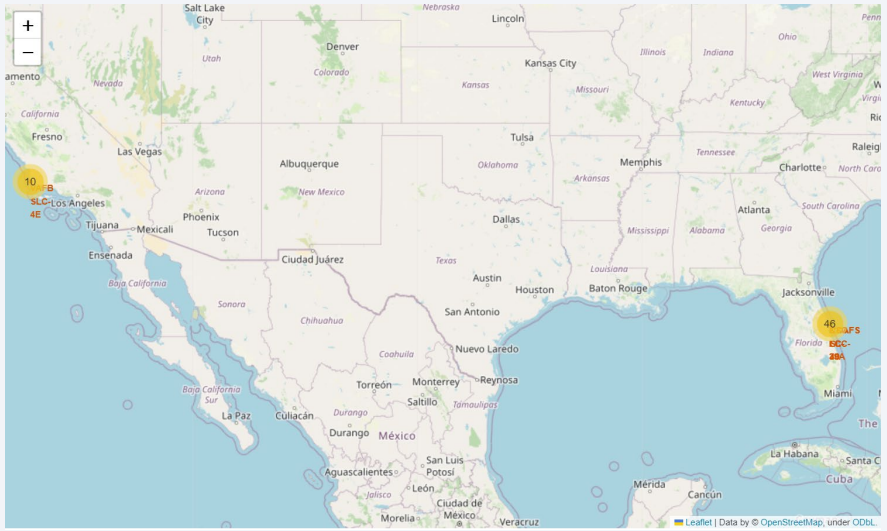

Section 3
Launch Sites
Proximities Analysis

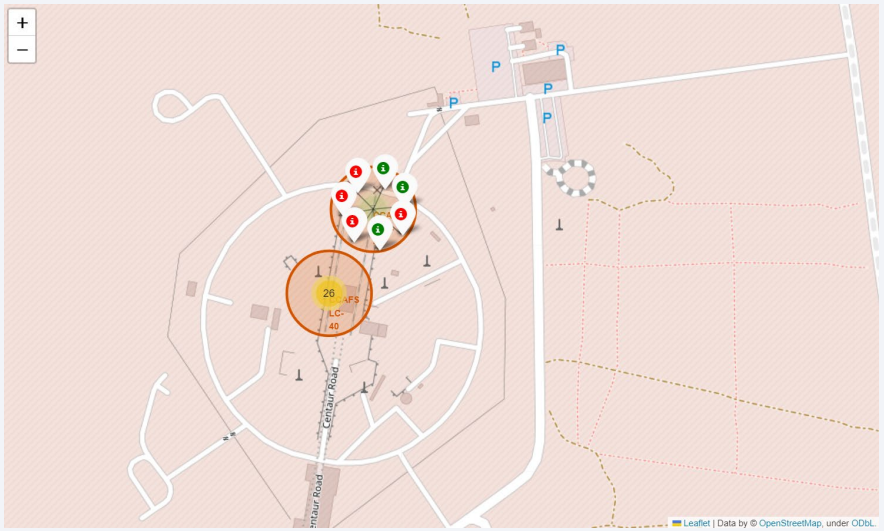
Global map showing launching locations in the US



We see some Space X launch locations that are located in the US coasts.

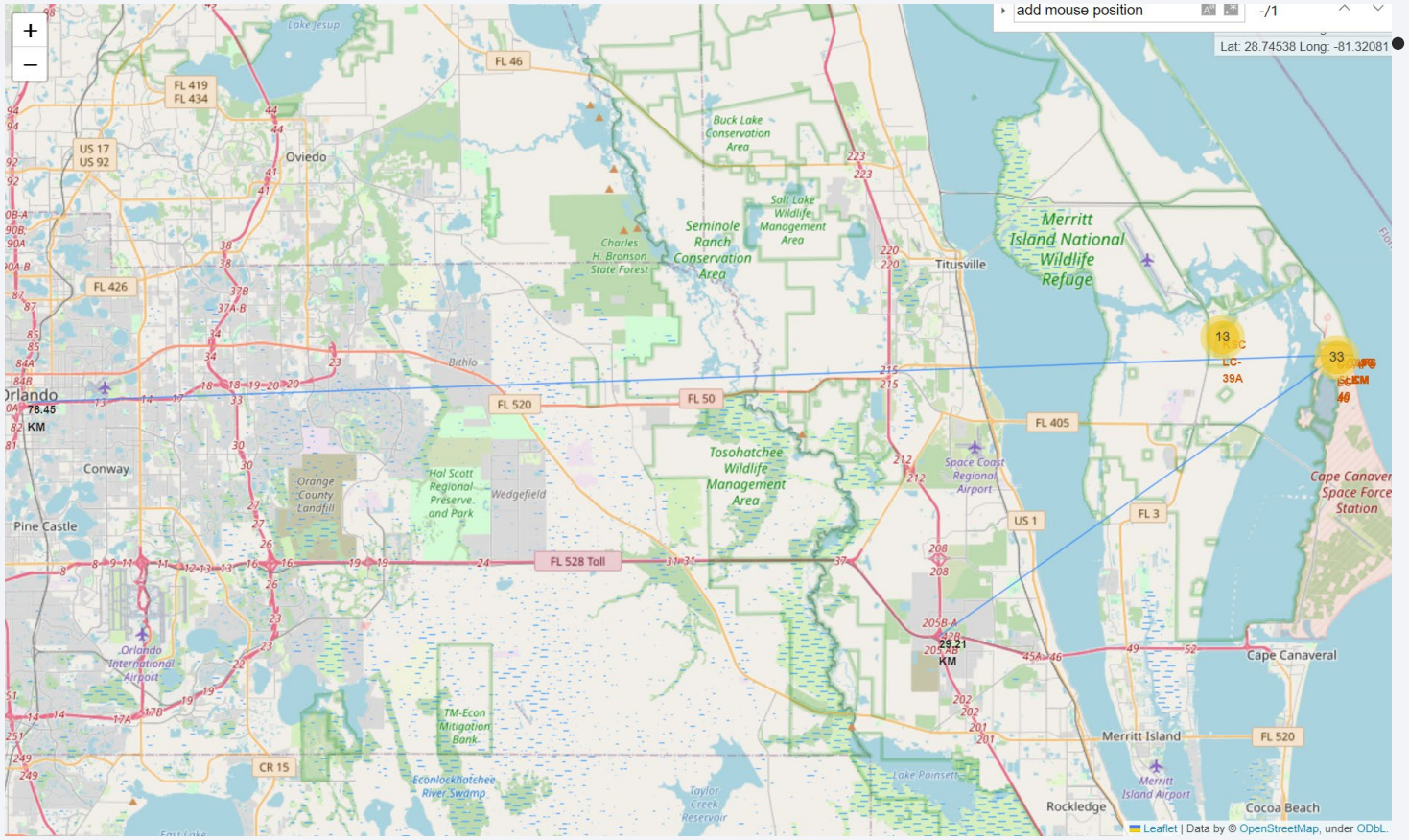
Color-labeled launch outcomes



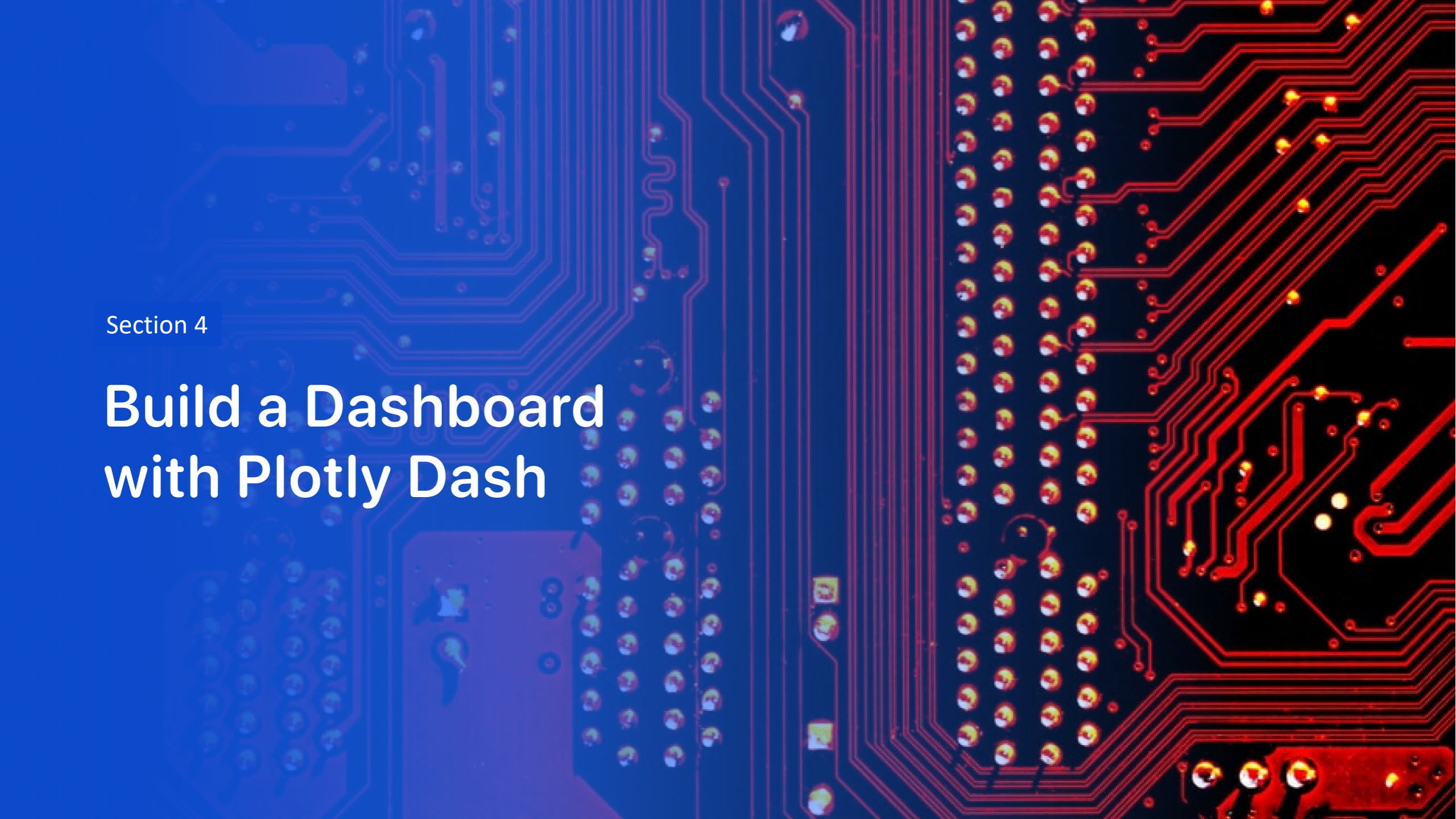


- For an specific launching site, the outcomes are shown in a color label, where the Green marker represent a successful launch and a red one represent unsuccessful ones.

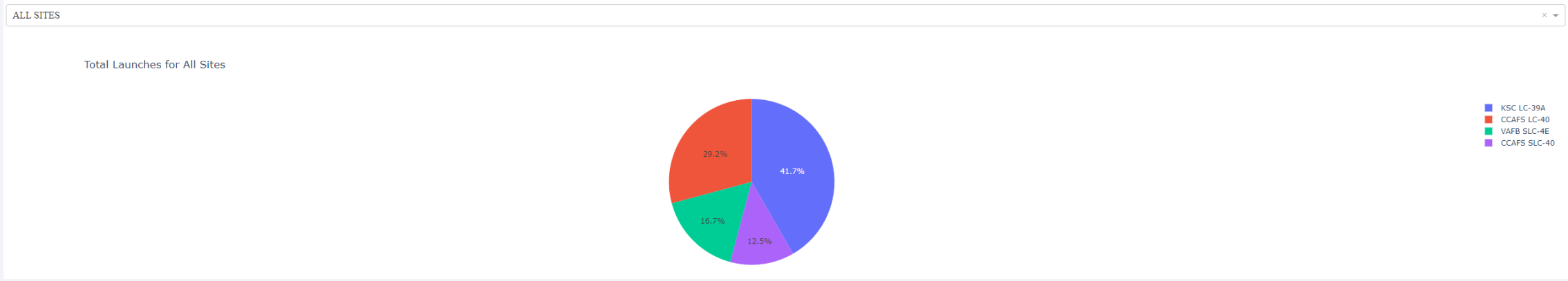
Distances between locations



- It's shown a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed


Section 4
Build a Dashboard with Plotly Dash

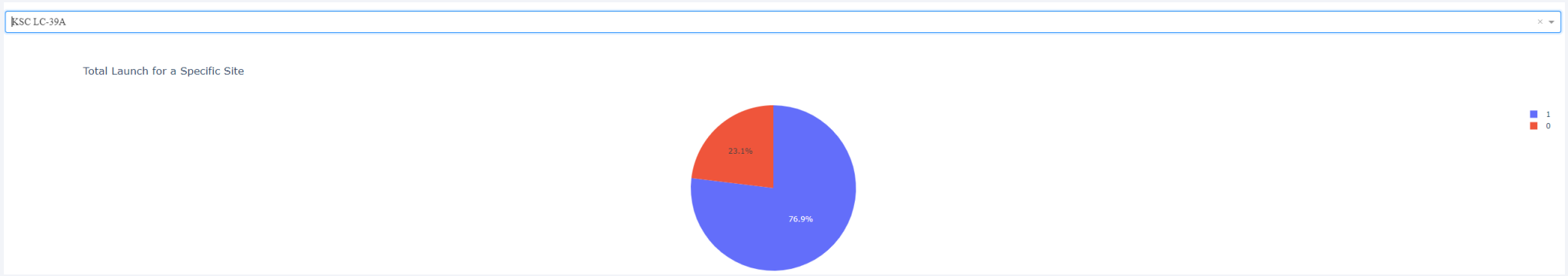
Success launches for all sites - Pie chart



- We can observe that KSC LC-39 A has the best success rate of launches with a 41.7% of the total successful launches.

We can see that ‘All sites’ is selected in the dropdown filter for the interactive created pie chart, generating also its legend based on the different categories

Success launches for KSC LC-39A - Pie chart



- We can see a pie chart for the launch site KSC LC-39A, the one with the highest success ratio, with a 76.9% of its launchings being a success.

Here the site KSC LC-39A is selected in the dropdown menu, creating a new pie-chart with it's correspondent legend

Scatter plot of Payload vs Launch Outcome in all sites

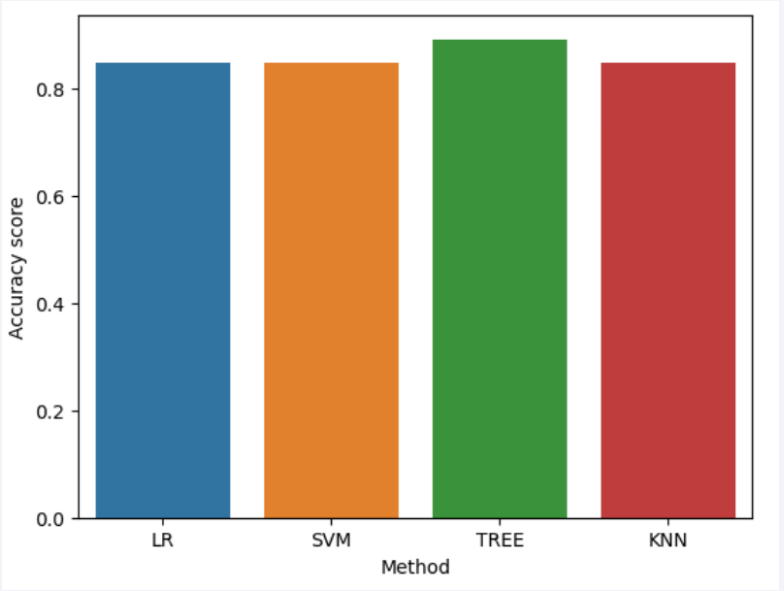
- The scatter plots are generated for all sites using the dropdown menu, at the top with the payload being on the range 0-5000 using the slider, and in the top in the range 5000-10000.
- It's seen that the loweighted paylods (<5000) have a better success rate than the heavy weighted ones.

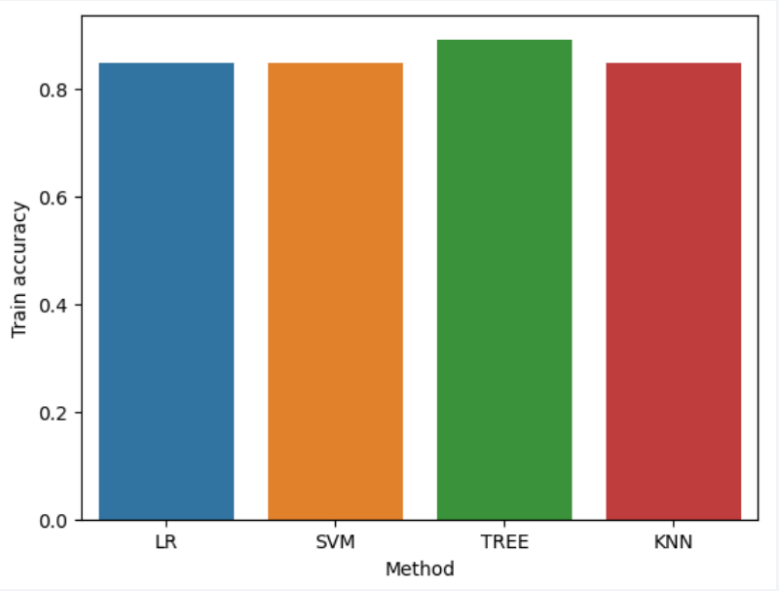
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Method	Test accuracy	Accuracy score
LR	0.833	0.847
SVM	0.833	0.848
TREE	0.833	0.891
KNN	0.833	0.848

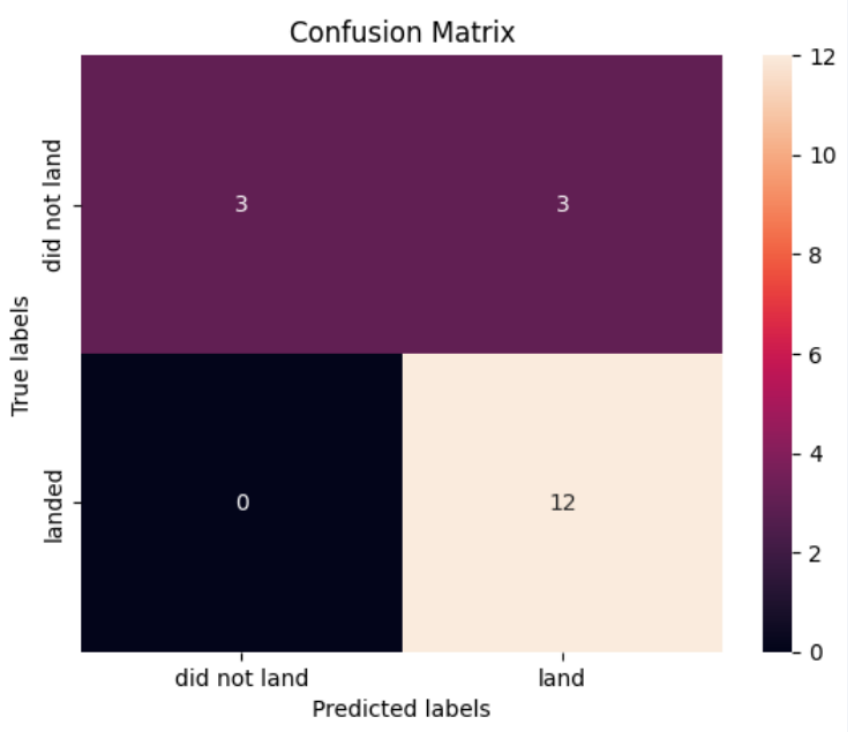




- The accuracy on the test dataset all methods performed equally, but for the accuracy score the Decision Tree performed slightly better with a score of 0.891

Confusion Matrix

- The confusion matrix of the Decision Tree is shown:



It's seen that the true positive rate is of 1!

But both the false positives and true negatives rates are only 0.5

In this way the accuracy of the model is 0.83

Conclusions

- The success of a mission can attributed to different factors. It can be seen that Success rates have been increasing over time.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- The payload mass is a big influence on how likely the output is to be a success. Some orbits require a light or heavy payload mass. Generally low weighted payloads perform better than the heavy weighted payloads.
- KSC LC-39A is currently the launch site with the better outcomes.
- It can be seen that the Decision Tree Classifier is the best ML model for the dataset available.

Appendix

GitHub repository with all the notebooks

Thank you!