Applied Data Science Capstone

Tejashwini PM

17/11/2023

Table Of Contents

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

Executive Summary

- Summary of Methodologies
- Data Collection via API, Web Scraping
- Exploratory Data Analysis (EDA) with Data Visualization
- EDA with SQL
- Interactive Map with Folium
- Dashboards with Plotly Dash
- Predictive Analysis

- Summary of Results
- Exploratory Data Analysis results
- Interactive maps and dashboard
- Predictive analysis results

Introduction

- Project background and context :

The objective of this project is to predict the successful landing of the Falcon 9 first stage. According to SpaceX's website, the launch cost of the Falcon 9 rocket is $62 million, while other providers charge upwards of $165 million for each launch. The significant price difference is attributed to SpaceX's ability to reuse the first stage. By determining the likelihood of a successful landing, we can calculate the cost of a launch. This information holds importance for other companies looking to compete with SpaceX in the rocket launch industry.

- Problems to answer :

1. What are the main characteristics of a successful or failed landing ?
2. What are the effects of each relationship of the rocket variables on the success or failure of a Landing?
3. What are the conditions which will allow SpaceX to achieve the best landing success rate ?

Methodology

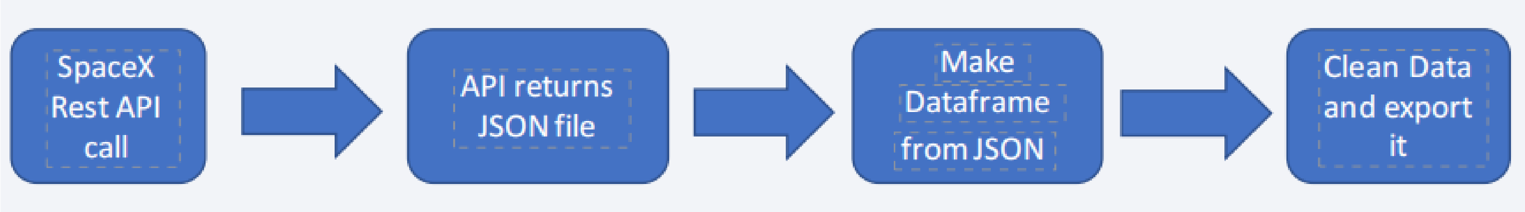
Methodology

Executive Summary

- Data Collection Methodology:
- SpaceX REST API
- Web scrapping
- Perform Data Wrangling
- Dropping unnecessary columns
- One Hot Encoding for classification models
- 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

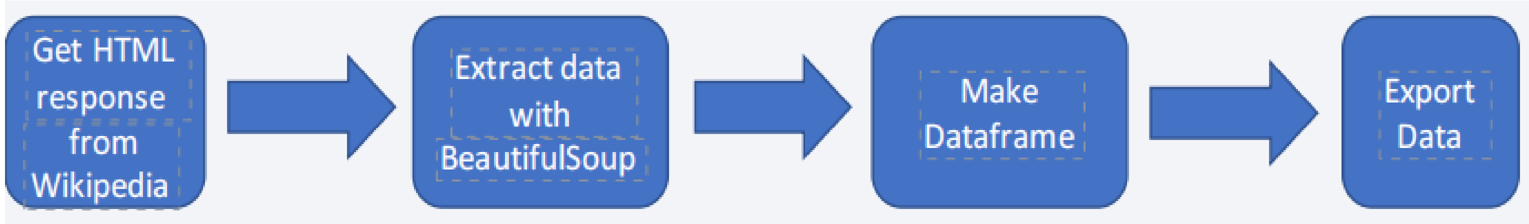
Data Collection

- Datasets are collected from Rest SpaceX API and web scrapping Wikipedia
- The information obtained by the API are rocket, launches, payload information
- The Space X REST API URL is api.spacexdata.com/v4/



SpaceX Rest API call → API returns JSON file → Make Dataframe from JSON → Clean Data and export it

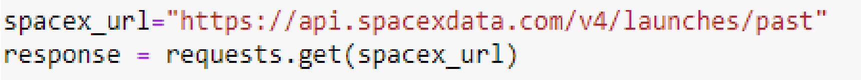
The information obtained by the web scrapping of Wikipedia are launches, landing, payload information.



Get HTML response from Wikipedia → Extract data with BeautifulSoup → Make Dataframe → Export Data

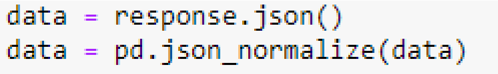
Data Collection – SpaceX API

1.Getting Response from API



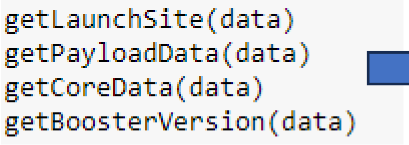
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON file



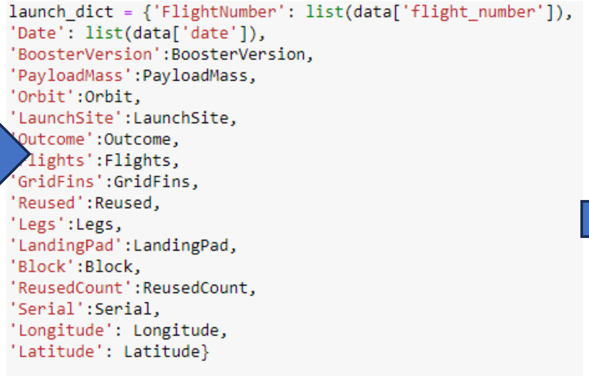
```
data = response.json()
data = pd.json_normalize(data)
```

3. Transform Data



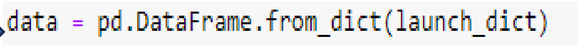
```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

4. Create dictionary with data



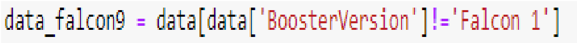
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe



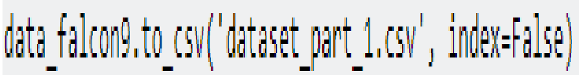
```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe



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

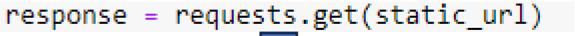
7. Export to file



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

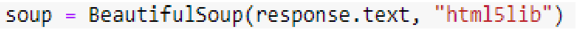
Data Collection - Scraping

1. Getting Response from HTML



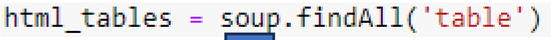
```
response = requests.get(static_url)
```

2. Create Beautiful Soup Object



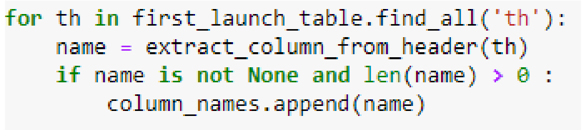
```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables



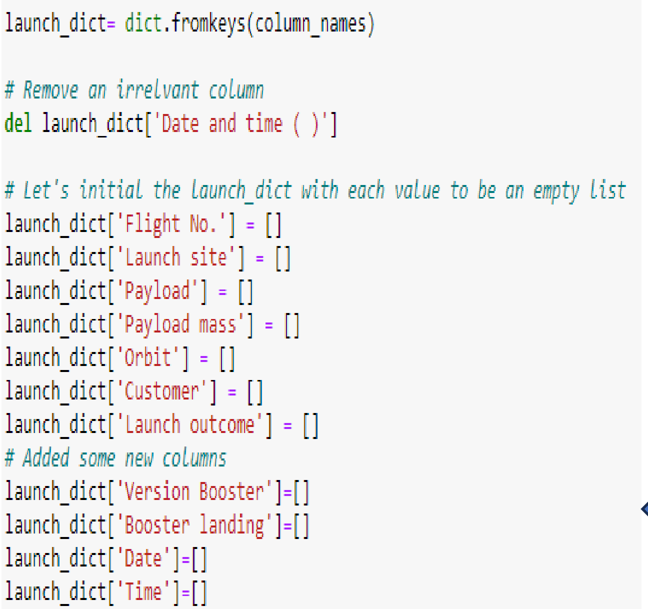
```
html_tables = soup.findAll('table')
```

4. Get column names



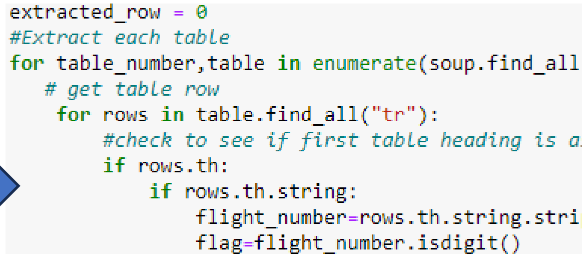
```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary



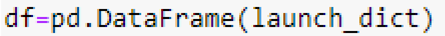
```
launch_dict= dict.fromkeys(column_names)

# 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



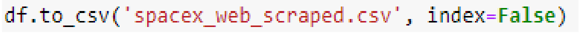
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
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

7. Create dataframe from dictionary



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

8. Export to file

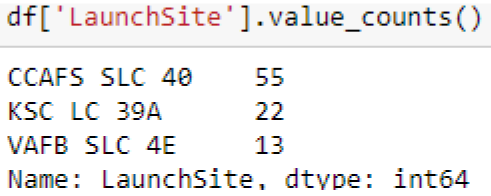


```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
- 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 transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.

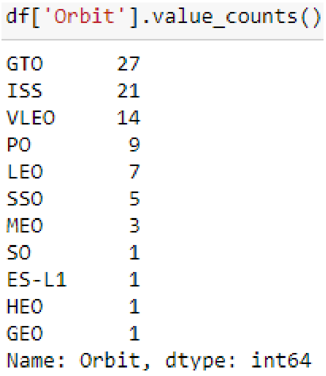
1. Calculate launches n umber for each site



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

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

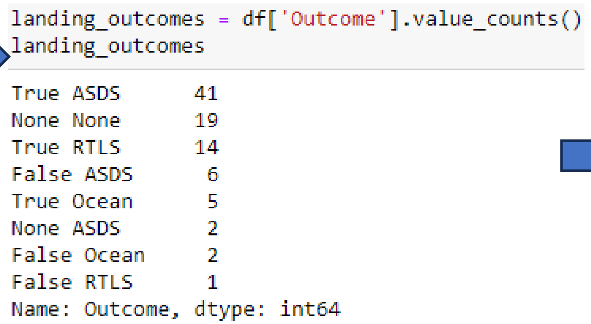
2. Calculate the number and occurence of each orbit



```
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
```

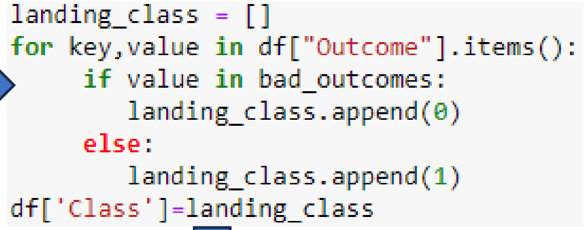
3. Calculate number and occurrence of mission outcome per orbit type



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

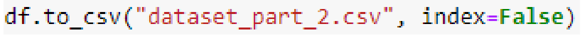
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

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)
```

EDA with Data Visualization

Scatter Graphs

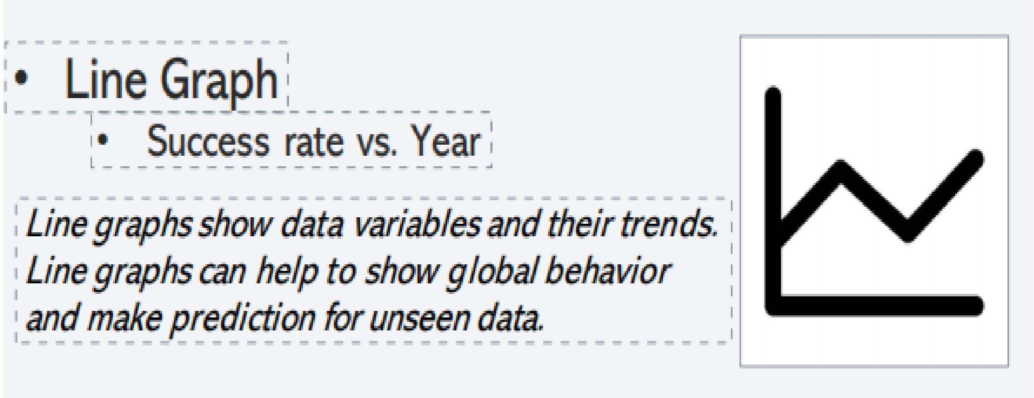
1. Flight Number vs. Payload Mass
2. Flight Number vs. Launch Site
3. Payload vs. Launch Site
4. Orbit vs. Flight Number
5. Payload vs. Orbit Type
6. Orbit vs. Payload Mass

Scatter plots show relationship between variables. This relationship is called the correlation.

Bar Graph

1. Success rate vs Orbit

Bar graphs show the relationship between numeric and categoric variables.



- Line Graph
    - Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.

EDA with SQL

We performed SQL queries to gather and understand data from dataset:

- Displaying the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass
- List the records which will display the month names, failure landing_ouutcome in drone ship, booster versions, launch_site for the

months in year2015.

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

Build an Interactive Map with Folium

Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

- Red circle at NASA Johnson Space Center's coordinate with label showing its name *(folium.Circle, folium.map.Marker)*.
- Red circles at each launch site coordinates with label showing launch site name *(folium.Circle, folium.map.Marker,*
- *folium.features.DivIcon).*
- The grouping of points in a cluster to display multiple and different information for the same coordinates

(folium.plugins.MarkerCluster).

- Markers to show successful and unsuccessful landings. G reen for successful landing and R ed for unsuccessful landing.

(folium.map.Marker,folium.Icon).

- Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them.

(folium.map.Marker,folium.PolyLine,folium.features.DivIcon)

- These objects are created in order to understand better the problem and the data. We can show easily

all launch sites, their surroundings and the number of successful and unsuccessful landings.

Predictive Analysis

Data preparation

1. Load dataset
2. Normalize data
3. Split data into training and test sets.

Model preparation

1. Selection of machine learning algorithms
2. Set parameters for each algorithm to GridSearchCV
3. Training Grid Search Model models with training dataset

Model evaluation

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

Model comparison

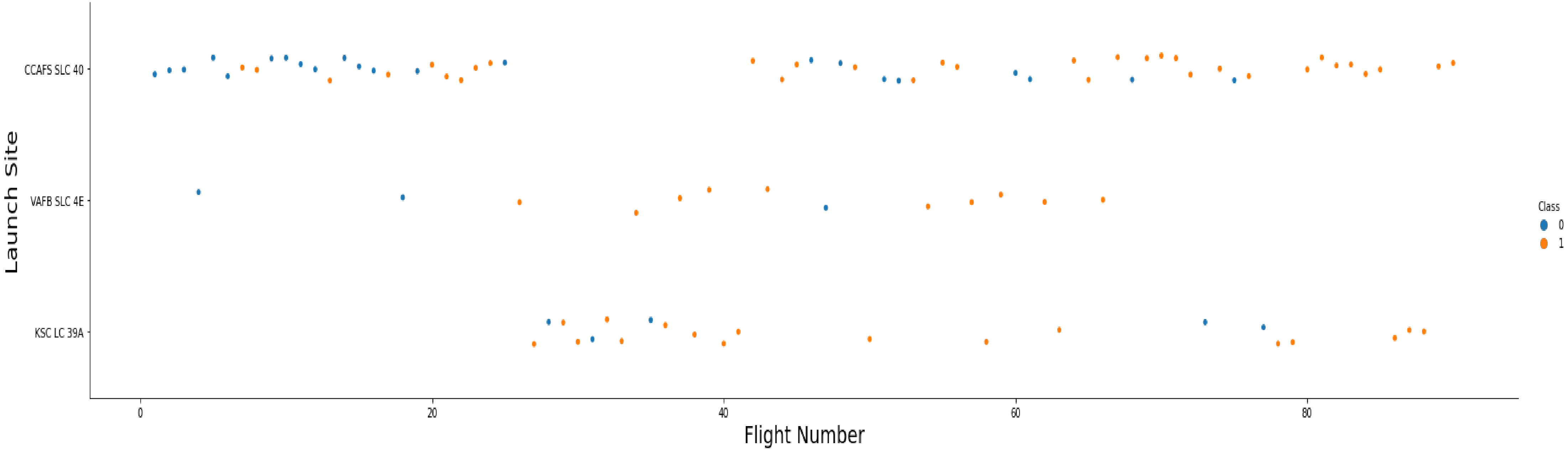
1. Comparison of models according to their accuracy
2. The model with the best accuracy will be chosen (see Notebook for result)

Results

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

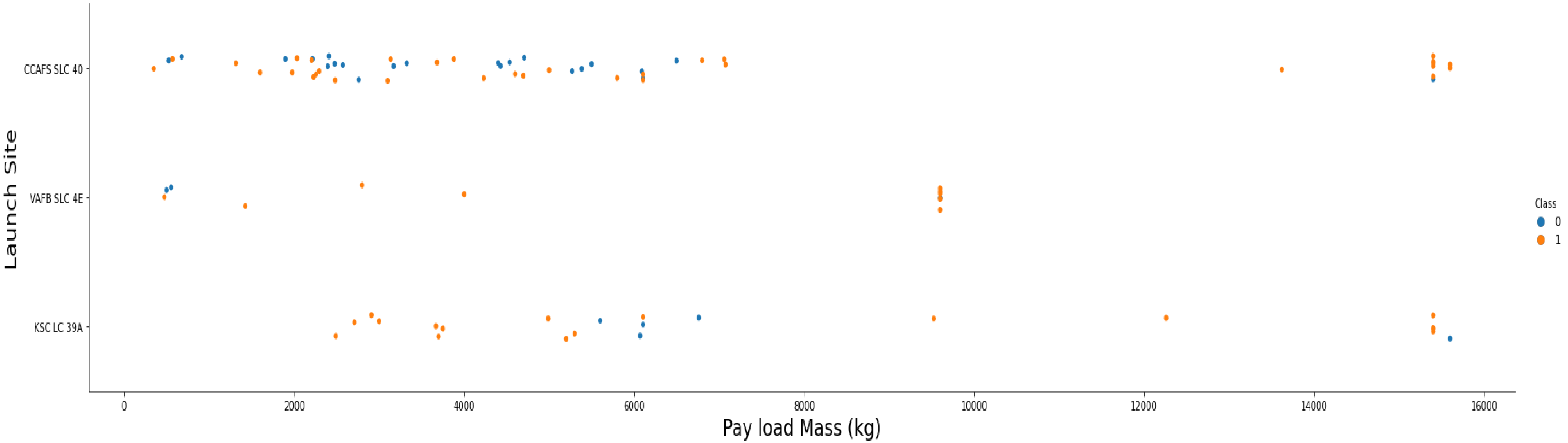
Insights drawn from EDA

Flight Number vs. Launch Site



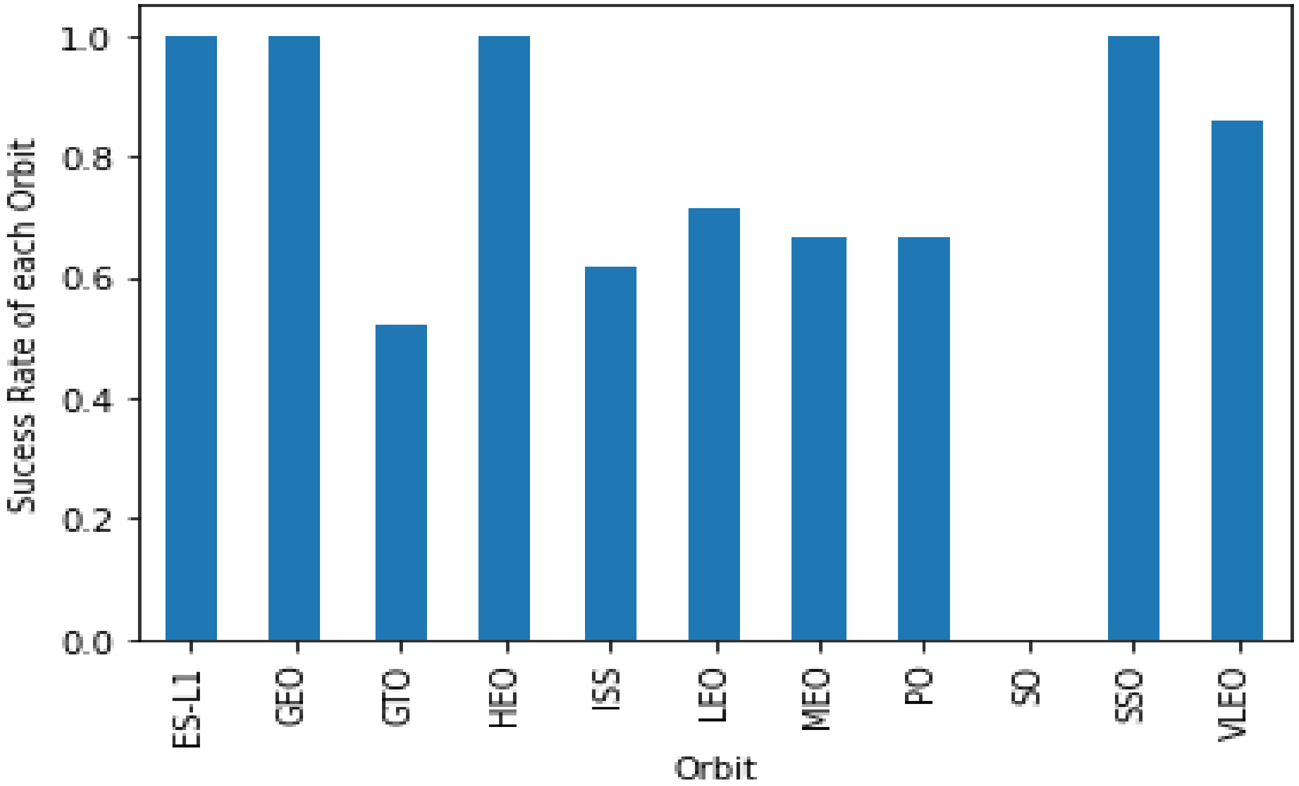
We observe that, for each site, the success rate is increasing.

Payload vs. Launch Site



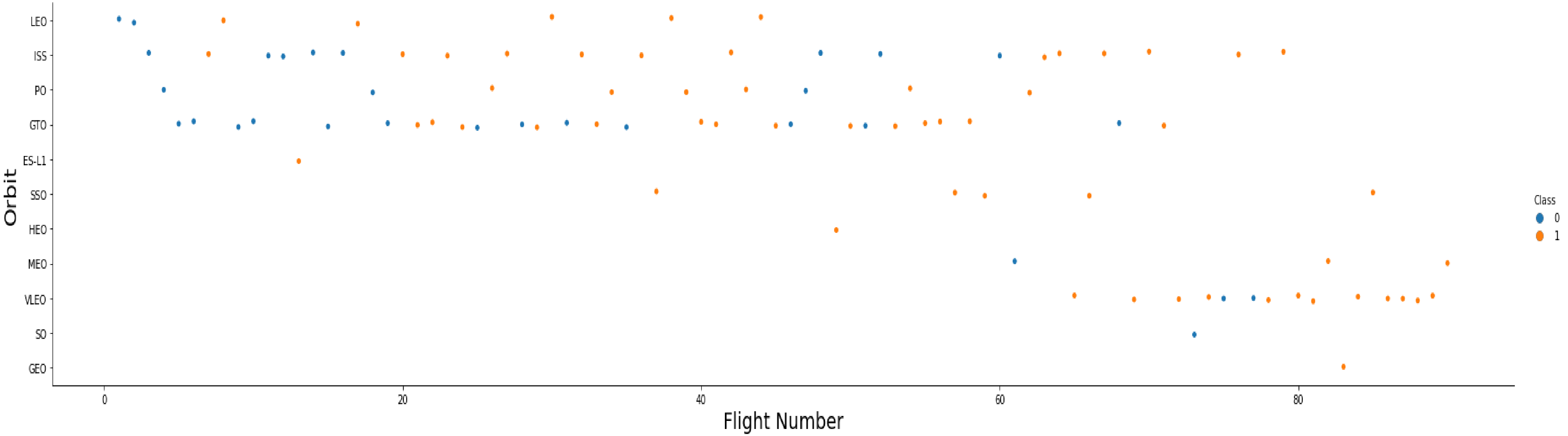
Depending on the launch site, a heavier payload may be a consideration for a successful landing.On the other hand a too heavy payload can make a landing fail.

Success Rate vs. Orbit Type



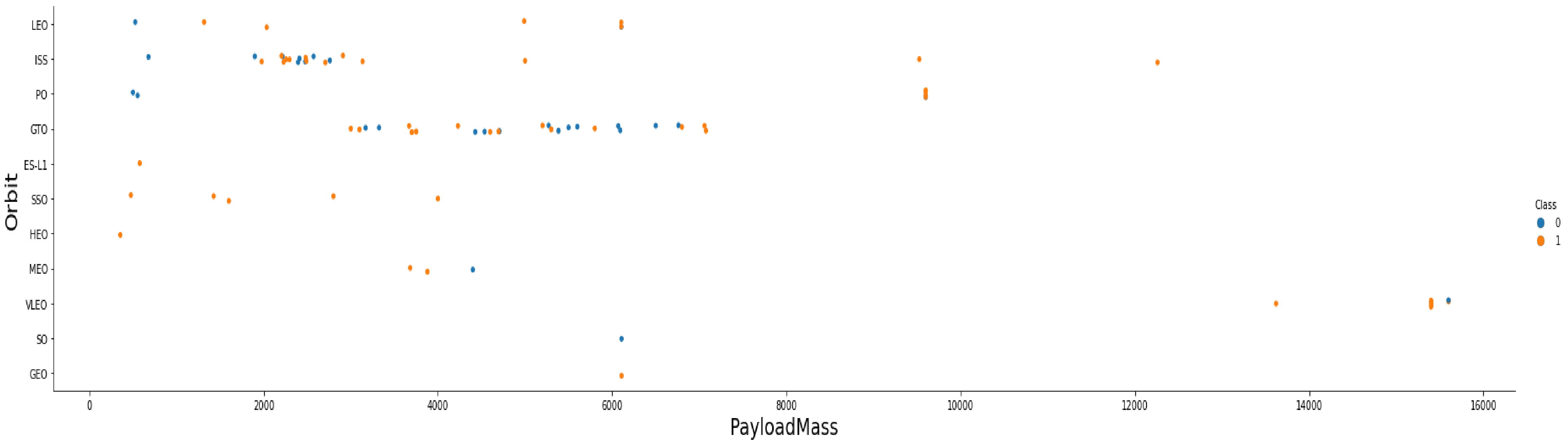
With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



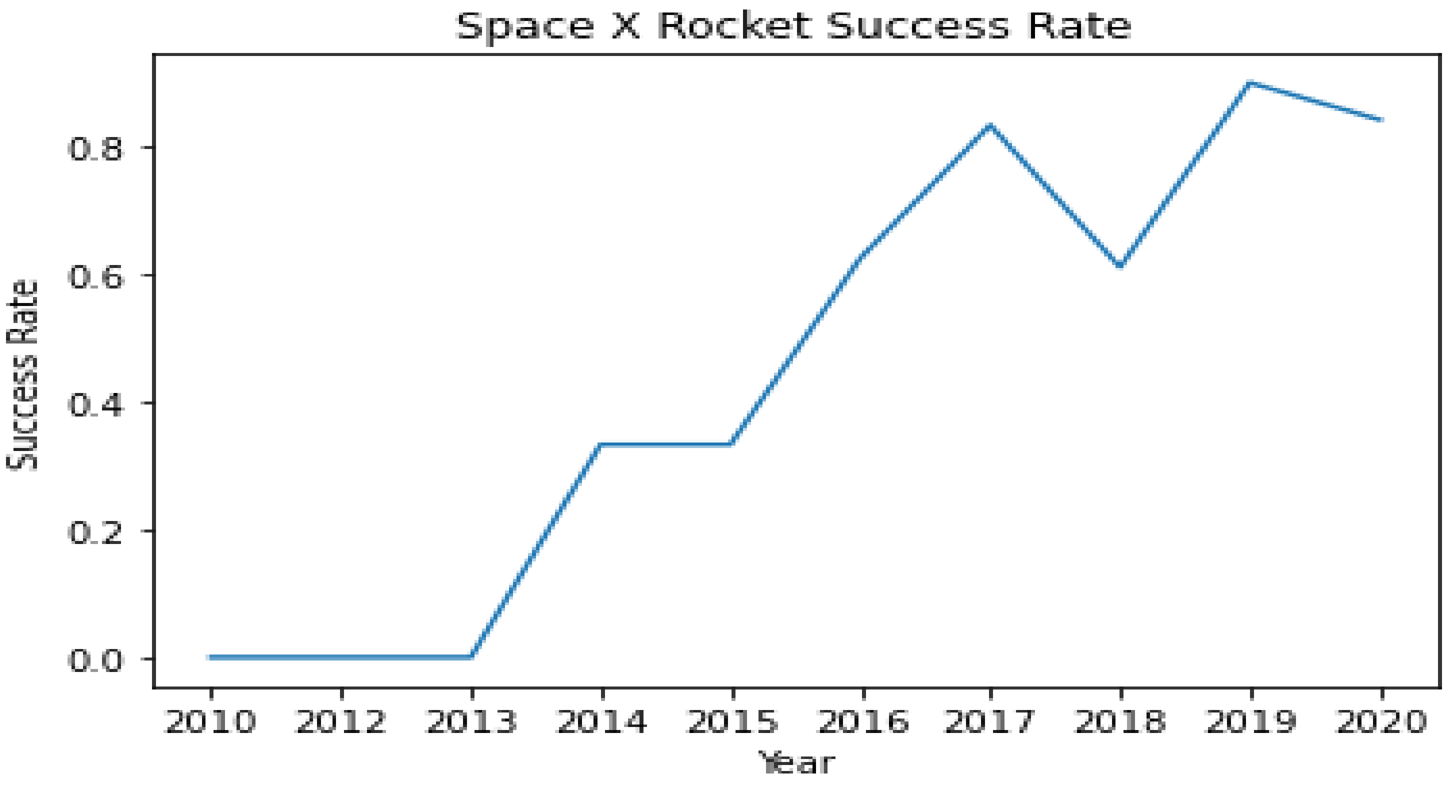
We notice that the success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of f lights. But we can suppose that the high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.

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 improve the success rate for the LEO orbit. Another finding is that Decreasing the payload weight for a GTO orbit improves the success of a launch.

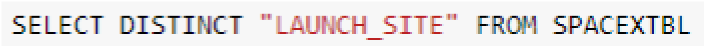
Launch Success Yearly Trend



Since 2013,we can see an increase in the Space X Rocket success rate.

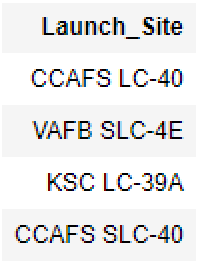
All launch site names

SQL Query



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

Results



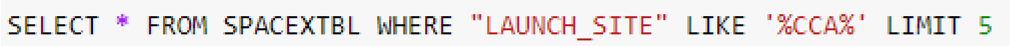
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Explanation

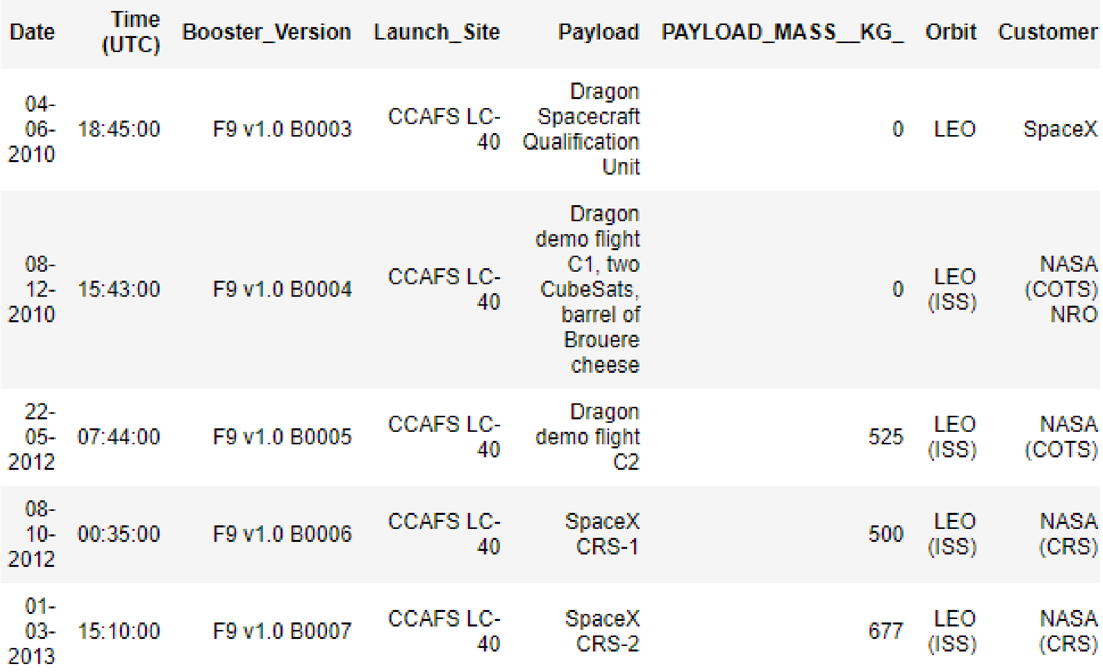
The use of DISTINCT in the query allows to remove. Duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SQL Query



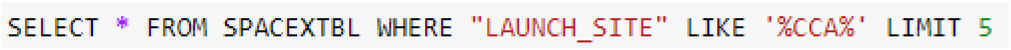
```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```



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)

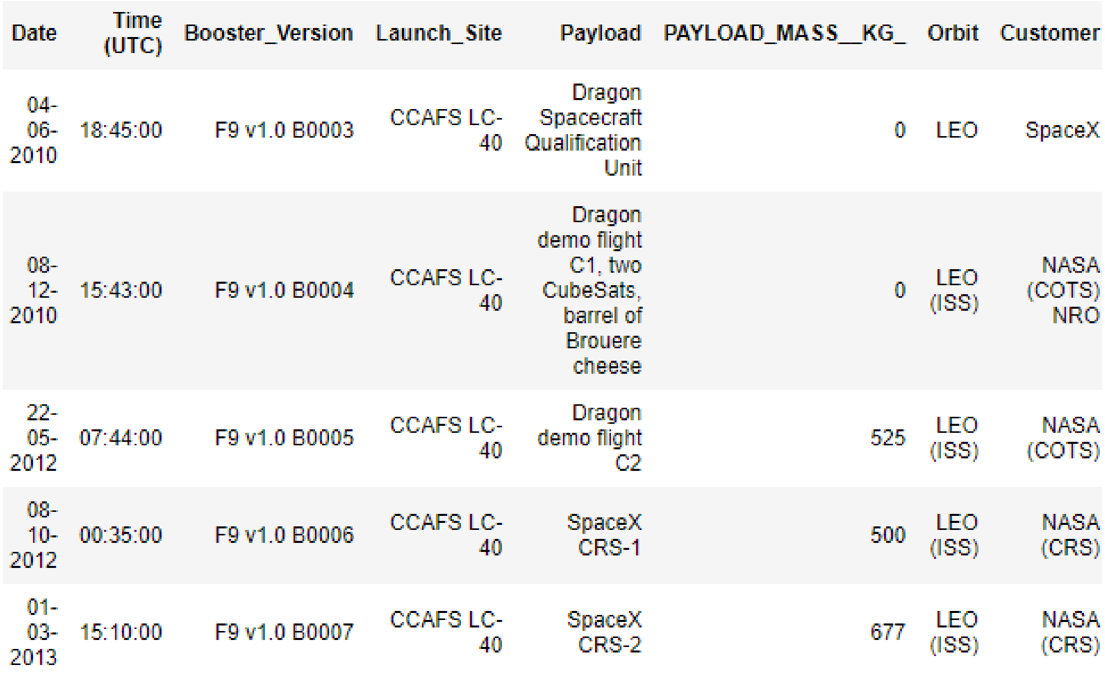
Launch Site Names Begin with 'CCA'

SQL Query



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

Results



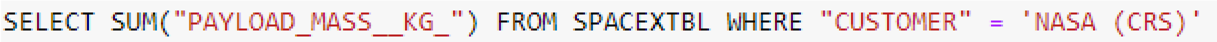
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	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)

Explanation

The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows5 records from filtering.

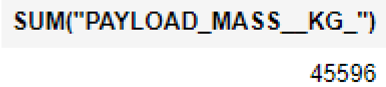
Total Payload Mass

SQL Query



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

Results



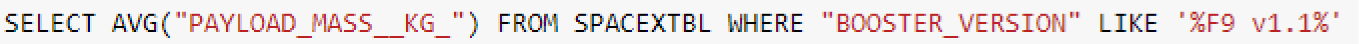
SUM("PAYLOAD_MASS__KG_")
45596

Explanation

This query returns the sum of all payload masses where the customer is NASA (CRS).

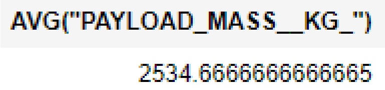
Average Payload Mass by F9 v1.1

SQL Query



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

Results



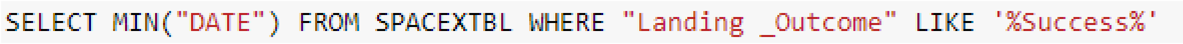
AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

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

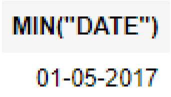
First Successful Ground Landing Date

SQL Query



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

Results



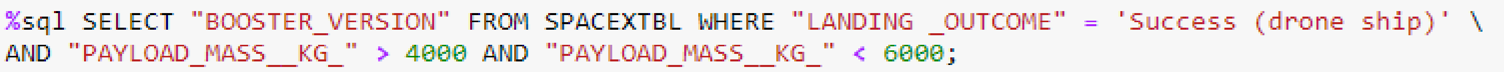
MIN("DATE")
01-05-2017

Explanation

With this query, we select the oldest successful landing. The WHERE clause filters data set in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

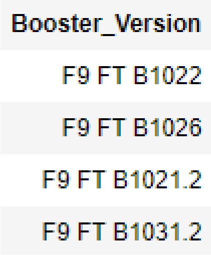
Successful Drone Ship Landing with Payload between 4000 and 6000

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



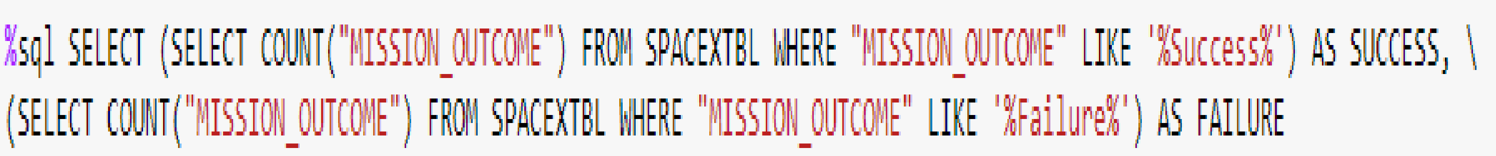
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 k g. The WHERE and AND clauses filter the dataset.

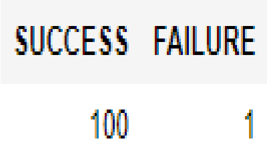
Total Number of Successful and Failure Mission Outcomes

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



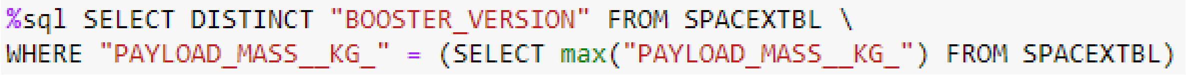
SUCCESS	FAILURE
100	1

Explanation

With the first SELECT, we show the subqueries that return results. The first subquery counts the successful Mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

SQL Query

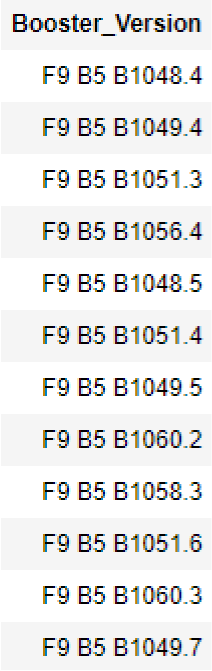


```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Explanation

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

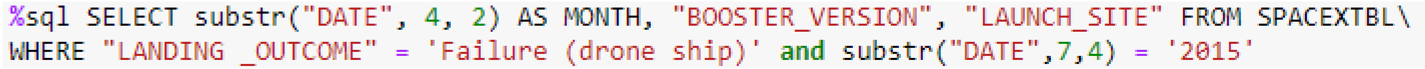
Results



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

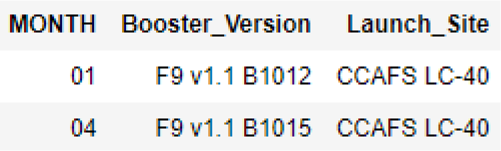
2015 Launch Records

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



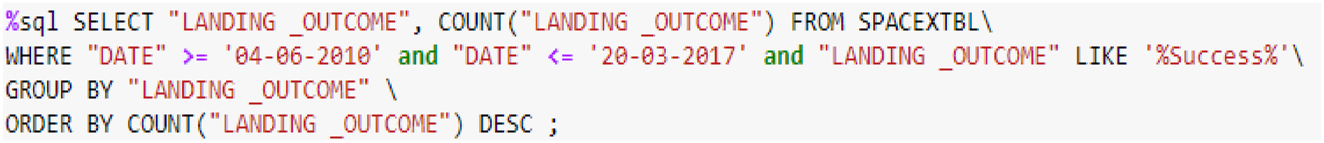
MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

The query provides information on the month, booster version, and launch site for unsuccessful landings that occurred in 2015. The Substring function is used to extract either the month or the year from the date. Specifically, Substring(DATE, 4, 2) extracts the month, and Substring(DATE, 7, 4) extracts the year.

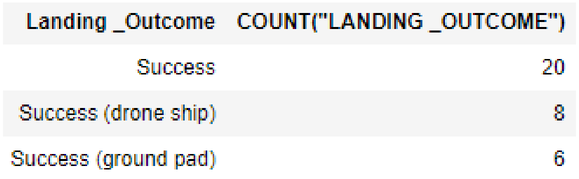
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

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 ;
```

Result



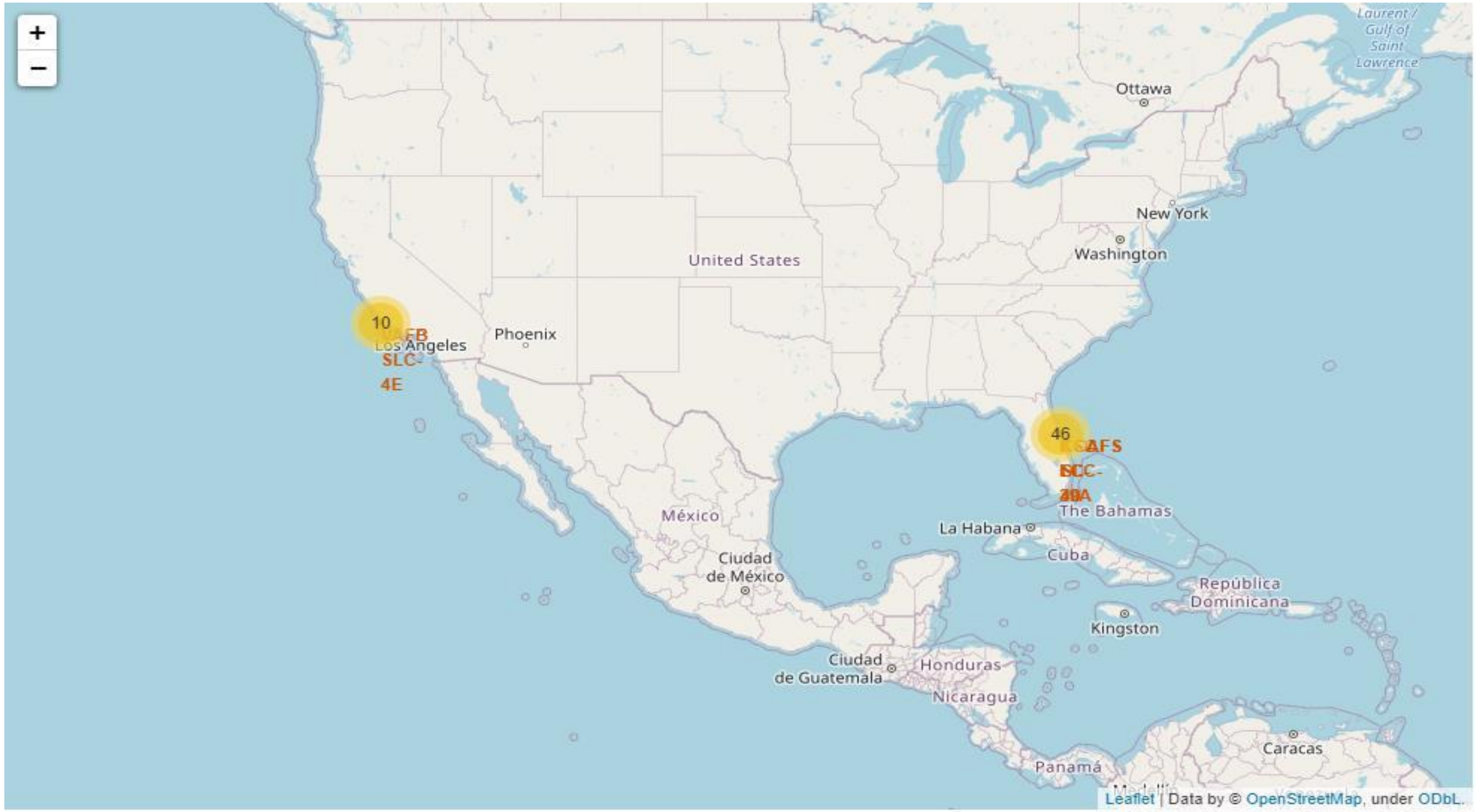
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Explanation

The query retrieves landing outcomes and their respective counts for missions that were successful, with dates falling between 04/06/2010 and 20/03/2017. The GROUP BY clause organizes the results based on landing outcomes, and the ORDER BY COUNT DESC arranges the results in descending order of counts

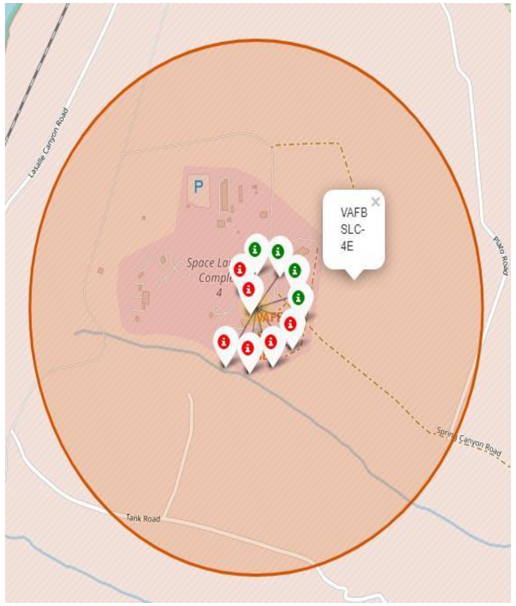
Launch sites Proximities Analysis

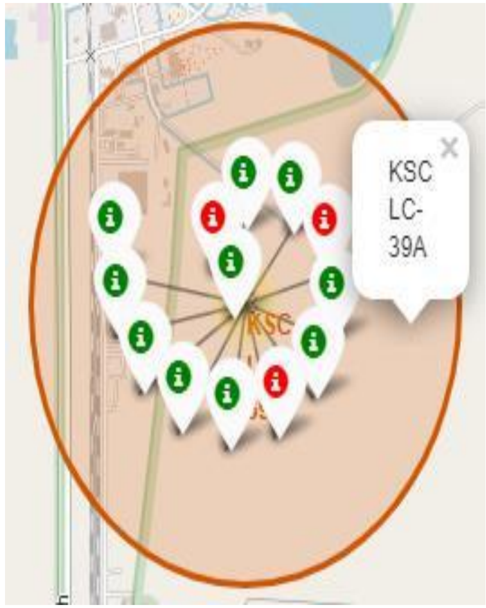
Folium map – Ground Stations

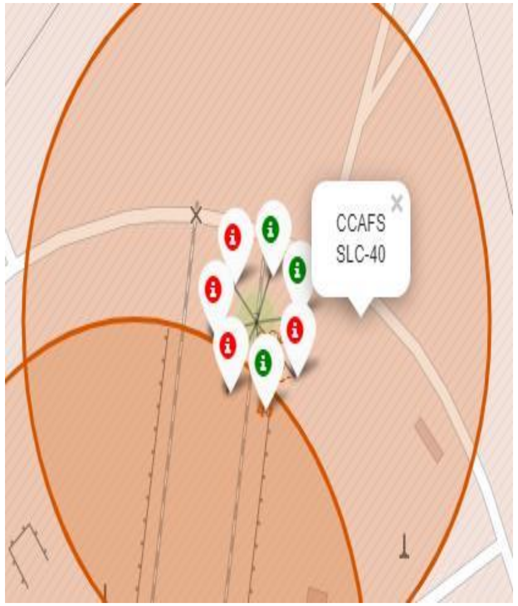


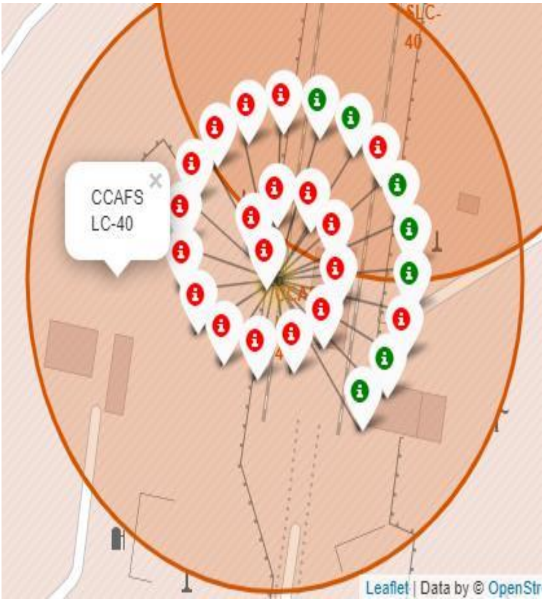
SpaceX launch site is located on the coast of the United States

Folium map – Color Labeled Markers



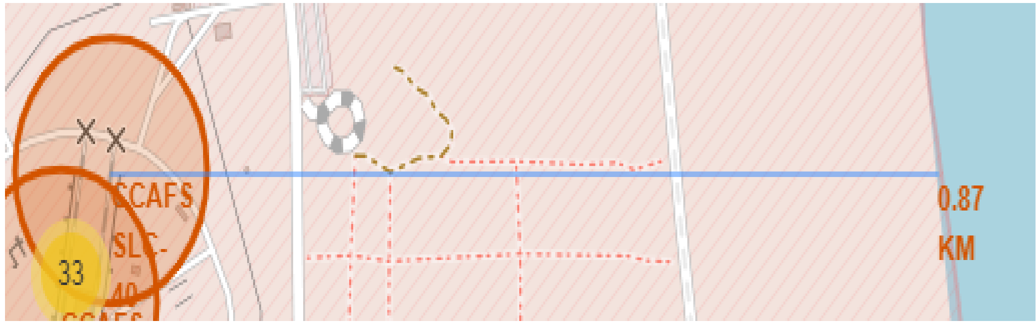


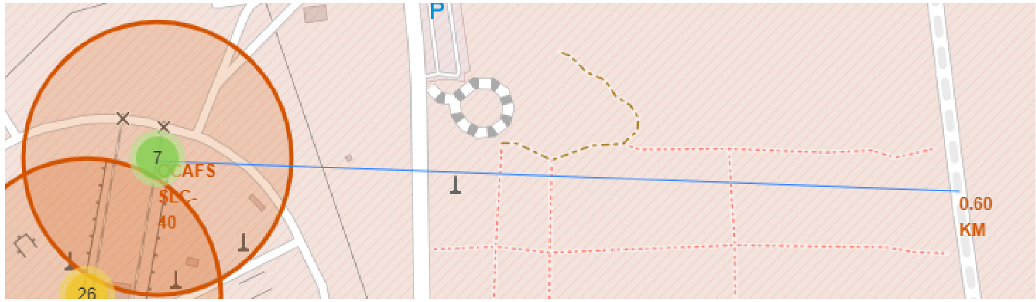


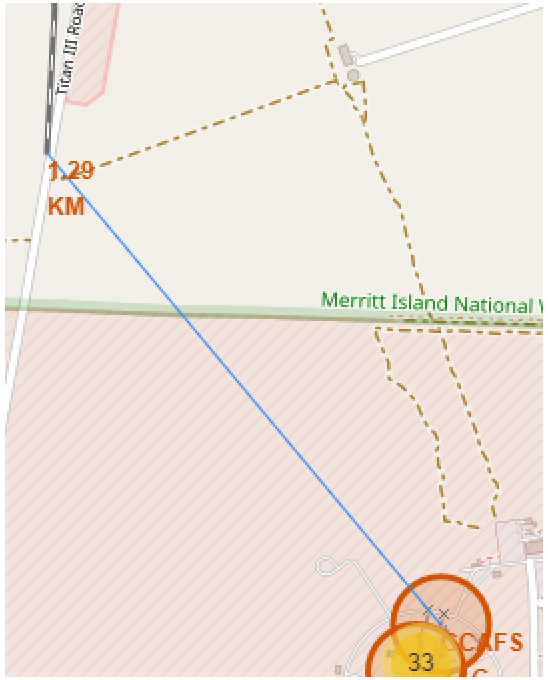


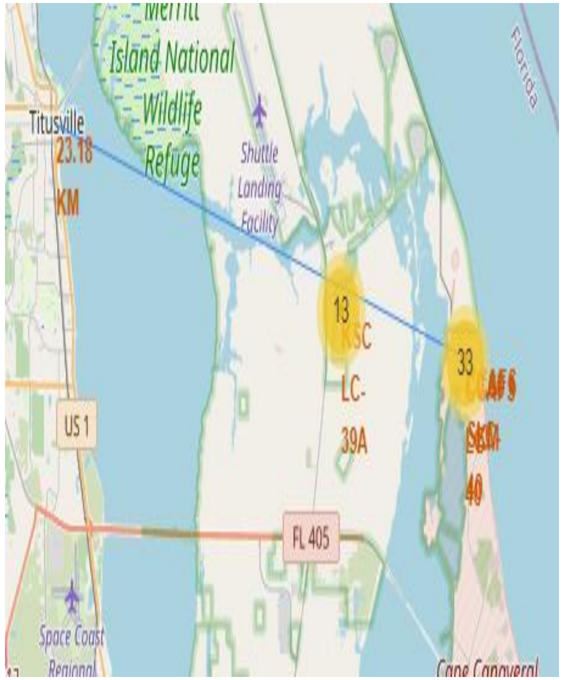
Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

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



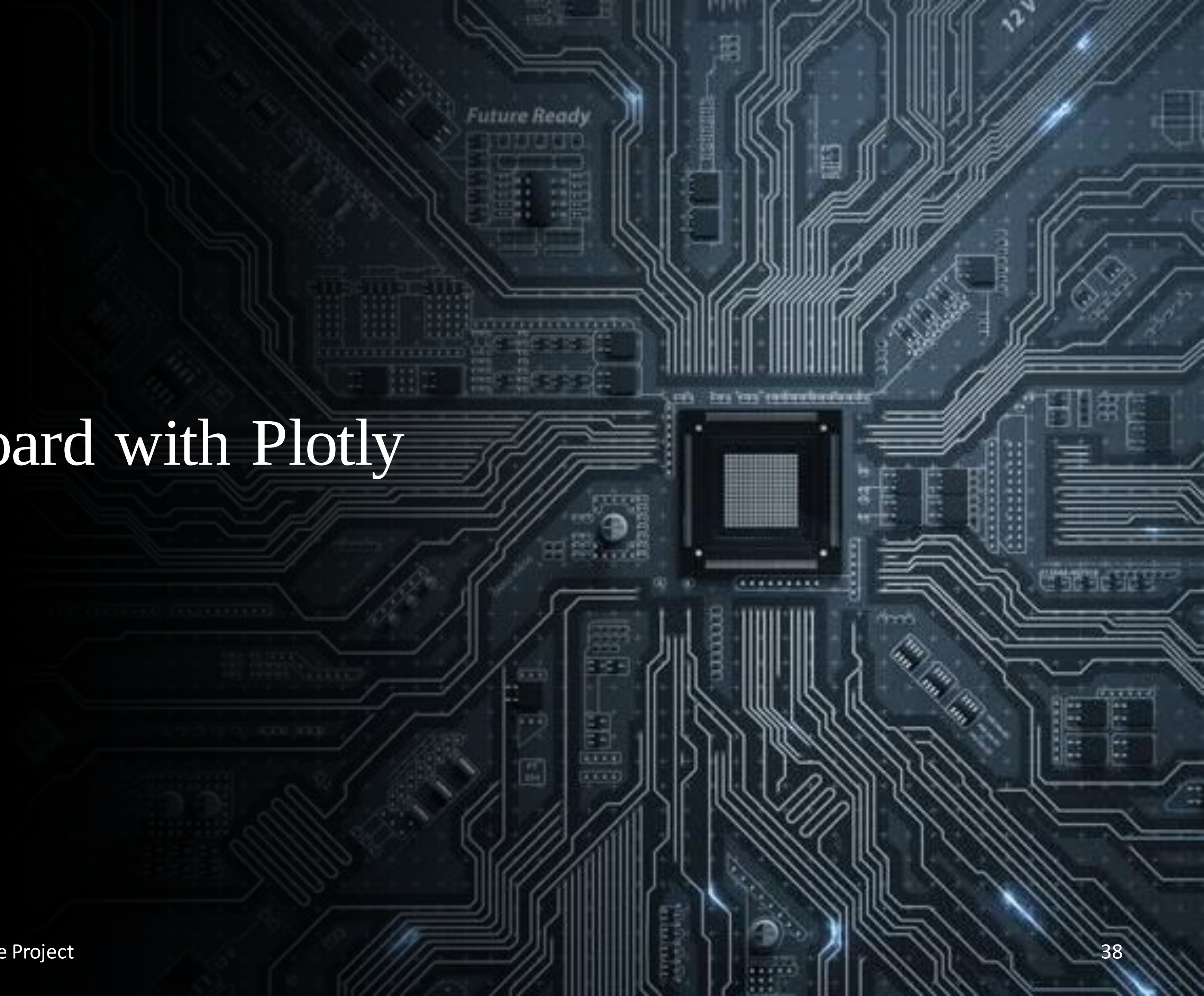




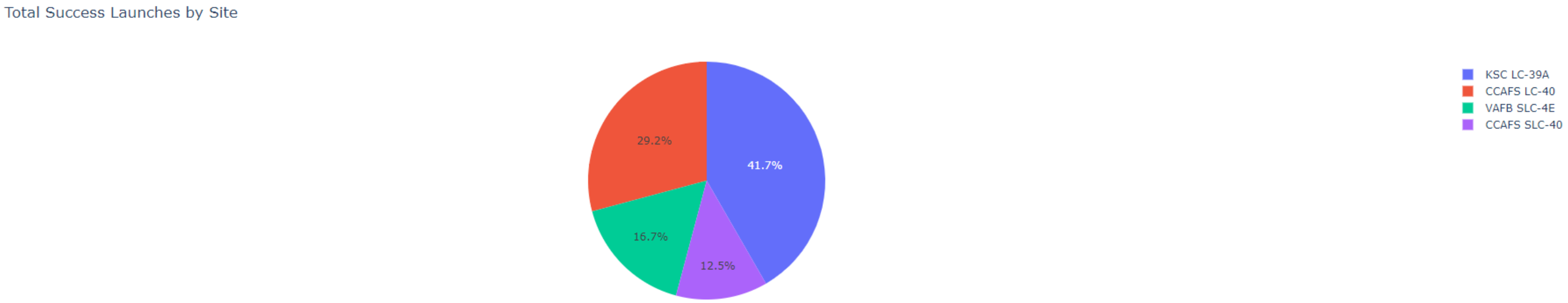


Is CCAFS SLC-40in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

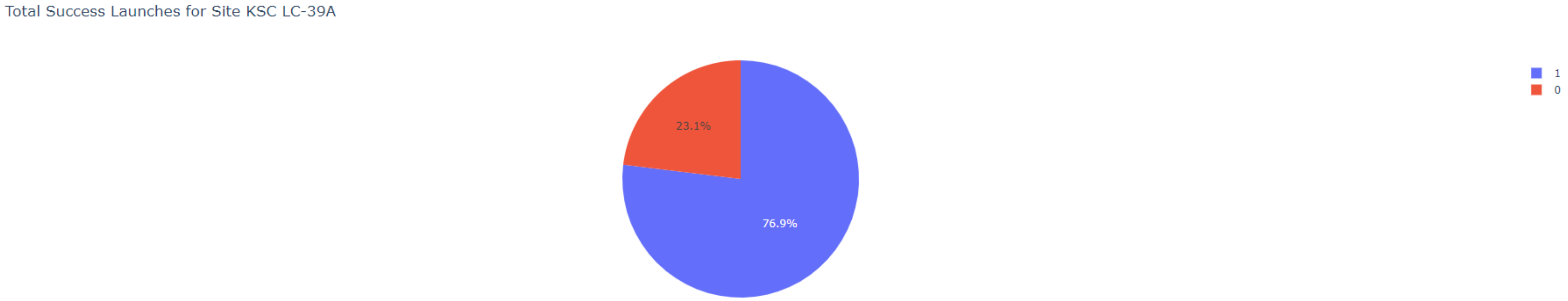
Building a Dashboard with Plotly and Dash



Dashboard – Total Success by Site



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

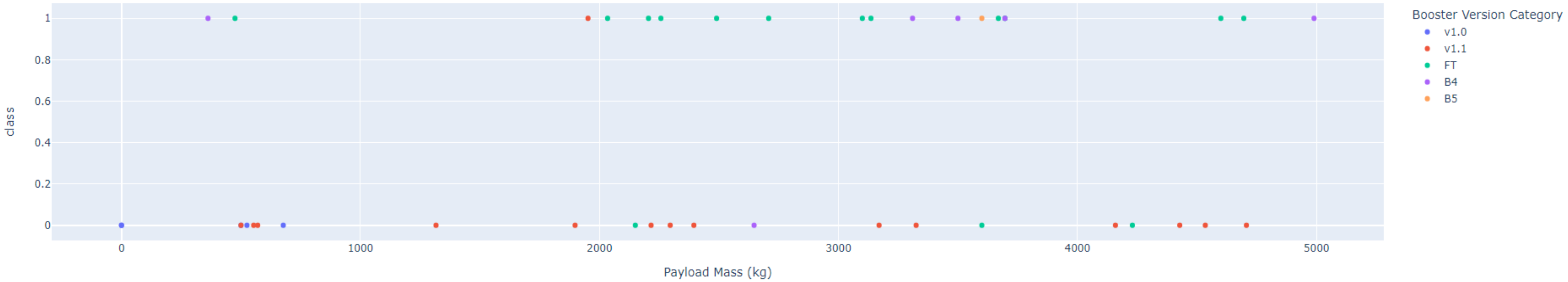


Total Success Launches for Site KSC LC-39A

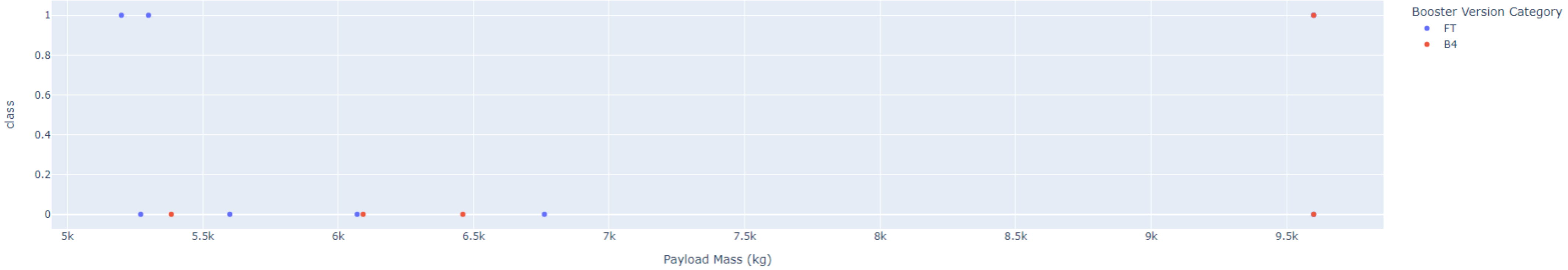
Dashboard – Payload mass vs Outcome for all sites with different payload mass selected

Low Weight Payload and High Weight Payload

Correlation between Payload and Success for all Sites



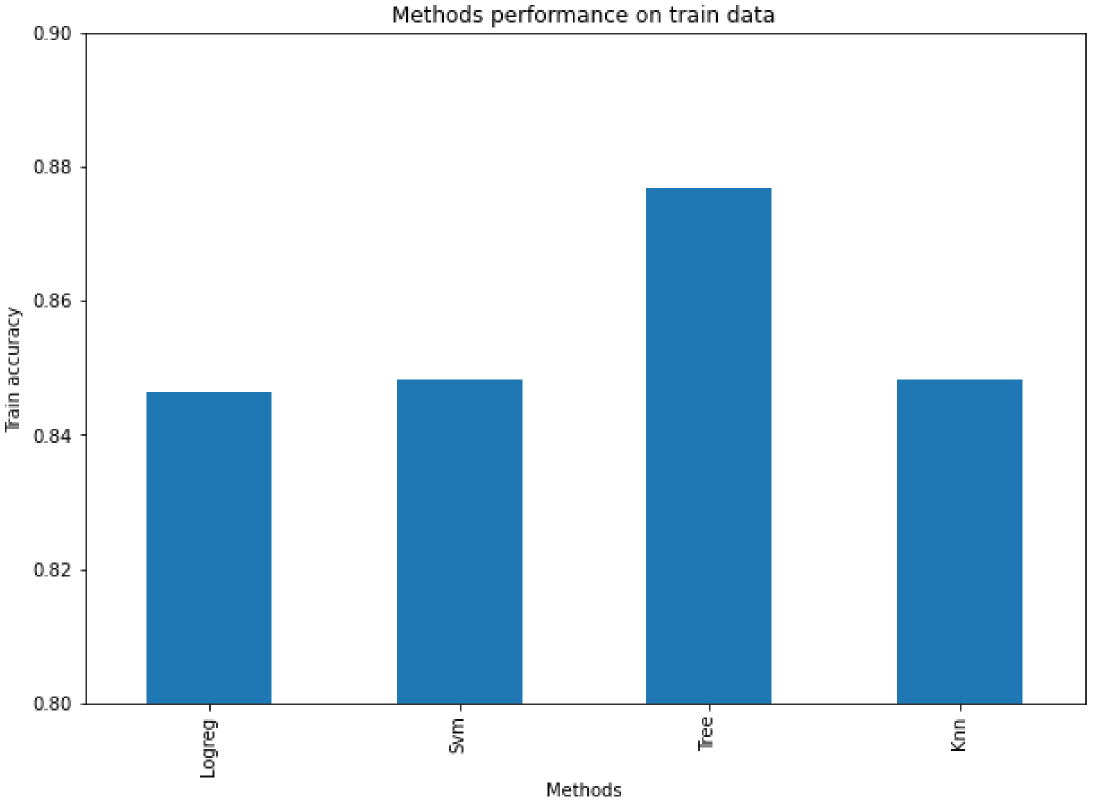
Correlation between Payload and Success for all Sites

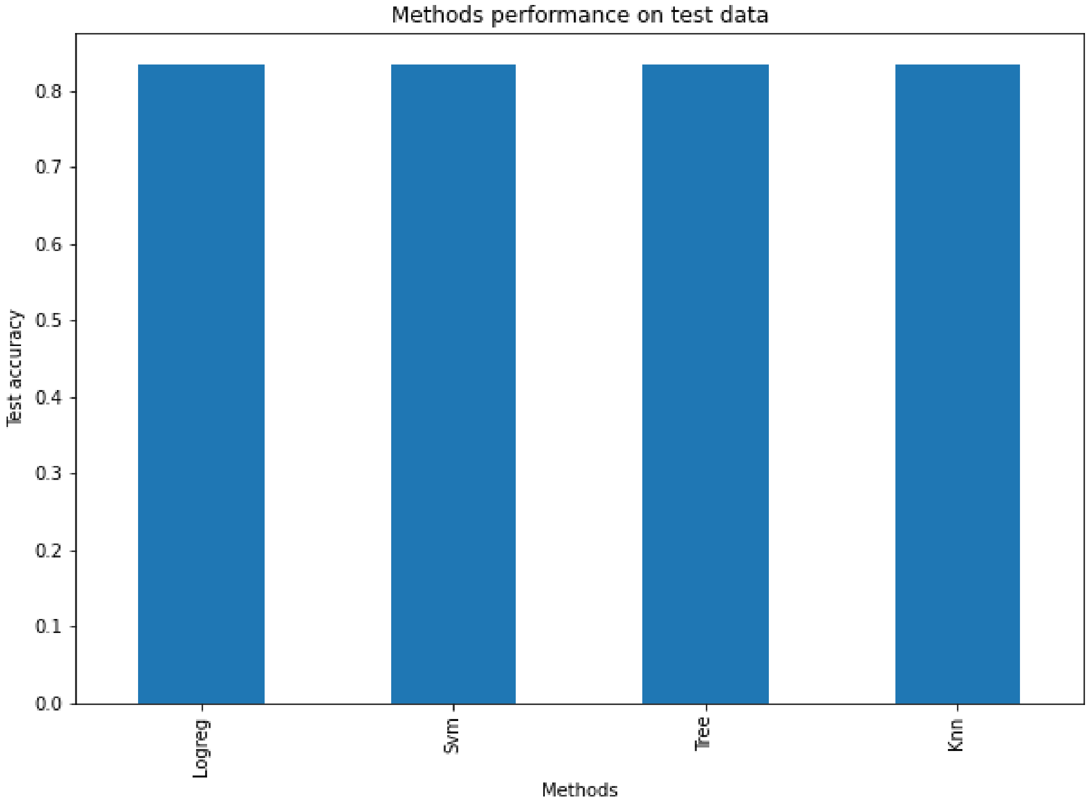


Predictive Analysis

Classification Accuracy

All methods exhibited similar performance in the accuracy test. To make a conclusive decision between them, obtaining additional test data could be beneficial. However, if an immediate choice is necessary, the decision would lean towards selecting the decision tree method.

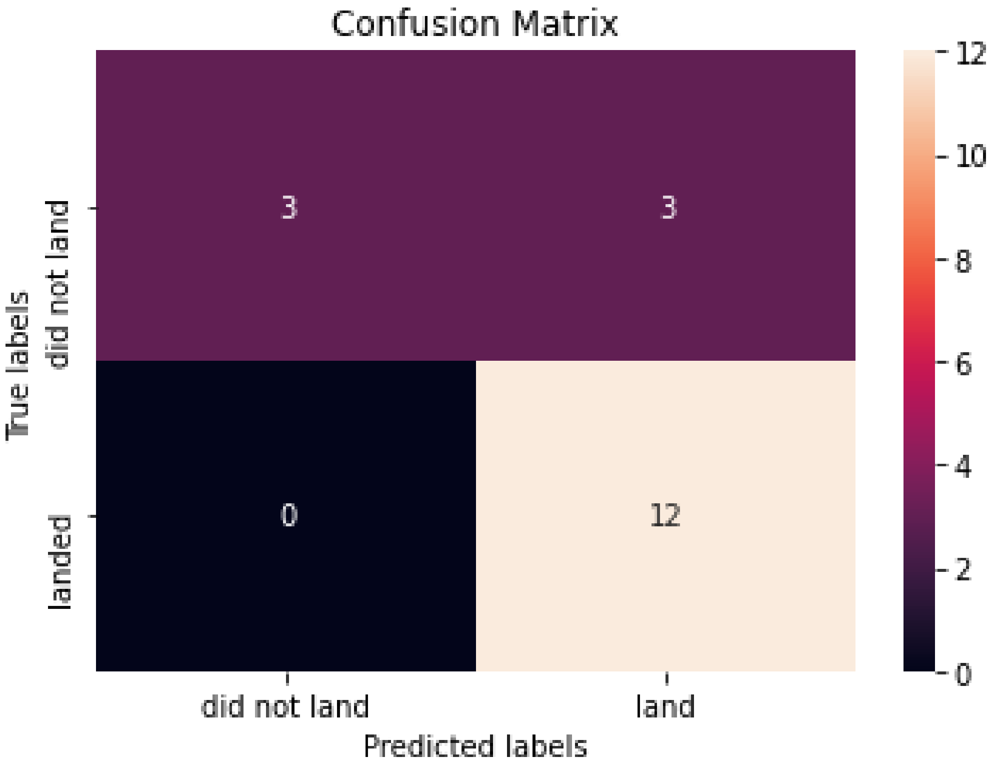




Confusion Matrix

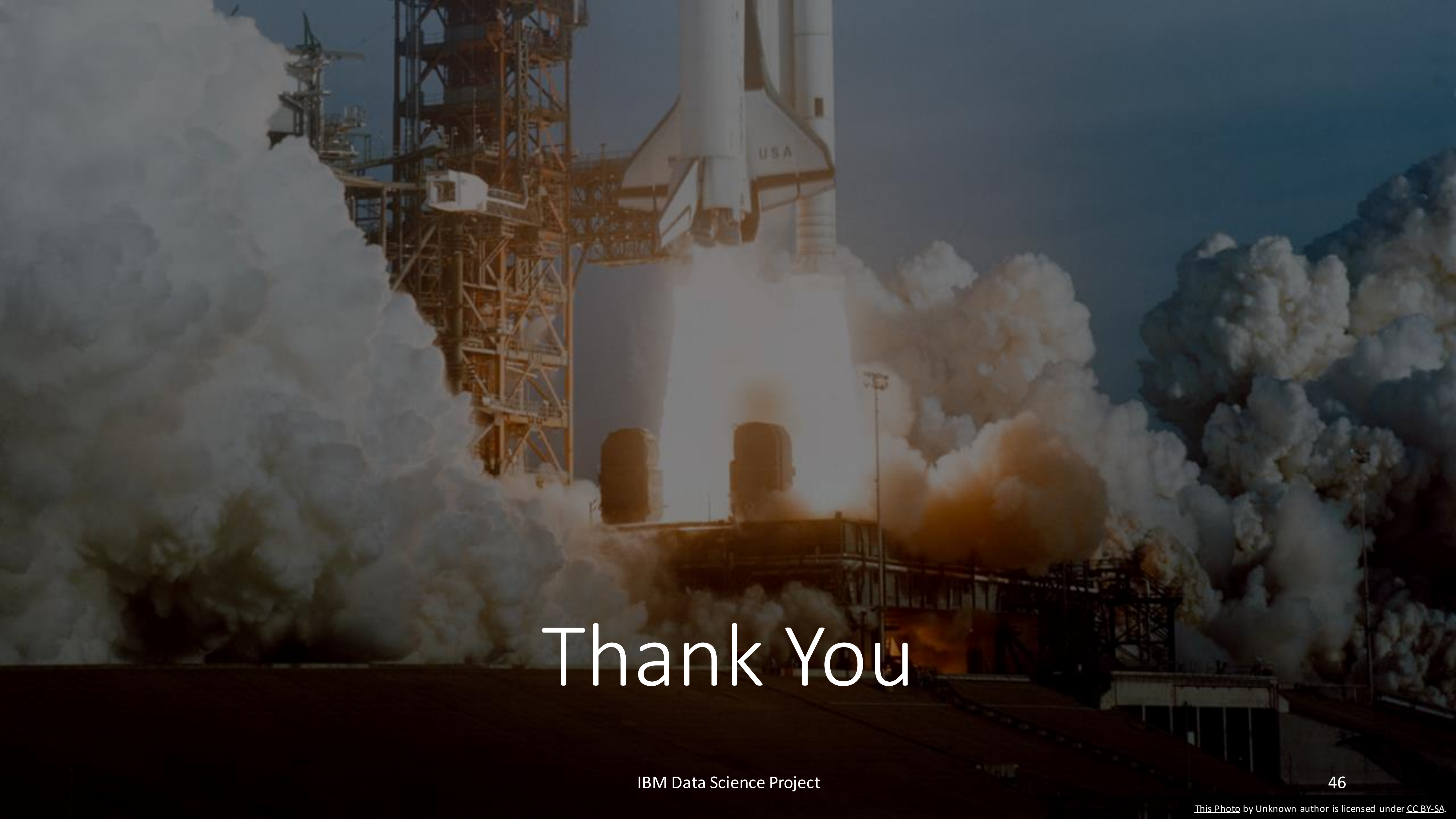
Since the test accuracies are equal, the confusion matrices are also identical among these models. The primary issue observed in these models pertains to false positives.

Confusion Matrix



Conclusion

- The success of a mission can be attributed to various factors, including the launch site, the orbit, and, notably, the number of previous launches. It can be assumed that there has been an accumulation of knowledge between launches, enabling a transition from launch failures to successful missions.
- For the success of a mission, the payload mass can be a crucial factor depending on the orbits involved. Certain orbits necessitate consideration of either light or heavy payload masses. However, as a general trend, low-weighted payloads typically exhibit better performance compared to their heavier counterparts.
- With the current data, we are unable to elucidate why some launch sites outperform others (with KSC LC-39A identified as the premier launch site). To address this issue, acquiring atmospheric or other pertinent data could provide the answers
- For this dataset, the Decision Tree Algorithm is selected as the preferred model, even though the test accuracy is identical among all the models used. The decision to choose the Decision Tree Algorithm is based on its superior training accuracy.



Thank You