Winning Space Race with Data Science

Santiago González Cruz
August 8th , 2023

Outline

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

Executive Summary

- Summary of methodologies

1. Data Collection using Web Scraping
2. Exploratory Data Analysis with Data Visualization and SQL
3. Interactive Dashboard with Folium
4. Predictive values with Machine Learning

- Summary of all results

1. Exploratory Data Analysis results
2. Interactive analytics in screenshots
3. Predictive analysis using ML

Introduction

- Project background and context

Predicting whether the Falcon 9 first stage will land successfully is the goal of this endeavor. The launch of the Falcon 9 rocket, according to SpaceX, cost 62 million dollars. The price of other providers is around 165 million dollars per one. The fact that SpaceX can reuse the first stage explains the pricing disparity. The price of a launch can be calculated by assessing if the stage will land. If another business plans to compete with SpaceX for a rocket launch, this information may be useful.

- Problems you want to find answers

Determining every element that affects how a landing turns out. The interrelationships between the factors and how they impact the result. The ideal circumstance required to raise the likelihood of a successful landing.

Section 1

Methodology

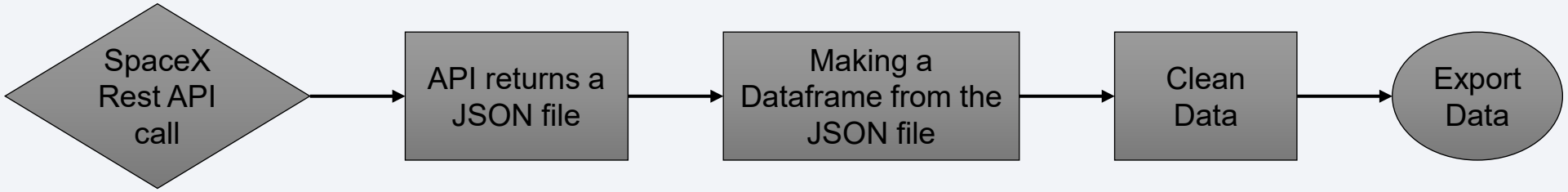
Methodology

Executive Summary

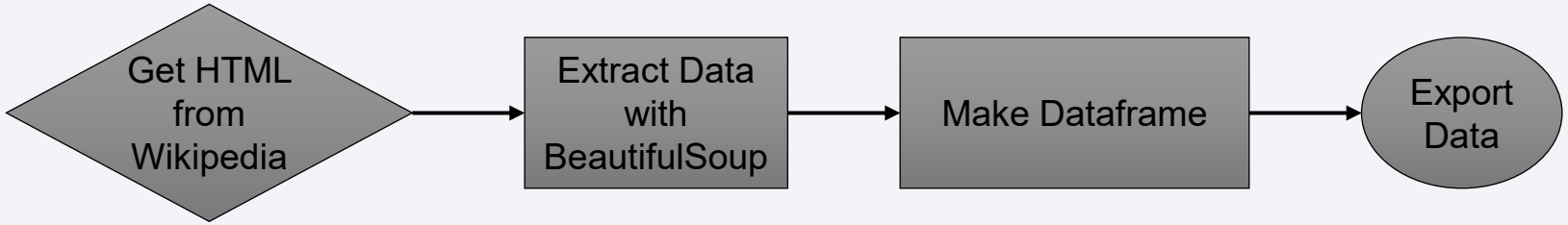
- Data collection methodology:
  - Utilizing the SpaceX REST API and web scraping from Wikipedia, data was gathered.
- Perform data wrangling
  - Dropping unnecessary columns and using One Hot Encoding for classification models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models.

Data Collection

- Utilizing the SpaceX REST API and web scraping from Wikipedia, data was gathered.



- Information about launches, landings, and payloads is collected through webscraping Wikipedia.



Data Collection – SpaceX API

1. Getting Response from API and Convert it to a JSON file

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

response = requests.get(spacex_url)

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

2. Transform Data

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

3. Create a dictionary

```
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}
```

4. Create and filter the dataframe

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

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

5. Export to a file

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

Link to code

Data Collection - Scraping

1. Getting Response from HTML and create object with BeatifulSoup.

```
data  = requests.get(static_url)

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

2. Find All tables and get its names

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


for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```

3. Create a dictionary

```
# 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']=[]
```

4. Add data to keys

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

5. Create Dataframe from dictionary

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

6. Export to a file

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

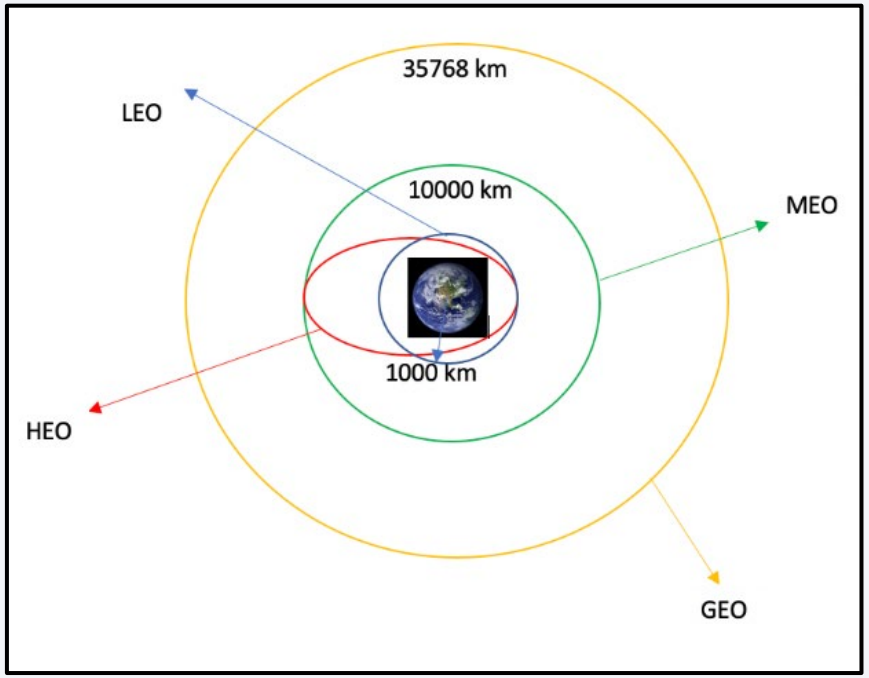
Link to code

Data Wrangling

Exploratory Data Analysis (EDA) requires the cleaning and unification of chaotic, complex data sets. This process is known as data wrangling.

The number of launches at each site will be determined first, followed by the number and frequency of mission outcomes for each type of orbit.

The outcome column is then used to build a landing outcome label. Further analysis, visualization, and machine learning will be made simpler as a result. The output will then be exported to a CSV file.

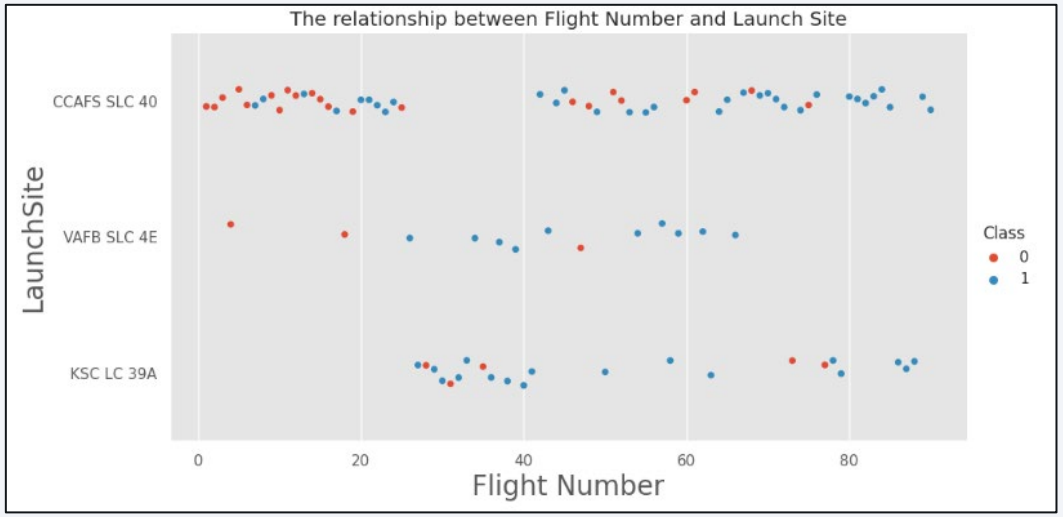


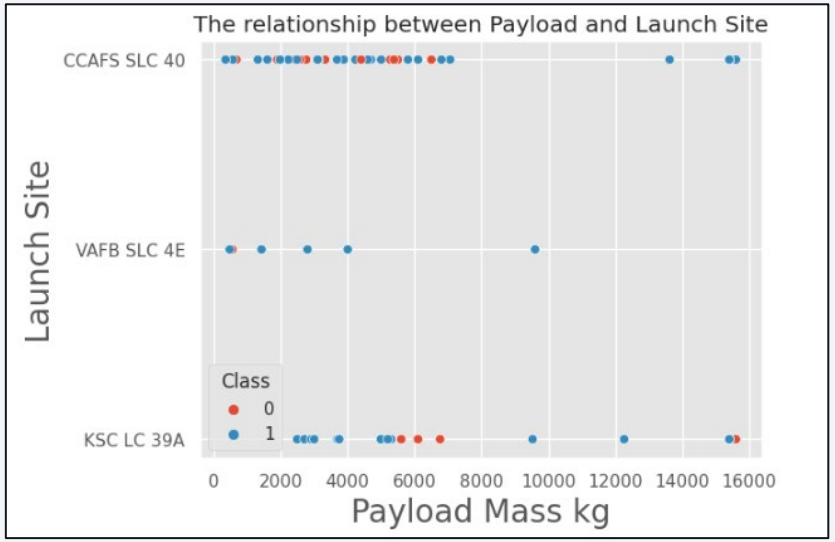
Link to code

EDA with Data Visualization

To determine the association between the qualities, we first used a scatter graph to compare:

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass





Scatter plots demonstrate the interdependence of qualities. Following the identification of a pattern in the graphs. It is relatively simple to identify the variables that have the most influence on the outcome of the landing.

Link to code

EDA with Data Visualization

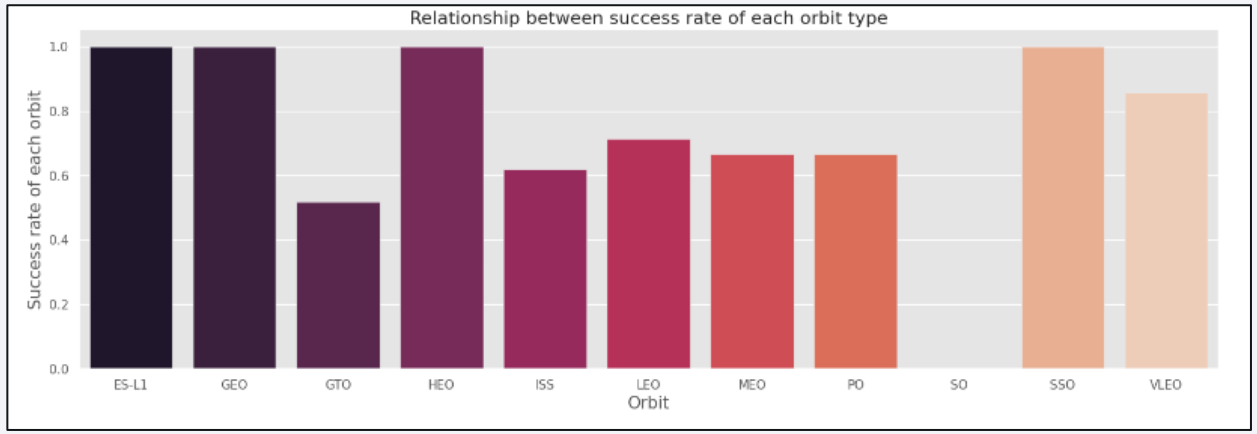
Once the scatter plot gives us an indication about the relationships. For further examination, we will then employ additional visualization tools like bar graphs and line graphs.

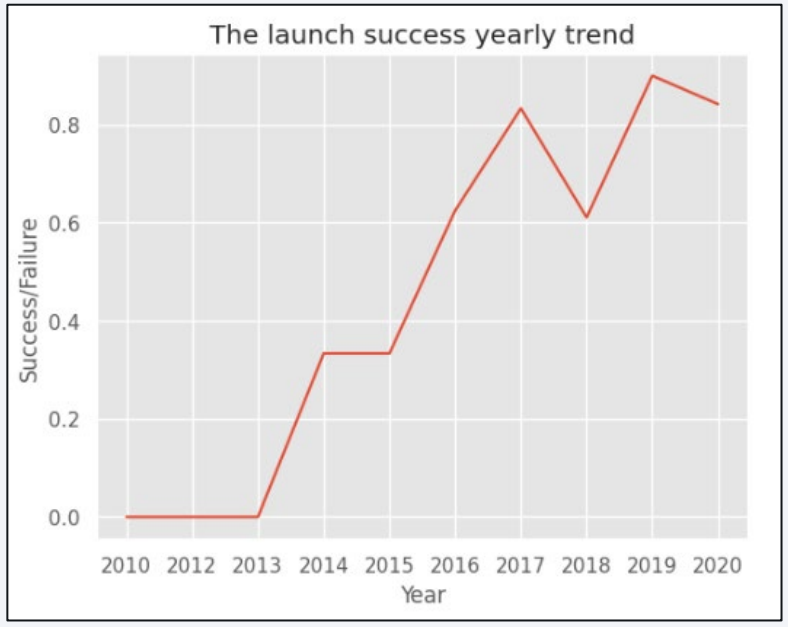
One of the simplest ways to understand how the qualities relate to one another is through bar graphs. The bar graph will be used in this situation to show which orbits have the best chance of succeeding.

We then employ the line graph to demonstrate a pattern or trend of the attribute across time, in this case, the yearly trend of launch success.

Then, by creating dummy variables to correspond to category columns, we employ feature engineering to be used in success prediction in the future module.

Link to code





EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - Displaying the names of the unique lauunch 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, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Link to code

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. Green for successful landing and Red 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)*

Link to code

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Link to code

Predictive Analysis (Classification)

Building the Model

- Load the dataset
- Transform the data and split into training and test datasets
- Decide which type of Machine Learning use
- Set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- Plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

- The model with the best accuracy score will be the best.

Link to code

Results

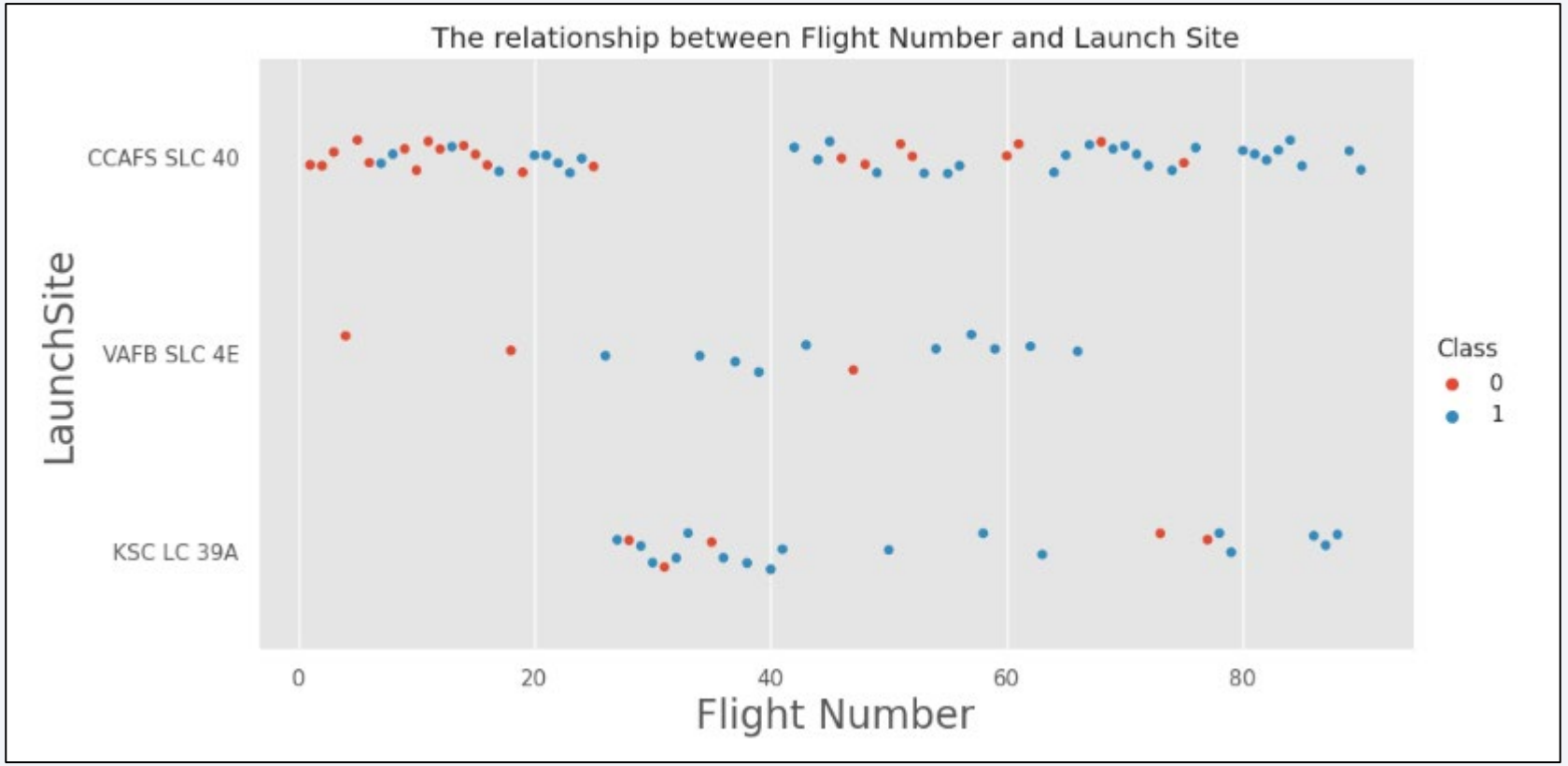
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

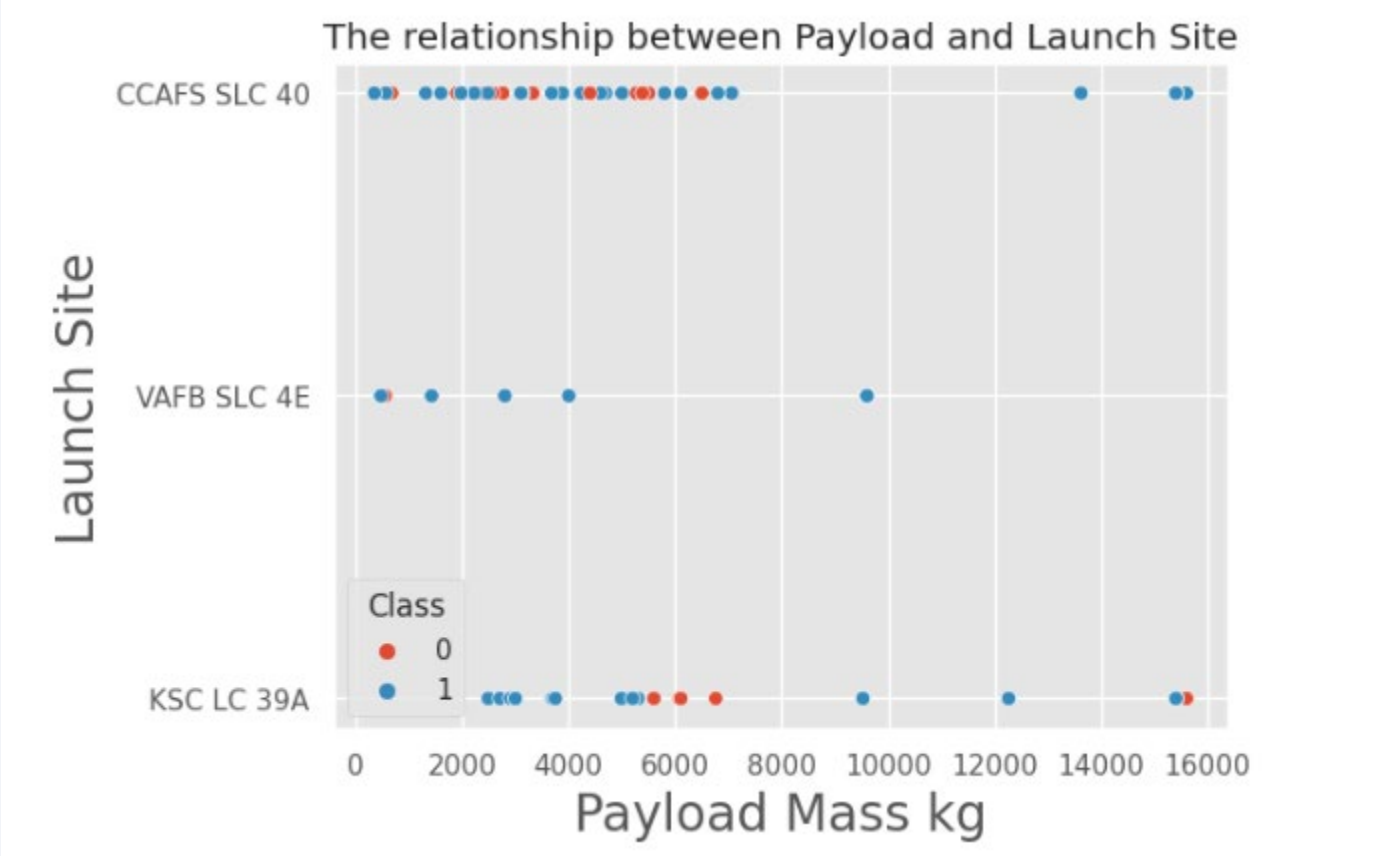
Flight Number vs. Launch Site

This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be.



Payload vs. Launch Site

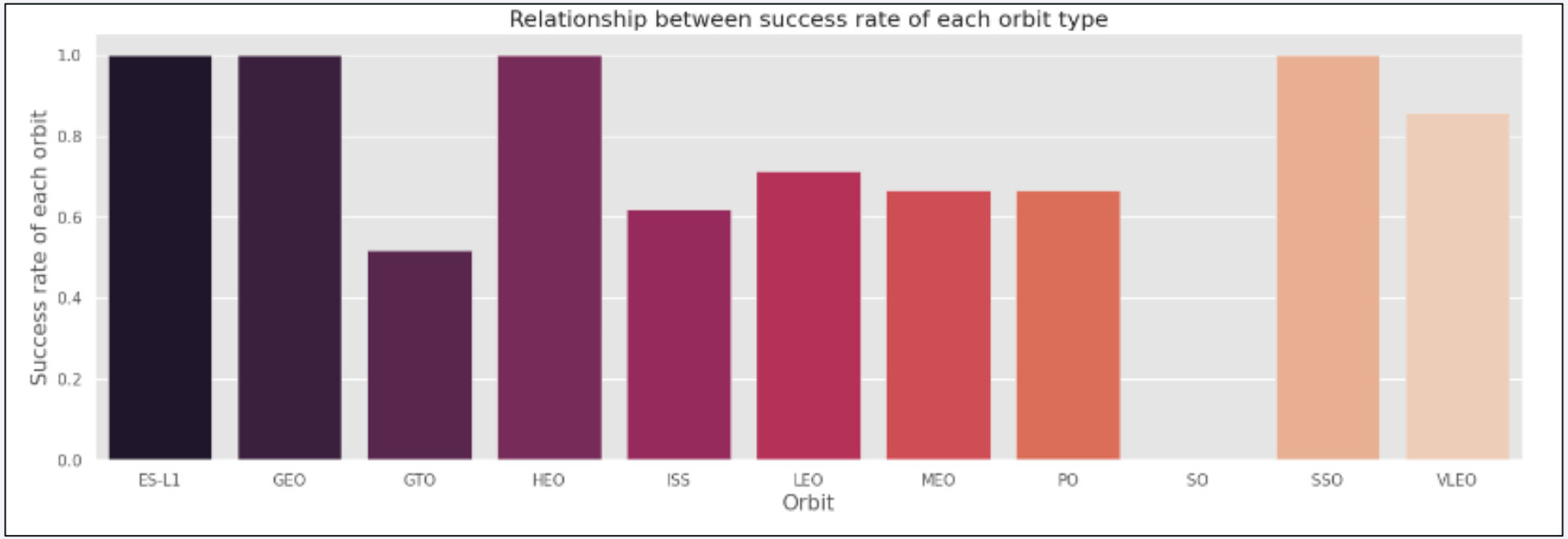
This scatter plot demonstrates that the success rate will be greatly boosted after the pay load mass exceeds 7000kg. However, there is no definite pattern to suggest that the success rate of the launch depends on the payload mass or the launch site.



Success Rate vs. Orbit Type

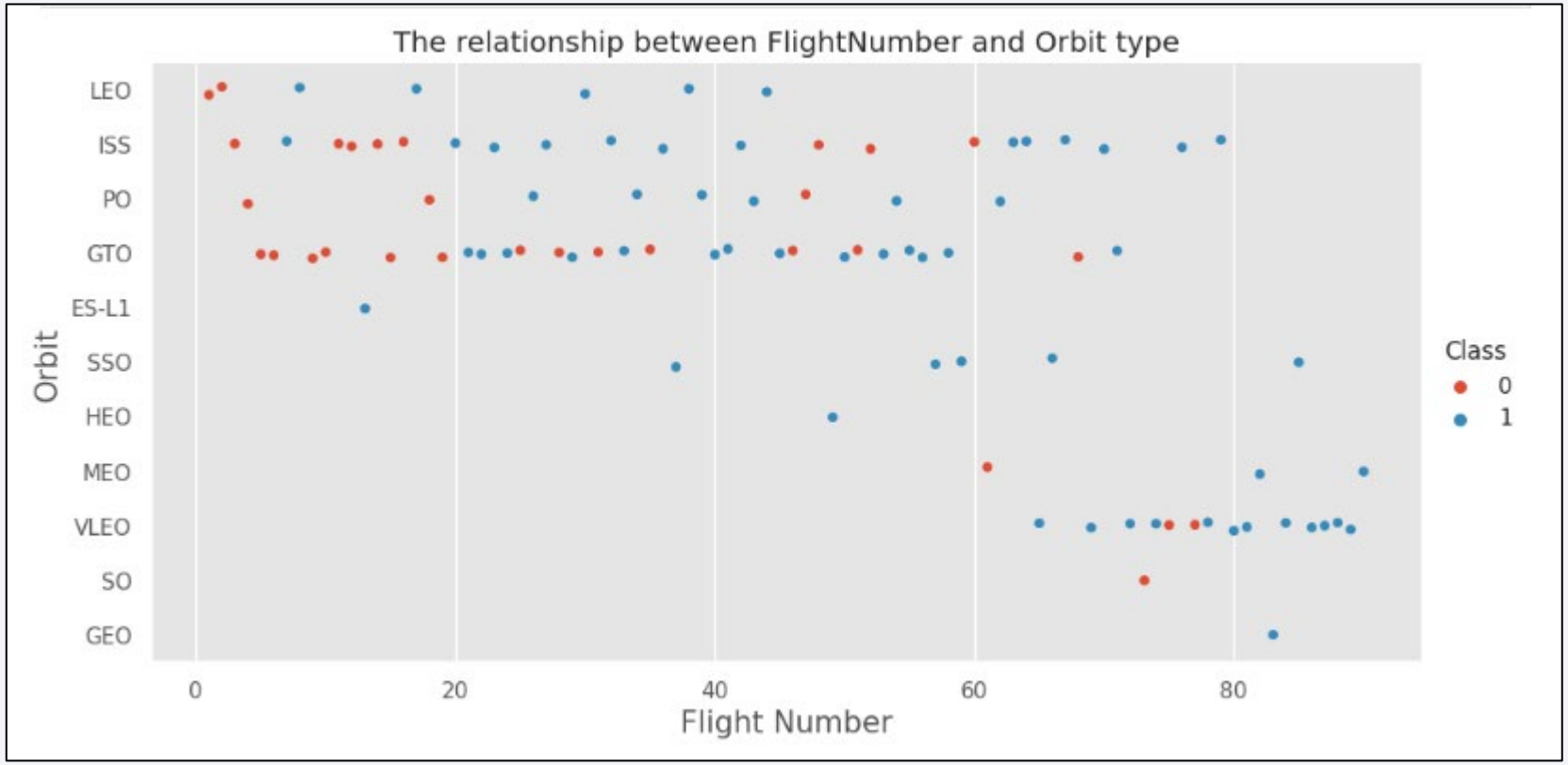
This diagram showed how the type of orbit could affect the success of a landing because some orbits, including SSO, HEO, GEO, and ES-L1, have 100% success rates whereas SO orbits have 0% success rates.

Deeper study reveals that several of these orbits, including GEO, therefore, HEO, and ES-L1, only occur once, therefore more datasets are required to observe any patterns or trends before any conclusions can be made.



Flight Number vs. Orbit Type

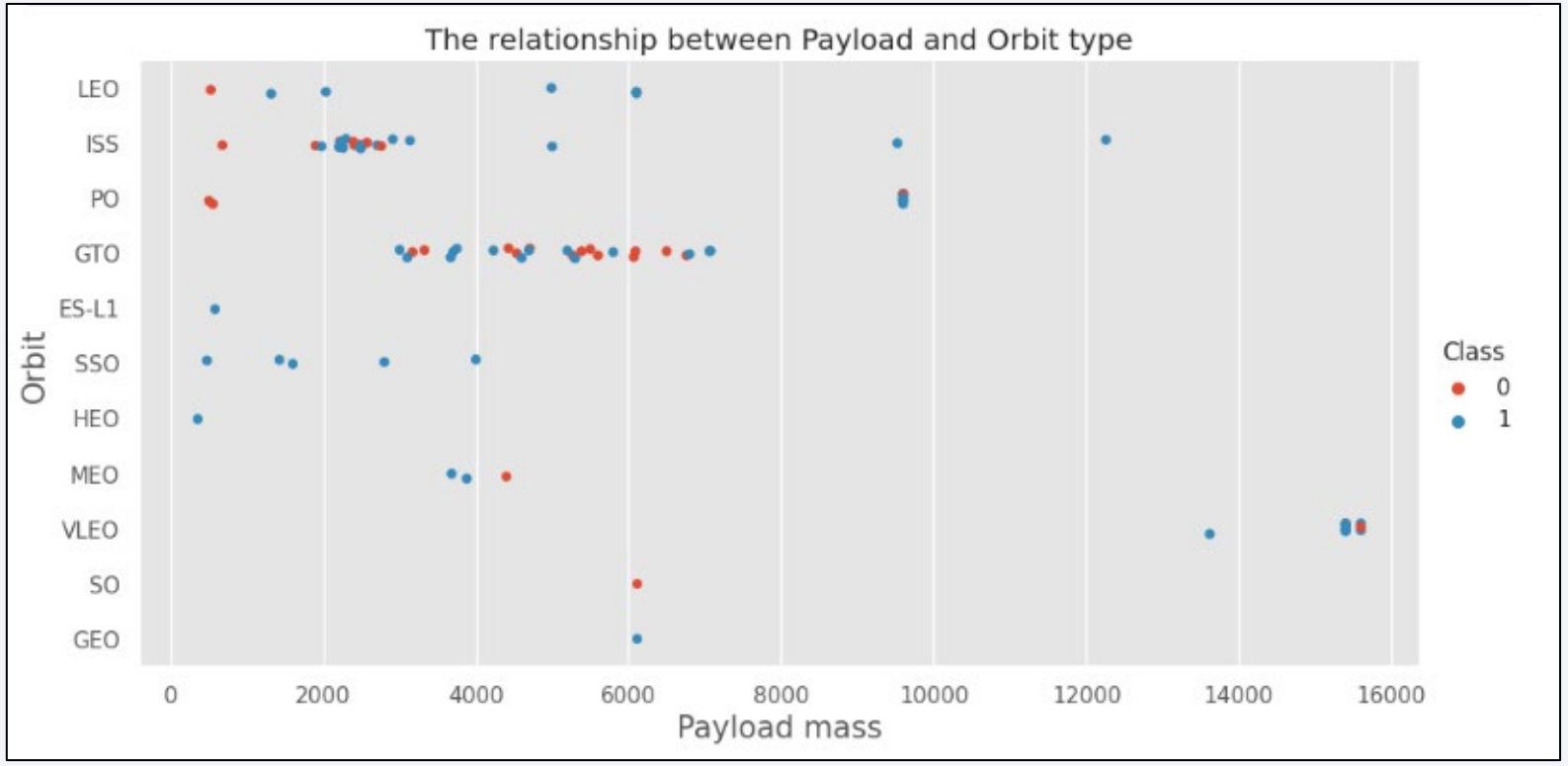
This scatter plot demonstrates that, generally speaking, the higher the flight number on each orbit, the higher the success rate (particularly on the LEO orbit), with the exception of the GTO orbit, where there is no correlation between the two characteristics.
The above statement should also eliminate orbits with a single occurrence because further data is required.



Payload vs. Orbit Type

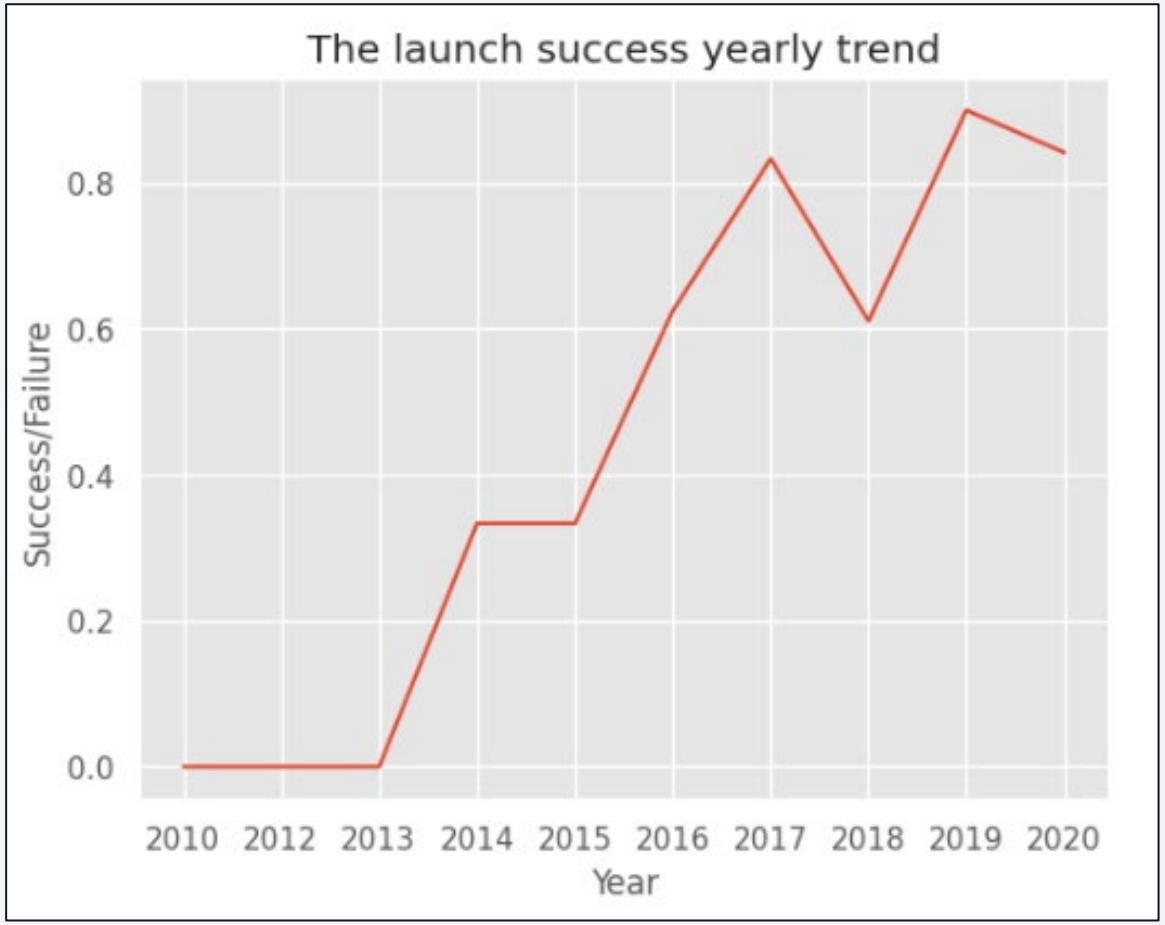
Heavier payload has positive impact on LEO, ISS and PO orbit. However, it has negative impact on MEO and VLEO orbit. GTO orbit seem to depict no relation between the attributes.

Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.



Launch Success Yearly Trend

These numbers showed a definite upward trend from 2013 to 2020.If this pattern continues into the following year. The success rate will rise gradually until it reaches a success rate of 1/100%.



All Launch Site Names

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

```
%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'

We used the query above to display 5 records where launch sites begin with `CCA`

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 619967 using the query below

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

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

payloadmass
619967

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

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

payloadmass
6138.287128712871

First Successful Ground Landing Date

We use the min() function to find the result andobserved that the dates of the first successful landing outcome on ground pad was 2010-04-06

```
%sql select min(DATE) from SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

min(DATE)
2010-04-06

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with 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_ BETWEEN 4000
```

```
 * 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

We used **COUNT** to filter for **GROUP BY** MissionOutcome was a success or a failure.

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SP
```

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

boosterversion
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

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT substr(Date, 6, 2) as month, DATE, Booster_Version,Launch_Site,Landing_Outcome from SPACEXTBL where Landing_Out
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
10	2015-10-01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

We selected Landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

We applied the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

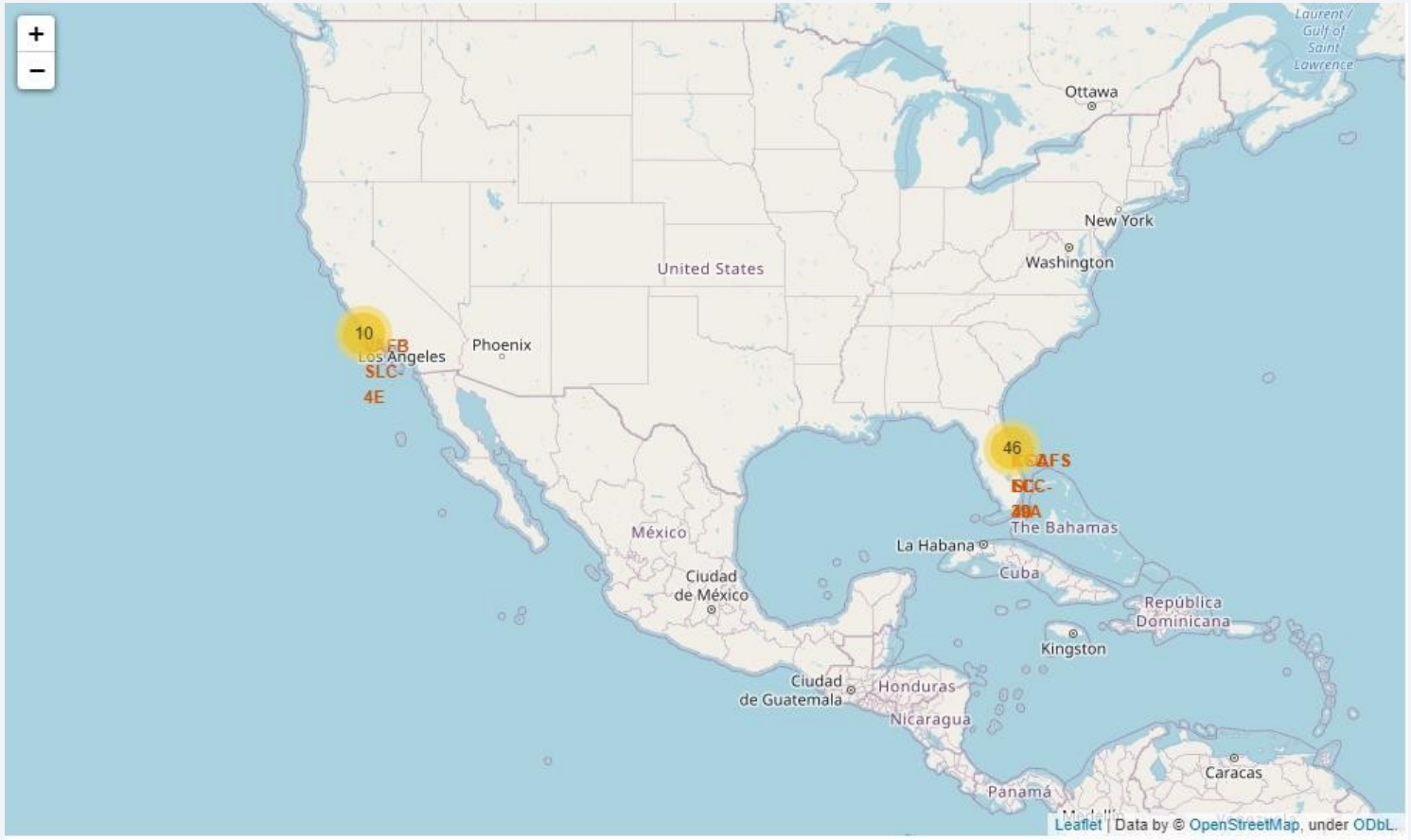
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome
No attempt
Success (ground pad)
Success (ground pad)
Success (drone ship)
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)


Section 3
Launch Sites
Proximities Analysis

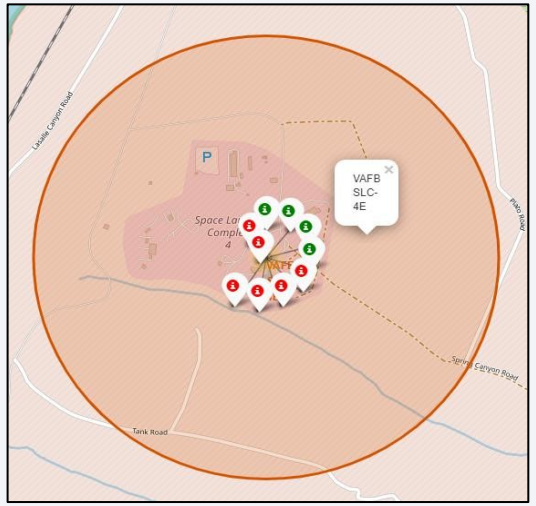
Location of all the Launch Sites

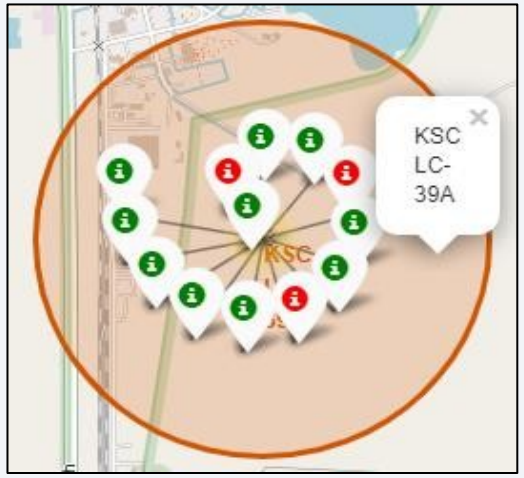
We can see that all the SpaceX launch sites are located inside the United States

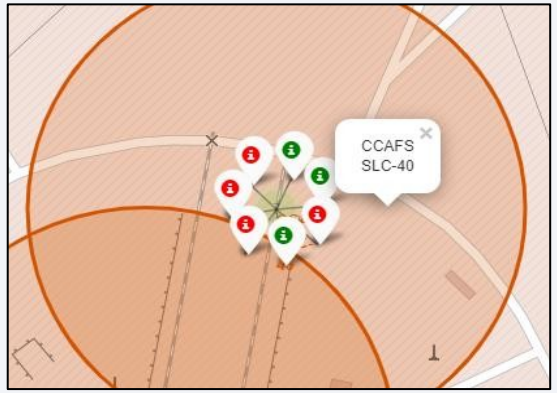


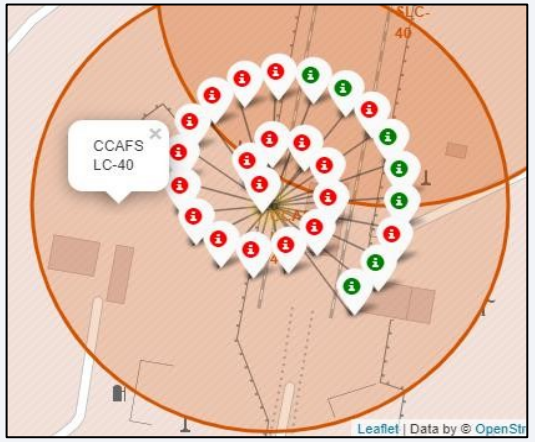
Color Labeled Markers

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

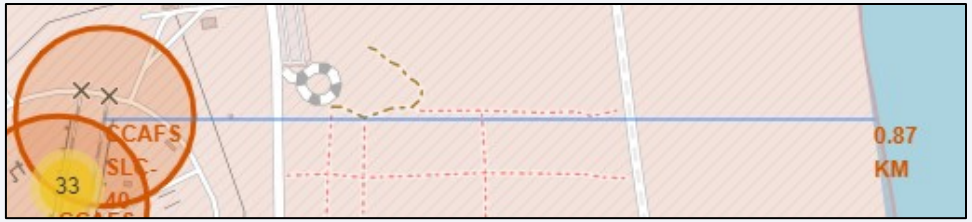


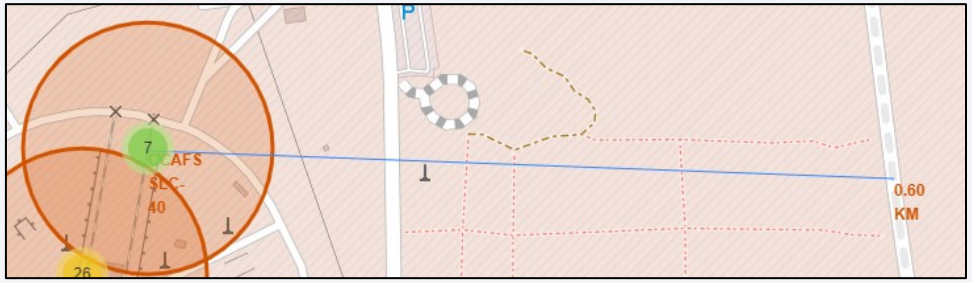




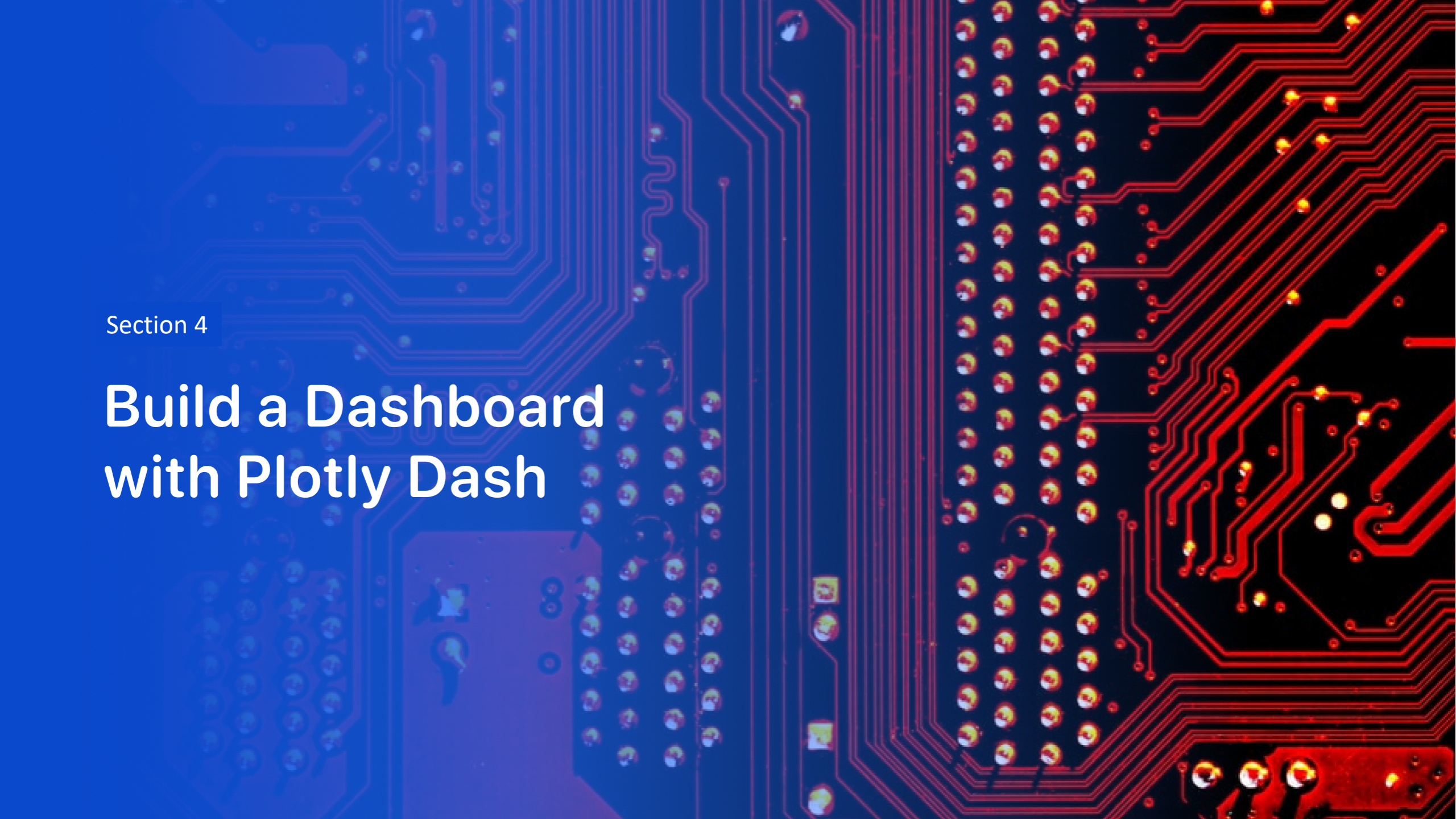


Launch Sites Distance to Landmarks



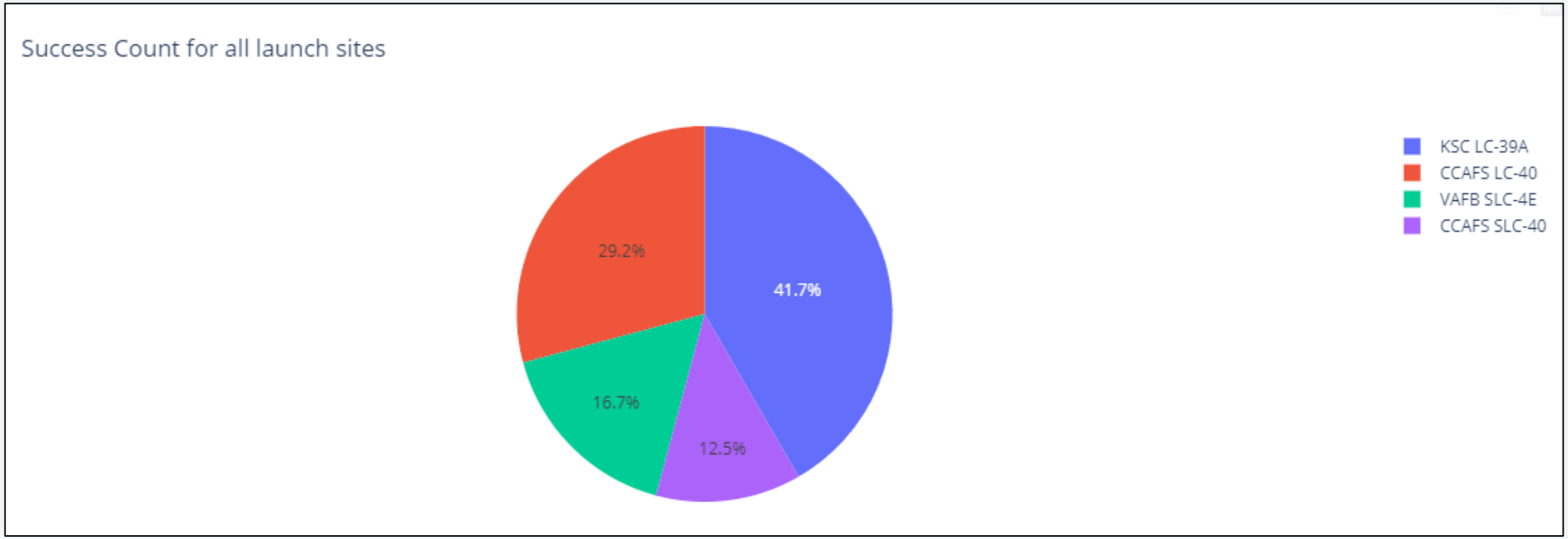


Is CCAFS SLC-40 in close proximity to railways ?
Yes
Is CCAFS SLC-40 in close proximity to highways ?
Yes
Is CCAFS SLC-40 in close proximity to coastline ?
Yes
Do CCAFS SLC-40 keeps certain distance away from cities ?
No


Section 4
Build a Dashboard with Plotly Dash

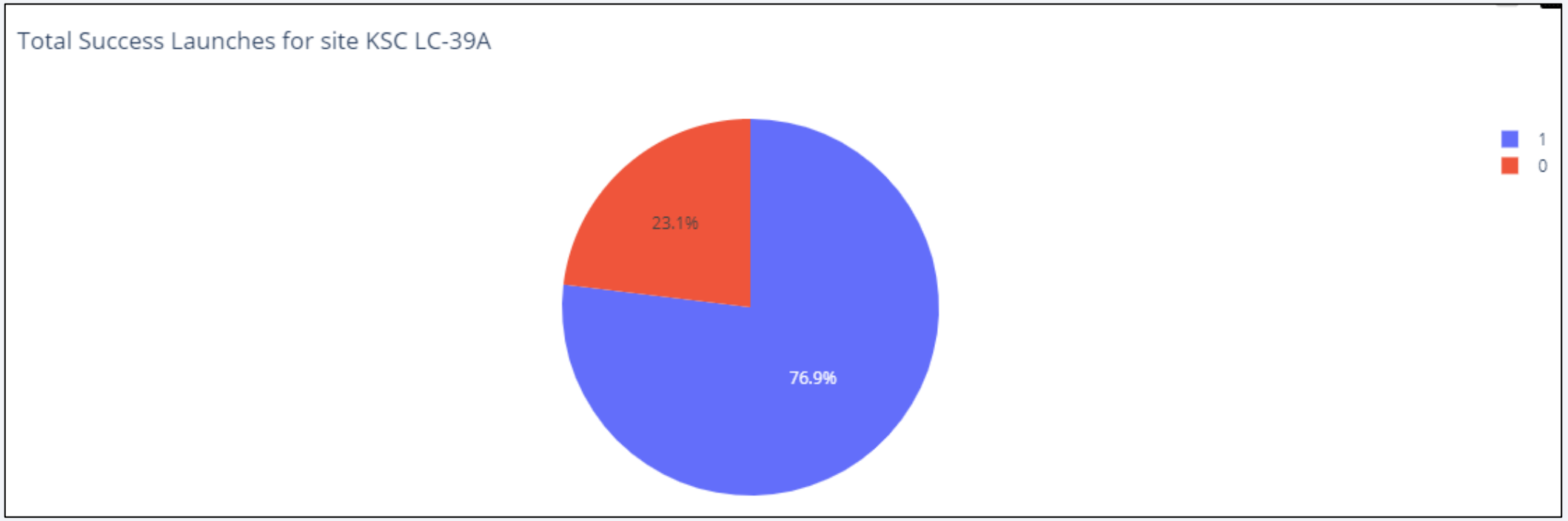
Total success by Site

We see that KSC LC-39A has the best success rate of launches.

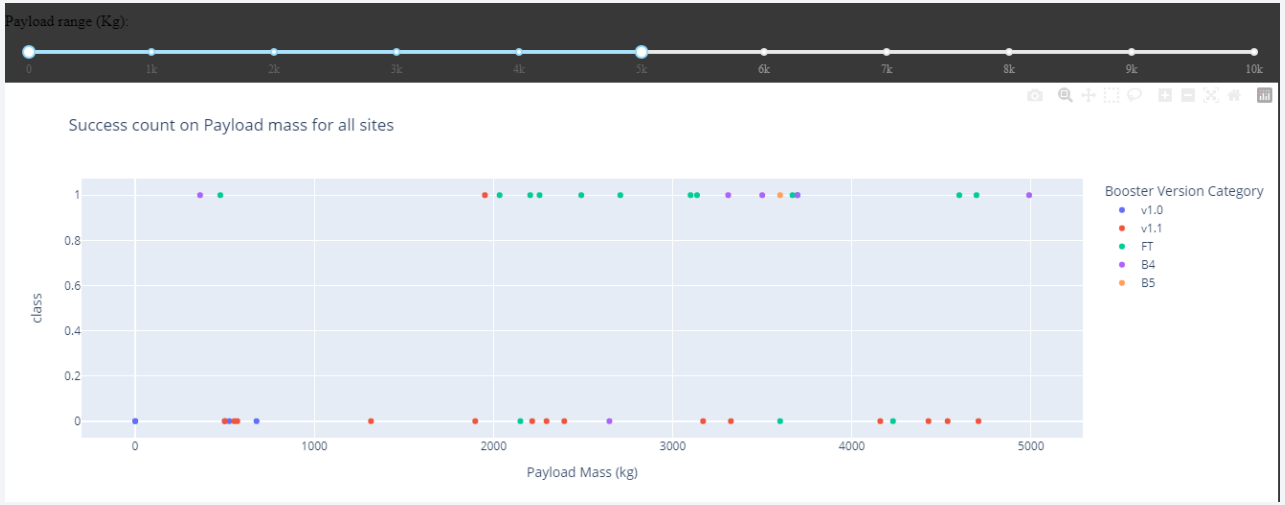


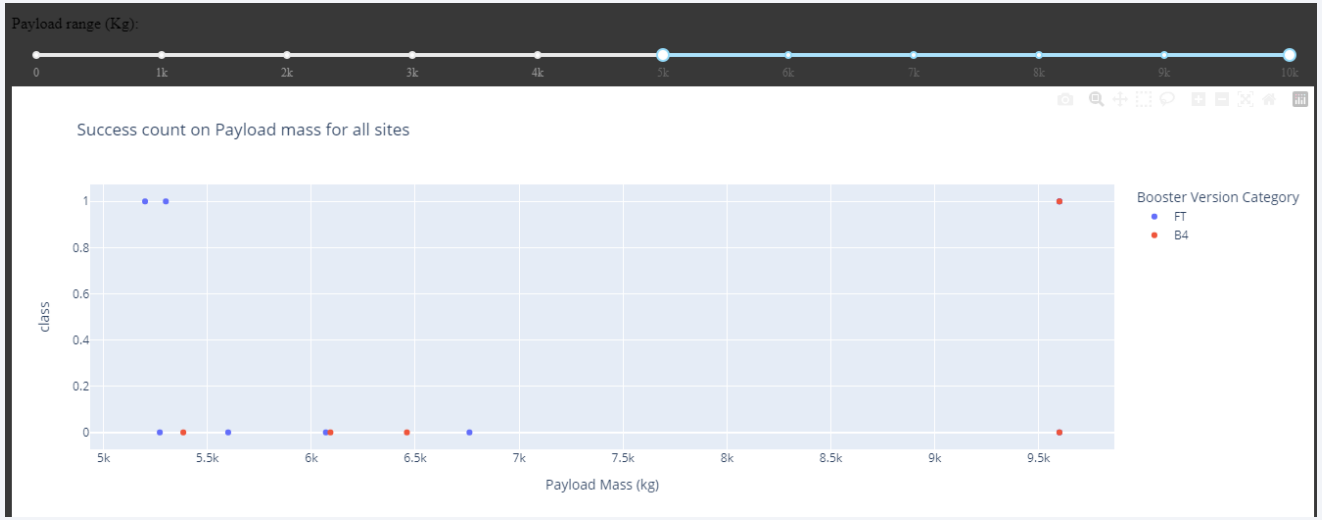
Total success launches for Site KSC LC-39A

We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.



Payload mass vs Outcome for all sites





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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we will take the decision tree.

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Decision tree best parameters

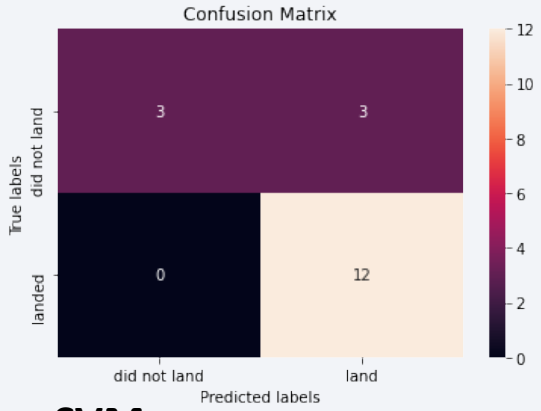
```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf':
2, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8732142857142856
```

Confusion Matrix

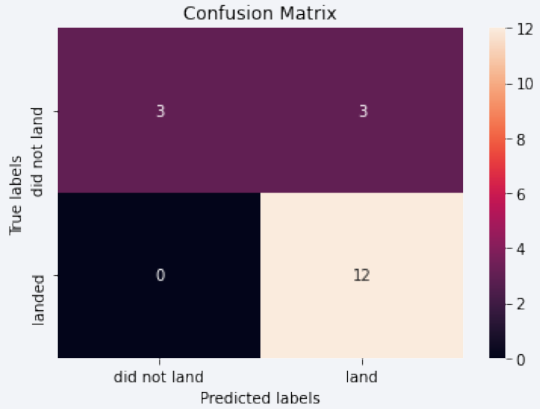
As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Logistic regression

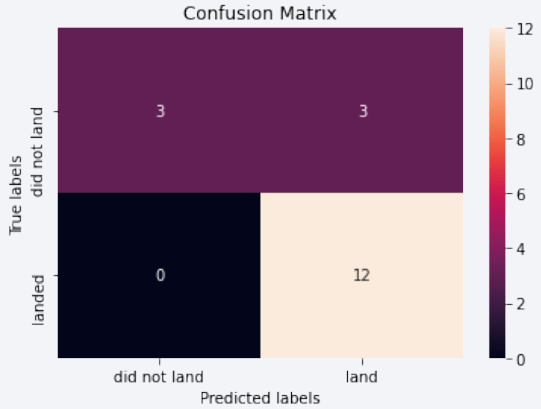
Decision Tree



kNN



SVM



Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Thank you!