Winning Space Race with Data Science

Waqas Haider
15-02-2024

Outline

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

Executive Summary

In a highly competitive business environment:

- SpaceX has revolutionized the launch of satellites with a concept of reusable launcher/booster: Falcon9/Falcon 9 Heavy
- The main advantage of this concept is the significant reduction in cost per kg.
- Reliability problems remain compared to classic launch vehicles like Soyuz or Ariane-5.
- For maintaining the “low cost” competitive advantage, compared with classic launchers, Falcon9 mission success is defined as the successful recovery or landing of the booster.
- Falcon9 booster successful recovery depends on features such as:
    - orbit
    - payload mass
    - booster versions
    - Launching sites…
- Based on these features, the best Machine Learning supervised classification model developed in this report, predicted booster recovery outcome with an accuracy close to 94%.

Introduction

- Project background and context SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage, Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

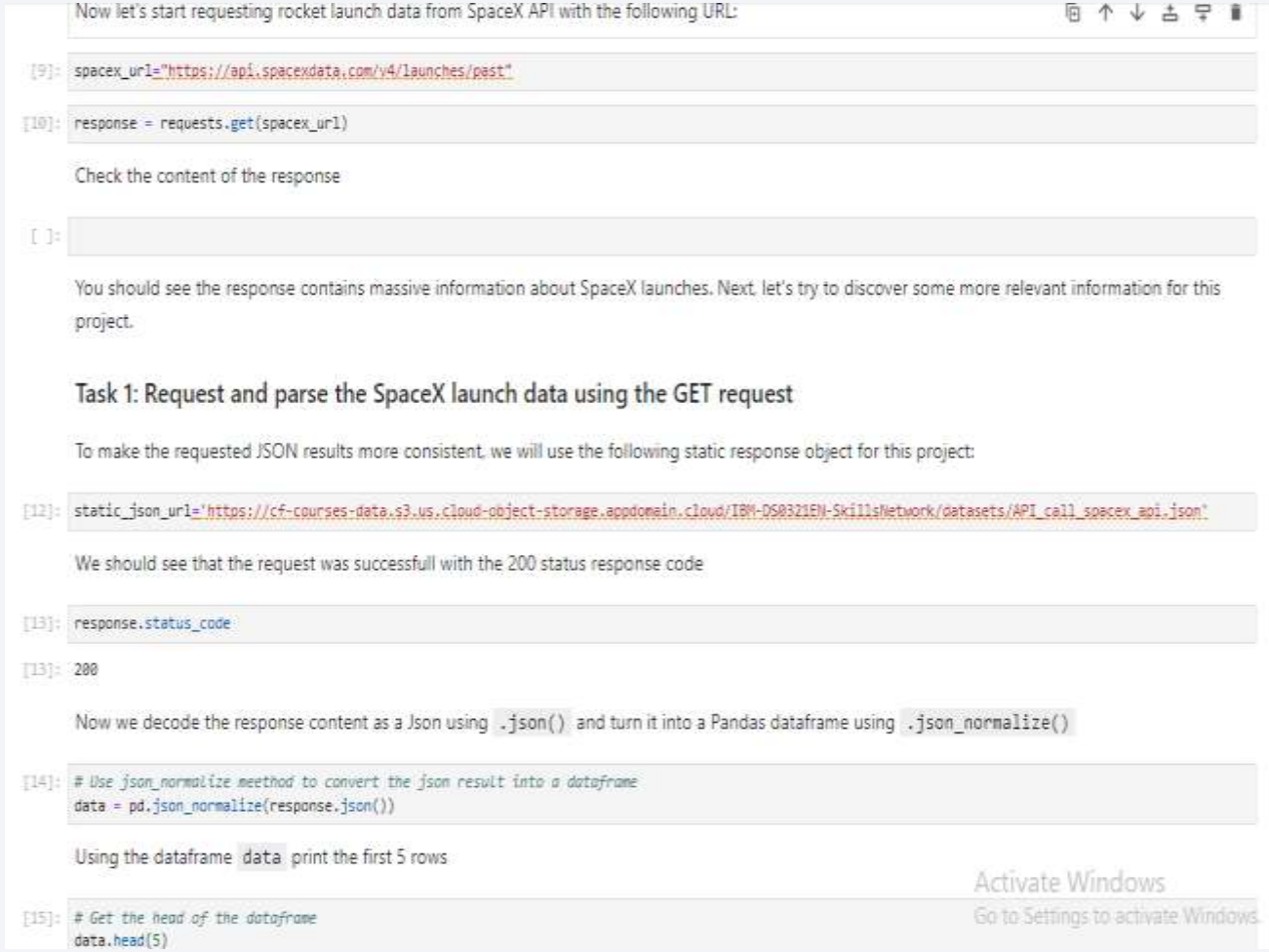
Methodology

Executive Summary

1. Data collection methodology
2. Perform data wrangling
3. Perform exploratory data analysis (EDA) using visualization and SQL
4. Perform interactive visual analytics using Folium and Plotly Dash
5. Perform predictive analysis using classification models

Data Collection – SpaceX API

- Data collected using SpaceX REST API by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a JSON result which was then converted into a Pandas data frame.
- GitHub URL of the completed SpaceX API calls notebook, https://github.com/WaqasHaider45/DataScienceProject/blob/main/Collecting%20the%20data.ipynb



Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

[10]: response = requests.get(spacex_url)
```

Check the content of the response

```
[ ]:
```

You should see the response contains massive information about SpaceX launches. Next, let's try to discover some more relevant information for this project.

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[12]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[13]: response.status_code

[13]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

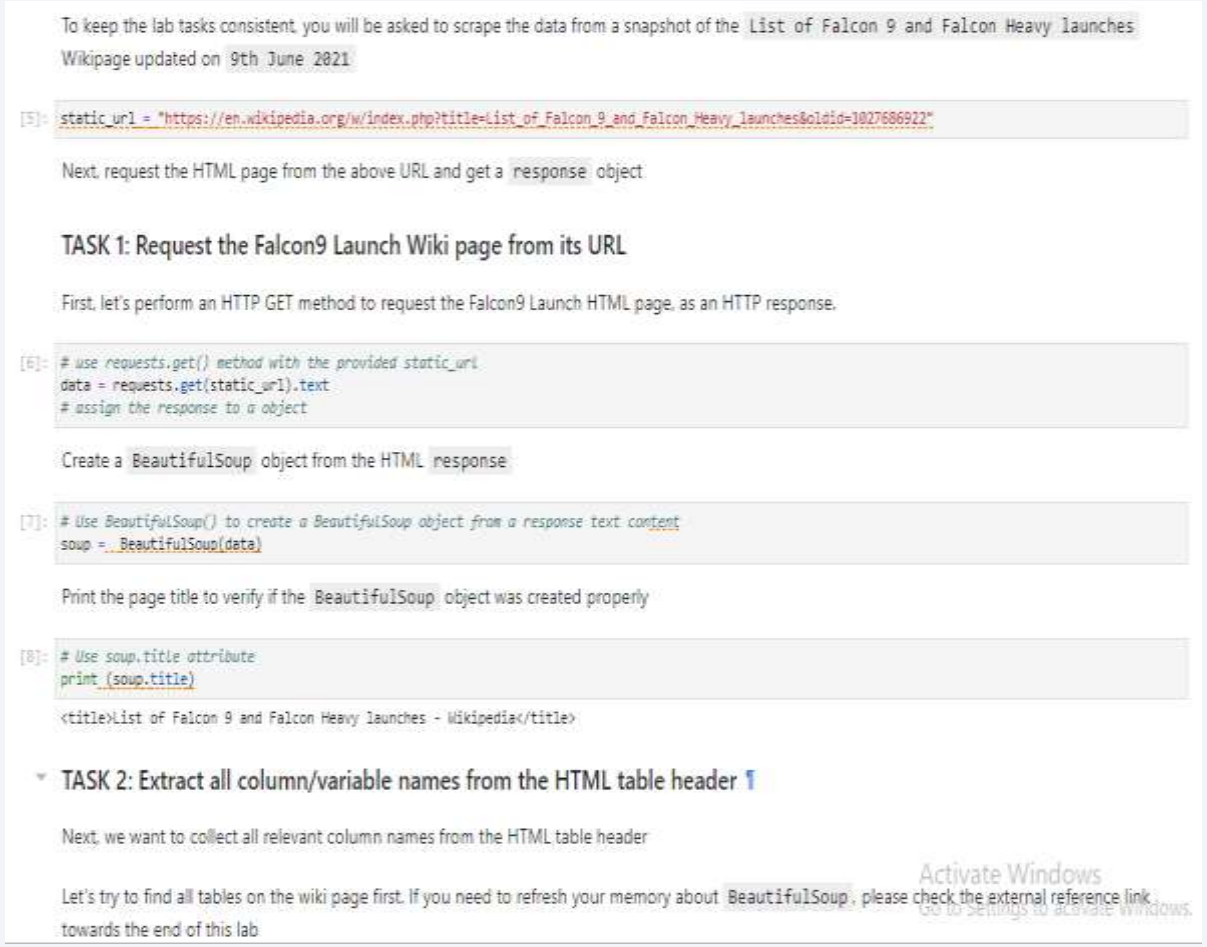
```
[14]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
[15]: # Get the head of the dataframe
data.head(5)
```

Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using Beautiful-Soup and request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.
- GitHub URL of the completed web scraping notebook, https://github.com/WaqasHaider45/Data-ScienceProject/blob/main/Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records.ipynb



To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

```
[5]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

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

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

```
[7]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data)
```

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

```
[8]: # Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

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

Data Wrangling

- After obtaining and creating a Pandas DF from the collected data, data was filtered using the Booster-Version column to only keep the Falcon 9 launches.
- Missing values in Payload Mass column are replaced by the mean value of the column.
- GitHub URL of completed data wrangling related notebooks, https://github.com/WaqasHaider45/Data-ScienceProject/blob/main/Data%20Wrangling.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

In [11]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

Out[11]:
```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

In [12]:
```
landing_class = df['Class']
df[['Class']].head(8)
```

Out[12]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1

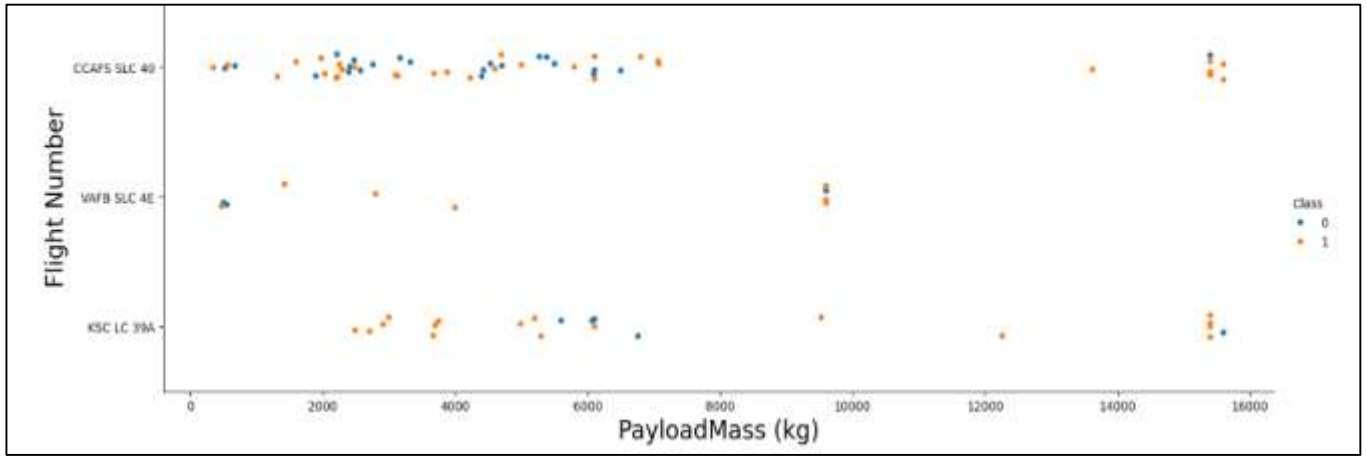
EDA with Data Visualization

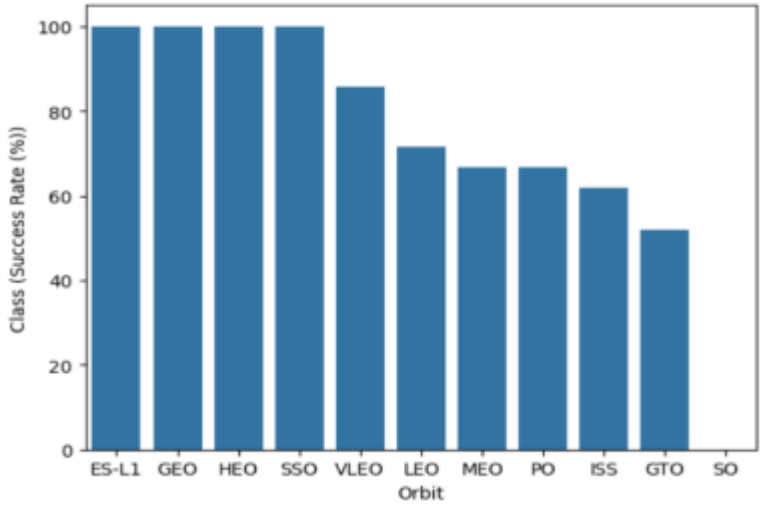
- Performed Exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib:

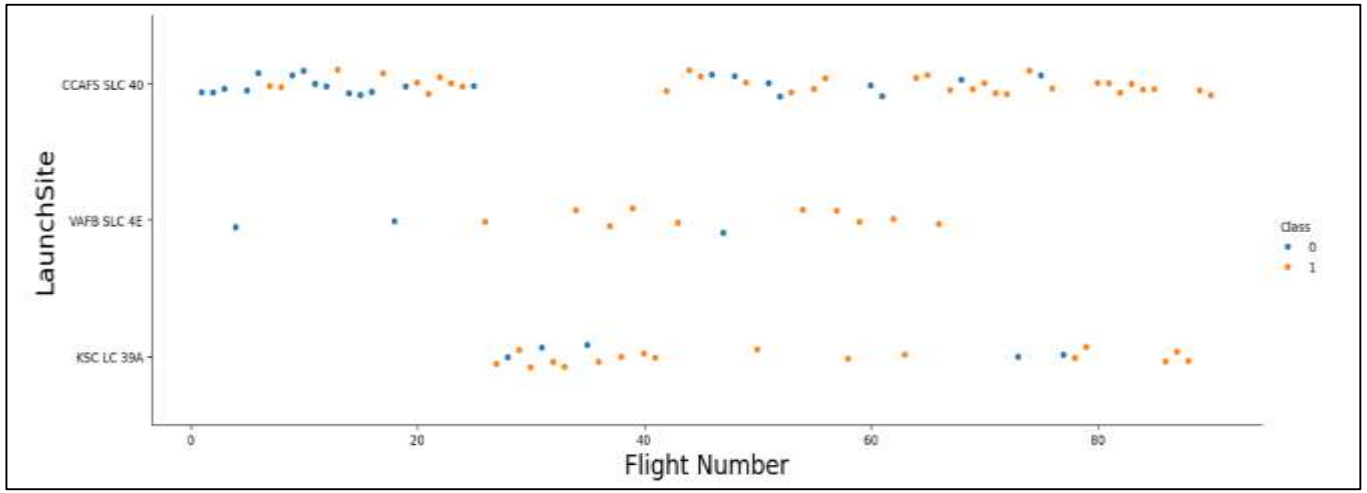
1. Exploratory Data Analysis
2. Preparing Data Feature Engineering
3. Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, Payload and Orbit type.
4. Used Bar chart to Visualize the relationship between success rate of each orbit type.
5. Line plot to Visualize the launch success yearly trend.

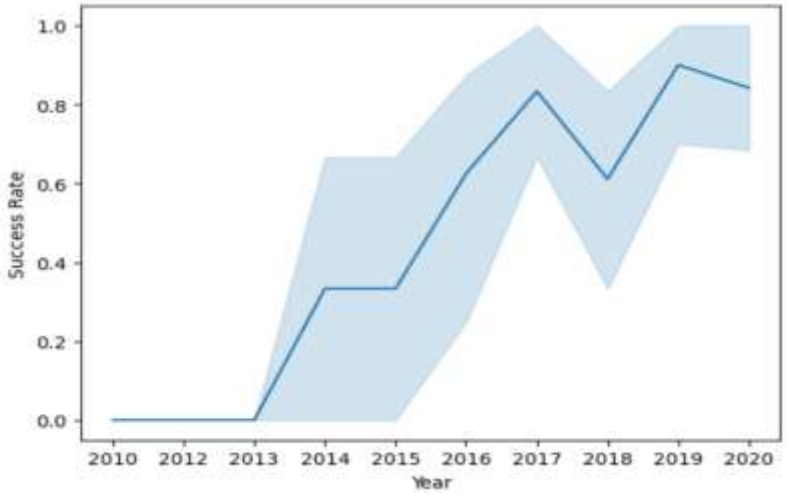
- GitHub URL of completed EDA with data visualization notebook, https://github.com/WaqasHaider45/DataScienceProject/blob/main/Exploring%20and%20Preparing%20Data.ipynb

EDA Visualization Charts:









EDA with SQL

The following SQL queries were performed for EDA:

- **Display the names of the unique launch sites in the space mission**.

%sql select Unique(LAUNCH_SITE) from SPACEXTBL

- **Display 5 records where launch sites begin with the string 'CCA’.**

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

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

%sql select sum(PAYLOAD_MASS__KG_) as payload-mass from SPACEXTBL;

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

%sql select avg(PAYLOAD_MASS__KG_) as payload-mass from SPACEXTBL;

- **List the date when the first successful landing outcome in ground pad was achieved.**

%sql select min(DATE) from SPACEXTBL;

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

%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

- **List the total number of successful and failure mission outcomes**

%sql select count(MISSION_OUTCOME) as mission outcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;

- **List the names of the booster versions which have carried the maximum payload mass. Use a subquery**

%sql select BOOSTER_VERSION as booster version from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);

- GitHub URL of completed EDA with SQL notebook, as an external reference and peer-review purpose, https://github.com/WaqasHaider45/DataScienceProject/blob/main/Overview%20of%20the%20DataSet.ipynb

Build an Interactive Map with Folium

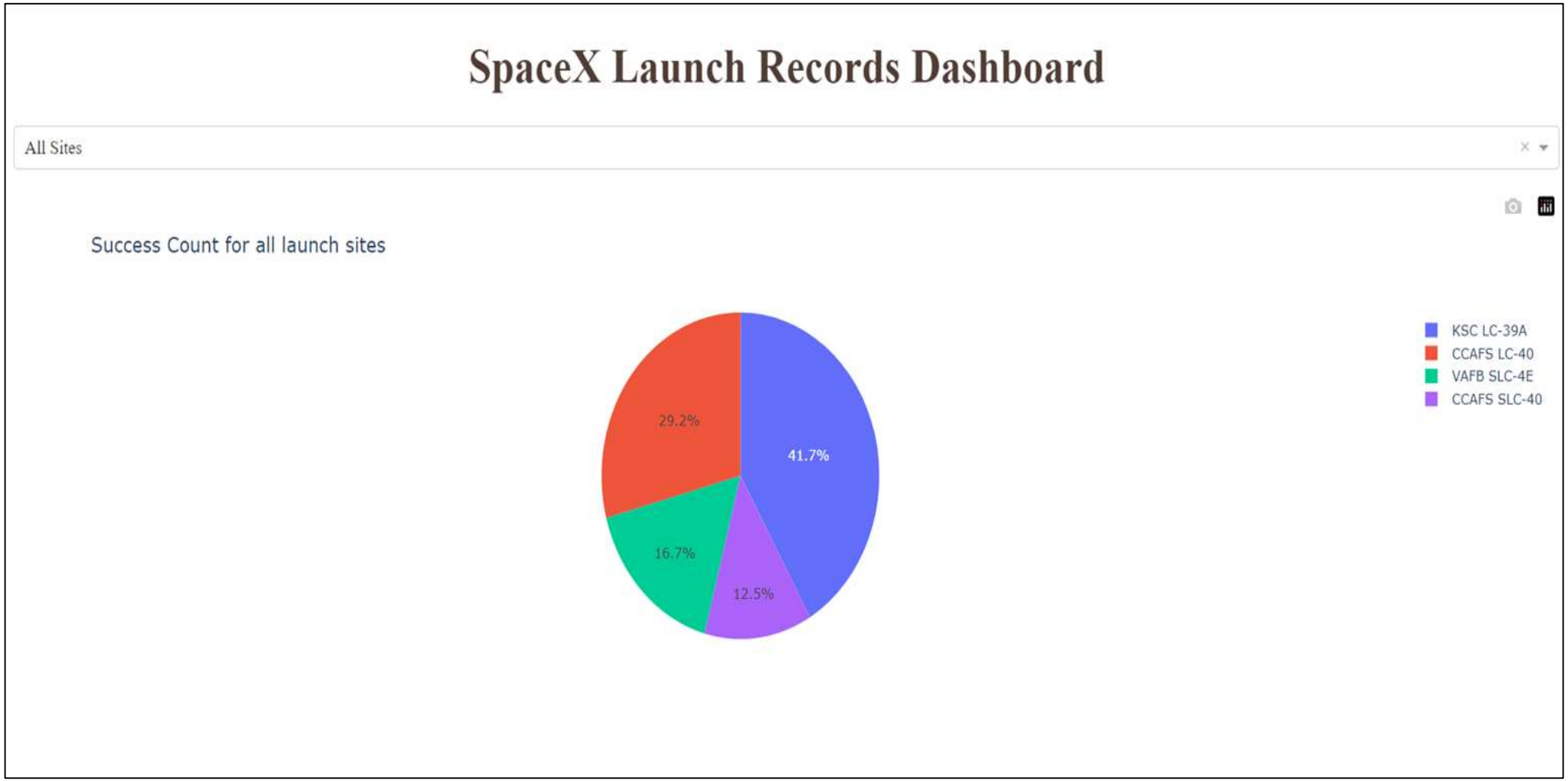
- A map have been created through folium library to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Launch set outcomes (failure=0 or success=1).
- GitHub URL of completed interactive map with Folium map, as an external reference and peer-review purpose, https://github.com/WaqasHaider45/DataScienceProject/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotly dash by:
  - Adding a Launch Site Drop-down Input Component.
  - Adding a callback function to render success-pie-chart based on selected site dropdown.
  - Adding a Range Slider to Select Payload.
  - Adding a callback function to render the success-payload-scatter-chart scatter plot.

- GitHub URL of completed Plotly Dash lab, as an external reference and peer-review purpose, https://github.com/WaqasHaider45/DataScienceProject/blob/main/SpaceX%20Launch%20Records%20Dashboard.py

SpaceX Launch Records Dashboard:



Predictive Analysis (Classification)

Short overview of finding the best predictive analysis technique.

- After loading the data as a Pandas Dataframe, perform Exploratory Data Analysis and determine Training Labels by;
  - Creating a NumPy array from the column Class in data, by applying the method to numpy() then assigned it to the variable Y as the outcome variable.
  - Then standardized the feature dataset (x) by transforming it using preprocessing. StandardScaler() function from Sklearn.
  - Finally data split into training and testing sets using the function train_test_split from sklearn.model selection with the test size parameter set to 0.2 and random state to 2.

Predictive Analysis (Classification)

- To find the best ML model/ method that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;

    1. First created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.
    2. For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to find best Hyperparameter.
    3. After fitting the training set, we output GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_.
    4. Finally using the method score to calculate the accuracy on the test data for each model and plotted a confussion matrix for each using the test and predicted outcomes.

Predictive Analysis (Classification)

- The table below shows accuracy score of test data for each of the methods comparing them to show which performed best.
- Techniques include SVM, Classification Trees, K Nearest Neighbors and Logistic Regression.

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333

- GitHub URL of completed predictive analysis lab, as an external reference and peer-review purpose, https://github.com/WaqasHaider45/Data-Science-Project/blob/main/SpaceX%20Machine%20Learning%20Prediction.ipynb

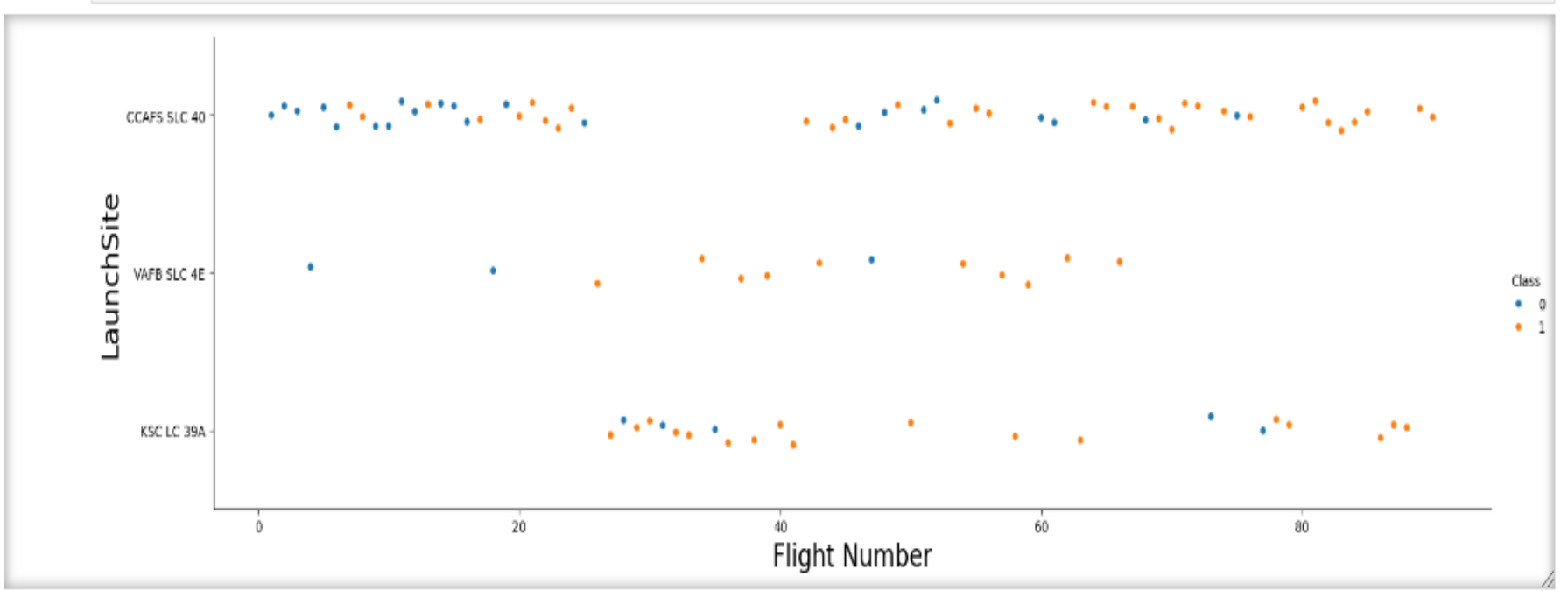
Results

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

Section 2

Insights drawn from EDA

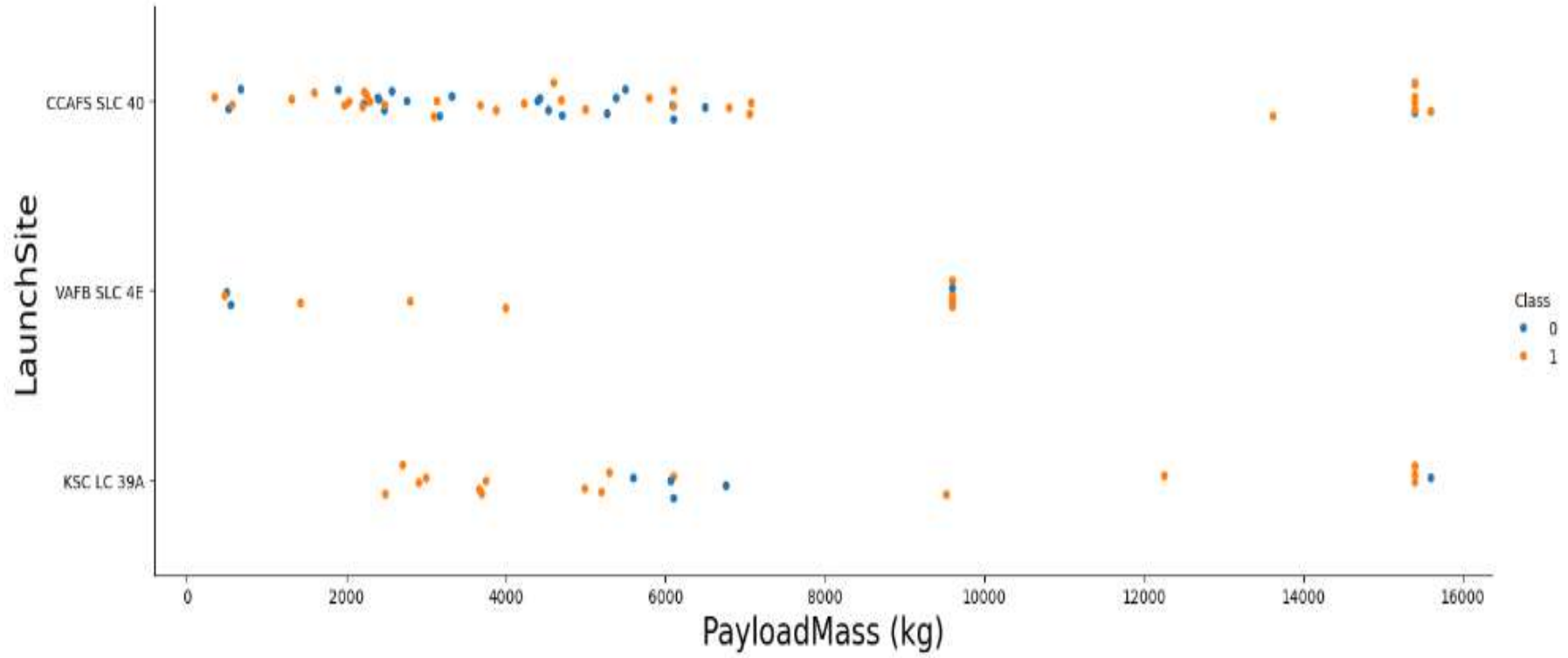
Flight Number vs. Launch Site



Now try to explain the patterns you found in Flight Number and Launch Site.

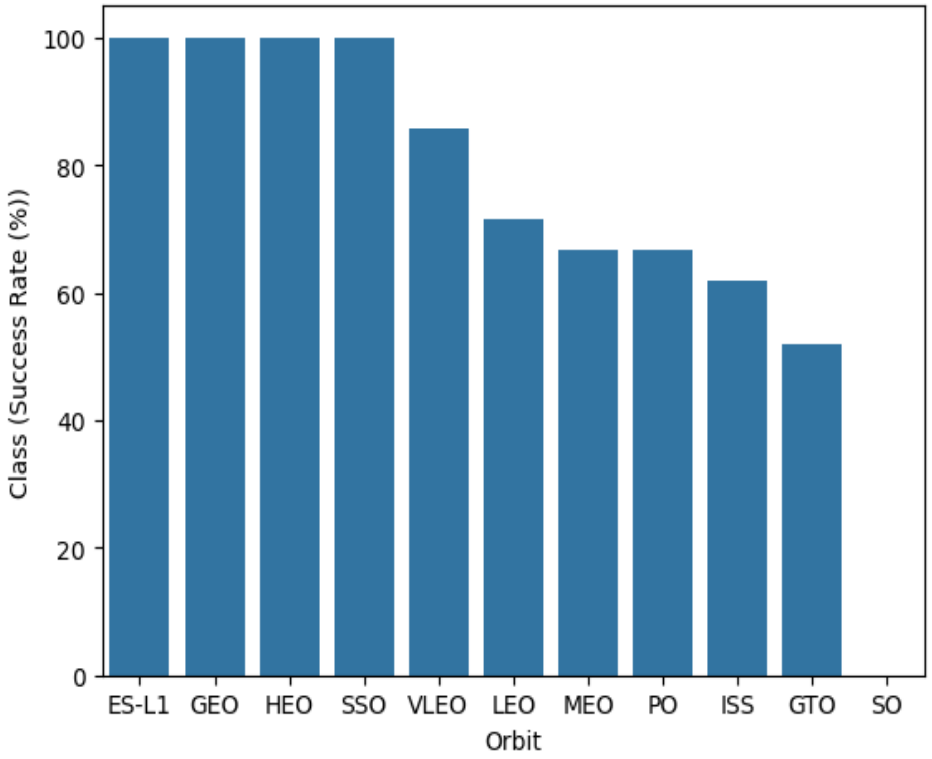
As the flight number increases it is clear to be seen that the success ration increases in all Launch sites. Focusing on VAFB SLC 4E almost all lauches after 20 Fight Number is successful except 1, while all the launches of other two are succesful after 80 Flights.

Payload vs. Launch Site



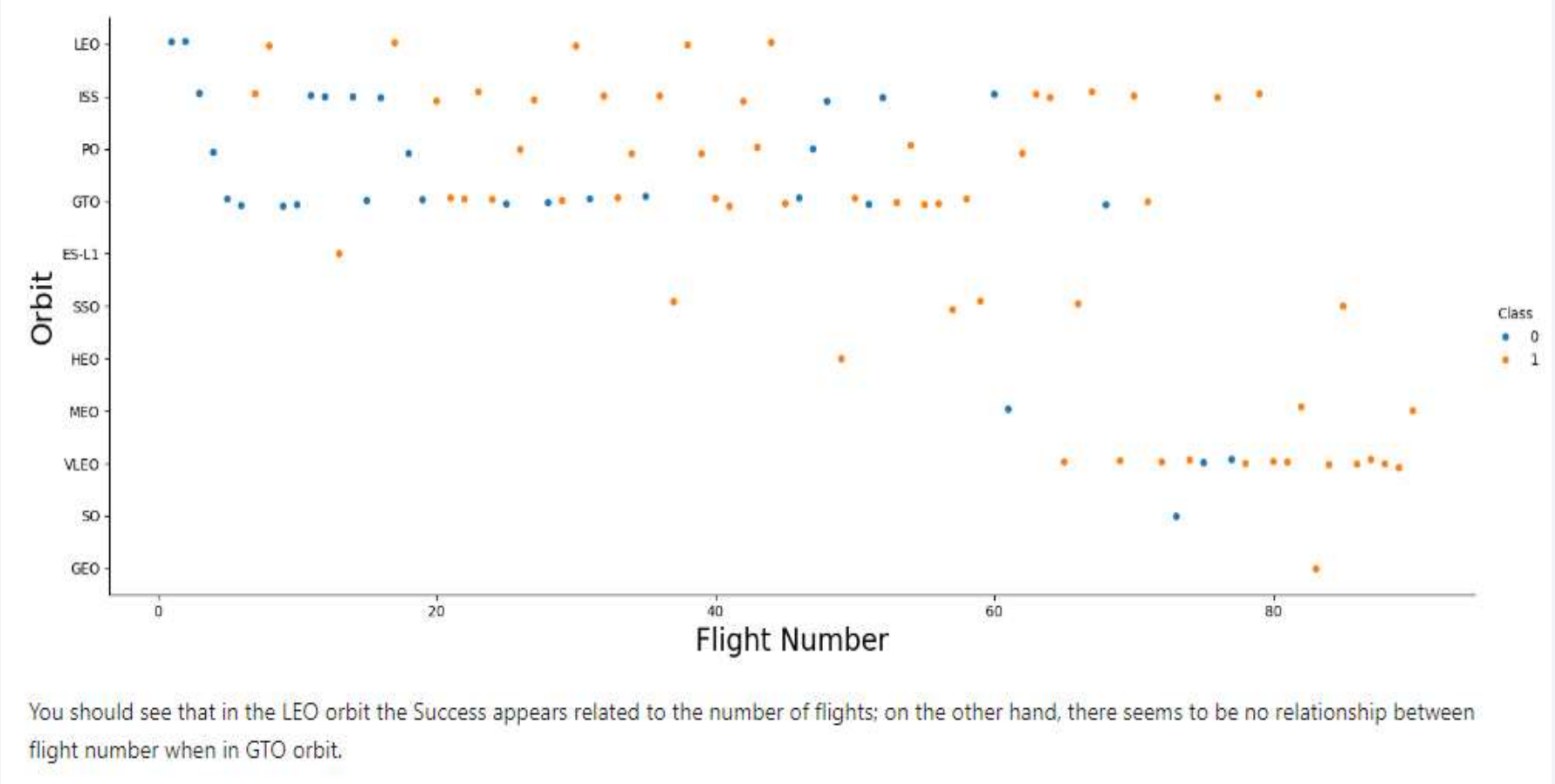
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



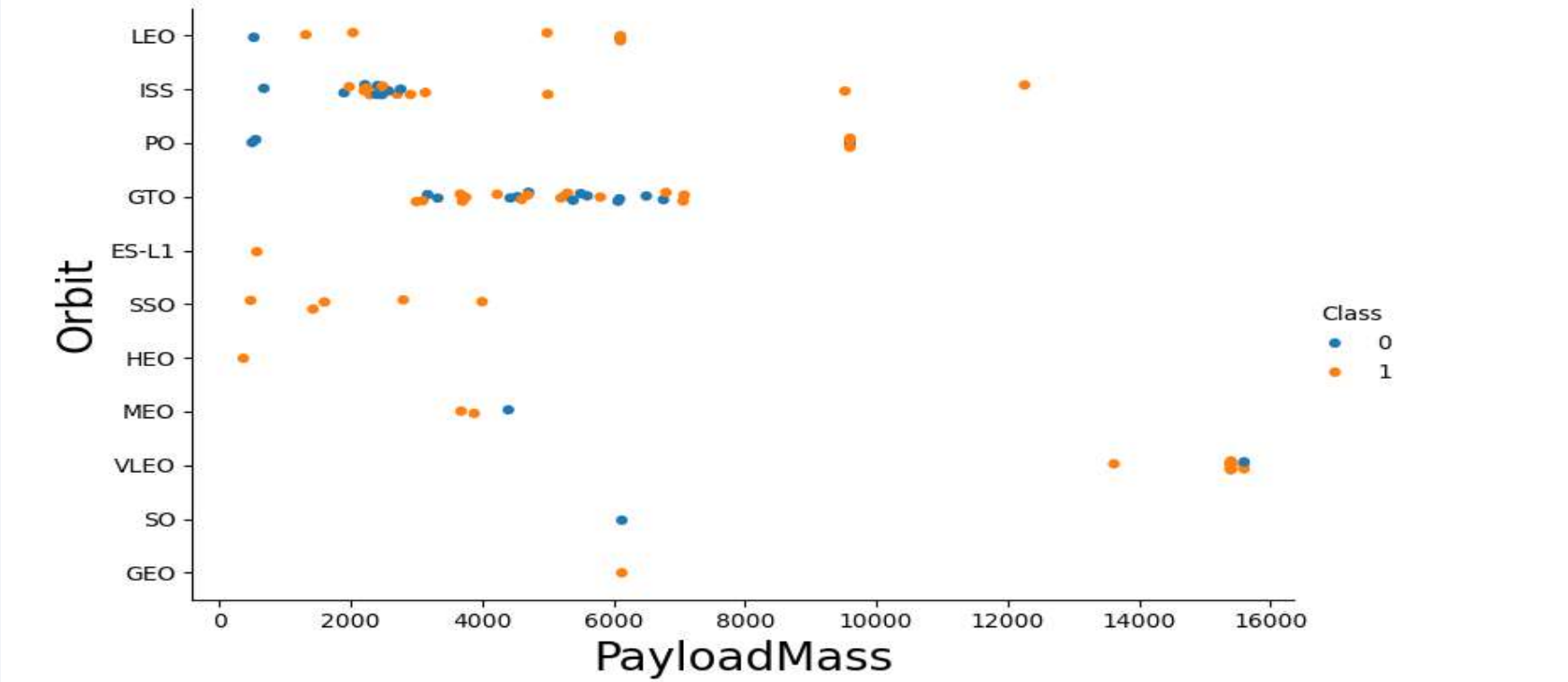
Analyze the ploted bar chart try to find which orbits have high sucess rate. Orbit ES-L1, GEO, HEO, SSO are the successful orbits having 100% success rate while orbit SO having success rate 0%. From orbits VLEO till GTO success rate is continuously decreasing and stay between 85% to 50% respectivley.

Flight Number vs. Orbit Type



You should see that 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.

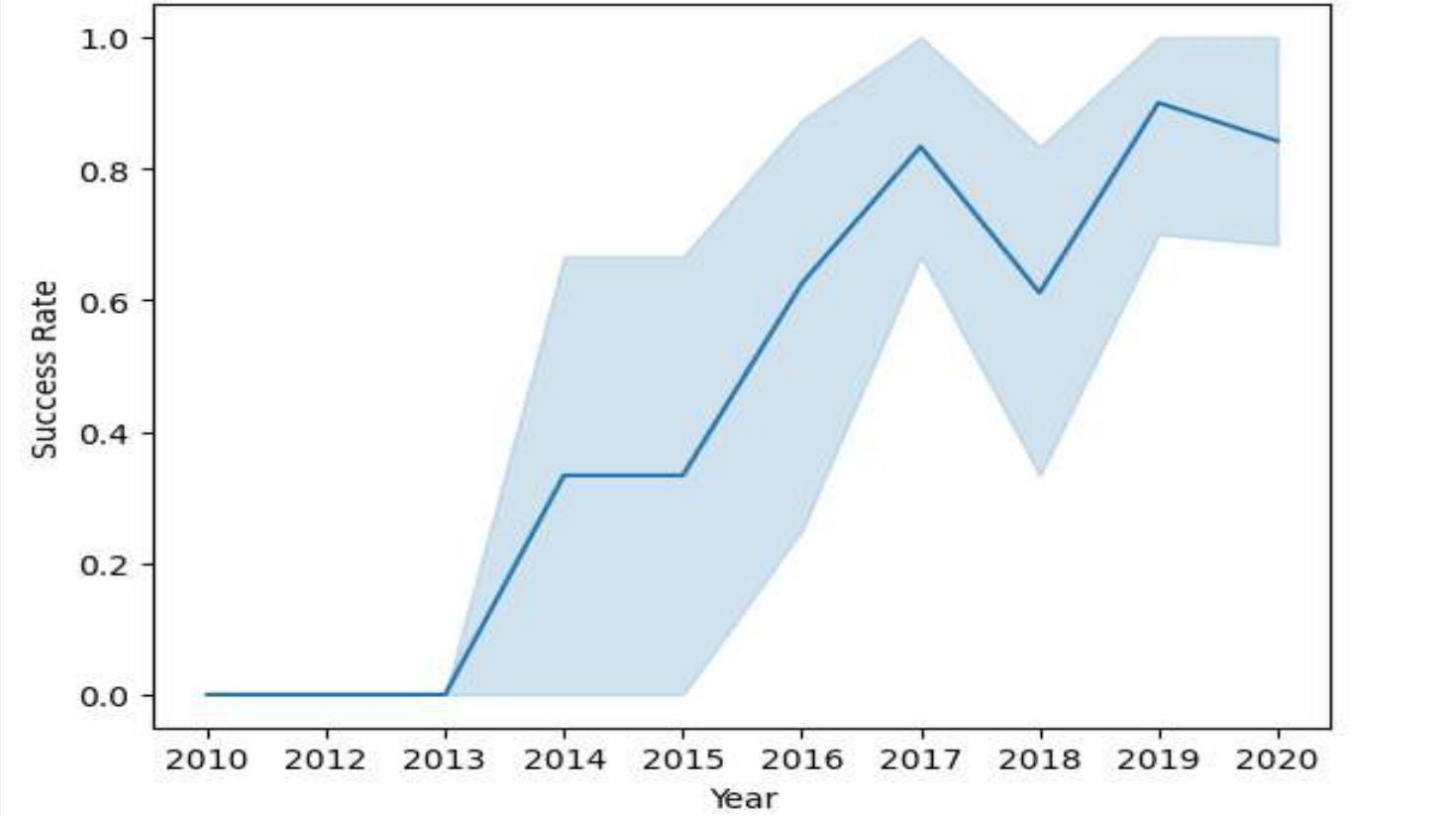
Payload vs. Orbit Type



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(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- All launch sites are taking as a list called launch site. Using distinct we get all the unique different sites used in launching.

```
%sql select distinct LAUNCH_SITE as "LauchSite" from SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- This is the list of launch site specifically starts with CCA with the of LIKE query and using LIMIT query to get only top 5 results.

```
%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

- Using SUM query can make the total of all payload mass from Payload_Mass_Kg_ column.

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

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

payloadmass
619967

Average Payload Mass by F9 v1.1

- The average payloadmass can be find by using AVG on the column of payload_mass_kg.

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

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

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

payloadmass
6138.287128712871

First Successful Ground Landing Date

- This is the .first date of a successful landing outcome by using MIN on the date column

```
%sql select min(DATE) from SPACEXTBL;
```

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

min(DATE)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- This is the list from BOOSTER Version column and applying two conditions on them:

1. Landing outcome must be on drone ship and should be successful.
2. Payload mass kg should be between 4000 to 6000. AND statement shows that all conditions must be fulfilled.

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Used the COUNT together with the GROUP BY statement to return total number of missions outcomes.

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

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

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

Boosters Carried Maximum Payload

- Using a Subquerry to return and pass the Max payload and used it list all the boosters that have carried the Max payload of 15600kgs.

```
%sql select BOOSTER_VERSION, payload as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

Booster_Version	boosterversion
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21

2015 Launch Records

- Used the subsrt in the select statement to get the month and year from the date column where substr(Date,7,4)='2015' for year and Landing_outcome was ‘Failure (drone ship) and return the records nmatching the filter.

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Ou
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order using DESC statement.

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDE
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)

Section 3

Launch Sites Proximities Analysis

Folium Map

There are total 4 launch sites of SpaceX:

- VAFB SLC-4E: Vandenberg Space Launch Complex 4 (CA)
- KSC-LC29A: Kennedy Space Center - Merritt Island (FL)
- CCAFS-LC40: Cape Canaveral Launch Complex 40 (FL)
- CCAF-SLC40: Cape Canaveral Space Launch Complex 40(FL)

Folium Map

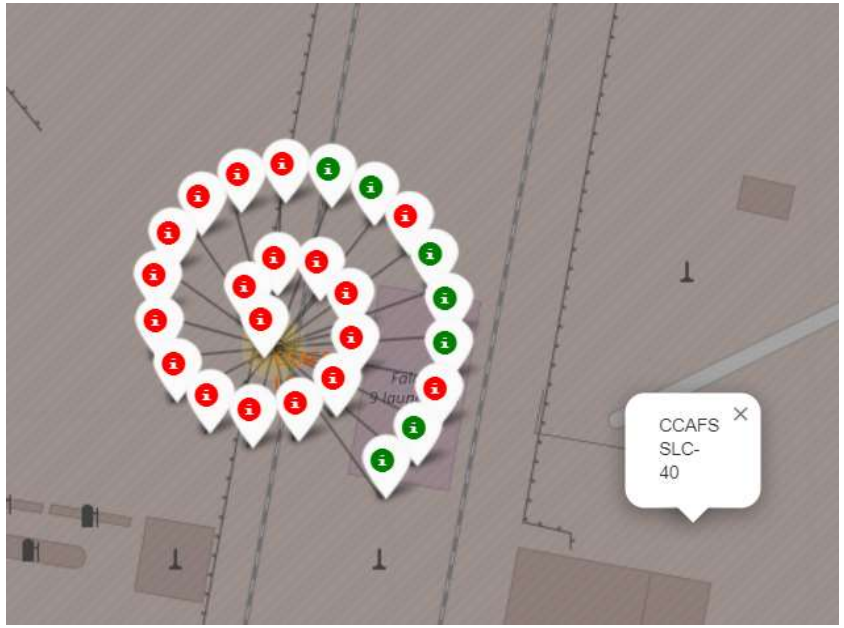
These are the successful and unsuccessful launch sites. Green shows successful launch while red are unsuccessful launches.

- The KSC LC 39A have mostly successful launches which is 10 successful while 3 unsuccessful.

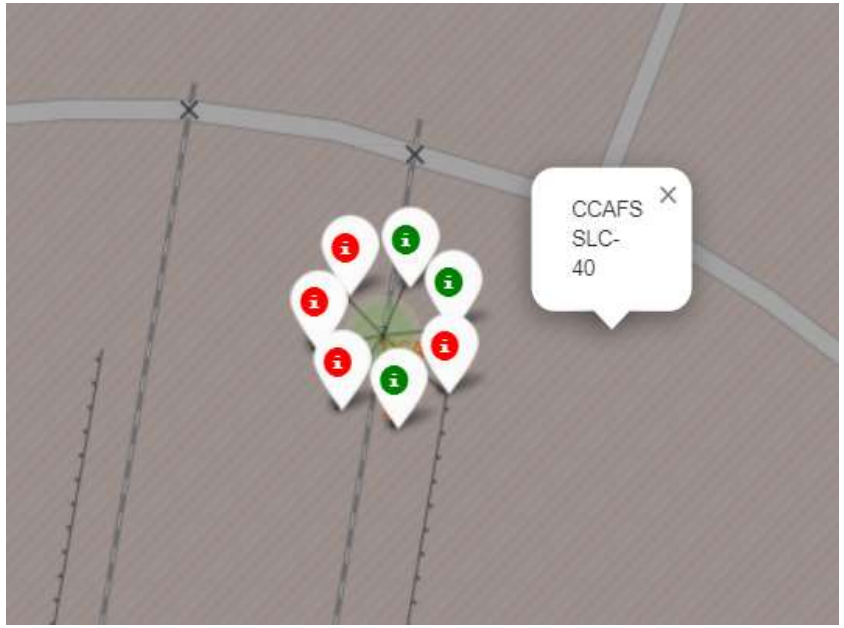
Kennedy Space Center (FL)
KSC LC 39A

Folium Map

- Launch site CCAFS SLC-40 have two launch site close to each other.
- The first site have 26 launches out of which only 7 were successful while the other left representation shows total 7 launches out of which only 3 were successful.
- In general it is clear that the launches from this area are mostly unsuccessful.



Cape Canaveral (FL)
CCAFS-LC40



Cape Canaveral (FL)
CCAFS-SLC40

Folium Map

- This is the launch site close to Los Angeles which is totally opposite to other three sites.
- The ratio of successful launches was not good enough having 6 unsuccessful launches with 4 successful in total of 10 launches.

Vandenberg Space Launch Complex 4 (CA)
VAFB SLC-4E

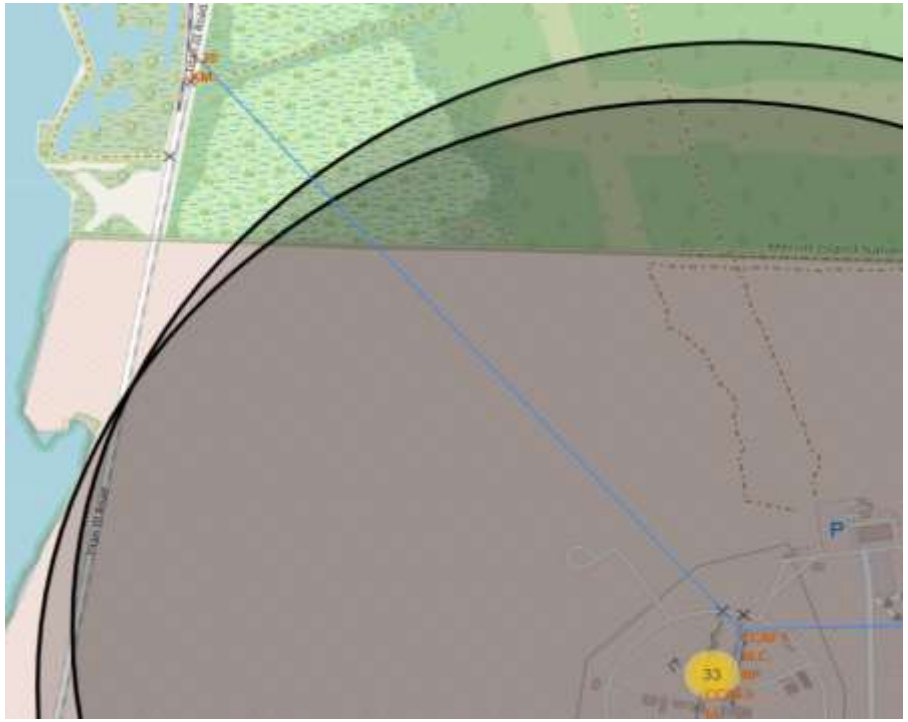
Folium Map

Launch site we use to check the distance with highways, train ways and city is **Cape Canaveral (FL) CCAFS-SLC40**

- This map shows the line that defines the closest distance of launch site with the highway.
- Distance with the highway is 0.58 km.

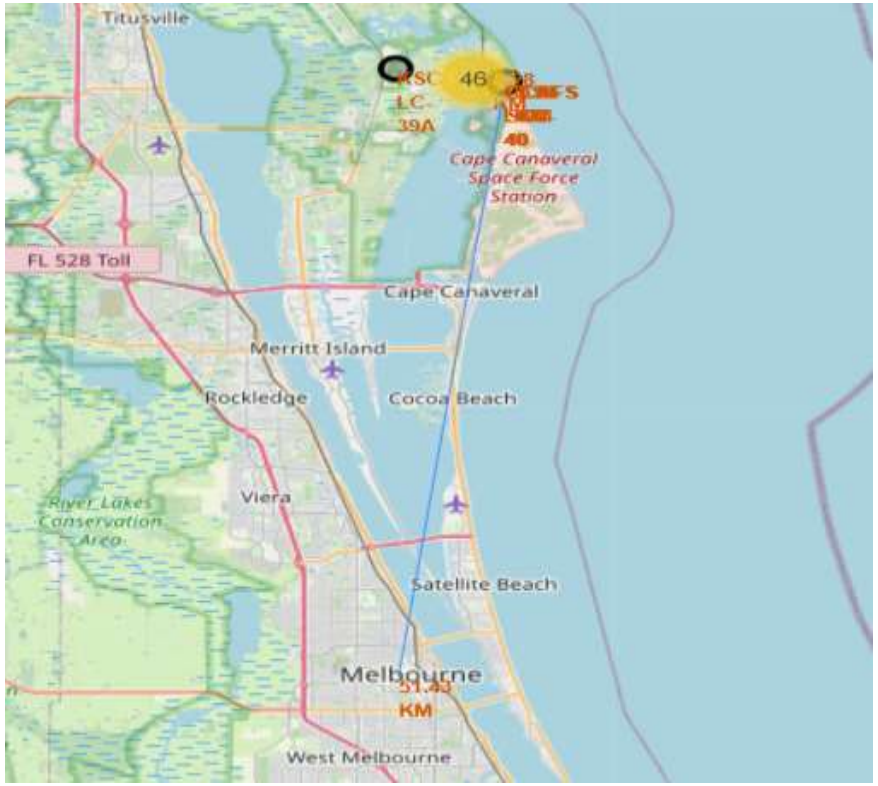
Folium Map

- The distance with railway line is almost 3.28 km.



Folium

- Melbourne is the closest big city from the launch site which is around 51.43 km.



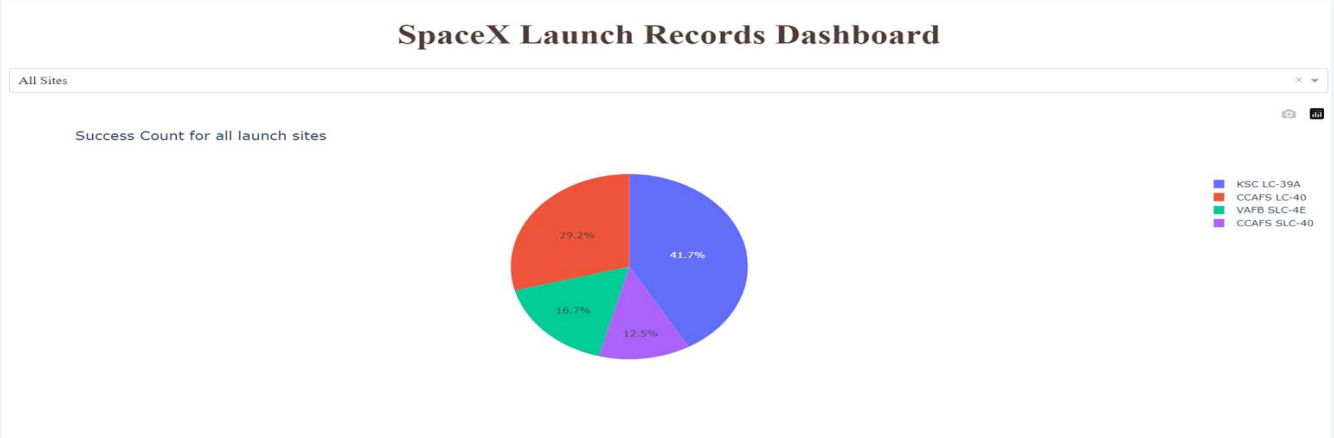
Section 4

Build a Dashboard with Plotly Dash

Plotly Dashboard

We built an interactive dashboard with Plotly including:

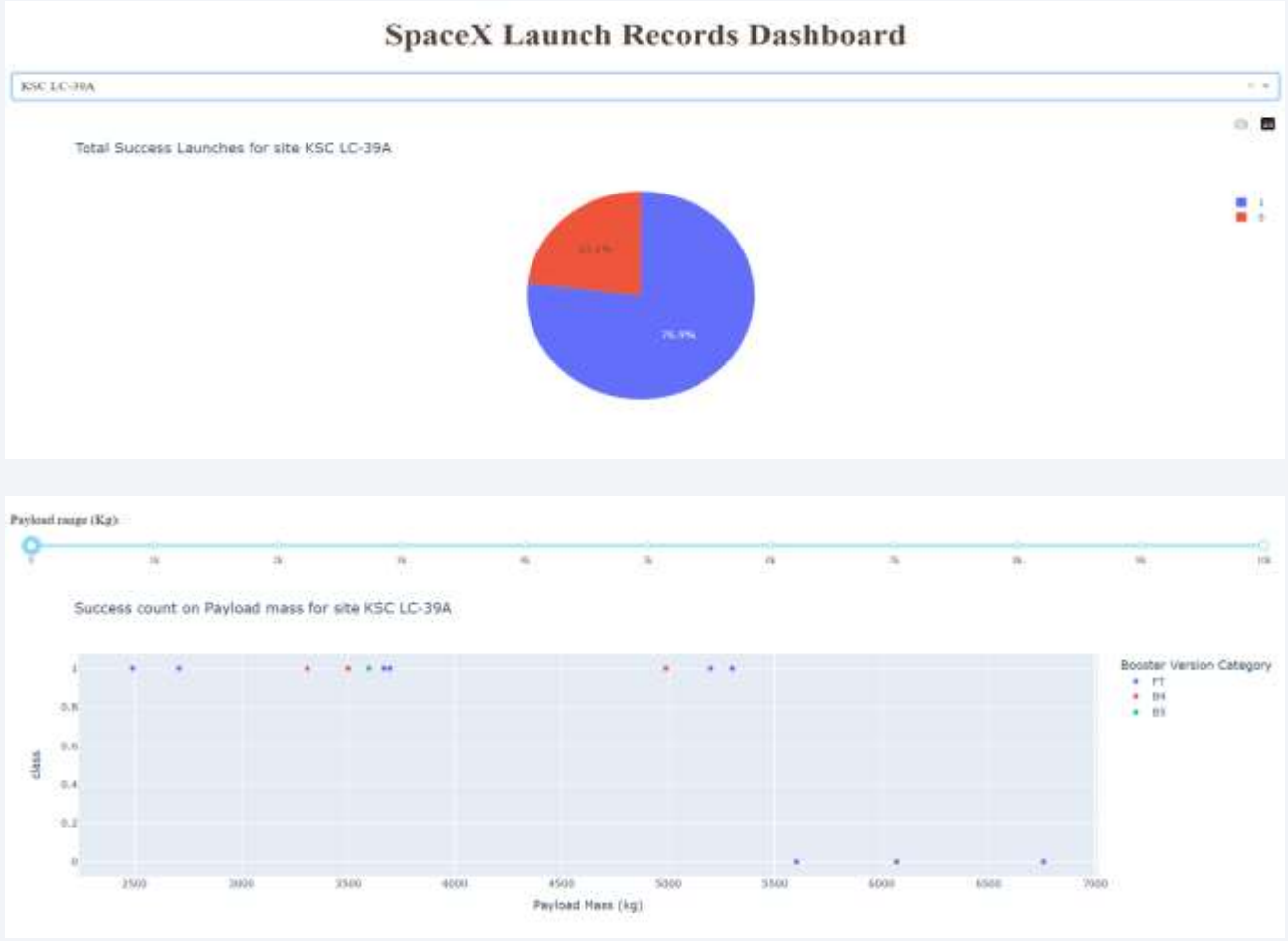
- Dropdown menu for selecting launch sites
- Pie charts displaying success rate.
- Scatter chart displaying launch site, payload mass, success/failure
- Range slider for selecting range of payload mass in kg.



Plotly Dashboard

Getting the following information by analyzing:

- Site with largest successful launches.
- Site with highest launch success rate
- Payload range(s) with highest launch success rate
- Payload range(s) with lowest launch success rate
- F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) with highest launch success rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

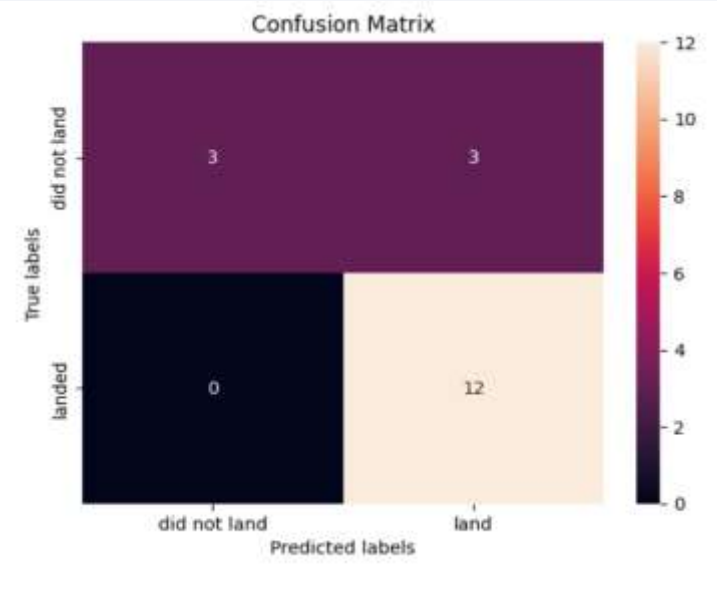
The final results shows in two column one is of predicted method column while other is Test Data Accuracy.

- Total 4 predictive methods use which is Logistic Regression, SVM, Decision Tree and KNN.
- Decision Tree shows more accuracy which is 0.88 as compare other three which 0.83.

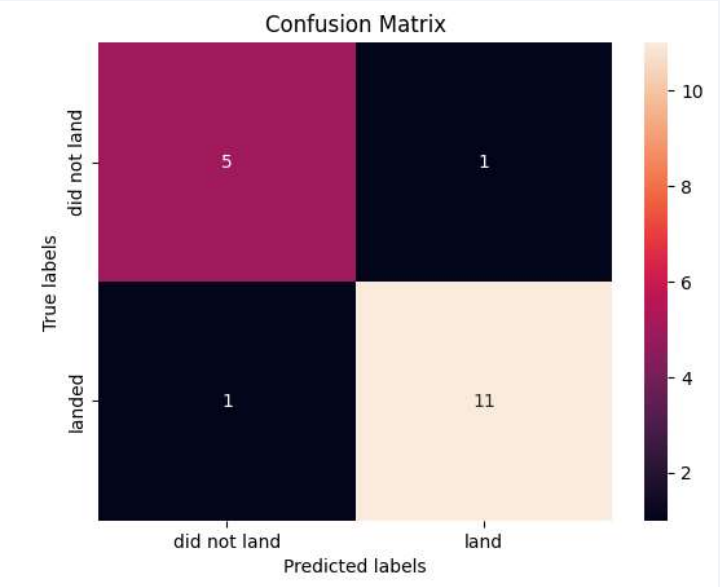
	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333

Confusion Matrix

- Decision Tree is the model predicted more accuracy than other three.
- Other predicted about unsuccessful landings were 100% correct while decision tree prediction about successful landing is more accurate than others.



Confusion matrix of KNN, Logistics Regression and SVM



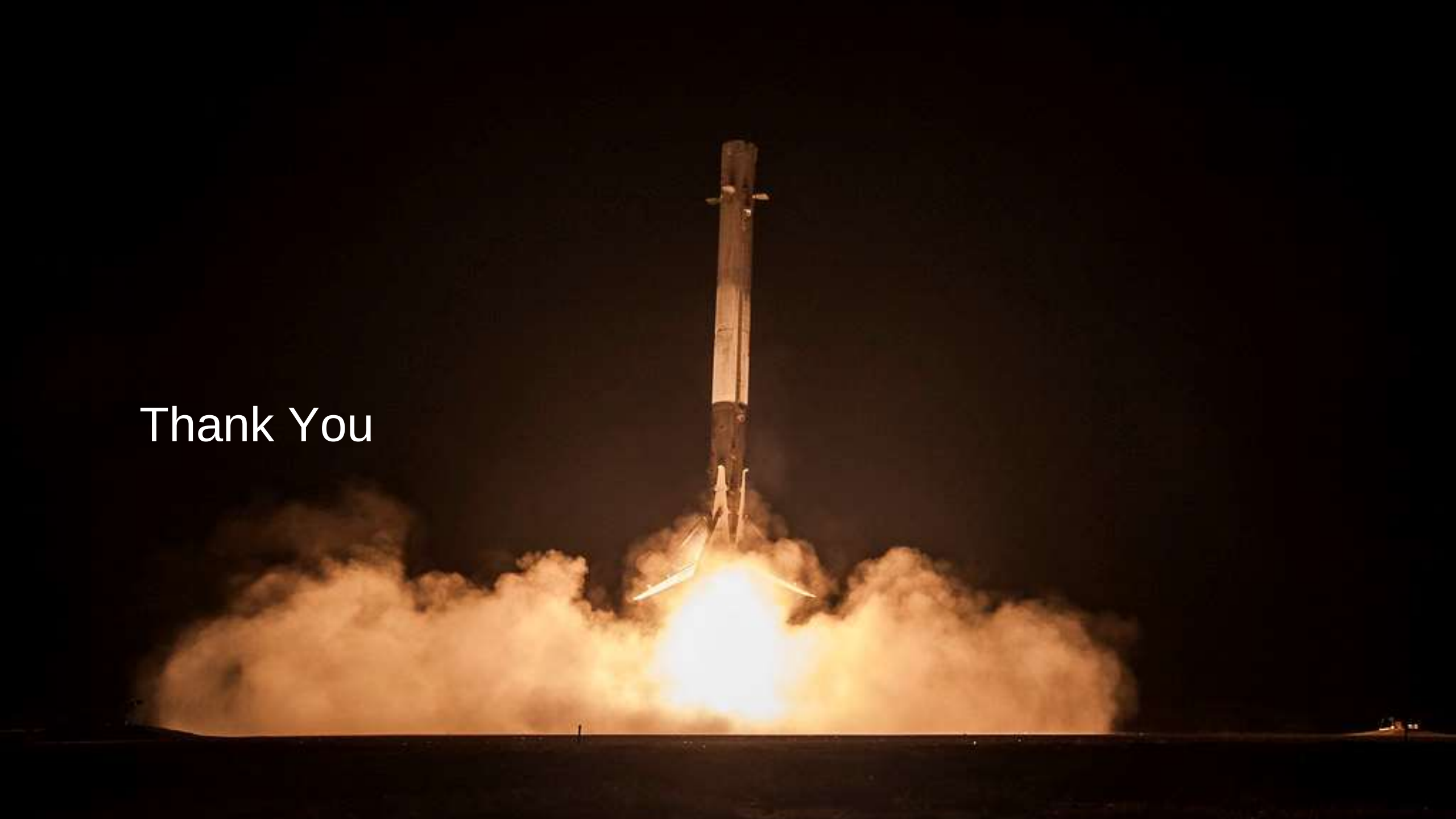
Confusion matrix of decision tree

Conclusions

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- We can deduce that, as the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight
- If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.
- 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

Conclusion Cont.

- 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(unsuccessful mission) are both there here
- A finally the success rate since 2013 kept increasing till 2020.


Thank You