IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Soufleros Konstantinos
15/01/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data collection via API, SQL and Web Scraping
  - Data Wrangling and Analysis
  - Interactive Maps with Folium
  - Predictive Analysis for each classification model
- Summary of all results
  - Data Analysis along with Interactive Visualizations
  - Best Model for Predictive Analysis

Introduction

- Project background and context

In this project we will predict if the Falcon 9 first stage will land successfully. SpaceX is a revolutionary company who has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving thanks to SpaceX astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price down even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

- Problems we want to find answers
  - Identifying all factors that influence the landing outcome.
  - The relationship between each variables and how it is affecting the outcome.
  - The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

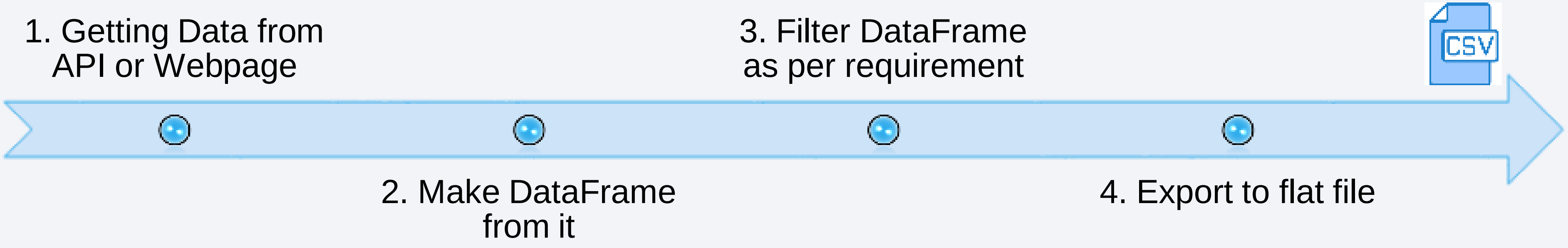
- Data collection methodology:
    - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
    - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Build and evaluate classification models

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia

For REST API, we started by using the get request. Then, we decoded the response content as .json and turn it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we used the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis.



Data Collection – SpaceX API

Getting response from API

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

Converting response to .json file

- jlist=requests.get(static_json_url).json()
- df2=pd.json_normalize(jlist)
- df2.head()

Apply custom functions to clean data

- getBoosterVersion(data)
- getLaunchSite(data)
- getPayloadData(data)
- getCoreData(data)

Assign list to dictionary then create dataframe

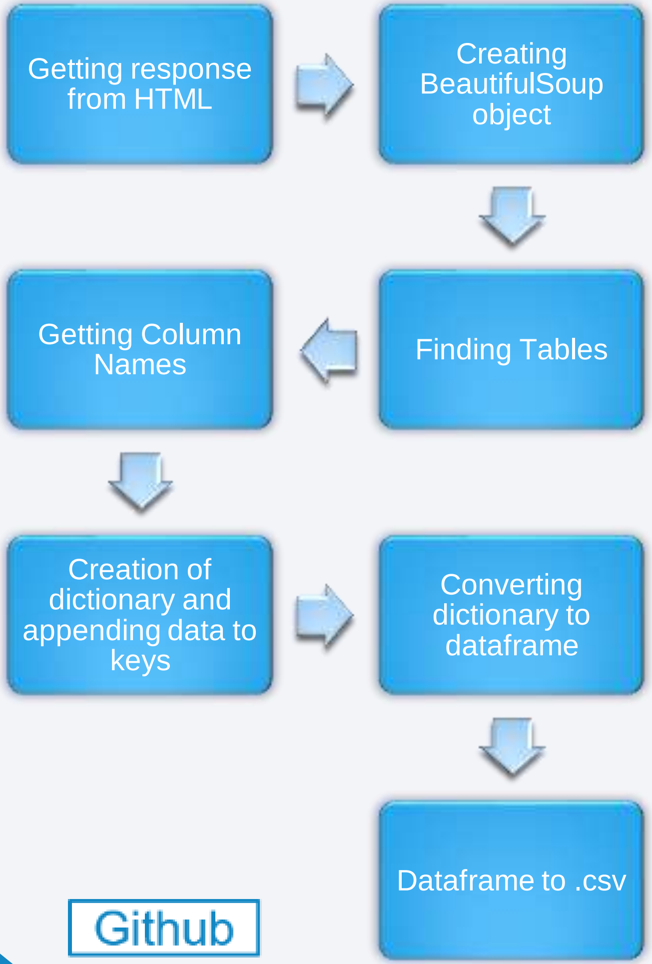
- launch_dict = {'FlightNumber': list(data['flight_number']),'Date': list(data['date']),'BoosterVersion':BoosterVersion,'PayloadMass':PayloadMass,'Orbit':Orbit,'LaunchSite':LaunchSite,'Outcome':Outcome, 'Flights':Flights,'GridFins':GridFins,'Reused':Reused,'Legs':Legs,'LandingPad':LandingPad,'Block':Block,'ReusedCount':ReusedCount,'Serial':Serial,'Longitude': Longitude,'Latitude': Latitude}
- data = pd.DataFrame({key:pd.Series(value) for key, value in launch_dict.items()})

Filter dataframe and export to flat file

- data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
- data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
- data_falcon9.to_csv('dataset_part_1.csv', index=False)

Github

Data Collection - Scraping



Github

```
data  = requests.get(static_url).text

soup = BeautifulSoup(data, 'html5lib')

html_tables=soup.find_all("table")

column_names = []
ths = first_launch_table.find_all('th')
for th in ths:
    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)

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

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.0B0007.1	No attempt	1 March 2013	15:10

Data Wrangling

Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type. We then create a landing outcome label from the outcome column.

This will make it easier for further analysis, visualization and ML. Lastly, we will export the result to a CSV.

Calculate the number of launches on each site

df['LaunchSite'].value_counts()

Calculate the number and occurrence of each orbit

df['Orbit'].value_counts()

Calculate the number and occurence of mission outcome per orbit type

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

Creating a 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['C

Export dataset to .csv

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

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

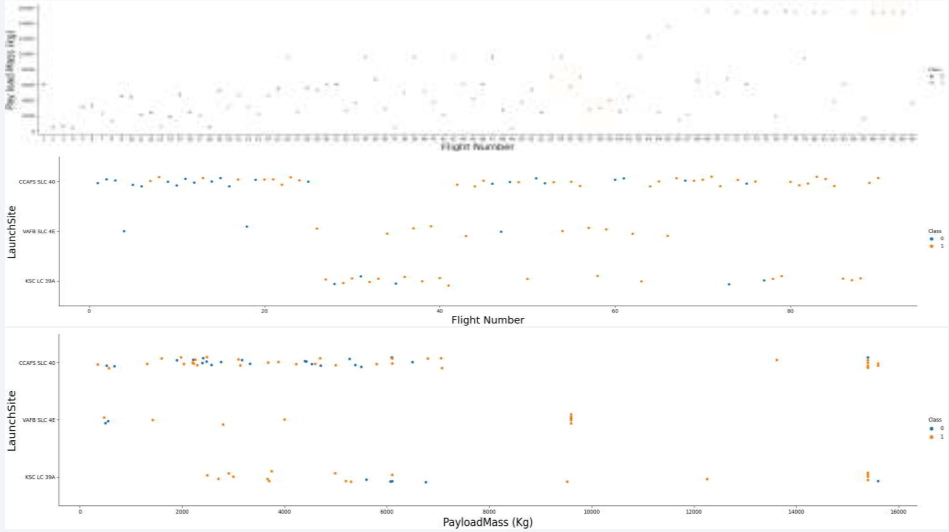
Github

EDA with Data Visualization

Scatter Graphs Drawn

- Payload and Flight Number
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit Type
- Payload and Orbit Type.

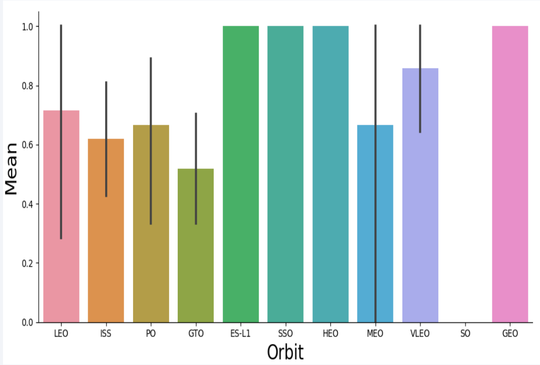
Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.



Bar Graph Drawn

- Success Rate vs Orbit

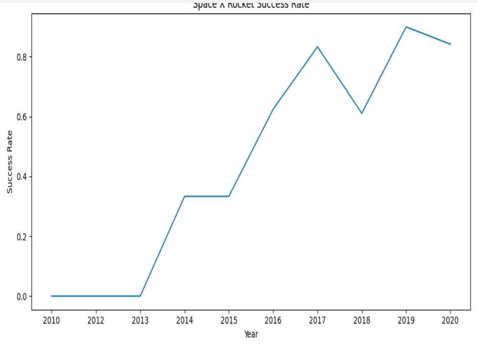
Bar graphs is one of the easiest way to interpret the relationship between the attributes. In this case, we will use the bar graph to determine which orbits have the highest probability of success.



Line Graph Drawn

- Launch Success Yearly Trend

Line graphs are useful for showing trends and make prediction for unseen data. In this case, we will use the line graph to observe the launch success yearly trend.



Github

EDA with SQL

Using SQL, we performed many queries to get better understanding of the dataset:

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string ‘CCA’.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.
- Rank the count of landing outcomes of success between the date 2010-06-04 and 2017-03-20, in descending order.

Github

Build an Interactive Map with Folium

Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. In our case we took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.

We then assigned the dataframe launch_outcomes(failure, success) to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().

We then calculated the distance of the launch sites to various landmarks to find answers to the questions such as:

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?

Map Objects	Code	Result
Map Marker	folium.Marker()	Map object to make a mark on map
Icon Marker	folium.Icon()	Create an icon on map
Circle Marker	folium.Circle()	Create a circle where Marker is placed
Polyline	folium.Polyline()	Create a line between points
Marker Cluster Object	MarkerCluster()	Simplify a map containing many markers having the same coordinates

Github

Build a Dashboard with Plotly Dash

In the Dashboard we included two interactive graphs:

- A Pie Graph, showing the total success for all sites or for specific launch site. We included this graph to show the percentage of success in relation to launch site.
- A Scatter Graph, showing the correlation between Payload and Success for all sites or for specific launch site. We included this graph to show the relationship between Success Rate and Booster Version Category.

Github

Predictive Analysis (Classification)

Building Model

- Load our dataset into dataframe.
- Normalize the data and then split into training and test datasets.

Model preparation

- Selection of machine learning 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
- The model with the best accuracy will be chosen

Github

Results

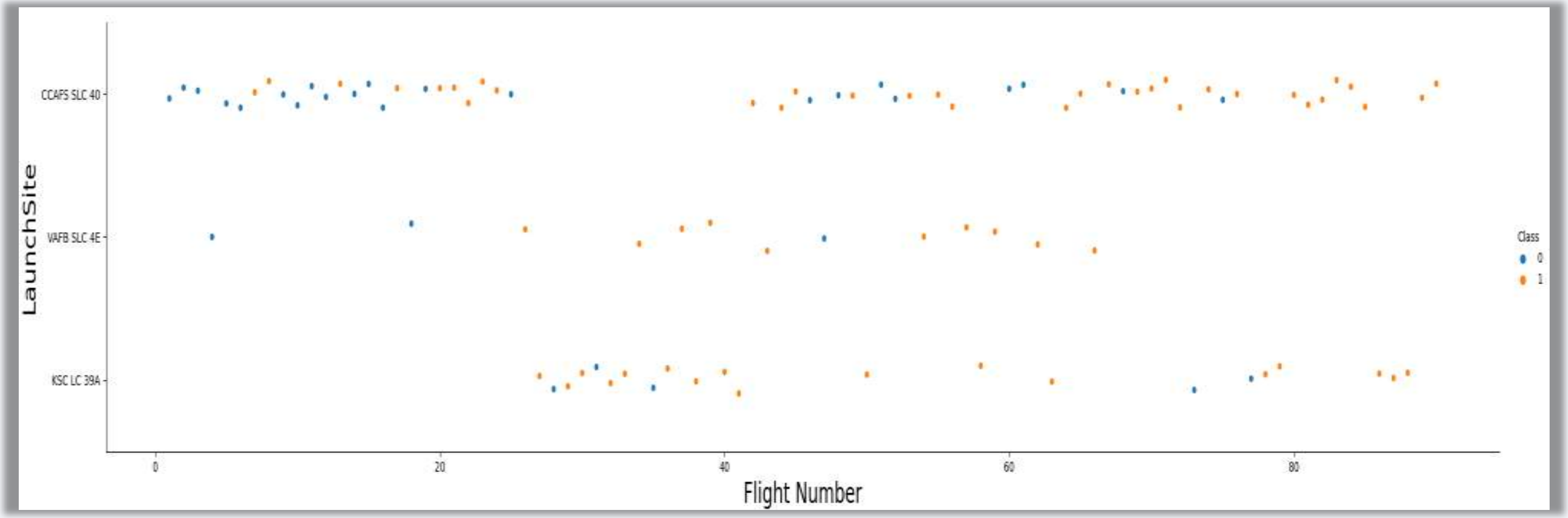
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

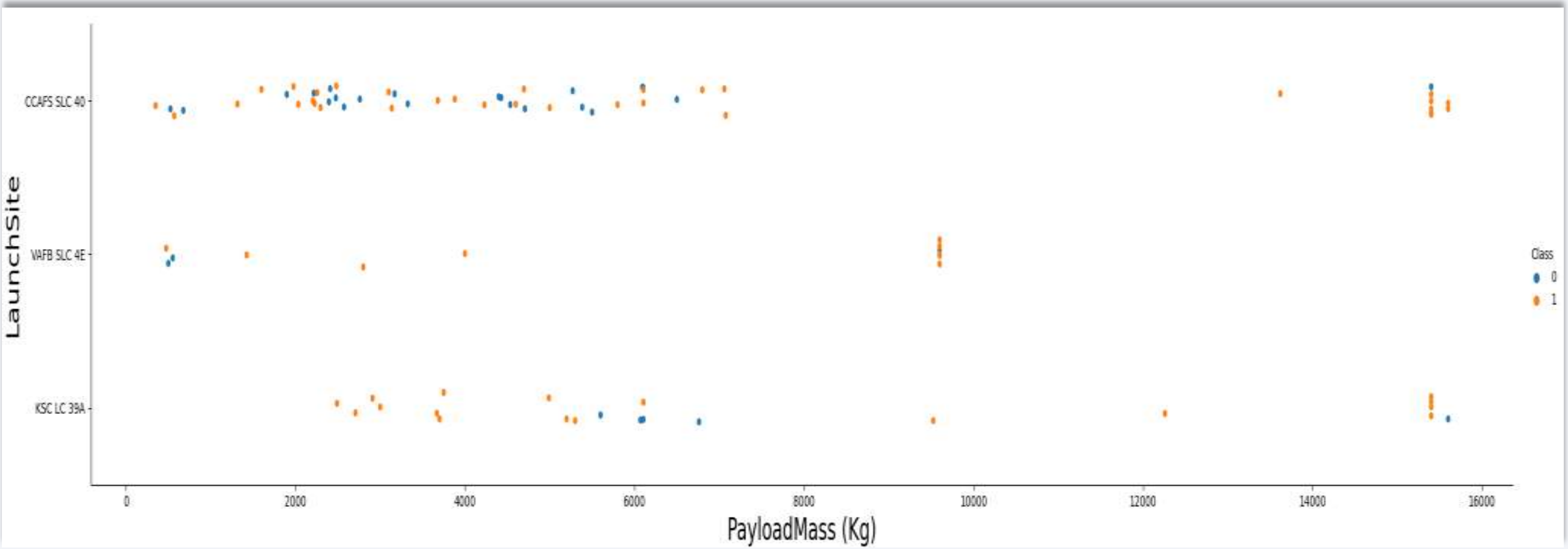
Flight Number vs. Launch Site

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



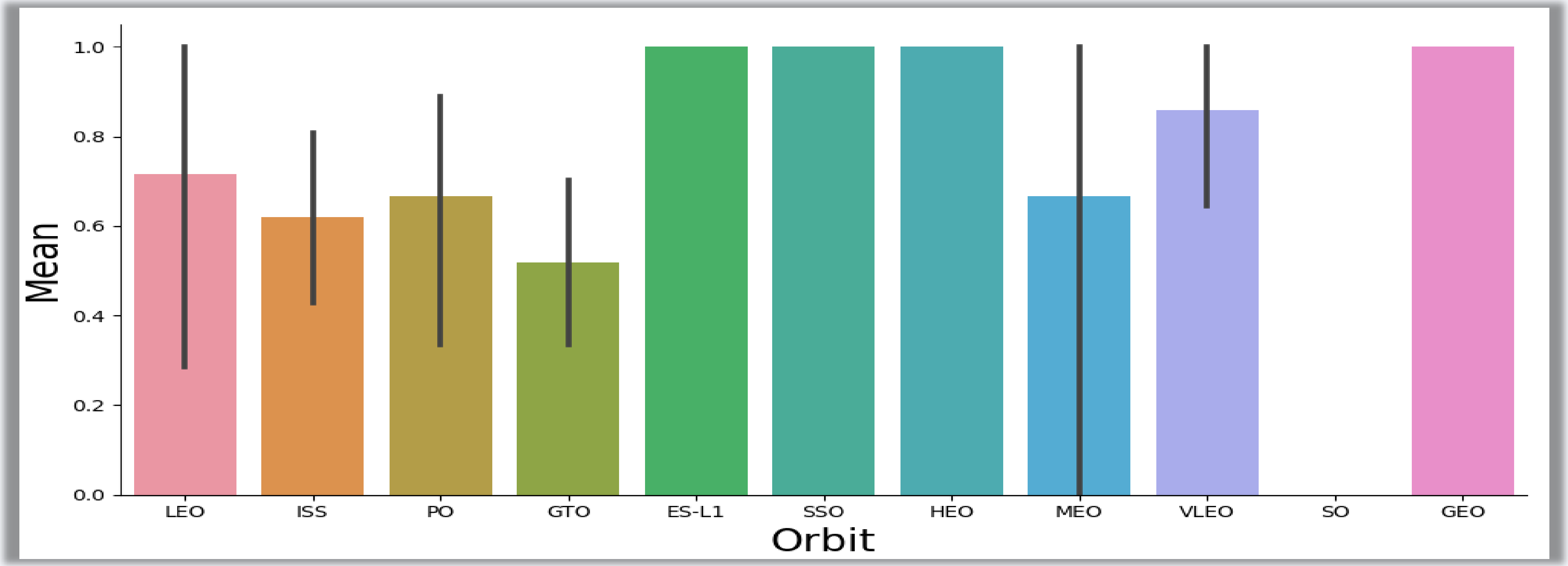
Payload vs. Launch Site

This scatterplot shows the greater the payload mass (>7000 kg) the higher the success rate for the Rocket. But there is no clear pattern to take a decision if the launch site is dependent on Payload Mass for a successful launch.



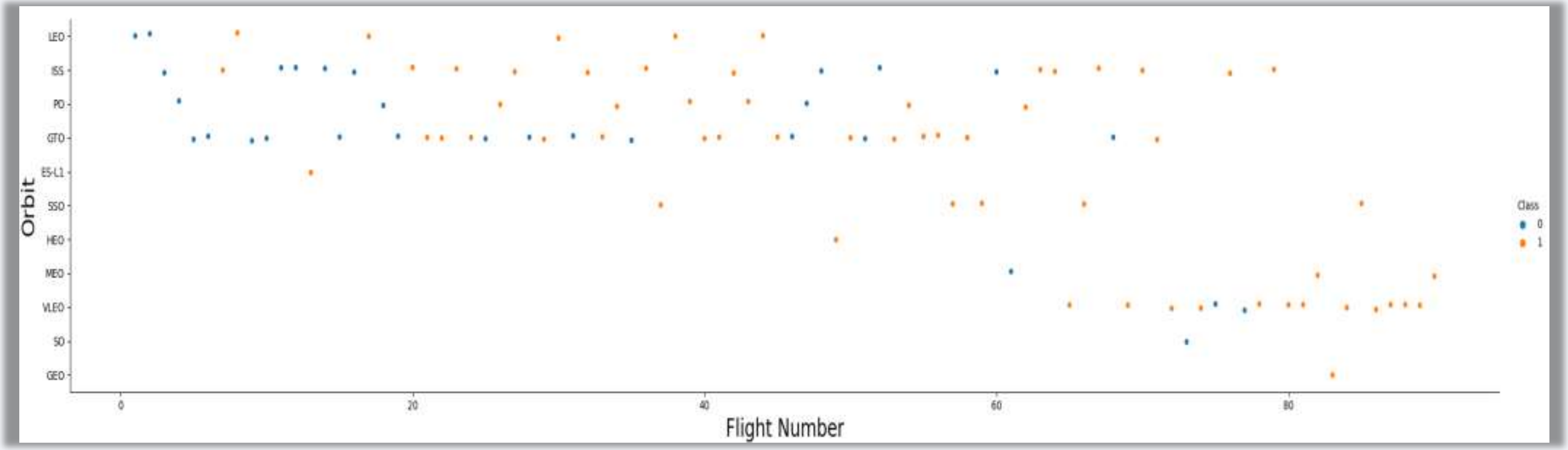
Success Rate vs. Orbit Type

This bar chart shows the success rate of each orbit type. ES-L1, SSO, HEO, GEO have the highest success rates.



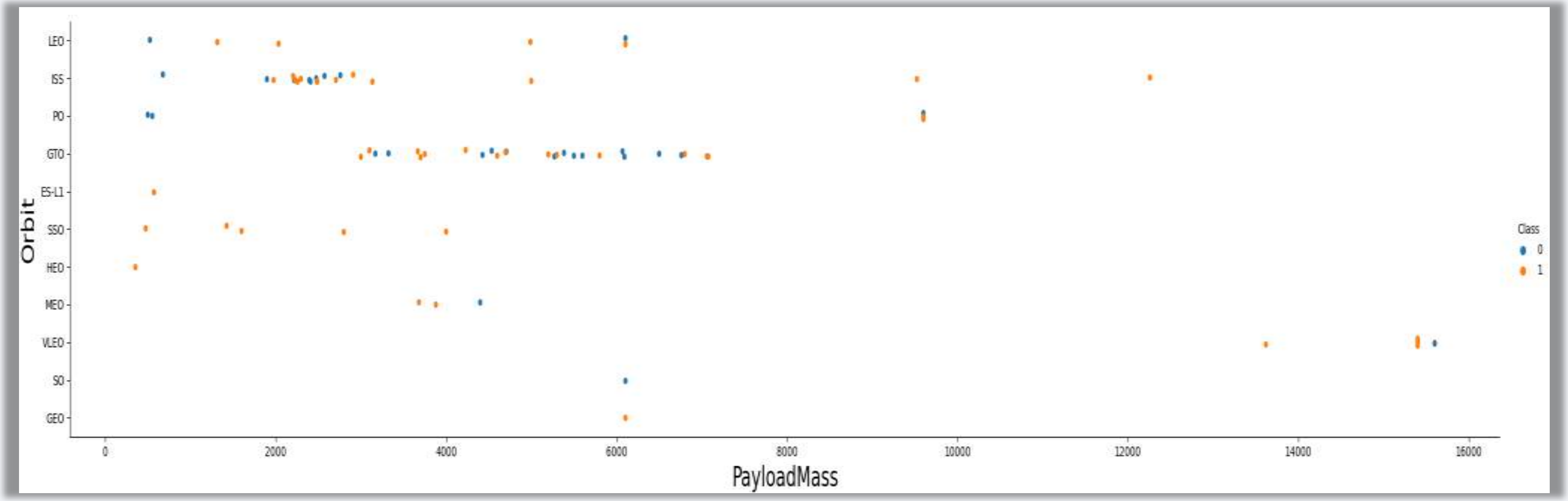
Flight Number vs. Orbit Type

This scatterplot shows 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 flights.



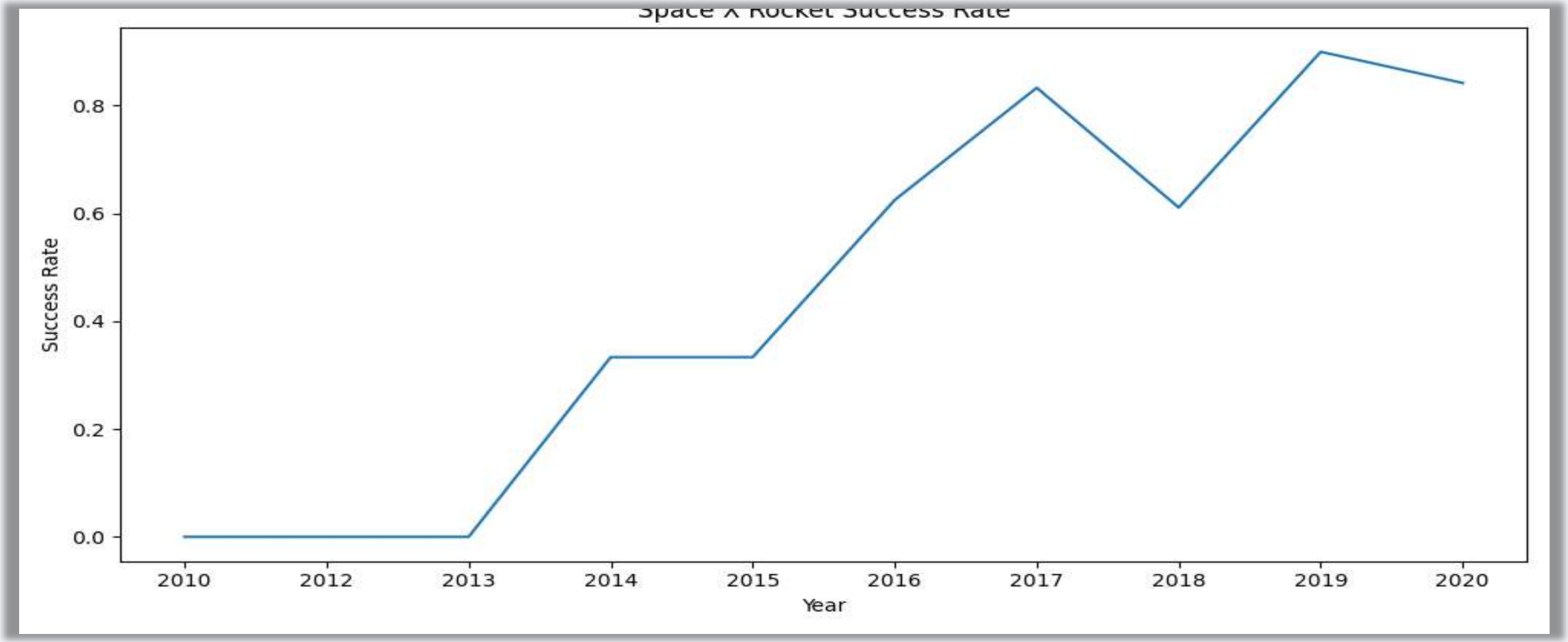
Payload vs. Orbit Type

This scatterplot shows that the weight of the payloads can have great influence on the success rate of the launches in certain orbits. Heavier payloads improve the success rate for the LEO orbit. Also, decreasing the payload weight for the GTO orbit improves the success of the launch.



Launch Success Yearly Trend

- This line chart of yearly average success rate shows that the success rate since 2013 kept increasing relatively, having only a dip during 2018.



All Launch Site Names

SQL QUERY

```
%sql SELECT DISTINCT Launch_Site as "Launch_Sites" FROM SPACEXTBL;
```

- The word DISTINCT in the query pulls the unique values for the Launch_Site column from the table SPACEXTBL.

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

Launch Site Names Begin with 'CCA'

SQL QUERY

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

- We used the keyword ‘LIMIT 5’ to fetch 5 Launch_Site from the table SPACEXTBL which begin with ‘CCA’.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

SQL QUERY

%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';

- The SUM function calculates the total in the column PAYLOAD_MASS_KG_ and WHERE clause filters the data to fetch Customer by name “NASA(CRS)

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

SQL QUERY

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEXTBL \
WHERE Booster_Version = 'F9 v1.1';
```

- The function AVG fetches the average of the column PAYLOAD_MASS_KG_ and WHERE clause filters the dataset to only perform calculations on Booster_version “F9 v1.1”.

Average Payload Mass by Booster Version F9 v1.1
2928.4

First Successful Ground Landing Date

SQL QUERY

%sql SELECT MIN(DATE) AS "First Successful Landing Outcome in Ground Pad" FROM SPACEXTBL \

WHERE [Landing _Outcome] = 'Success (ground pad)';

- MIN function converts the DATE column into minimum date and WHERE clause filters the data to perform calculations on Landing_Outcome with values “Success (ground pad)”.

First Succesful Landing Outcome in Ground Pad
01-05-2017

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;
```

- Selecting only Booster_Version and WHERE clause filters the dataset to Landing_Outcome = Success(drone ship) AND filters additionally for Payload_Mass_KG_ >4000 and <6000.

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

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

```
%sql SELECT COUNT(Mission_Outcome) AS "Successful Mission"
FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Success%’;

%sql SELECT COUNT(Mission_Outcome) AS "Failure Mission"
FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%';
```

- Wilcard % is used to filter with WHERE clause for Mission_Outcome either for success or a failure.

Successful Mission
100

Failure Mission
1

Boosters Carried Maximum Payload

SQL QUERY

%sql SELECT DISTINCT Booster_Version AS "Booster Versions with Maximum Payload Mass" FROM SPACEXTBL \

WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

- MAX function is used for the maximum payload in the column PAYLOAD_MASS_KG_ and WHERE clause filters Booster_Version that had the maximum payload.

Booster Versions with Maximum Payload Mass
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';
```

- The combination of the WHERE clause, LIKE,AND and BETWEEN operators to filter for failed outcomes in drone ship, their booster versions and launch site names for the year 2015.

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

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

SQL QUERY

```
%sql SELECT [Landing _Outcome], COUNT([Landing _Outcome]) AS Number 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 ;
```

- Selecting LANDING_OUTCOME with WHERE clause to filter the date between '04-06-2010' and '20-03-2017', and searching for successful landing grouping by LANDING_OUTCOME and ordering by COUNT(LANDING_OUTCOME) in descending order.

Landing _Outcome	Number
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

Launch Sites Proximities Analysis

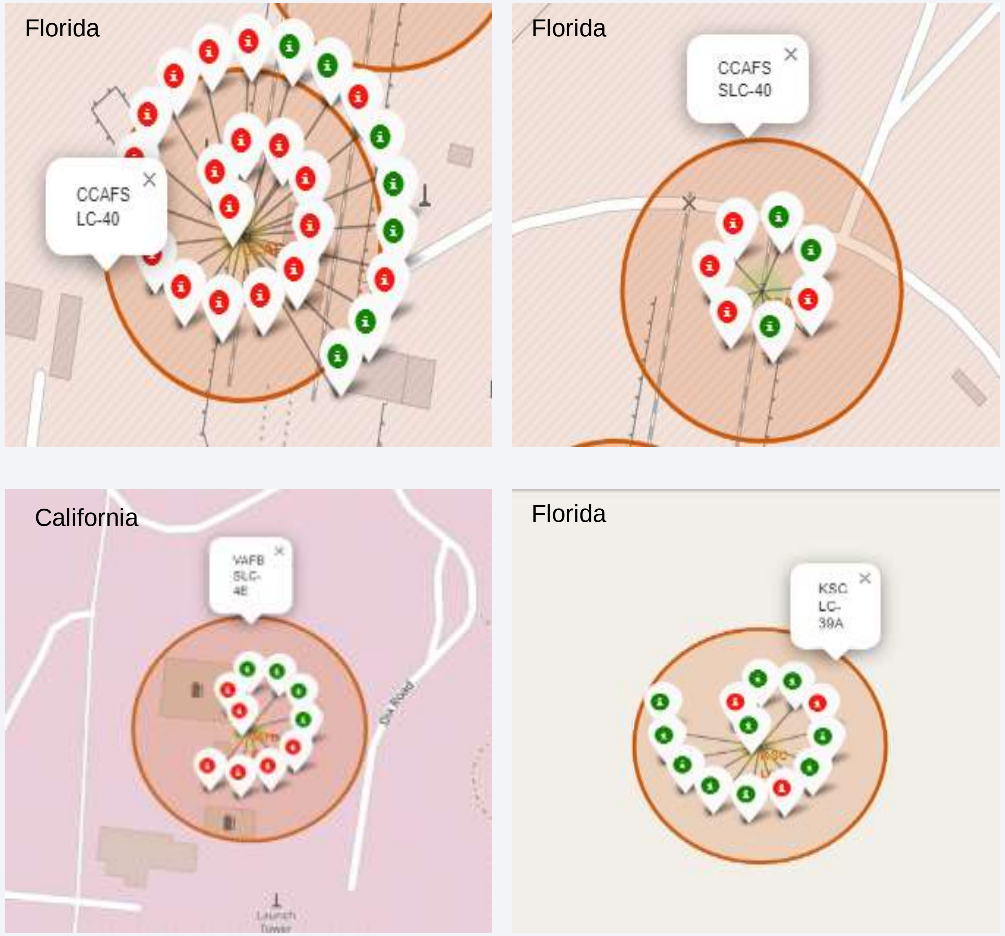
Location of all the Launch Sites

We can see that all the SpaceX launch sites are located inside the United States and more specifically in the coastlines of California and Florida.

Color Labeled Launch Records

- Green marker shows successful launches and Red marker shows failures.

- Here we notice that KSC LC-39A has the most successful launches.



Launch Sites Distance From Landmarks

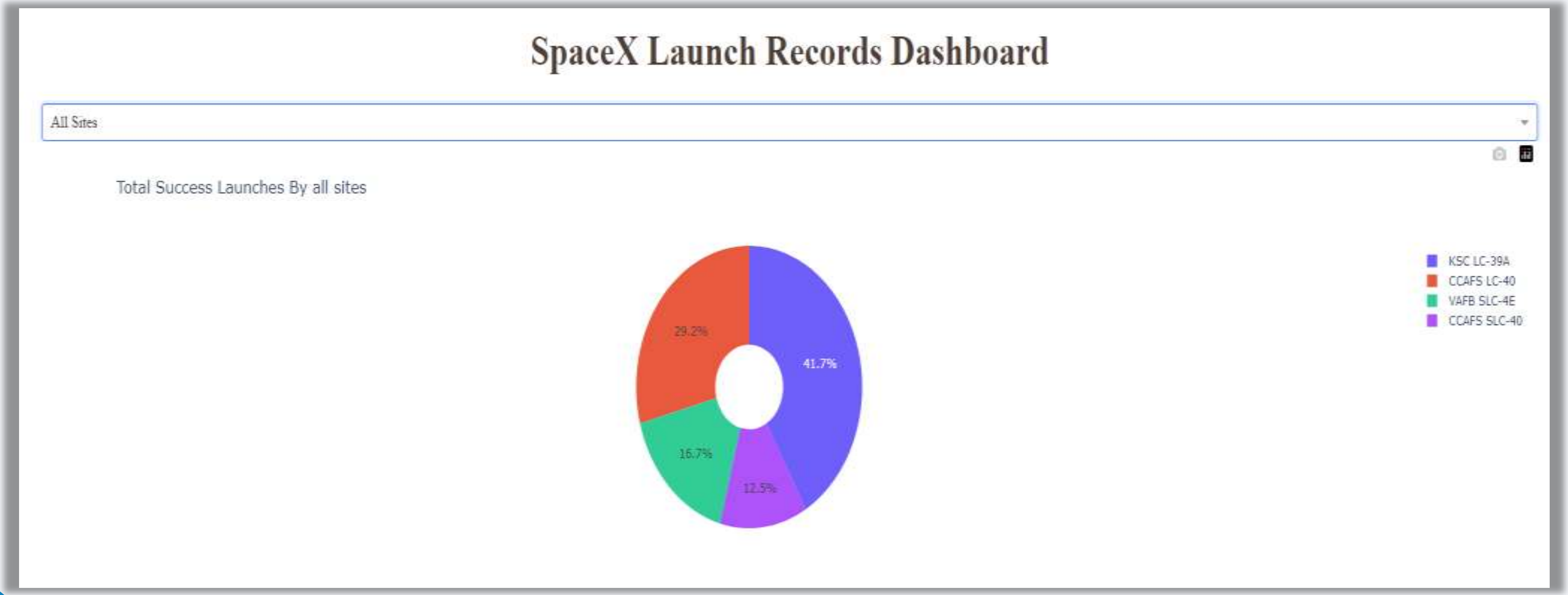
- Are launch sites in close proximity to railways? **NO**
- Are launch sites in close proximity to highways? **NO**
- Are launch sites in close proximity to coastline? **YES**
- Do launch sites keep certain distance away from cities? **YES**

Section 4

Build a Dashboard with Plotly Dash

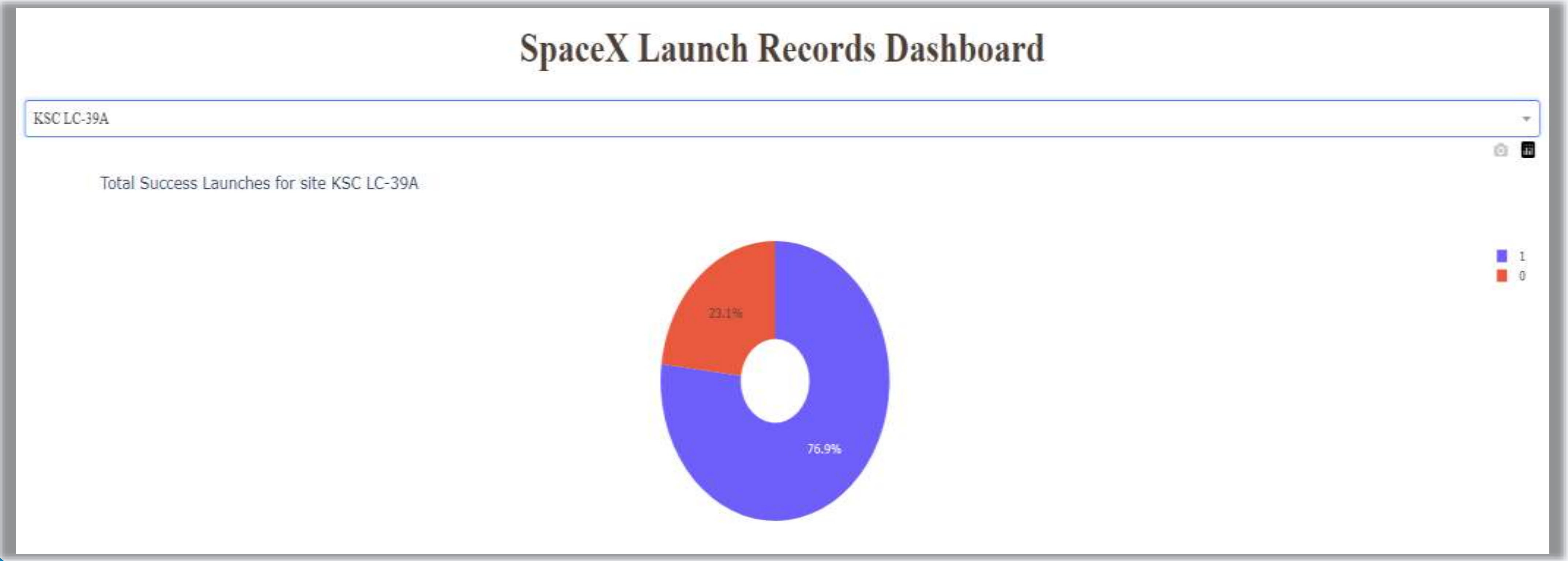
Launch Success Percentage for All Sites

Here, we notice clearly that KSC LC-39A had the most successful launches from all the sites.



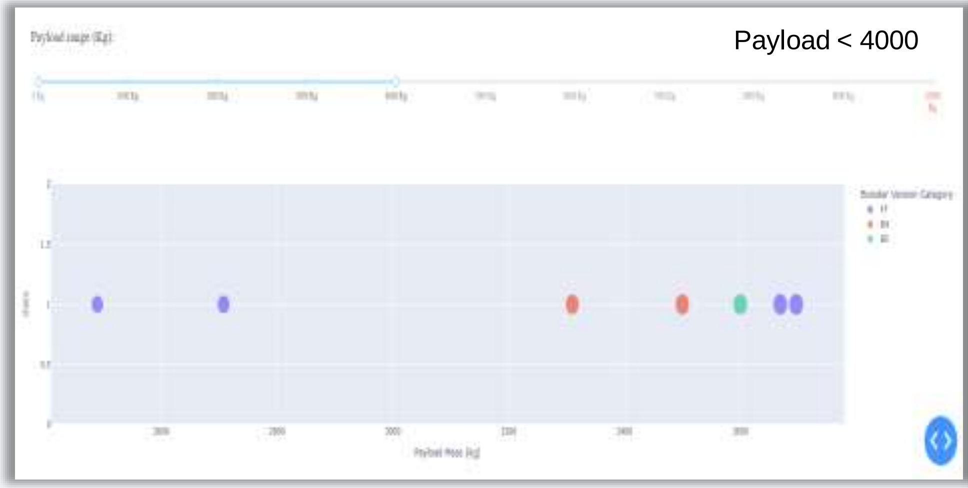
Launch Site With Highest Launch Success Ratio

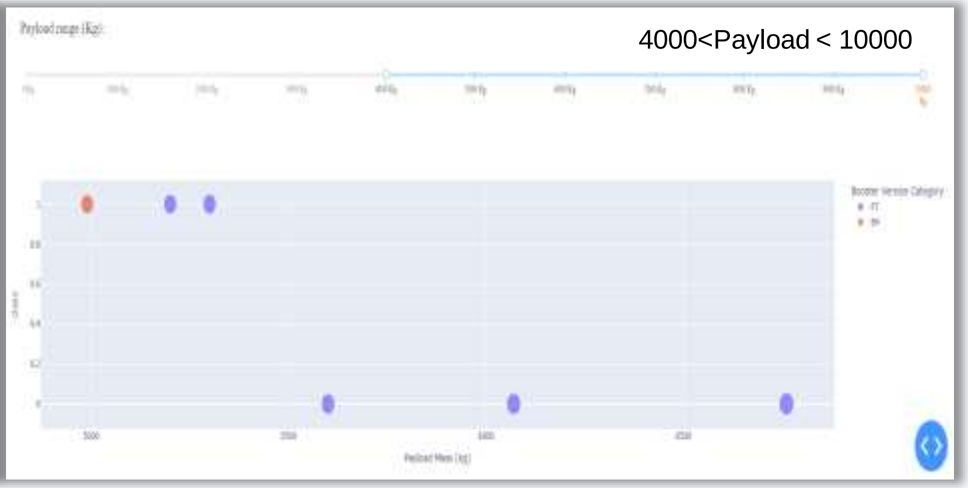
KSC LC-39A achieved 76.9% success rate while getting 23,1% failure rate.



Payload vs Launch Outcome Scatter plot

The success rate for low weighted payload (0-4000kg) is higher than heavy weighted payload (4000-10000kg).





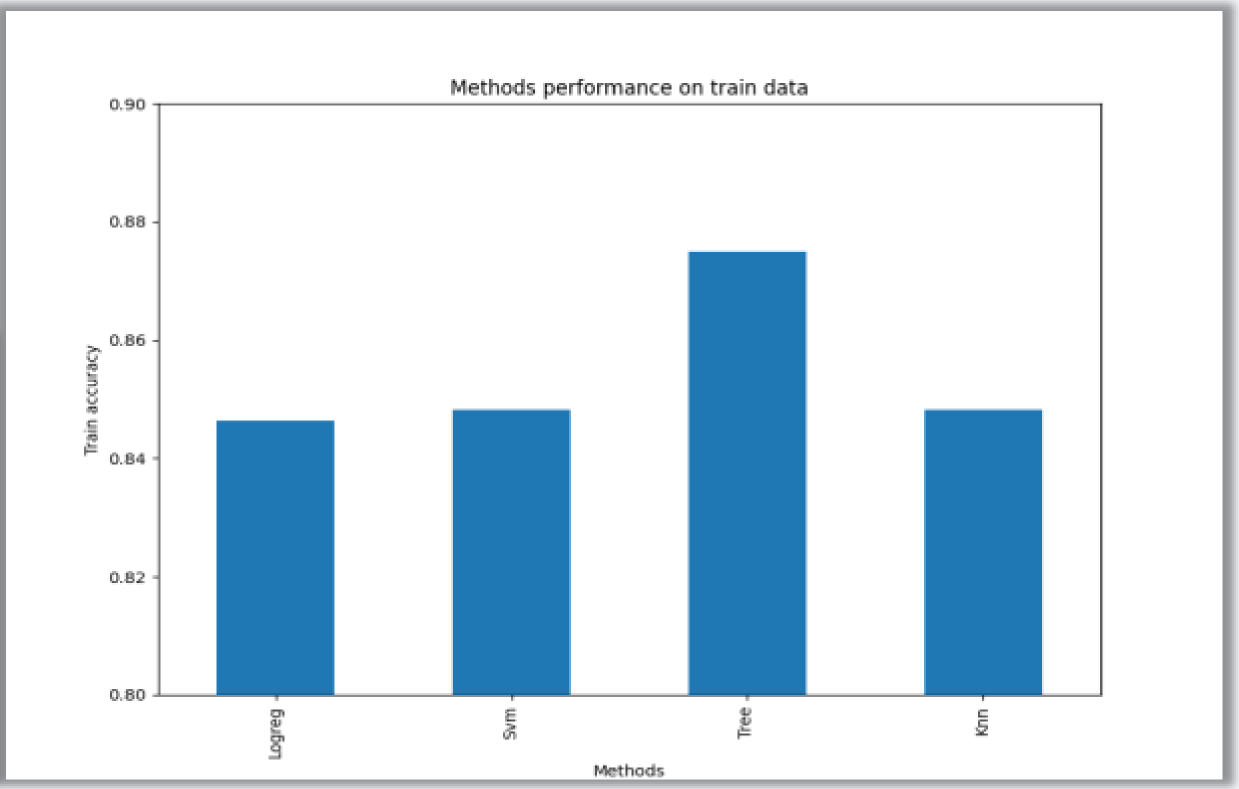
Section 5

Predictive Analysis (Classification)

Classification Accuracy

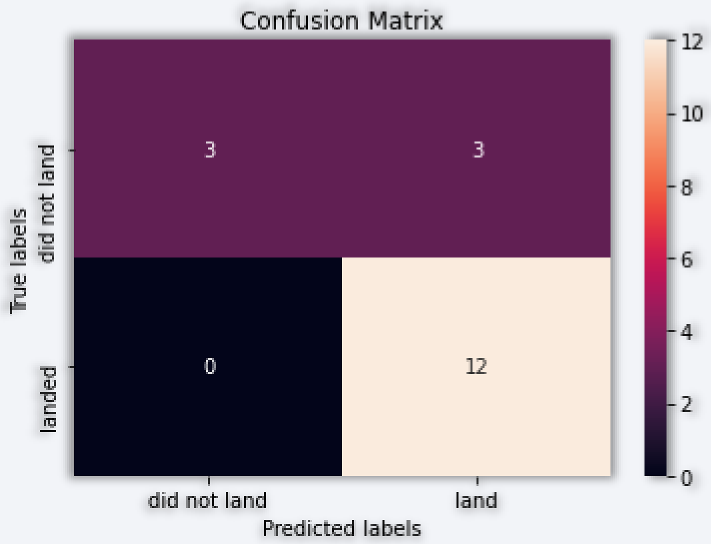
Accuracy is very close to all the models, but we can identify that the best algorithm is the Decision Tree Algorithm which has the highest accuracy 0.875000.

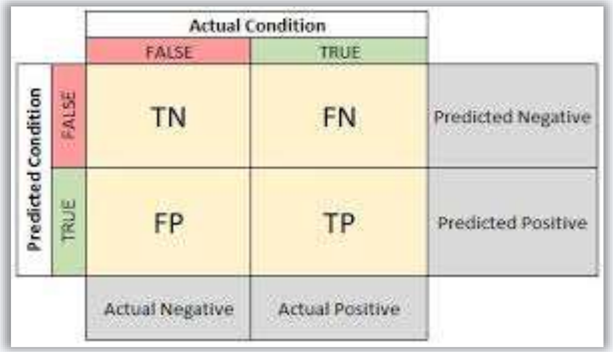
	Accuracy Train	Accuracy Test
Tree	0.875000	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



Confusion Matrix

The confusion matrix for the Decision Tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





Conclusions

- Orbits ES-L1, GEO, HEO, SSO have the highest success rates.
- Success rates for SpaceX launches have been increasing relatively with time.
- KSC LC-39A launch site has the most successful launches but increasing payload mass seems to have negative impact on success.
- Decision Tree Classifier Algorithm is the best for Machine Learning Model for the provided dataset.

Thank you!