WINNING SPACE RACE WITH DATA SCIENCE

BY : RITU SINGH

25 SEPT 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Interactive map with Folium
- Interactive Dashboard with Plotly Dash
- Predictive Analysis (Classification)

Summary of results

- Exploratory data analysis (EDA)
- Interactive analytics demo in screenshots
- Predictive analysis

Introduction

Project background and context

We predict if the Falcon 9 first stage will land successfully. 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.

Problems we want to find answers

- **What factors make launch successful.**
- **What factors affect launch sites success rate.**
- **Relationship between different variable and how they affect launch success rate.**


Methodology

Methodology

Executive Summary

- Data collection methodology: It can be done by two method :

➢ By make a get request to the SpaceX API. And clean up the requested data.

➢ By web scraping from Wikipedia.

- Perform data wrangling
    - Data was cleaning, organizing, and transforming raw data into the desired format for analysts to use for prompt decision-making.
- 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

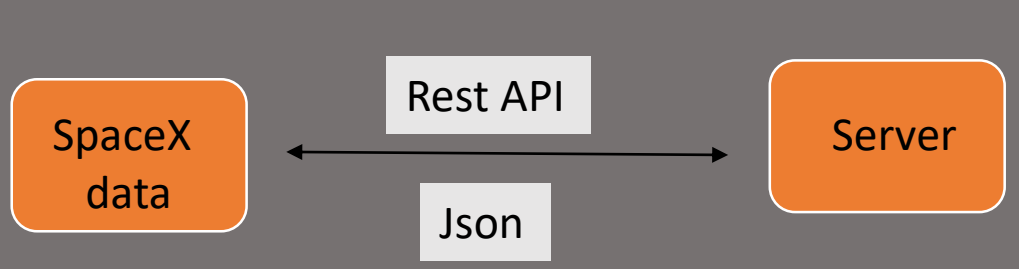
Data Collection

- We would like to get a SpaceX data to predict whether Falcon 9 first stage will land successfully or not. For that 1st we do data collection by following steps :

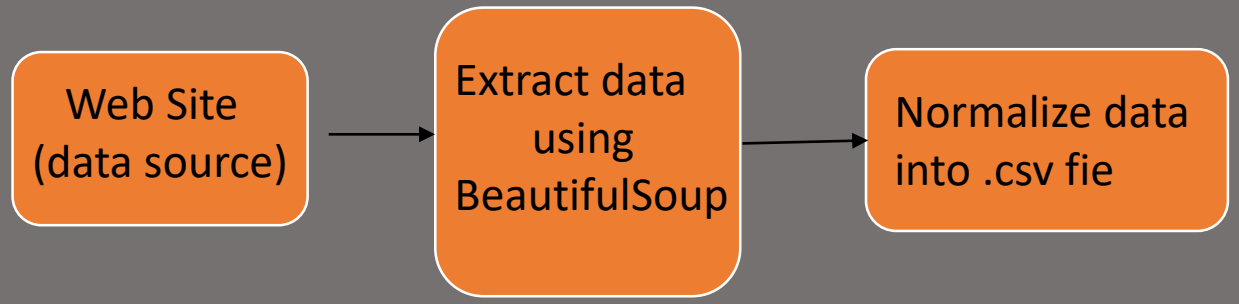
(1) We start with requesting rocket launch data from SpaceX REST API with a URL : https://api.spacexdata.com/v4/launches/past.

(2) After converting the data into suitable format ,this API gives information about booster name, payload, launch site, landing outcomes, No. of flights, type of landing , about core and etc.

(3) Another method by which we can obtain SpaceX data is by web scraping Wikipedia using BeautifulSoup.



SpaceX API



Web Scraping

Data Collection – SpaceX API

(1) Request rocket launch data from SpaceX API :

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

(2) Decode the response content as a Json file :

```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url)
data = pd.json_normalize(response.json())
```

(3) Clean data by helper function :

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

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

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

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

(4) Create new 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(launch_dict)
data
```

(5) Filter the dataframe to only include Falcon 9 launch

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

(6) Convert file into .csv format :

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

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/5b1a0146b82db58339fc2988837dc475f2d11601/spacex-data-collection-api.ipynb

Data Collection - Scraping

(1) HTTP GET method to request the Falcon9 Launch HTML page :

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

(2) Create a BeautifulSoup object from the HTML response :

```
soup = BeautifulSoup(response, 'html.parser')
```

(3) Extract table :

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

(4) Extract column name :

```
cols = first_launch_table.find_all('th')
for col in cols:
        name = extract_column_from_header(col)
        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) Create Dataframe from dictionary :

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

(7) Convert Dataframe file into .csv file :

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

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/5b1a0146b82db58339fc2988837dc475f2d11601/webscraping.ipynb

Data Wrangling

- Data wrangling can be defined as the process of cleaning, organizing, and transforming raw data into the desired format for analysts to use for prompt decision-making.
- With our data we will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means successfully landed to a specific region of the ocean while False Ocean means unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed to a drone ship False ASDS means the mission outcome was unsuccessfully landed to a drone ship. None ASDS and None None these represent a failure to land.
- We will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.

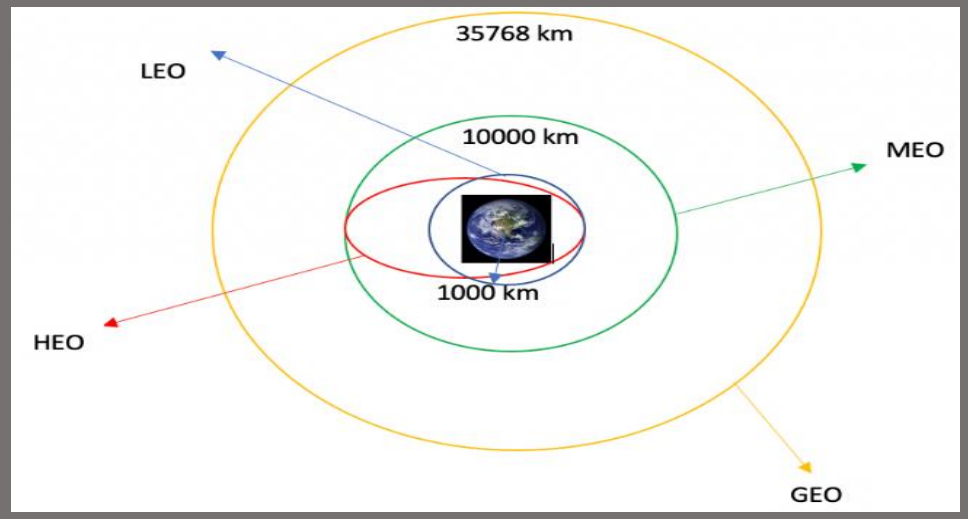


Fig : Each launch aims to an dedicated orbit, and here are some common orbit types

Data Wrangling

- The following steps were taken during data wrangling :

Load Space X dataset
And calculate percentage of missing value .

Calculate the number of launches on each site :
launch_sites=df['LaunchSite'].value_counts()

Calculate the number and occurrence of each orbit :
df['Orbit'].value_counts()

Calculate the number of landing outcomes :
landing_outcomes= df['Outcome'].value_counts()

Create a landing outcome label from Outcome column:

```
landing_class= []
for row in df['Outcome']:
  if row in bad_outcomes:
    landing_class.append(0)
  else:
    landing_class.append(1)
df['Class']=landing_class
df[['Class']].head(8)
```

Calculate success rate of landing_class :
df["Class"].mean()

Export the file into .csv format

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/0b82af432f215898c8eb1bf6630ed0767f5cd197/spacex-Data%20wrangling.ipynb

EDA with Data Visualization

SCATTER PLOT

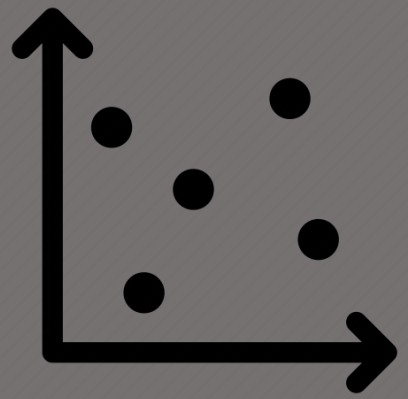
(1) FlightNumber vs PayloadMass

(2) FlightNumber vs LaunchSite

(3) PayLoad vs LaunchSite

(4) FlightNumber vs Orbit

(5) PayLoad vs Orbit



A scatter diagram **visualizes the relationship between two variables and how they are correlated**.

A scatter diagram is one of the best tools for Large data and to show a non-linear pattern .

BAR PLOT

(1) Success Rate vs Orbit

The bar graph **helps to compare the different sets of data among different groups easily**.

LINE PLOT

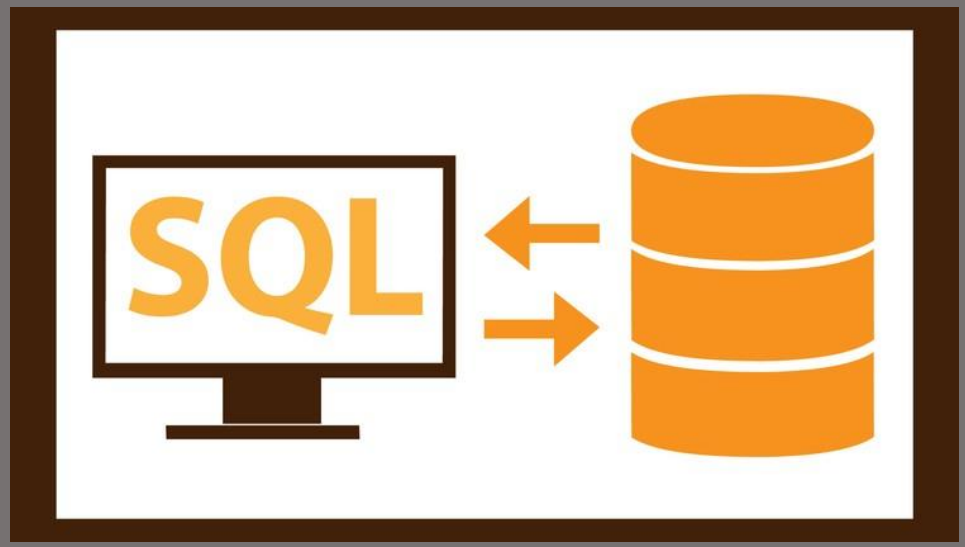
(1) Plot of Launch success yearly trend.

Line graphs are used **to track changes over short and long periods of** time.

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/874803ce50c6ddbe151700719a3d6ea5ebc6860d/eda-dataviz.ipynb

EDA with SQL

- Performed SLQ queries to find insights of dataset.
- Following are the queries asked :



(1) Display the names of the unique launch sites in the space mission.

(2) Display 5 records where launch sites begin with the string 'CCA‘.

(3) Display the total payload mass carried by boosters launched by NASA (CRS)

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

(5) List the date when the first succesful landing outcome in ground pad was acheived.

(6) List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

(7) List the total number of successful and failure mission outcomes

(8) List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

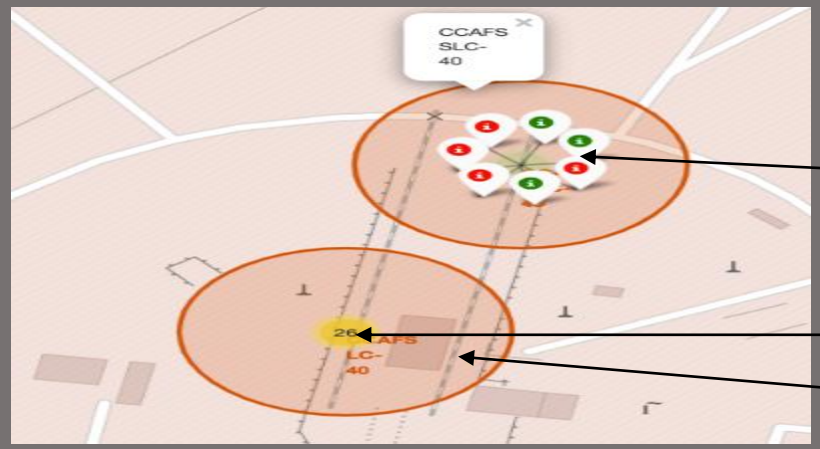
(9) Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

(10) List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/f2adf5c03d8e535f7a319b7b18477aa3b8e39965/eda-sql-coursera_sqllite%20(1).ipynb

Interactive Map with Folium

- The launch success rate depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and we could discover some of the factors by analyzing the existing launch site locations. Hence we perform interactive visual analytics using Folium.
- Folium provides the **folium.Map( )** class which takes site's location parameter in terms of latitude and longitude and generates a map around it.
- **folium.Circle( )** class is used to add a highlighted circle area with a text label on a specific coordinate.
- **folium.Marker( )** class is used to mark specific locations. We used marker for all launch records like green marker for successful launch and red marker for unsuccessful launch.
- **Folium.MarkerCluster( )** class is used to simplify a map containing many markers having the same coordinate.
- Added `**MousePosition**` on the map to get coordinate for a mouse over a point on the map.
- We can use markers to specific certain locations like hospitals , Police stations, schools etc.



https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/aba58282f594d76e6d2c706ee6012d5a33dc3cce/launch_site_location.ipynb

Interactive Dashboard with Plotly Dash

Our dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.

Launch Site Drop-down

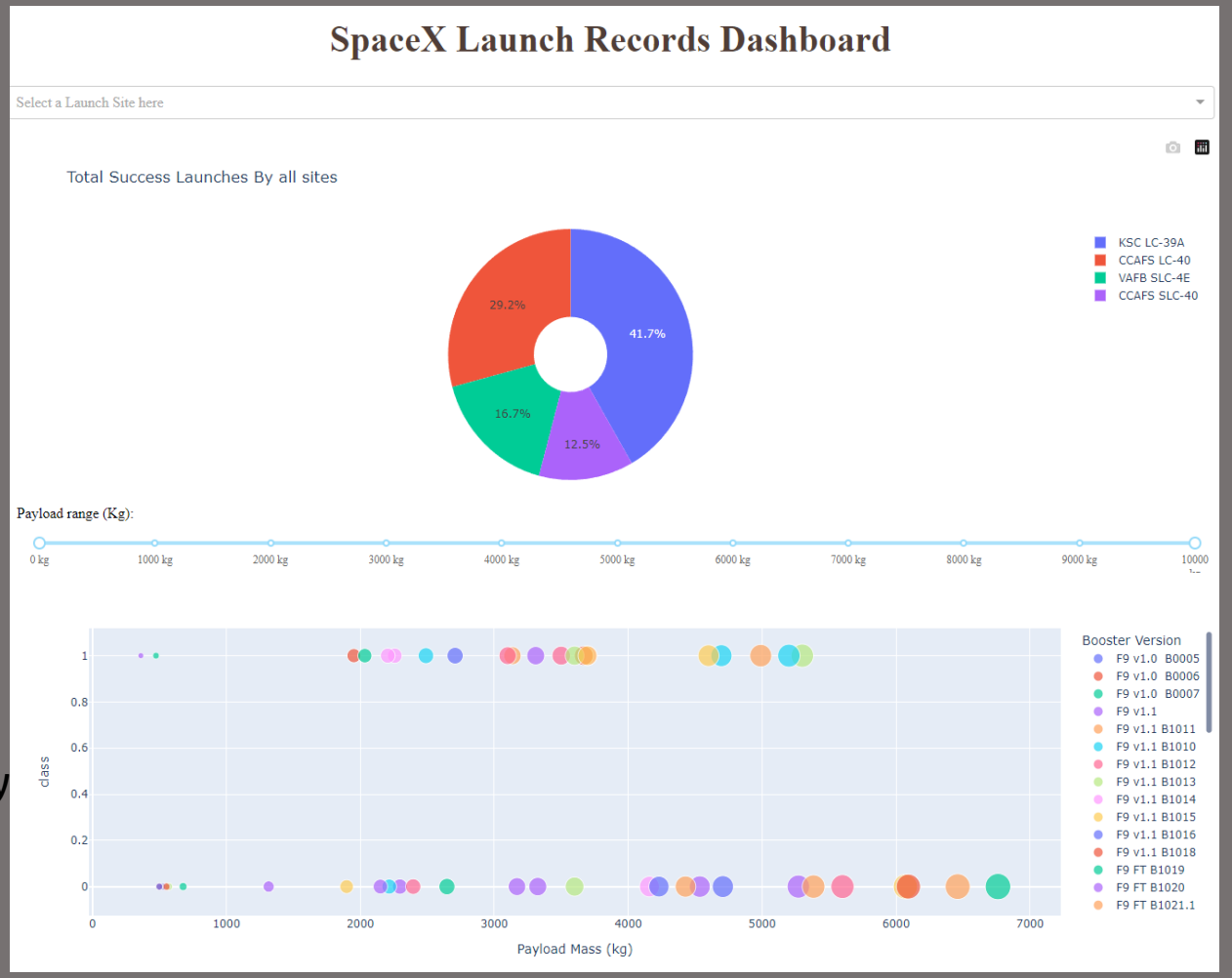
dropdown menu so that we can select different launch sites.

PIE CHART

It shows the success rate of each launch site . It is the best way to visualize success rate of launch site's.

SCATTER PLOT

It is a success payload scatter chart . It basically show how payload may be correlated with mission outcomes for selected site's.



- By analyzing dashboard we can find useful insights like:

1.Which site has the largest successful launches?
2.Which site has the highest launch success rate?
3.Which payload range(s) has the highest launch success rate?
4.Which payload range(s) has the lowest launch success rate?
5.Which F9 Booster version (`v1.0`, `v1.1`, `FT`, `B4`, `B5`, etc.) has the highest launch success rate?

https://u1508ritusin-8050.theiadocker-3-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/1b5a43336d52223c9c9239e62866ed421ffaa411/spacex_dash_app%20(1).py

Predictive Analysis (Classification)

- We create a machine learning pipeline to predict if the first stage will land or not.

Load the dataframe and create a column class in data

↓

Standardize the data in X and reassign X:
X= preprocessing.StandardScaler().fit(X).transform(X)

↓

split the data into training and testing data using the function train_test_split :
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

↓

Create Classification algorithms like logistic regression, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (KNN) and fit the object to find best parameter.

↓

Find the method perform best using .score() function

- In our test dataset we found that decision tree method having the best accuracy with 88 % .

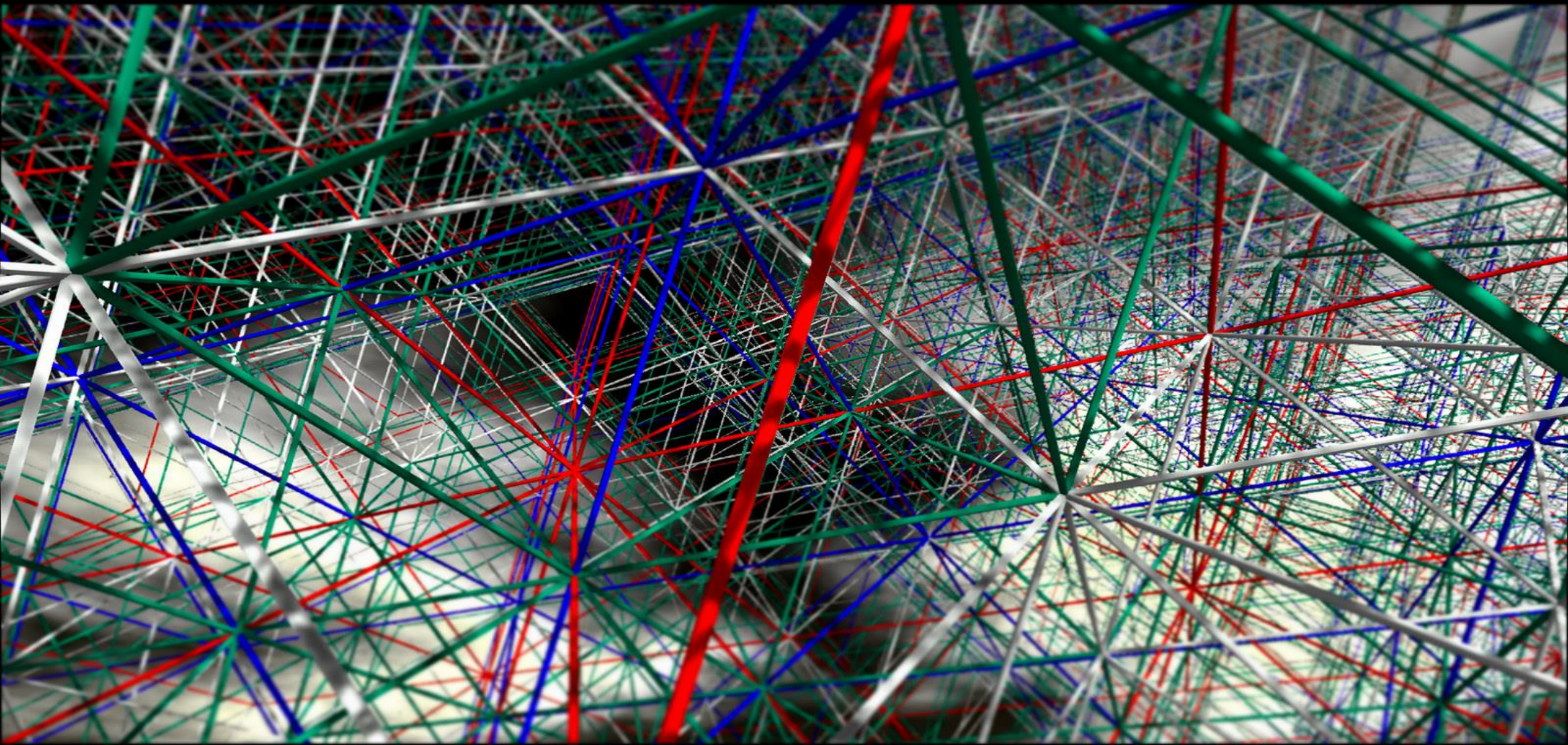
https://github.com/Ritusingh1500/Applied-Data-Science-Capstone-/blob/17789397b5aeb54158f53c5493322ad580b43bd2/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

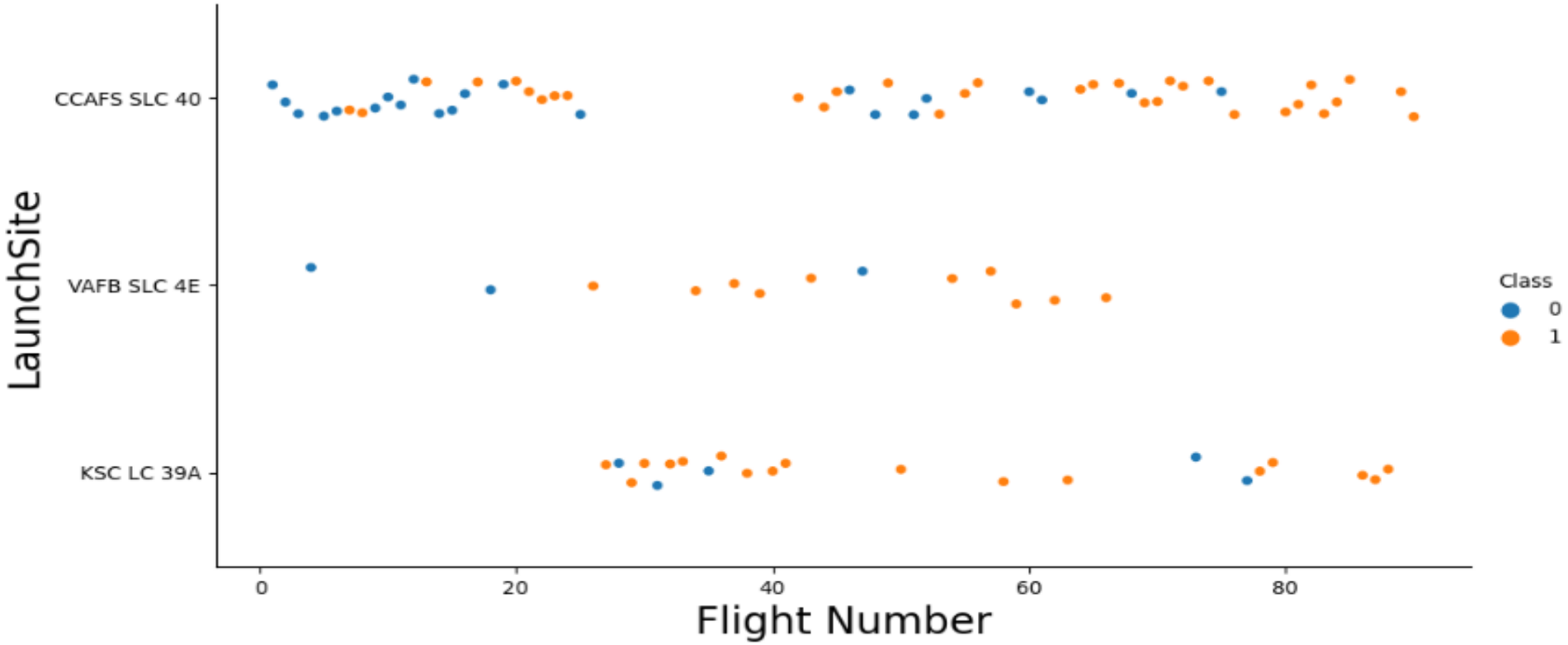
- Exploratory data analysis (EDA)
- Interactive analytics demo

  in screenshots
- Predictive analysis

EDA Data Visualization



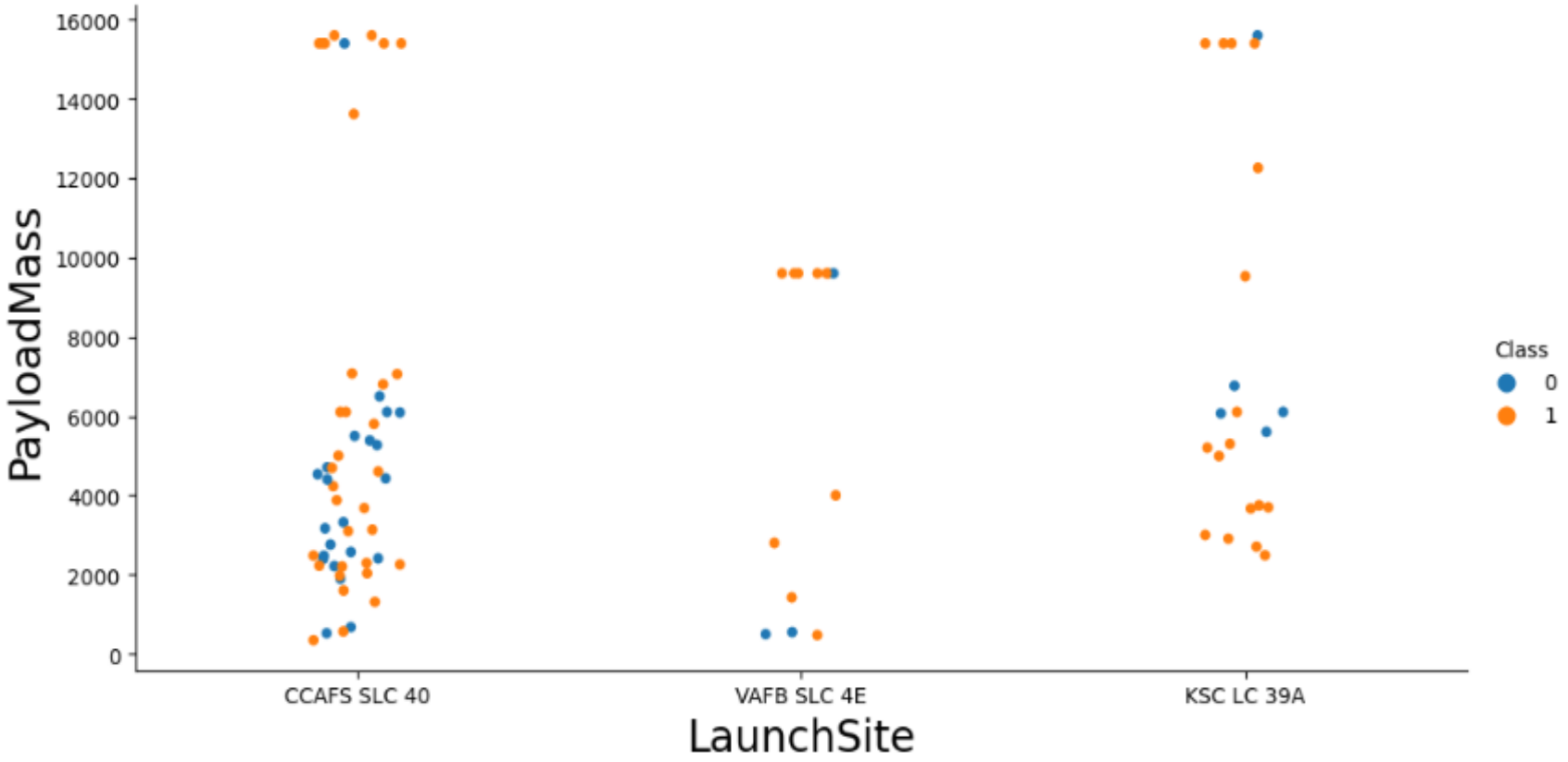
Flight Number vs. Launch Site



Conclusion :

- **Launch Site with Flight Number greater then 20 show better success rate of successful rocket landing.**
- **Launch site VAFB SLC 4E and KSC LC 39A have better success rate with around 77 % . As comparing to CCFS SCL 40 launch site.**
- **Flight number above 20 show better success rate.**

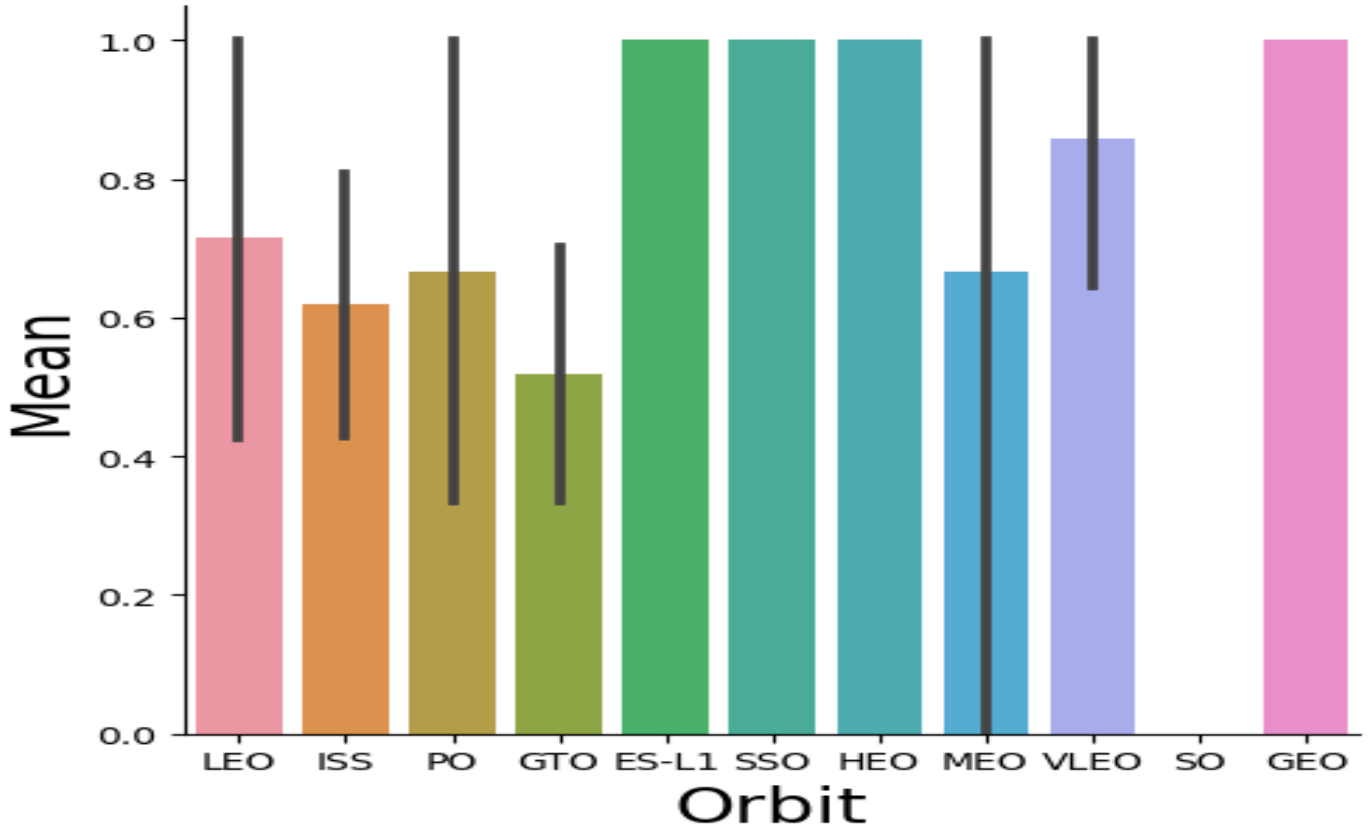
Payload vs. Launch Site



Conclusion :

- **We can say that for VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).**
- **Launch site's have better success rate for payload mass greater then 8000.**

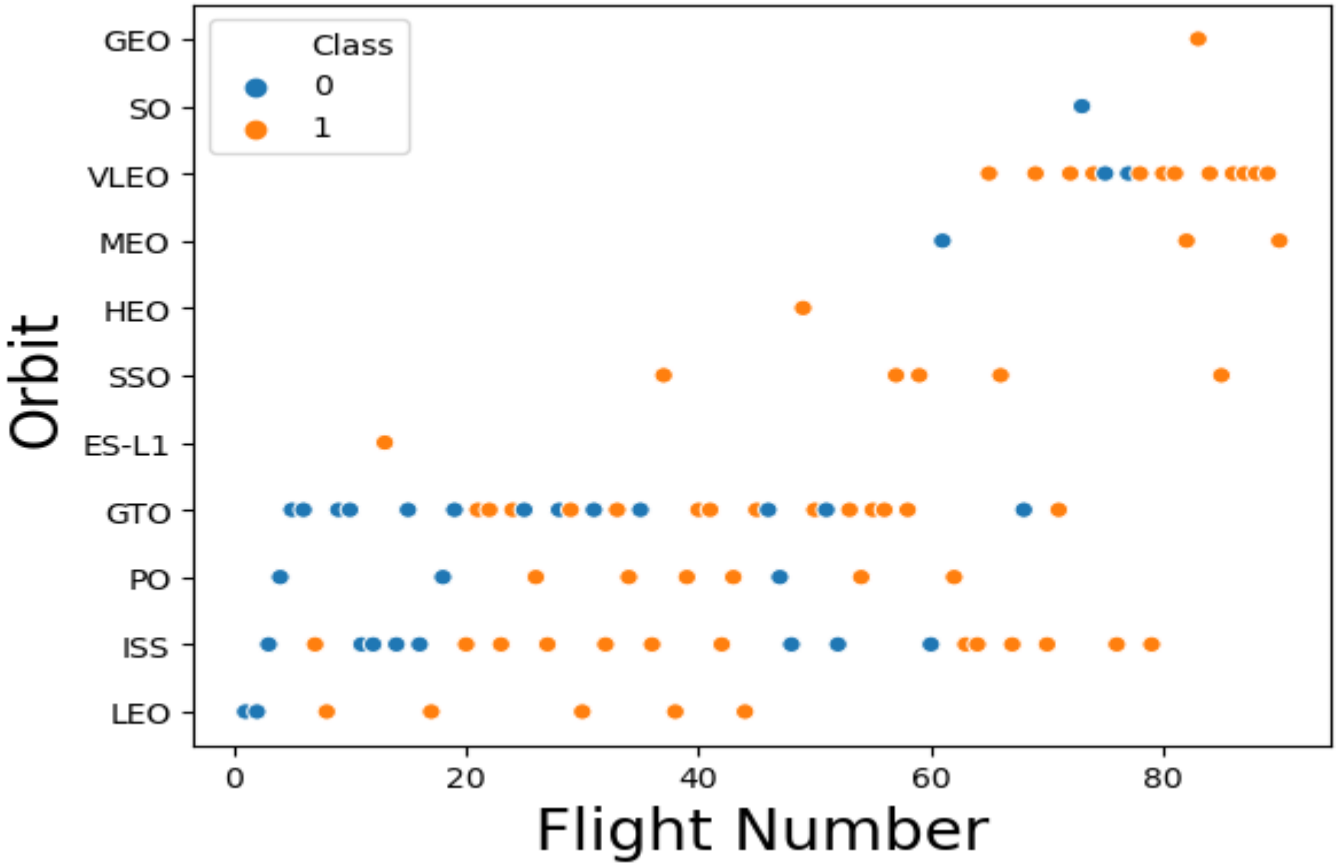
Success Rate vs. Orbit Type



Conclusion :

- **Orbit ES-L1, SSO, HEO, GEO have best success rate while GTO have worst success rate .**

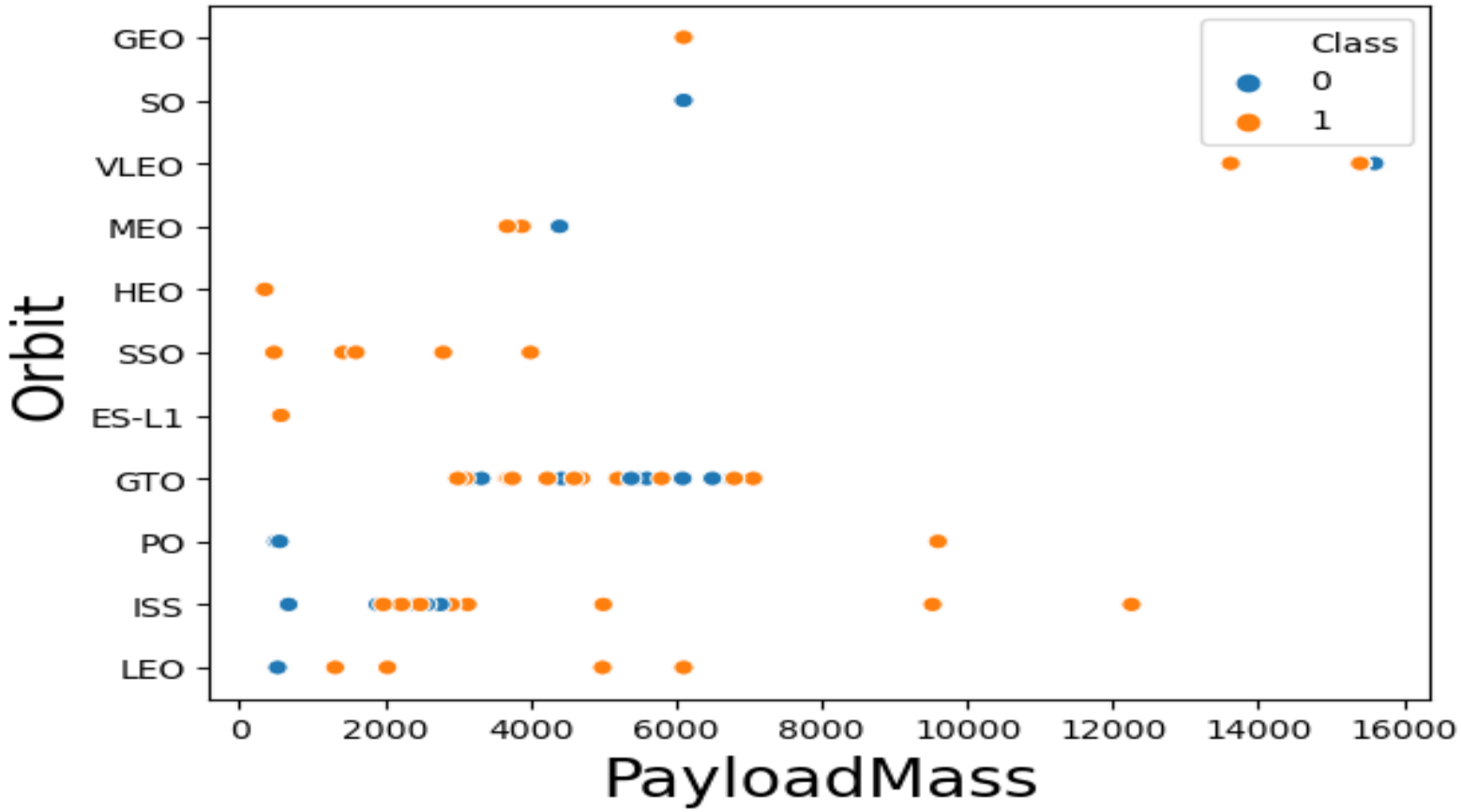
Flight Number vs. Orbit Type



Conclusion :

- **We can't see a clear relationship between Orbit and flight number. But we can say that LEO and VLEO orbit have high success rate and GTO orbit have lowest success rate.**

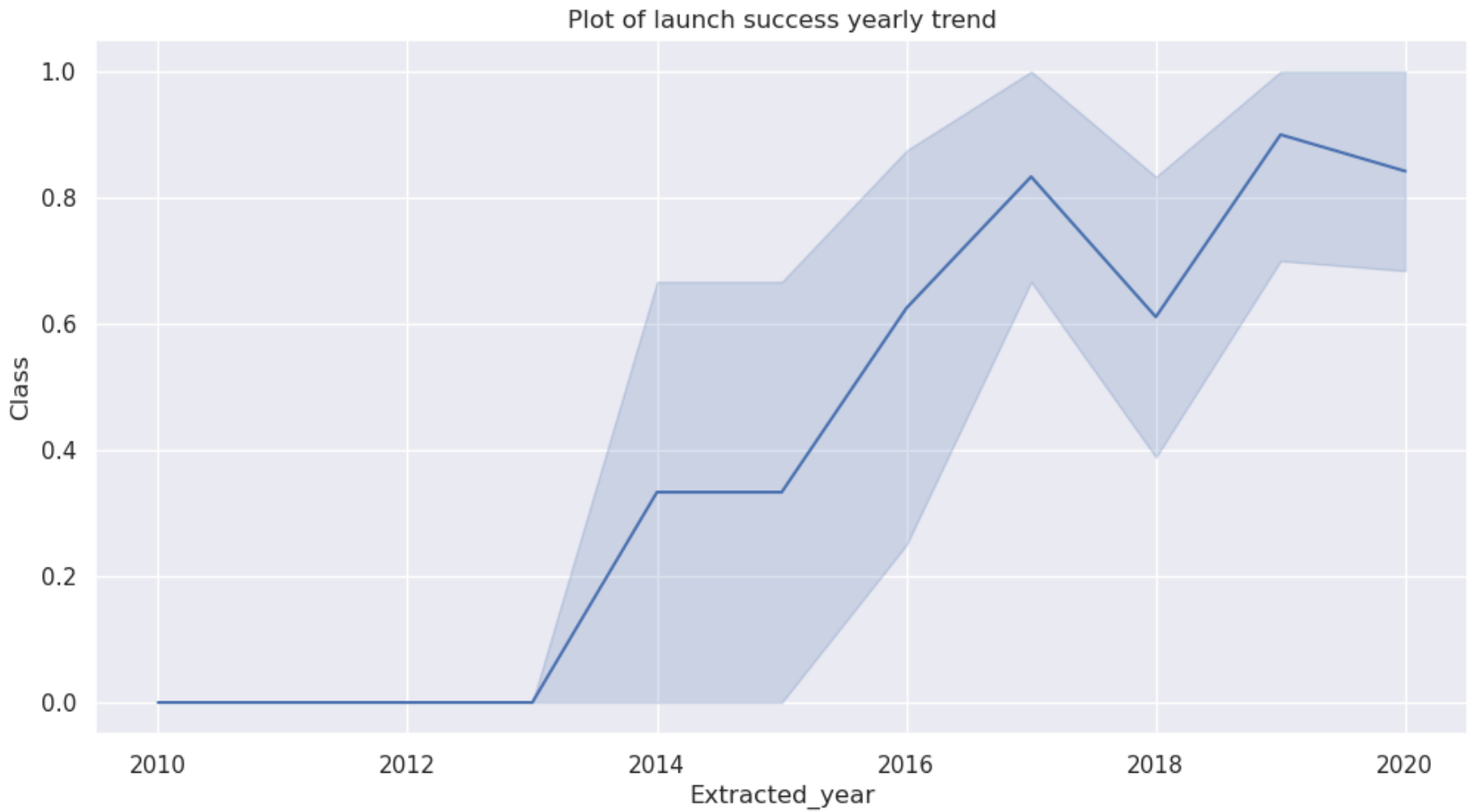
Payload vs. Orbit Type



Conclusion :

- **We can say that 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



Conclusion :

- **The success rate since 2013 kept increasing till 2020**

Exploratory data analysis with SQL

All Launch Site Names

SQL Query

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

%sql select DISTINCT(LAUNCH_SITE) from SPACEXTBL;

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

Query Explanation :

Select function is used to find certain data like here launch site and **DISTINCT** function is used to find unique name of launch sites from **SPACEXTBL** dataset.

Launch Site Names Begin with 'CCA'

SQL Query

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

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

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

Query Explanation :

SELECT (*) function is use to show all columns from **SPACEXTBL** dataset. And **where** function is used to filter data with condition **LIKE 'CCA%' LIMIT 5** to show launch site whose launch site name start with CCA string with record limit 5.

Total Payload Mass

SQL Query

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

```
%sql select customer, sum(PAYLOAD_MASS__KG_) as total_payload_mass
from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

Customer	total_payload_mass
NASA (CRS)	45

Query Explanation :

In this code **customer** and **sum(PAYLOAD_MASS_KG)** is selected with in **sum** function is used find total payload mass whose booster launched is by **NASA (CRS)** from SPACEXTBL dataset

Average Payload Mass by F9 v1.1

SQL Query

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

%sql select BOOSTER_VERSION, avg(PAYLOAD_MASS__KG_) as average_payload_mass from SPACEXTBL where BOOSTER_VERSION ='F9 v1.1';

Booster_Version	average_payload_mass
F9 v1.1	29

Query Explanation :

In this code **select BOOSTER_VERSION,avg(PAYLOAD_MASS_KG)** in which **avg** function is used to find average of payload mass and **where** function is to filter with the condition having **booster version F9 v1.1**

First Successful Ground Landing Date

SQL Query

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

%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEXTBL WHERE "LANDING__OUTCOME" = 'Success (ground pad)';

First Succesful Landing Outcome in Ground Pad
2015-12-22

Query Explanation :

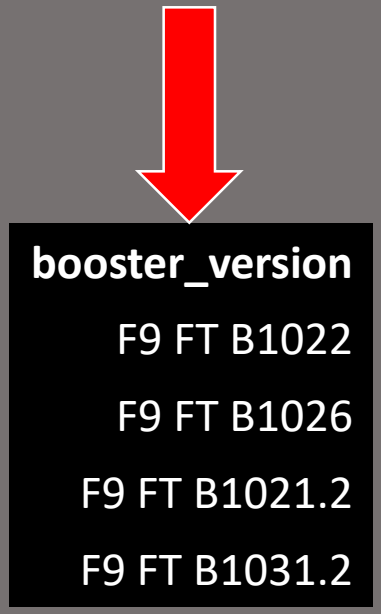
In this code we **select min(DATE)** where **min** function is used to show minimum date and **where** function is to filter with the condition **LANDING_OUTCOME=success(ground pad)** which gives successful landing outcome in ground pad in **SPACEXTBL** dataset

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

- 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 MISSION_OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```



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

Query Explanation :

In this code **select BOOSTER_VERSION** is use to find booster version ,**where** with a condition of having **MISSION_OUTCOME = 'Success (drone ship)'** for successful drone ship mission outcome and **PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000** for having payload mass between 4000 to 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query

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

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTBL;
```

Successful Mission	Failure Mission
100	1

Query Explanation :

In this code **select sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission"** is used to show sum all mission outcome with success result. And also **select sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission"** is used to show sum of all the mission outcome with failure result from **SPACEXTBL** dataset.

Boosters Carried Maximum Payload

SQL Query

- List the names of the booster_versions which have carried the maximum payload mass.

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

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

Query Explanation :

Select BOOSTER_VERSION which we are finding .**where PAYLOAD_MASS_KG = (select max(PAYLOAD_MASS_KG)** is use to filter data with condition of having maximum payload mass from **SPACEXTBL** dataset.

2015 Launch Records

SQL Query

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

%sql SELECT "monthname(DATE)" as Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE "year(DATE)" = '2015' AND "LANDING__OUTCOME" = 'Failure (drone ship)';

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

Query Explanation :

In this code **Select monthname(DATE) , BOOSTER_VERSION, LAUNCH_SITE** which we want to find from **SPACEXTBL** dataset . **monthname(DATE)** function dispay month name from the date.And **where** is used to impose condition like **"year(DATE)" = '2015' AND "LANDING__OUTCOME" = 'Failure (drone ship)';** which means year should be 2015 and landing outcome should be failure in drone ship .

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

SQL Query

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

%sql SELECT "LANDING__OUTCOME" AS "Landing outcome" COUNT("LANDING__OUTCOME") AS "Rank of successful landing between 2010-06-04 and 2017-03-20" FROM SPACEXTBL WHERE"LANDING__OUTCOME" LIKE '% Success %' AND DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "LANDING__OUTCOME" ORDER BY COUNT("LANDING__OUTCOME") DESC;

Landing outcome	Rank of successful landing between 2010-06-04 and 2017-03-20
Success (drone ship)	5
Success (ground pad)	3

Query Explanation :

In this code **SELECT "LANDING__OUTCOME" ,COUNT("LANDING_OUTCOME)** ,count function is used to count no of landing outcome. **WHERE** function is use to impose **"LANDING__OUTCOME" LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20'** condition which means count landing outcomes whose result contain success word and having date between 04-06-2010 and 20-03-2017. **GROUP BY "LANDING__OUTCOME"** function is used to group landing outcome and **ORDER BY COUNT("LANDING__OUTCOME") DESC** function is used to count landing outcome in descending order.

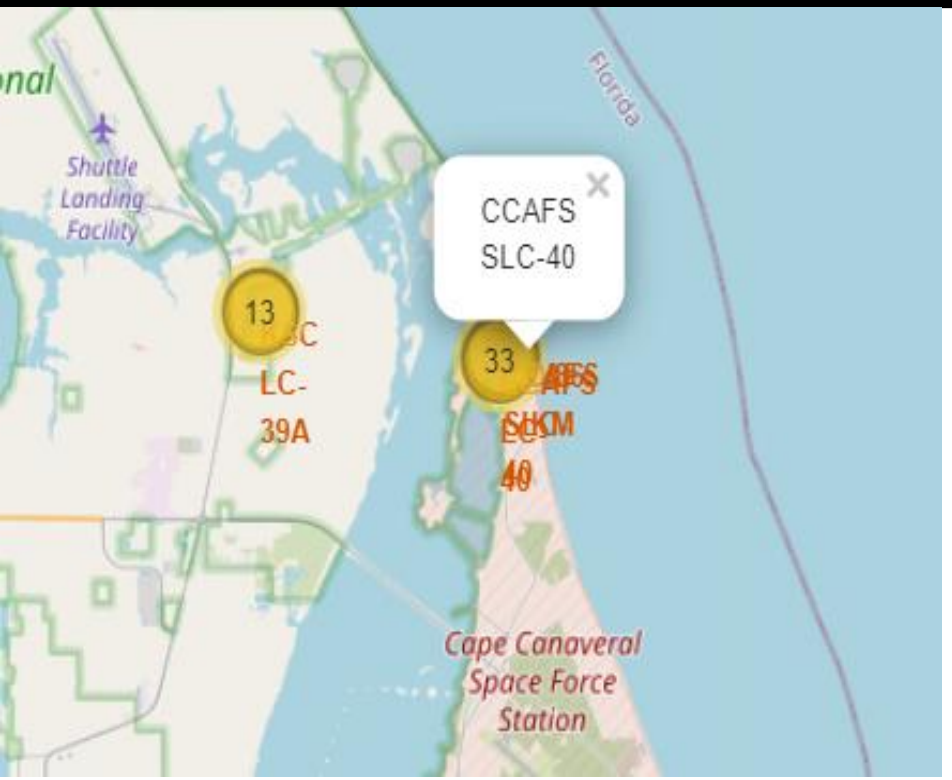

Launch sites
Proximities Analysis

ALL LAUNCH SITE

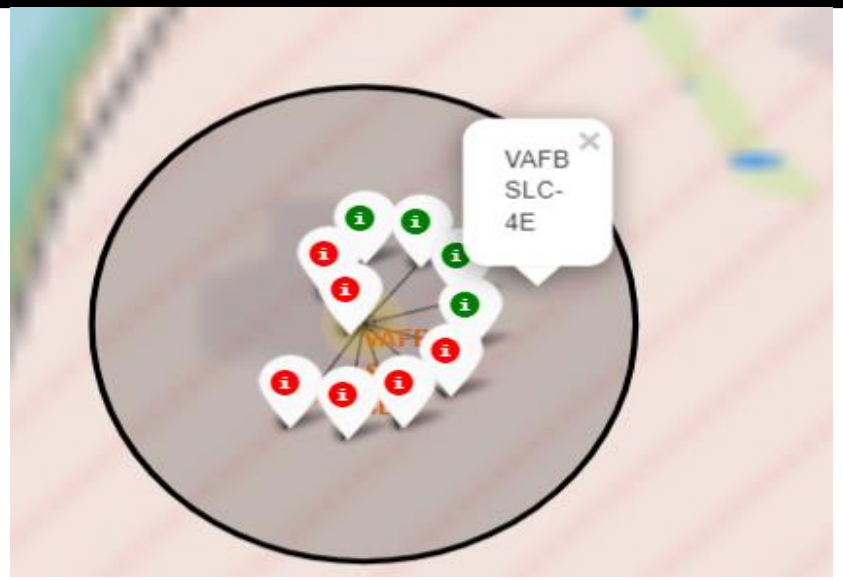
World map showing all Launch site of SpaceX . We can see all launch site are around coastal area of USA.

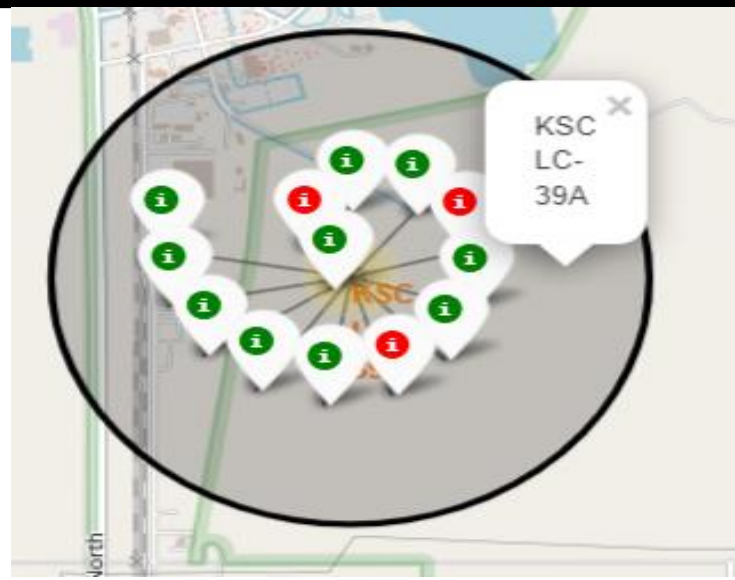
Zoomed images

MARKED LAUNCH SITES



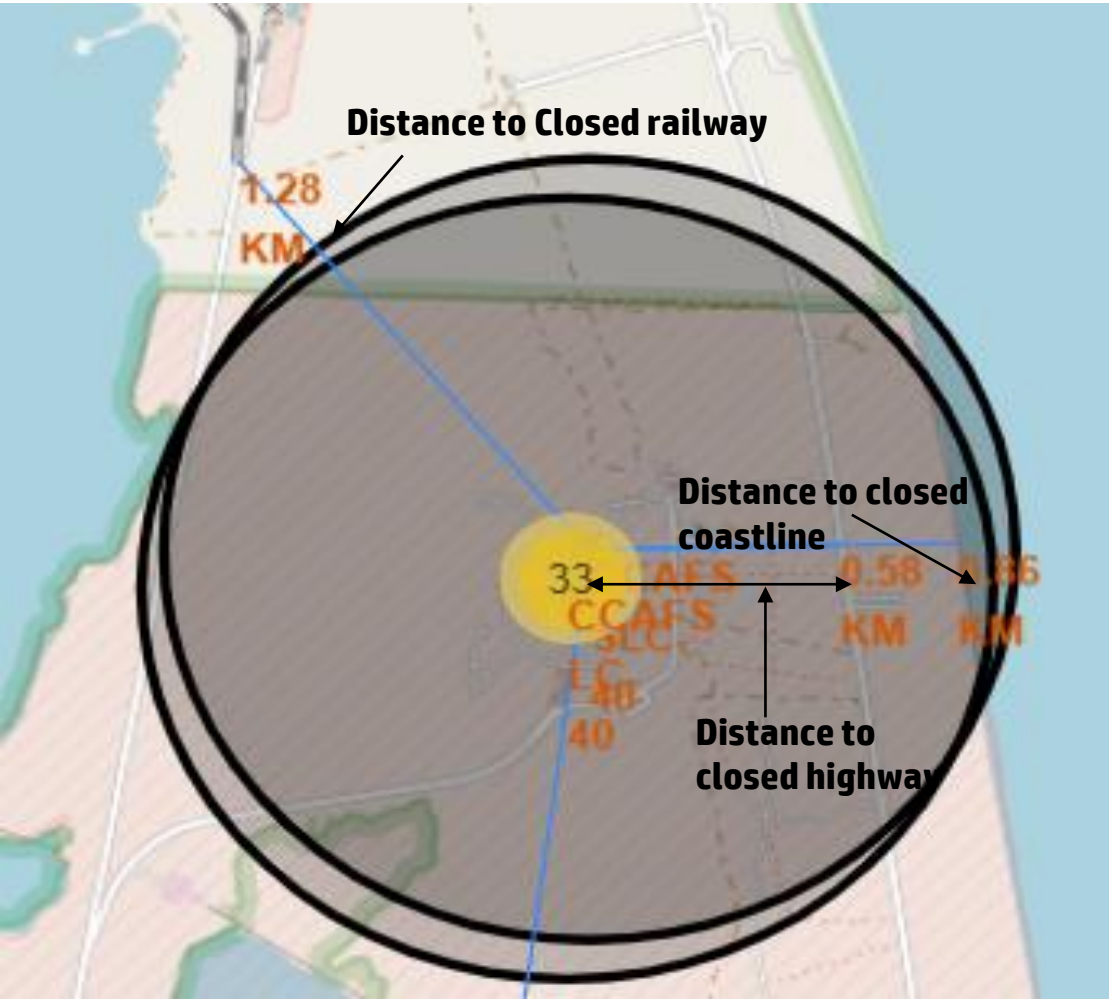
Yellow circle is a marked cluster which indicate sites are very close to each other. So all close site indicate by single one circle and number inside circle represent number of sites in that place.

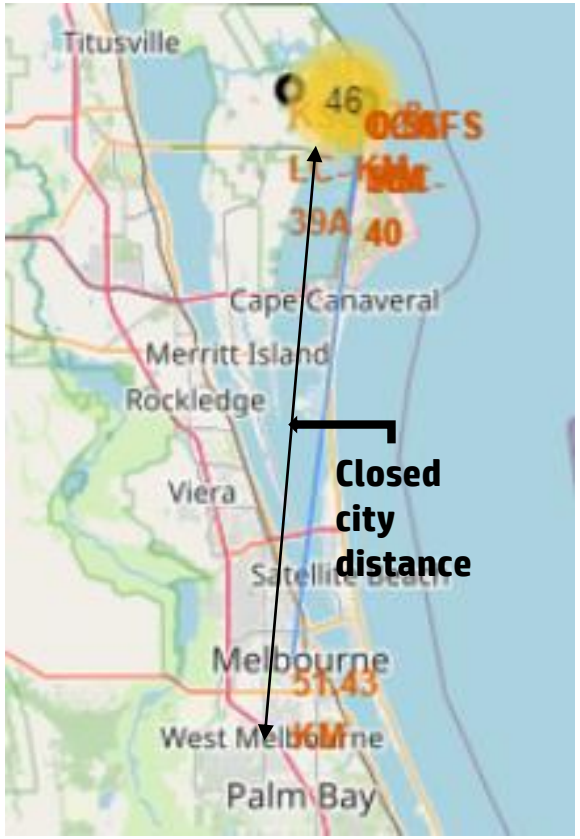


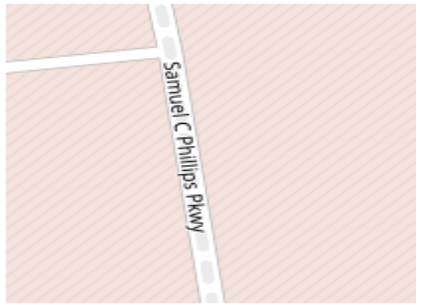


Green marker indicate successful launch and **red marker** show unsuccessful launch . By that we can find which launch site have greater success rate

INSIGHTS OF LAUNCH SITES

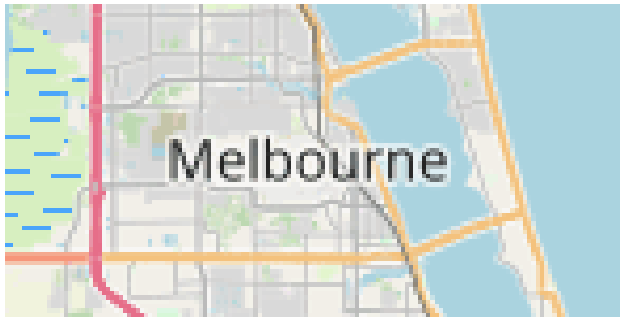






Closed highway symbol

Closed Railway symbol



Closed City symbol

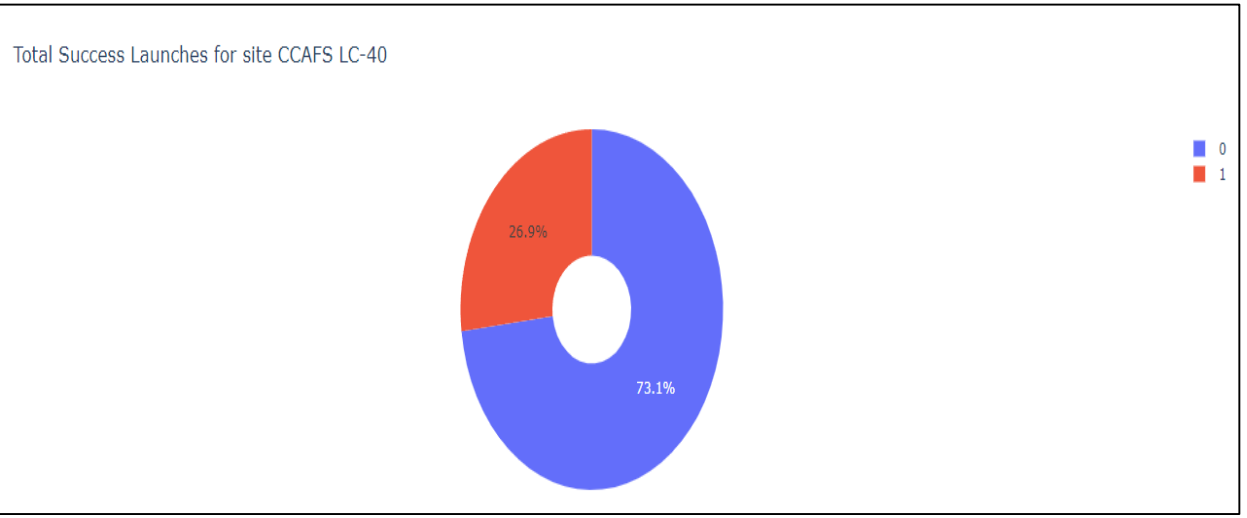
```
distance to closest highway = 0.5834695366934144  km
distance to closest railroad = 1.2845344718142522  km
distance to closest city = 51.43416999517233  km
```

By analyzing maps we can answer following things :

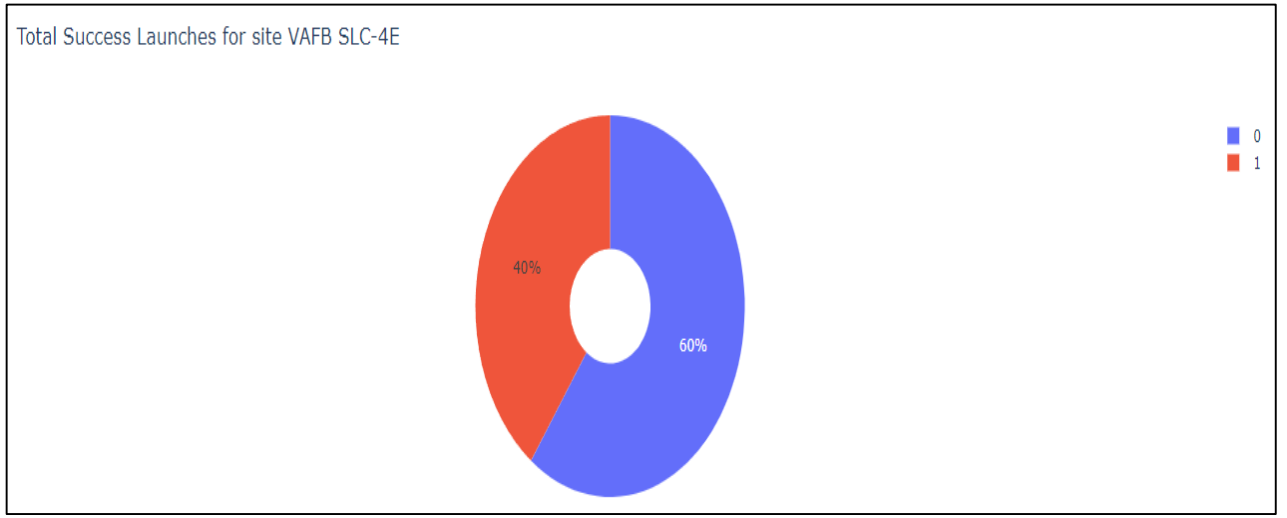
- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?

Interactive analytics with Dashboard

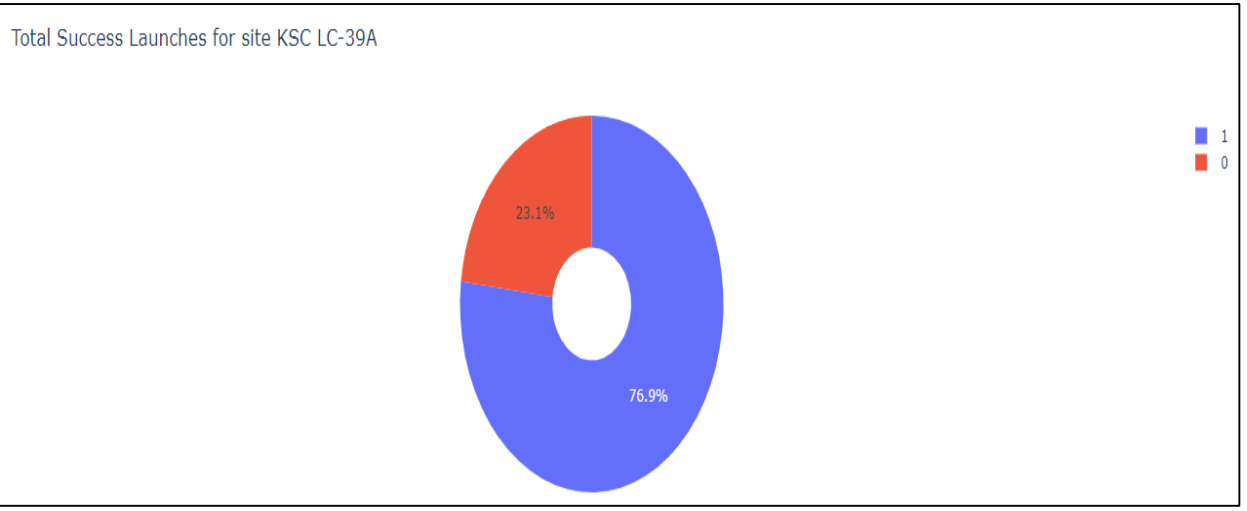
PEICHART LAUNCH SUCCESS COUNT



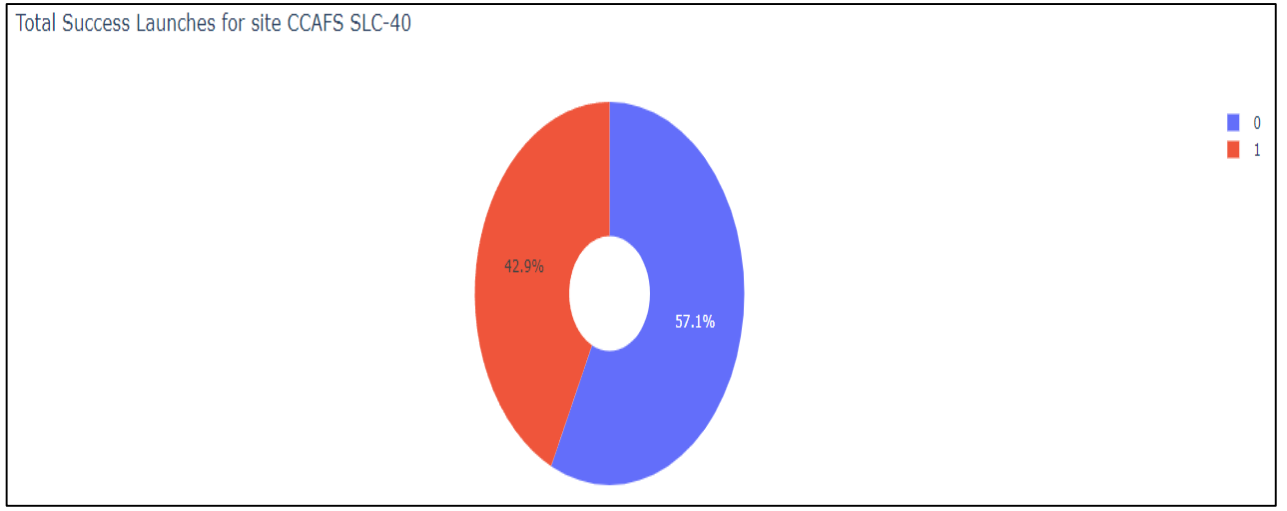
CCAFS LC - 40 has 73.1% success rate



VAFB SLC - 4E has 60% success rate

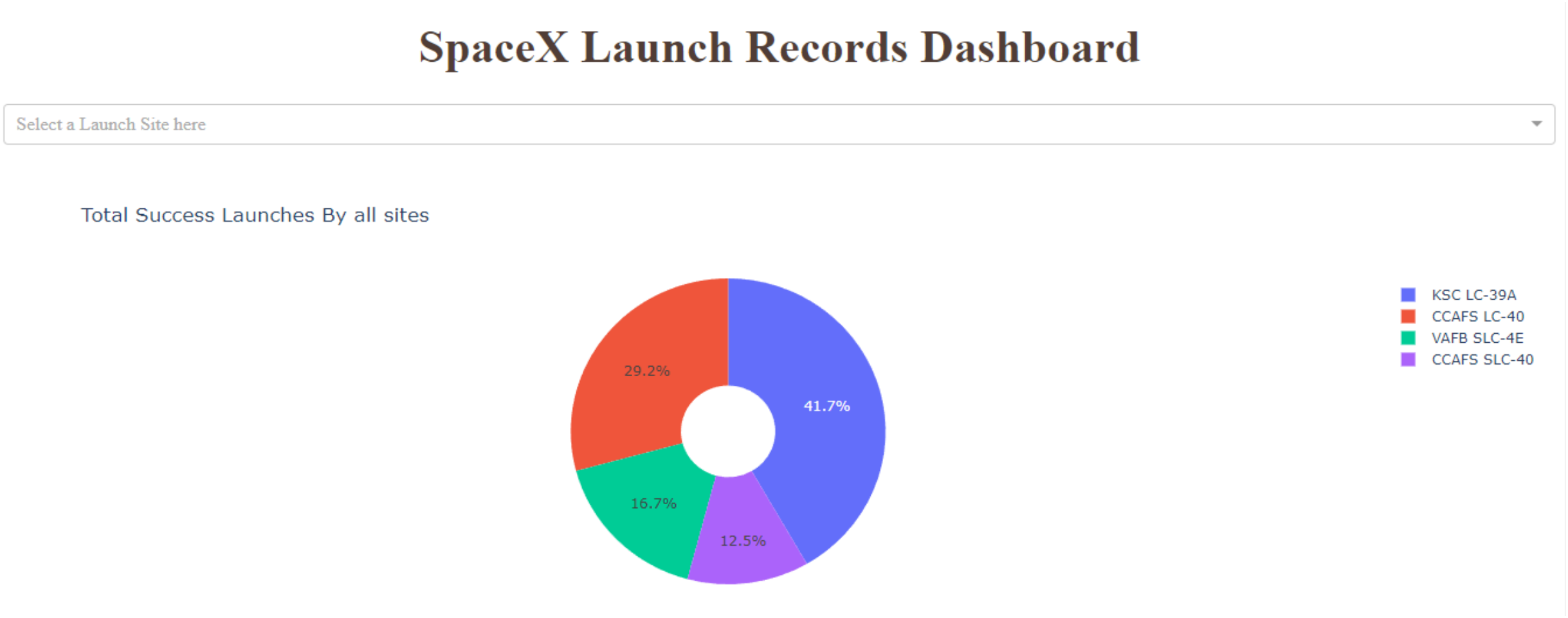


KSC LC – 39A has 76.9% success rate



CCAFS SLC – 40 has 57.1% success rate

PEICHART FOR ALL LAUNCH SITES

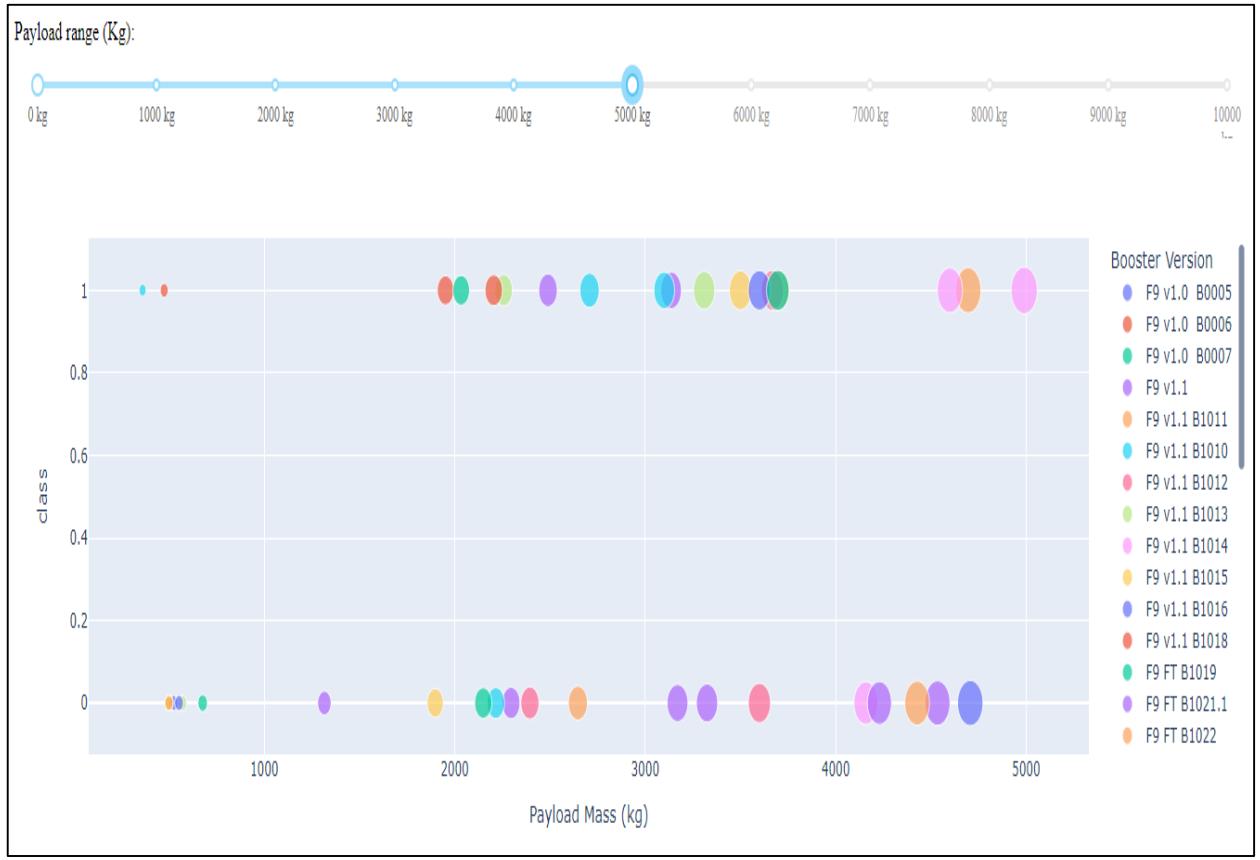


Conclusion

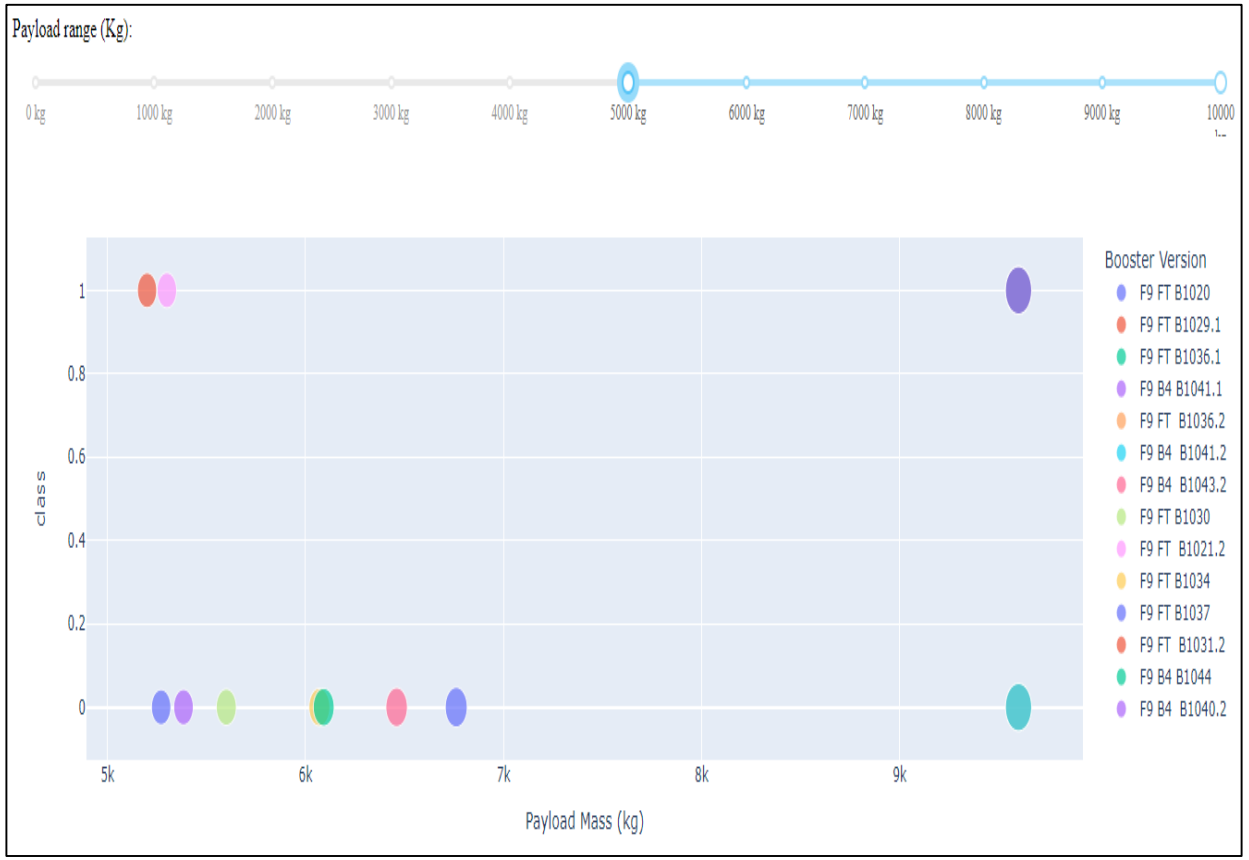
- Pie chart shows that KSC LC- 39A has the highest success rate of 41.2% from all the launch sites.

Payload vs. Launch Outcome scatter plot

For payload range 0kg – 5000kg

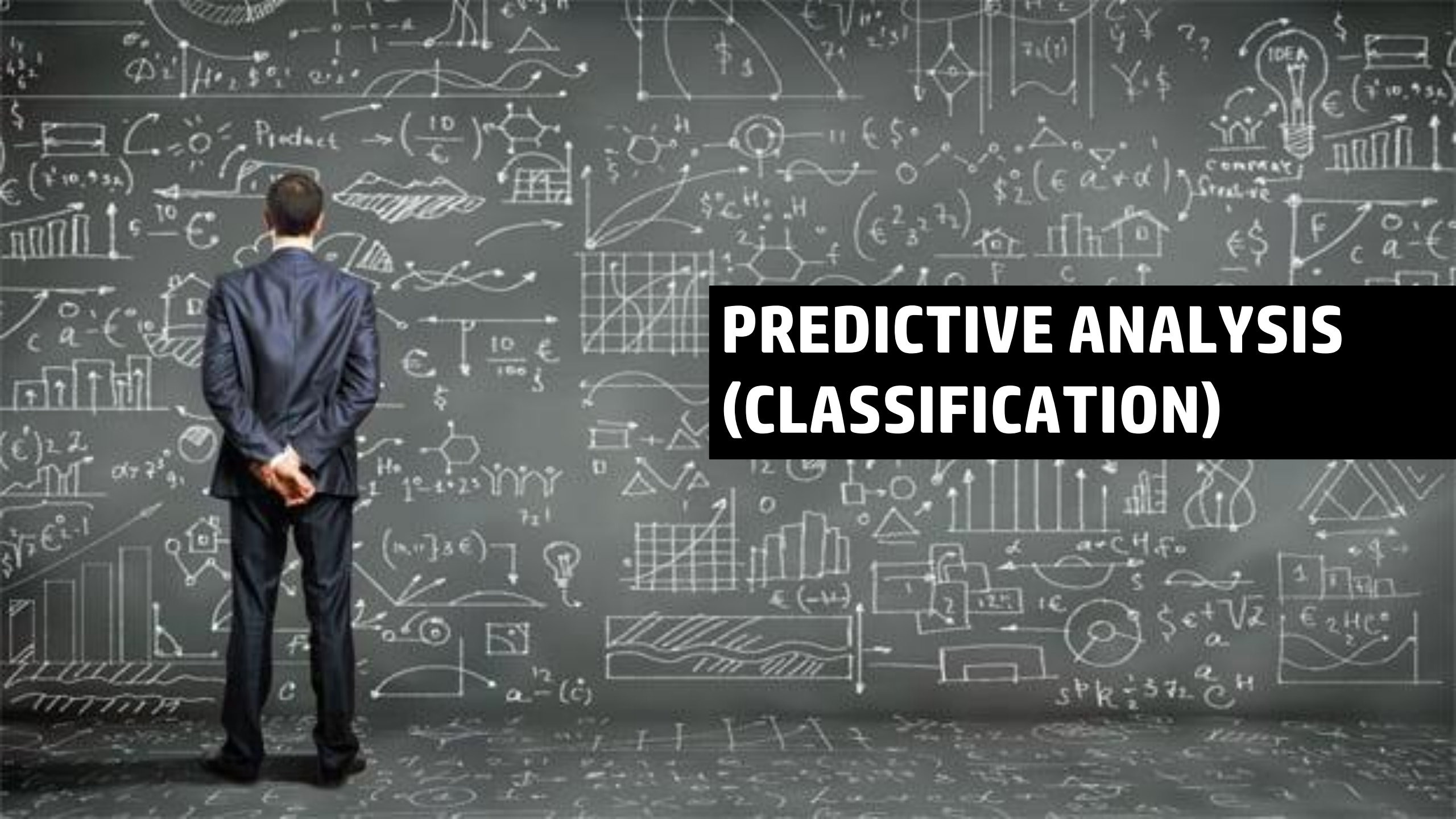


For payload range 5000kg -10000kg



Conclusion

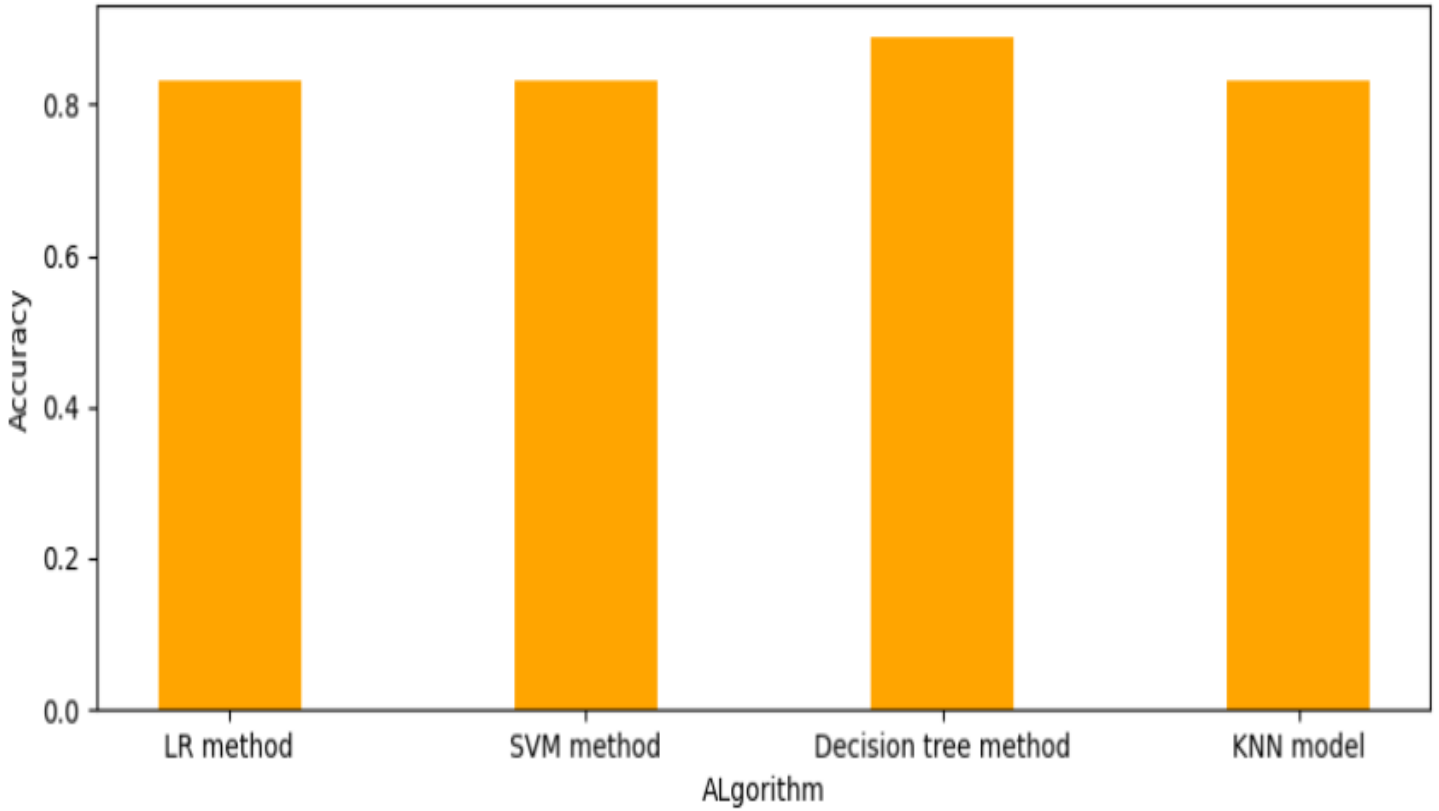
We can see success rate is higher for lower payload mass range as comparing to higher payload mass range


PREDICTIVE ANALYSIS
(CLASSIFICATION)

Classification Accuracy

Methods accuracy values :

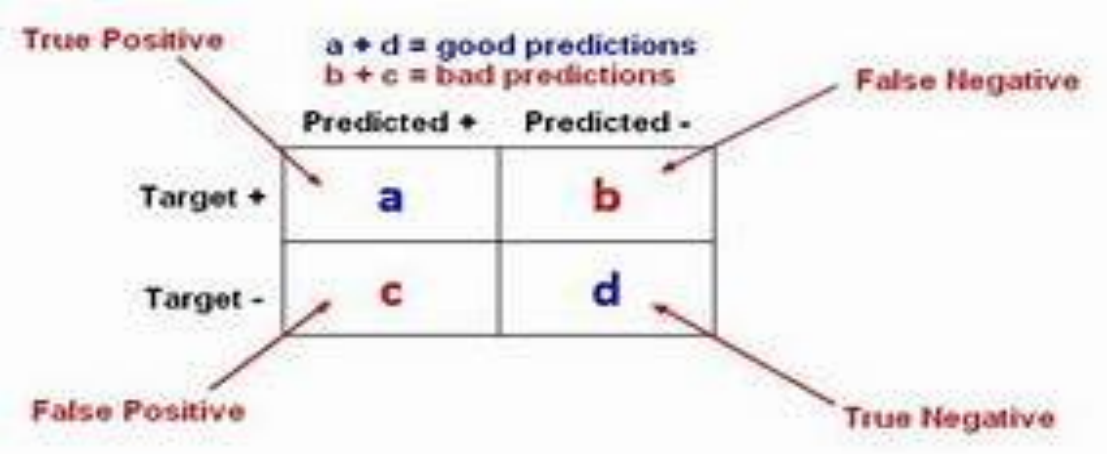
```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8888888888888888
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

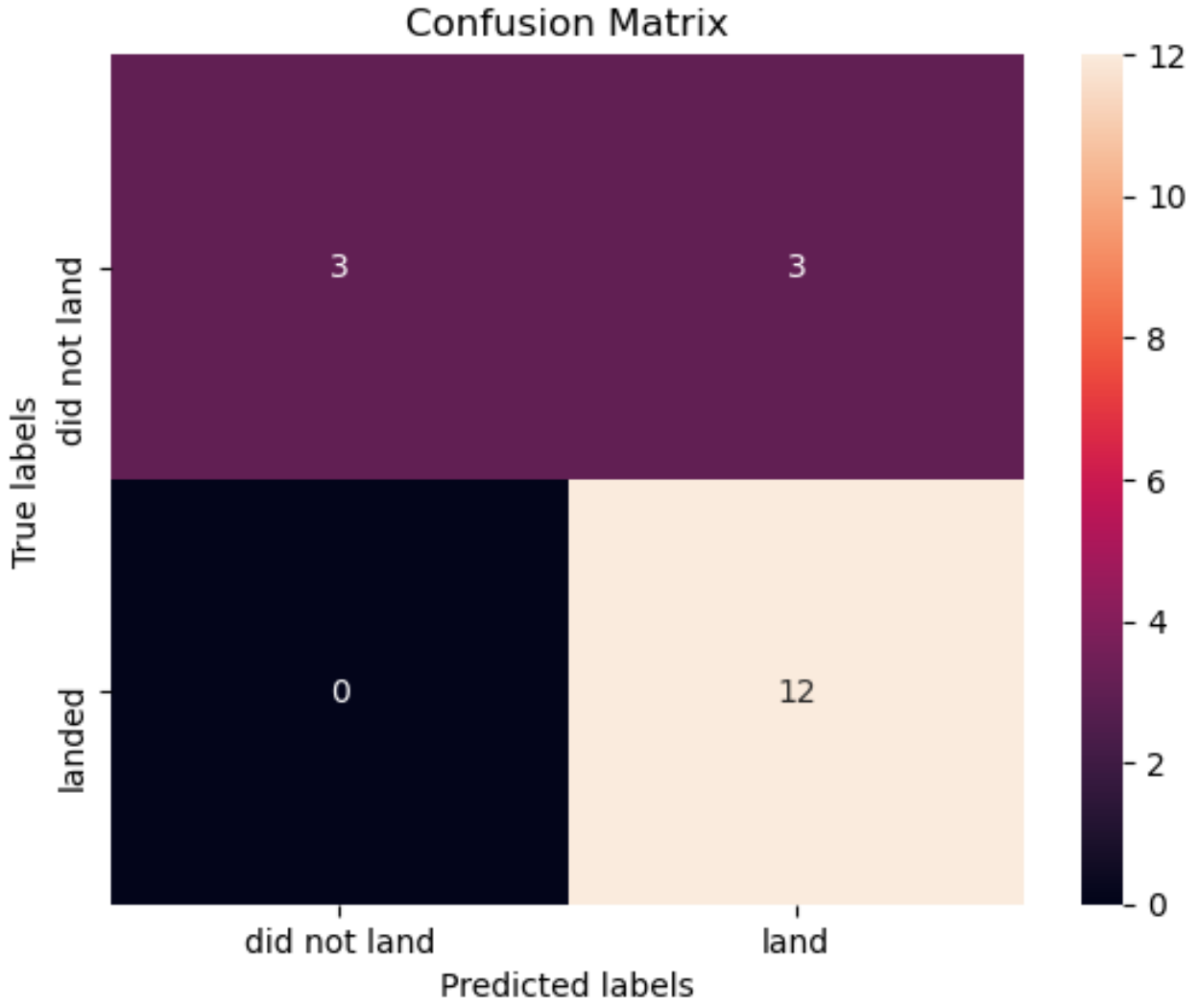


We can say that **decision tree method** is the best classification method with 0.888 accuracy . It means decision tree method is best to use in analyics.

Confusion Matrix

A confusion matrix is **a table that is used to define the performance of a classification algorithm**.
This is the decision tree confusion matrix. We can distinguish between the different classes. We see that the major problem is false positive.





Conclusions

- Launch Site with Flight Number greater then 20 show better success rate of successful rocket landing.
- Launch success rate is higher for lower payload mass range as comparing to higher payload mass range.
- Orbit ES-L1, SSO, HEO, GEO have best success rate while GTO have worst success rate .
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- KSC LC – 39A has highest successful launch sites.
- Decision tree method is best classification method to use.

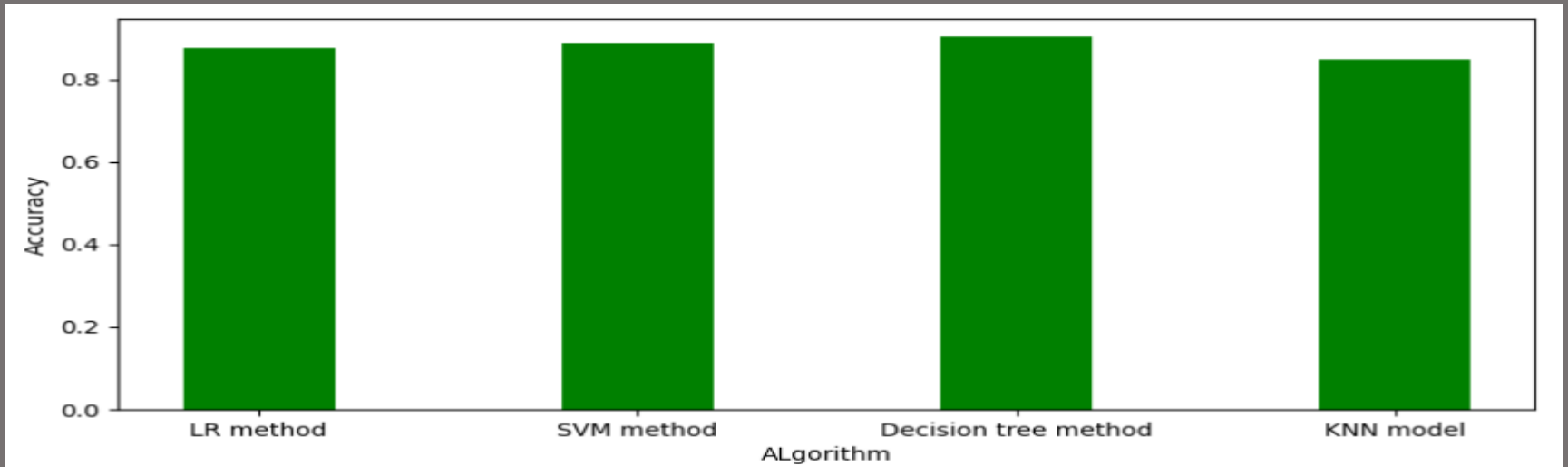
Appendix

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Total Number of Successful and Failure Mission" FROM SPACEXTBL \
WHERE MISSION_OUTCOME LIKE 'Success%' OR MISSION_OUTCOME LIKE 'Failure%';
```

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

Total Number of Successful and Failure Mission
101

Use to show combine mission outcomes



Observe bar chart of training Data set

Did some Google research on data collection, data wrangling, confusion matrix etc.


THANK YOU !