IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bhaskar Saha
15-Dec-2021

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data collection methodology
  - Perform data wrangling
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly Dash
  - Perform predictive analysis using classification models

- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context
  - About sending spacecraft, SPACEX is the most successful, among others. That is because SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX uses Falcon 9 rocket launches on its website with a cost of 62 million dollars where other providers cost upwards 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. Also, we need to determine if SpaceX will reuse the first stage.

- Problems you want to find answers
  - What influences if the first stage will land successfully?
  - The effect each relationship with certain variables will impact in determining the success rate of a successful landing.
  - What are the conditions SpaceX need to have to achieve to get the best results and ensure the best first stage success landing rate.

Section 1

Methodology

Methodology

- Executive Summary
  - Data collection methodology
    - SpaceX Rest API
    - Web Scrapping from Wikipedia
    - Clean the requested data
  - Perform data wrangling
    - Gather preliminary insight of data
  - 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

- The following data were collected for this project

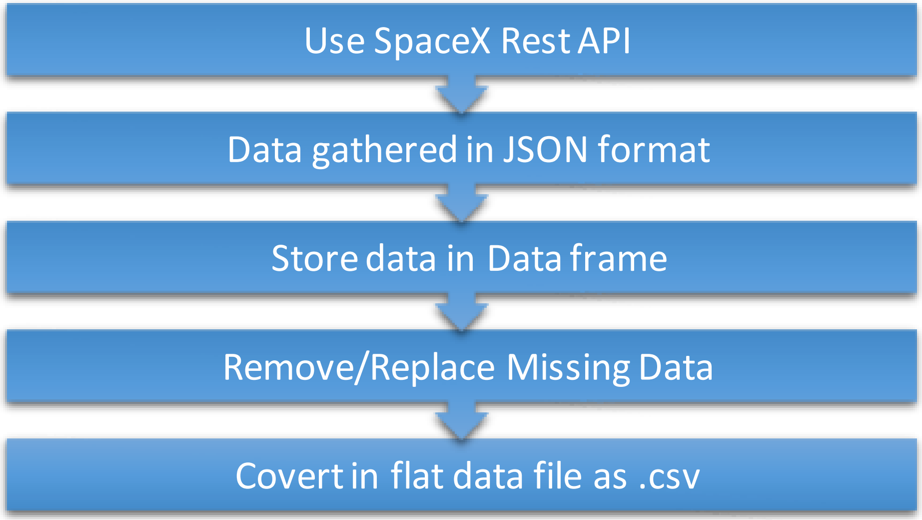
❑ SpaceX REST API

✓ We worked with SpaceX launch data that is gathered from the SpaceX REST API.

✓ This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

✓ Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.

✓ The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/



❑ Web scraping Wikipedia

✓ Another popular data source for obtaining Falcon 9 Launch data is Web scraping Wikipedia using Beautiful Soup.

Data Collection – SpaceX API

Git hub URL : [Data fetch from rest API]

1. Request and parse the SpaceX launch data using the GET request

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

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

2. Converting the data into .json format

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

3. Apply custom functions to fetch required data and in right format

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

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

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

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

4. Construct a dictionary with required data

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

5. Fetch only data related to Falcon 9 launches

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

6. Replace Missing Data and save into csv file

```
# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,mean,inplace=True)
data_falcon9.isnull().sum()
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

Git hub URL : [Web Scaping link]

1. Getting response from HTML

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

2. Creating BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(spacex_data.text, 'html.parser')
```

3. Finding tables and getting Column Names

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

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

4. Creation of Dictionary and append data to key

```
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
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict["Flight No."].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])
```

5. Converting dictionary to Data frame and Data frame to .csv

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

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

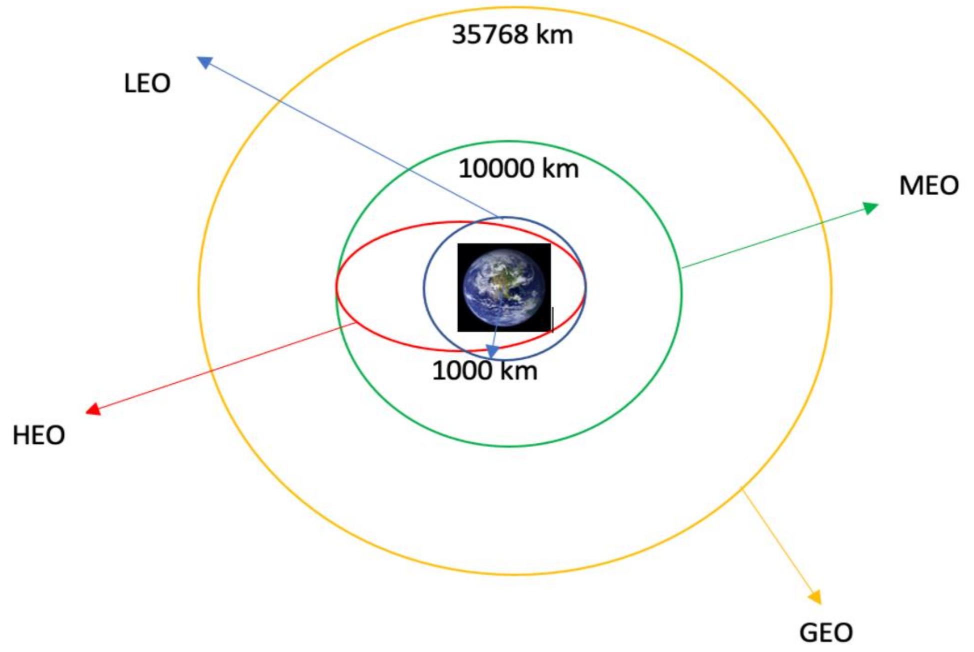
Data Wrangling

Introduction:

In the data collected from Sources, 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 the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was 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 on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

In Data Wrangling, those outcomes are converted into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

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



Git hub URL : [Data Wrangling Link]

- Perform exploratory Data Analysis on Dataset
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Create a landing outcome label from Outcome column
- Export into .csv

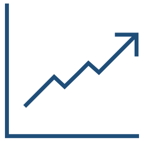
EDA with Data Visualization

- **Scatter Graphs being drawn for**:
  - ***Flight Number*** *vs* ***Payload Mass***
  - ***Flight Number*** *vs* ***Launch Site***
  - ***Payload Mass*** *vs* ***Launch Site***
  - ***Flight Number*** *vs* ***Orbit Type***
  - ***Payload*** *vs* ***Orbit Type***

Scatter are used ***to observe and show relationships between two numeric variables****. The dots in a scatter plot not only report the values of individual data points, but also patterns when the data are taken as a whole. This graph shows how much one variable is affected by another.*

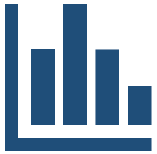
Git hub URL : [EDA with Data Visualization link]

- **Line Graph being drawn for**
  - ***Launch success*** *vs* ***Year***



Line graphs are used to track changes over short and long periods of time. It helps to make predictions about the results which are not recorded yet or predictions for future year.

- Bar graph being drawn for
  - ***Success Rate*** *vs* ***Orbit Type***



A bar graph makes it easy to compare sets of data between different groups immediately. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar graphs can also show big changes in data over time

EDA with SQL

Some SQL queries are performed to get insight of the dataset.

Some information are needed which are based on some conditions which can be easily fetched using some SQL queries.

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

Git hub URL : [EDA with SQL]

Build an 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 hopefully we could discover some of the factors by analyzing the existing launch site locations.

Folium is used

- To visualize the Launch Data into an interactive map. 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 and count of launches from that site.
- We assigned launch outcomes(failures, successes) to classes 0 and 1 with **Green** and **Red** markers on the map in a Marker Cluster()
- Using **Haversine's formula** we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks
- Example of some trends in which the Launch Site is situated in.
    - •Are launch sites near railways? ***Yes***
    - •Are launch sites near highways? ***Yes***
    - •Are launch sites near coastline? ***Yes***
    - •Do launch sites keep certain distance away from cities? ***Yes***

Git hub URL : Interactive Map with Folium

Build a Dashboard with Plotly Dash

- Plotly Dash application is built for users to perform interactive visual analytics on SpaceX launch data in real time
- This dashboard application contains ***Launch site*** as a dropdown list and a ***Payload mass*** range slider to interact with a pie chart and a scatter point chart.
    - **Pie chart** visualizing launch success counts and percentage which will answer
        - Which site has the largest successful launches?
        - Which site has the highest launch success rate?
    - **Scatter chart** to visually observe how payload may be correlated with mission outcomes for selected site. Also, to color-label the Booster version on each scatter point to observe mission outcomes with different boosters. This chart will answer
        - Which payload range(s) has the highest launch success rate?
        - Which payload range(s) has the lowest launch success rate?
        - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success

Git hub URL : [Source Code of Plotly Dash]

Predictive Analysis (Classification)

General Steps for Predictive Analysis

Pre-processing
- to standardize our data

Train Test Split
- to split our data into training and testing data

Train model
- Train Model with Train set of data

Grid Search
- to find the hyperparameters that allow a given algorithm to perform best
- Using the best hyperparameter values, we will determine the model with the best accuracy using the training data
- Test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors

Test Model
- Test the model with Test Data to find accuracy

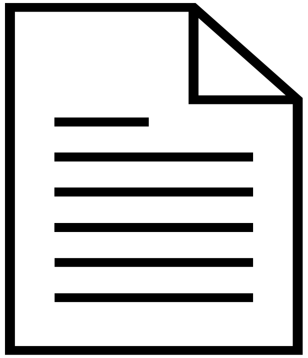
Confusion matrix creation
- to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known

Steps followed in Predictive Analysis and Training

Load data into NumPy and Pandas → Standardize the data → Split the Data into Test and train set(Test size = 0.2) → Set hyper parameter for different Test models → Train the model → Figure out the accuracy for the model using GridSearchCV object → Test the model with test data → Create Confusion matrix to figure out Model performance → Repeat the processes from setting hyper parameter to confusion matrix creation for other Models

Git hub URL : [Predictive Analysis]

Results

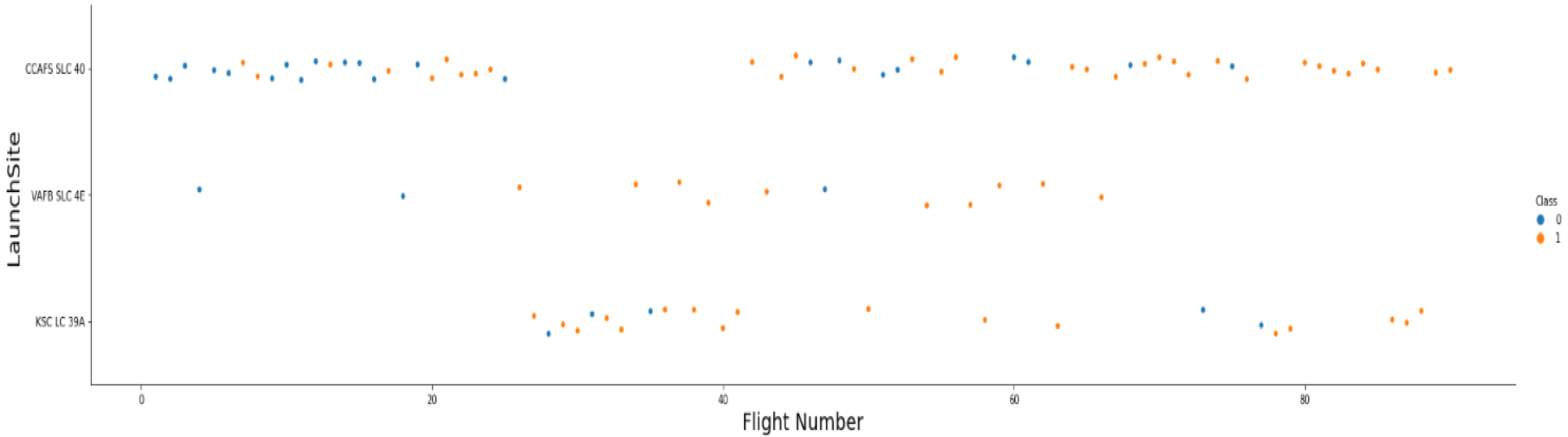


- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

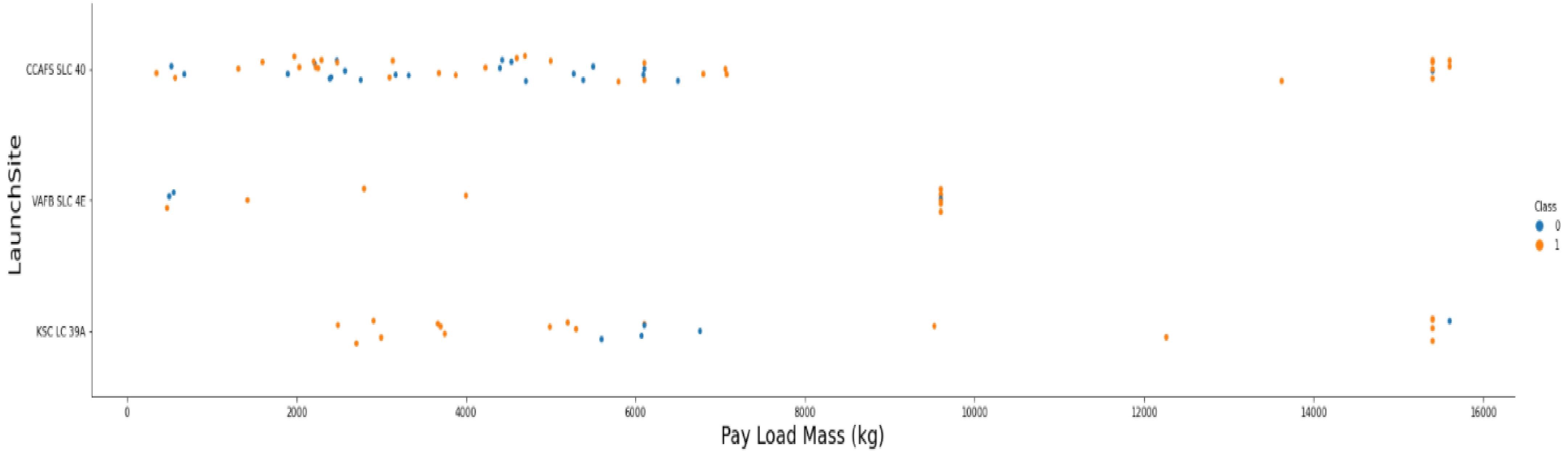
Insights drawn from EDA

Flight Number vs. Launch Site



- Launch Site CCAFS SLC 40 launched most flights. VAFB SLC 4E launched a smaller number of flights.
- CCAFS SLC 40 is the first who landed the first Stage successfully.
- All the flights after Flight number 77 are successfully landed First Stage irrespective of Launch Site.

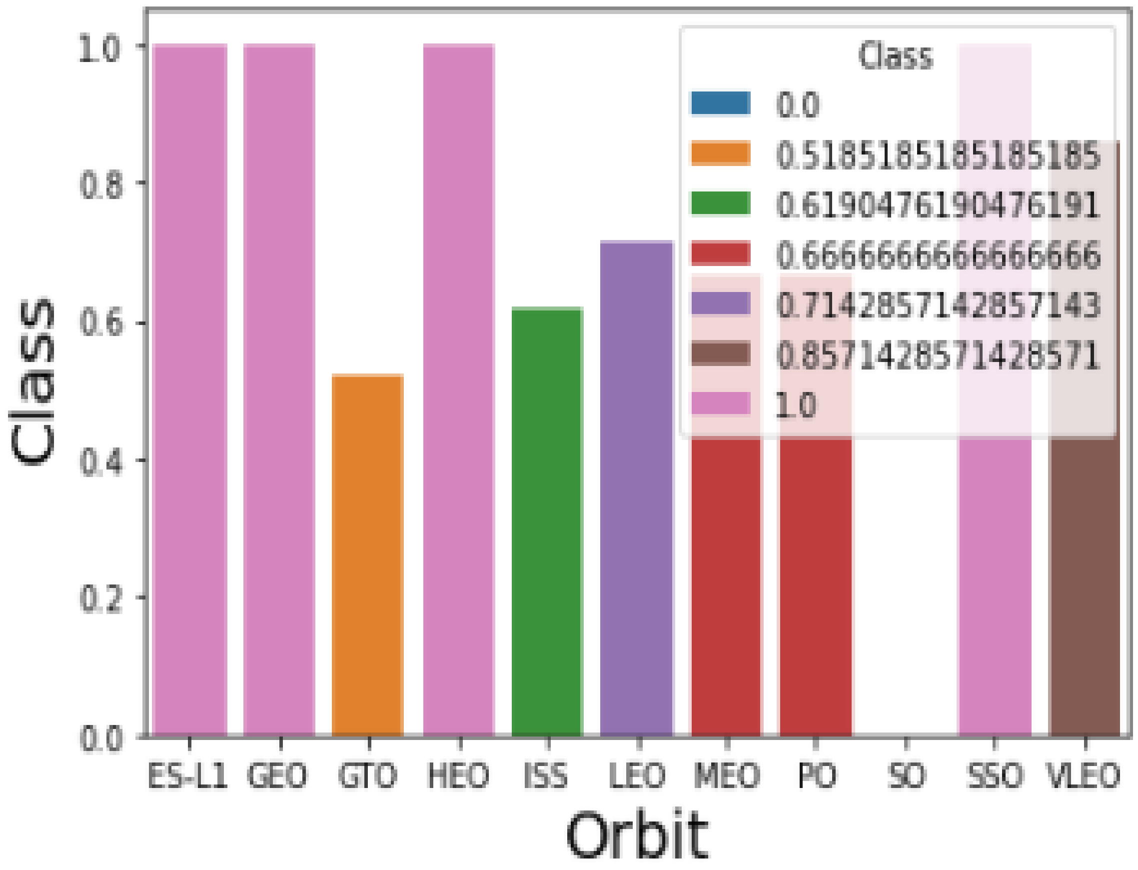
Payload vs. Launch Site



- Launches with highest Payload mass which is successfully launched is from CCAAFS SLC 40
- VAFB SLC 4B launches are all less than 10000 kg Pay Load.
- Higher Payload launches are mostly successful

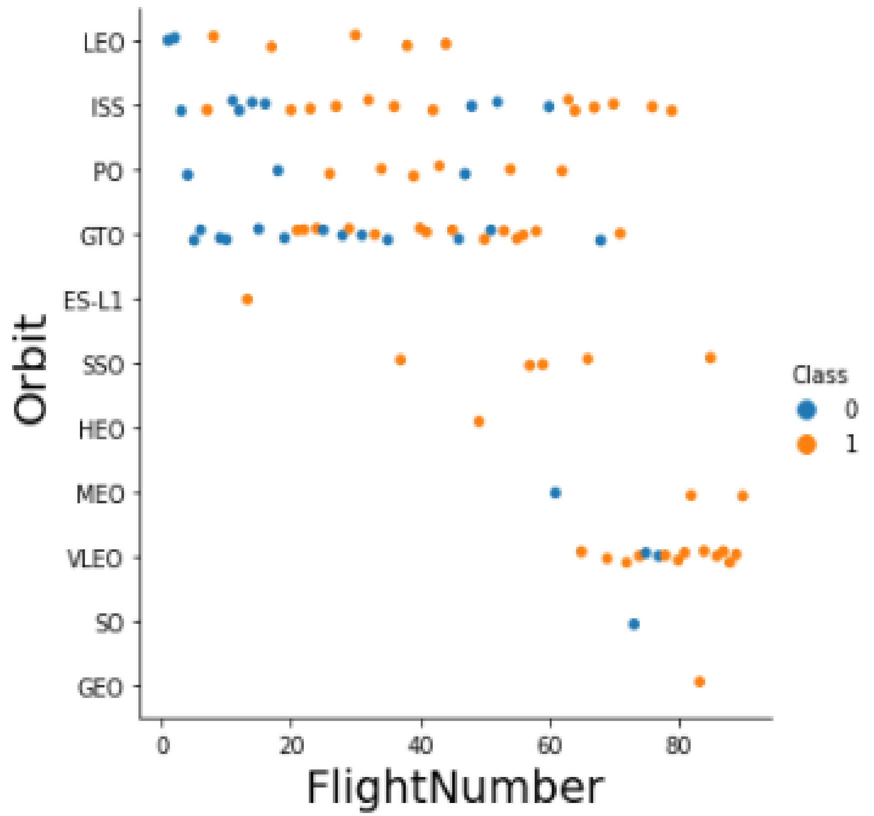
Success Rate vs. Orbit Type

- All launches to Orbits ES-L1,GEO,HEO and SSO are Successful
- VLEO has 85% Success Rate
- Launches to SO orbit is Unsuccessful



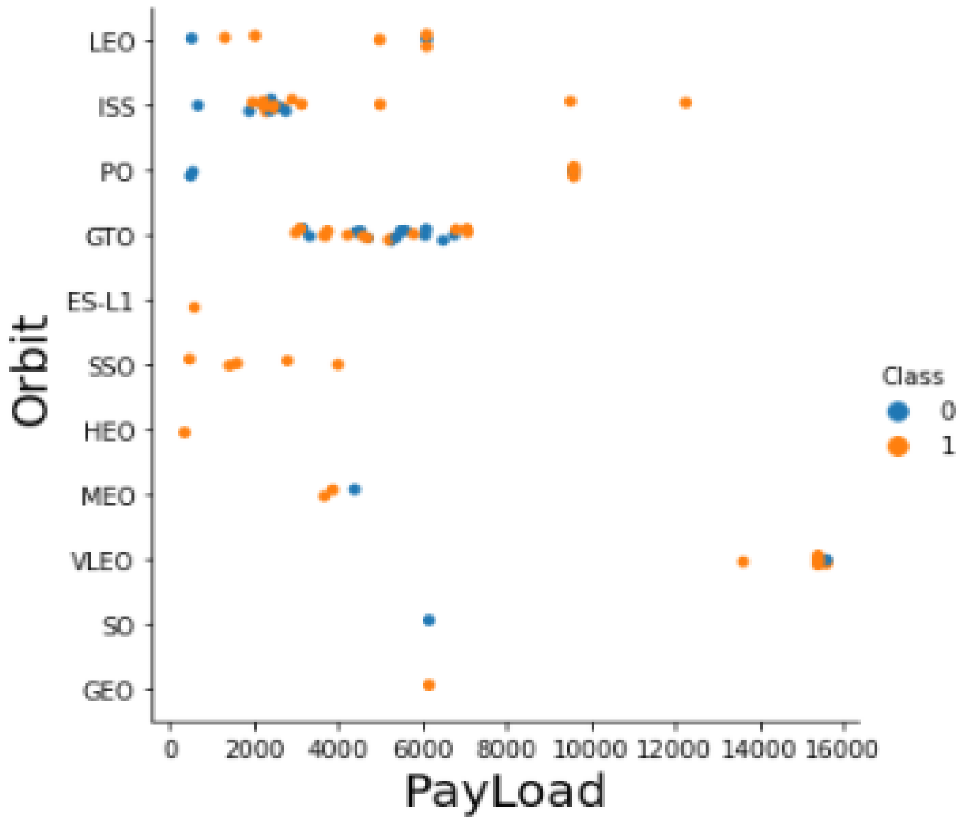
Flight Number vs. Orbit Type

- Most of the Launches are belongs to ISS and GTO orbit Types
- LEO has mostly the success in respective to number of flights
- Current launches are mostly VLEO orbit types and mostly successful
- There is no relation between success regarding Flight number in case of obit type GTO
- In First phase of launches mostly Flights are related to LEO,ISS,PO and GTO whereas in later phase of launches are mainly concentrated on VLEO orbit types

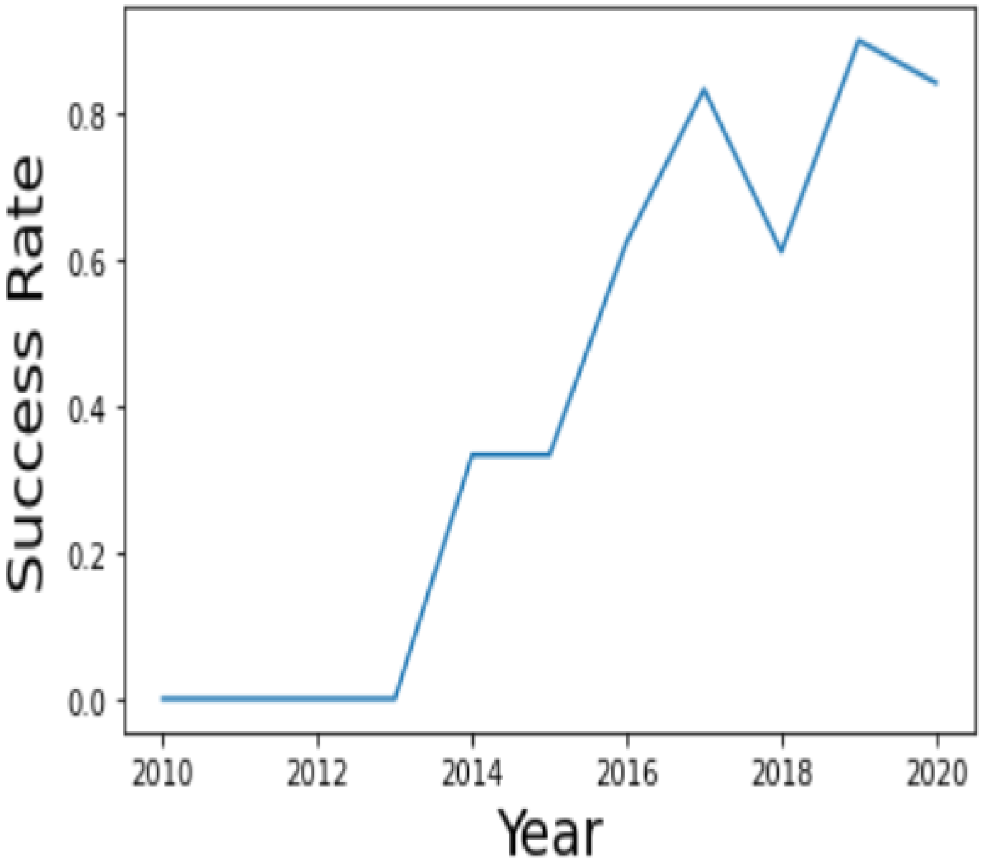


Payload vs. Orbit Type

- Most of the Launches are belongs to less than 10000 kg Payloads and most of them are successful
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- There is no clear relation is present for GTO orbit with Pay Load mass



Launch Success Yearly Trend



- The success rate since 2013 kept increasing till 2017
- There is decrease in 2018 on success rate and it again increased to 90% success in 2019
- Success Rate again decreased in 2020

All Launch Site Names

SQL Query

```
Select Distinct LAUNCH_SITE from SPACEXTBL;
```

Output

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

Query Explanation:

- There are 4 Unique Launch sites are present in US as shown in the output.
- Distinct function is used to fetch only unique LAUNCH_SITE from SPACEXTBL table

Launch Site Names Begin with 'CCA'

SQL Query

```
%%sql
SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' FETCH FIRST 5 ROWS ONLY;
```

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing _Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Query Explanation

- Using Like function we have fetched required data for the Launch_Site starts with ‘CCA’
- “Fetch First 5 rows only” will fetch only 5 rows as per requirement

Total Payload Mass

SQL Query

```
SELECT SUM(payload_mass__kg_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)';
```

Output

total_payload_mass
45596

Query Explanation:

- **This Query used aggregate function “SUM” to get to total.**
- **“WHERE” clause is used to specify the aggregate “SUM” restricted to CUSTOMER “NASA (CRS)”**

Average Payload Mass by F9 v1.1

SQL Query

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

Output

average_payload_mass
2534

Query Explanation:

- This Query used aggregate function “AVG” to get to average.
- “WHERE” clause is used to specify the aggregate “AVG” restricted to BOOSTER_VERSION ‘F9 V1.1’

First Successful Ground Landing Date

SQL Query

```
SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)';
```

Output

first_successful_landing
2015-12-22

Query Explanation:

- This Query used aggregate function “Min” to get to minimum/lowest date.
- “WHERE” clause is used to specify the aggregate “MIN” restricted to LANDING_OUTCOME which are success to Ground Pad.
- First Successful landing on ground pad achieved on 22nd Dec,2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
SELECT BOOSTER_VERSION,LAUNCH_SITE,PAYLOAD FROM SPACEXTBL
WHERE LANDING_OUTCOME='Success (drone ship)' AND
payload_mass__kg_ BETWEEN 4000 AND 6000;
```

Output

booster_version	launch_site	payload
F9 FT B1022	CCAFS LC-40	JCSAT-14
F9 FT B1026	CCAFS LC-40	JCSAT-16
F9 FT B1021.2	KSC LC-39A	SES-10
F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105

Query Explanation:

- Query has required fields as booter version, launch site, payload to fetch
- This Query used "WHERE" clause
  - To Restrict the result set only for successful landing on drome ship.
  - To Restrict the result set only for the landing with pay load mass between 4000 and 6000 kg.
- There are 4 Booster versions are present which satisfy given criteria. All of them are of Booster version **F9 FT B10*** and launched from either **CCAFS LC-40** or **KSC LC-39A** Launch Sites

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
SELECT MISSION_OUTCOME,COUNT(*) Count_mission_Outcome FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

Output

mission_outcome	count_mission_outcome
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Query Explanation:

- This Query used “Group By” clause to get the counts based on the Mission Outcome
- There are 101 total missions on which 1 failed in flight,99 are successful and 1 is successful but payload status is unclear

Boosters Carried Maximum Payload

SQL Query

```
SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE payload_mass__kg_ =(SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

Output

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:

- Query has required fields as booter version to fetch for flight with maximum payload
- This Subquery used to fetch
  - Max Payload mass in Kg by a flight in SPACEXTBL

All Versions are F9 B5 B10* series which satisfies the criteria.

2015 Launch Records

SQL Query

```
SELECT booster_version,launch_site,landing_outcome FROM SPACEXTBL
WHERE YEAR(DATE) ='2015' and
LANDING_OUTCOME='Failure (drone ship)';
```

Output

booster_version	launch_site	landing_outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Query Explanation:

- Query has required fields as booter version, Launch site and landing Outcome to fetch
- This "Where" clause is used to fetch
  - Year function is used to fetch Year from the Date column and checked if it is '2015'
  - Landing Outcome is checked to verify if landing outcome related to failure in Drone Ship.

All Versions are F9 v1.1 B101* series which satisfies the criteria and Launched from CCAFS LC-40.

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

SQL Query

```
SELECT * FROM (SELECT LANDING_OUTCOME,COUNT(*) OUTPUT_COUNT FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME)
ORDER BY OUTPUT_COUNT DESC;
```

Output

landing_outcome	output_count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Query Explanation:

- Internal query is created to create an intermediate table with landing outcome and corresponding count(using Group By) within the given Date range
- Outer Query sorts the internal intermediate table in descending order to get desired output

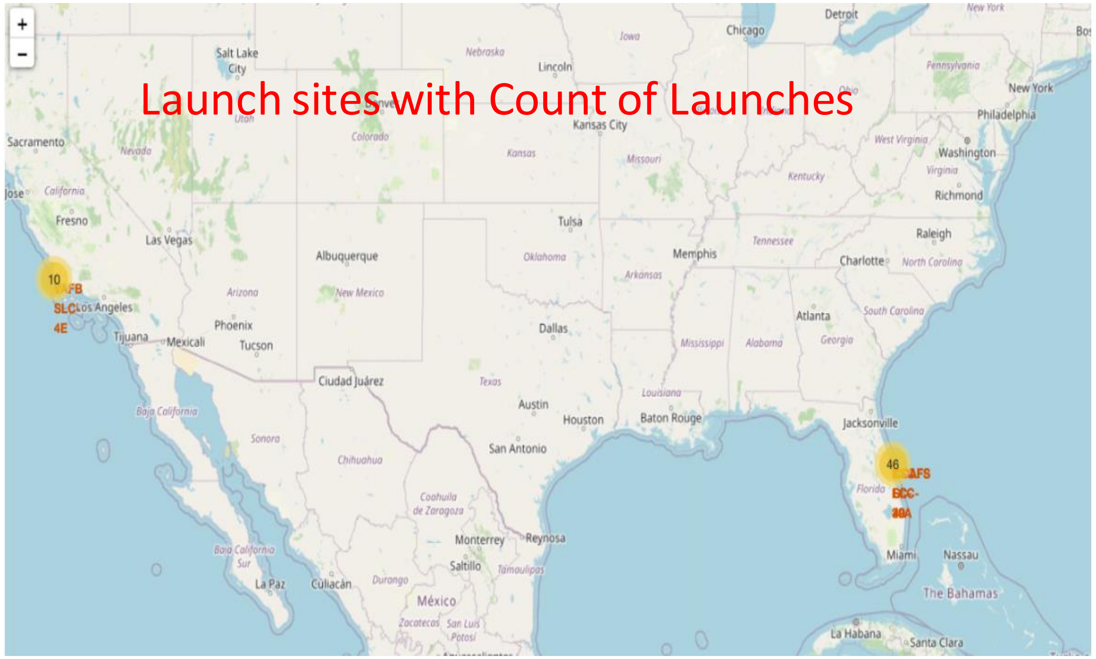
One third of the time attempts are not done to land the first stage which is highest in count in landing outcome, There are 11 times Drone ship landing is attempted out of which 5 times it is successful. All the 3 attempts to Ground pad landing is successful.

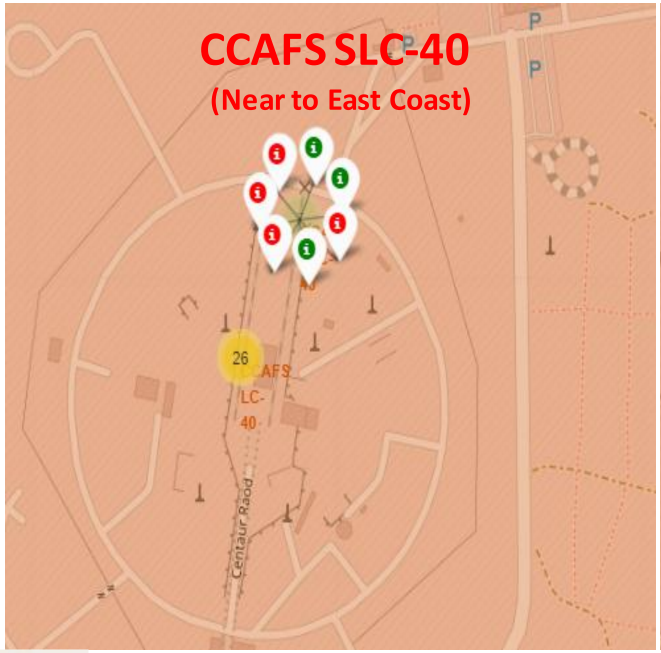
Section 4

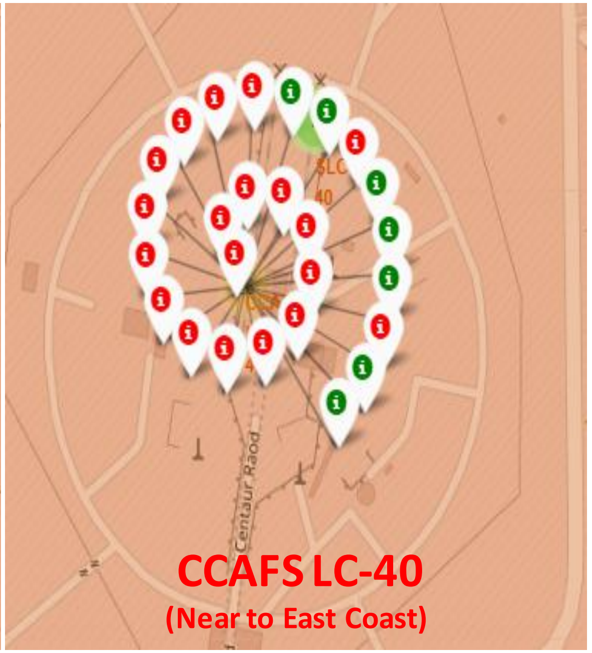
Launch Sites Proximities Analysis

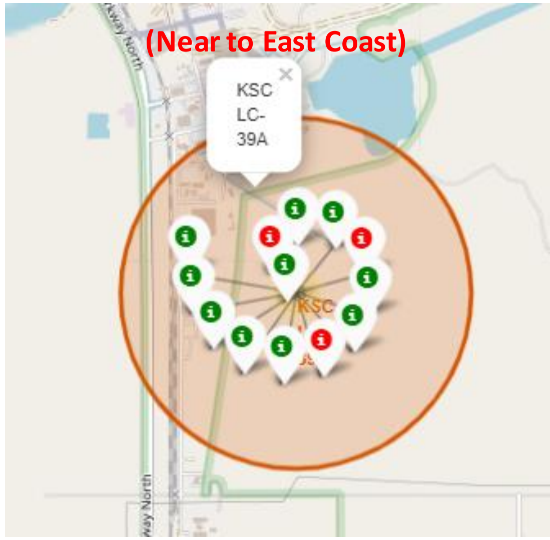
All Launch sites in USA Maps

Success/Failed launches for each site on the map





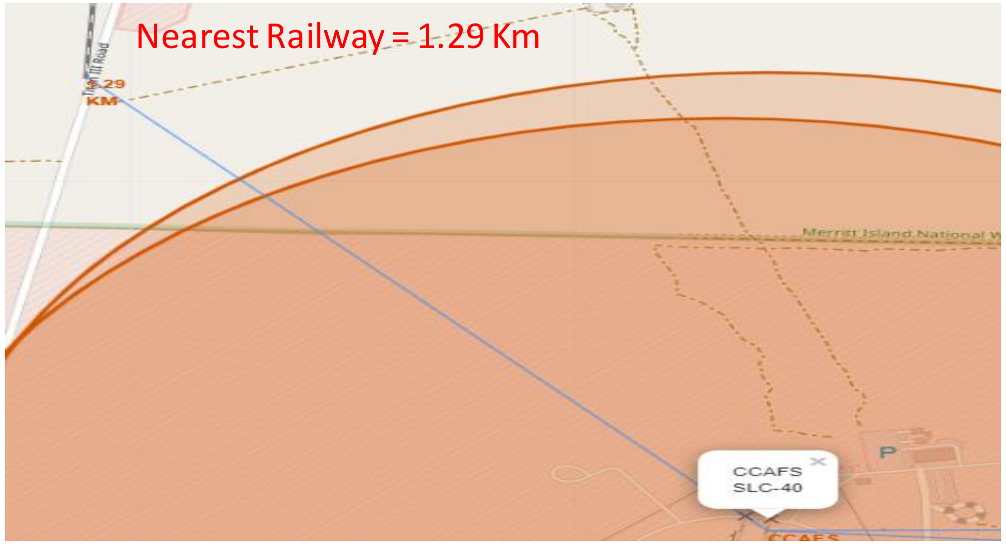




- **Number of launches are from Launch site near East Coast are 46 and from Launch site near West Coast is 16.**
- **In East Coast there are 3 Launch Sites whereas in Coast there is only one Launch Site.**
- **Launches are mostly done from CCAFS LC-40 launch site, but it is not most successful site. Most Successful Launch Site is KSC LC-39A.**

(Success and Failure are shown by Color labeled markers as Green and Red respectively)

Distances between Launch Site to its proximities
(For reference CCAFS SLC-40)



Are launch sites near railways? **Yes**

Are launch sites near highways? **Yes**

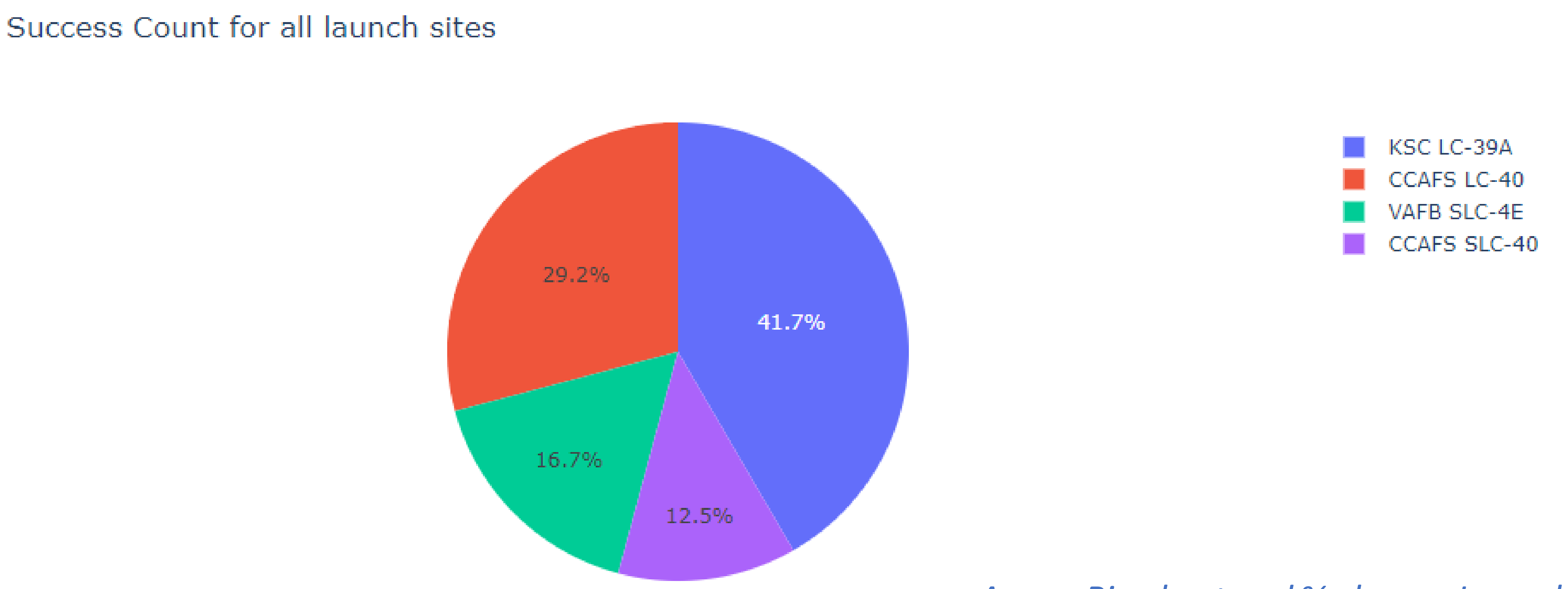
Are launch sites near coastline? **Yes**

Do launch sites keep certain distance away from cities? **Yes**

Section 5

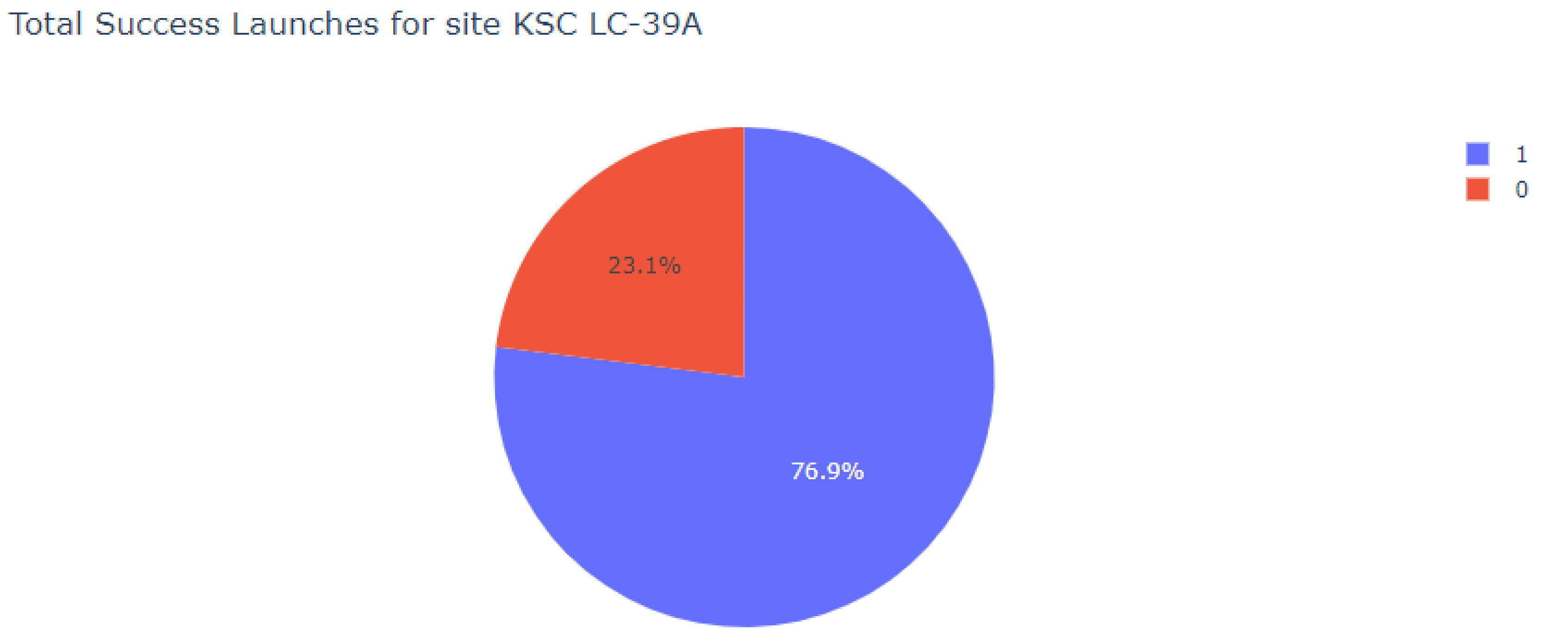
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each Launch Site



*As per Pie chart and % shown, Launch Site “**KSC LC-39A**” has more Success rate*

Pie chart for the launch site with highest launch success ratio (KSC LC-39A)

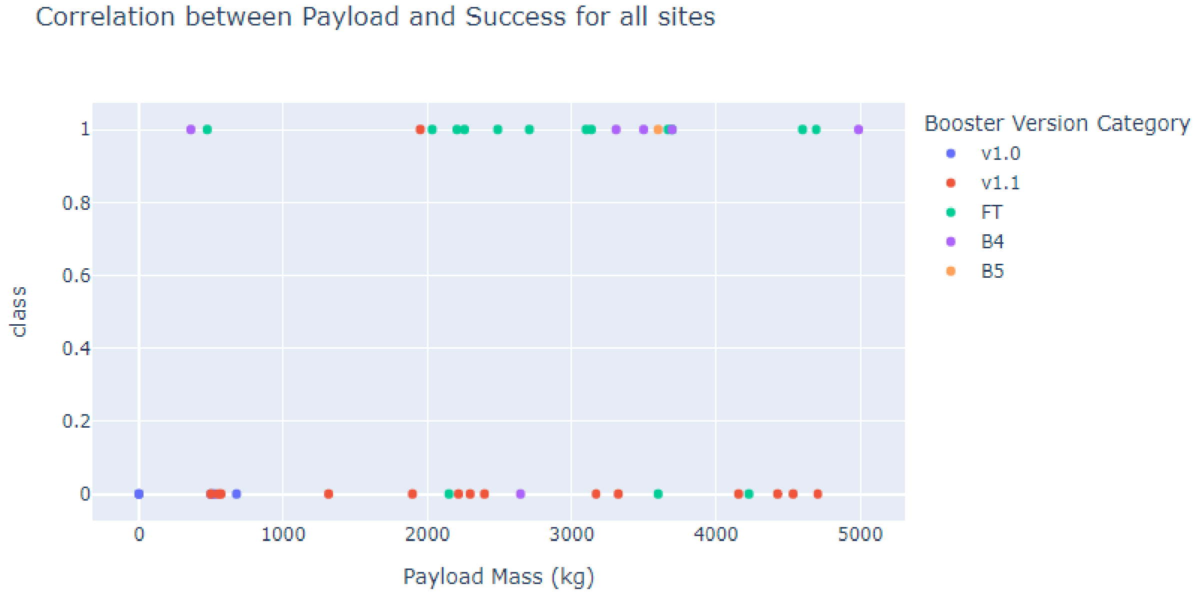


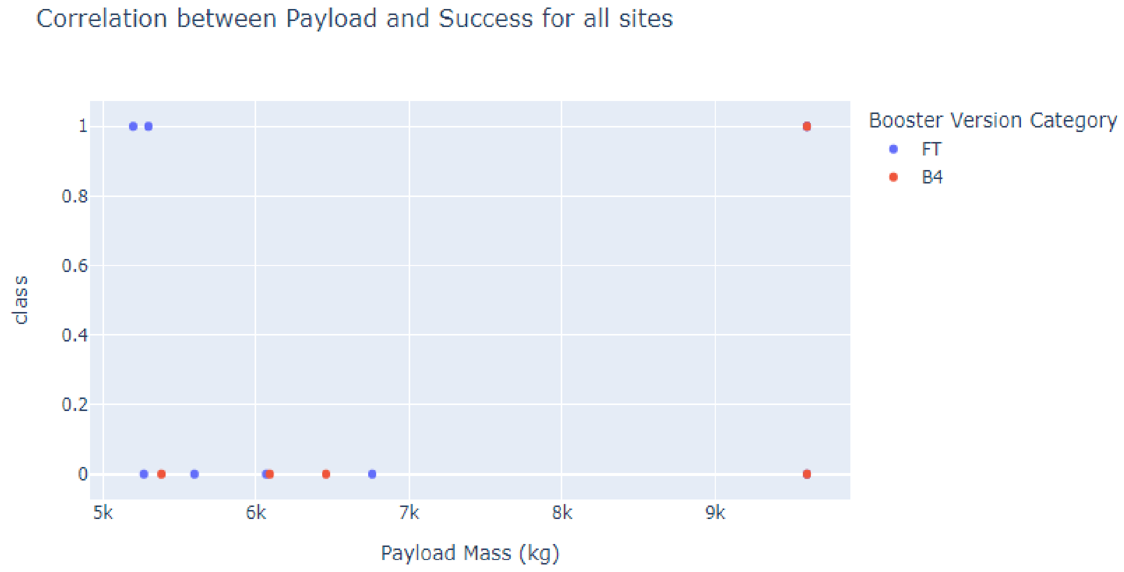
KSC LC-39A has Success Rate 76.9%

*(**1** is Success where as **0** is failure)*

Payload vs. Launch Outcome scatter plot for All Sites

(with different payload selected in the range slider)





Most of the Lunches are with Payload mass <= 5000 kg and Most of them are successful

*Within 5000 kg range Most Successful is Booster version is **FT** and less Successful is **V1.1***

*Only there are **two versions** which has launches with high Payload mass **(>5000 kg)***

*Only **B4** is the version has both success and failure with High Payload Mass**(>10000 Kg)***

Section 6

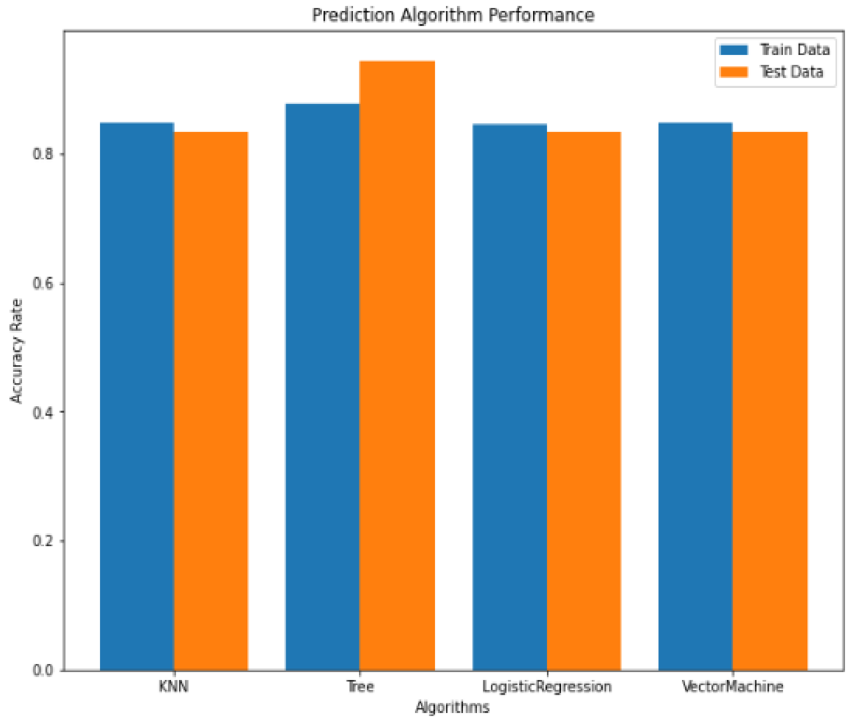
Predictive Analysis (Classification)

Classification Accuracy using Training and Test data

*As per the bar graph and display below it is clear that with Training Data "**Tree**" algorithm has highest accuracy (87.67%)*

*With Test data **"Tree"** Algorithm has highest accuracy 94.44%*

*So "**Tree**" algorithm wins the race with maximum accuracy with Training data as well with test data*



Algorithm	Training Data Accuracy	Test Data Accuracy
KNN	0.848214	0.833333
Tree	0.876786	0.944444
LogisticRegression	0.846429	0.833333
VectorMachine	0.848214	0.833333

```
Best Algorithm is Tree with accuracy :  0.8767857142857143
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
```

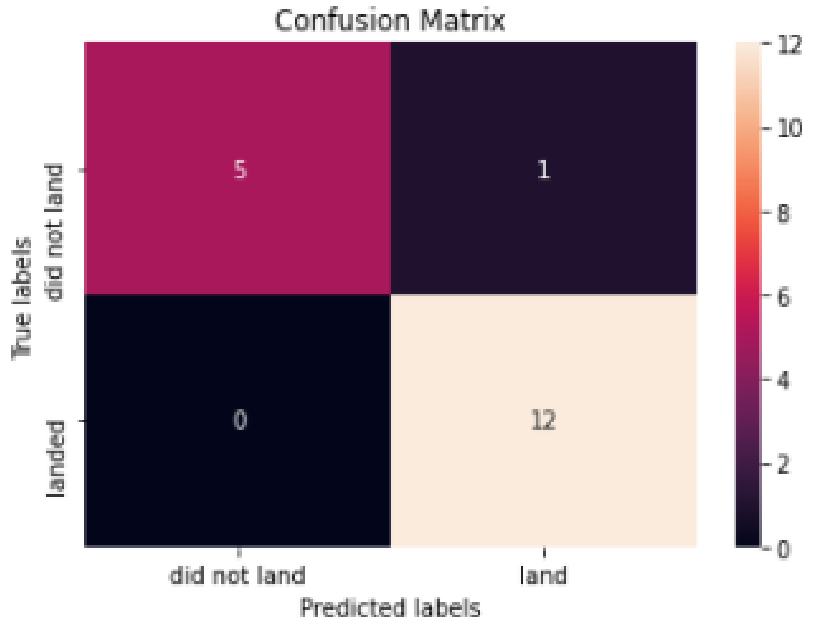
Above is the best Hyperparameter used by the "Tree" Algorithm used to achieve highest accuracy

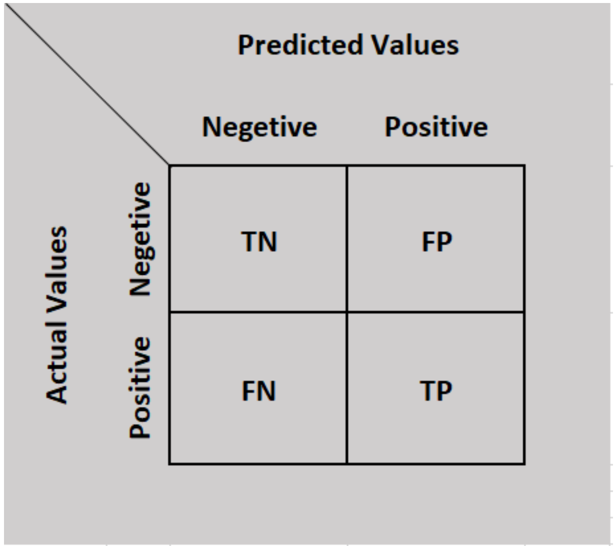
Confusion Matrix for Best Algorithm (Tree Algorithm)

*Examining the Confusion matrix, We can understand that **"Tree"** Algorithm with Best hyper parameter is capable of distinguish between different classes.*

We can see 94.44% times this Algorithm is able to identify True Positives and True Negative (17 out of 18)

Only for 5.56 % time it is giving False Positives which is only drawback of this Algorithm.





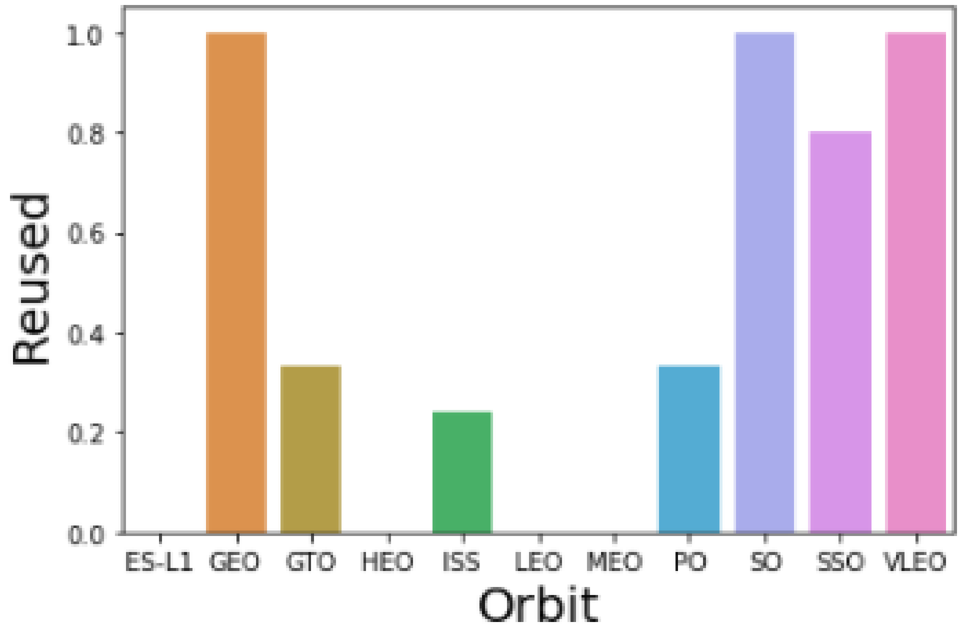
Conclusions

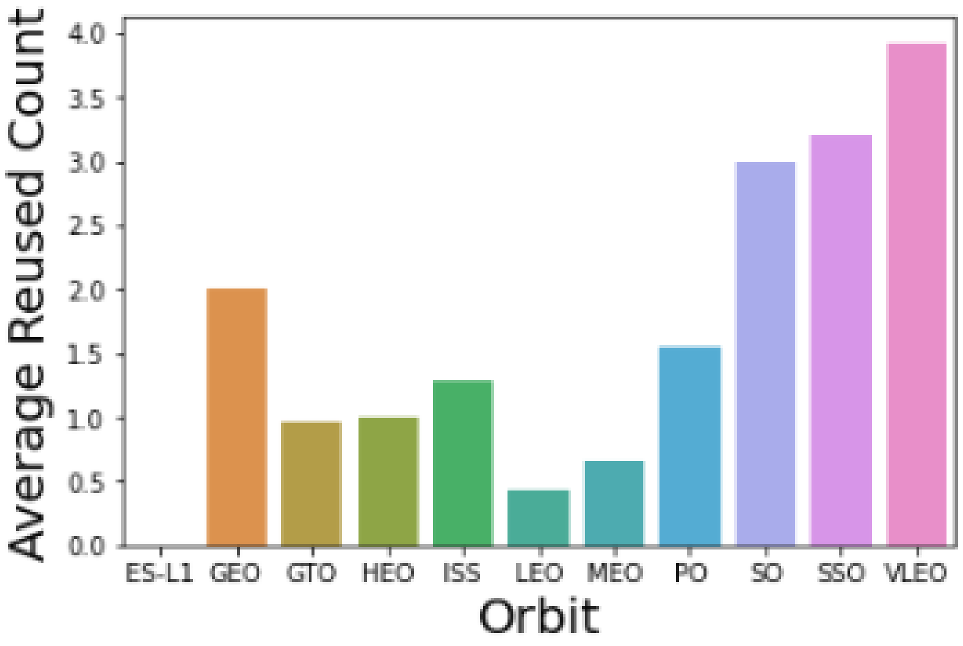
- *All the flights after Flight number 77 are successfully landed First Stage irrespective of Launch Site.*
- *All launches to Orbits ES-L1,GEO,HEO and SSO are Successful where as VLEO has 85% Success Rate*
- *With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.*
- *Launch successes are increased till 2017 and dropped in 2018 and again raised in 2019 which is again dropped in 2020.*
- *Most of the Mission are successful (100 out of 101 missions)*
- *F9 B5 B10* Booster version Launches are high pay load launches*
- *Launch sites have closeness with High Road, Coastline and railways but maintains certain distance from City*
- *KSC LC-39A Launch site has most Success Rate(76.9%)*
- *Most of the Launches are Low Payload(<5000 Kg) and Successful*
- *Tree Algorithm Providing best Prediction on Launch Success and failure(90% accuracy with Training data and 83.33% accuracy with Test Data)*

Appendix

Link for further analysis

Orbit Type vs Reusability

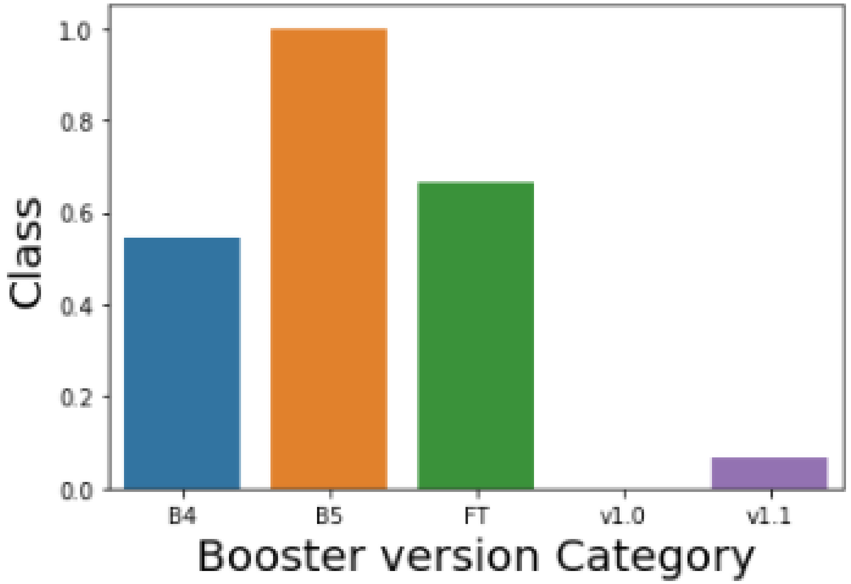


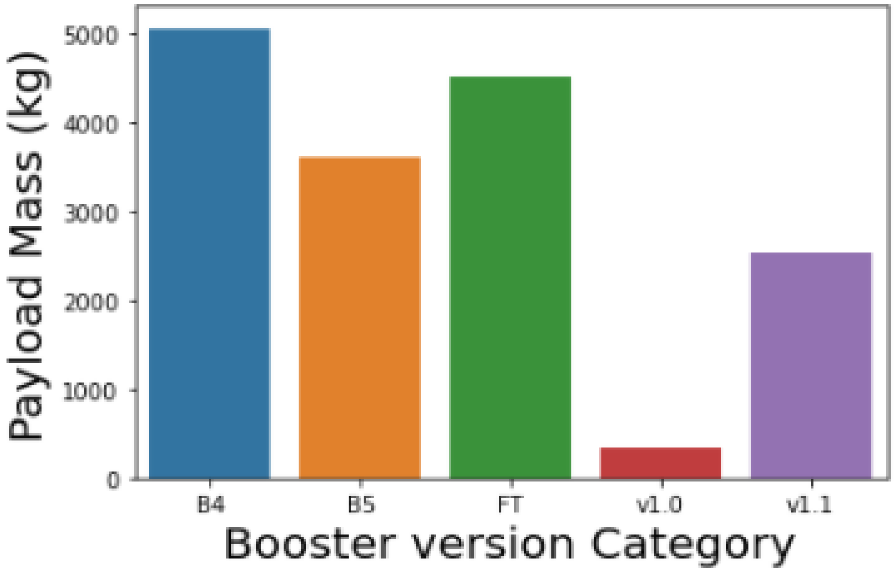


First stage to the Launches to Orbit type GEO,SO,VLEO are reused and for Orbit Type SSO it is used 80% of the times

First stage of VLEO orbit type launches are reused 4 times as an average

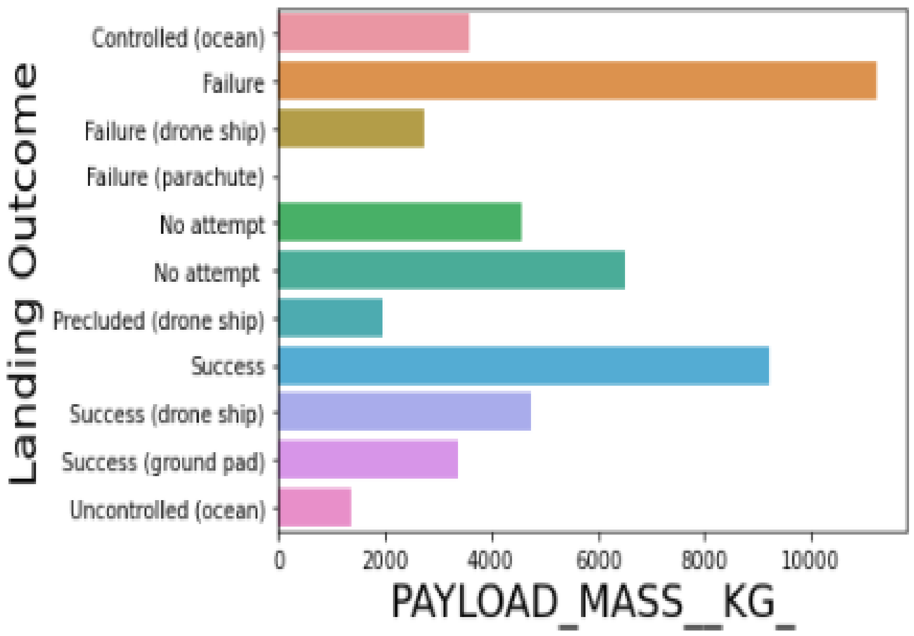
Booster version Category





- Booster version B4 has highest average Payload mass among other versions but Success rate of landing First Stage is less than 60%.
- Booster version B5 using average Pay load mass less than 4000 kg but providing 100% Success in landing the First Stage

Payload Mass Vs Landing Outcome



Launches with **High Payload Mass** are resulting in **“Failure”** in Landing Outcome

Reference

- Link of REST API
  - https://api.spacexdata.com/v4/rockets/
  - https://api.spacexdata.com/v4/launchpads/
  - https://api.spacexdata.com/v4/payloads/
  - https://api.spacexdata.com/v4/cores/
  - https://api.spacexdata.com/v4/launches/past

- Link of Wiki Page for Web Scraping
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Thank you!