IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sonica Sinha
23rd June, 2022

Outline

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

Executive Summary

❖ Summary of methodologies

- ✓ Data Collection through API
- ✓ Data Collection with Web Scraping
- ✓ Data Wrangling
- ✓ Exploratory Data Analysis with SQL
- ✓ Exploratory Data Analysis with Data Visualization
- ✓ Interactive Visual Analytics with Folium
- ✓ Machine Learning Prediction

❖ Summary of all results

- ✓ Exploratory Data Analysis result
- ✓ Interactive analytics in screenshots
- ✓ Predictive Analytics result from Machine Learning Lab

Introduction

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.

Problems -

1.) To determine the factors associated for successful landing of the rocket at first stage.
2.) Relationship with each rocket variables for each outcome at the end.
3.) Past history and its future prediction for successful landing using Machine Learning.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - From SpaceX Rest API provided by course instructors
  - Wikipedia via web scraping
- **Perform data wrangling**
  - Data was processed using one-hot encoding for categorical features
- **Perform exploratory data analysis (EDA) using visualization and SQL** by Scatter plot and bar graphs
- **Perform interactive visual analytics using Folium** by Geospatial map marking **and Plotly Dash** by dashboards
- **Perform predictive analysis using classification models**
  - Used 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.

General steps being used for collection of data are as follows---

1. Getting data from SpaceX Rest API and Wikipedia
2. Converted to dataframe via data scraping and wrangling
3. Extraction of required information from dataframe by filtration
4. Exported to table file
5. Finally to CSV for further analysis

Data Collection – SpaceX API

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	6123.547647	1.0
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	6123.547647	1.0
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	6123.547647	1.0
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	6123.547647	1.0
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	6123.547647	1.0

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

Getting response from SpaceX API

Converting it to .jason file

Custom functions to clean data applied

```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

List assigned to Dictionary to create Dataframe

Filtered and Exported to CSV

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

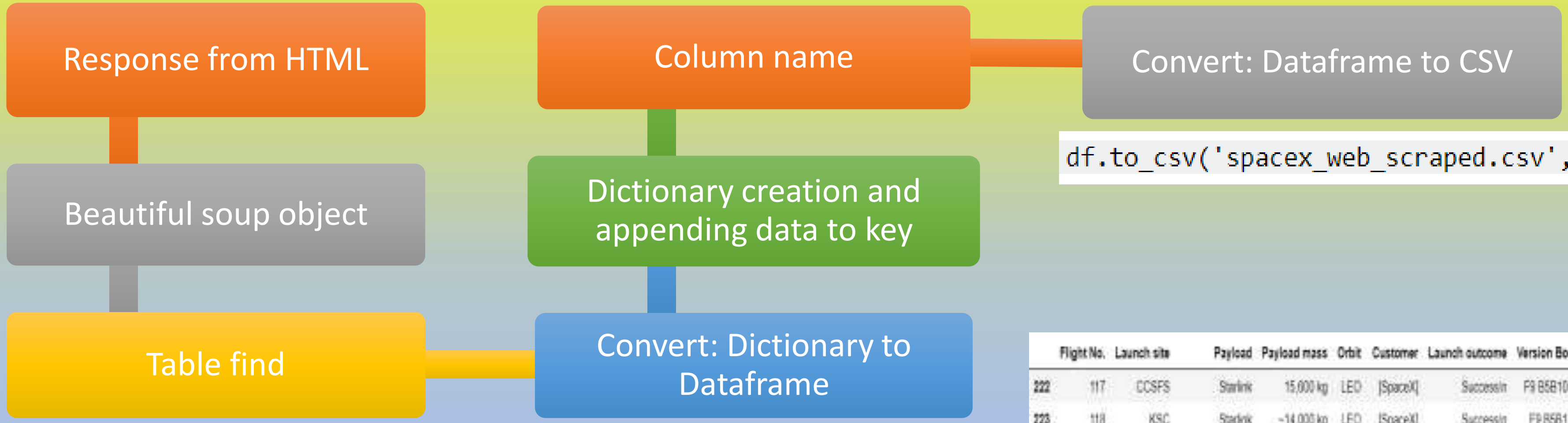

GitHub URL-https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/Capstone%20Project%20SpaceX%20.ipynb

Data Collection - Scraping

Data scraping, in its most general form, refers to **a technique in which a computer program extracts data from output generated from another program**. Data scraping is commonly manifest in web scraping, the process of using an application to extract valuable information from a website.

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

```
print(column_names)
```




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

```
# Use soup.title attribute
soup.title
```

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
df=pd.DataFrame(launch_dict)
df.tail(5)
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
222	117	CCSFS	Starlink	15,600 kg	LEO	[SpaceX]	Success\n	F9 B5B1051.10	Success	9 May 2021	06:42
223	118	KSC	Starlink	~14,000 kg	LEO	[SpaceX]	Success\n	F9 B5B1058.8	Success	15 May 2021	22:56
224	119	CCSFS	Starlink	15,600 kg	LEO	[NASA]	Success\n	F9 B5B1063.2	Success	26 May 2021	18:59
225	120	KSC	SpaceX CRS-22	3,328 kg	LEO	[Sirius XM]	Success\n	F9 B5B1067.1	Success	3 June 2021	17:29
226	121	CCSFS	SXM-8	7,000 kg	GTO	0	0	F9 B5	0	6 June 2021	04:26

GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/Data%20Collection%20with%20Web%20Scraping%20.ipynb

Data Wrangling

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis. With the amount of data and data sources rapidly growing and expanding, it is getting increasingly essential for large amounts of available data to be organized for analysis.

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

Number of launches calculated for each specific site

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Number of occurrence of each orbit calculated

Number of mission outcome per orbit type calculated

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

landing_outcomes
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key,value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

From outcome column, landing outcome label outcome

Exported to CSV

```
df.to_csv("dataset_part\_2.csv", index=False)
```


GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/Data%20Wrangling%20.ipynb

EDA with Data Visualization

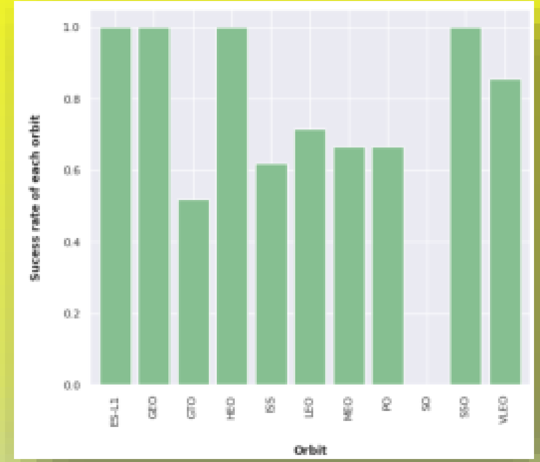
Scatter Plot Graphs

Following kinds of graphs has been prepared using matplotlib.pyplot.

- Flight number vs Payload Mass
- Flight number vs Launch site
- Payload vs Launch site
- Flight number vs Orbit type
- Payload vs Orbit type

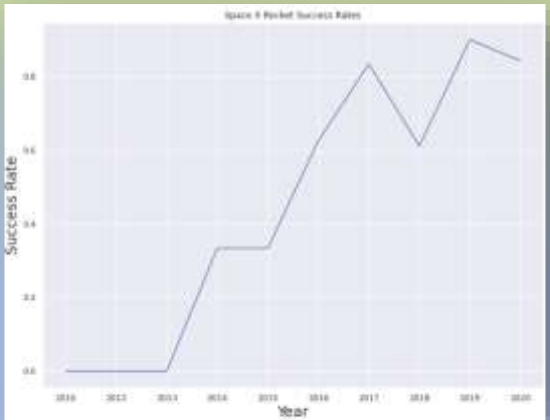
Bar Graphs

- Success rate of each orbit Vs Orbit



Line Graphs

- Success rate vs Year



GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

The processed for performing SQL, has been summarized in following points--

- 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 of 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 or success between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/EDA%20with%20SQL%20%20.ipynb

Build an Interactive Map with Folium

- Folium actually makes it very easy to understand and visualize data that has been processed using Python on interactive leaflet map. This library uses coordinates (latitude and longitude) for locating the each specific site and being circled with labeled name. Although, with easy interactive understanding to location on map, launch sites has also been demarcated, successful launches with **class '1'** as **Green** and failure launches with **class '0'** as **Red**. And also distance from the coastline also been calculated in km.

```python
# Import folium MarkerCluster plugin
from folium.plugins import MarkerCluster
# Import folium MousePosition plugin
from folium.plugins import MousePosition
# Import folium DivIcon plugin
from folium.features import DivIcon
```

```python
# Function to assign color to launch outcome
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)
```

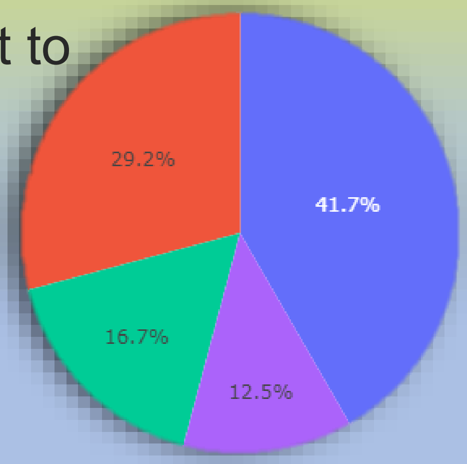
	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

```python
coordinate = [28.56213,-80.56751]
icon_ = folium.DivIcon(html=str(round(distance_coastline, 2)) + " km")
marker = folium.map.Marker(
    coordinate,
    icon=icon_
)
marker.add_to(site_map)
site_map
```


GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/Interactive%20Visual%20Analytics%20and%20Dashboard%20.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly dash is built which is completely user friendly and fun to play for the seeing the changes happening with different components.
- Pie chart is prepared showing total launches by a certain site and as a whole.
- A scatter plot graph is also prepared showing relation with outcome and payload mass (kg) for different booster version – which is again an user interactive plot to play with.



GitHub URL- https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets.
- Decide which type of ML to use.
- Set the parameters and algorithms to Grid Search CV and fit it to dataset.

Evaluating the Model

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

Finding Best Performance Classification Model

- Use Feature Engineering and Algorithm Tuning
- The model with the best accuracy score will be the best performing model.

GitHub URL - https://github.com/Soni-Test/Applied-Data-Science-Capstone-Project-/blob/Master/Machine%20Learning%20Prediction%20.ipynb

Results

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

Section 2

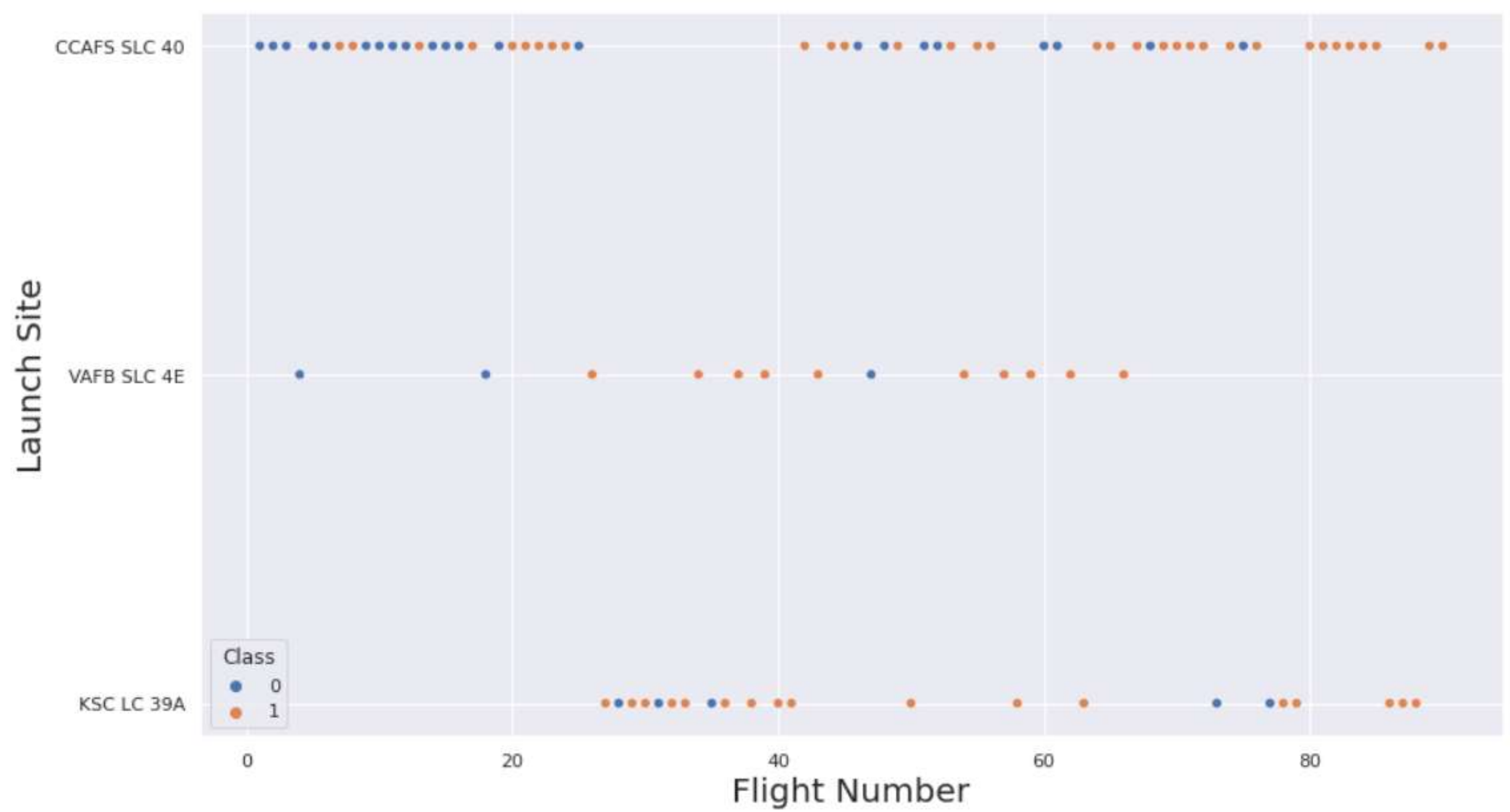
Insights drawn from EDA

Flight Number vs. Launch Site

Description

Presented scatter plot shows the larger the flights amount in the launch site, the greater will be the success rate.

However, site CCAFS SLC40 shows the least pattern.

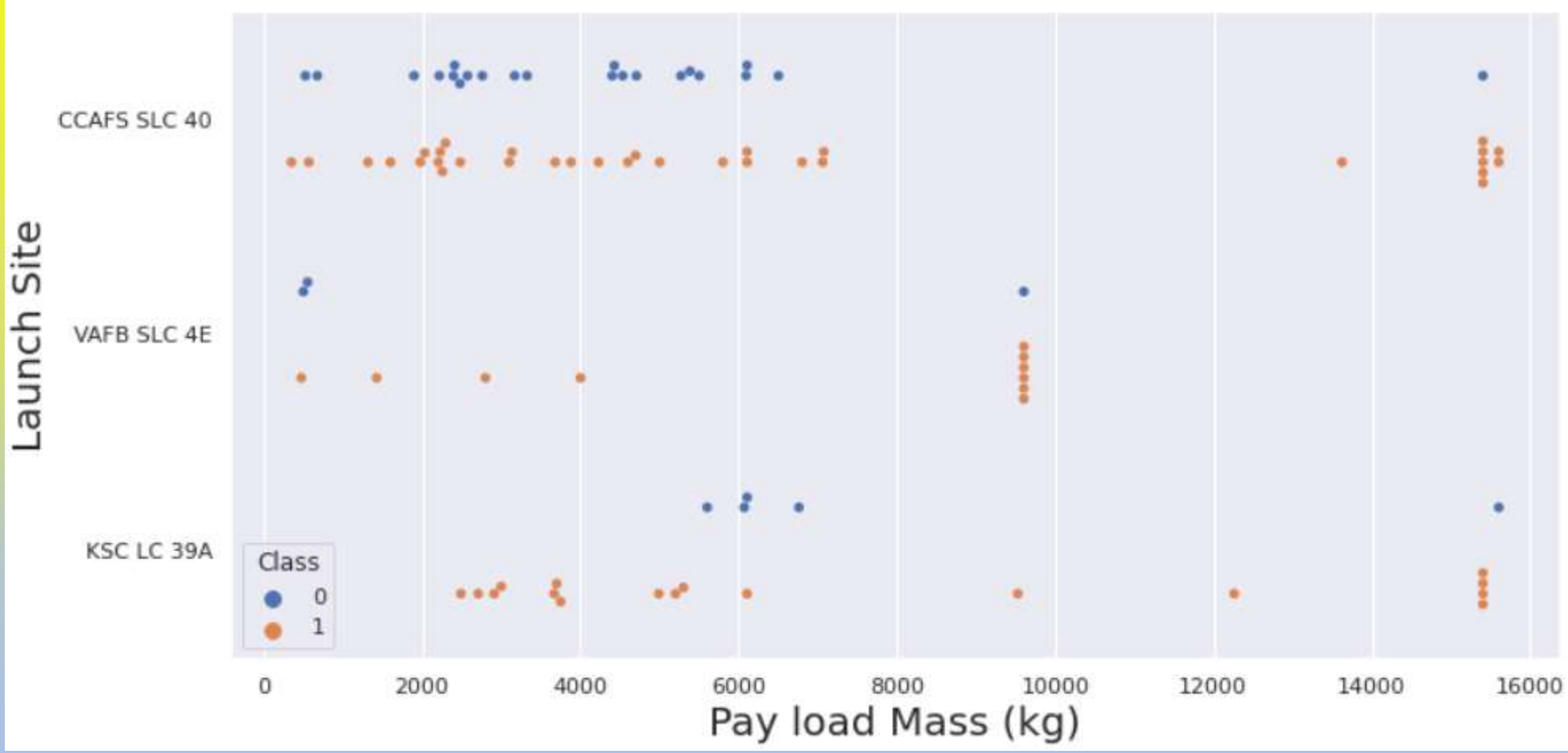


Payload vs. Launch Site

Description

Presented scatter plot shows the pay load mass is greater than 7000kg, and the probability of the success rate will be highly increased.

However, there is no clear and particular pattern for saying the launch site is completely dependent to the pay load mass for the success rate.

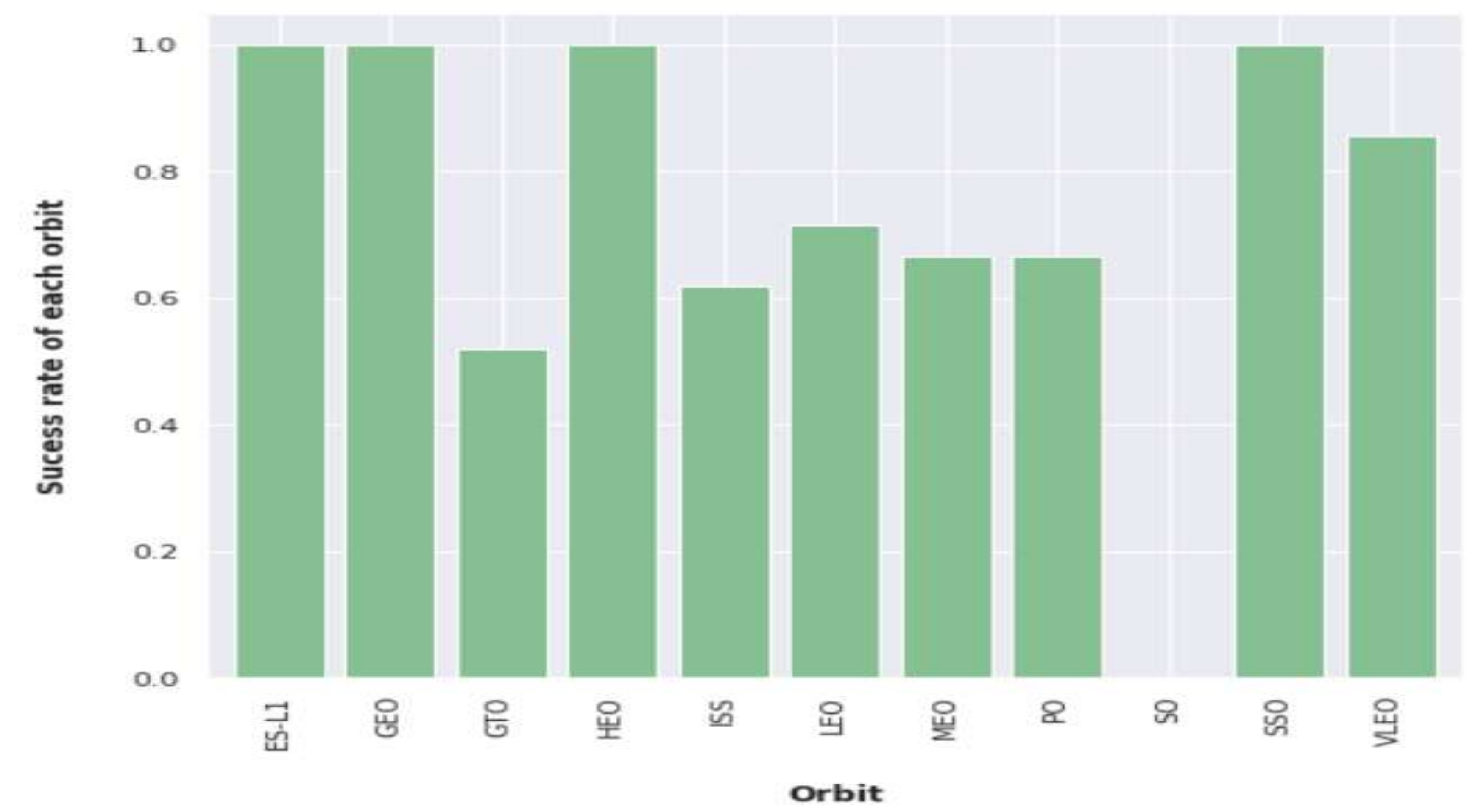


Success Rate vs. Orbit Type

Description

Present graph shows the possibility of the orbits to influences the landing outcomes which is for some orbits it is 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success.

However, deeper analysis show
that 4 orbits has occurrence of 1 such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to seen pattern or trend before we draw any conclusion.

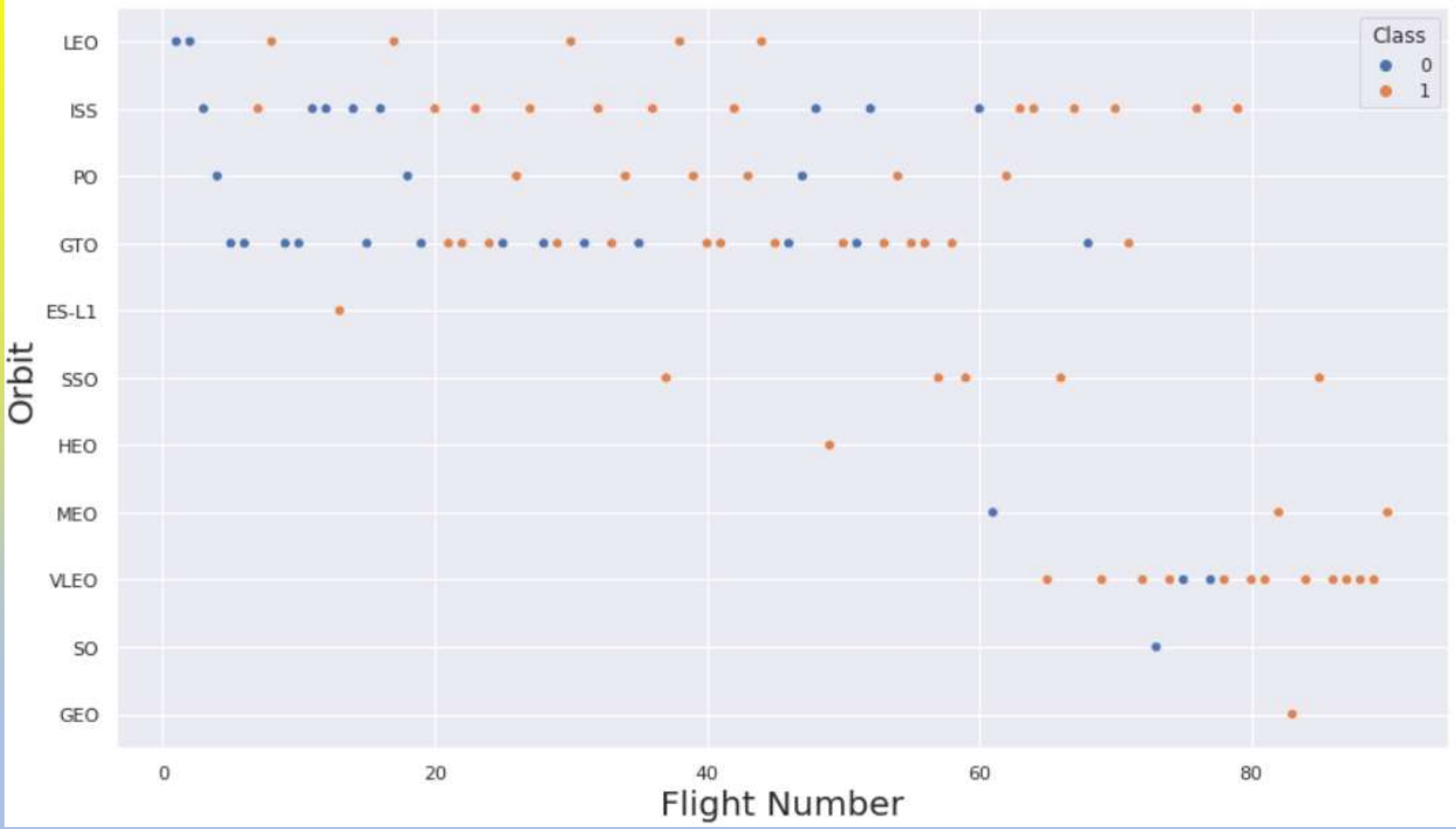


Flight Number vs. Orbit Type

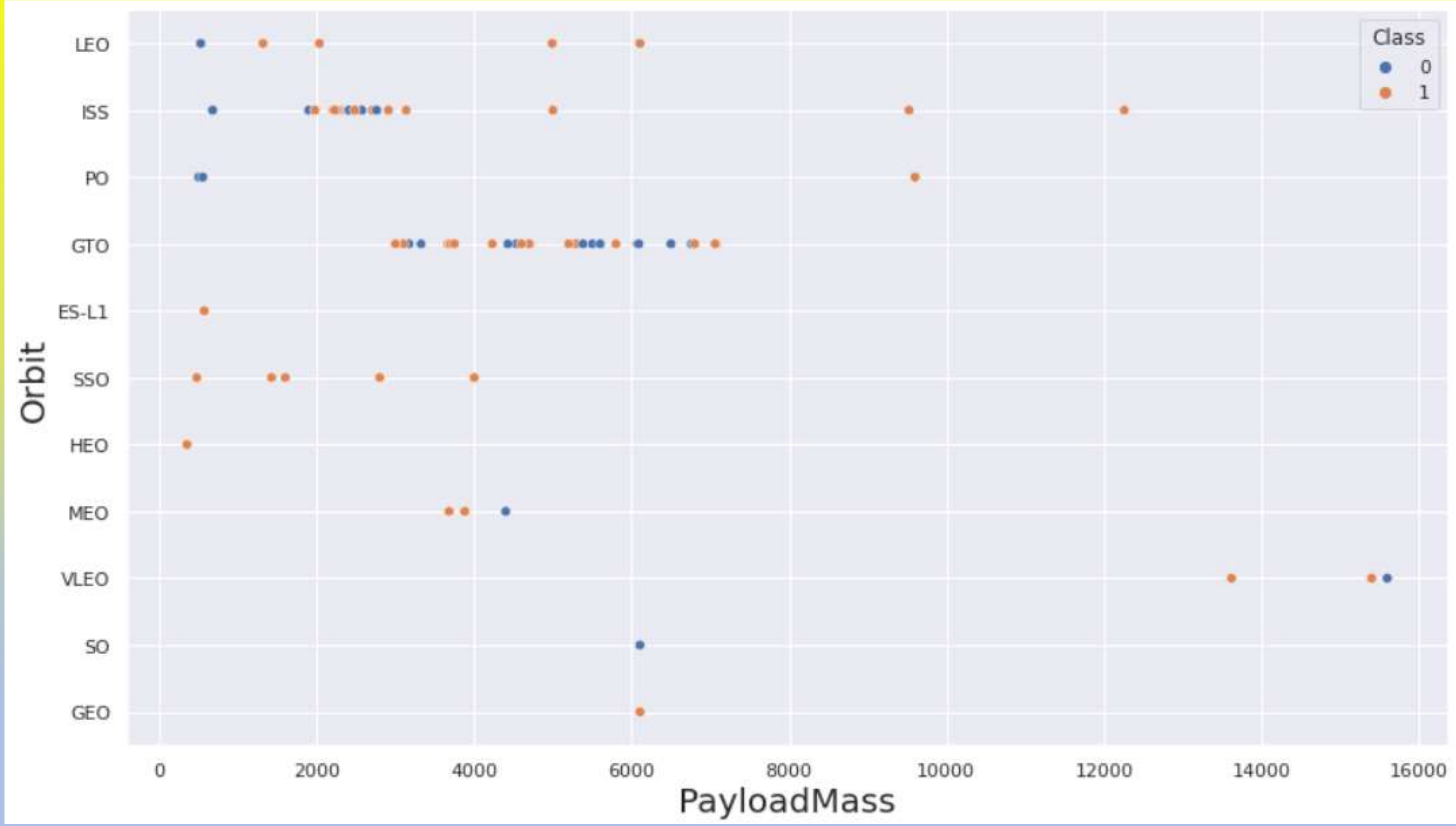
Description

Present scatter plot shows that, larger the flight number on each orbits, the greater will be the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.

Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.



Payload vs. Orbit Type



Description

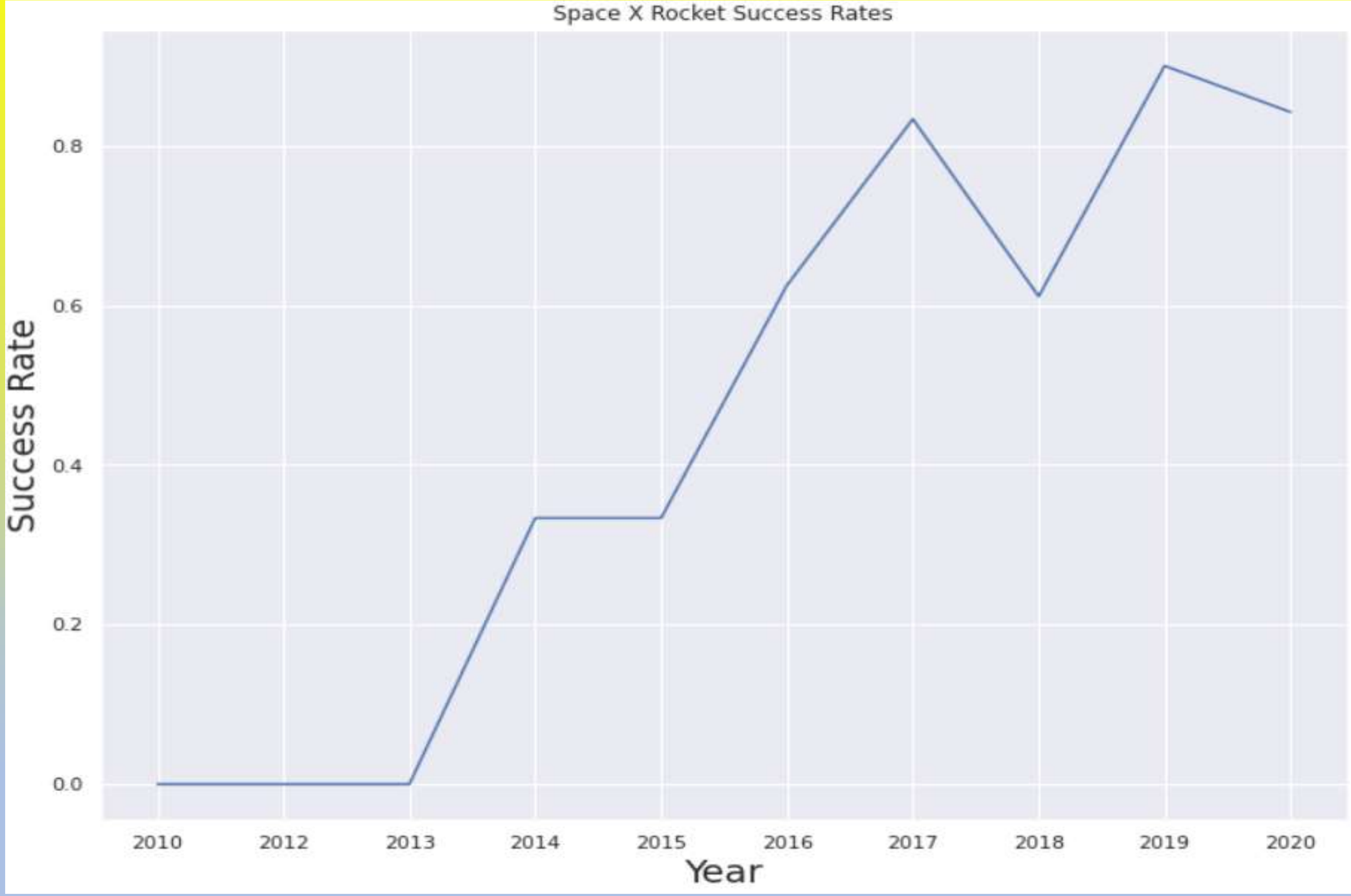
Heavier payload is showing the positive impact on LEO, ISS and P0 orbit. However, it is also showing the negative impact on MEO and VLEO orbit. GTO orbit seem to depict no relation between the attributes. Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.

Launch Success Yearly Trend

Description

The present trend clearly showcases the increasing trend straight from year 2013 to 2020 but with some minor dips in year 2015, 2018 and bit in 2020.

If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

SQL Query

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

Launch Sites

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

Description

Here the query word DISTINCT is used to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

SQL Query

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

Launch Sites

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

Description

Here the '%sql' is used to display the 5 records where launch sites begin with `CCA`

Total Payload Mass

SQL Query

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

Total Payload Mass

Total Payload Mass by NASA (CRS)
45596

Description

Here the total payload mass is been calculated with ‘SUM’ query to carried by boosters from NASA which is 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';
```

Total Payload Mass

Average Payload Mass by Booster Version F9 v1.1
2928

Description

Here, the average payload mass is for the carried booster version F9 v1.1 as 2928.4.

First Successful Ground Landing Date

SQL Query

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

Result

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

Description

Here, the min() function is used to find out the result, which shows the date of the first successful landing outcome on ground pad on 22nd December, 2015.

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;
```

Result

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

Description

Here the WHERE clause is used to filter out for the boosters which have successful landed on the drone ship and applied the AND condition to determine whether the successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query with Result

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

 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Successful Mission
100

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

 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Failure Mission
1

```
%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%';
```

 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Total Number of Successful and Failure Mission
101

```
%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;
```

 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.

Successful Mission	Failure Mission
100	1

Description

Here for selecting the number of successful and failure mission outcomes, subquery of ‘Success%’ and ‘Failure%’ is used to filter for WHERE Mission Outcome.

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

Result

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

Description

Here for the boosters carried maximum payload, subquery in the WHERE clause and the MAX() function is used.

2015 Launch Records

SQL Query with Result

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

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

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

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

MONTH	booster_version	launch_site
1	F9 v1.1 B1012	CCAFS LC-40
4	F9 v1.1 B1015	CCAFS LC-40

```
%sql SELECT {fn MONTHNAME(DATE)} as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

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

Description

Here for the launch records of 2015, the combinations of the WHERE clause, LIKE, AND, and BETWEEN
conditions are used to filter the failed landing outcomes in drone ship, their booster versions, and launch site names.

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

SQL Query with Result

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Landing Outcome	Total 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

```
%sql SELECT COUNT(LANDING__OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-20" FROM SPACEXTBL \
WHERE LANDING__OUTCOME LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

```
 * ibm_db_sa://zrb00667:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Rank success count between 2010-06-04 and 2017-03-20
8

Description

Here the Landing outcomes is selected along with the COUNT of landing outcomes from the processed data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

Further the GROUP BY clause is applied to group all the landing outcomes and then the ORDER BY clause is used to order the grouped landing outcome in descending order.

Section 3

Launch Sites Proximities Analysis

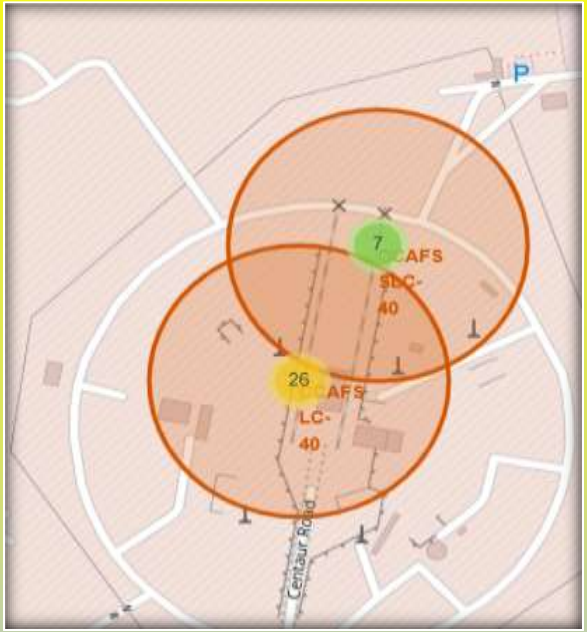
Site Location of Launch Area

SpaceX sites at Florida.

SpaceX sites at California.

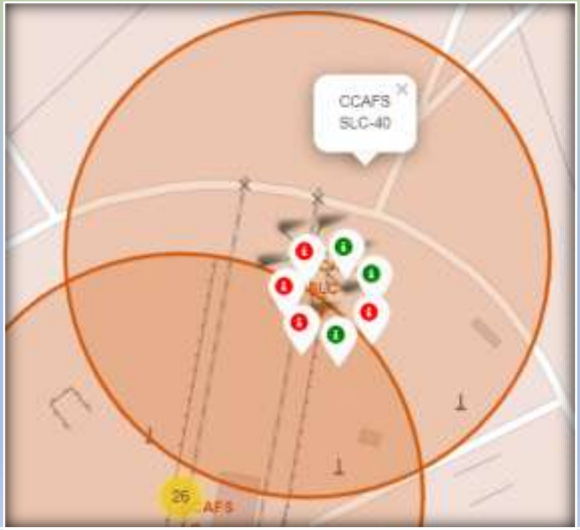
All the SpaceX sites located in United States of America in different states of Florida and California.

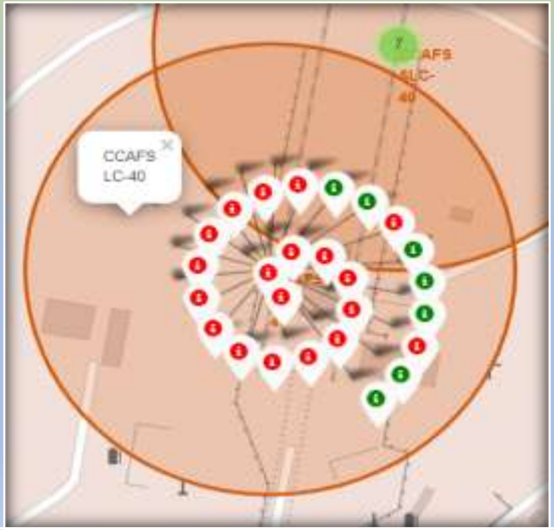
Launch Sites With Color Labels

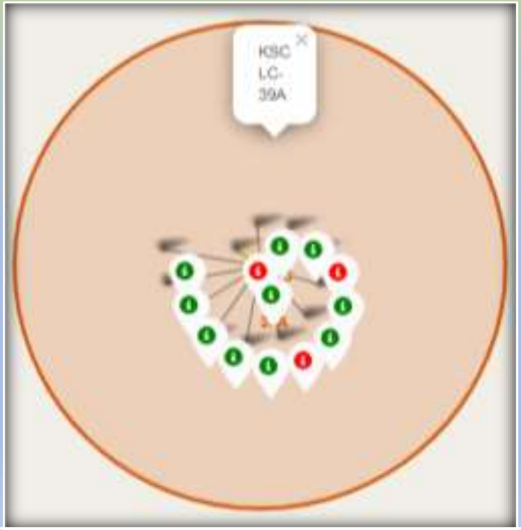


<u>*Green*</u> marker shows the successful sites.
<u>*Red*</u> marker shows the failure sites.

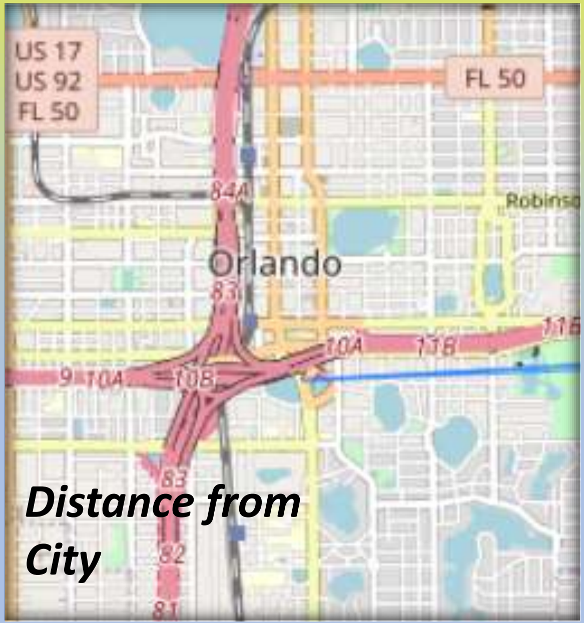
Among all the launch sites, it is clearly seen that **KSC LC-39A** has most successful launches.







Launch Sites Distance to Landmarks

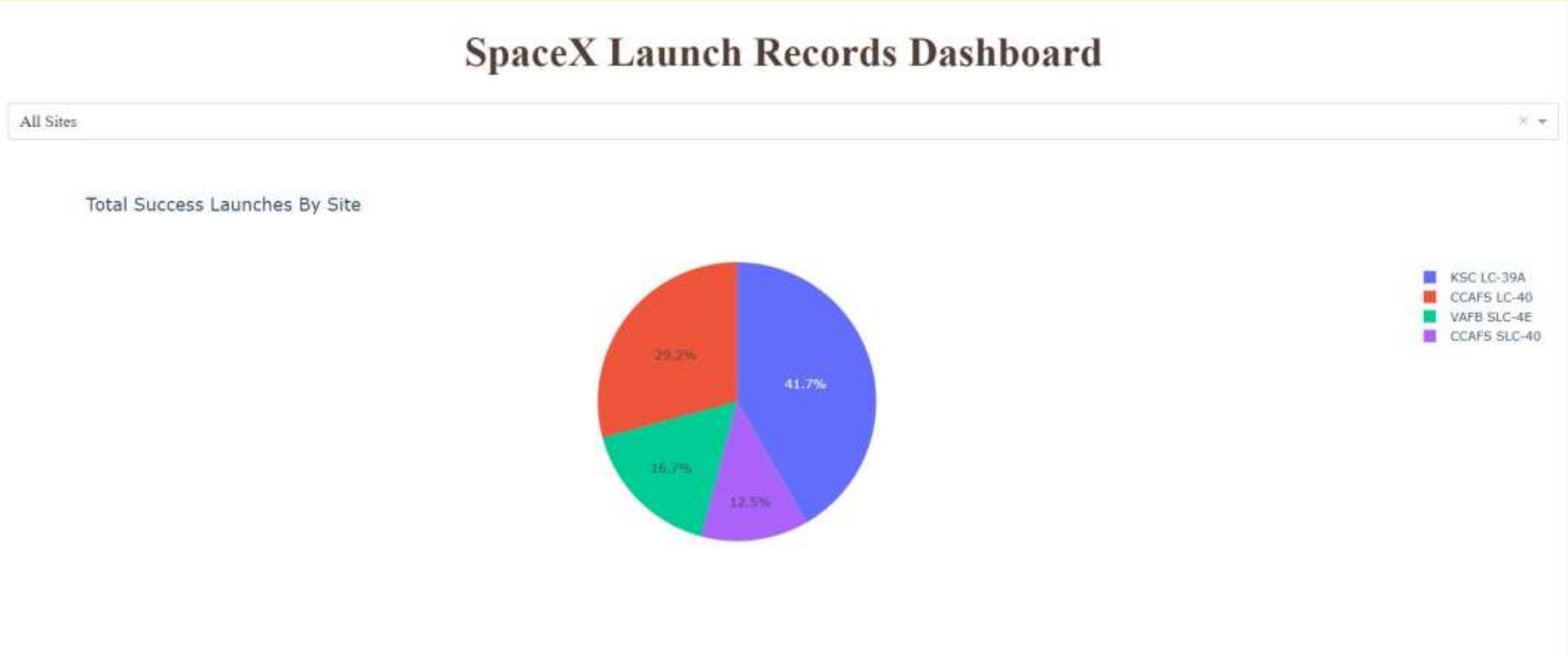


- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

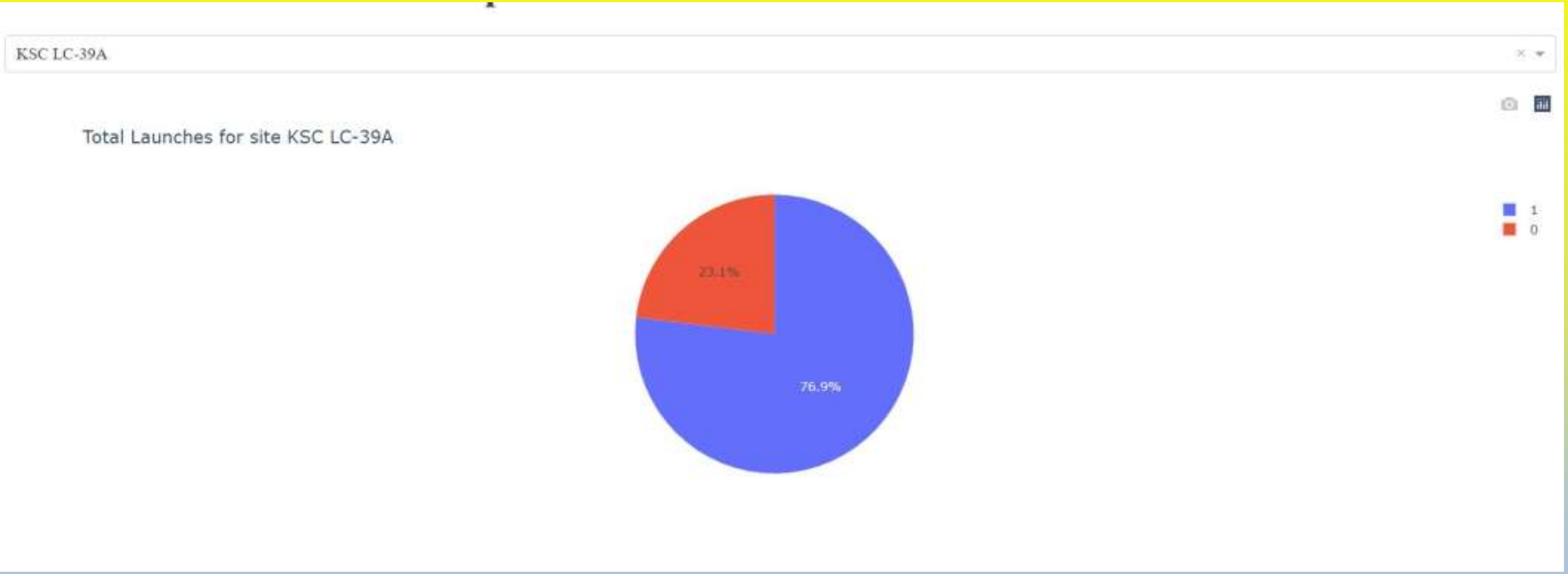
Section 4

Build a Dashboard with Plotly Dash

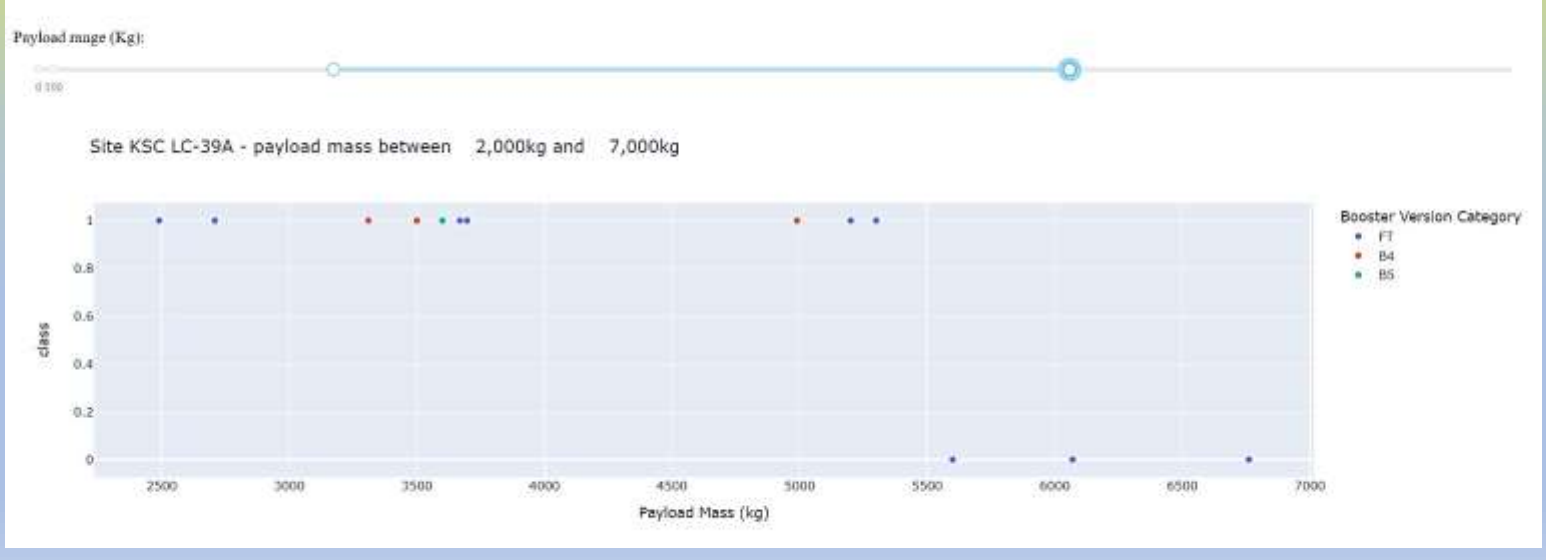
Success Launch of All Site



Highest Launch Success Ratio



Payload vs Launch Outcome Scatter Plot



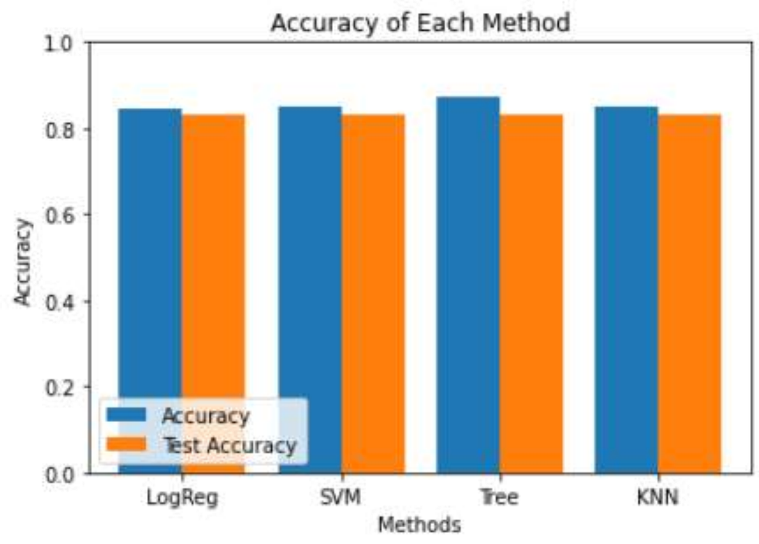
Section 5

Predictive Analysis (Classification)

Classification Accuracy

As it can be seen that the best classification accuracy using Machine Learning algorithm came up with **Decision Tree**, i.e. **<u>0.875</u>**.

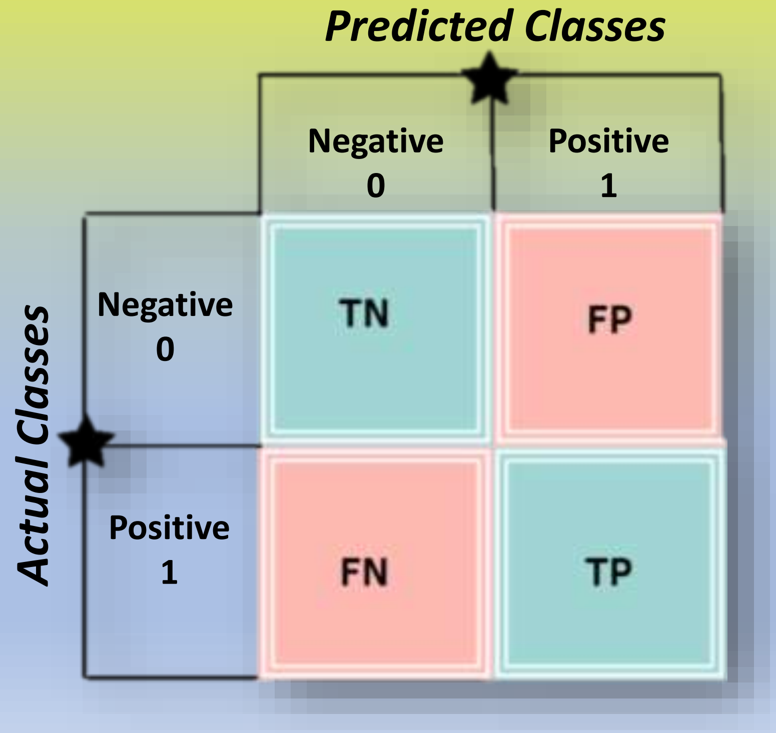
Model	Accuracy	Test Accuracy
Log Reg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.875	0.83333
KNN	0.84821	0.83333

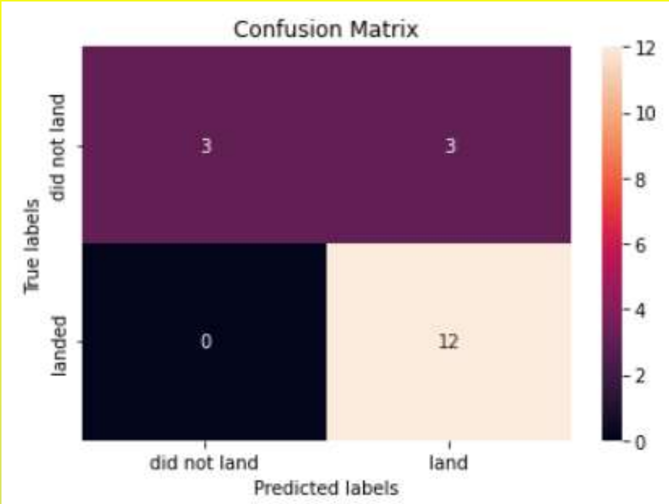


Confusion Matrix

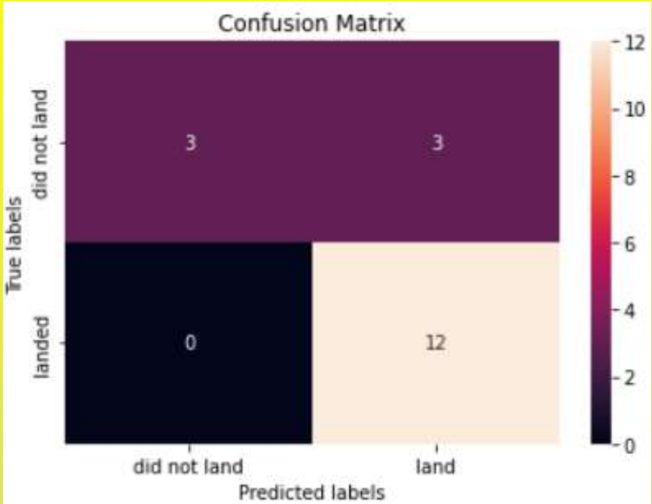
A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:

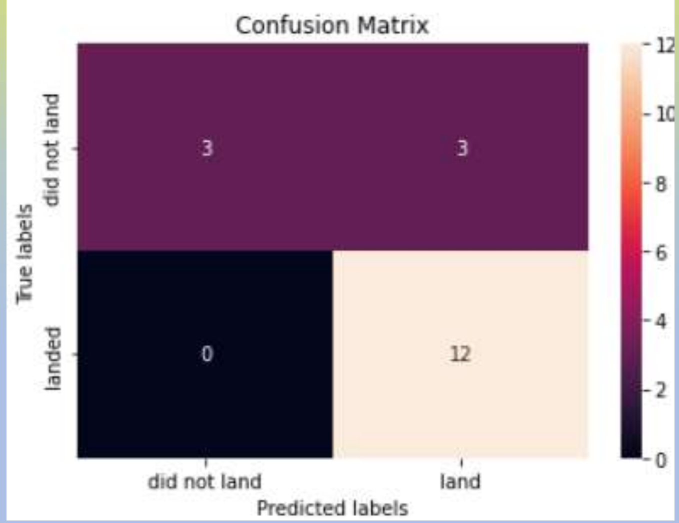




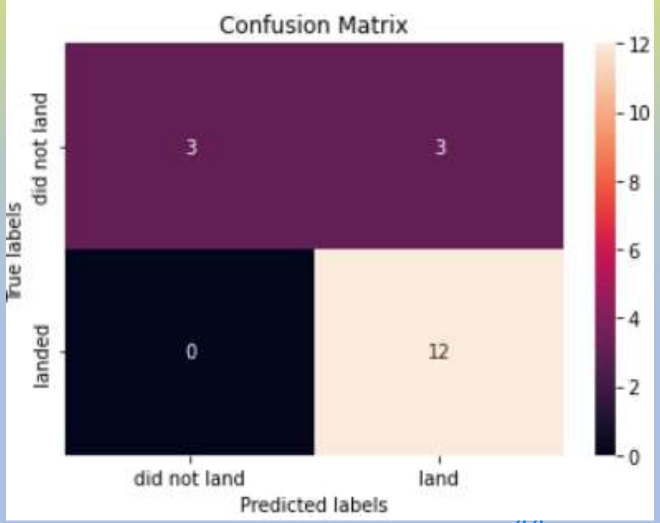
Logistic Regression



SVM

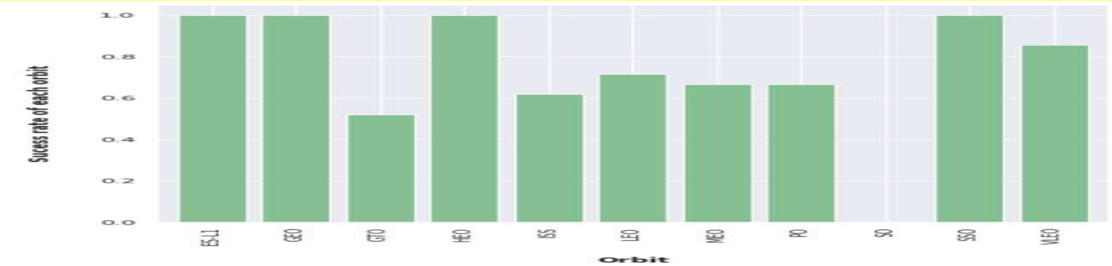


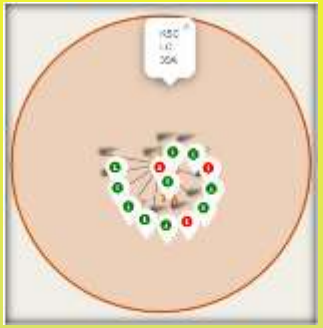
Decision Tree

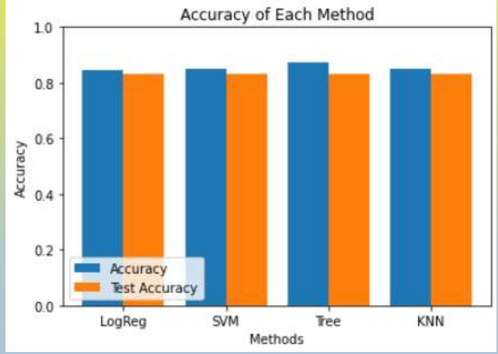


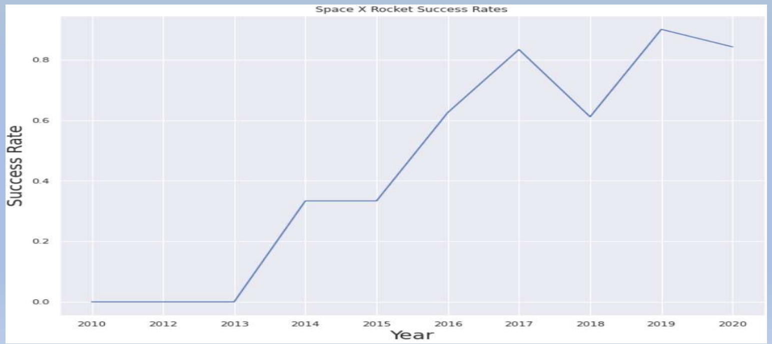
KNN

Conclusions









- Landing outcomes for some orbits SSO, HEO, GEO and ES-L1 has 100% success rate while SO orbit produced 0% rate of success.
- Among all the launch sites, KSC LC-39A has most successful launches.
- Trend showcases the increasing trend straight from year 2013 to 2020 but with some minor dips in year 2015, 2018 and bit in 2020 but still high.
- Best classification accuracy using Machine Learning algorithm came up with Decision Tree, i.e. 0.875.
- The average payload mass is for the carried booster version F9 v1.1 as 2928.4.

Appendix

SpaceX sites at Florida.

SpaceX sites at California.

All the SpaceX sites located in United States of America in different states of Florida and California.

Thank you!