IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Erwin Ide
2021-09-27

Outline

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

Executive Summary

Summary of methodologies:

- Data collection: SpaceX API and web scraping
- Data wrangling
- Exploratory Data Analysis (EDA) and Visualization using SQL, Pandas, and Matplotlib
- Interactive Visual Analytics and Dashboard using Folium and Plotly Dash
- Predictive Analysis

Summary of all results:

- Higher experience better success rate.
- Most efficient Orbits: ES-L1, GEO, HEO, SSO and VLEO
- Best launch site: KSC LC-39A
- Best accuracy model: Decision Tree Classifier

Introduction

Space Y that would like to compete with SpaceX.

We want to determine:

- Factors of a successful landing
- The price of each launch.
- SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

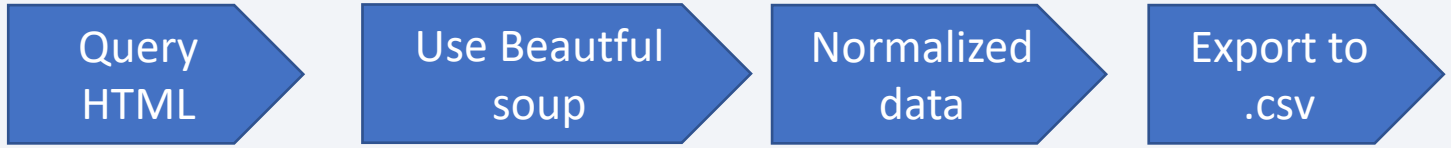
Executive Summary

- Data collection methodology:
  - Data collection: SpaceX API and web scraping
- Perform data wrangling
  - One hot encoding for non-numeric fields
  - Dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
  - SQL queries
  - Graphs to visualize patterns.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Maps and interactive dashboard.
- Perform predictive analysis using classification models
  - Maching learning models

Data Collection

- I used data from two sources:
    1. Wikipedia for the list of falcons and launches
    2. Rest API from SpaceX for the launches specifications and results.

- Wikipedia DATA:



- API Data:

Data Collection – SpaceX 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.- Json normalize file

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

3.- Apply Custom funtion to clean data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4.- Dict, Filter and Dealing with Missing Values

```
data = pd.DataFrame.from_dict(launch_dict)
```

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

5.- Dataframe and exporting to .csv

https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/a53d31de4e960e16fca36c1da1961a17c5a8c08c/Complete%20the%20Data%20Collection%20API%20Lab.ipynb

Data Collection - Scraping

1.- Getting response from HTML

```
wiki = requests.get(static_url).content
```

2.- Creating BeautifulSoup object

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

3.- Finding tables

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

4.- Dictionary and append data

```
launch_dict= dict.fromkeys(column_names)
```

5.- Converting dictionary to DataFrame.

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

6.- Exporting dataframe into .csv

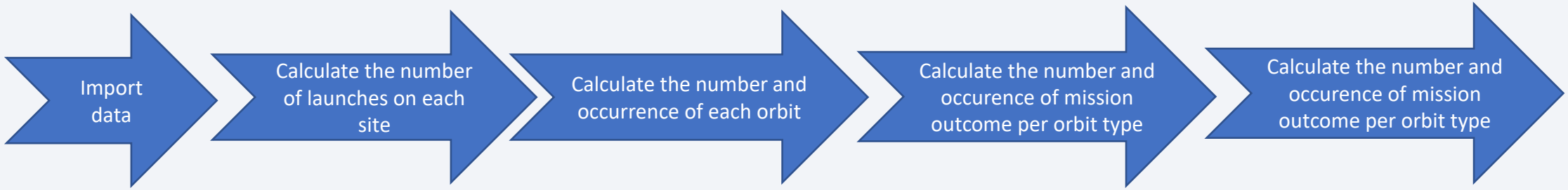
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

- https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/19cc06ab15cef0bfef719f140770c21acb5deb69/jupyter-labs-webscraping.ipynb

Data Wrangling

We performed an exploratory data analysis (EDA) to find some patterns in the data and determine what the label would be for training supervised models.

Process:



https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/bf219a29586a0b8f2ce4e7a5cd54ea3b63014caa/labs-jupyter-spacex-Data%20wrangling.ipynb

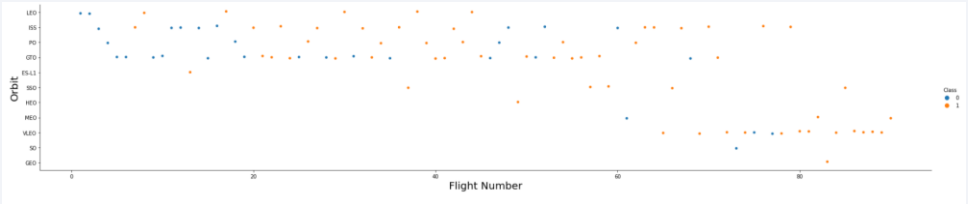
EDA with Data Visualization

SCATTER PLOTS:

Scatter plots were used here to show the relationship between two variables

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type

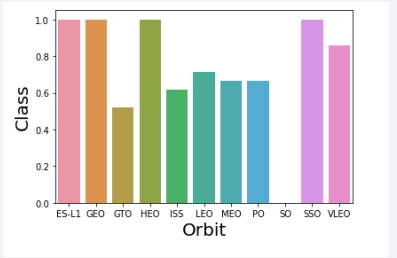
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose



BAR CHART:

Bar charts are helpful in comparing between categories to easily see which performs best.

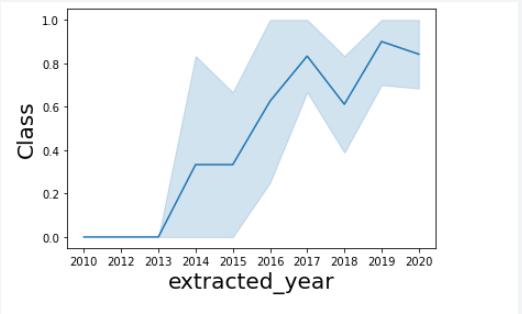
- Success rate of each orbit type.



LINE GRAPH:

A line graph is a useful tool in showing trends in a set of data..

- Success rates per year



https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/78df355b171547f4a2aeef77bba5d88dd9cc3182/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

Following SQL queries were use for our analysis:

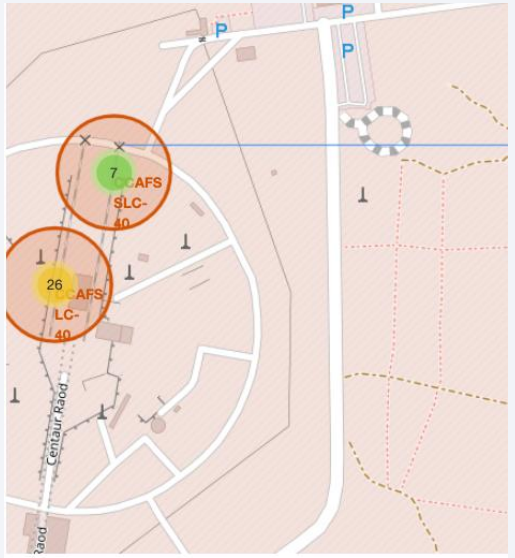
- **Displaying the names of the unique launch sites in the space mission**
- **Displaying 5 records where launch sites begin with the string 'KSC'**
- **Displaying the total payload mass carried by boosters launched by NASA (CRS)**
- **Displaying average payload mass carried by booster version F9 v1.1**
- **Listing the date where the successful landing outcome in drone ship was achieved.**
- **Listing the names of the boosters which have success in ground pad 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 records which will display the month names, successful landing_outcomes in ground pad ,booster**
- **versions, launch_site for the months in year 2017**
- **Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.**

https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/cea8bf248f6761cbb6fb4133103e3fe9592f6995/Eda_sql.ipynb

Build an Interactive Map with Folium

- We used Folium in this project to find geographical patterns among launch sites. Here I used the following Folium map objects:

Map Object	Purpose
Circle Marker	Mark the location of interest
Map Marker	Shows a mark on the map
Icon Marker	Creates an icon
PolyLine	Shows a line between two points
Marker Cluster Object	Multiple markers on the map when zoomed out



Build a Dashboard with Plotly Dash

I used Plotly Dash to create an interactive dashboard containing the following:

- **Dropdown:** Used to choose different launch sites
- **Range Slider:** Used to choose a payload range (values between 0kg to 1000kg)
- **Pie Chart:** It shows the distribution of success launches per site and it shows the distribution of successful vs. failed launches.
- **Scatter Plot :** Payload range using slider and the booster version categories.

https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/cb28cdaba727f86c7a7ee8642b304b4178bf1358/dash.py

Predictive Analysis (Classification)

The model was built using the Sklearn library from Python

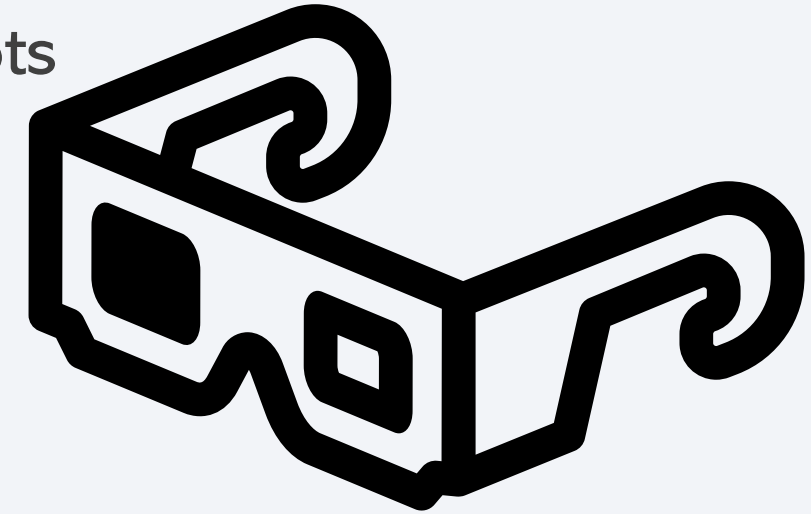
- We evaluated each model by splitting the dataset with 20% of test data and 80% of train data for a total of 18 test records.
- The best preforming model was found by comparing the result of each model over the given dataset
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Process:

https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project/blob/14c3efa5cc8b8b4f850da089c740f674aad68cc3/Hands%20on%20Lab:%20Complete%20the%20Machine%20Learning%20Predic.ipynb

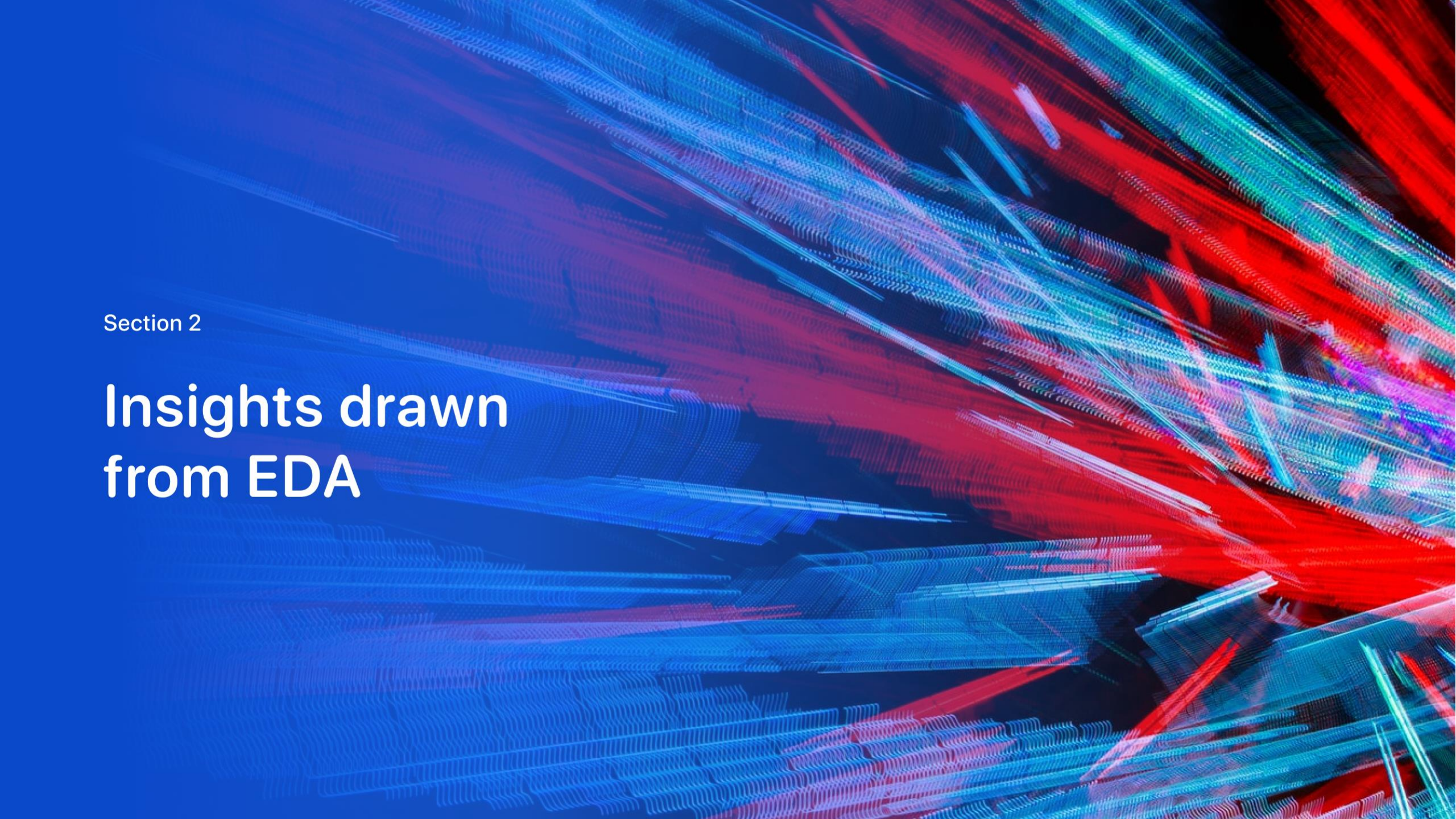
Results

1. Exploratory data analysis results
2. Interactive analytics demo in screenshots
3. Predictive analysis results

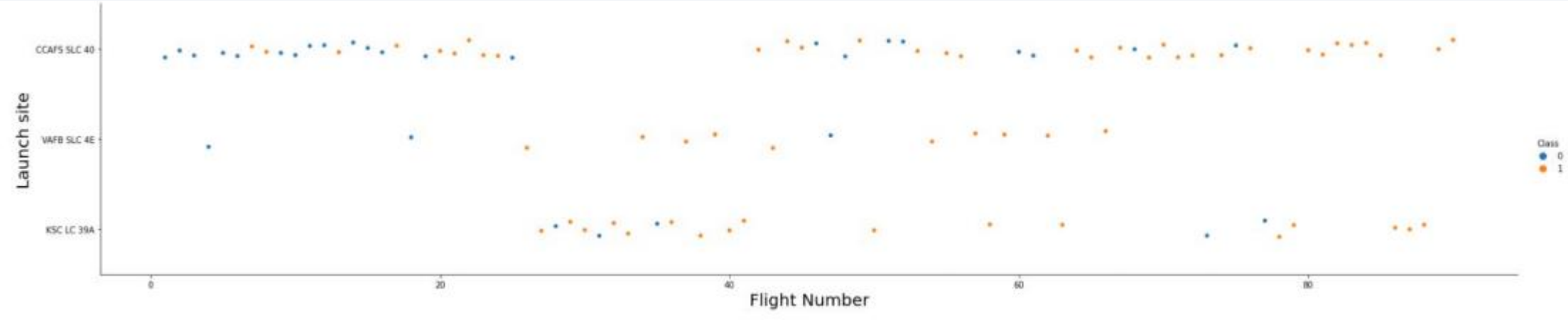


Section 2

Insights drawn from EDA

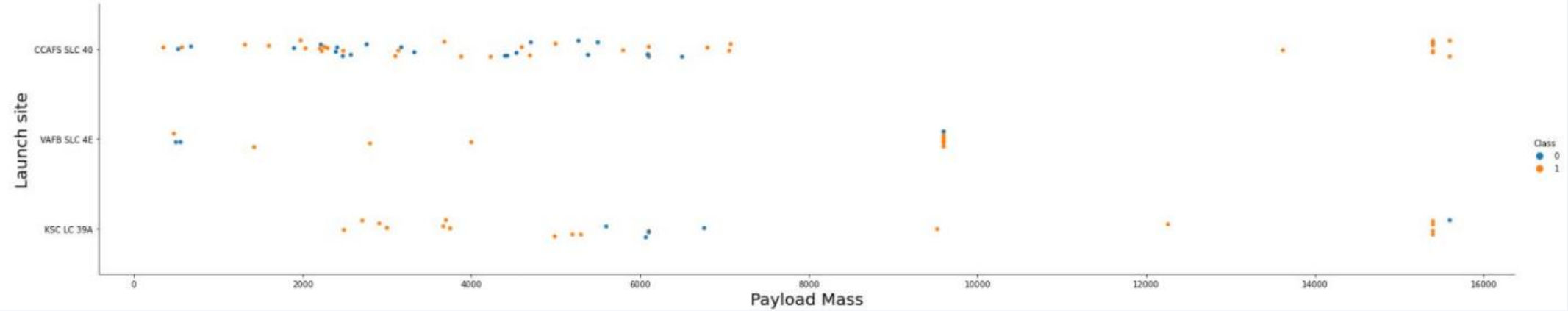


Flight Number vs. Launch Site



- The scatter plot has Flight Number in the x-axis and Launch Site in the y-axis.
- CCAFS SLC 40 has more successful landings

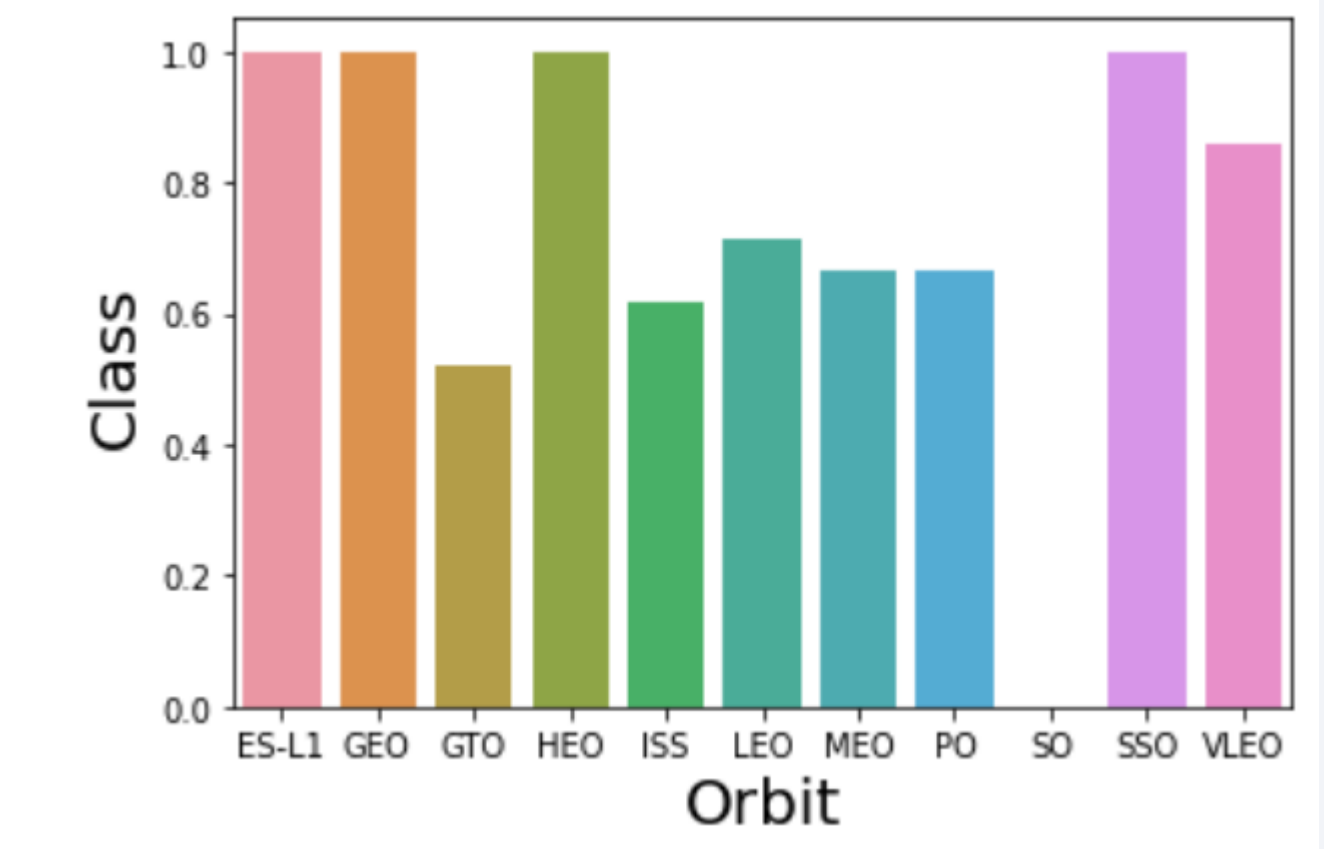
Payload vs. Launch Site



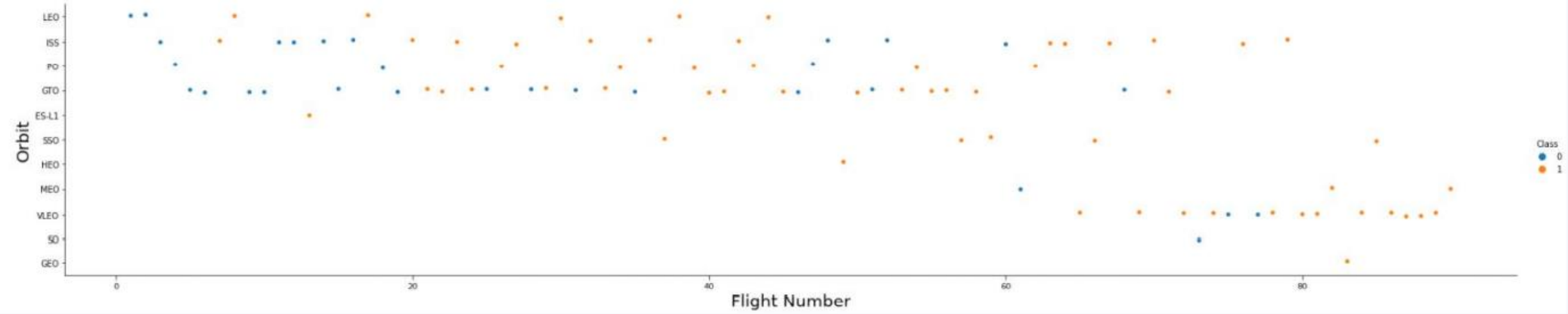
- The scatter plot has Payload Mass (kg) in the x-axis and Launch Site in the y-axis.
- CCAFS is performing better with higher mass

Success Rate vs. Orbit Type

The orbit types ES-L1, GEO, HEO, and SSO has the highest success rates(100%). Meanwhile, the orbit type SO has 0% success rate.

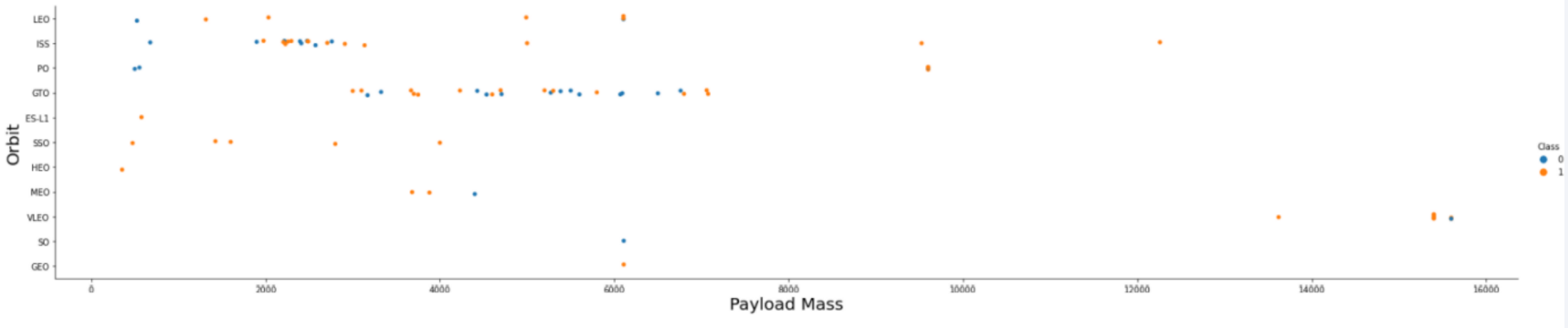


Flight Number vs. Orbit Type



- the success rate for LEO orbit increases as the number of flights increase.

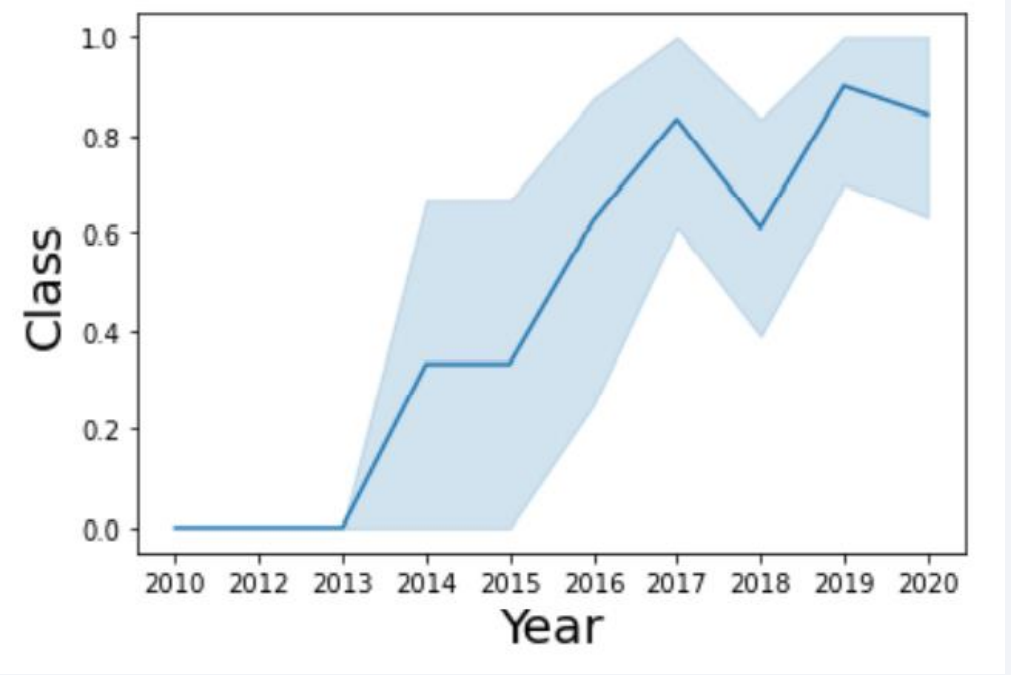
Payload vs. Orbit Type



- The orbits have a big impact on the success ratio of the launches at middle and lower masses

Launch Success Yearly Trend

- Performance improvement from the start



All Launch Site Names

- Using the DISTINCT() function will return only the unique elements.

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

Launch Site Names Begin with '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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Using the LIKE keyword accompanied by 'CCA%' to search for all launch site names beginning with 'CCA' and specifying the LIMIT will return a selected number of rows.

Total Payload Mass

SQL Query:

```
payload_mass_nasa_crs = %sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTBL where CUSTOMER='NASA (CRS)'
```

total_payload_mass
45596

- Using the SUM() function will return the sum of the column chosen and specifying a condition.

Average Payload Mass by F9 v1.1

SQL Query:

```
avarage_payload_mass_carried_F9_v1_1 = %sql select avg(PAYLOAD_MASS__KG_) as Avarage_Payload_Mass from SPACEXTBL where BOOSTER_VERSION =
'F9 v1.1'
```

avarage_payload_mass
2928.400000

- Using the AVG() function will return the average of the column chosen and specifying a condition.

First Successful Ground Landing Date

SQL Query:

```
successful_ground_pad_landing = %sql select min(DATE) as Date_first_successful_outcome_ground_pad from SPACEXTBL where LANDING__OUTCOME =
'Success (ground pad)'
```

date_first_successful_outcome_ground_pad
2015-12-22

- Using the MIN() function will return the minimum value of the column chosen and specifying a condition.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query:

```
successful_drone_landing_payload_mass = %sql select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and PA
YLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

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

- Using the AND keyword allows you to specify more than one argument or condition, and the BETWEEN-AND keywords for a specified range.

Total Number of Successful and Failure Mission Outcomes

SQL Query:

```
success_mission_outcome = %sql select count(MISSION_OUTCOME) as success_mission_outcome from SPACEXTBL where MISSION_OUTCOME like '%Succe
ss%'
success_mission_outcome
```

```
failure_mission_outcome = %sql select count(MISSION_OUTCOME) as failure_mission_outcome from SPACEXTBL where MISSION_OUTCOME like '%Failu
re%'
failure_mission_outcome
```

	successful_mission_outcome	failure_mission_outcome
0	(100)	(1)

- The COUNT() function returns the total count of the selected column.

Boosters Carried Maximum Payload

SQL Query:

```
load = %sql SELECT DISTINCT BOOSTER_VERSION , max(PAYLOAD_MASS__KG_) as max_payload FROM SPACEXTBL GROUP BY BOOSTER_VERSION ORDER BY max_
payload DESC;
load
```

booster_version	max_payload
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600
F9 B5 B1049.6	15440
F9 B5 B1059.3	15410
F9 B5 B1051.5	14932
F9 B5 B1049.3	13620
F9 B5B1058.1	12530

- Using the Distinct() function and group by.

M
MONTHNAME

2015 Launch Records

SQL Query:

```
succ = %sql SELECT MONTHNAME(DATE,'%m') as monthname, BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME FROM SPACEXTBL WHERE (LANDING__OUTCO
ME LIKE '%Failure (drone ship)%') AND EXTRACT(YEAR FROM DATE) = '2015'
succ
```

monthname	booster_version	launch_site	landing__outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Using the CONVERT(), MONTHNAME() and EXTRACT function.

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

SQL Query:

```
rank_count = %sql select COUNT(LANDING__OUTCOME) as successful_landing_outcome from SPACEXTBL where (LANDING__OUTCOME LIKE '%Success%') A
ND (DATE > '2010-06-04') AND (DATE < '2017-03-20')
```

successful_landing_outcome
8

- The COUNT() function returns the total count of the selected column, in the WHERE using LIKE condition.

Section 4

Launch Sites Proximities Analysis

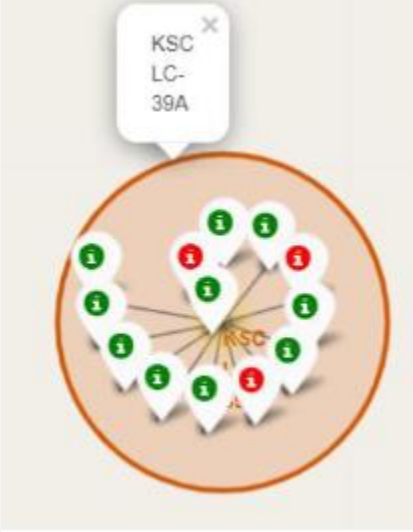
Spacex launch sites.

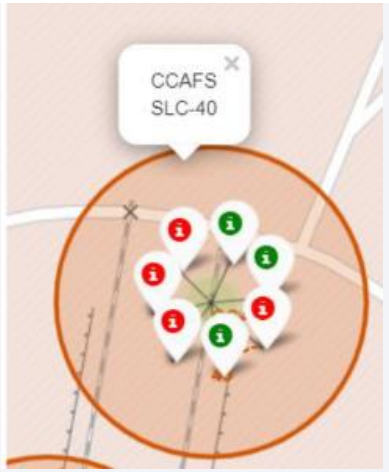
- The launch sites locations are concentrated in the USA.

Labelled Markers

- Green Markers: Successful lunches
- Red Markers: no succe lunches

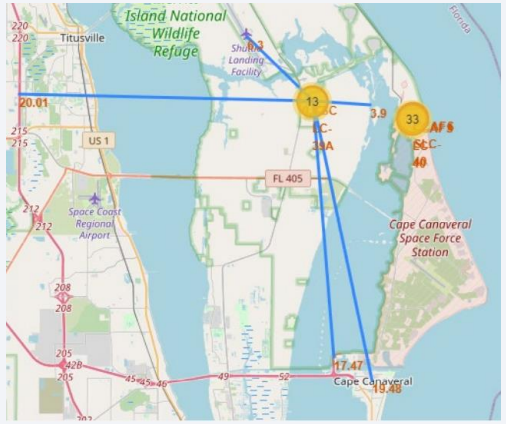
We can see that the launch site KSC LC-39A has the better results and CCAF – SLC 40 the worst ones

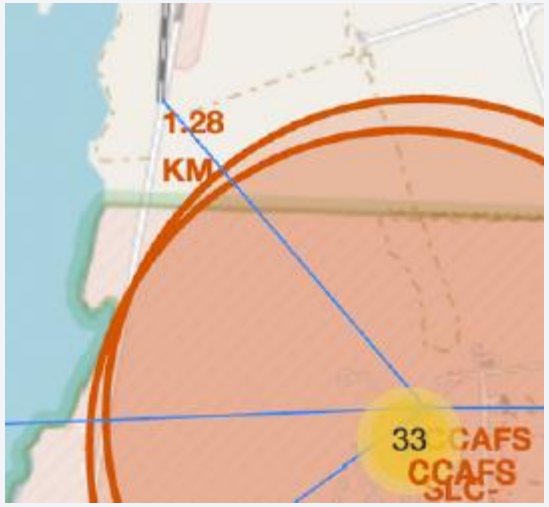




Distances

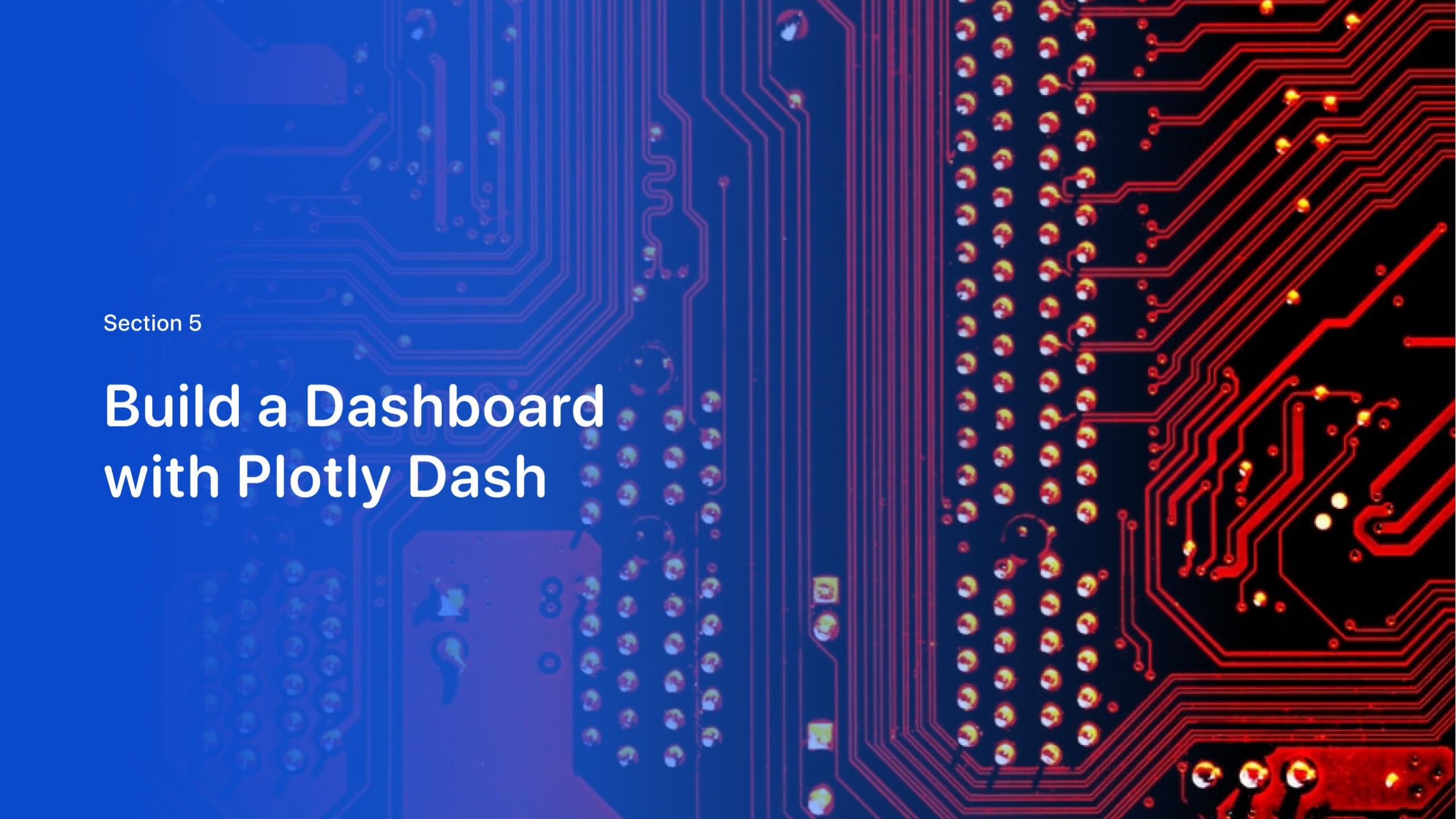
- Launch sites are easy to access, and all the transport infrastructures are near by with a train station at 700m





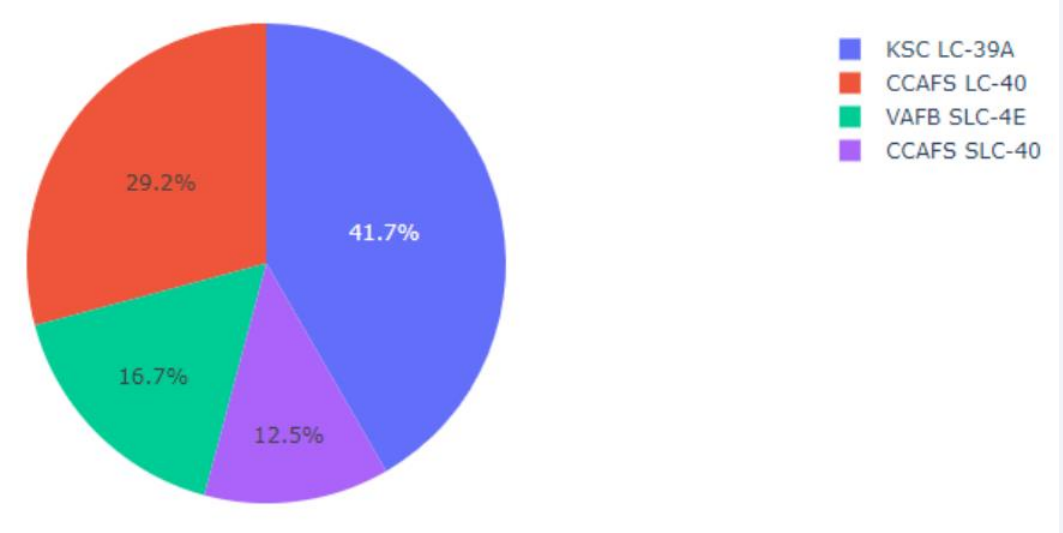
Section 5

Build a Dashboard with Plotly Dash



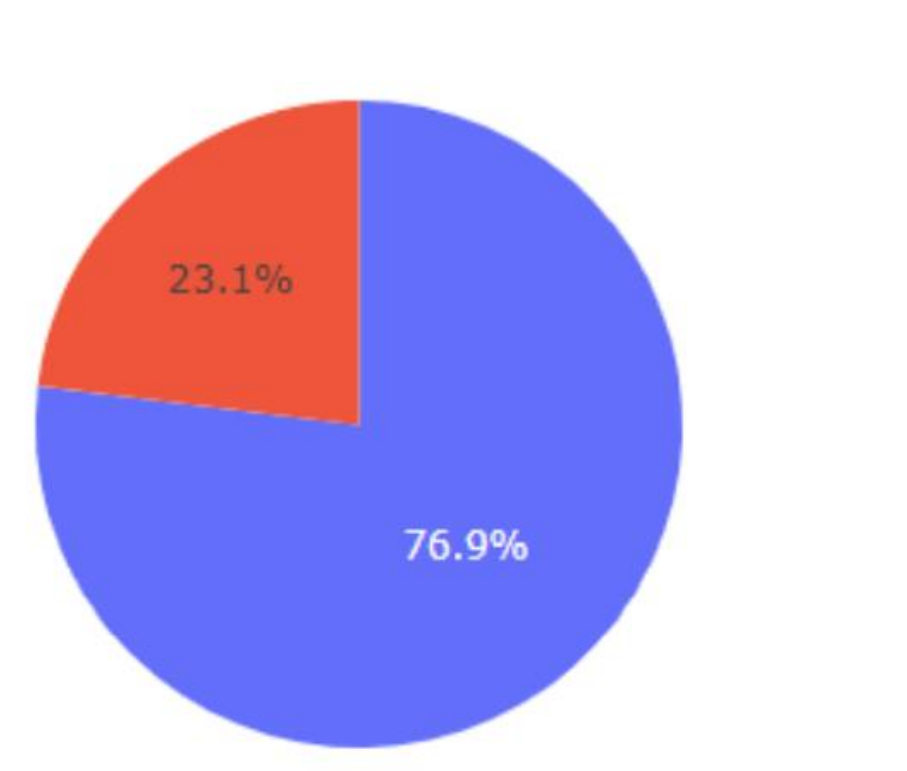
Launch success

- KSC LC-39A as having the most successful launch count.

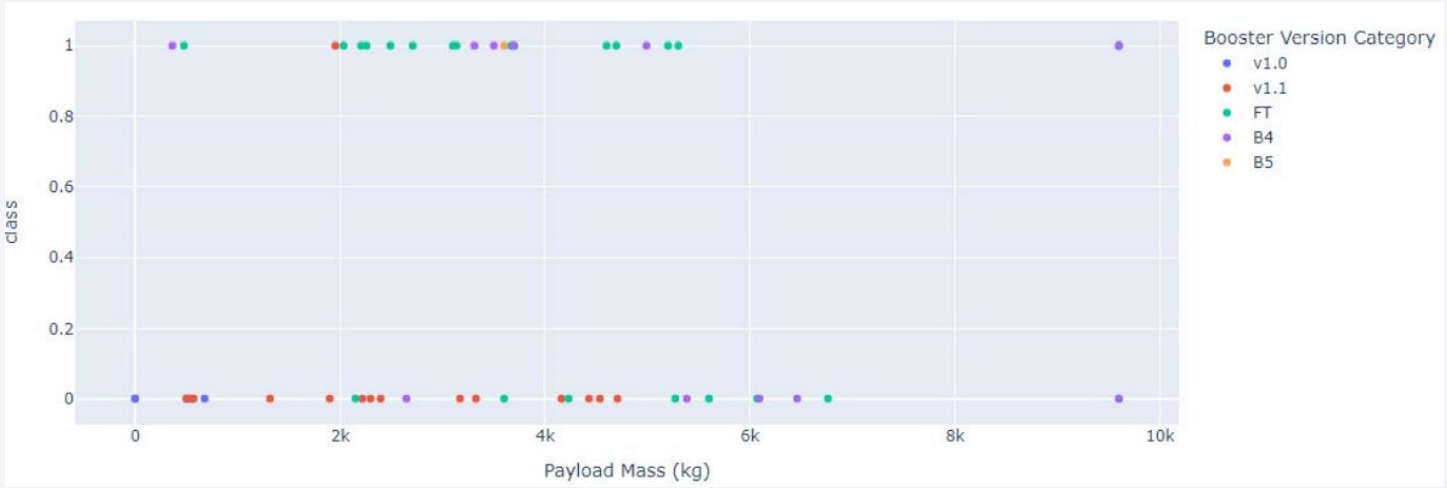


Lauch site with highest launch success ratio

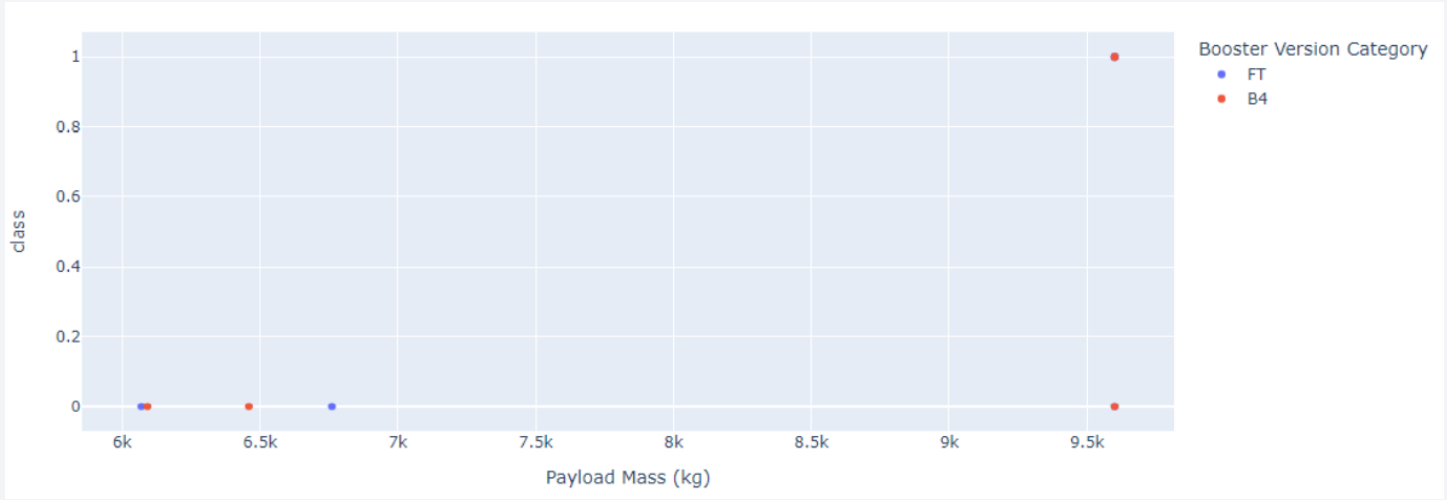
- KSC LC-39A has a ratio higher than 77% of success launch.



Payload vs. Launch



payloads from 0 to 2500kg,



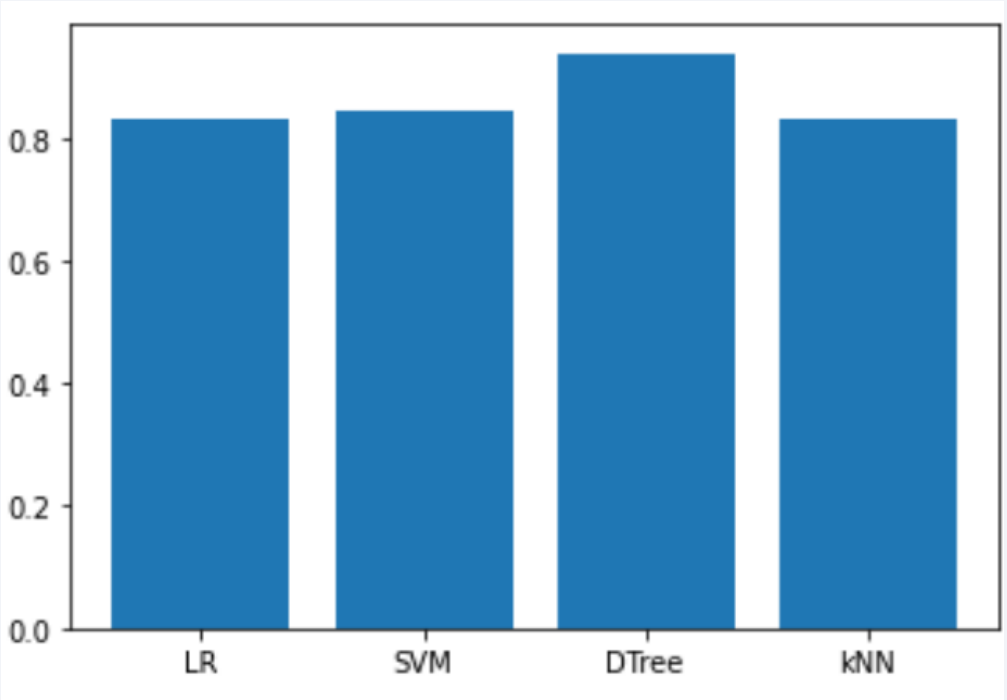
payloads above 6000kg

Section 6

Predictive Analysis (Classification)

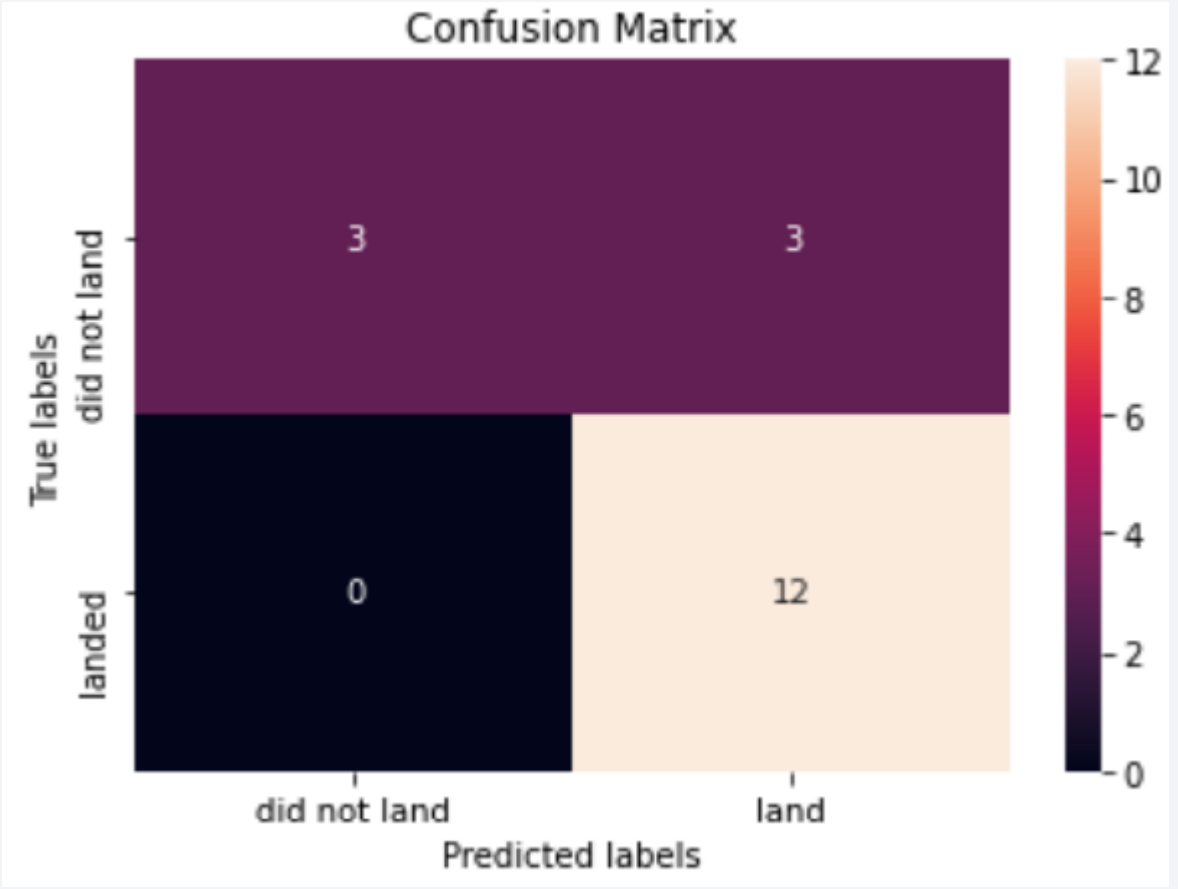
Classification Accuracy

- The accuracy is extremely close
- The tree algorithm wins
- I achieved 0.83 accuracy on the test data.



Confusion Matrix

- Examining the confusion matrix, we see that Tree can distinguish between the different classes.
- I see that the major problem is false positives.



Conclusions

- Higher payload mass perform better
- Experience increase launches success rates
- Most efficient Orbits are ES-L1, GEO, HEO, SSO and VLEO
- Best launch site is KSC LC-39A
- Launch sites are near transportations means and away from citizens
- Decision Tree Classifier has the best accuracy

Appendix

- The complete project can be found on this GitHub repository :
- https://github.com/erwinideb/Data-Science-and-Machine-Learning-Capstone-Project

Thank you!