Winning Space Race with Data Science

Faizan Ul Haq
Feb. 20th, 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- **Summary of Methodologies**
  - Data Collection
  - Data Wrangling
  - EDA with SQL
  - EDA with Visualization
  - Building an Interactive Map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- **Summary of all results**
  - Exploratory Data Analysis Results
  - Interactive Analytics Demo
  - Predictive Analysis Results

Introduction

Project background and context

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- In this project, we would like to find out the price for each rocket launch for company SpaceY, competitor of SpaceX. Since the rocket cost highly depends on the successful rate of reuse of the first stage of racket, we would like to study parameter of the First stage of rocket on successful launch.

Problem Objective - Problems you want to find answers

- **As a Data Scientist for SpaceY, we would like to find out:**
  - How the parameters in the first stage affect on the successful landing rate of Falcon 9?
  - Can we find out the best parameters in the first stage to ensure the successful landing rate of our First Stage rocket so that we can be able to determine the best cost for the Rocket Launch?

Section 1

Methodology

Methodology

Data collection methodology:

- SpaceX REST API
- Web Scrapping from Falcon9 Wikipedia List of Falcon 9 and heavy launches

Perform data wrangling

- Performed One Hot Encoding to find some patterns in the data and determine what would be the label for training supervised models which will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed and 0 means it was unsuccessful.

Perform exploratory data analysis (EDA) using visualization and SQL

- Exploring using SQL queries and preparing data using Scatter and Bar graphs to show some patterns of data.

Perform interactive visual analytics using Folium and Plotly Dash

- To find some geographical patterns and prepare Dashboards about launch sites by using Folium and Plotly Dash.

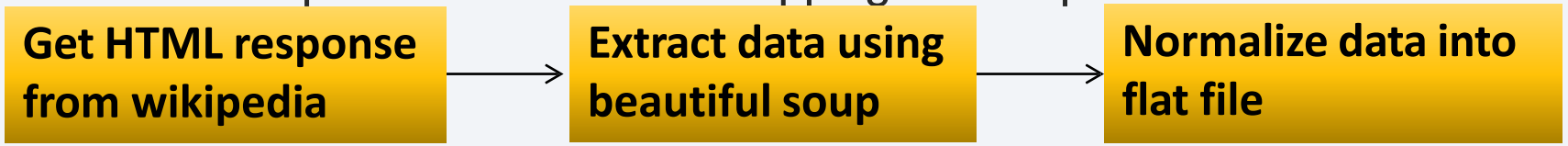
Perform predictive analysis using classification models

- How to build, tune, evaluate classification models

Data Collection

- **The following datasets were collected by two ways:**
  - **By Using SpaceX API**
    - ❖ The API provides us the data about launches, rocket used, payload delivered, launch and landing specifications and landing outcomes etc.

  - **By Using Web Scrapping**
    - ❖ BeautifulSoup is used for Web Scrapping on Wikipedia.



Our goal is to use this data to predict whether SpaceX rocket will be successful to land the rocket or not.

Data Collection – SpaceX API

simplified flow chart

1 .Getting Response from API

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

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

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

4. Assign list to dictionary then 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}
```

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

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

Github URL to Notebook

Data Collection – Web Scraping

[Github URL to Notebook]

1. Getting Response from HTML

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

2. Creating BeautifulSoup Object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
# create an instance of the BeautifulSoup class to parse our document
soup = BeautifulSoup(response.content, 'html.parser')
```

3. Finding Tables

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
html_tables
```

4. Getting Column Names

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name)>0):
        column_names.append(name)
```

5. Creating 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. Appending Data to Keys (please refer to notebook block 15)

```
In [15]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(soup.find_all('table',"wikita
            # get table row
             for rows in table.find_all("tr"):
                 #check to see if first table heading is as number corresp
                 if rows.th:
                     if rows.th.string:
                         flight_number=rows.th.string.strip()
```

7. Converting Dictionary to DataFrame

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

8. DataFrame to .csv file

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

Data Wrangling

Introduction

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 the mission outcome was successfully landed to a specific region of the ocean. 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.

We mainly converted those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Process

Perform Exploratory Data Analysis EDA on Dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

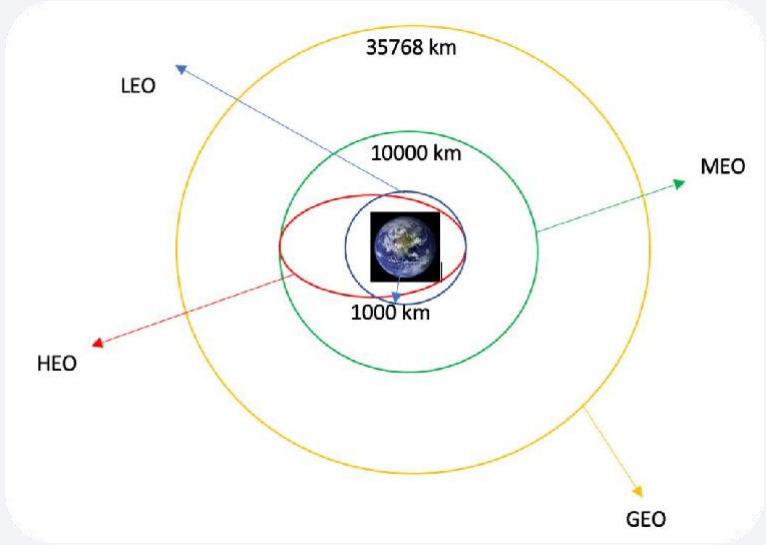
Export dataset as .CSV file

Create a landing outcome label from Outcome Column.

Create a landing outcome label from Outcome Column.

Github URL to Notebook

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



EDA With Data Visualization

Scatter Graphs:

Scatter Plot show how much one variable is affected by another. Using Scatter plot, we can check their correlation between 2 variables.

1. Flight Number Vs Payload Mass
2. Flight Number Vs Launch Site
3. Payload Vs Launch Site
4. Orbit Vs Flight Number
5. Payload Vs Orbit Type
6. Orbit Vs Payload Mass

Bar Charts:

Bar Chart shows big changes in data over time. It makes easy to compare dataset between multiple group at a glance.

1. Orbit Type Vs Success Rate

Line Charts:

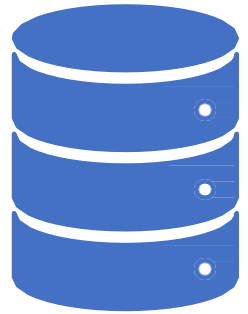
Line Graphs are useful as they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.

1. Success Rate Vs Year

Github URL to Notebook

EDA with SQL

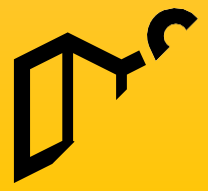
Performed SQL Queries to gather information about the dataset.



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

Github URL to Notebook

Build an Interactive Map with Folium




Objects created and added to a Folium Map:

- Markers that show all launch sites on map.
- Markers that show the success/failed launches for each site on the map.
- Lines that show the distances between a launch site to its proximities.




By adding these objects, following geographical patterns about launch sites are found:

- Are launch sites in close proximity to railways? Yes
- Are launches 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

Build a Dashboard with Plotly Dash

The Dashboard is built with Plotly Dash.
The dashboard application contains a pie chart and a scatter plots.

Graphs used:

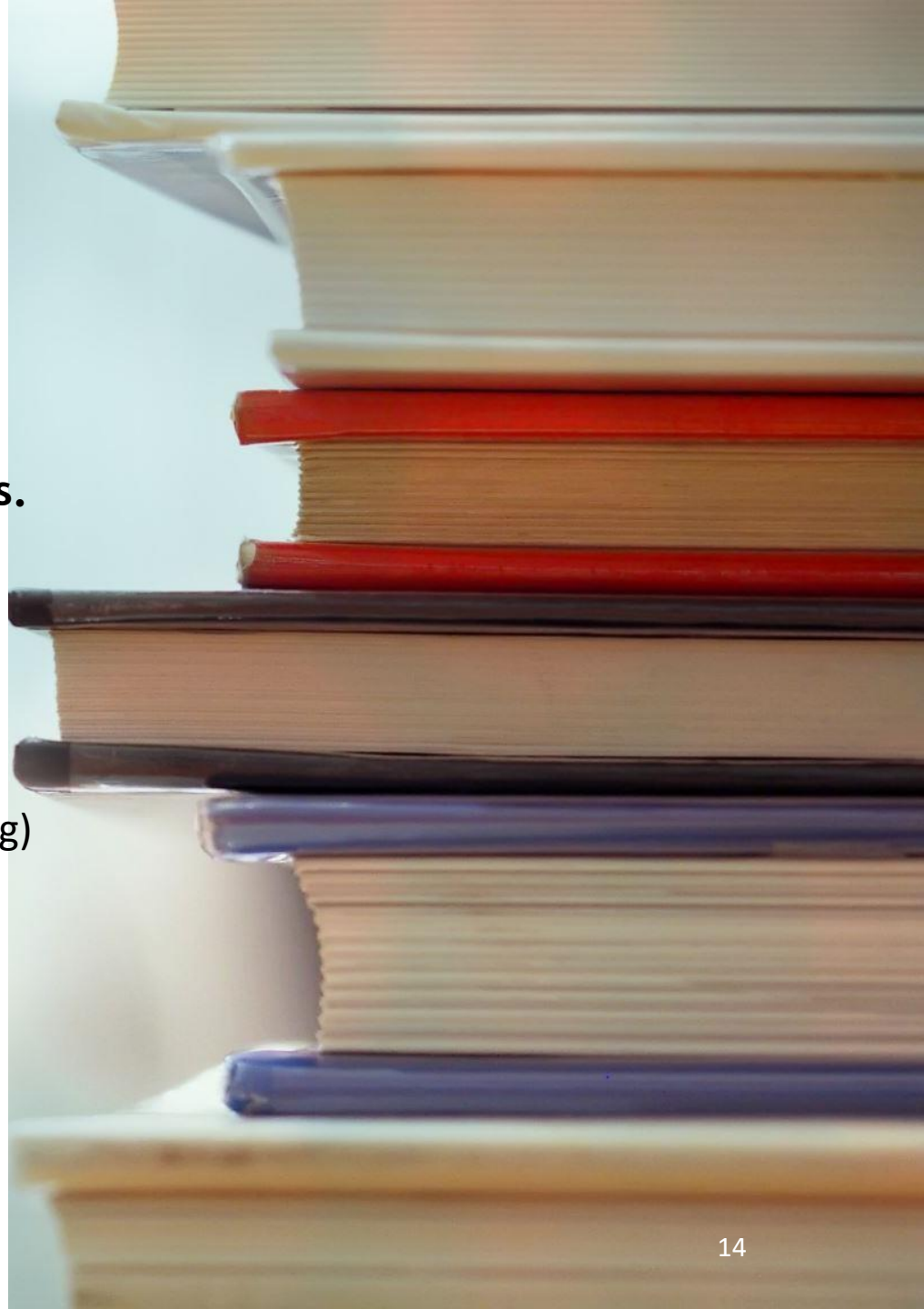
Pie Chart:

- Pie Chart shows the total launches by a certain site/all sites.

Scatter Graph:

- Scatter Graph shows the relationship with Outcome and Payload Mass (Kg) for the different Booster Version.

[Github URL to Source Code]



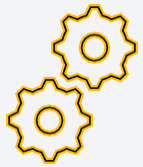
Predictive Analysis (Classification)

Building Model

- Load our data set into Numpy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples
- Decide on which type of machine learning algorithms to apply

Evaluating Model

- Check accuracy for each model
- Get tuned hyperparameters foreach type of algorithms



Improvable Model

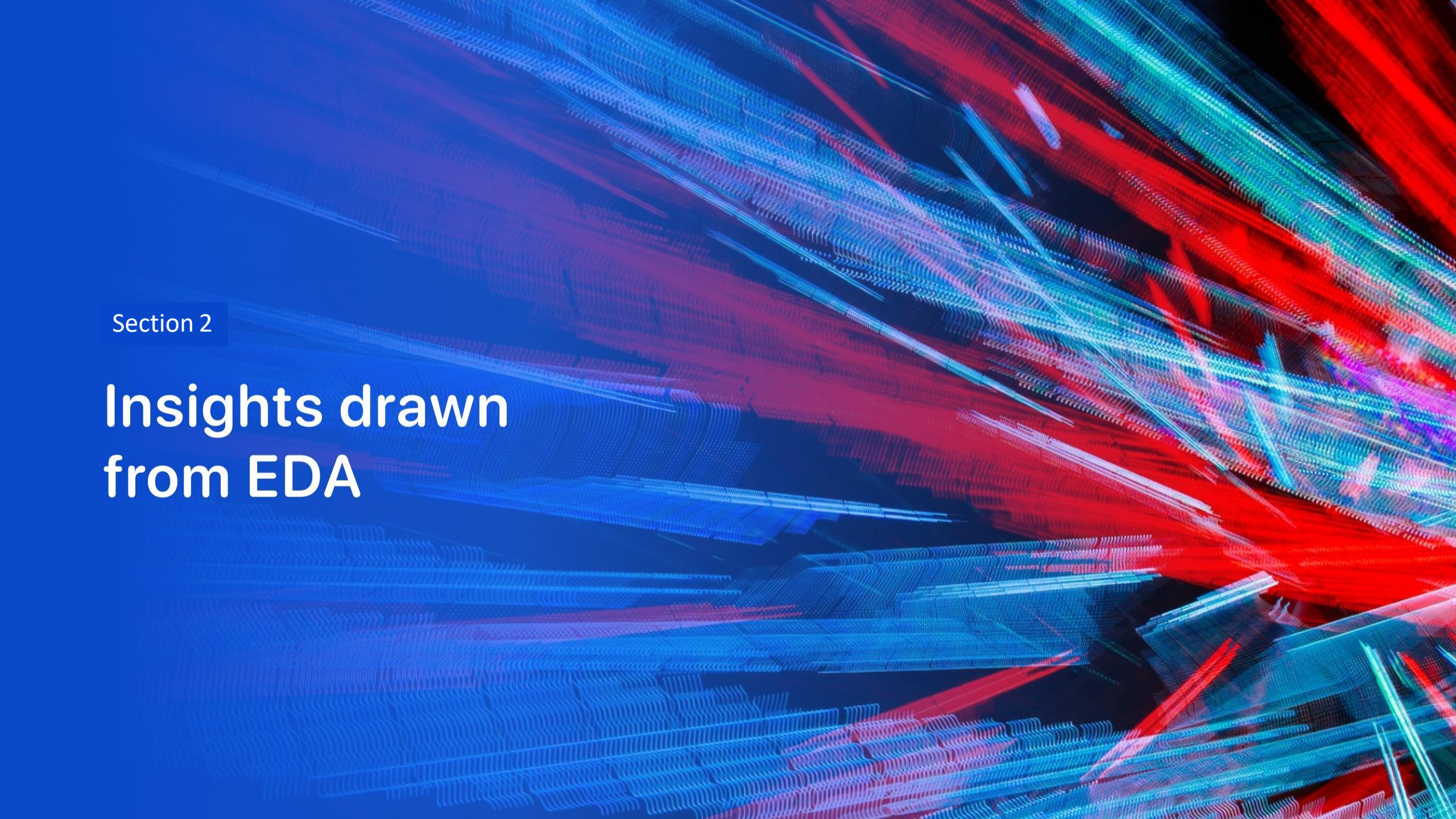
- Feature Engineering
- Algorithm Tuning

Finding The best Performing Classification Model

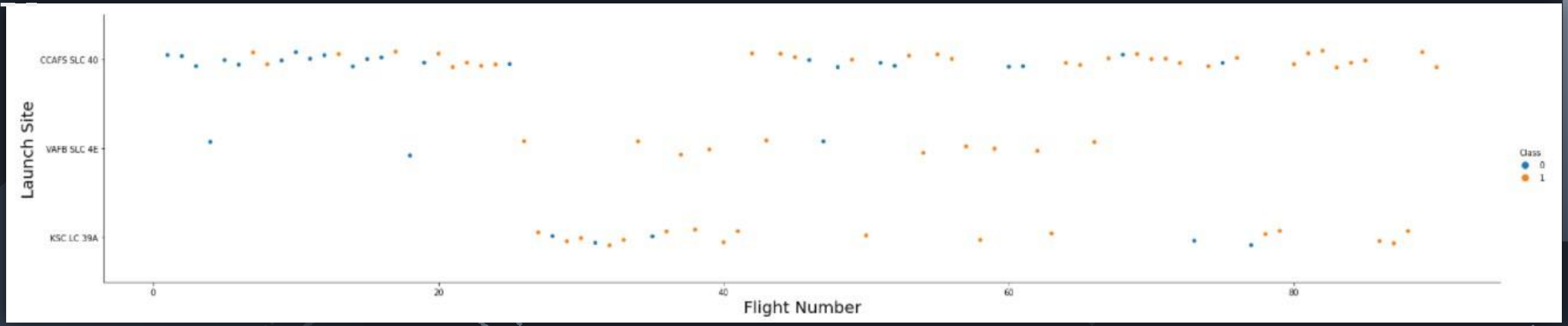
The model with the best accuracy score wins the best performing model

Results

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

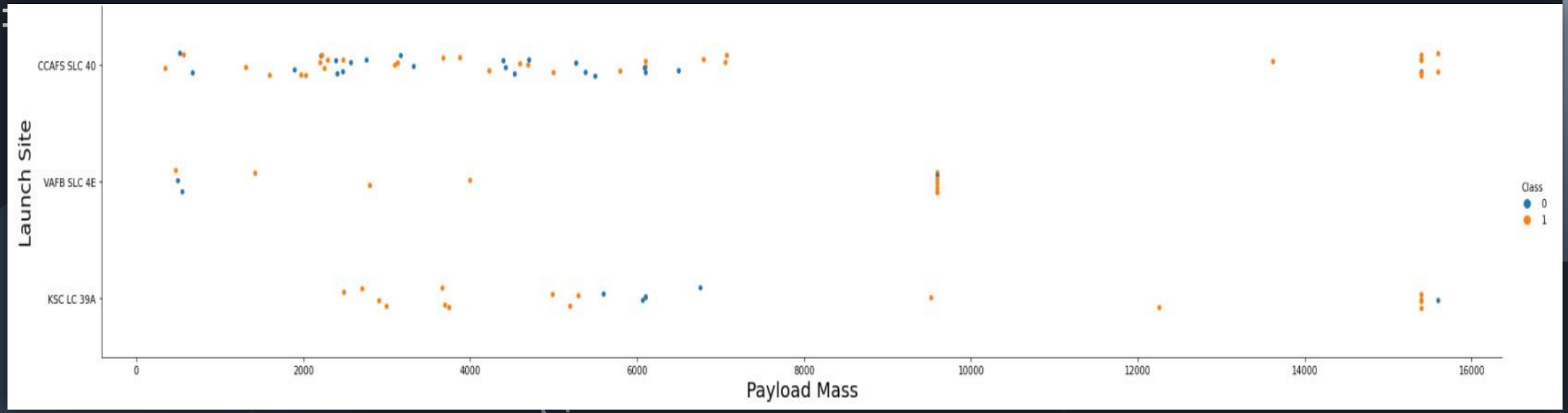

Section 2
Insights drawn from EDA

EDA with Visualization



Flight Number Vs Launch Site

- Class 0 (Blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- This figure shows that the success rate increased as the number of flights increased.
- As the success rate has increased considerably since the 20th flights. This point seems to be a big breakthrough.

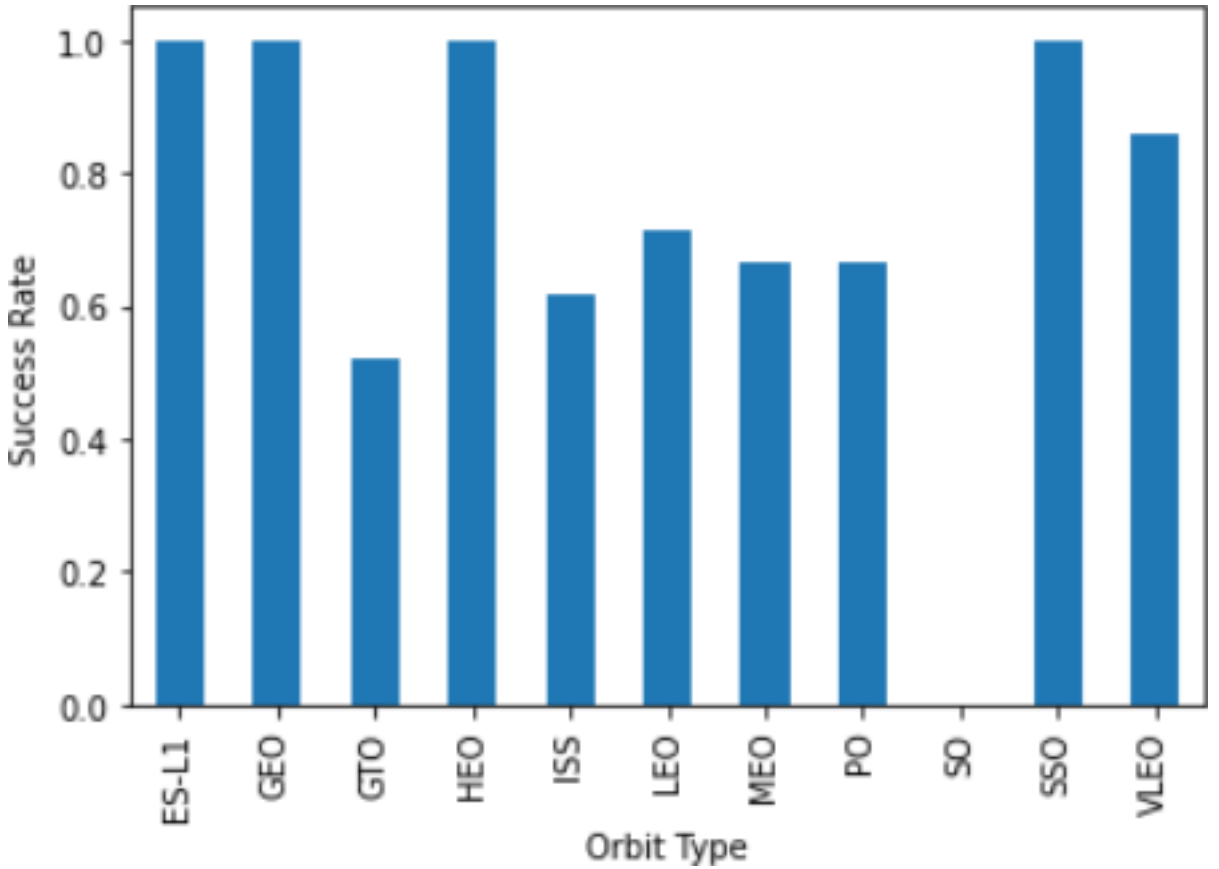


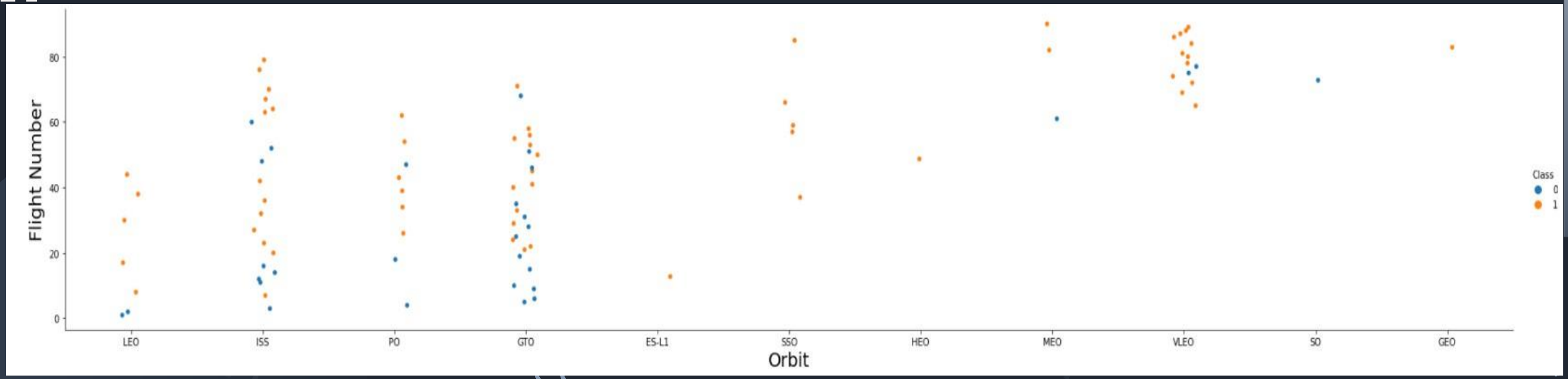
Payload Vs Launch Site

- The greater the payload mass for Launch Site **CCAFS SLC 40**, the higher the success rate for the Rocket.
- But it seems difficult to make decisions based on this figure because no clear pattern can be found between successful launch Pay Load Mass.

Success Rate vs. Orbit Type

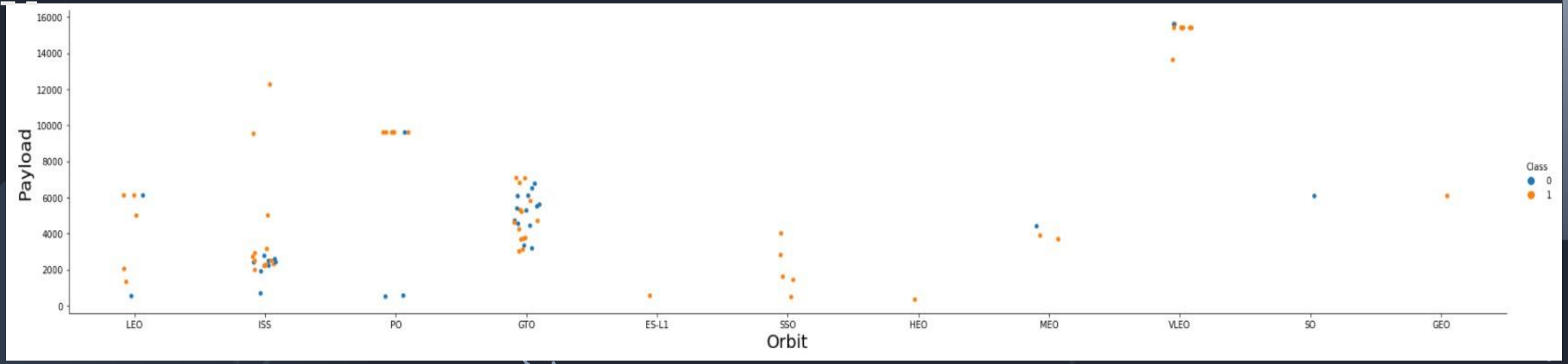
- Orbit types **SSO, HEO, GEO, and ES-L1** have the **highest success rates (100%)**
- On the other hand, the success rate of orbit type **GTO** is only **50%**, and it is the **lowest** except for type **SO**, which **recorded failure in a single attempt**.





Flight Number vs. Orbit Type

- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in GTO orbit there seems to be no relationship between flight numbers and success rate.
- SpaceX starts with LEO with a moderate success rate, and it seems that VLEO, which has a high success rate, is used the most in recent launches

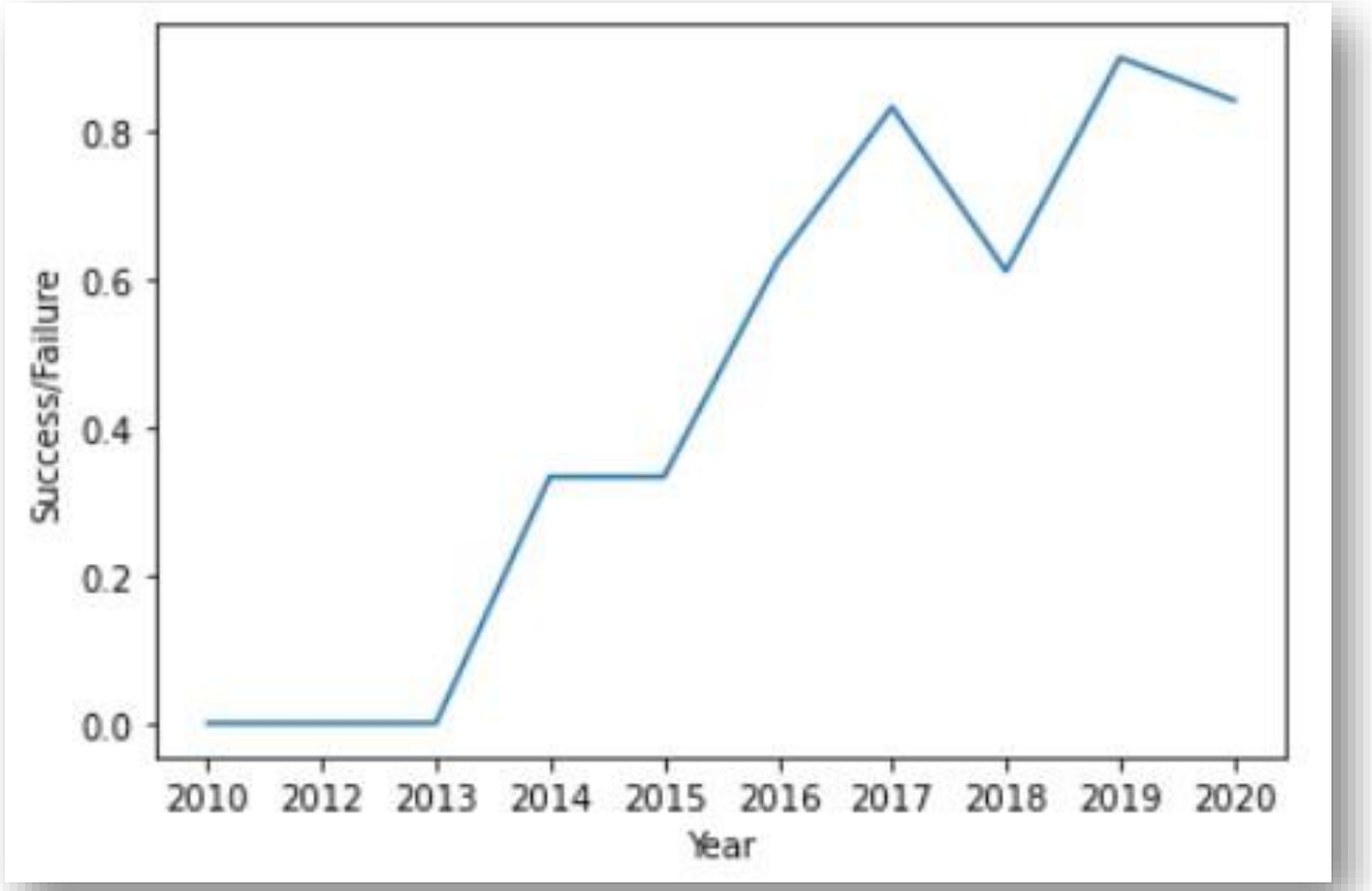


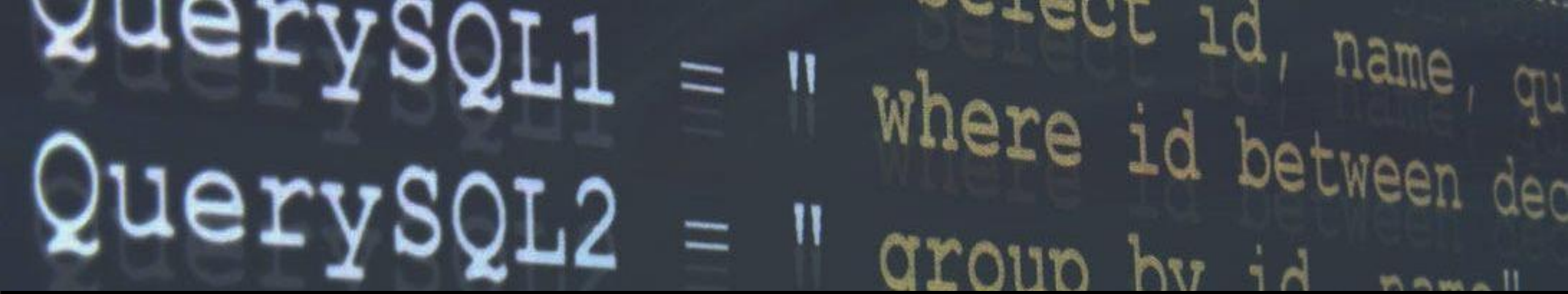
Payload vs. Orbit Type

- 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

- The success rate since 2013 kept increasing till 2017.
- The rate decreased slightly in 2018 to 60%.
- Recently, it has shown a success rate of about 80%.



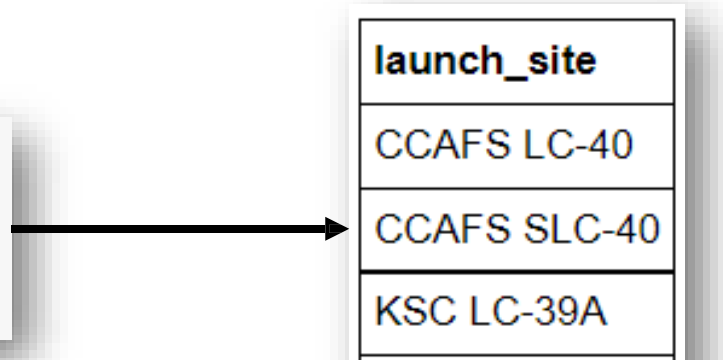


EDA with SQL

All Launch Site Names

SQL Query:

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXDATASET;
```



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

Query Explanation:

- Using the word DISTINCT in the query means that it will only show Unique values in the Launch_Site column from SpaceXDATASET Table

Launch Site Names Begin with 'CCA'

SQL Query:

```
%%sql
SELECT * FROM SPACEXDATASET
        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	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

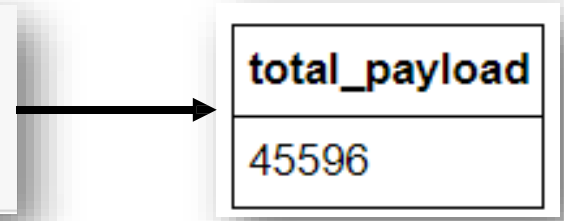
Query Explanation:

- 5 records of the SpaceX table were displayed using LIMIT 5 clause in the query
- Using the LIKE operator and the percent sign (%) together, the LAUNCH_SITE name starting with CAA will be called.

Total Payload Mass

SQL Query:

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD
        FROM SPACEXDATASET
        WHERE CUSTOMER = 'NASA (CRS)';
```



total_payload
45596

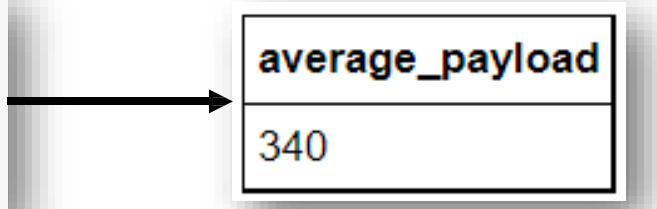
Query Explanation:

- Using the SUM() function to calculate the sum of column PAYLOAD_MASS_KG
- The WHERE Clause filter the dataset to only perform calculations on CUSTOMER, WHERE CUSTOMER = ‘NASA (CRS)’.

Average Payload Mass by F9 v1.1

SQL Query:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload
    FROM SPACEXDATASET
    WHERE Booster_Version LIKE 'F9 v1.0%';
```



average_payload
340

Query Explanation:

- Using the AVG() function to calculate the average value on column PAYLOAD_MASS_KG and store the value in average_payload.
- The WHERE clause filters the dataset to only perform calculation on Booster_version = F9 v1.1

First Successful Ground Landing Date

SQL Query:

```
%%sql
SELECT MIN(Date) AS First_Successful_Landing
    FROM SPACEXDATASET
    WHERE Landing__Outcome = 'Success (ground pad)';
```

first_successful_landing
2015-12-22

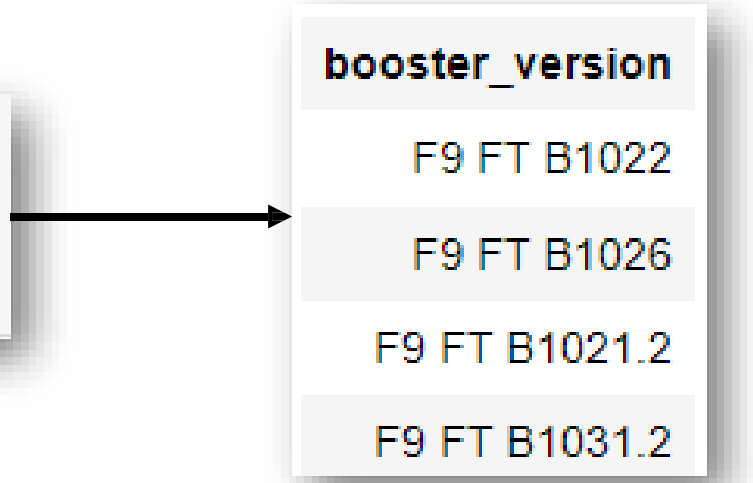
Query Explanation:

- Using the MIN() function to calculate the minimum date value on column DATE and store the value in First_Successful_Landing.
- The WHERE clause filters the dataset to only perform calculation on Landing__Outcome = ‘Success (ground pad)’.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query:

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXDATASET
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND (Payload_Mass__Kg_ BETWEEN 4000 AND 6000);
```



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

Query Explanation:

- Selecting only Booster_Version Column
- In the WHERE Clause filters the Landing__Outcome column to Success (drone ship)
- The AND & BETWEEN clause specifies additional filter condition PAYLOAD_MASS_KG BETWEEN 4000 AND 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query:

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXDATASET
GROUP BY MISSION_OUTCOME;
```

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

Query Explanation:

- Using the COUNT() function to filter the total number of columns
- Using GROUP BY function to group rows that have same values into summary rows to find the total number in each MISSION_OUTCOME.
- SpaceX successfully completed nearly 99% of its mission based on the dataset

Boosters Carried Maximum Payload

SQL Query:

```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXDATASET);
```

booster_version
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

Query Explanation:

- Using the COUNT() function to filter the total number of columns
- Using GROUP BY function to group rows that have same values into summary rows to find the total number in each MISSION_OUTCOME.
- SpaceX successfully completed nearly 99% of its mission based on the dataset

2015 Launch Records

SQL Query:

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXDATASET
WHERE Landing__Outcome = 'Failure (drone ship)'
    AND YEAR(DATE) = 2015;
```

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

Query Explanation:

- In the WHERE clause, filter the dataset to perform a search if Landing_Outcome is Failure(drone ship).
- Use AND operator to display a record if YEAR is 2015
- There were two landing failures on drone ships in 2015

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

SQL Query:

```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL_NUMBER
FROM SPACEXDATASET
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

landing__outcome	total_number
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:

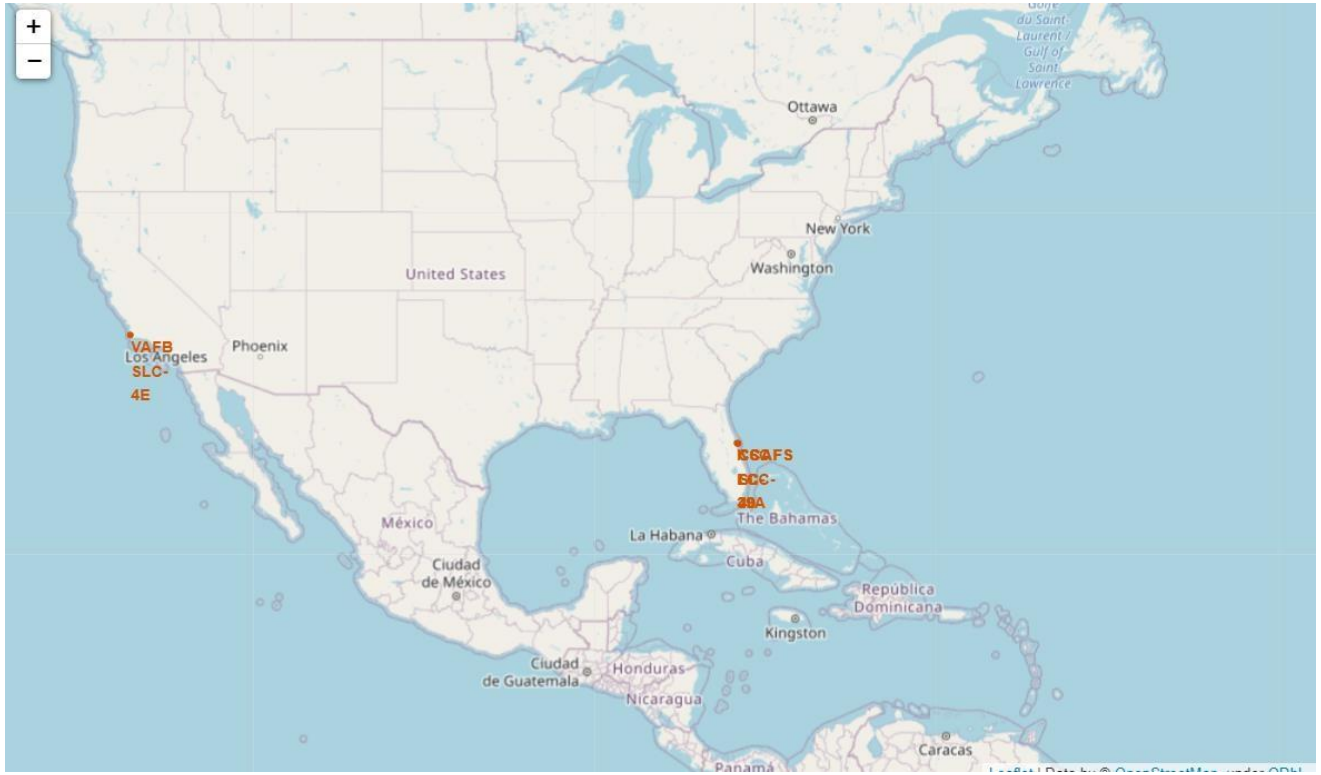
- In the WHERE clause, filter the dataset to perform search for DATE between 2010-06-04 and 2017-03- 20
- Using ORDER clause to sort the records by total number of landing and DESC clause to sort the records in descending order.

Section 3

Launch Sites Proximities Analysis

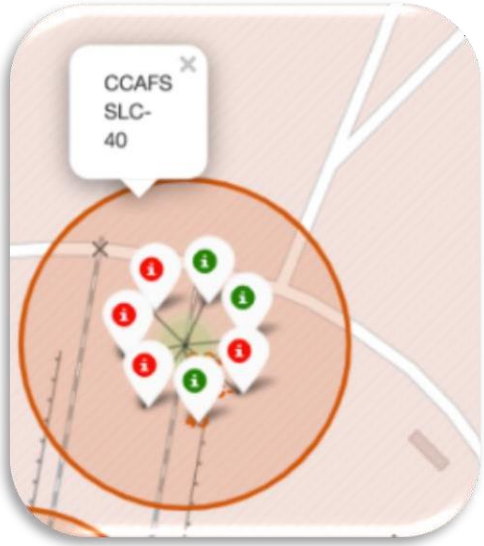
Interactive Map with Folium

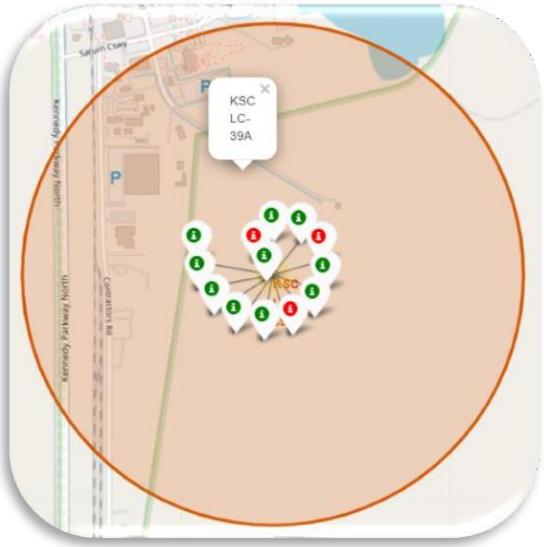
All Launch Site Locations

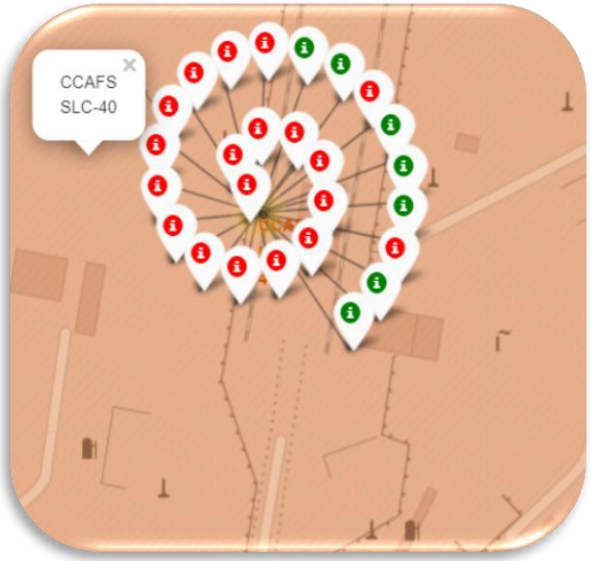


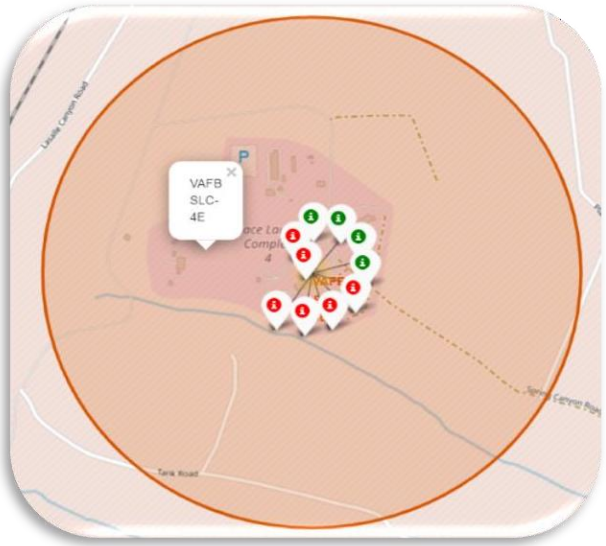
Most of SpaceX Launch Sites are in US Coast Area .i.e., **Florida and California**

Color Labelled Markers







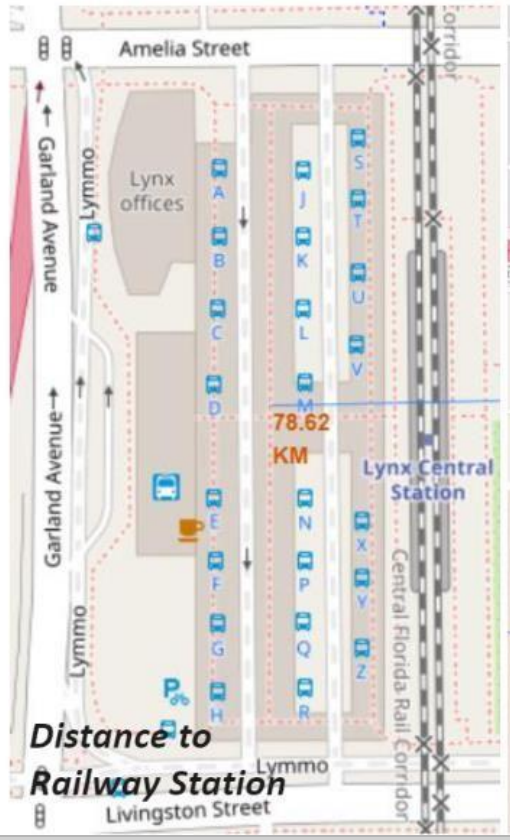


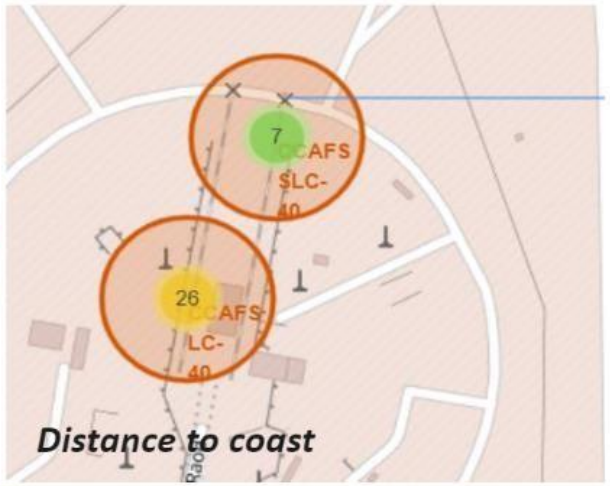
Florida Launch Sites

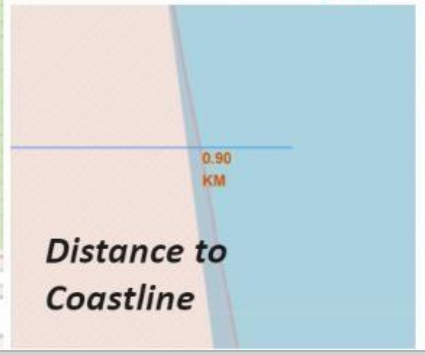
California Launch Sites

Note: Green Marker shows Successful Launches and Red Marker shows Failures.

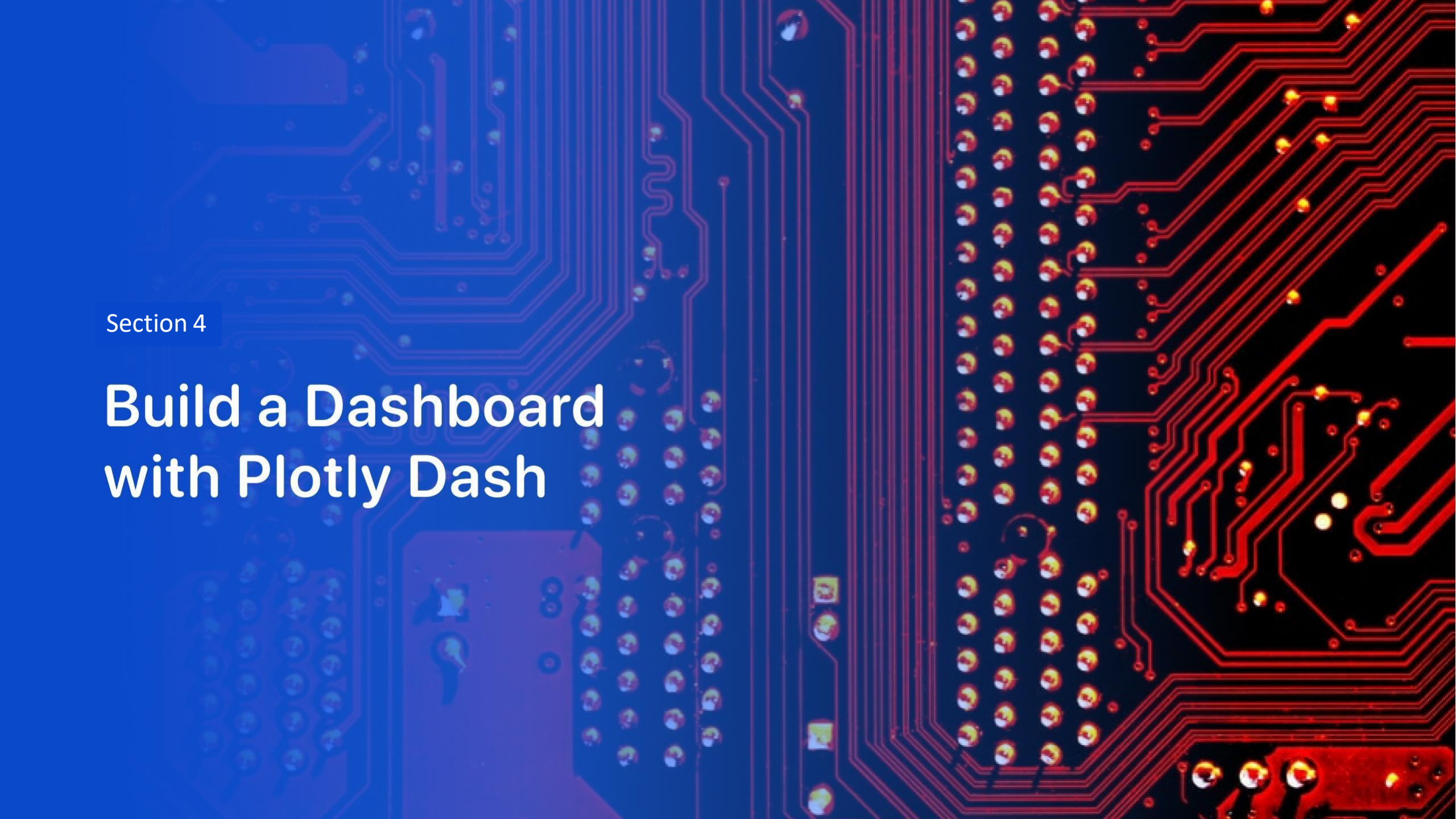
Distance of Launch Sites from Landmarks

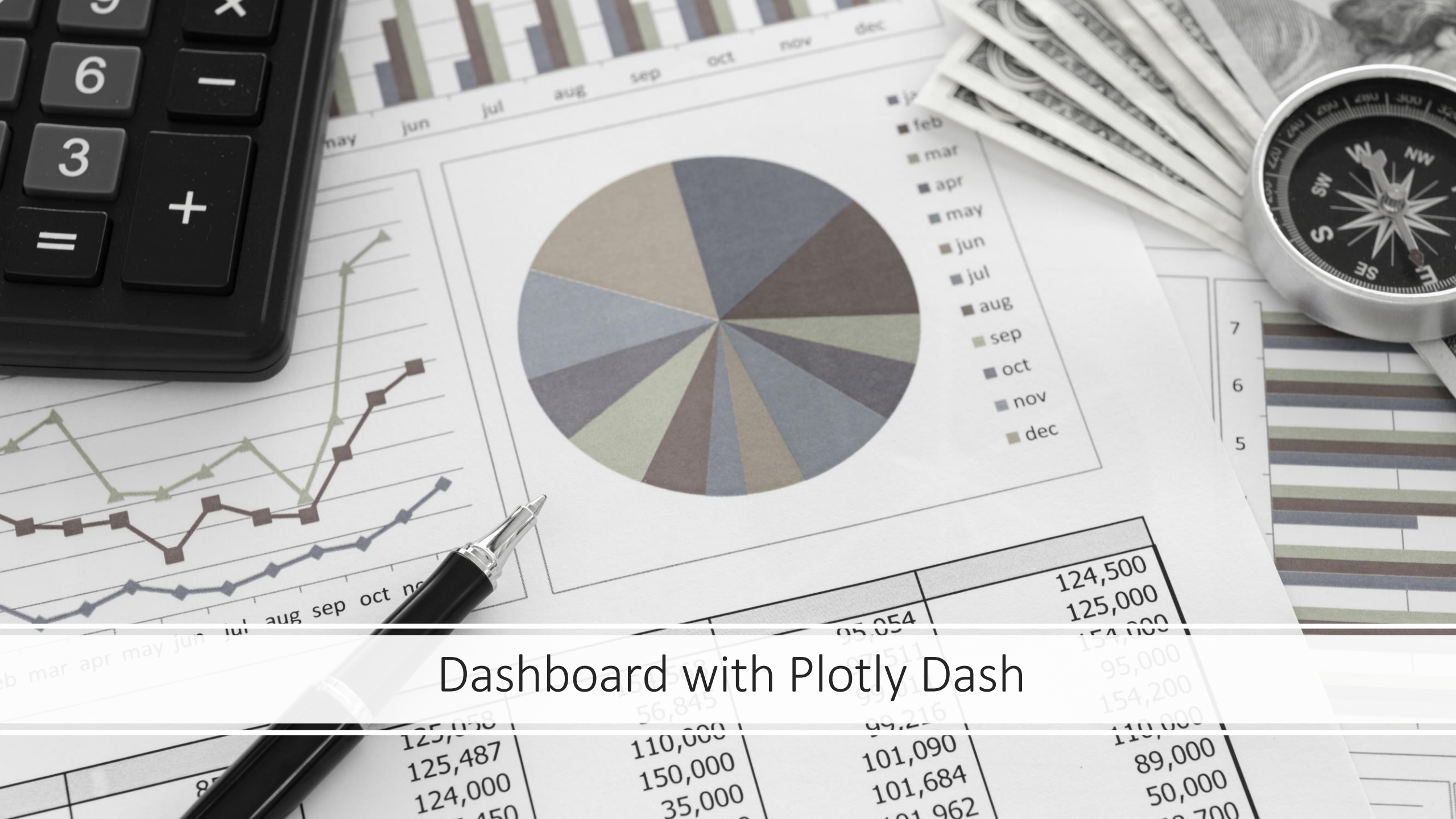






- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

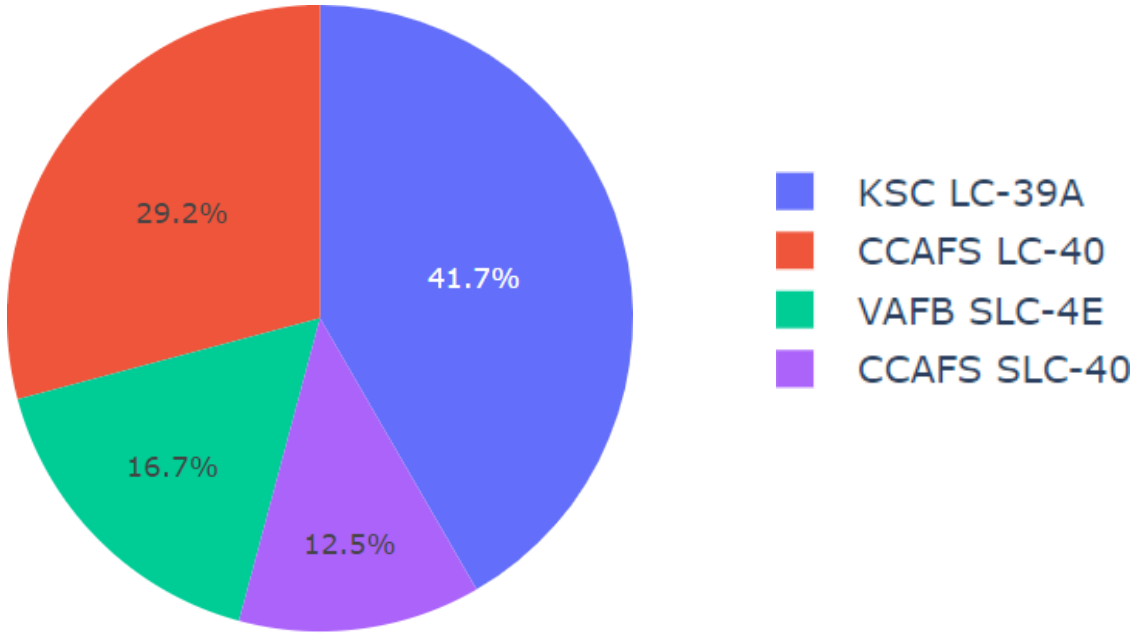


Dashboard with Plotly Dash

Dashboard (Pie Chart)

Total Success Launches by all sites

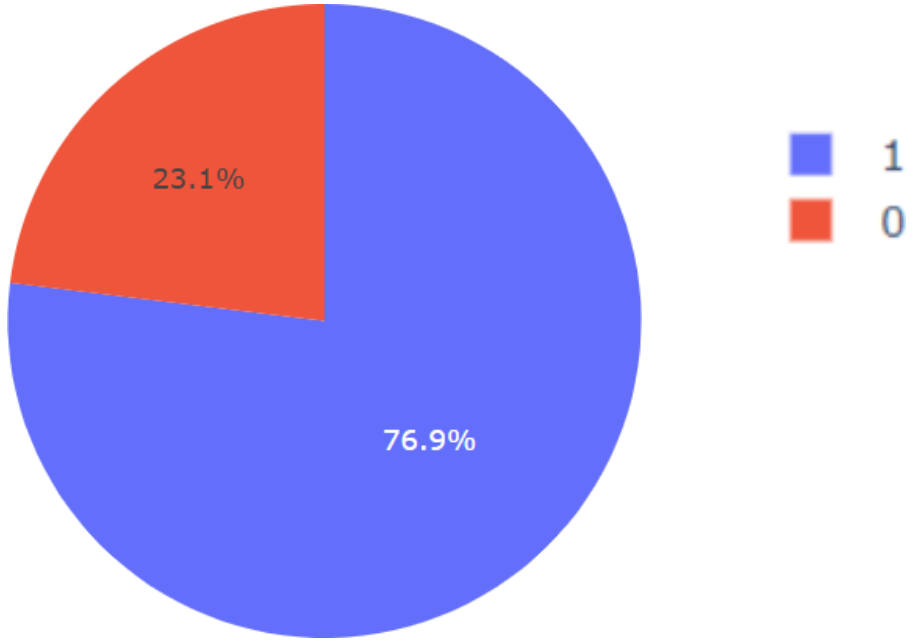
- Here, This Pie Chart shows the success percentage achieved by each launch site.
- KSC LC-39A records had the most launch successful launches among all sites.
- VAFB SLC-4E has the lowest success launch



Dashboard (Pie Chart)

Launch Site with Highest launch Success Ratio

- Here, This Pie Chart shows the launch site with highest launch success ratio.
- KSC-LC-39A achieved a 76.9% success rate with total of 13 landing while getting a failure rate of 23.1%.



Payload Vs Launch Outcome Scatter Plot for all sites

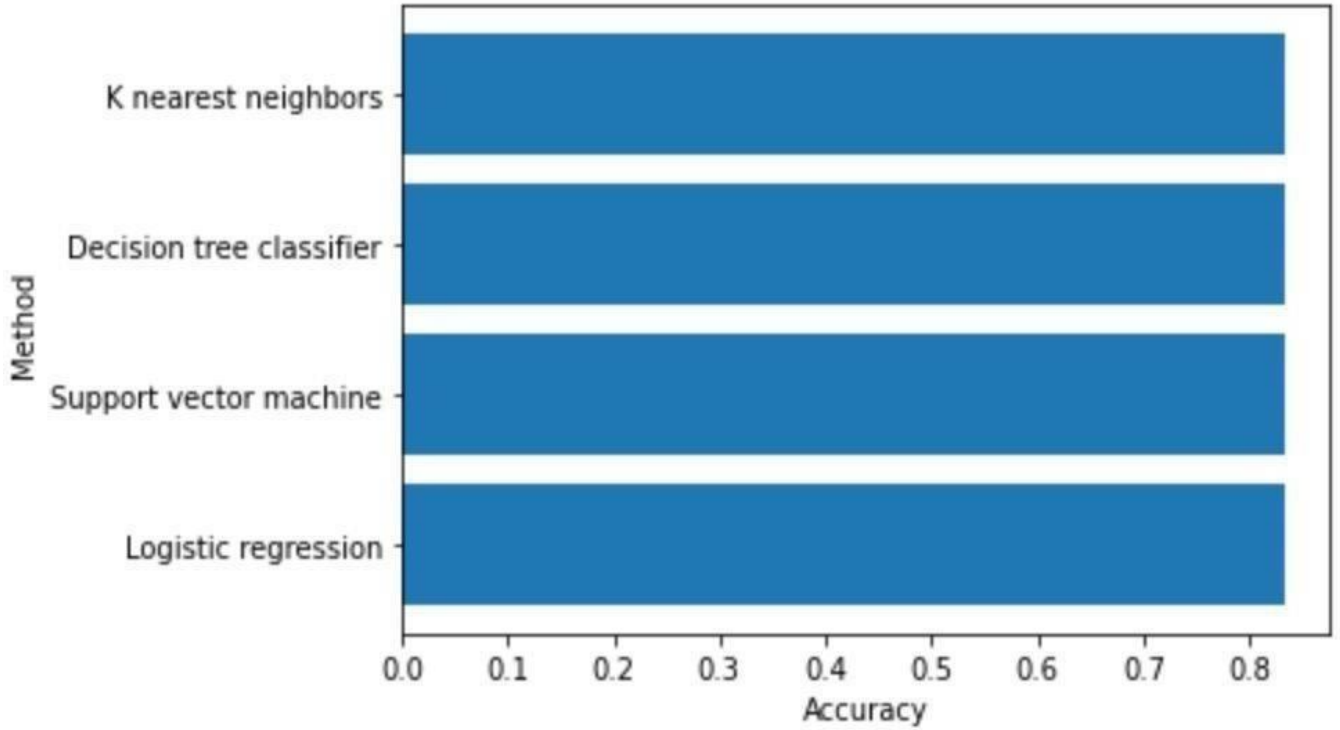
Success rate for low weight payloads is higher than heavy weighted.

Section 5

Predictive Analysis (Classification)

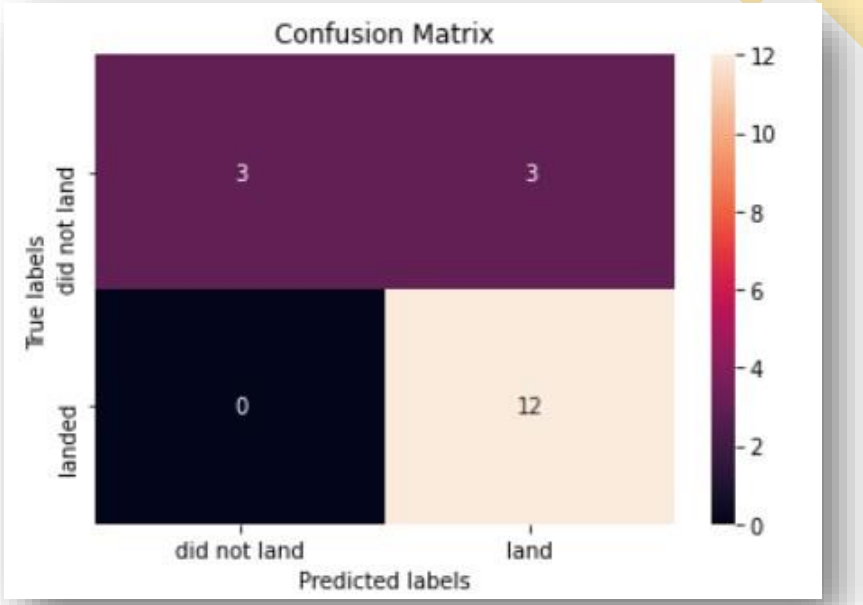
Classification Accuracy

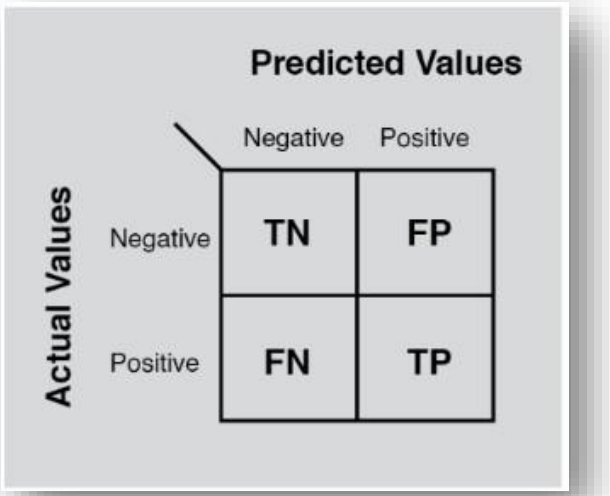
- In the test set, the accuracy of all models was virtually the same at 83.33%
- More data is needed to improve the model



Confusion Matrix

- The confusion matrix is the same for all models
- The models predicted 12 successful landing when the true label was successful.
- The model predict 3 failed landing while the actual label was not successful.
- Overall, the model predict quite good at successful landings.





Conclusions

Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate(100%).

KSLC-39A has the highest number of launch successes and the highest success rate among all sites.

Low weighted payloads perform better than the heavier payloads.

In this dataset, all models have the same accuracy (83.33%), but it seems that more data is needed to determine the optimal model due to the small data size.

Thank you!