IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

MINAL KANADE
2/5/2022

# Outline

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

# Executive Summary

- Summary of methodologies
  - ➢ Data Collection
  - ➢ Data Wrangling
  - ➢ EDA with Data Visualization
  - ➢ EDA with SQL
  - ➢ 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 in screenshots
  - ➢ Predictive analysis Results

# Introduction

- Project background and context
- In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - What are the factors influencing the rocket will land successfully?
  - What will be the interaction among various features that will determine the success rate of successful landing?
  - What conditions the SpaceX have to achieve to get best results and  best success  landing rate?

Section 1

# Methodology

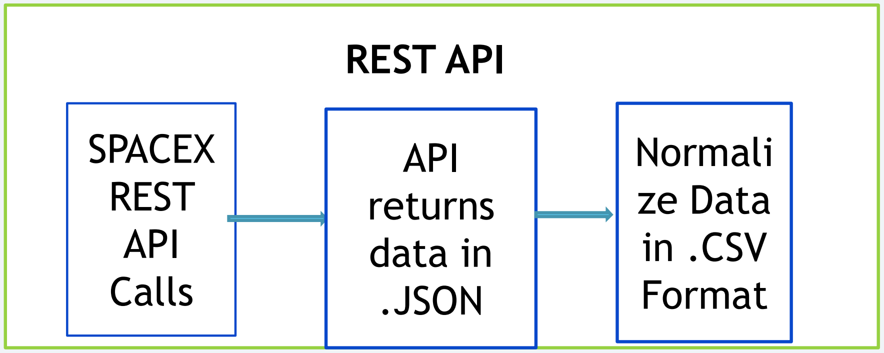
# Methodology

## Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - One Hot encoding was applied to the features for machine learning.
- 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 sets were collected using :
- Launch Data is pulled thru requests to SpaceX Rest API.
- The REST API provided data about Launces ,rockets used, payload, landing outcome, launch specification, etc.
- Another data source used was collecting Falcon 9 Launch Data from web scraping Wikipedia using BeautifulSoup object.

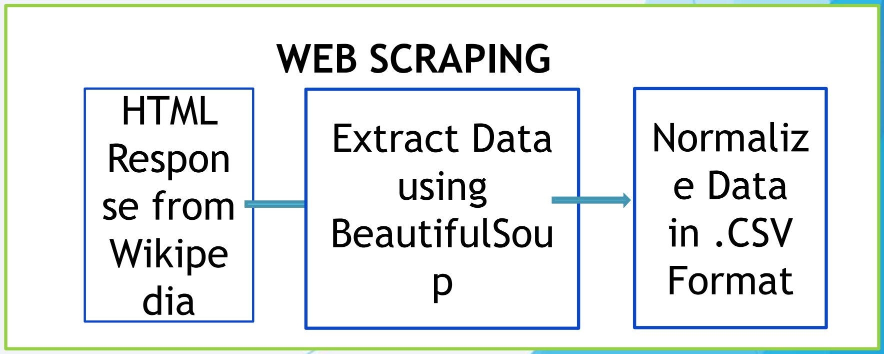


**REST API**

SPACEX REST API Calls

API returns data in .JSON

Normalize Data in .CSV Format



**WEB SCRAPING**

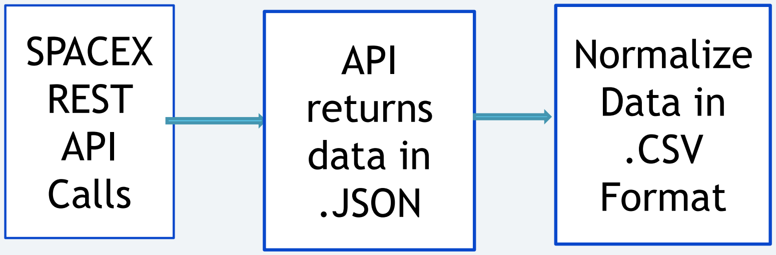
HTML Response from Wikipedia

Extract Data using BeautifulSoup

Normalize Data in .CSV Format

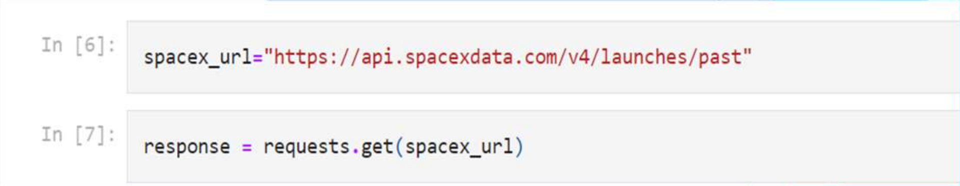
# Data Collection – SpaceX API

- The get request call to the SpaceX API for Data . Then did some basic data wrangling and formatting. Also cleaned the Data thru API. Normalized the JSON Data to Flat File.



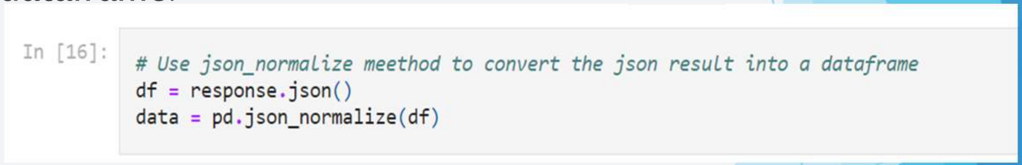
- Github URL to the Notebook

1. Request rocket launch data from SpaceX API with the following URL



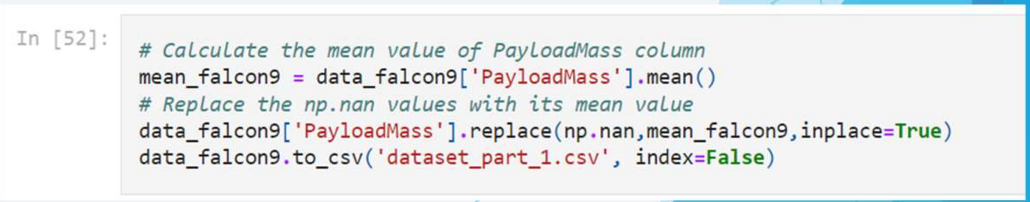
```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]:  response = requests.get(spacex_url)
```

2. Use json_normalize method to convert the json results to dataframe.



```
In [16]:  # Use json_normalize meethod to convert the json result into a dataframe
          df = response.json()
          data = pd.json_normalize(df)
```

3. Then perform data cleaning and filling in the missing values.

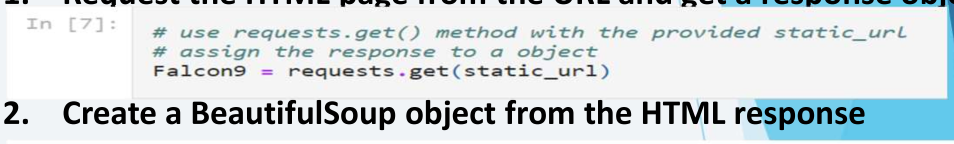


```
In [52]:  # Calculate the mean value of PayloadMass column
          mean_falcon9 = data_falcon9['PayloadMass'].mean()
          # Replace the np.nan values with its mean value
          data_falcon9['PayloadMass'].replace(np.nan,mean_falcon9,inplace=True)
          data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

# Data Collection - Scraping

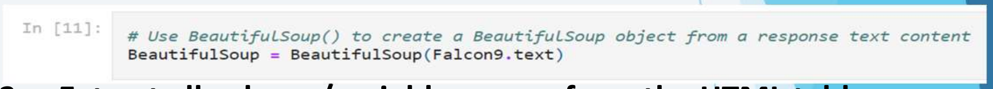
- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia
- Web scrap Falcon 9 launch records with BeautifulSoup
- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame
- GITHUB URL  to the Notebook

1. **Request the HTML page from the URL and get a response object.**



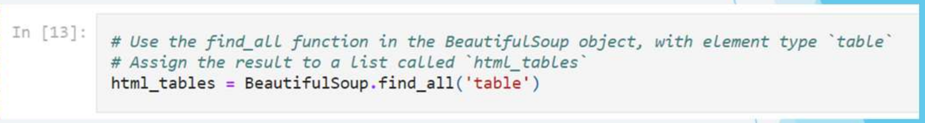
```
In [7]:  # use requests.get() method with the provided static_url
         # assign the response to a object
         Falcon9 = requests.get(static_url)
```

2. **Create a BeautifulSoup object from the HTML response**



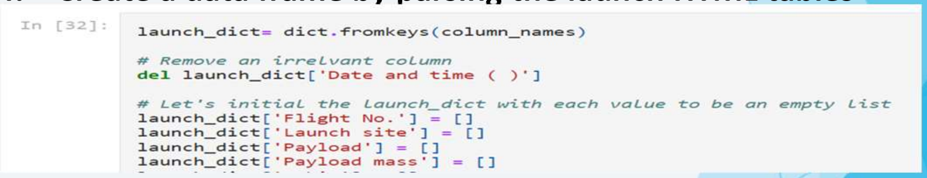
```
In [11]:  # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
          BeautifulSoup = BeautifulSoup(Falcon9.text)
```

3. **Extract all column/variable names from the HTML table**



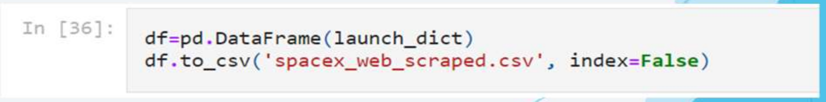
```
In [13]:  # Use the find_all function in the BeautifulSoup object, with element type `table`
          # Assign the result to a list called `html_tables`
          html_tables = BeautifulSoup.find_all('table')
```

4. **Create a data frame by parsing the launch HTML tables**



```
In [32]:  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'] = []
```

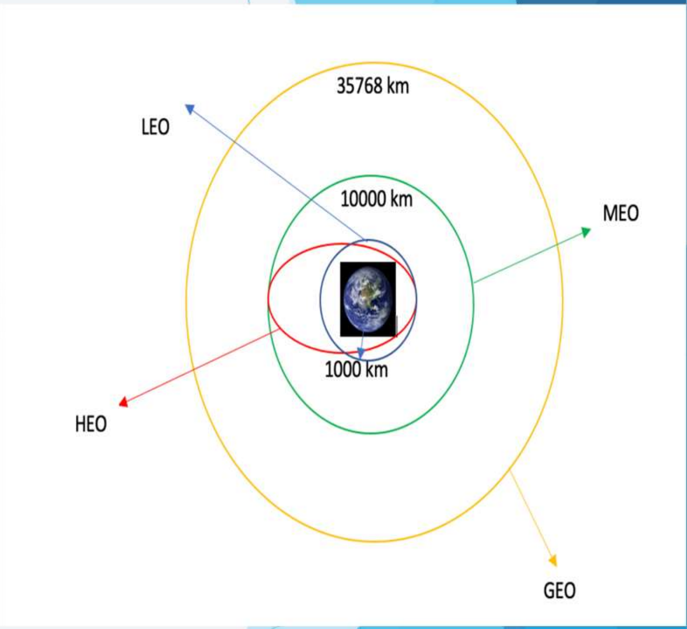
5. **Create a dataframe and export to .csv**



```
In [36]:  df=pd.DataFrame(launch_dict)
          df.to_csv('spacex_web_scraped.csv', index=False)
```

# Data Wrangling

- Performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



Calculate the number of launches for 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

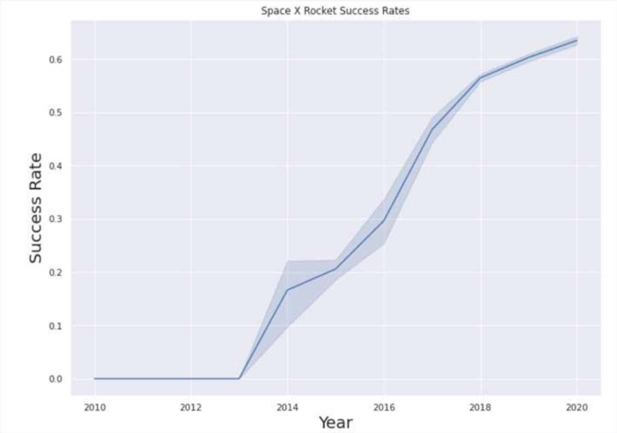
Determine the success rate

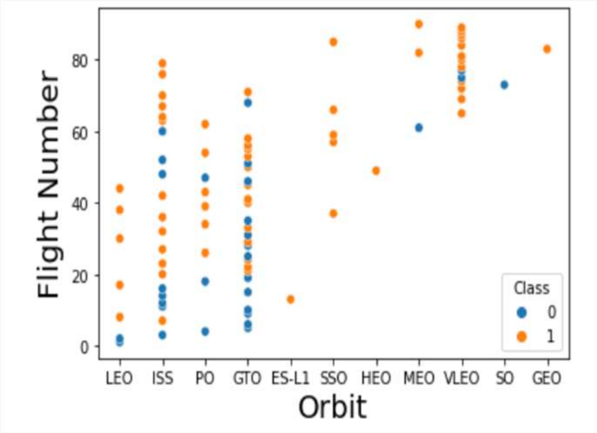
Export Dataset to .CSV

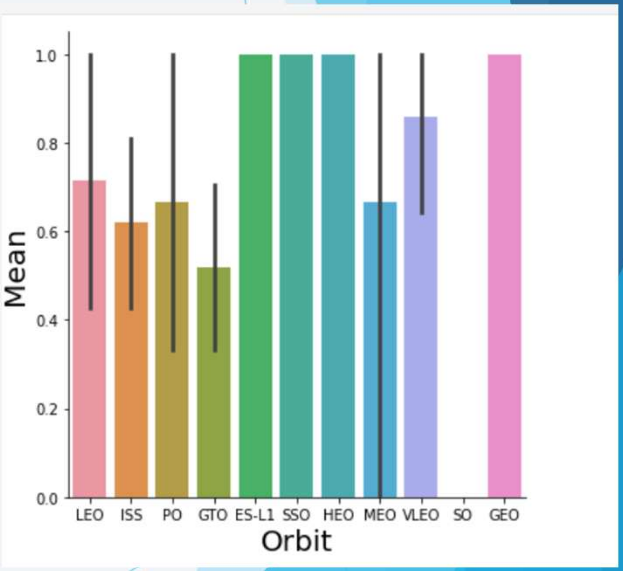
- Github URL to the Notebook

# EDA with Data Visualization

- Performed exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib. Explored the data by visualizing the relationship between Flight number & launch site, launch success yearly trend,etc.







- [Github URL to the Notebook]

# EDA with SQL

- Performed different SQL queries to understand the dataset
  - **Display the names of the unique launch sites in the space mission**
  - **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 the Notebook

# Build an Interactive Map with Folium

- I have marked  all launch sites on a map, with the success/failed launches for each site on the map with the map objects such as markers, circles, lines to a folium map.
- Assigned feature launch outcome(Success or Failure) to class 0 and 1 .
- Using color label marker cluster identified the launch site with high success rate
- Calculated the distances between a launch site to its proximities.
- Tried to answer below questions
  - 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

- GitHub URL to the Notebook

# Build a Dashboard with Plotly Dash

- Build and interactive dashboard using Plotly Dash
- I have plotted the pie charts showing the total launches by launch sites.
- I have plotted the scatter graph showing the relationship with outcome and payload mass for booster version.

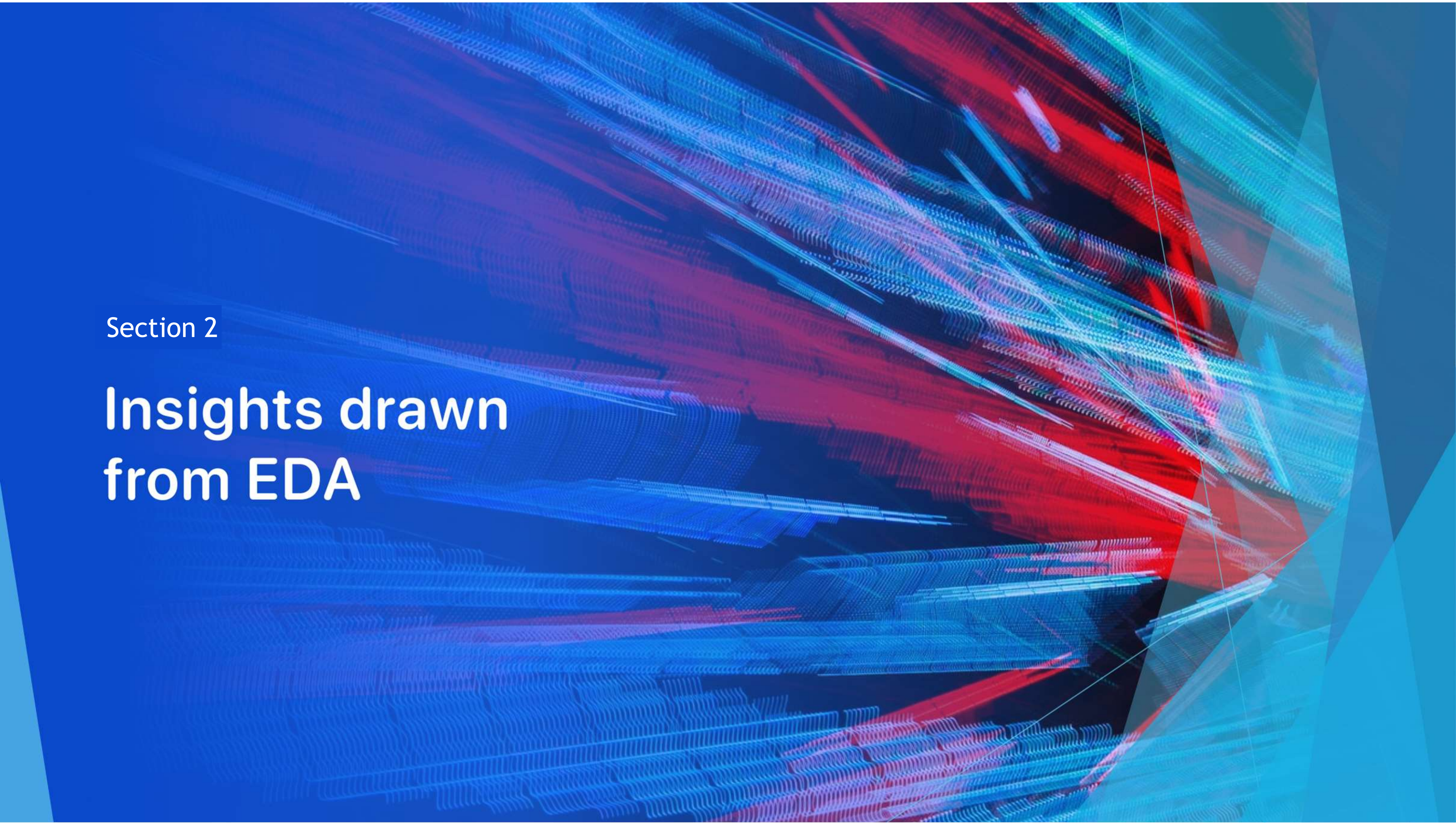
- GitHub URL to the Notebook

# Predictive Analysis (Classification)

- I have loaded the data using Panda and numpy, transformed the data and divided data into Test and Training set
- I have build different machine learning models using GridSearchCV
- Evaluated model by using - Check accuracy for each model, get tuned parameters for each algorithm, Plot confusion Metrix
- Improved the Model using Feature Engineering and Algorithm Tuning
- Finding the best performing Classification Model
- GitHub URL to the Notebook

# Results

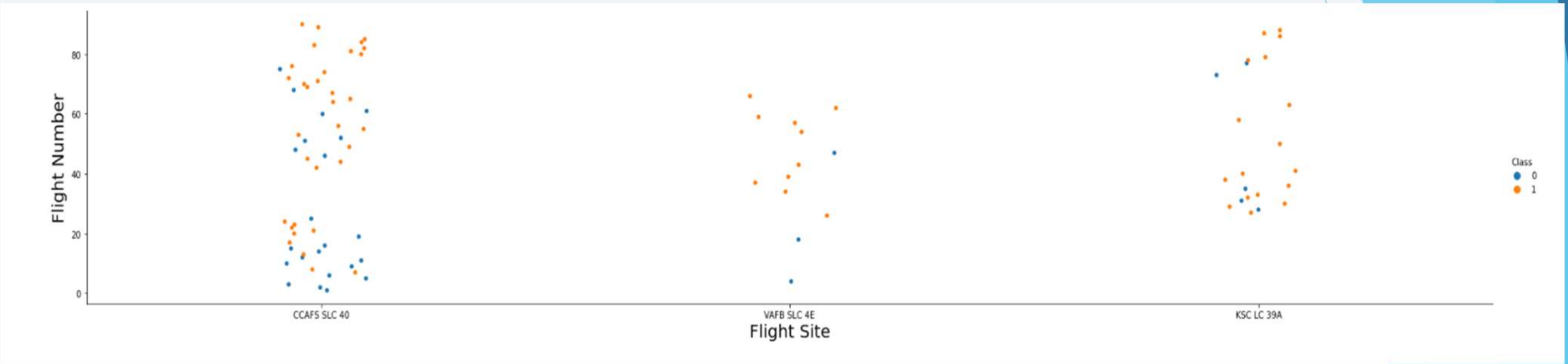
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

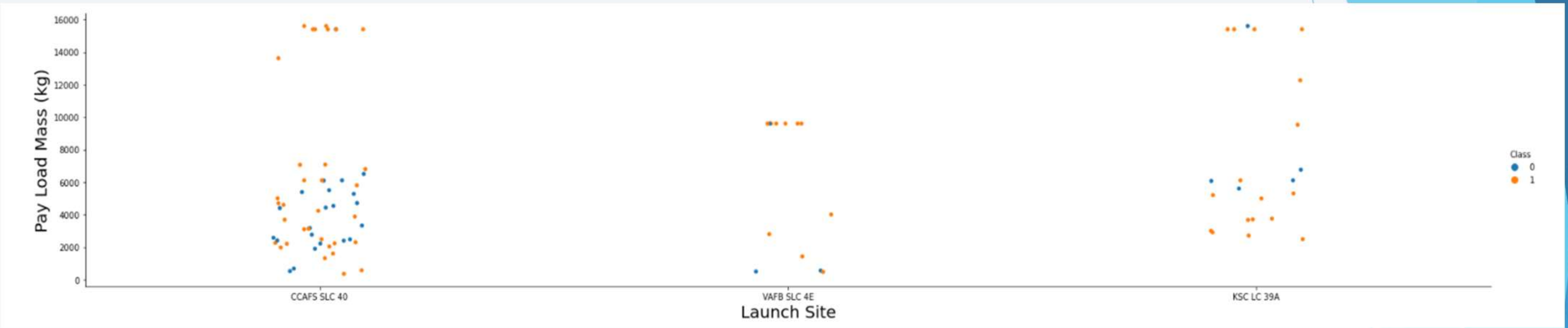
# Insights drawn from EDA

# Flight Number vs. Launch Site



- From the above scatter plot , shows the more number of the flights at launch sight greater the success rate at that site.

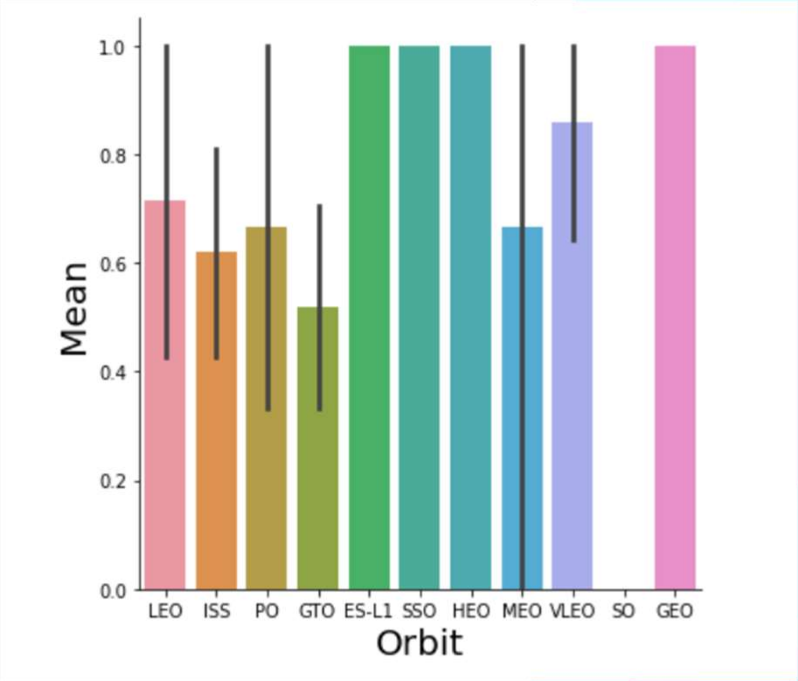
# Payload vs. Launch Site



- VAFB-SLC launch site there are no rockets launched for heavy payload mass(kg)(greater than 10000).
- CCAFS SLC40 Launch site have more success rate for heavy payload mass(kg)(greater than 12000)

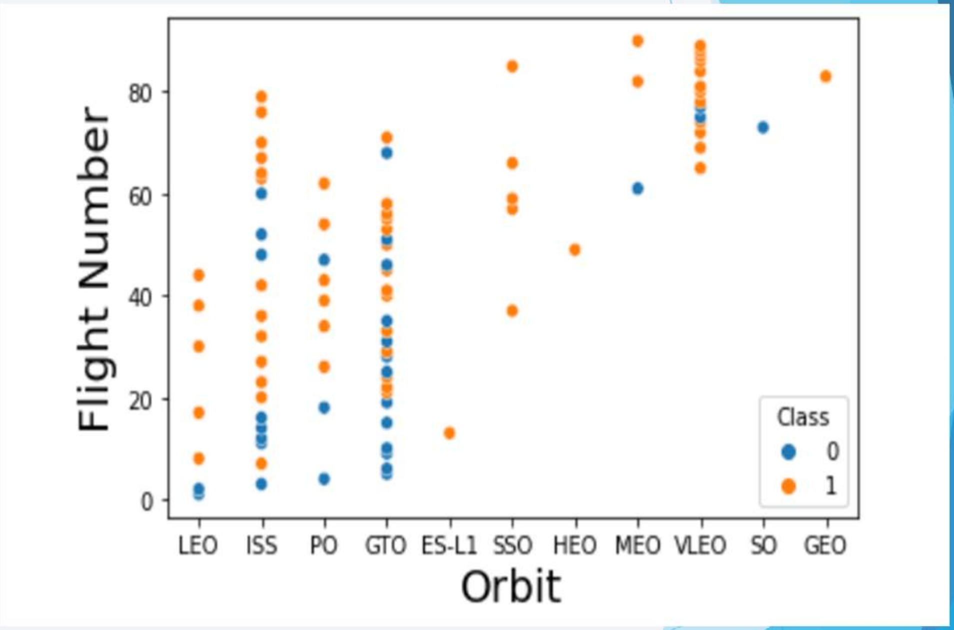
# Success Rate vs. Orbit Type

- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- Orbit ES-L1,SSO,HEO have the highest success rate



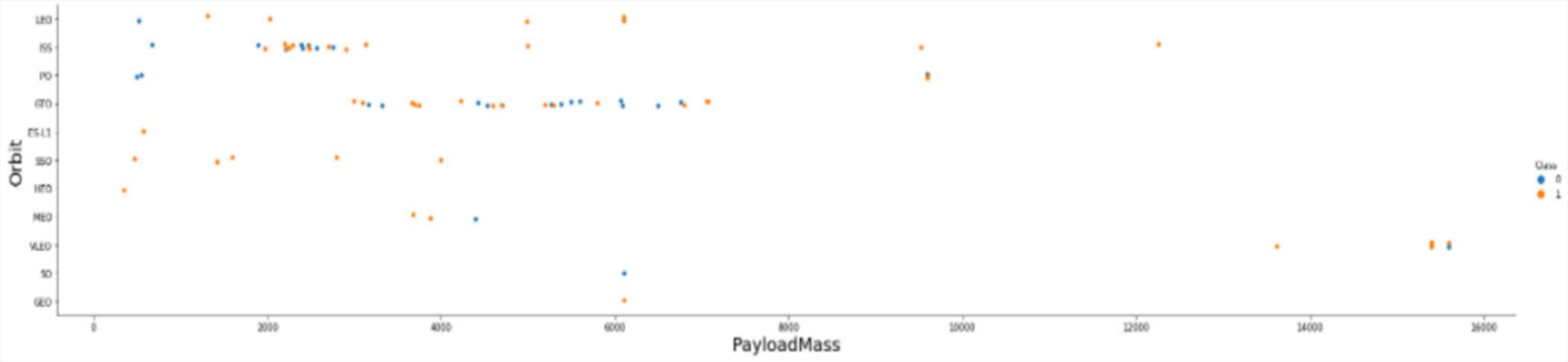
# Flight Number vs. Orbit Type

- 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.



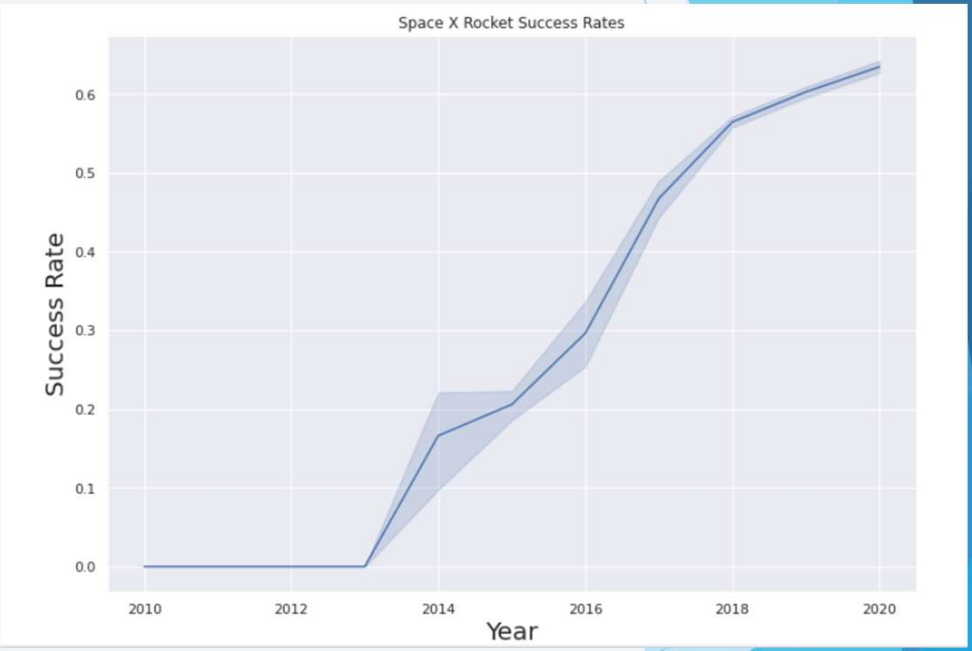
# Payload vs. Orbit Type

- With Heavy payload ,successful landing for PO, LEO and ISS Orbits



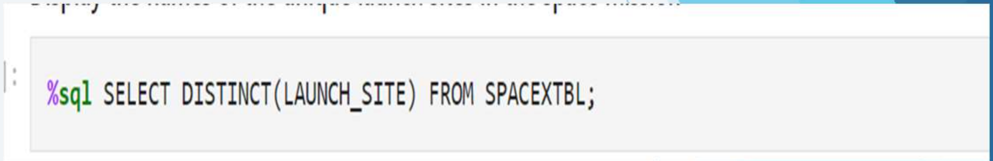
# Launch Success Yearly Trend

- From chart we can observe that the success rate from 2013 to 2020 was increasing.
- Till 2013 the success rate as zero, so first successful attempt was after year 2013

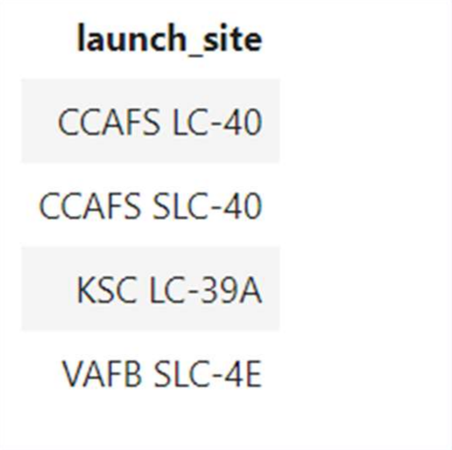


# All Launch Site Names

- SQL Query used to find the names of the unique launch sites :-
- Used Distinct() for listing unique Launch site names.



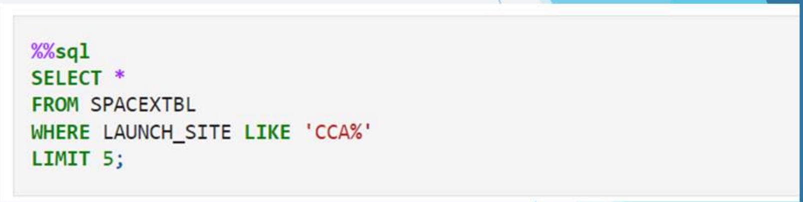
```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```



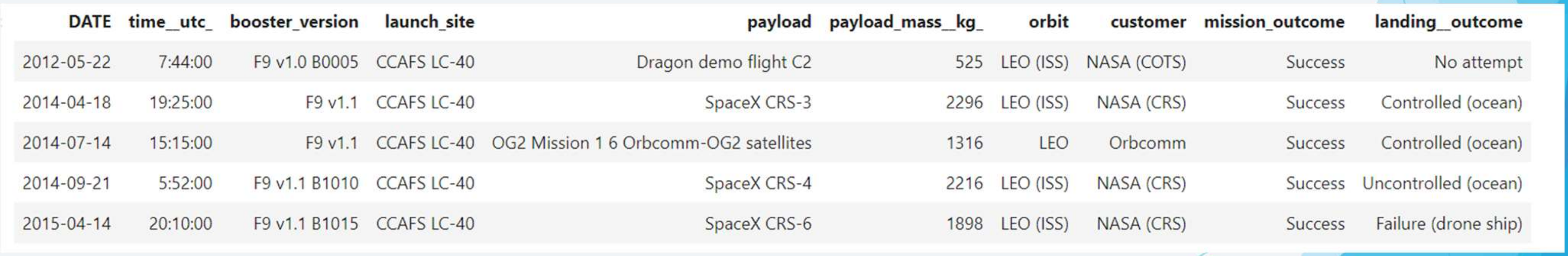
| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

# Launch Site Names Begin with 'CCA'

- Query Used to Find 5 records where launch sites begin with `CCA` -
- Used “LIKE” statement to query the site names starting with “CCA”
- “LIMIT” is used for only finding 5 records



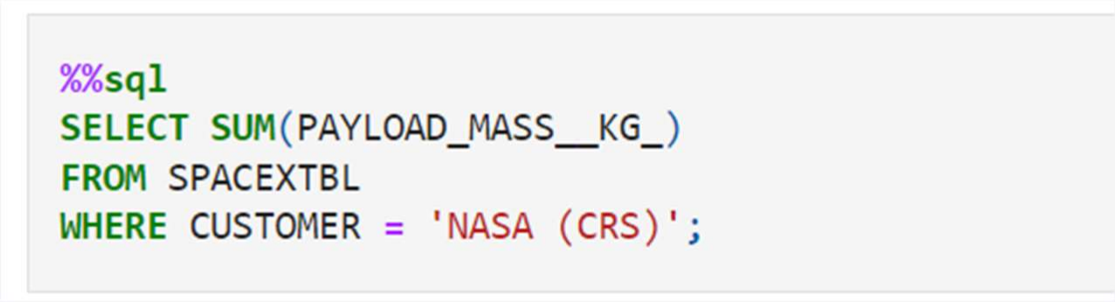
```
%%sql
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```



| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2014-04-18 | 19:25:00 | F9 v1.1 | CCAFS LC-40 | SpaceX CRS-3 | 2296 | LEO (ISS) | NASA (CRS) | Success | Controlled (ocean) |
| 2014-07-14 | 15:15:00 | F9 v1.1 | CCAFS LC-40 | OG2 Mission 1 6 Orbcomm-OG2 satellites | 1316 | LEO | Orbcomm | Success | Controlled (ocean) |
| 2014-09-21 | 5:52:00 | F9 v1.1 B1010 | CCAFS LC-40 | SpaceX CRS-4 | 2216 | LEO (ISS) | NASA (CRS) | Success | Uncontrolled (ocean) |
| 2015-04-14 | 20:10:00 | F9 v1.1 B1015 | CCAFS LC-40 | SpaceX CRS-6 | 1898 | LEO (ISS) | NASA (CRS) | Success | Failure (drone ship) |

# Total Payload Mass

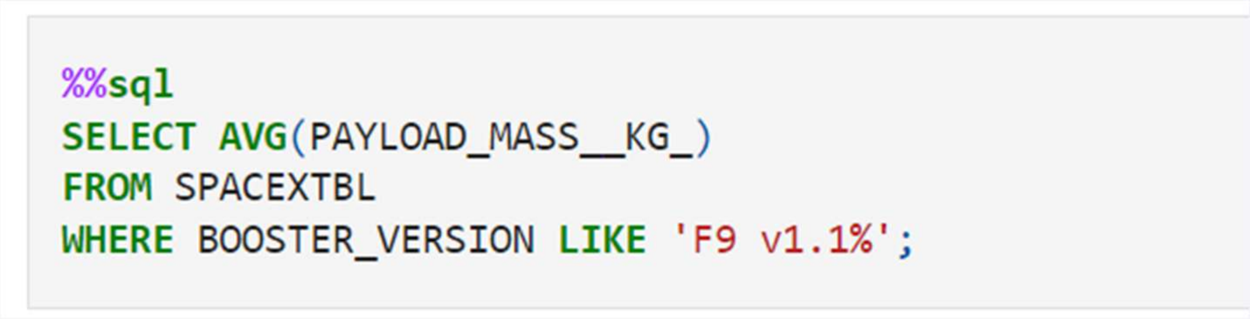
- Total payload carried by boosters from NASA - 45596
- Below Query used to calculate the Total payload



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

# Average Payload Mass by F9 v1.1

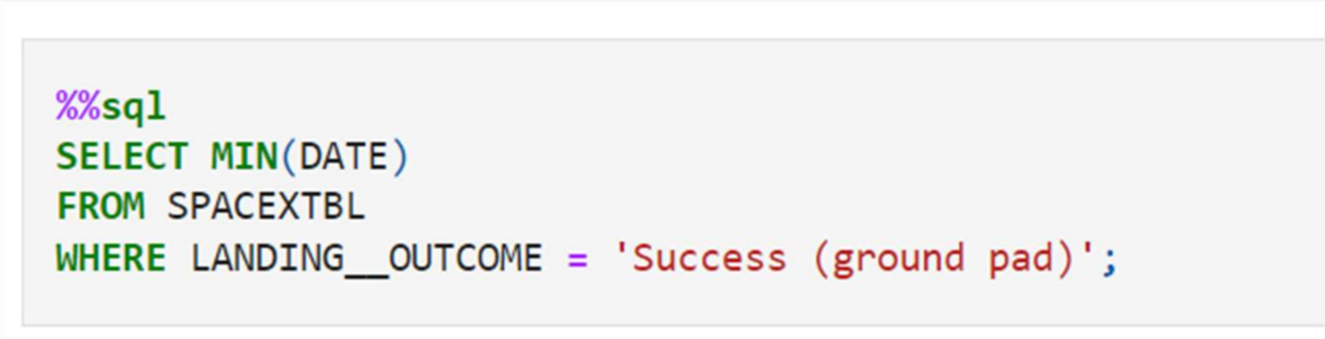
- Average payload mass carried by booster version F9 v1.1 -  2928.4
- Below query is used :



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

# First Successful Ground Landing Date

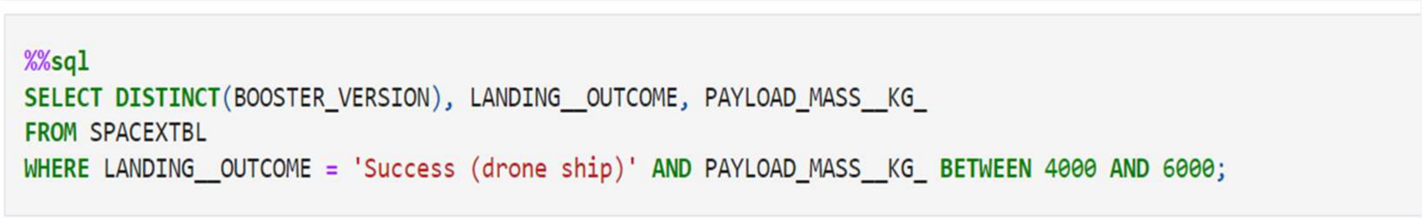
- Find the dates of the first successful landing outcome on ground pad
  - -2015 -12 -25
- Min Function used to first date for Successful landing outcome



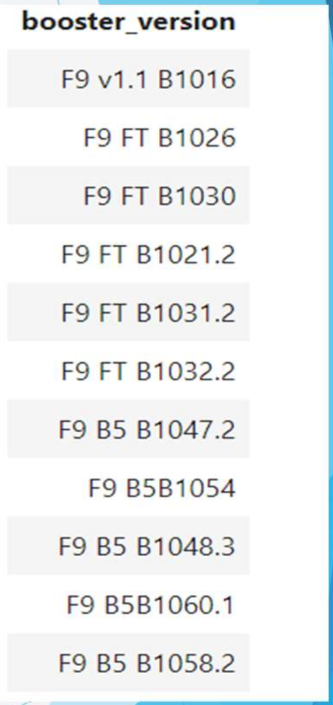
```
%%sql
SELECT MIN(DATE)
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

# Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Below query is used , it checks the Payload mass by using “BETWEEN” Keyword.



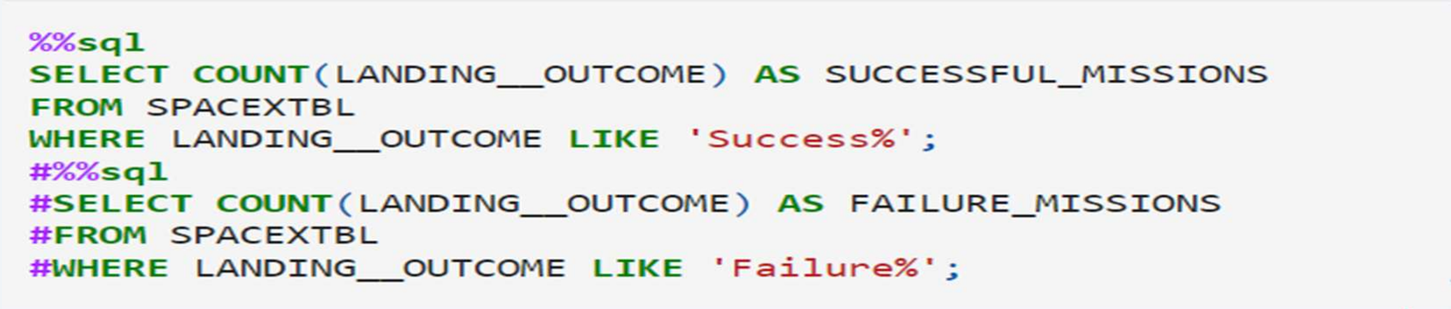
```
%%sql
SELECT DISTINCT(BOOSTER_VERSION), LANDING__OUTCOME, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```



| booster_version |
|---|
| F9 v1.1 B1016 |
| F9 FT B1026 |
| F9 FT B1030 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |
| F9 FT B1032.2 |
| F9 B5 B1047.2 |
| F9 B5B1054 |
| F9 B5 B1048.3 |
| F9 B5B1060.1 |
| F9 B5 B1058.2 |

# Total Number of Successful and Failure Mission Outcomes

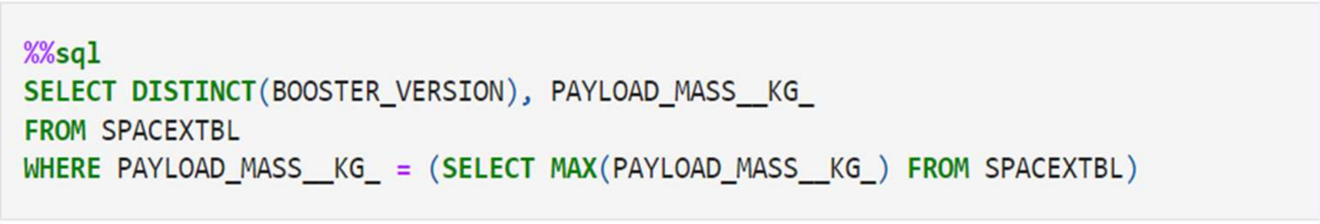
- Total number of successful mission outcomes -100
- Total number of failure mission outcomes - 1
- Below query is used .
- Total numbers are calculated using “COUNT” function and by “LIKE” to check whether th landing oucome was success or failure



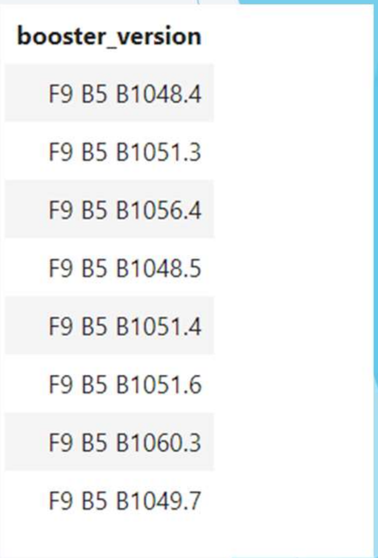
```
%%sql
SELECT COUNT(LANDING__OUTCOME) AS SUCCESSFUL_MISSIONS
FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE 'Success%';
#%%sql
#SELECT COUNT(LANDING__OUTCOME) AS FAILURE_MISSIONS
#FROM SPACEXTBL
#WHERE LANDING__OUTCOME LIKE 'Failure%';
```

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Below Query is used :-
- MAX Function is used to calculated maximum payload
- Used a subquery to find distinct names of Booster



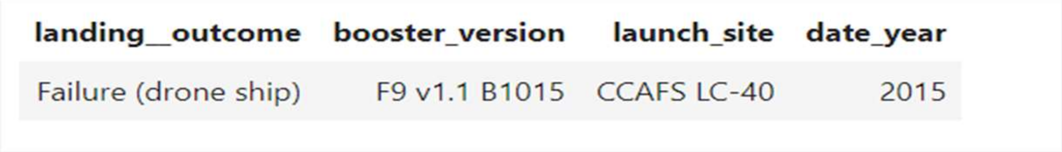
```
%%sql
SELECT DISTINCT(BOOSTER_VERSION), PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```



| booster_version |
|---|
| F9 B5 B1048.4 |
| F9 B5 B1051.3 |
| F9 B5 B1056.4 |
| F9 B5 B1048.5 |
| F9 B5 B1051.4 |
| F9 B5 B1051.6 |
| F9 B5 B1060.3 |
| F9 B5 B1049.7 |

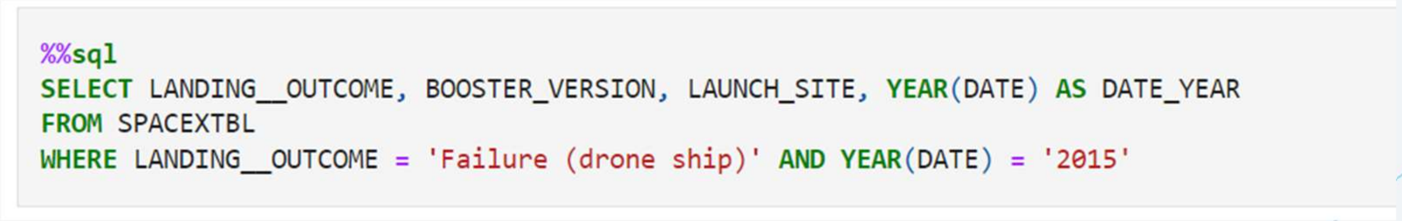
# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



| landing__outcome | booster_version | launch_site | date_year |
|---|---|---|---|
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 | 2015 |

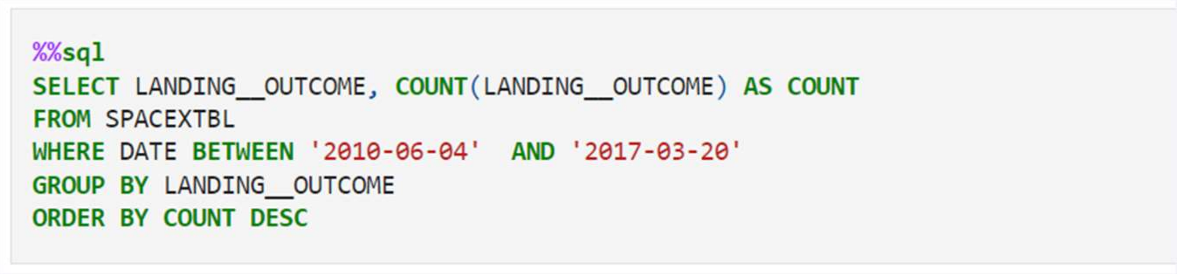
- Below Query is used –
- Used the “YEAR” function to extract the Year from the Date column and used where clause to select Failure and 2015 year



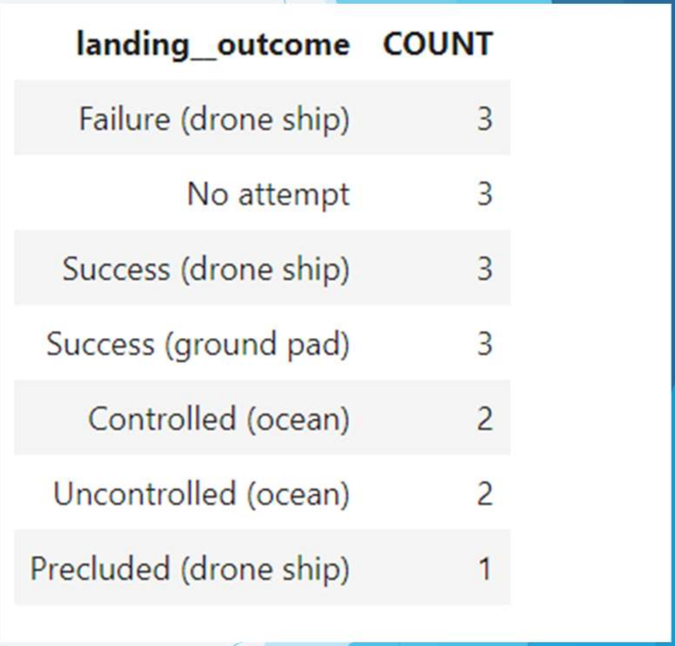
```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, YEAR(DATE) AS DATE_YEAR
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

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

- Below Query is used –
- Used “WHERE” clause to filter dates and “GROUP BY” Clause to Landing outcome. ORDER by “COUNT” to rank in “DESC” for Descending order



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



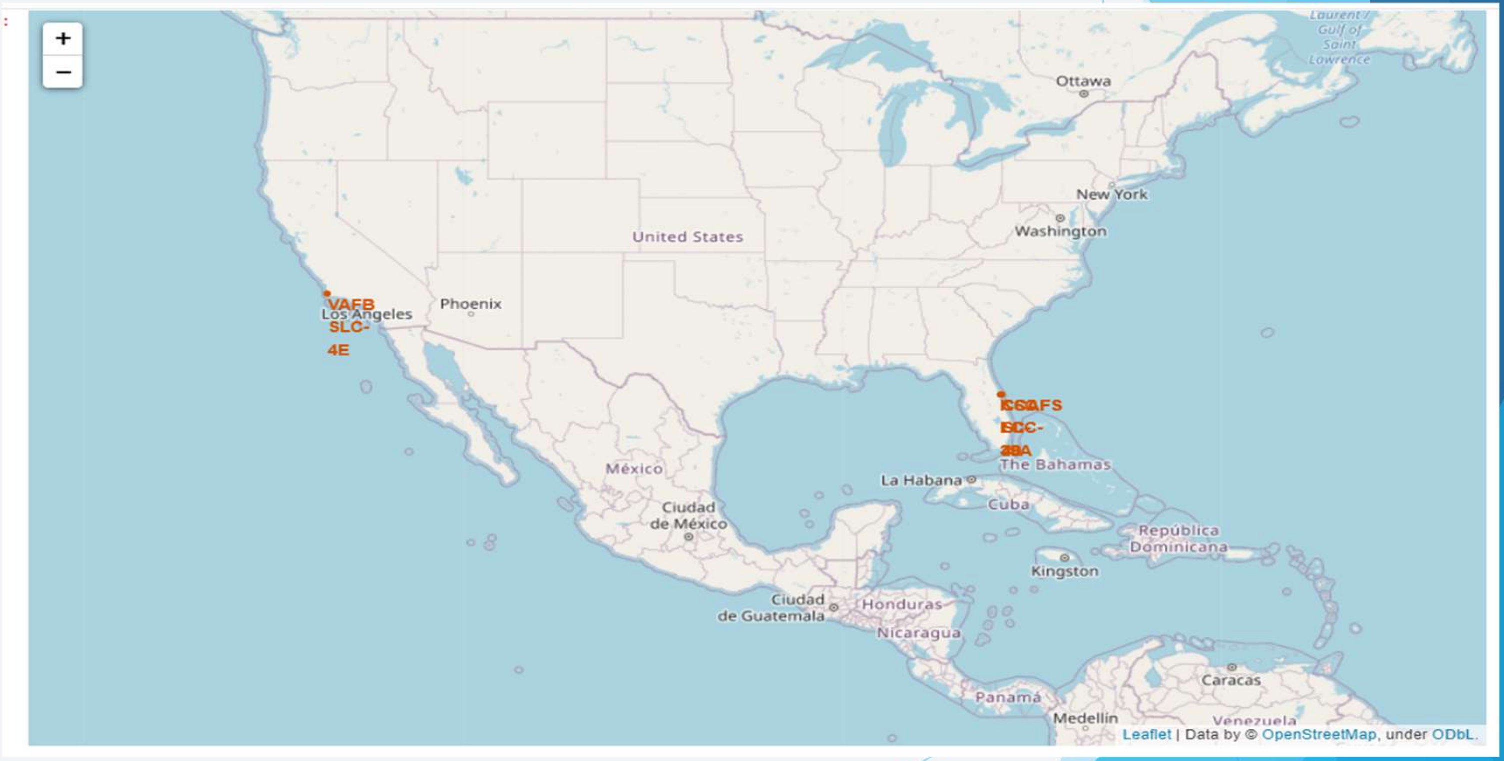
| landing__outcome | COUNT |
|---|---|
| Failure (drone ship) | 3 |
| No attempt | 3 |
| Success (drone ship) | 3 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

Section 3

# Launch Sites Proximities Analysis

# ALL Lunch Site on global Map Marker

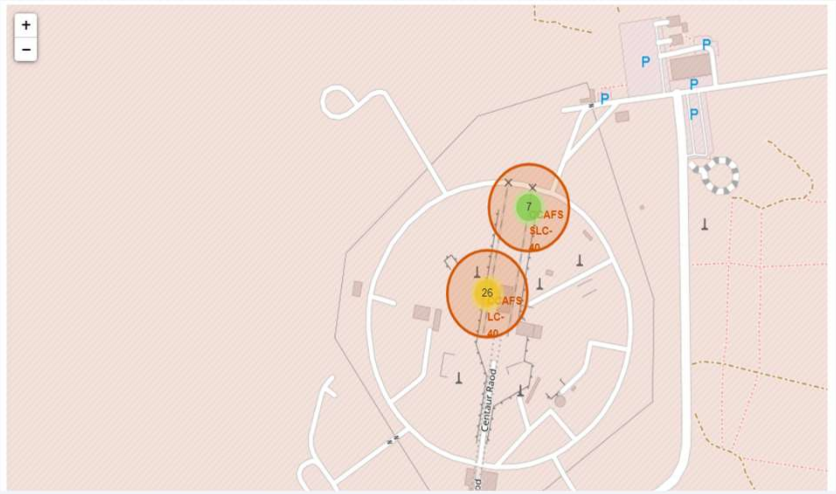
- From Map, We can see the SPACEX Launch sites are in California and Florida near Coasts

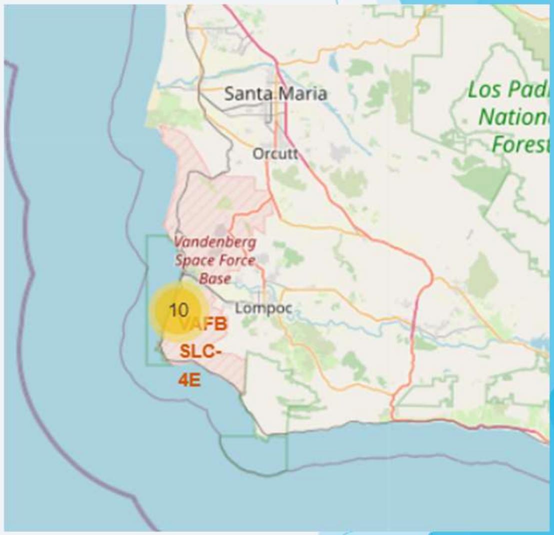


# Success/failed launches using Color Labeled launch

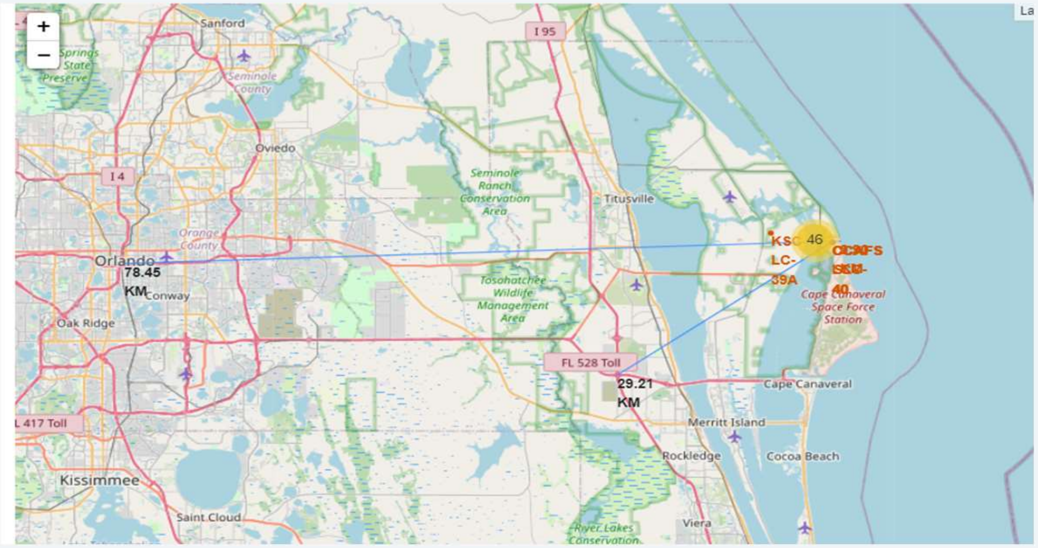
- FLORIDA Launch Site
- Green marker shows successful Launches

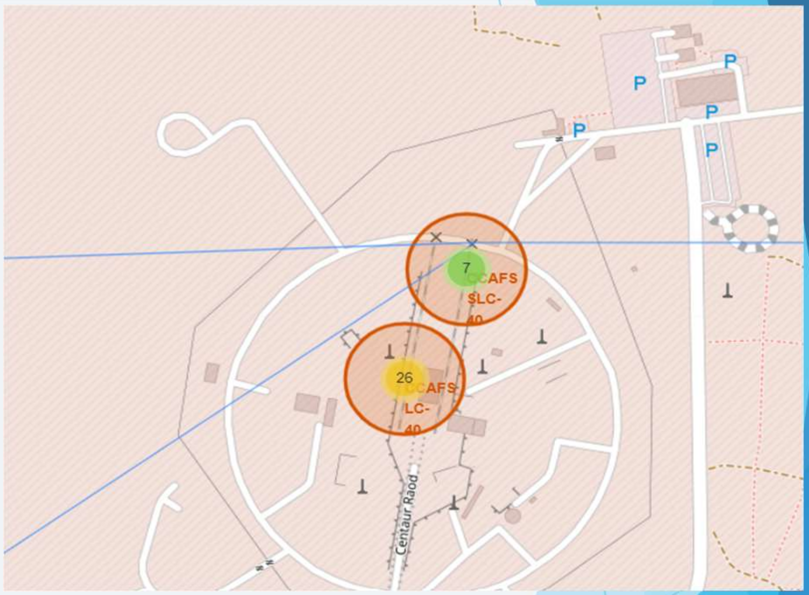
- California Launch Site



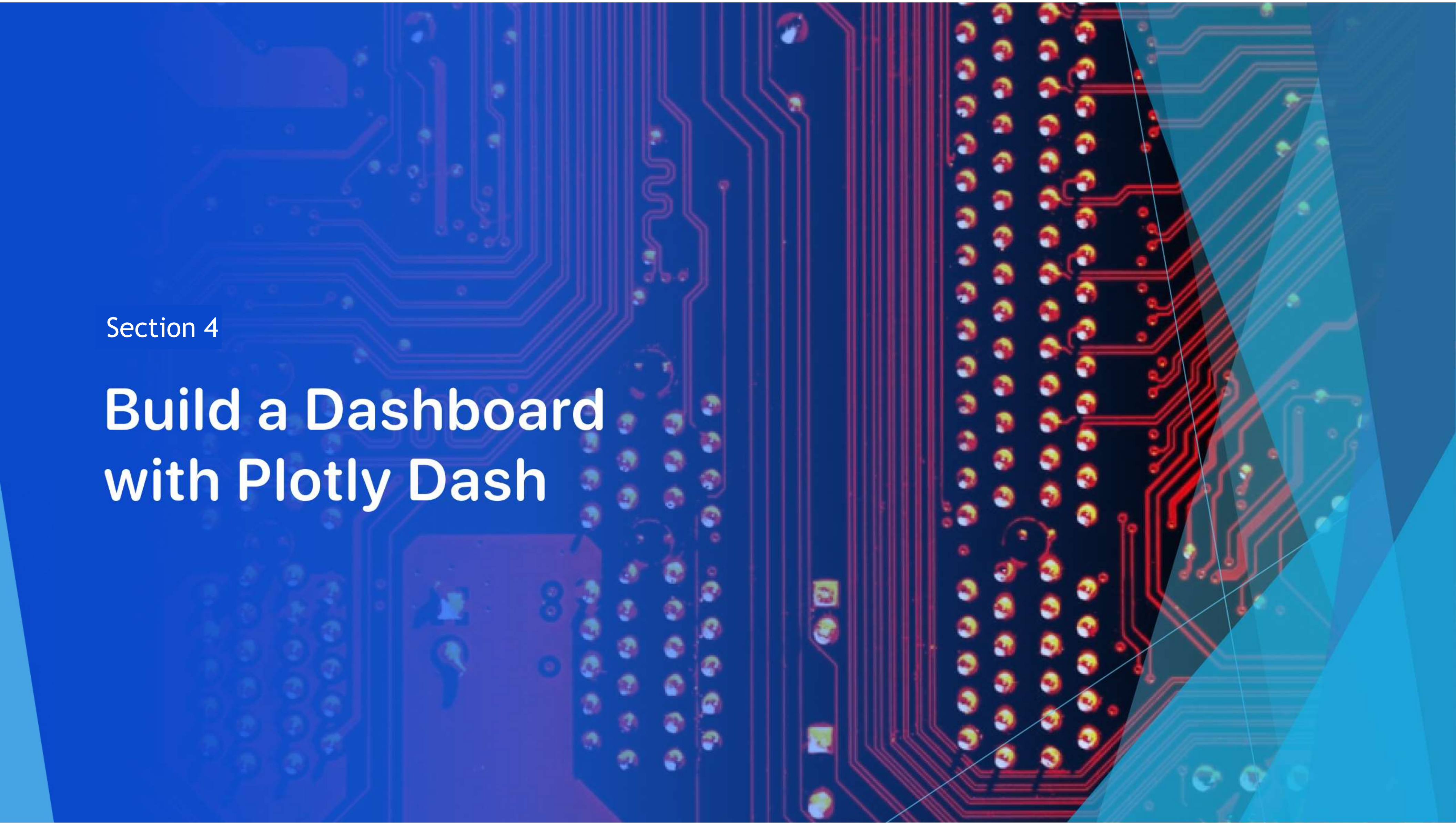


# Launch Site Distance to Land Marks





- 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

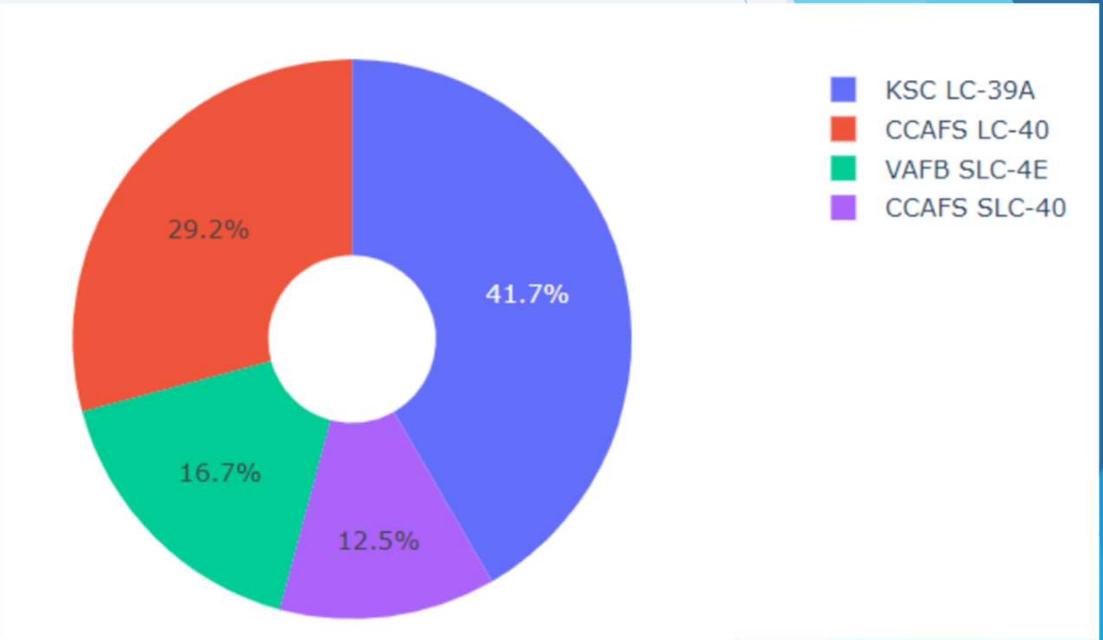


Section 4

# Build a Dashboard with Plotly Dash

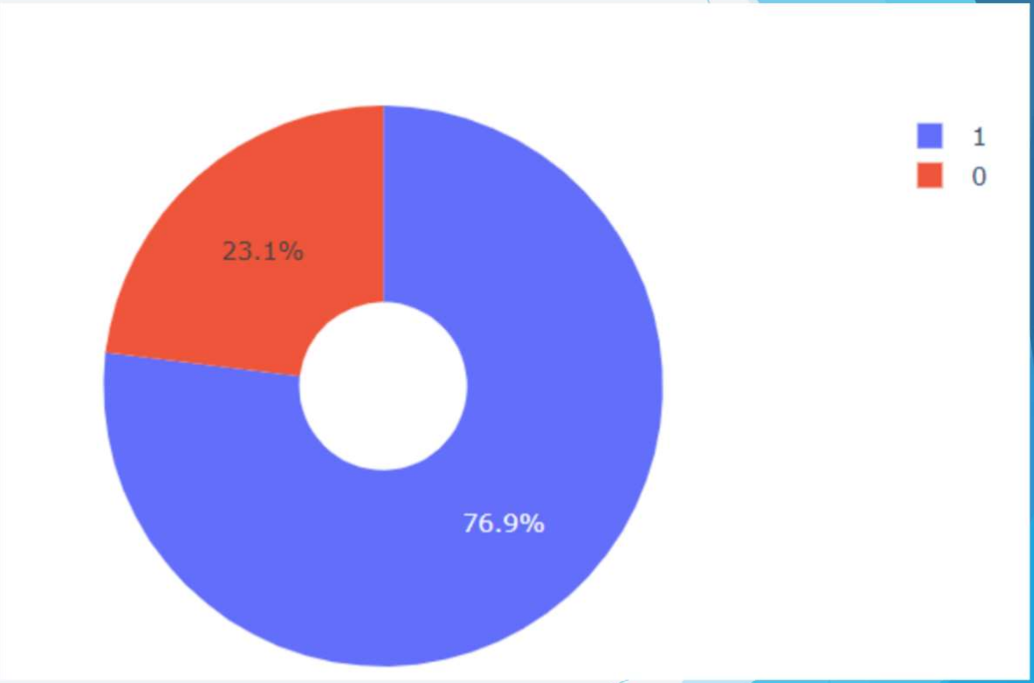
# Pie chart showing launch success for all Sites

- From the Pie chart, KSC LC - 39 A had the most successful launches from all sites



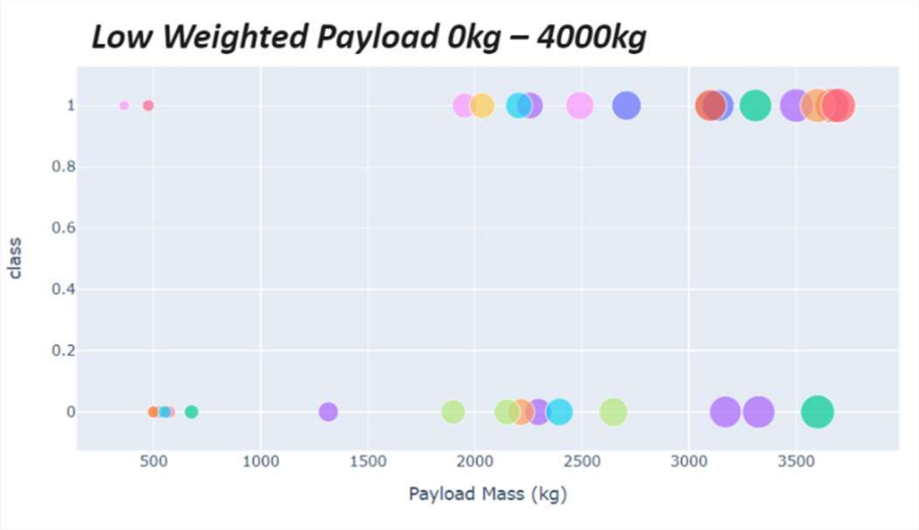
# Piechart for the launch site with highest launch success ratio

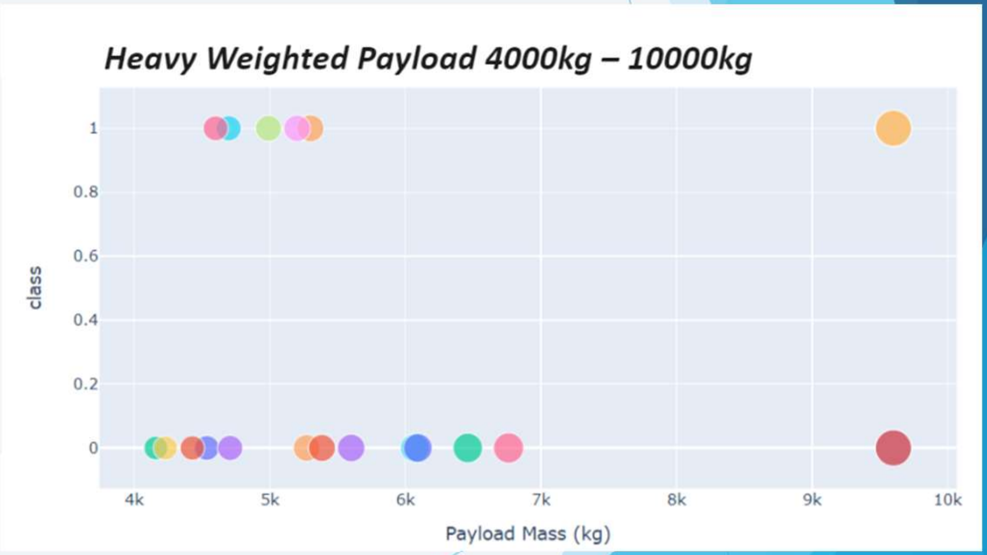
- KSC LC -39A which have highest launch success ratio of 76.9%, while the failure rate is 23.1%



# Payload vs. Launch Outcome scatter plot for all sites

- From the chart below , the conclusion is the success rate for low weighted payload is higher than heavy weighted payload



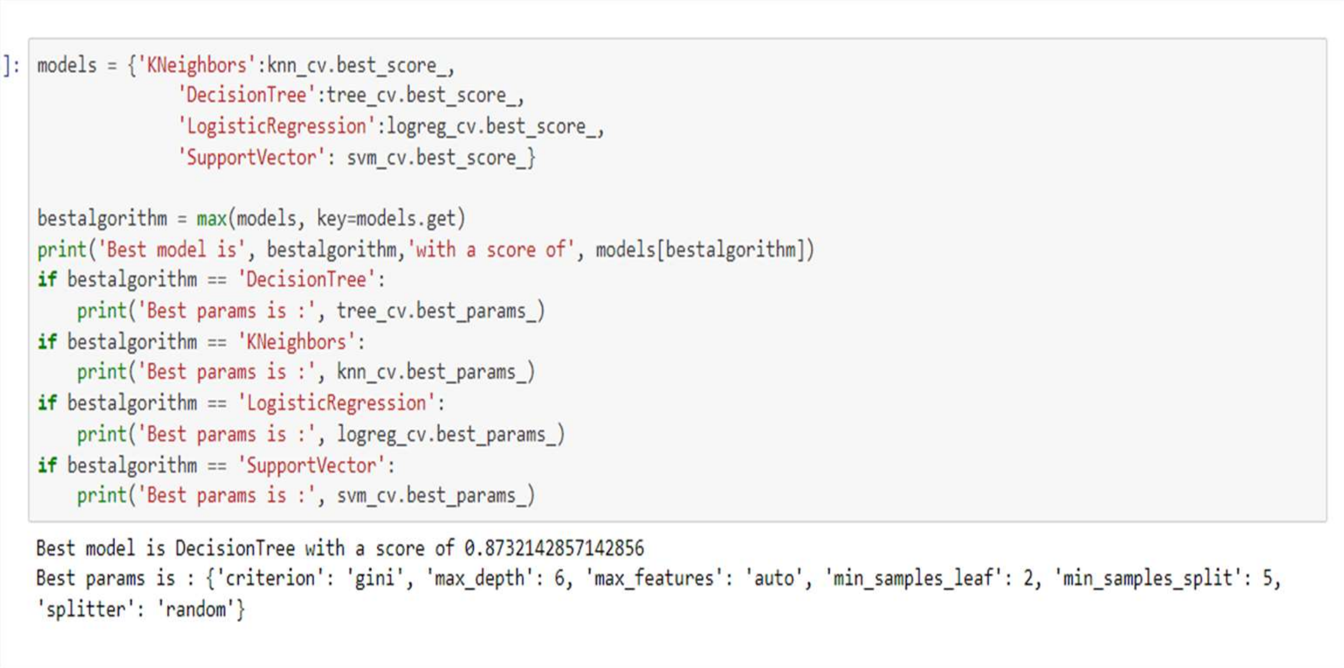


Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- Decision Tree Classified Model have the Highest classification accuracy

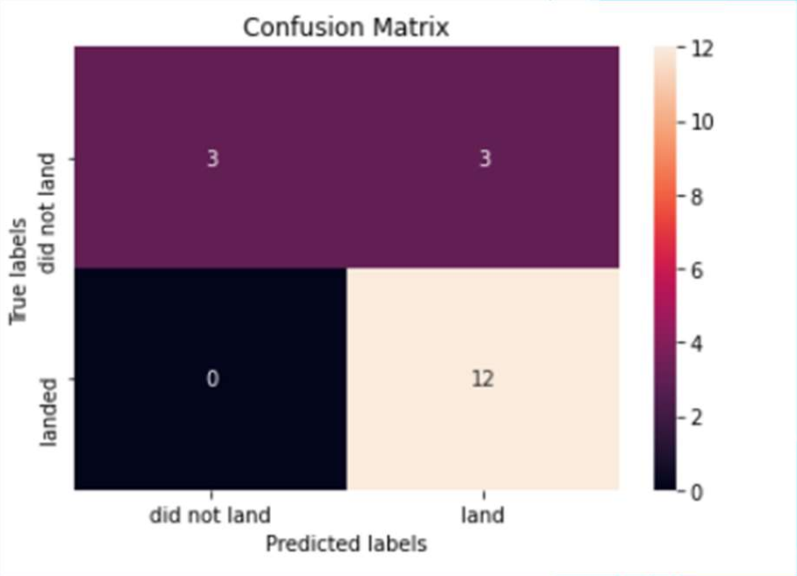


```
]: models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

   bestalgorithm = max(models, key=models.get)
   print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
   if bestalgorithm == 'DecisionTree':
       print('Best params is :', tree_cv.best_params_)
   if bestalgorithm == 'KNeighbors':
       print('Best params is :', knn_cv.best_params_)
   if bestalgorithm == 'LogisticRegression':
       print('Best params is :', logreg_cv.best_params_)
   if bestalgorithm == 'SupportVector':
       print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'random'}
```

# Confusion Matrix

- The Confusion Matrix for Classifier Tree show it can distinguish between the different classes. We see that the major problem is false positives , i.e. unsuccessful landing marked as successful landing marked by classifier.



# Conclusions

Conclusions from the SpaceX project areas below :

- The more number of the flights at launch sight greater the success rate at that site.
- The launch success rate started increasing from 2013 to 2020
- Orbits SSO,HEO,GEO have the most success rate.
- Launch site KSC LC - 39 had the most successful launches from all sites.
- The Decision tree Classifier was the best Classifier for this Data. (with highest accuracy)

# Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!