Winning Space Race with Data Science

Atharva Jirafe
29-Nov-2021

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data collection and data wrangling methodology
  - EDA and interactive visual analytics methodology
  - Predictive analysis methodology
- **Summary of all results**
  - EDA with visualization results
  - EDA with SQL results
  - Interactive map with Folium results
  - Plotly Dash dashboard results
  - Predictive analysis (classification) results

Introduction

- Project background and context

This project 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
  - If we can determine if the first stage will land successfully, we can determine the cost
  - Determine the price of each launch
  - Determine if SpaceX will reuse the first stage

Section 1

Methodology

Methodology

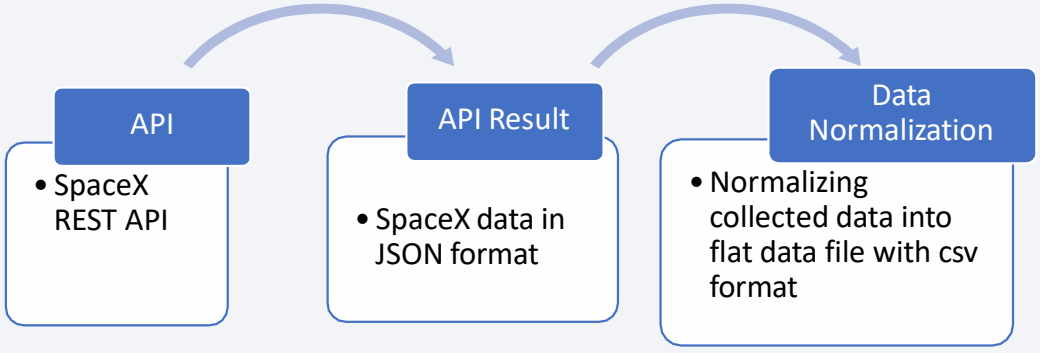
Executive Summary

- Data collection methodology:
  - Collecting the Data by using SpaceX REST API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Exploratory Data Analysis and Feature Engineering to predict if the Falcon 9 first stage will land successfully
- Perform interactive visual analytics using Folium and Plotly Dash
  - To visualize the Launch Data into an interactive map.
- Perform predictive analysis using classification models
- To build, tune, evaluate classification models

Data Collection

The datasets was collected by using SpaceX REST API and Web Scraping

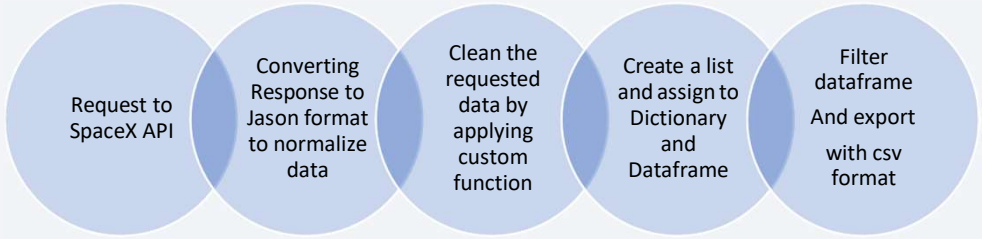
- API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Our objective is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints is a URL api.spacexdata.com/v4/.
- Web scraping used to obtain Falcon 9 Launch data from Wikipedia site using BeautifulSoup library.



SpaceX API Process

Web Scraping Process

Data Collection - SpaceX API



Data Collection with SpaceX API

1 .Getting Response from API

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-
  storage.appdomain.cloud/IBM-DS0321EN-
  SkillsNetwork/datasets/API_call_spacex_api.json'
```

2. Converting Response to a .json file

```
response.json()
data=pd.json_normalize(response.json())
```

3. Apply custom functions to clean data

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

GitHub link

https://github.com/usailky/Applied-Data-Science-Capstone/blob/master/Data%20Collection%20%20API.ipynb

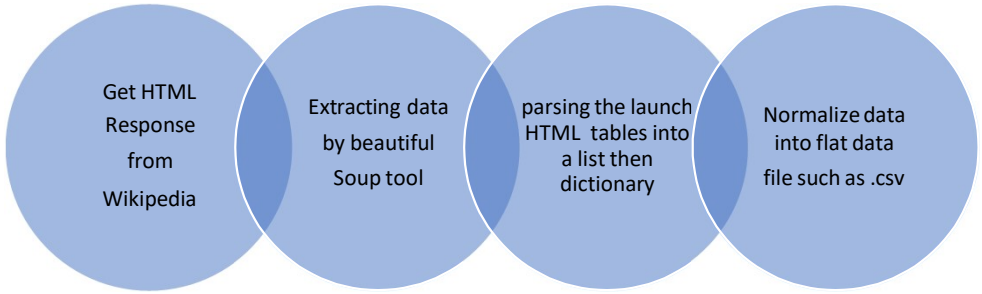
4. Assign list to dictionary then dataframe

```
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(launch_dict)
```

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

```
data_falcon9=df[(df.BoosterVersion=="Falcon 9")]
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping



GitHub link

https://github.com/usailky/Applied-Data-Science-Capstone/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb

Data Collection with SpaceX API

1. Getting Response from HTML

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

2. BeautifulSoup to create object

```
soup=BeautifulSoup(page.text,'html.parser')
```

3. Finding tables

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

4. Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
try :
name =
 extract_column_from_header(
 temp[x])
if (name is not None and
len(name)> 0):
column_names.append(name)
except:
Pass
```

5. Creation of dictionary

```
launch_dict=
 dict.fromkeys(column_names)
del launch_dict['Date and time ( )']
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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table
 in……
```

7. Converting dictionary to dataframe

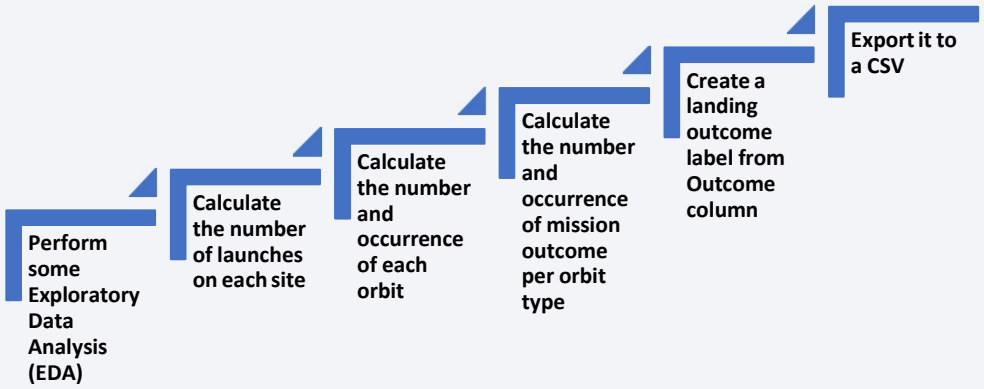
```
df =
 pd.DataFrame.from_dict(l
 aunch_dict)
```

8. Dataframe to .CSV

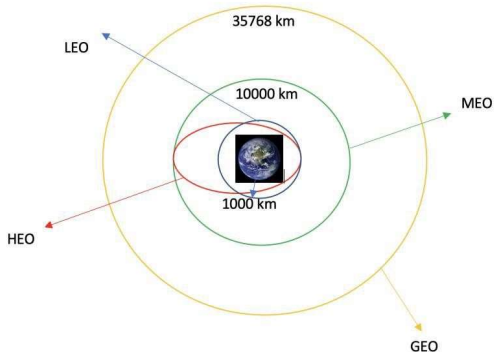
```
df.to_csv('spacex_web_scraped.c
 sv', index=False)
```

Data Wrangling

There are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.



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



GitHub Link

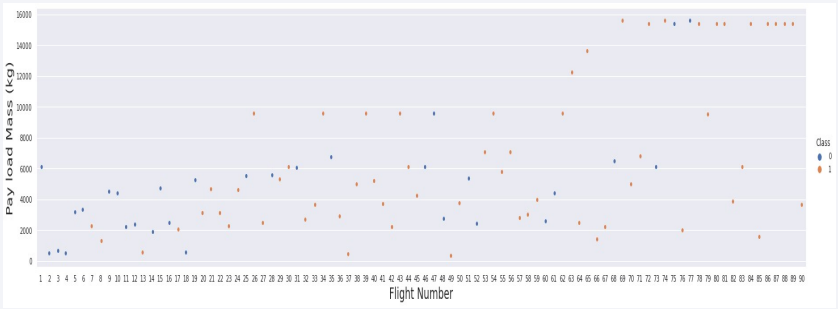
https://github.com/atharvajirafe/IBM_Data_Science/blob/Master/Data%20wrangling.ipynb

EDA with Data Visualization

Scatter Graphs

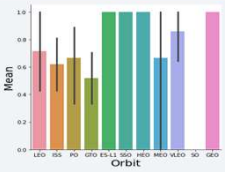
- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit Type
- Orbit VS. Payload Mass

Scatter plots illustrates how one variable effects another one and the correlation of the relationship between two variables. Scatter plots in general consist of large body of data.



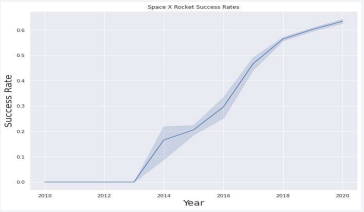
Bar Graph

To compare sets of data between coming from different groups. The bar graph represents categories on one axis and a discrete value in the other. It show the relationship between the two axes and it can show the changes in data over time.



Line Graph

Line graph illustrates data variables and trends. It helps to make a prediction about the result of data not yet recorded



GitHub link

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/ea10e202-4014-4052-a808-0e3342831aa5/view?access_token=f85f16b48f79457d15ae16acdb95461dc4824c8d4a6e11ad87d7e81a8edd5646

EDA with SQL

- **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 acheived.**
- **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. Use a subquery**
- **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

https://github.com/atharvajirafe/IBM_Data_Science/blob/Master/EDA%20with%20SQL%20lab.ipynb

Build an Interactive Map with Folium

1. **Visualizing the Launch Data into an interactive map**
   We used the Latitude and Longitude coordinates at each launch site and added a *Circle Marker around each launch site with a label of the name of the launch site.*
2. **Assigned the dataframe launch_outcomes**
   A success and a failure outcomes were assigned to classes 0 and 1 with Green for success and Red for failure markers on the map using Marker Cluster() function
3. **Calculating the distance**
   Haversine's formula was used to calculate the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns.
4. **Drawing Lines on the map**
   Lines were drawn to measure distance to landmarks

Below are some example of some trends in which the Launch Site is situated in.

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

https://github.com/atharvajirafe/IBM_Data_Science/blob/Master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- **Launch Site Drop-down Input Component**

  To see which one has the largest success count and then to select one specific site and check its detailed success rate (class=0 vs. class=1).

- **Callback function to render success-pie-chart based on selected site dropdown**

  Pie chart to visualizing launch success counts.

- **Range Slider to Select Payload**

  To find if variable payload is correlated to mission outcome and select different payload range and see if we can identify some visual patterns.

- **Callback function to render the success-payload-scatter-chart scatter plot**

  To visually observe how payload may be correlated with mission outcomes for selected site(s) and to color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters.

GitHub URL

https://github.com/usailky/Applied-Data-Science-Capstone/blob/master/Dashboarding.ipynb

Predictive Analysis (Classification)

BUILDING MODEL

- Load dataset into NumPy and Pandas
- Transform Data
- Split data into training and test data sets
- Check how many test samples available
- Decide which type of machine learning algorithms to be used
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

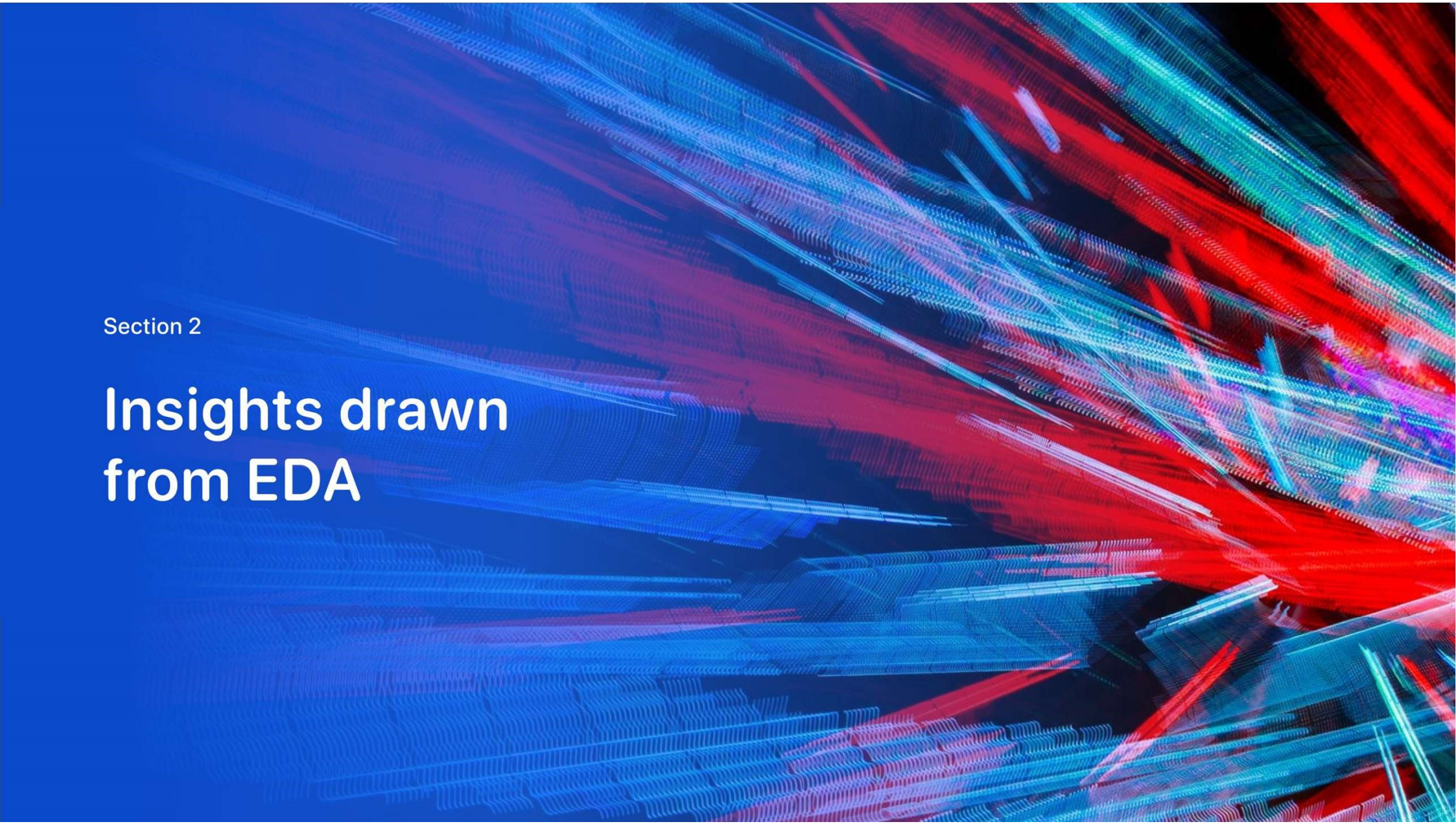
- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook

GitHub URL

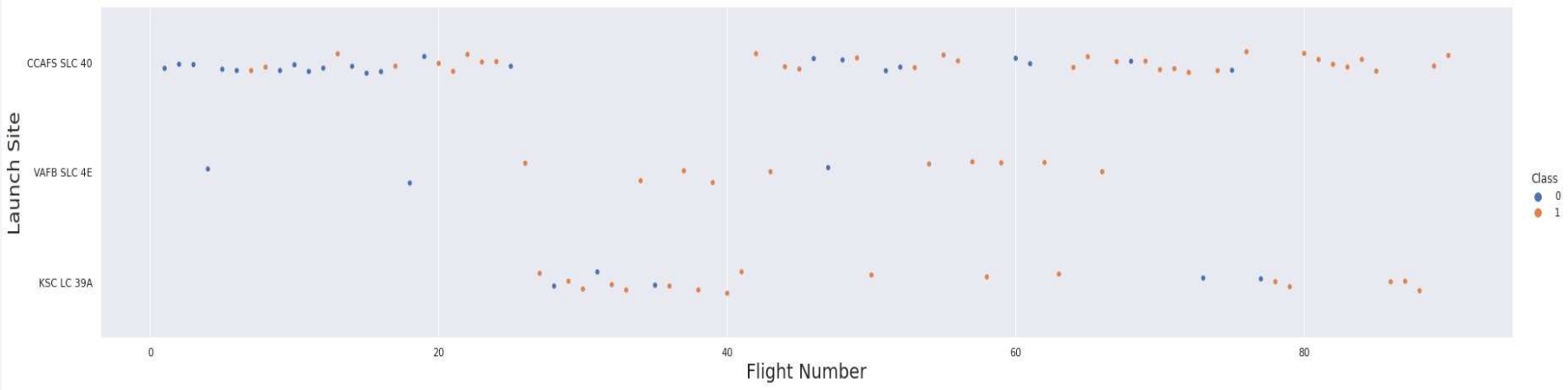
https://github.com/atharvajirafe/IBM_Data_Science/blob/Master/Machine%20Learning%20Prediction.ipynb

Results

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

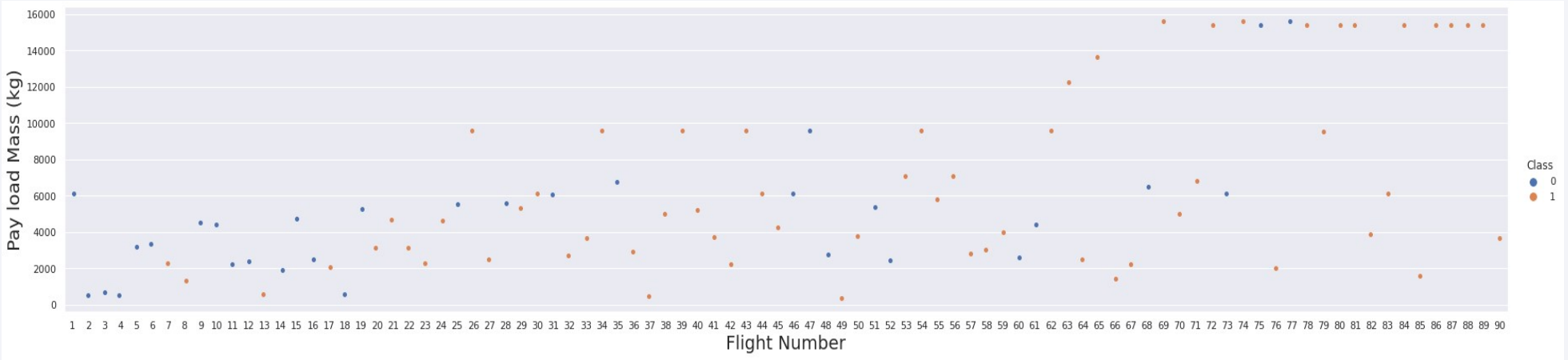

Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site



More number of flights at a launch site indicates a greater success rate at a launch site.

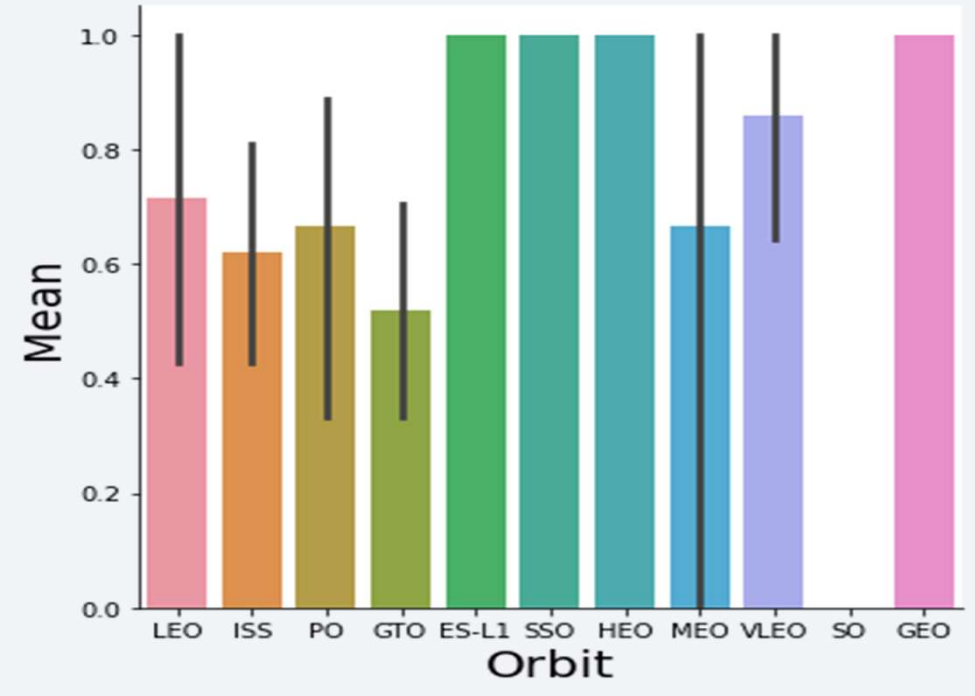
Payload vs. Launch Site



We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return. We see different launch sites have different success rates. For example CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77% . There is no clear pattern to be found with this visualization for making a decision whether the launch site is dependent on Pay Load Mass for a success launch or not.

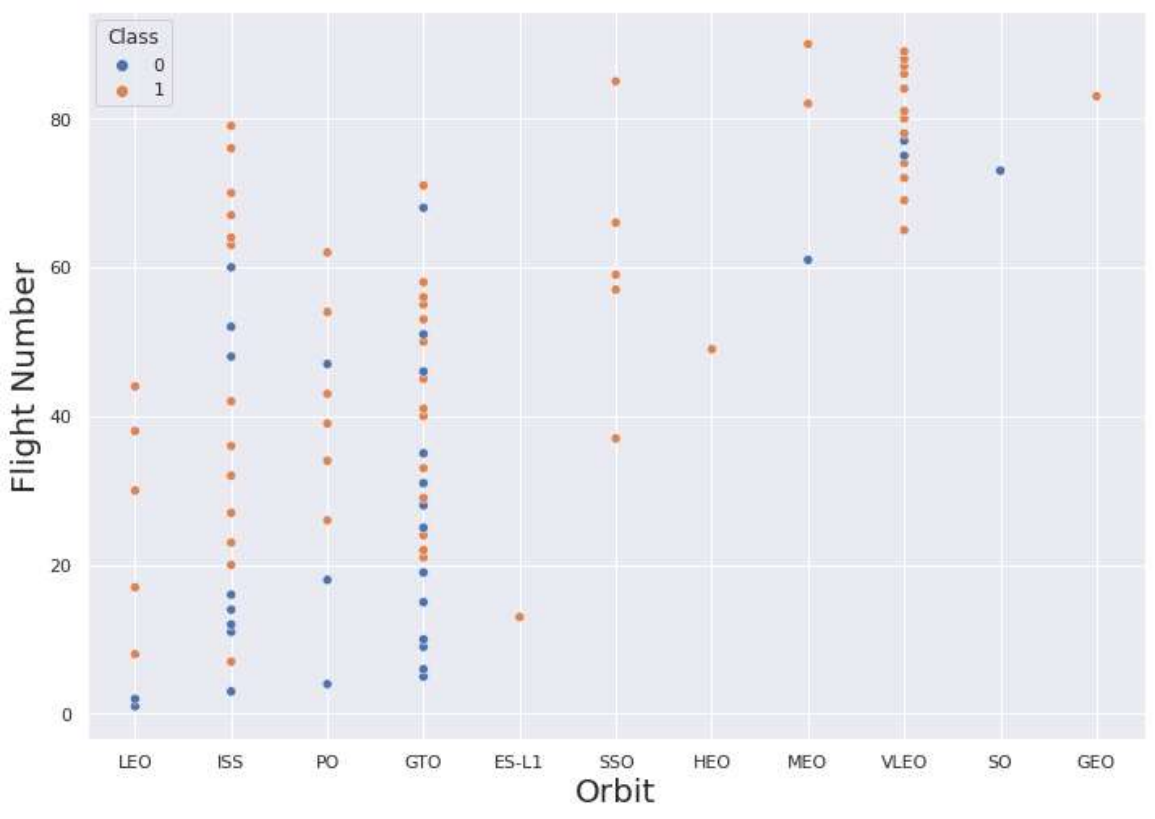
Success Rate vs. Orbit Type

Orbit ES-L1, SSO, HEO, and GEO have the highest Success Rate



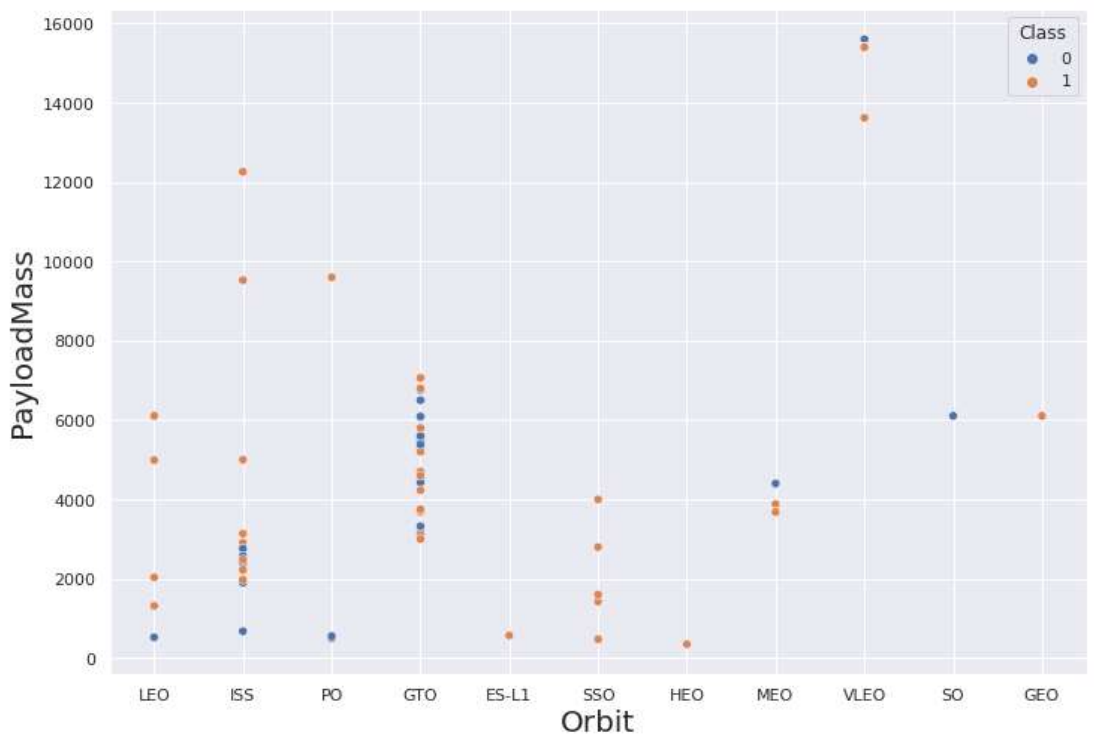
Flight Number vs. Orbit Type

In the 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.



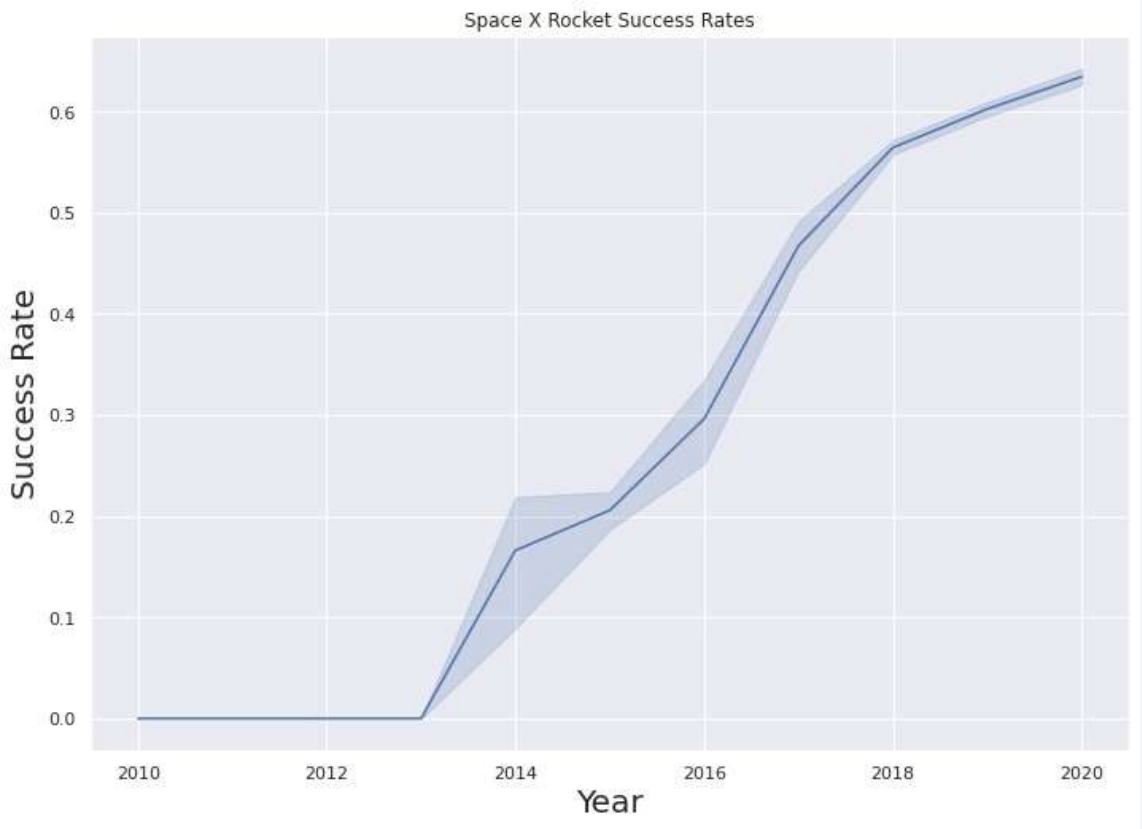
Payload vs. Orbit Type

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

You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

Select DISTINCT launch_site from Spacex

Using the word *DISTINCT* in the query shows only unique values in the *launch_site* column from *SpaceX table*

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

Launch Site Names Begin with 'CCA'

*SELECT * FROM Spacex where launch_site LIKE 'CCA%' FETCH NEXT 5 ROWS ONLY*

Using the word *TOP 5* in the query means that it will only show 5 records from *SpaceX table* and *LIKE* keyword has a wild card
with the words *'CCA%'* the percentage in the end suggests that the Launch_Site name must start with KSC.

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

Total Payload Mass

Select SUM(payload_mass__kg_) as total_payload_mass from Spacex where customer = 'NASA (CRS)'

Using the function SUM summates the total in the column PAYLOAD_MASS_KG_. The WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS).

total_payload_mass
45596

Average Payload Mass by F9 v1.1

Select AVG(payload_mass__kg_) as Average_Payload_Mass from Spacex where booster_version ='F9 v1.1'

Using the function *AVG* calculates the average in the column *PAYLOAD_MASS_KG_*. The *WHERE* clause filters the dataset to only perform calculations on *Booster_version F9 v1.1.*

average_payload_mass
2928

First Successful Ground Landing Date

Select MIN(Date) successful_landing_outcome from Spacex where landing__outcome = 'Success'

Using the function MIN calculates the minimum date in the column Date
The WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success

successful_landing_outcome
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Select booster_version from Spacex where landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000

Selecting only *Booster_Version and t*he *WHERE* clause filters the dataset to *Landing_Outcome = Success (drone ship).* The *AND* clause specifies additional filter conditions *Payload_MASS_KG_*>4000AND*Payload_MASS_KG_<6000*

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

Total Number of Successful and Failure Mission Outcomes

Select count(mission_outcome) As Succes_Mission_Outcome ,(select count(mission_outcome) from SPACEX where mission_outcome LIKE '%Failure%') AS Failed_Mission_Outcome from SPACEX where mission_outcome LIKE '%Success%';

List the total number of successful and failure mission outcomes using subqueries.

succes_mission_outcome	failed_mission_outcome
100	1

Boosters Carried Maximum Payload

SELECT DISTINCT booster_version, MAX(payload_mass__kg_) as Maximum_Payload_Mass from Spacex GROUP BY booster_version ORDER BY Maximum_Payload_Mass DESC

DISTINCT in the query shows only unique values in the Booster_Version column from SpaceX table and GROUP BY groups the list in order set to a certain condition. DESC was used to arranging the dataset into descending order

booster_version	maximum_payload_mass
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600

2015 Launch Records

Select booster_version, launch_site, landing__outcome from spacex where YEAR(DATE)='2015' AND landing__outcome = 'Failure (drone ship)'

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

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

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

SELECT 'Success (ground pad)' as Landing_Outcome, COUNT(landing__outcome) AS Outcomes_Between_2010_06_04_and_2017_03_20 from spacex WHERE ((DATE) >'2010-06-04') AND ((DATE) < '2017-03-20') and landing__outcome = 'Success (ground pad)' UNION SELECT 'Failure (drone ship)', COUNT(landing__outcome) from spacex WHERE ((DATE) >'2010-06-04') AND ((DATE) < '2017-03-20') and landing__outcome = 'Failure (drone ship)'

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. The Function COUNT counts records in column WHERE filters data (wildcard) and two (conditions)

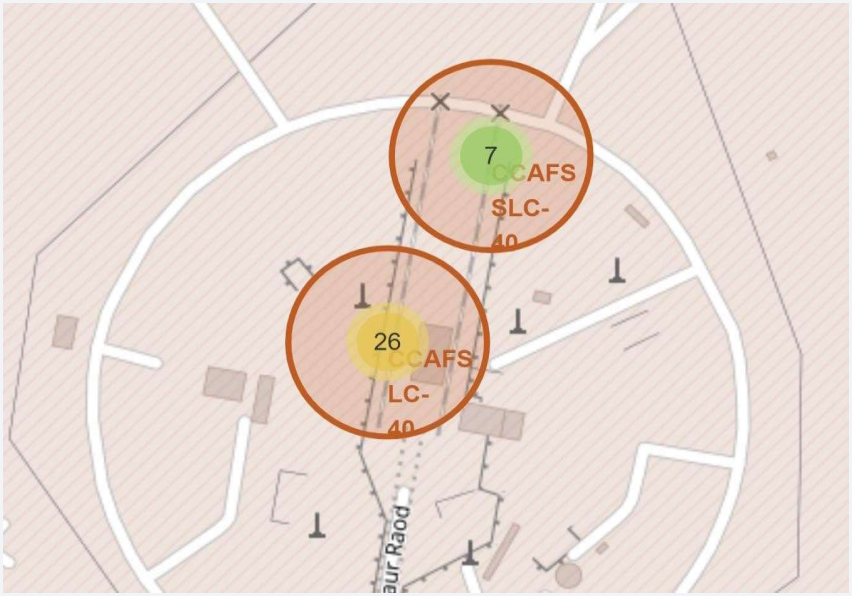
landing_outcome	outcomes_between_2010_06_04_and_2017_03_20
Success (ground pad)	3
Failure (drone ship)	5


Section 4
Launch Sites
Proximities Analysis

All launch sites on a global map

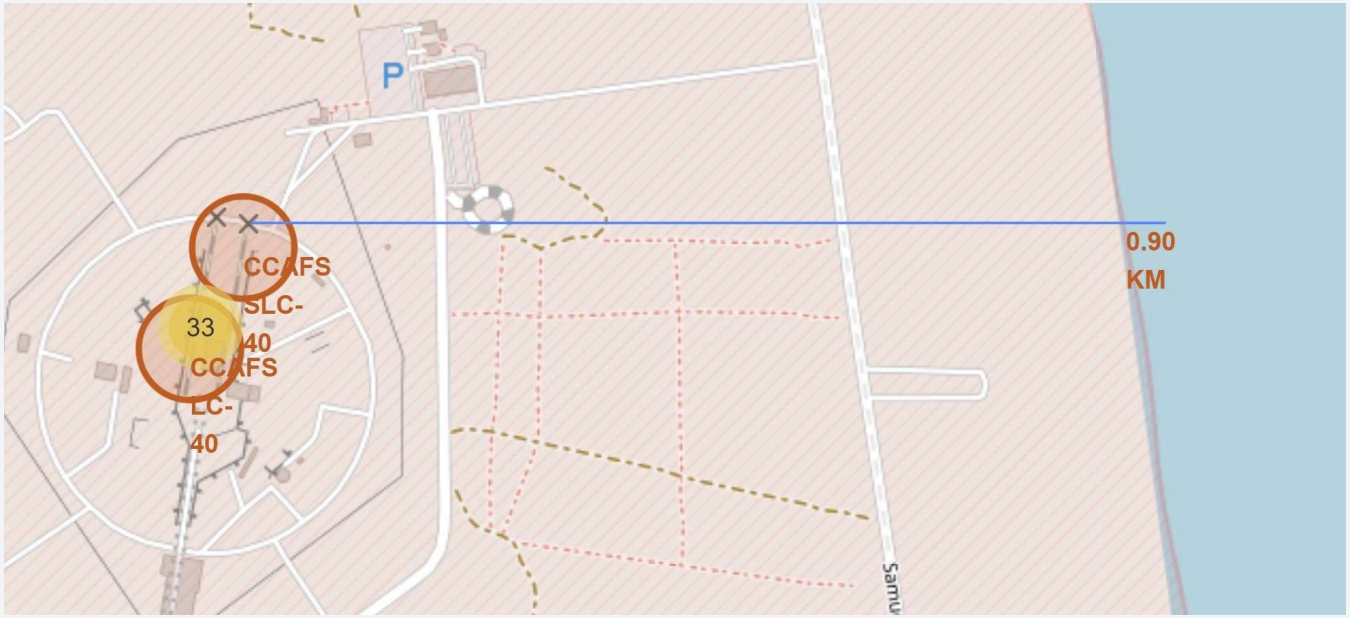
SpaceX launch sites are in the United States of America coasts Florida and California.

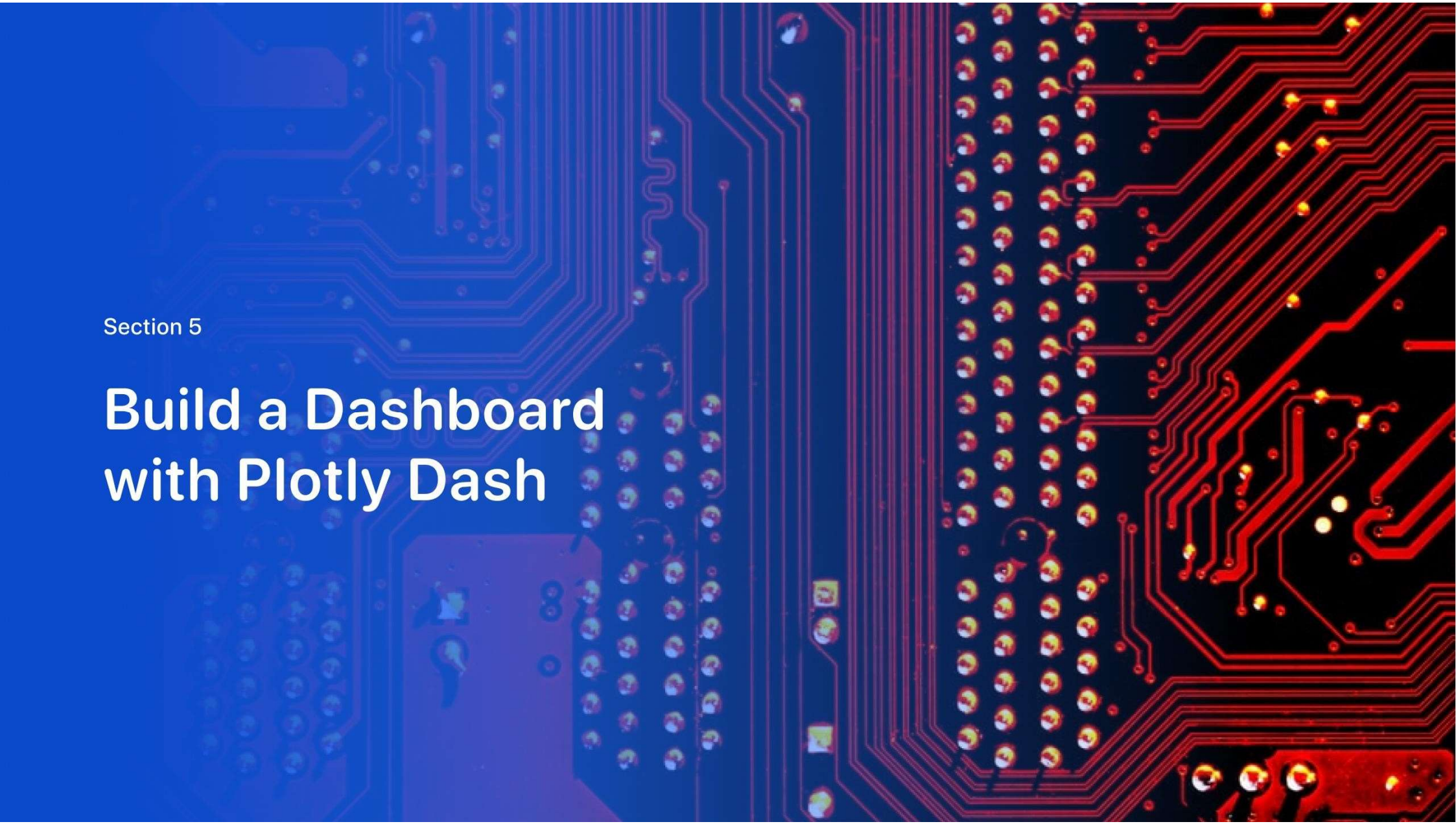
Colour Labelled Markers



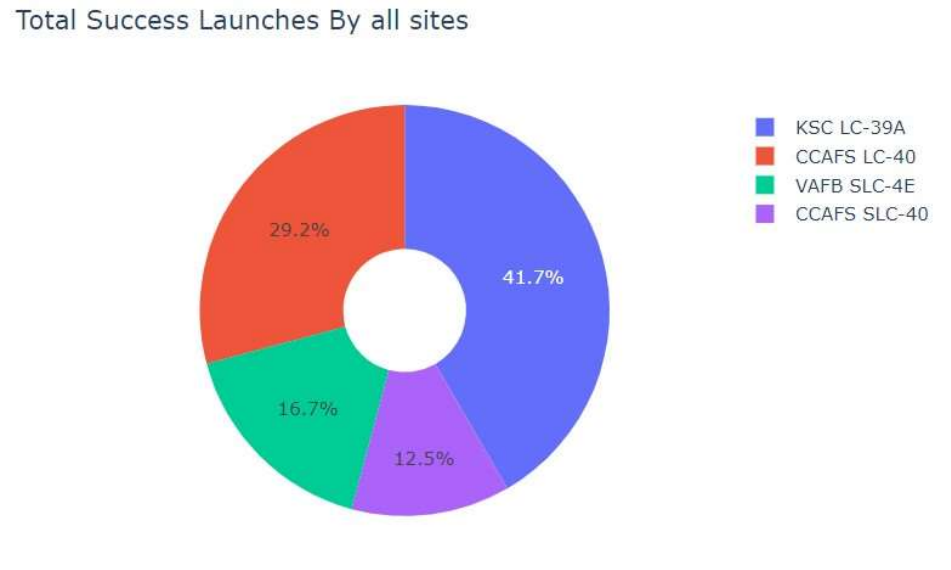
Green Marker shows successful Launches and Red Marker shows Failures

Launch Sites distance to landmarks to find trends



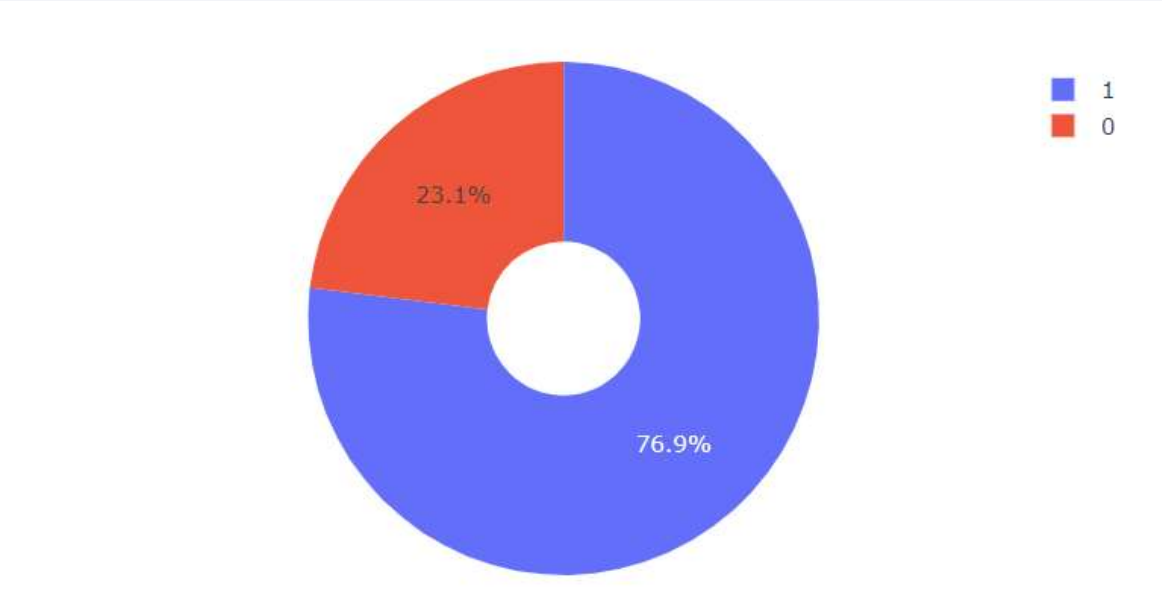

Section 5
Build a Dashboard with Plotly Dash

DASHBOARD-Pie chart showing the success percentage



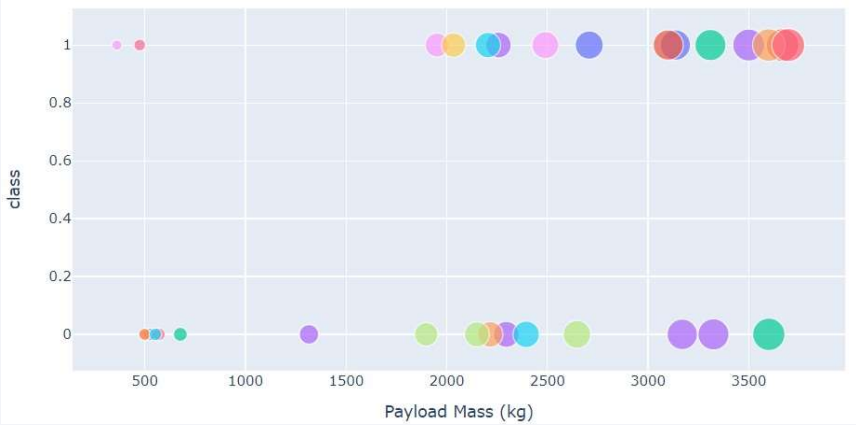
The site KSC LC 39A had the most successful launches from all the sites

DASHBOARD-Pie chart for the launch site with highest launch success ratio

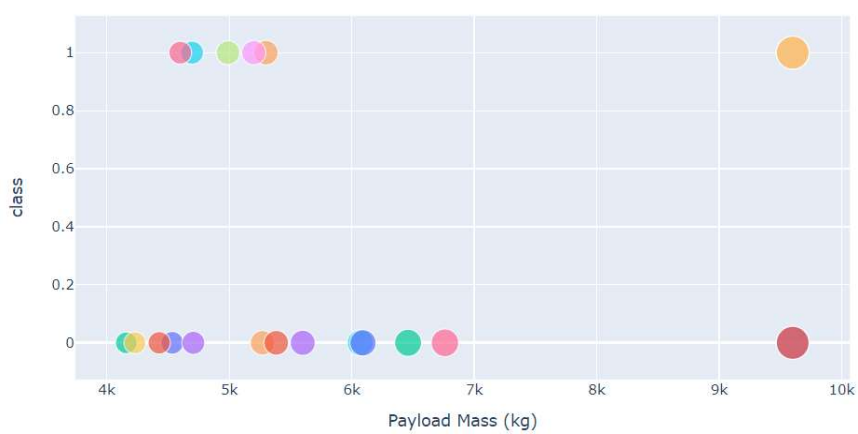


KSC LC 39A has 76.9% success rate while getting 23.1% failure rate

DASHBOARD–Payload vs. Launch Outcome scatter plot for all sites with different payload selected in the range slider



Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg

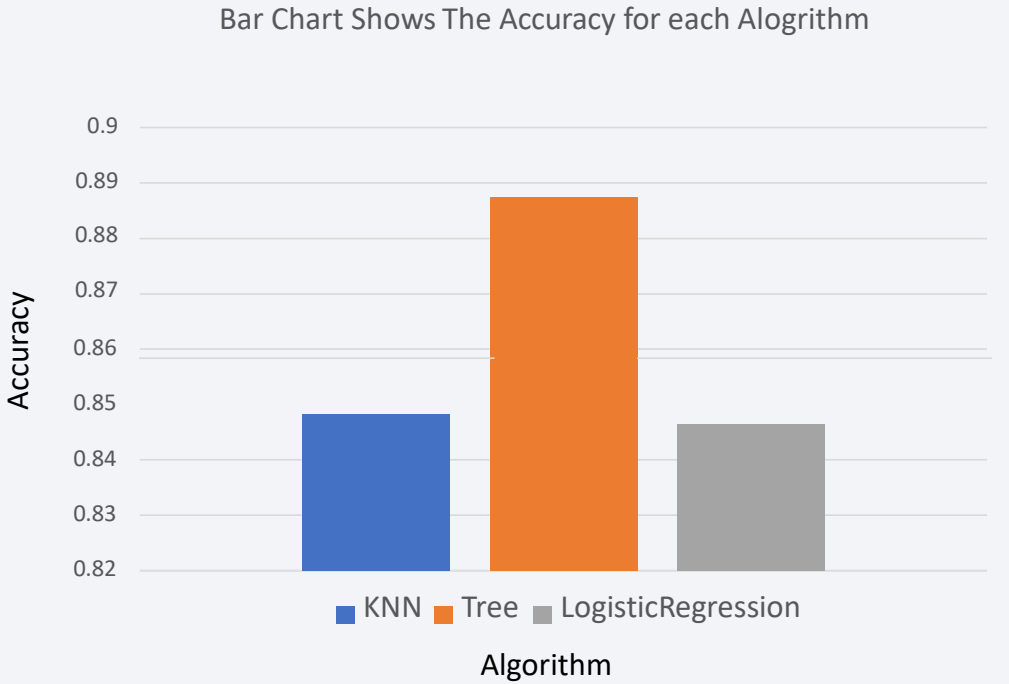
The success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)

Classification Accuracy

Algorith	Accuracy
KNN	0.848214286
Tree	0.887499999
LogisticRegression	0.846428571



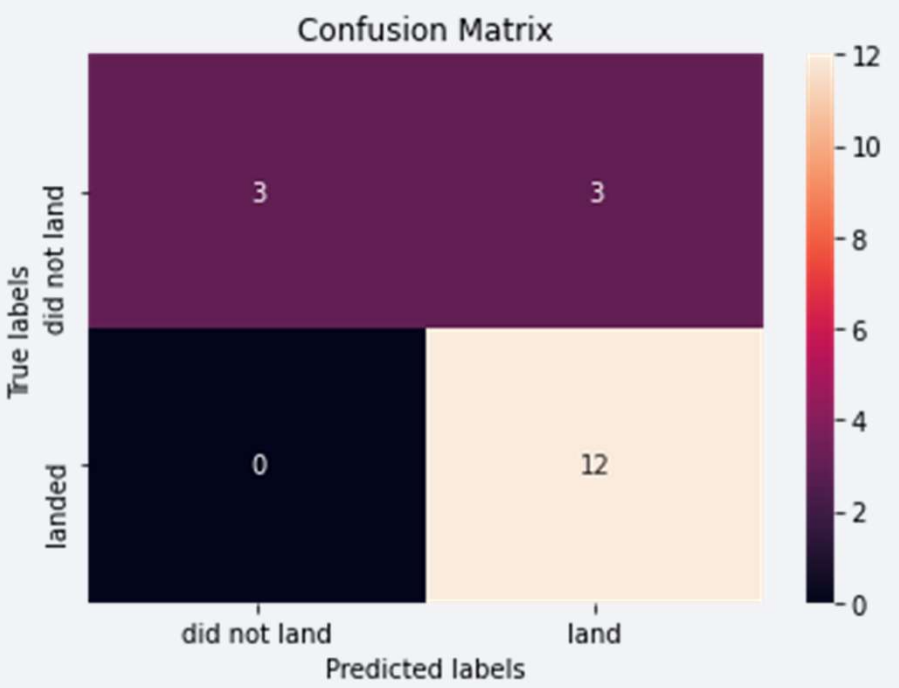
The accuracy is extremely close Tree algorithm has the highest accuracy.

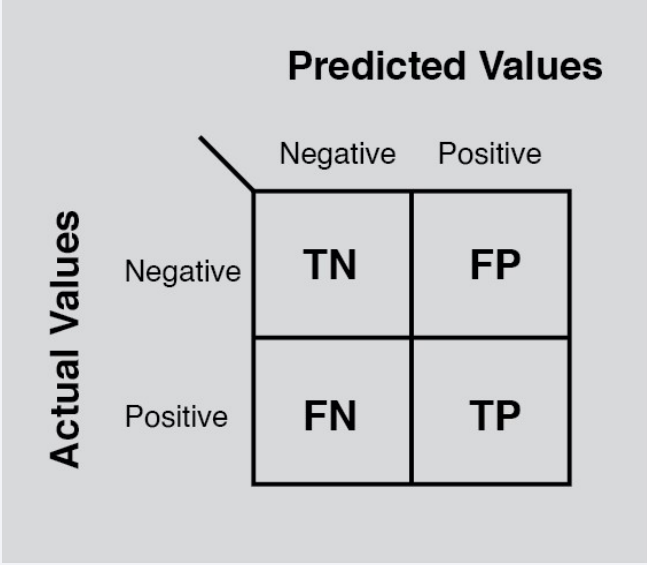
bestalgorithm = max(algorithms, key=algorithms.get)

Best Algorithm is Tree with a score of 0.8874999999999998
Best Params is : {'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}

Confusion Matrix

Examining the confusion matrix, Tree algorithm can distinguish between the different classes.





Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is proportional time
- KSC LC-39A had the most successful launches from all the sites
- Orbit ES-L1, SSO, HEO, GEO have the best Success Rate

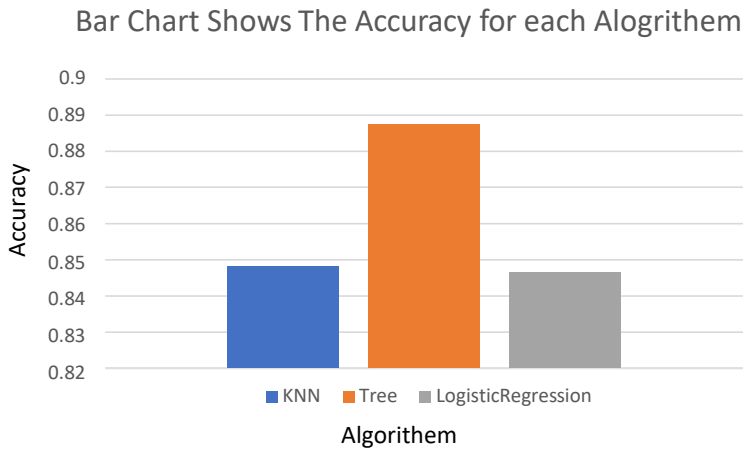
Appendix

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

Microsoft Excel

I have used Microsoft Excel to represent the accuracy of the algorithm by Bar chart using table that has each algorithm with its accuracy value

Algorith	Accuracy
KNN	0.848214286
Tree	0.887499999
LogisticRegression	0.846428571



Thank you!