IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sapna Yadav
24-11-2021

Outline

- **Executive Summary**
- **Introduction**
- **Methodology**
- **Results**
- **Conclusion**

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

We predicted 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 influences if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
- What conditions does SpaceX have to achieve to get the best results and ensure the best rocket 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 data fields for Machine Learning and dropping irrelevant columns

Perform exploratory data analysis (EDA) using visualization and SQL

Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.

Perform predictive analysis using classification models

Build, tune, evaluate classification models

Data Collection

The following datasets was collected by

- We worked with SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.

Data Collection – SpaceX API

1.

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

2.

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

3.

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

```
getBoosterVersion(data)
```

4.

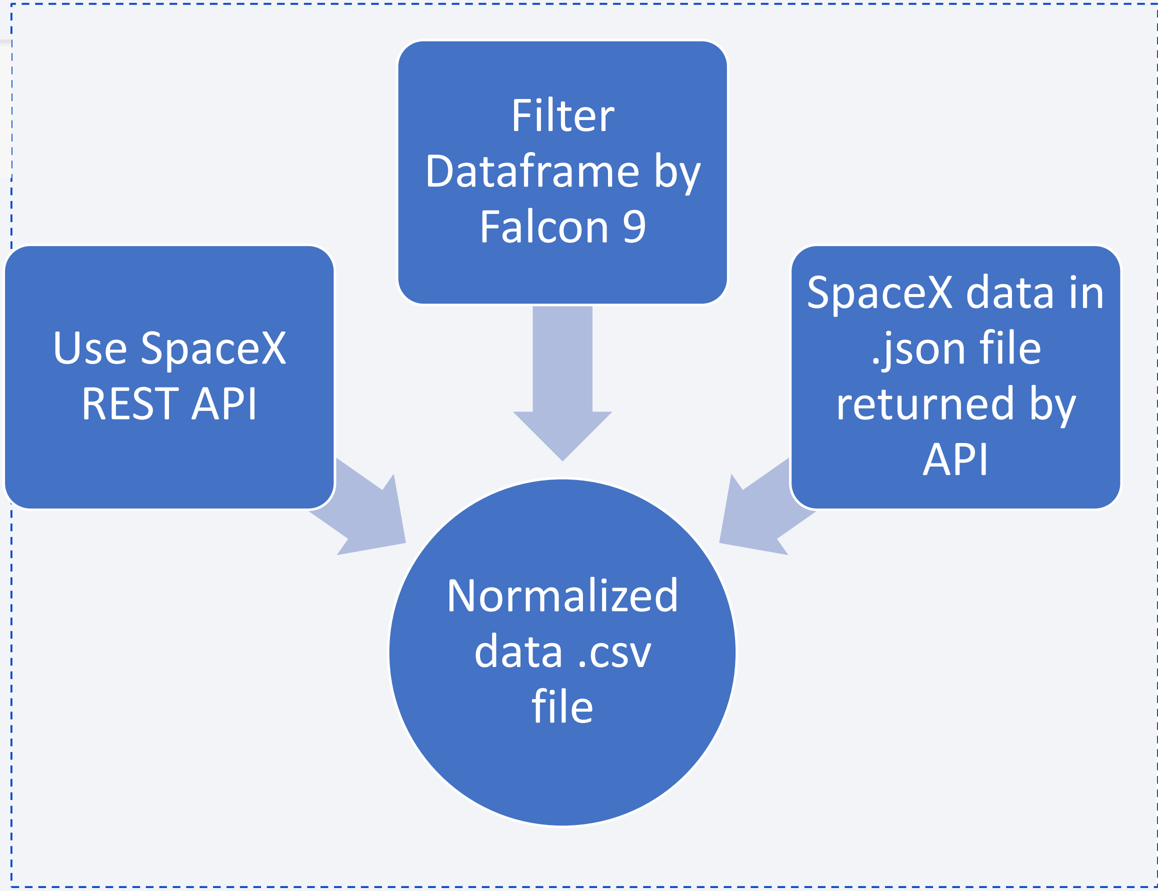
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

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

5.

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

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



Github URL

Data Collection - Scraping

1.
```
page = requests.get(static_url)
```

2.
```
soup = BeautifulSoup(page.text, 'html.parser')
```

3.
```
html_tables = soup.find_all('table')
```

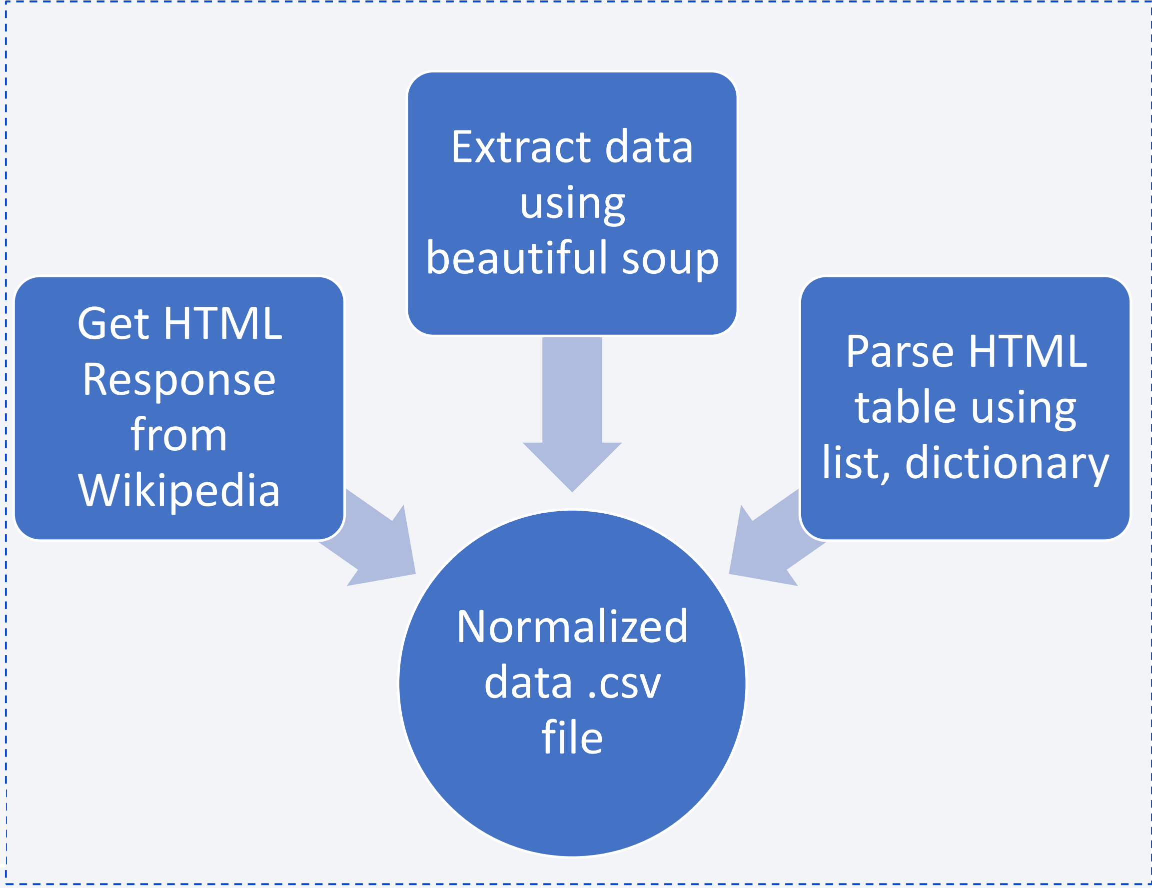
4.
```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x]
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5.
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
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
```
extracted_row = 0
#Extract each table
for table_number,table in enumera...e
   # get table row
    for rows in table.find_all("tr"
        #check to see if first tabl
```

7
```
df = pd.DataFrame.from_dict(launch_dict)
```

8
```
df.to_csv('spacex_web_scraped.csv', index=False)
```



GitHub URL

Data Wrangling

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

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

Create a landing outcome label from Outcome column

Work out success rate for every landing in dataset

GitHub URL

EDA with Data Visualization

Scatter Graphs being drawn:

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

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data

Mean VS. Orbit:

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

Line Graph being drawn:

Success Rate VS. Year Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

GitHub URL

EDA with SQL

Performed SQL queries to gather information about the dataset. For example of some questions we were asked about the data we needed information about. Which we are using SQL queries to get the answers in the dataset :

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- 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 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.

GitHub URL

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map. We took 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.

We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()

Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks 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

Explain why you added those objects

GitHub URL

Build a Dashboard with Plotly Dash

Used Python Anywhere to host the website live 24/7 so your can play around with the data and view the data.

The dashboard is built with Flask and Dash web framework.

Graphs

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

• display relative proportions of multiple classes of data.

• size of the circle can be made proportional to the total quantity it represents.

- Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

• It shows the relationship between two variables.

• It is the best method to show you a non-linear pattern.

• The range of data flow, i.e. maximum and minimum value, can be determined.

• Observation and reading are straightforward.

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- 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

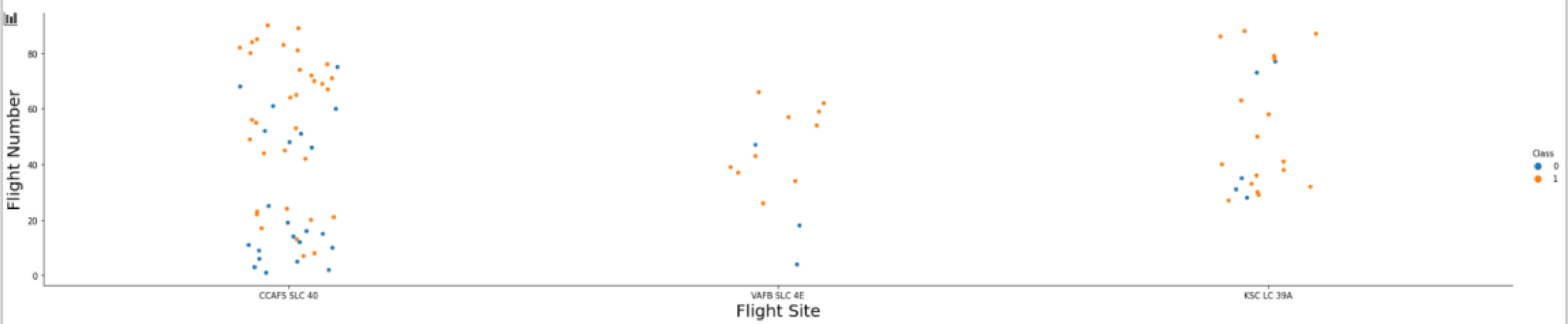
Results

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

Section 2

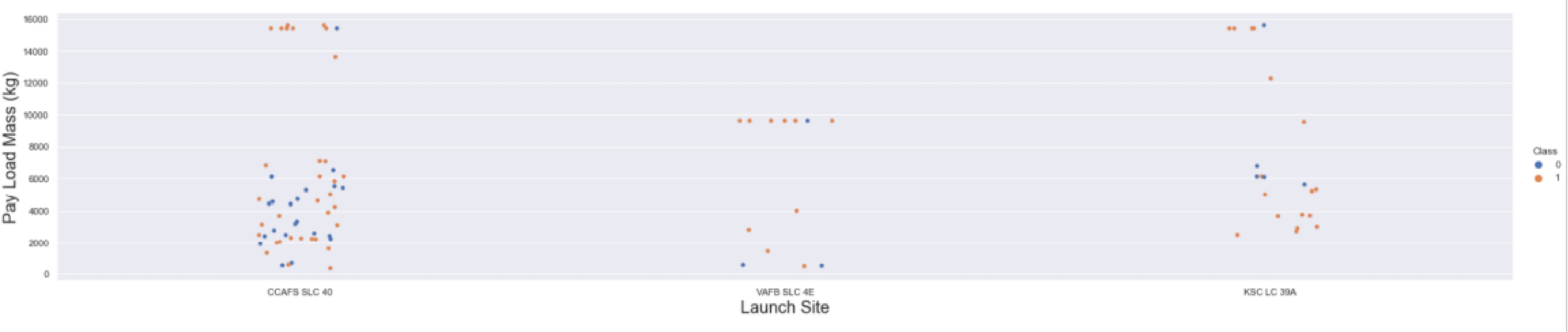
Insights drawn from EDA

Flight Number vs. Launch Site



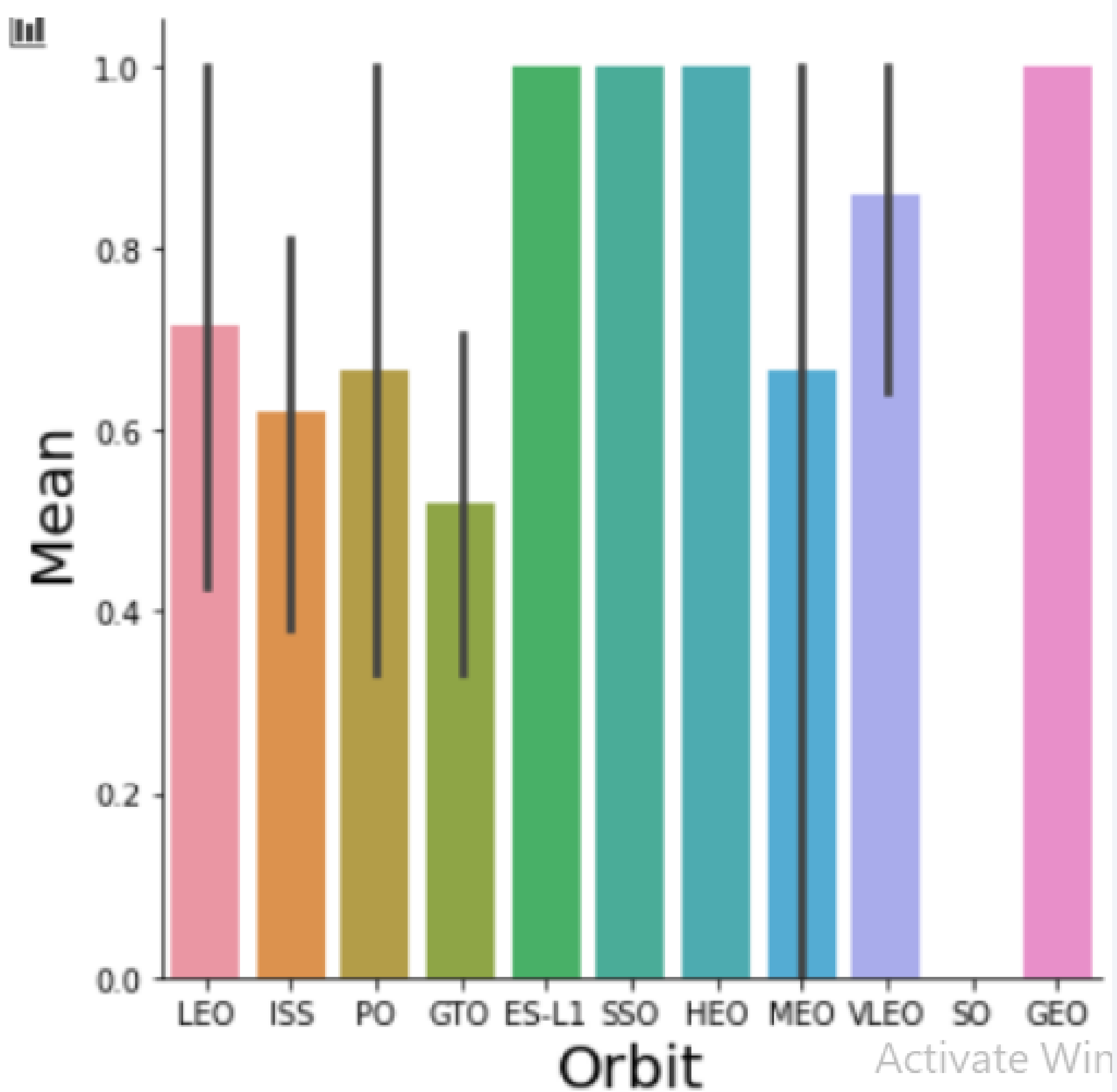
Success rate is greater as the amount of flights at a launch site increase

Payload vs. Launch Site



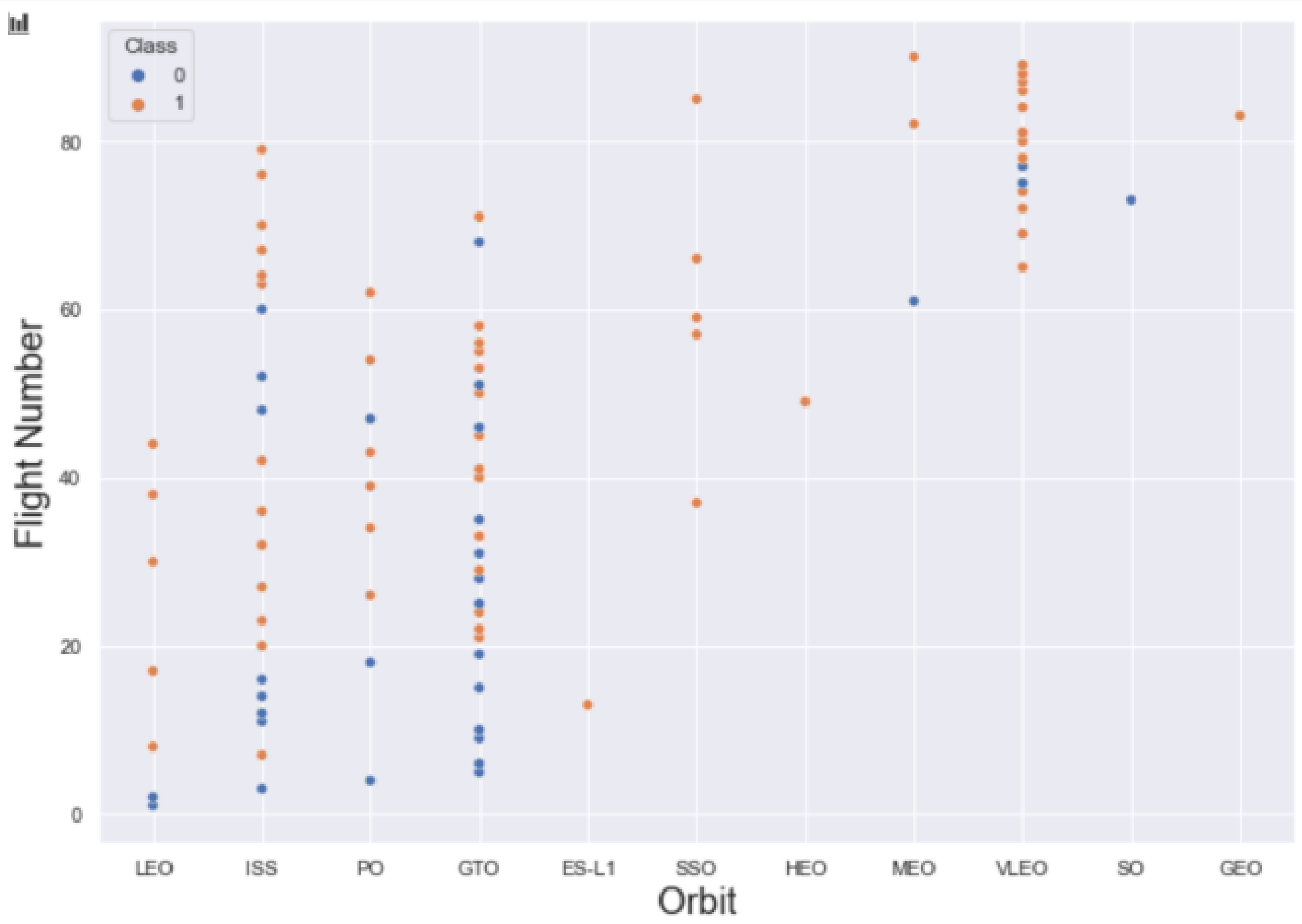
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependent on Pay Load Mass for a success launch.

Success Rate vs. Orbit Type



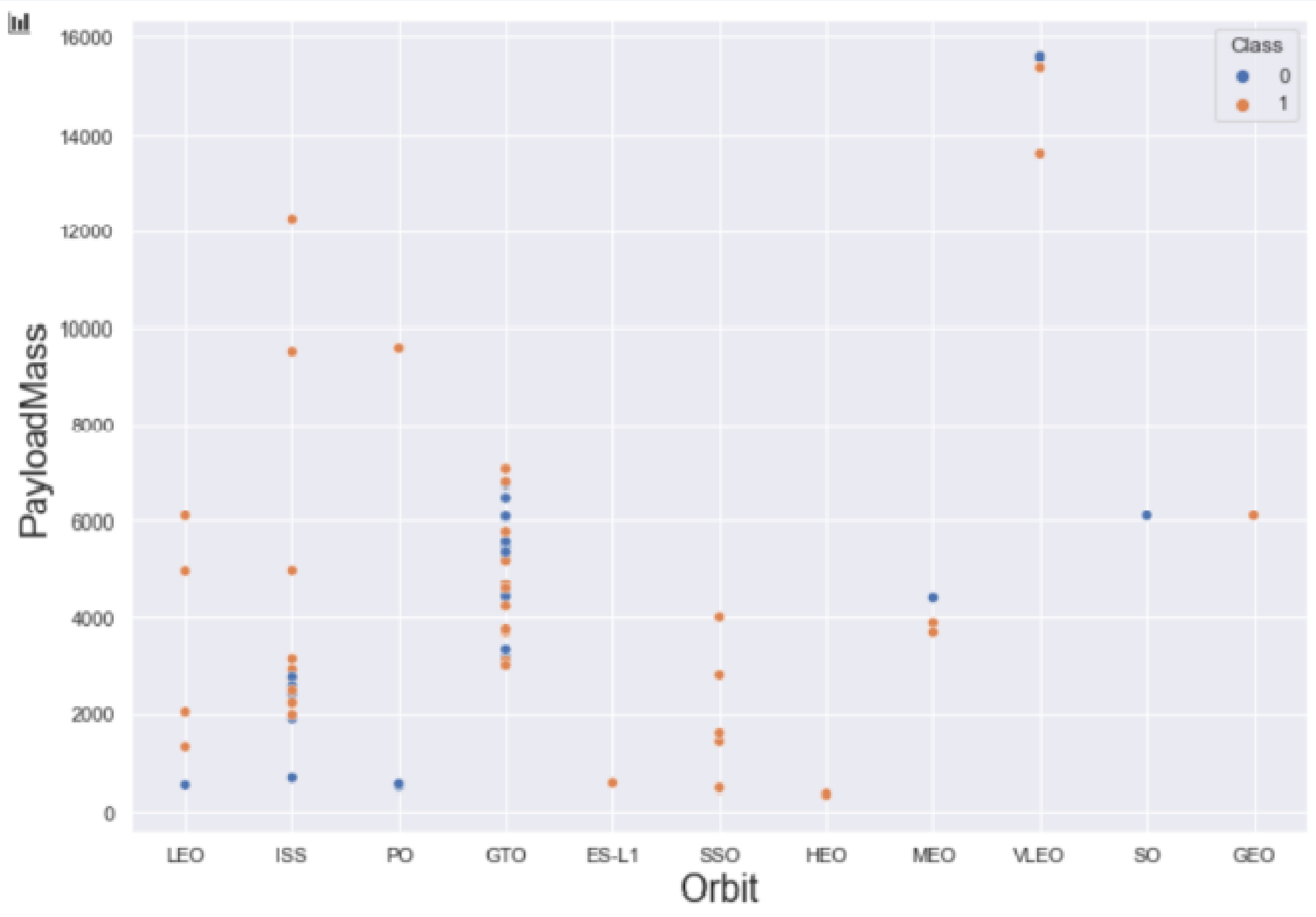
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Flight Number vs. Orbit Type



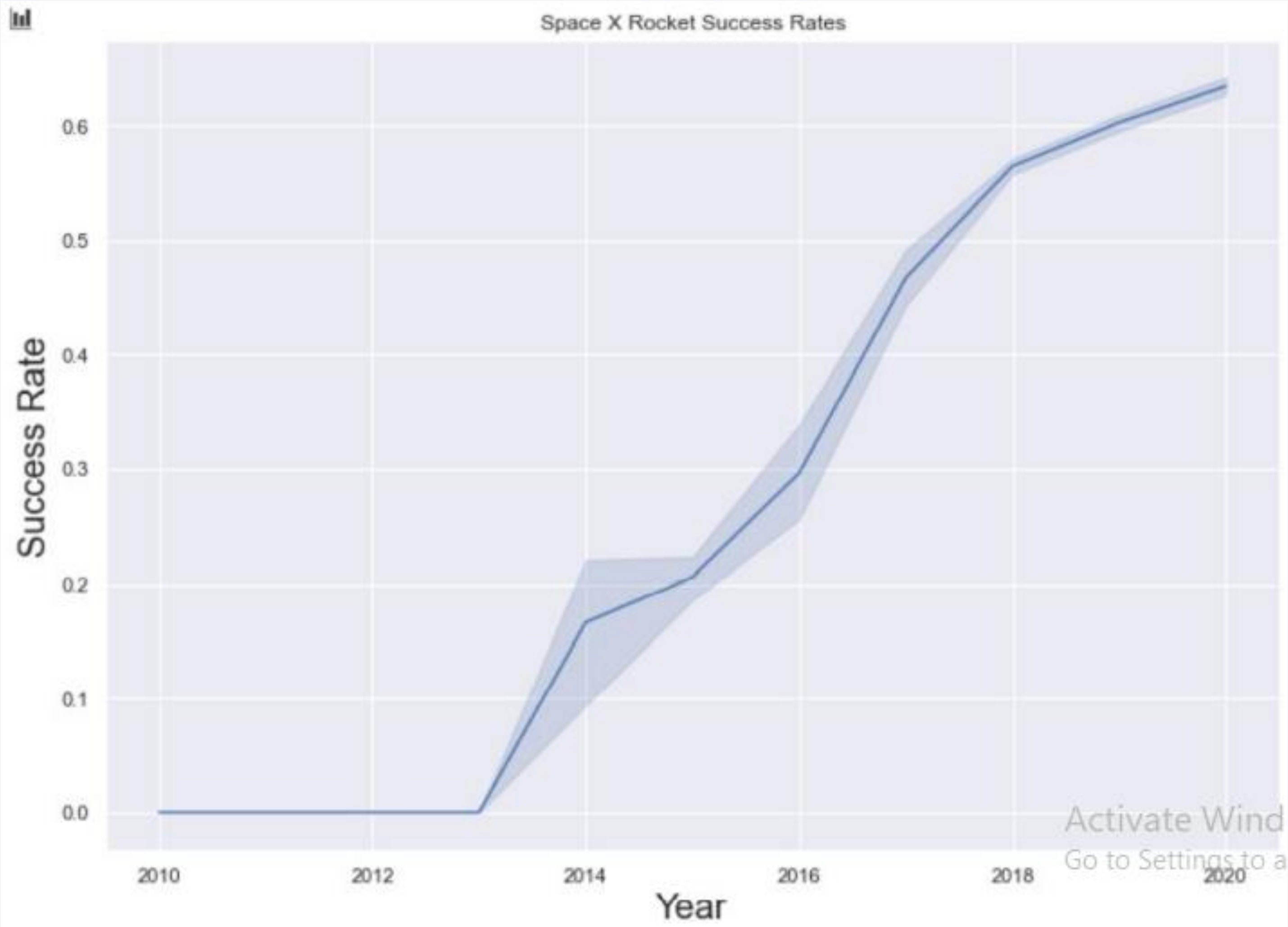
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type



Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020

All Launch Site Names

Unique launch sites

Query:

select DISTINCT Launch_Site from tblSpaceX

CCAFS LC-40

CCAFS SLC-40

CCAFS SLC-40

KSC LC-39A

VAFB SLC-4E

QUERY EXPLAINATION

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

Launch Site Names Begin with 'CCA'

Query:

select TOP 5 * from tblSpaceX WHERE Launch_Site LIKE ‘CCA%'

QUERY EXPLAINATION

Using the word TOP 5 in the query means that it will only show 5 records from tblSpaceX 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 CCA.

Total Payload Mass

Query:

select SUM(PAYLOAD_MASS_KG_) TotalPayloadMass from tblSpaceX where Customer = 'NASA (CRS)'",'TotalPayloadMass

Query Result

Total Payload Mass = 45596

QUERY EXPLAINATION

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)

Average Payload Mass by F9 v1.1

Query:

select AVG(PAYLOAD_MASS_KG_) AveragePayloadMass from tblSpaceX where Booster_Version = 'F9 v1.1'

Average Payload Mass = 2928

QUERY EXPLAINATION

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

First Successful Ground Landing Date

Query:

select MIN(Date) SLO from tblSpaceX where Landing_Outcome = ”Success (drone ship)”

First successful ground landing date = 06-05-2016

QUERY EXPLAINATION

Using the function MIN works out the minimum date in the column Date The WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

Successful Drone Ship Landing with Payload between 4000 and 6000

Query:

select Booster_Version from tblSpaceX where Landing_Outcome = 'Success (ground pad)’ AND Payload_MASS_KG_ > 4000 AND Payload_MASS_KG_ < 6000

Query Result:

F9 FT B1032.1

F9 B4 B1040.1

F9 B4 B1043.1

QUERY EXPLAINATION

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

Total Number of Successful and Failure Mission Outcomes

Query:

SELECT(SELECT Count(Mission_Outcome) from tblSpaceX where Mission_Outcome LIKE '%Success%') as Successful_Mission_Outcomes, (SELECT Count(Mission_Outcome) from tblSpaceX where Mission_Outcome LIKE '%Failure%') as Failure_Mission_Coutcomes

Query Result:

Successful_Mission_Outcomes = 100

Failure_Mission_Coutcomes = 1

QUERY EXPLAINATION

We used subqueries here to produce the results. The LIKE '%foo%' wildcard shows that in the record the foo phrase is in any part of the string in the records for example. PHRASE "(Drone Ship was a Success)" LIKE '%Success%' Word 'Success' is in the phrase the filter will include it in the dataset

Boosters Carried Maximum Payload

Query:

SELECT DISTINCT Booster_Version, MAX(PAYLOAD_MASS _KG_) AS [Maximum Payload Mass] FROM tblSpaceX GROUP BY Booster_Version ORDER BY [Maximum Payload Mass] DESC

QUERY EXPLAINATION

Using the word DISTINCT in the query means that it will only show Unique values in the Booster_Version column from tblSpaceX GROUP BY puts the list in order set to a certain condition. DESC means its arranging the dataset into descending order

2015 Launch Records

Query:

SELECT DATENAME(month, DATEADD(month, MONTH(CONVERT(date, Date, 105)), 0) - 1) AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM tblSpaceX WHERE (Landing_Outcome LIKE N'%Success%') AND (YEAR(CONVERT(date, Date, 105)) = '2015')

QUERY EXPLAINATION

Date fields in SQL Server stored as NVARCHAR the MONTH function returns name month. The function CONVERT converts NVARCHAR to Date. WHERE clause filters Year to be 2015

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

Query:

SELECT COUNT(Landing_Outcome) FROM tblSpaceX WHERE (Landing_Outcome LIKE '%Success%’) AND (Date > '04-06-2010’) AND (Date < '20-03-2017')

Query Result:

Landing Outcomes Between 2010-06-04 and 2017-03-20 = 34

QUERY EXPLAINATION

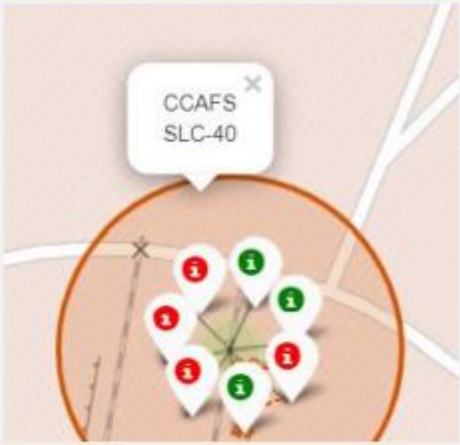
Function COUNT counts records in column WHERE filters data LIKE (wildcard) AND (conditions) AND (conditions)

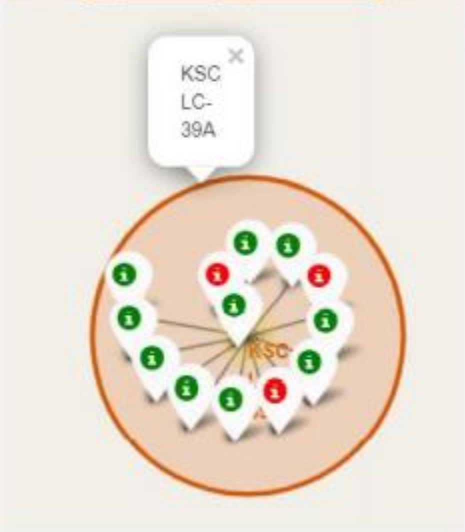
Section 4

Launch Sites Proximities Analysis

Launch sites global map markers

Colour Labelled Launch Outcomes



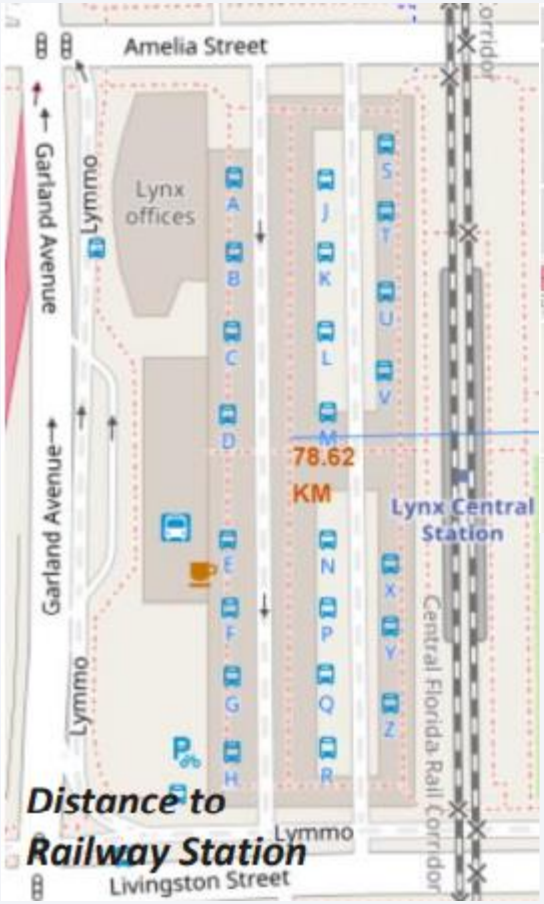


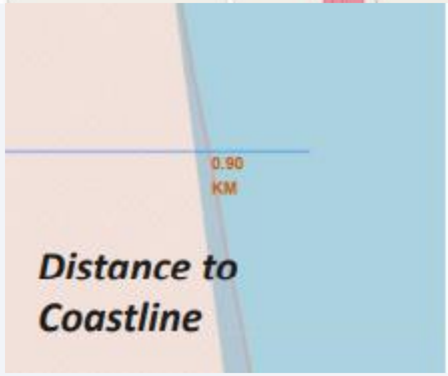
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Analyze Launch Sites distance to landmarks



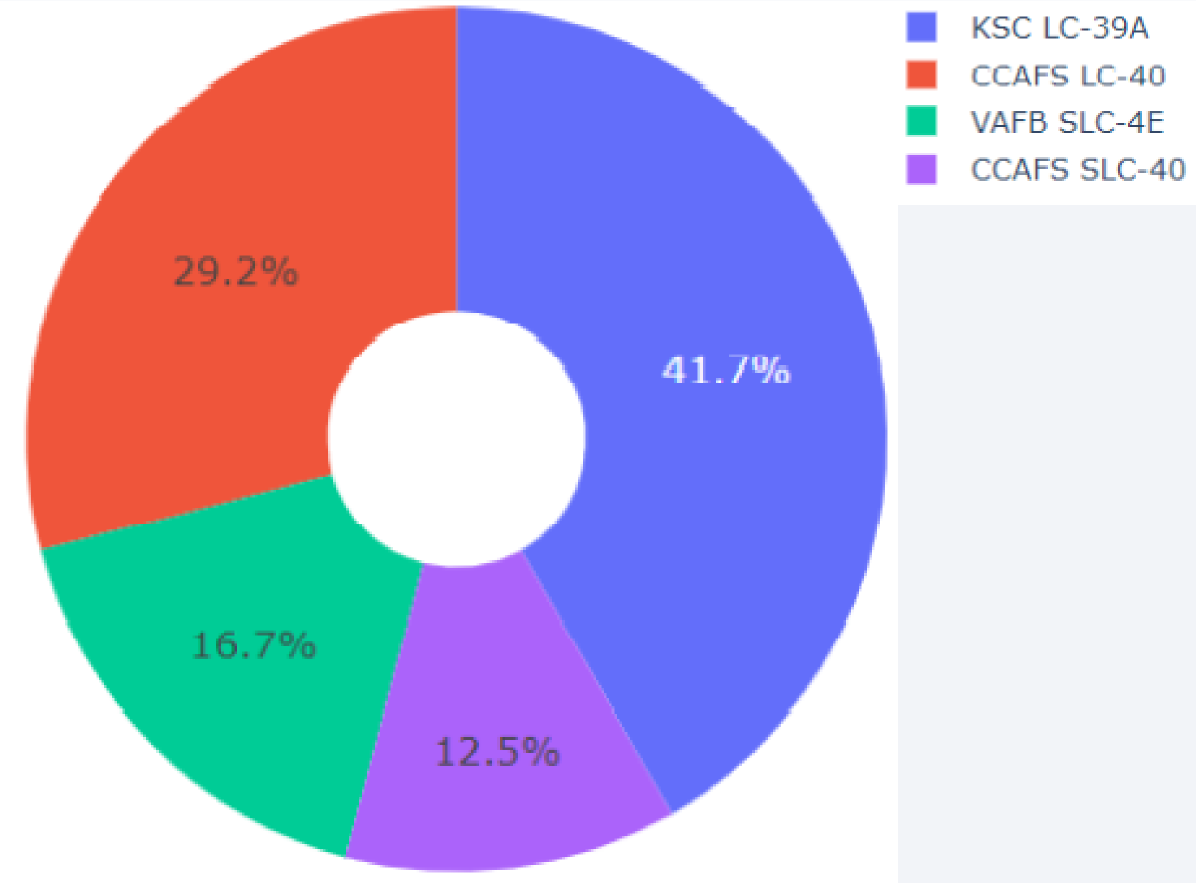


- Launch sites in close proximity to railways - No
- Launch sites in close proximity to highways - No
- Launch sites in close proximity to coastline - Yes
- Launch sites keep certain distance away from cities - Yes

Section 5

Build a Dashboard with Plotly Dash

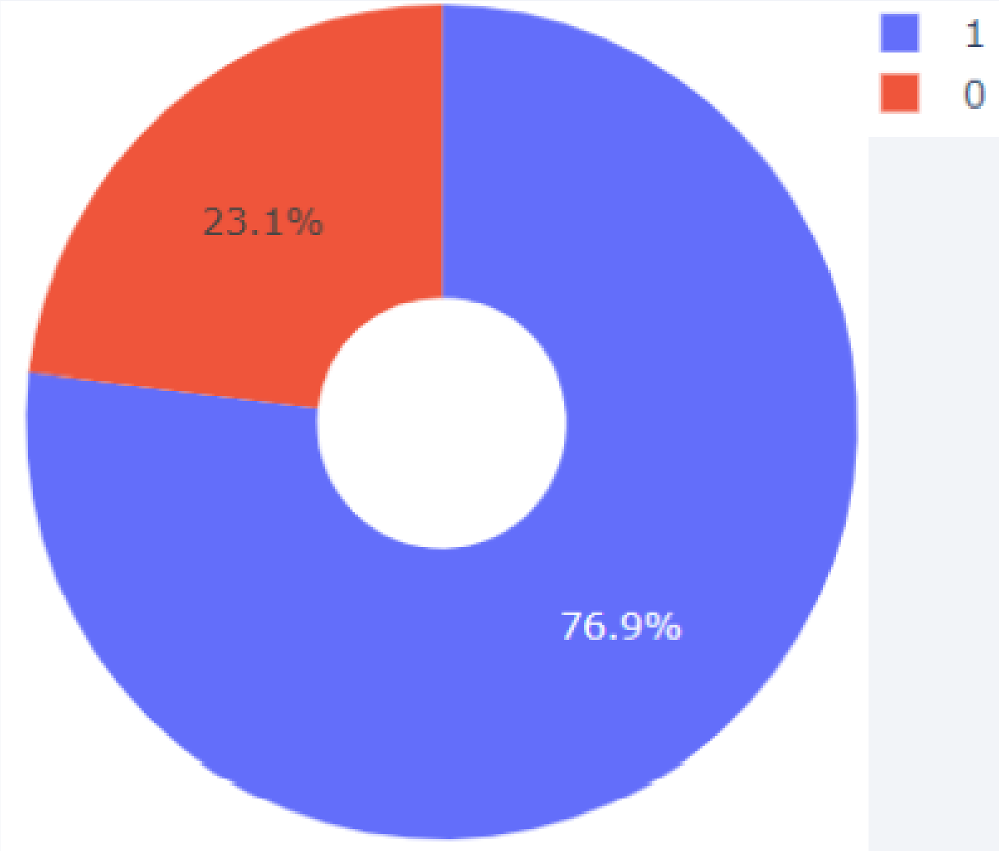
DASHBOARD – Pie chart showing the success percentage achieved by each launch site



Total success launches by all sites

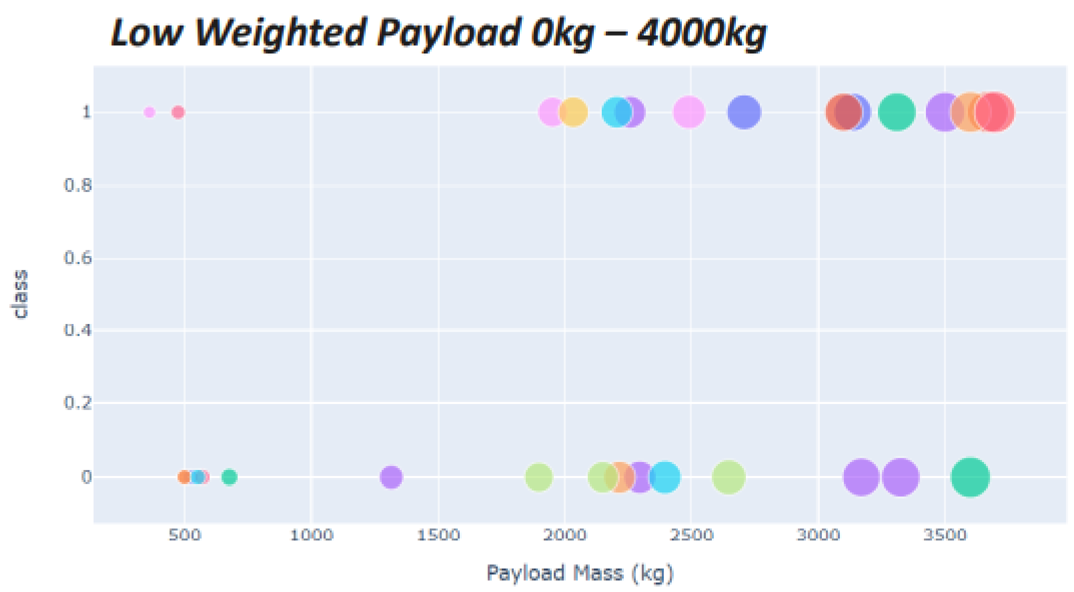
It can be observed that KSC LC-39A had the most successful launches from all the sites

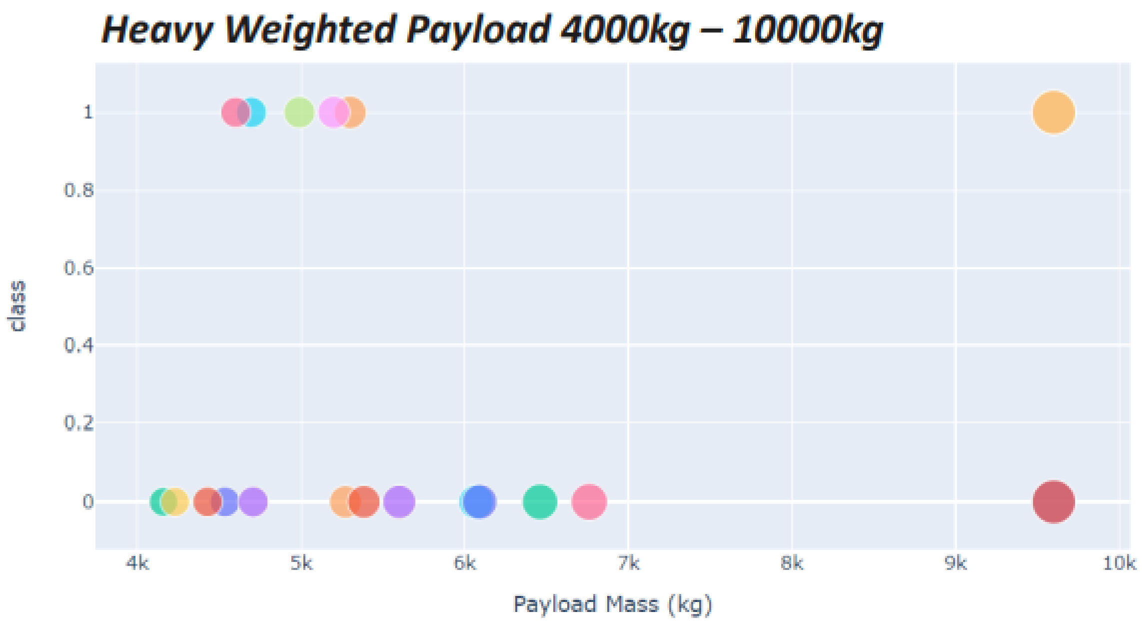
DASHBOARD – Pie chart for the launch site with highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

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





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)

Classification Accuracy

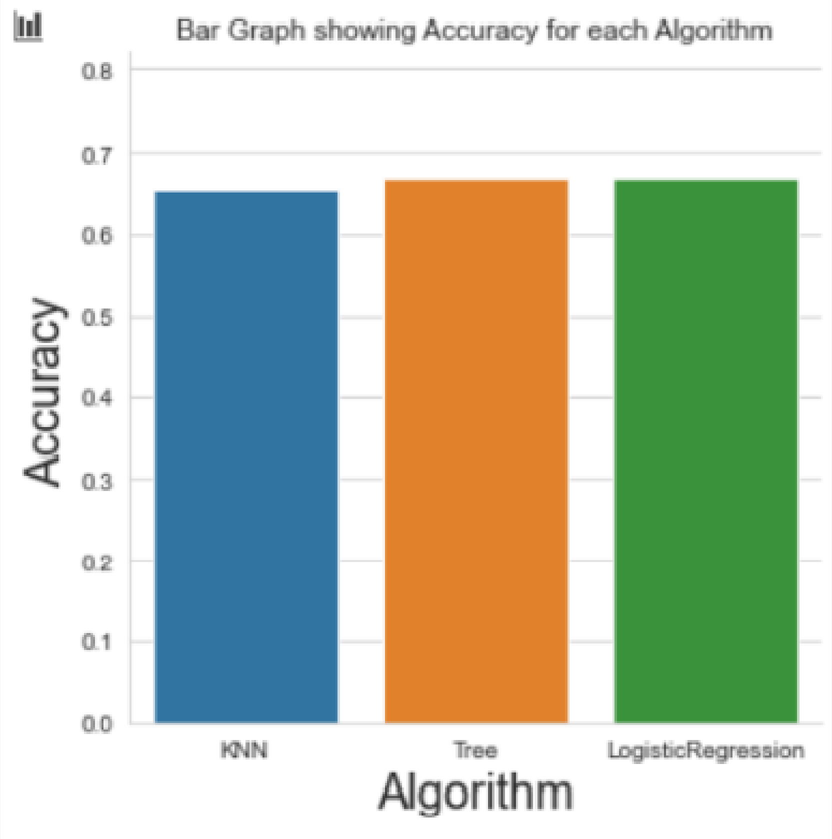
As you can see our accuracy is extremely close but we do have a winner its down to decimal places! using this function

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

Sr. No.	Accuracy	Algorithm
1	.6535	KNN
2	.6678	Tree
3	.6678	Logistic Regression

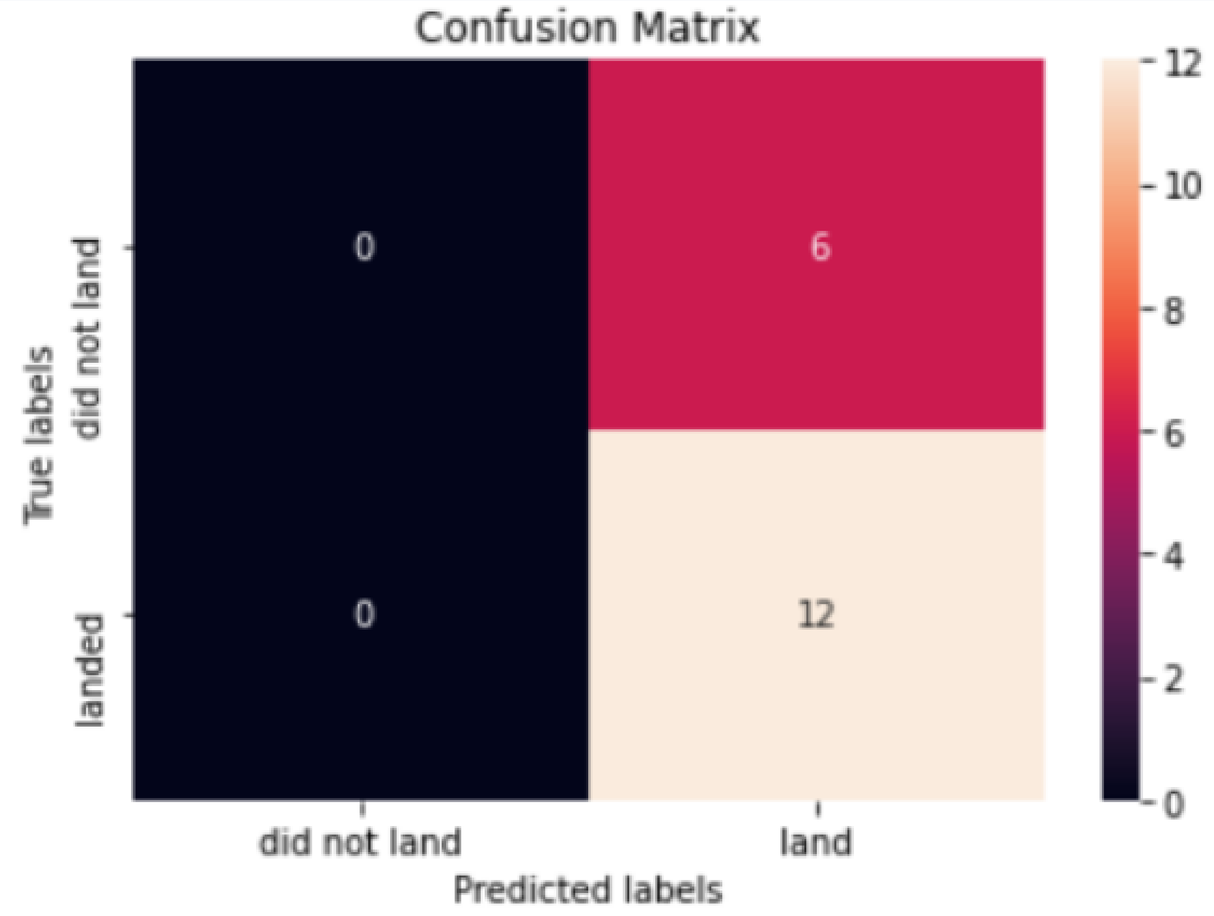
The Tree algorithm performs better.

The best accuracy for decision tree classifier is 83.33% accuracy on the test data.



Confusion Matrix

In the confusion matrix, it can be observed that Tree can distinguish between the different classes. And it can also be analyzed that the major problem is false positives.



Conclusions

- The Tree Classifier Algorithm is the best for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- It can be observed that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Thank you!