IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Durgesh Mishra
09/10/2021

Outline

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
  - During this capstone, we will determine if the first stage of the Falcon 9 will land successfully. According to SpaceX, Falcon 9 rocket launches cost about 62 million dollars, while other companies charge about 165 million dollars each. SpaceX saves so much money since the first stage is reusable. In order to figure out the cost of a launch we need to determine whether the first stage will land. In the event that another company wants to compete with SpaceX for a rocket launch, this information could be used.
  - Ultimately, the goal of this project is to create a machine learning pipeline that can predict whether the first stage will land based on historical data.

- 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 interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- How to build, tune, evaluate classification models

Data Collection

- Data is collected through get request (HTTP) to the SpaceX API.
- The API has been extended to include helper functions, which will allow us to retrieve information from the launch data based on identification numbers.
- Decode the response content as a Json using .json() function and turn it into a Pandas dataframe using .json_normalize() function.
- 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

- Request and parse the SpaceX launch data using the GET request and convert it to json
- Apply custom functions to clean data
- Filter the dataframe to only include `Falcon 9` launches.
    - Using the **BooosterVersion** column to only keep the Falcon 9 launches

- GitHub URL

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)
```

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
  - iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to extract column name one by one
- **Create a data frame by parsing the launch HTML tables**
- After filling the parsed launch record values into launch_dict, you can create a dataframe from it.
- GitHub URL

1 .Getting Response from HTML

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

2. Creating BeautifulSoup 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)

# 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. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
            # get table row
             for rows in table.find_all("tr")
                 #check to see if first table
```

7. Converting dictionary to dataframe

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

8. Dataframe to .CSV

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

Data Wrangling

Dealing with these missing values.

- The LandingPad column will retain None values to represent when landing pads were not used.
- Calculated the mean for the PayloadMass using the .mean(). Then used the mean and the .replace() function to replace np.nan values in the data with the mean that is calculated.

Calculate the number and occurrence of mission outcome per orbit type

- Use the method .value_counts() on the column Outcome to determine the number of landing_outcomes.Then assign it to a variable landing_outcomes.
- created a set of outcomes where the second stage did not land successfully

Create a landing outcome label from Outcome column

- Using the Outcome, created a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. Then assigned it to the variable landing_class.
- This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

GitHub URL

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 show how much one variable is affected by another. The relationship between two variables is called their correlation.
- GitHub URL

- Line Graph:
  - 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

- Bar Graph being drawn:
  - Mean VS. Orbit
- A bar diagram makes it easy to compare sets of data between different groups. 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.

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 the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
  - Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
  - Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- 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

[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 nonlinear pattern.
- The range of data flow, i.e., maximum and minimum value, can be determined.
- Observation and reading are straightforward.

URL Link to live website:

GitHub URL:

Predictive Analysis (Classification)

BUILDING MODEL

Use NumPy and Pandas to load our dataset

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 algorithm

Plot Confusion Matrix

IMPROVING MODEL

Feature Engineering

Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

Best performing classification model is the model with the best accuracy score

There is a dictionary of algorithms with ranks at the bottom of the notebook.

Results

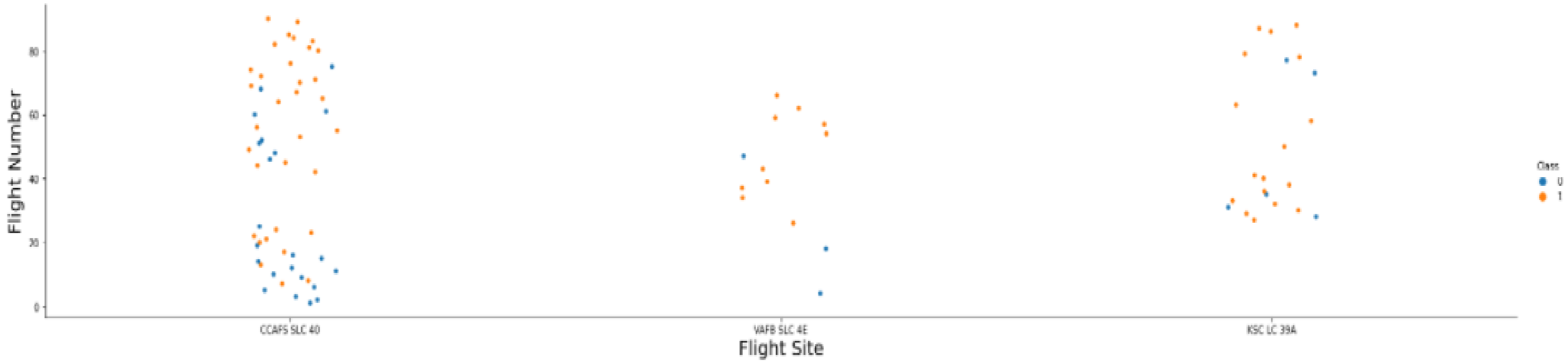
EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

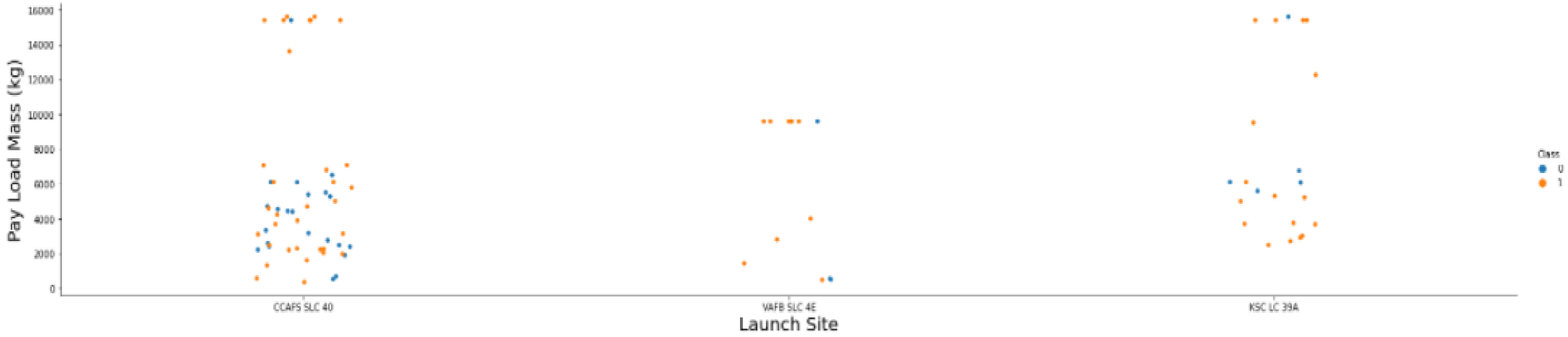
Insights Drawn From EDA

Flight Number vs. Launch Site



The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site

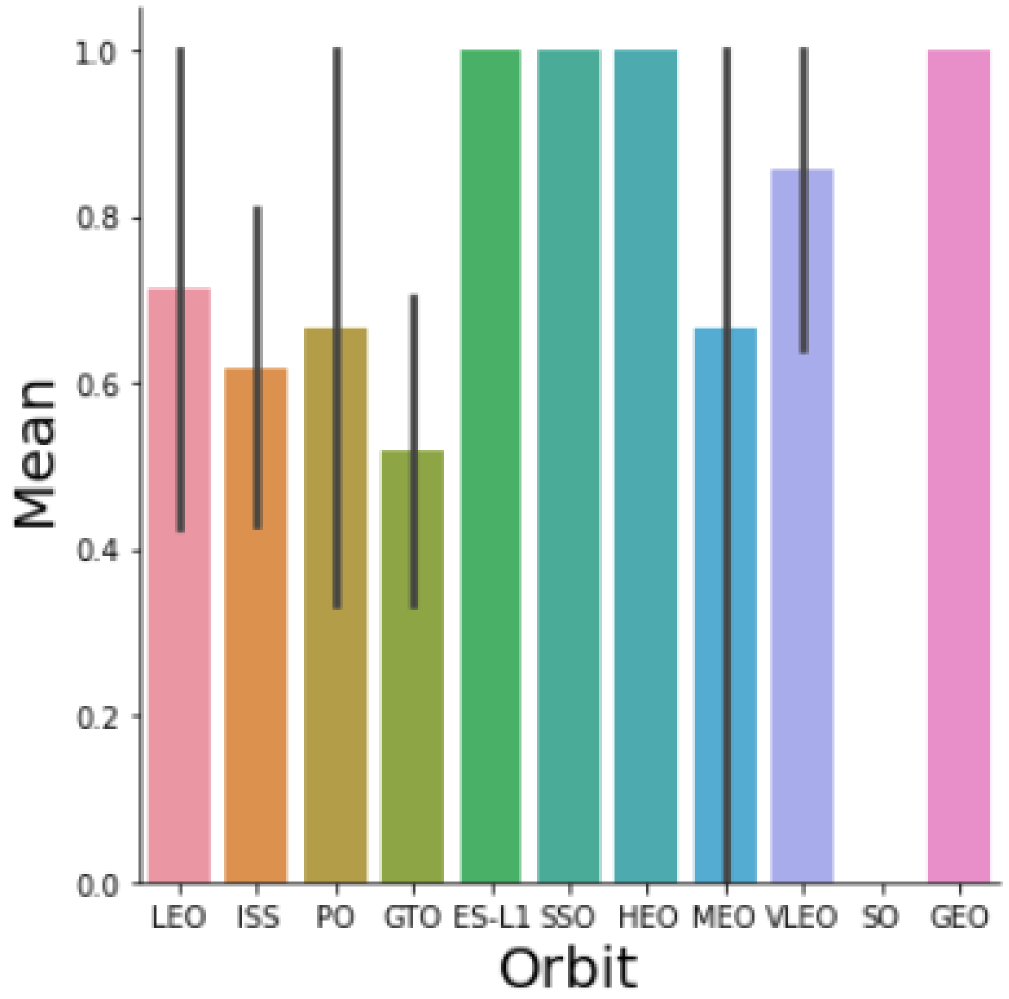


The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.

There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

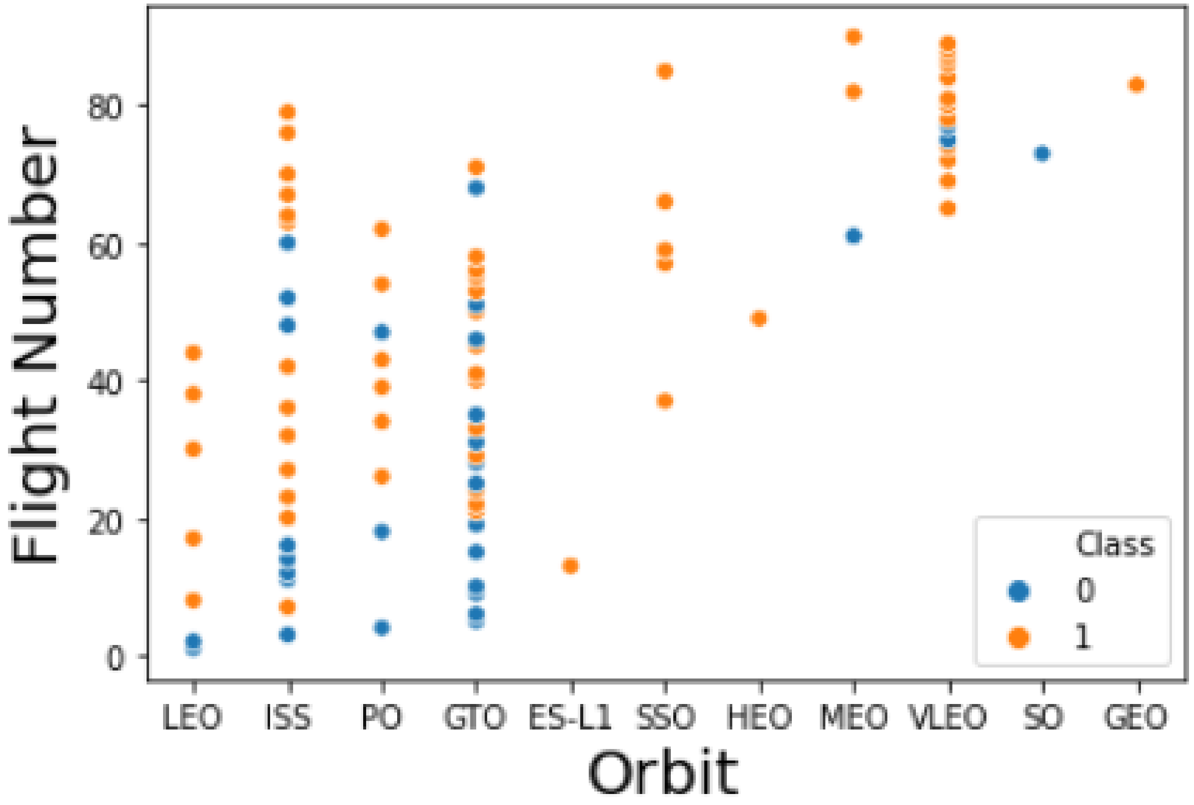
Success Rate vs. Orbit Type

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



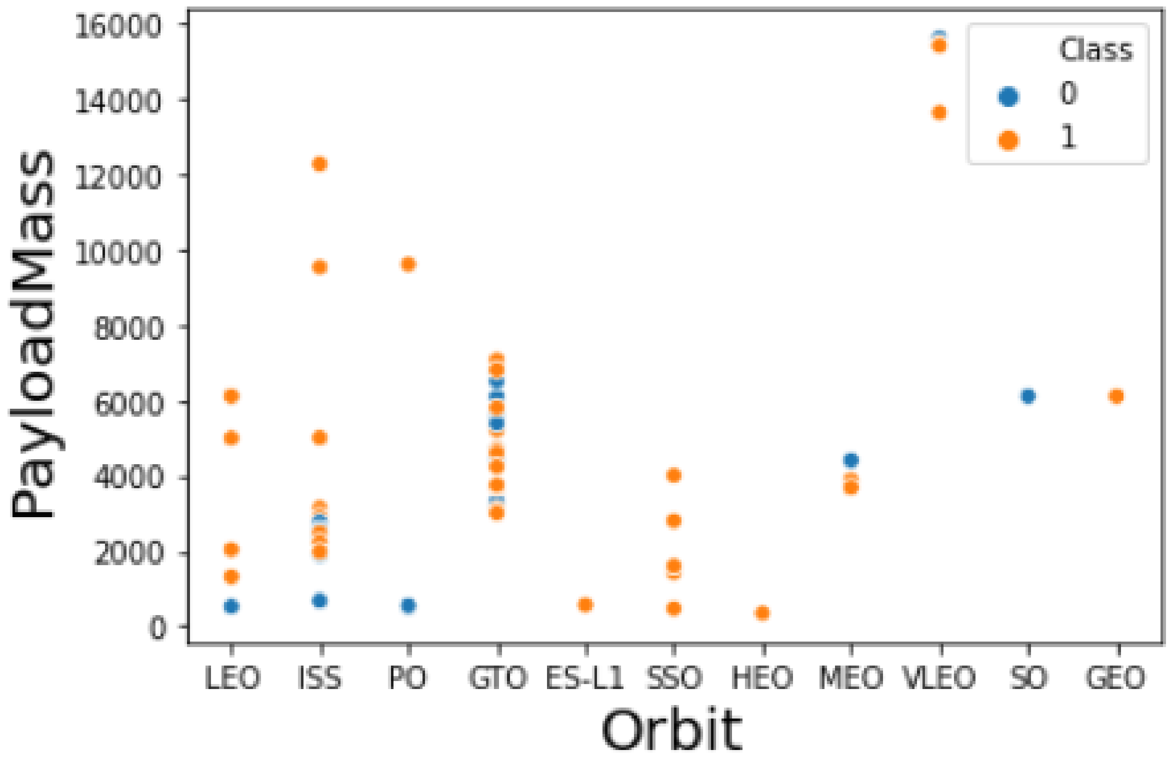
Flight Number vs. Orbit Type

- You should see that 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



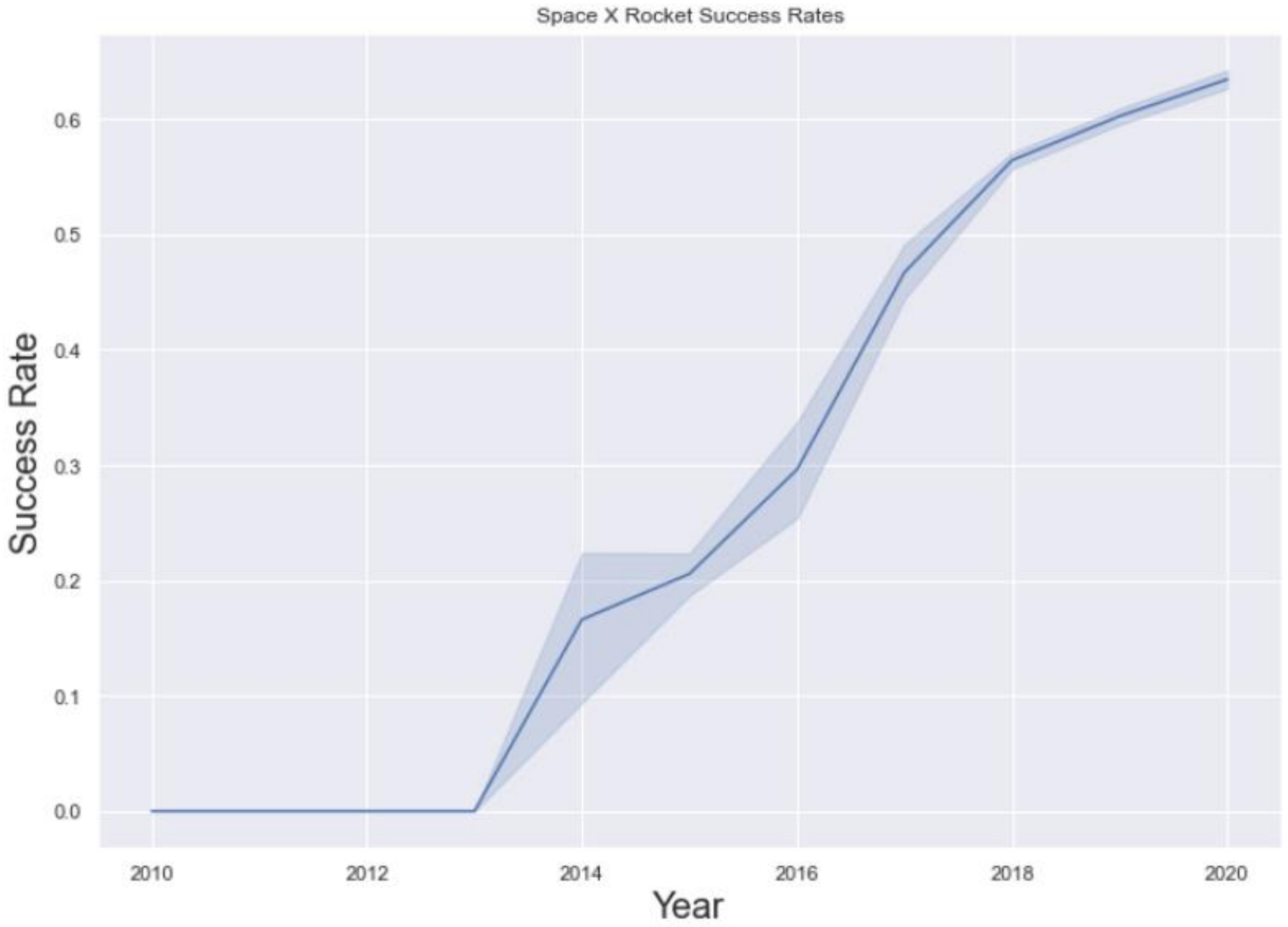
Payload vs. Orbit Type

- You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend

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



All Launch Site Names

SQL QUERY

Select DISTINCT Launch_Site from tblSpaceX

QUERY EXPLAINATION

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

Unique Launch Sites
CCAFS LC-40
CCAFS SLC-40
CCAFSSLC-40
KSC LC-39A
VAFB SLC-4E

	Date	Time_UTC	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	19-02-2017	2021-07-02 14:39:00.0000000	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
1	16-03-2017	2021-07-02 06:00:00.0000000	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2	30-03-2017	2021-07-02 22:27:00.0000000	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
3	01-05-2017	2021-07-02 11:15:00.0000000	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
4	15-05-2017	2021-07-02 23:21:00.0000000	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Launch Site Names Begin with 'CCA'

- SQL QUERY
  - Select TOP 5 * from tblSpaceX WHERE Launch_Site LIKE 'KSC%’
- 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 'KSC%' the percentage in the end suggests that the Launch_Site name must start with KSC.

Total Payload Mass

SQL QUERY

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

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)

	Total Payload Mass
0	45596

Average Payload Mass by F9 v1.1

SQL QUERY

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

	Average Payload Mass
0	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

SQL QUERY

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

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)

	Date which first Successful landing outcome in drone ship was acheived.
0	06-05-2016

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY

Select Booster_Version from tblSpaceX where Landing_Outcome = 'Success (ground pad)’AND Payload_MASS_KG 4000 AND Payload_MASS_KG 6000

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

	Date which first Successful landing outcome in drone ship was acheived.
0	F9 FT B1032.1
1	F9 B4 B1040.1
2	F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

SQL 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 '%F ailure%') as Failure_Mission _Coutcomes

QUERY EXPLAINATION

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.

	Successful_Mission_Outcomes	Failure_Mission_Outcomes
0	100	1

Boosters Carried Maximum Payload

SQL 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

	Booster_Version	Maximum Payload Mass
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
...	...	...
92	F9 v1.1 B1003	500
93	F9 FT B1038.1	475
94	F9 B4 B1045.1	362
95	F9 v1.0 B0003	0
96	F9 v1.0 B0004	0

97 rows × 2 columns

2015 Launch Records

SQL QUERY

SELECT Date, Booster_Version, Launch_Site, Landing_Outcome FROM tblSpaceX WHERE (Landing_Outcome LIKE N’%Faliure%') AND YEAR(CONVERT(date,Date, 105)) = ‘2015'

Date	Booster_Version	Launch_Site	Landing _Outcome
10/1/2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
14-04-2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

QUERY EXPLAINATION

Date fields in SQLServer stored as NVARCHAR. The function CONVERT converts NVARCHAR to Date WHERE clause filters Year to be 2015 and landing outcome to be failure.

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

SQL QUERY

SELECT COUNT(Landing_Outcome) AS sl FROM dbo.tblSpaceX WHERE ((Landing_Outcome = 'Failure (drone ship)') OR (Landing_Outcome = 'Success (ground pad)')) AND (Date >'04-06-2010') AND (Date < '20-03-2017')

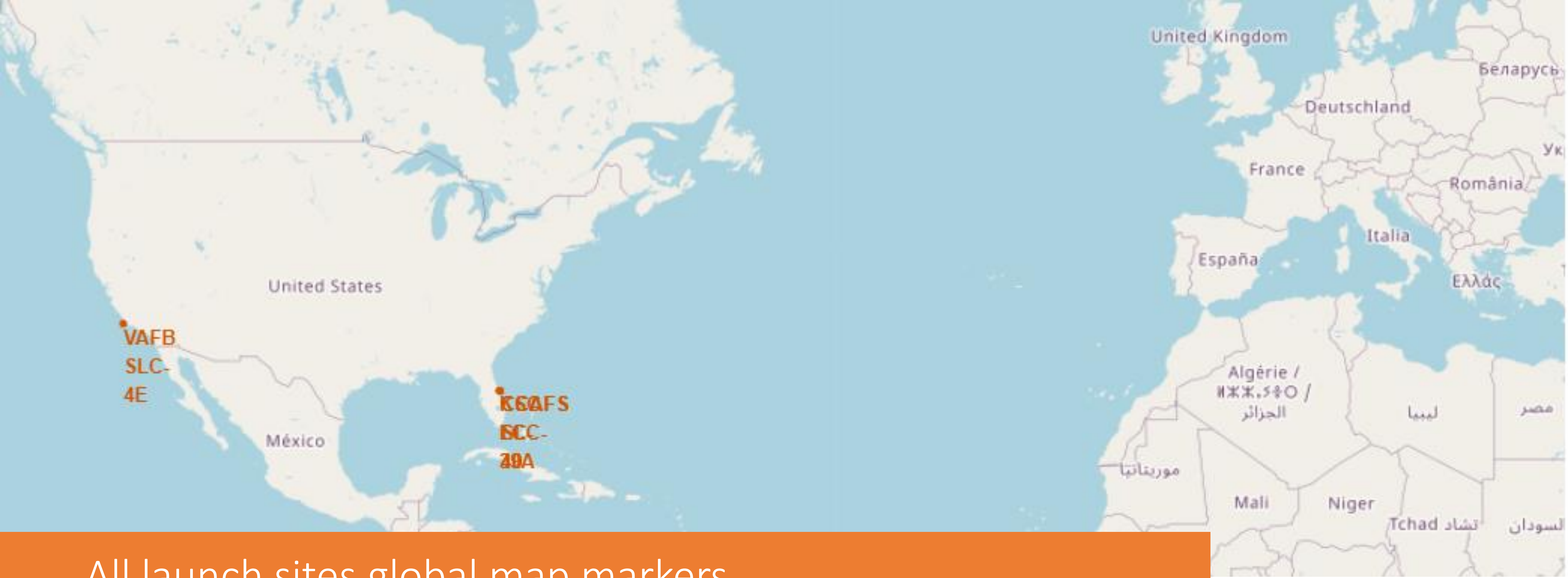
QUERY EXPLAINATION

Used subqueries here to produce the results. We need to find the count of landing outcomes that are successful ground pad or failure drone ship between 2010-06-04 and 2017-03-20

Landing Outcomes between 2010-06-04 and 2017-03-20
10

Section 4

Launch Sites Proximities Analysis



All launch sites global map markers

- We can see that the SpaceX launch sites are in the United States of America coasts.
    - Florida and California

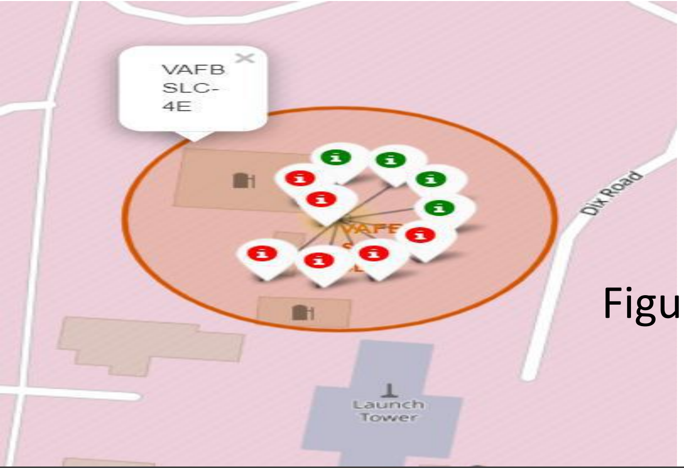
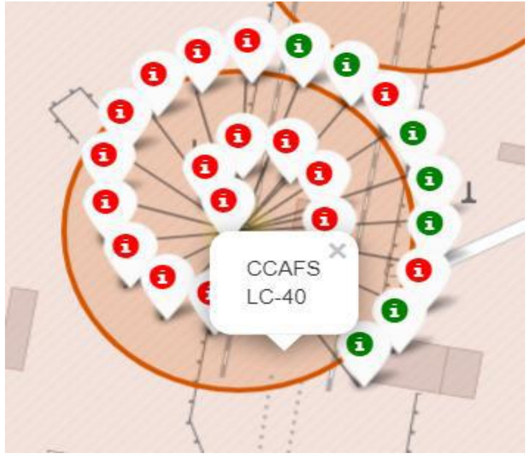
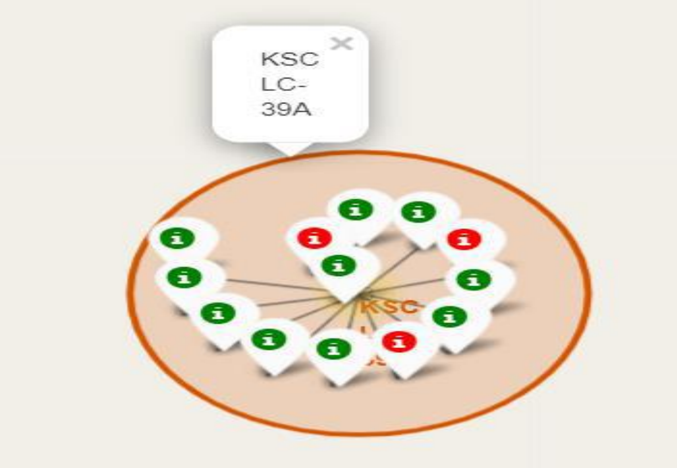


Figure 1

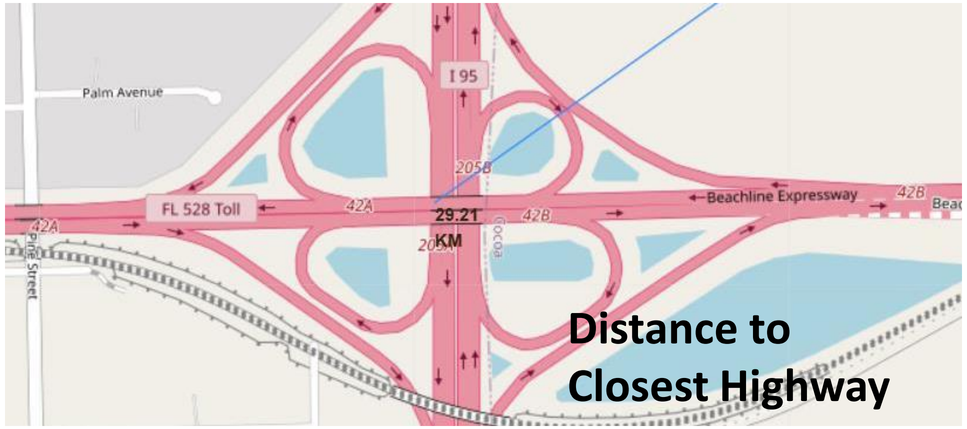


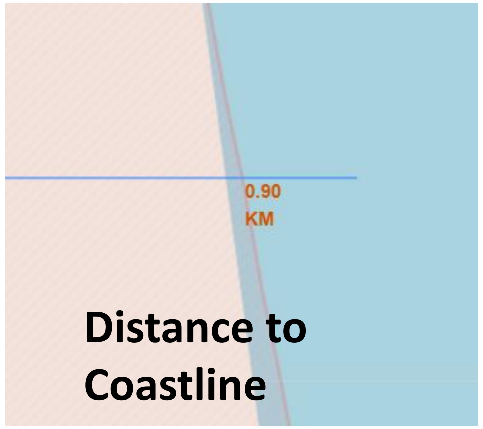


Color Labelled Markers

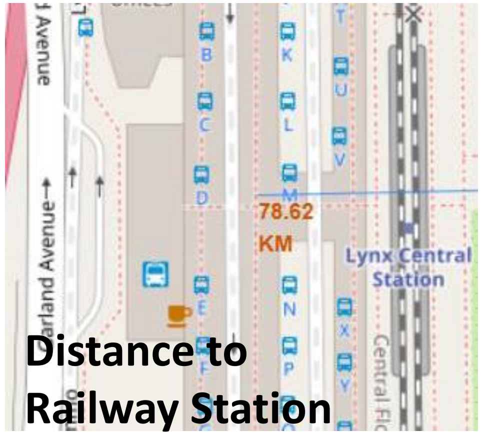
- Florida Launch Sites
    - Green Marker shows successful Launches and Red Marker shows Failures
- California Launch Site (Figure 1)

Working out Launch Sites distance to landmarks to find trends with Haversine formula using CCAFS SLC 40 as a reference





- 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

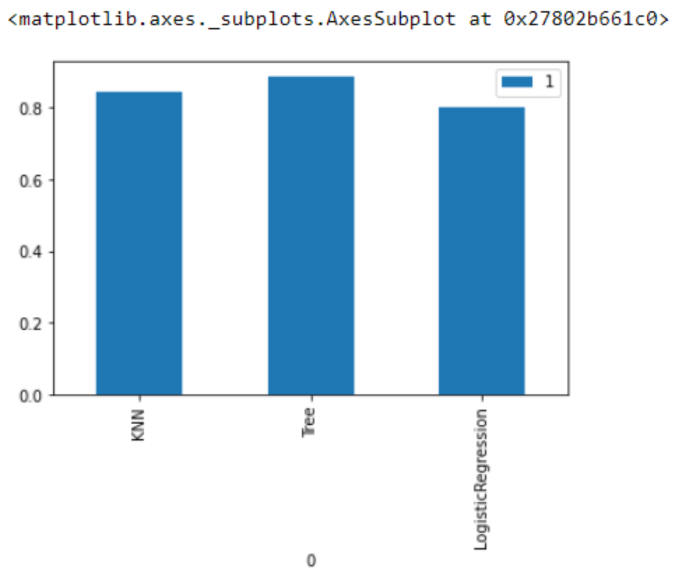


Section 6

Predictive Analysis (Classification)

Classification Accuracy

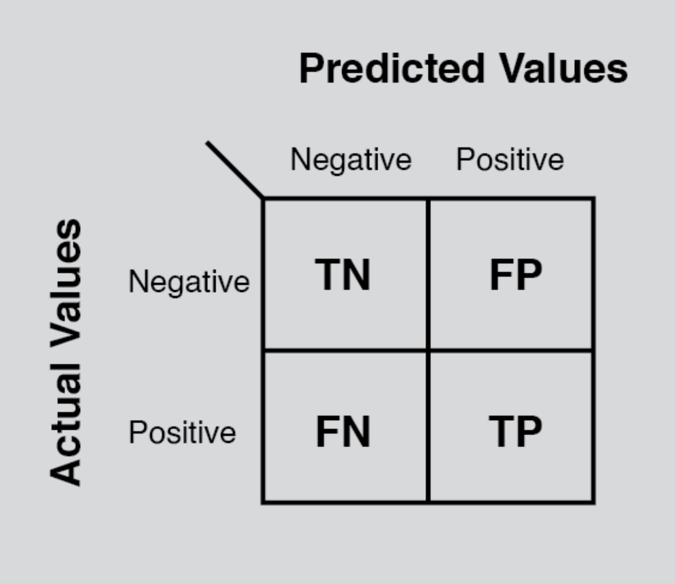
- As you can see our accuracy is extremely close, but we do have a winner its down to decimal places using below function.
- bestalgorithm = max(algorithms, key=algorithms.get)
- Best Algorithm is Tree with a score of 0.8875
- Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best’}
- After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 94.44% accuracy on the test data.

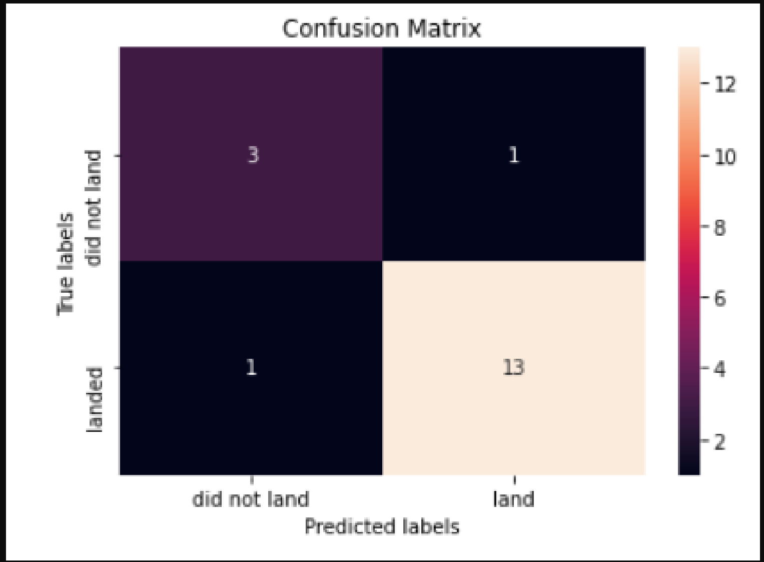


	0	1
0	KNN	0.844643
1	Tree	0.887500
2	LogisticRegression	0.803571

Confusion Matrix for the Tree

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





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 directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC 39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate

Appendix

- Haversine formula
- Module sqlserver (ADGSQLSERVER)
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Thank you!