IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Manok Jo
15 JUL 2024

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 result

Introduction

- Project background and context

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.

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

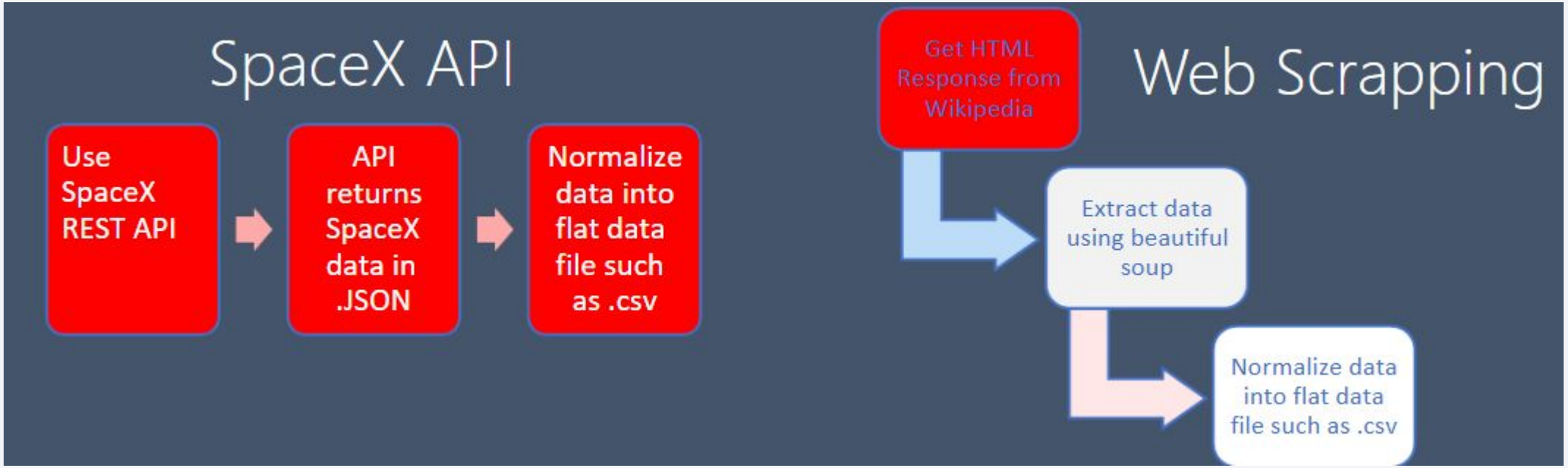
- Data collection methodology:

  SpaceX Rest API

  (web Scrapping) from Wikipedia

- Perform data wrangling – Transforming data for Machine Learning
  - One Hot Encoding data fields for Machine Learning and Dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

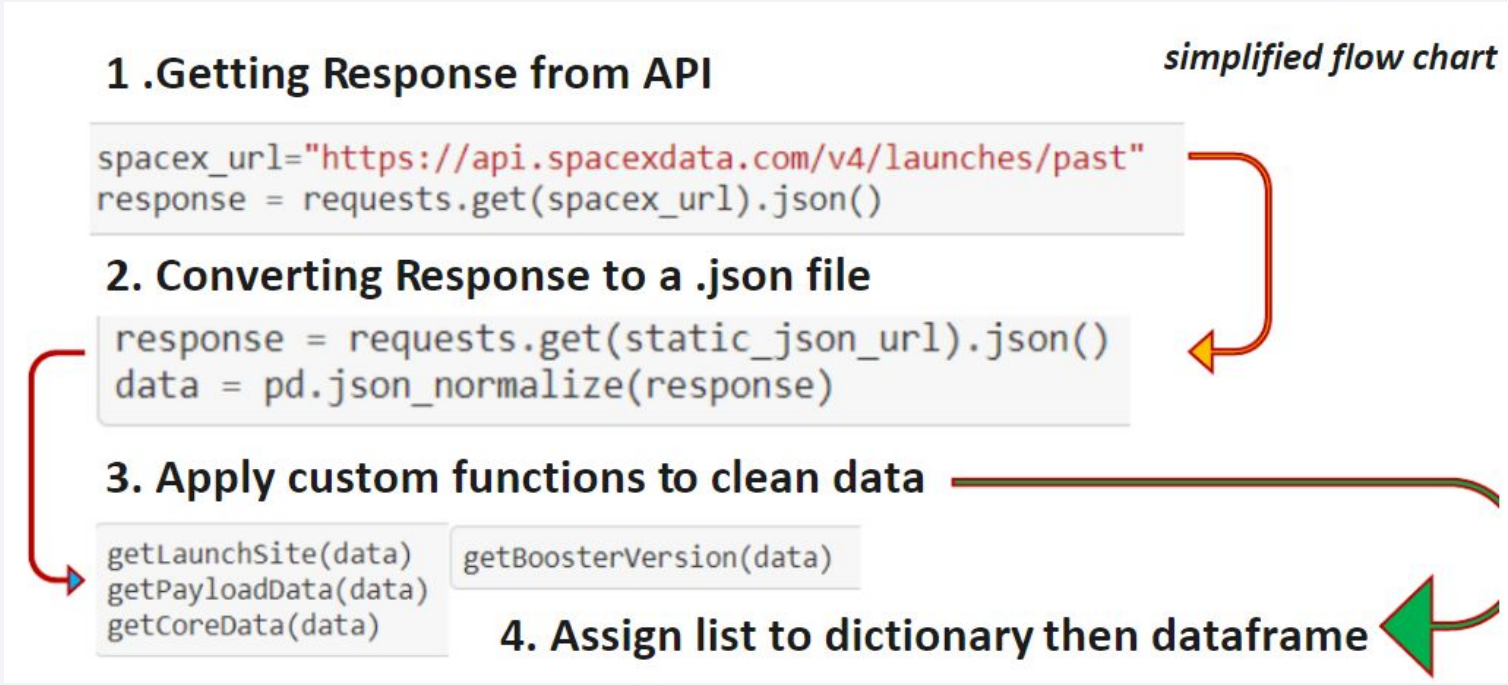
Data Collection



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose




```
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}
```

Data Collection – SpaceX API

```
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)
```

8

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

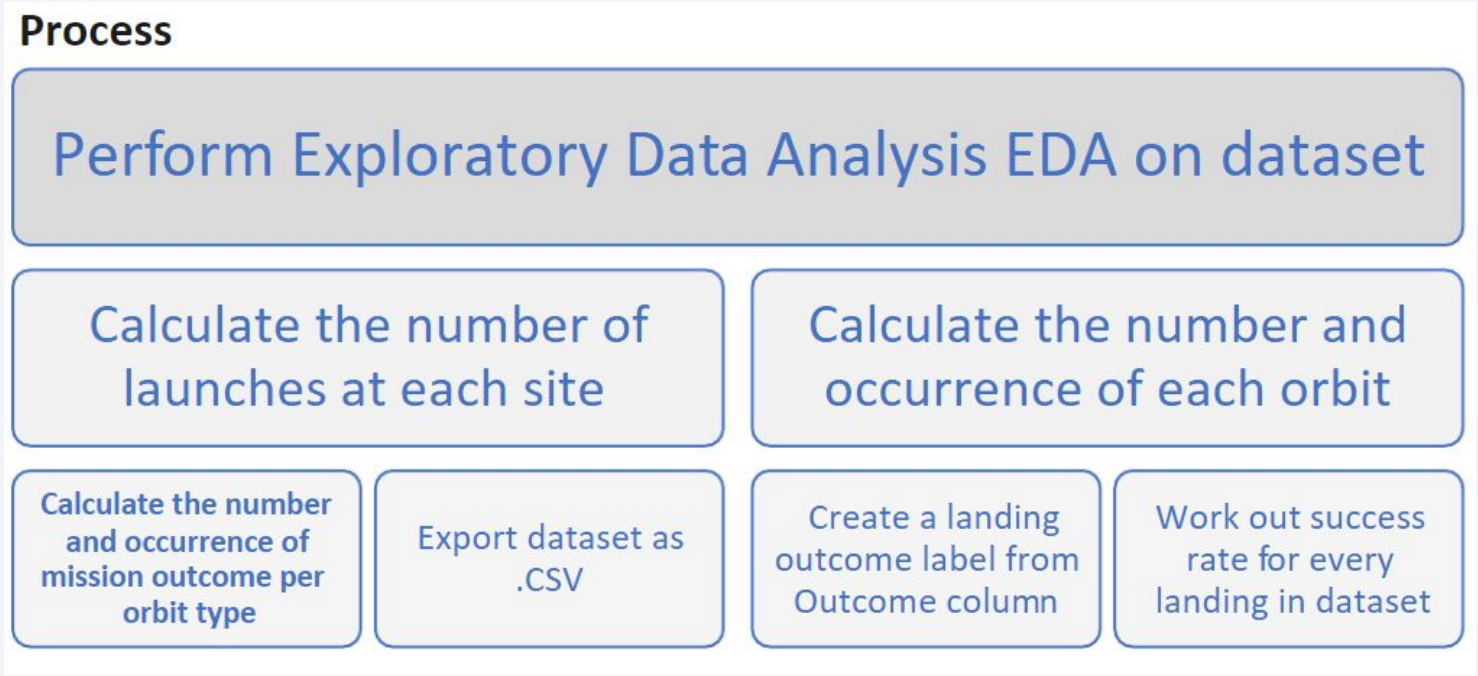
```
launch_dict= dict.fromkeys(column_names)

# 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

Data Wrangling



GitHub URL to Notebook

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 . Scatter plots usually consist of a large body of dat

Bar Graph :

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

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

GitHub URL to Notebook

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 to Notebook

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 to Notebook

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.

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

GitHub URL to Notebook

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

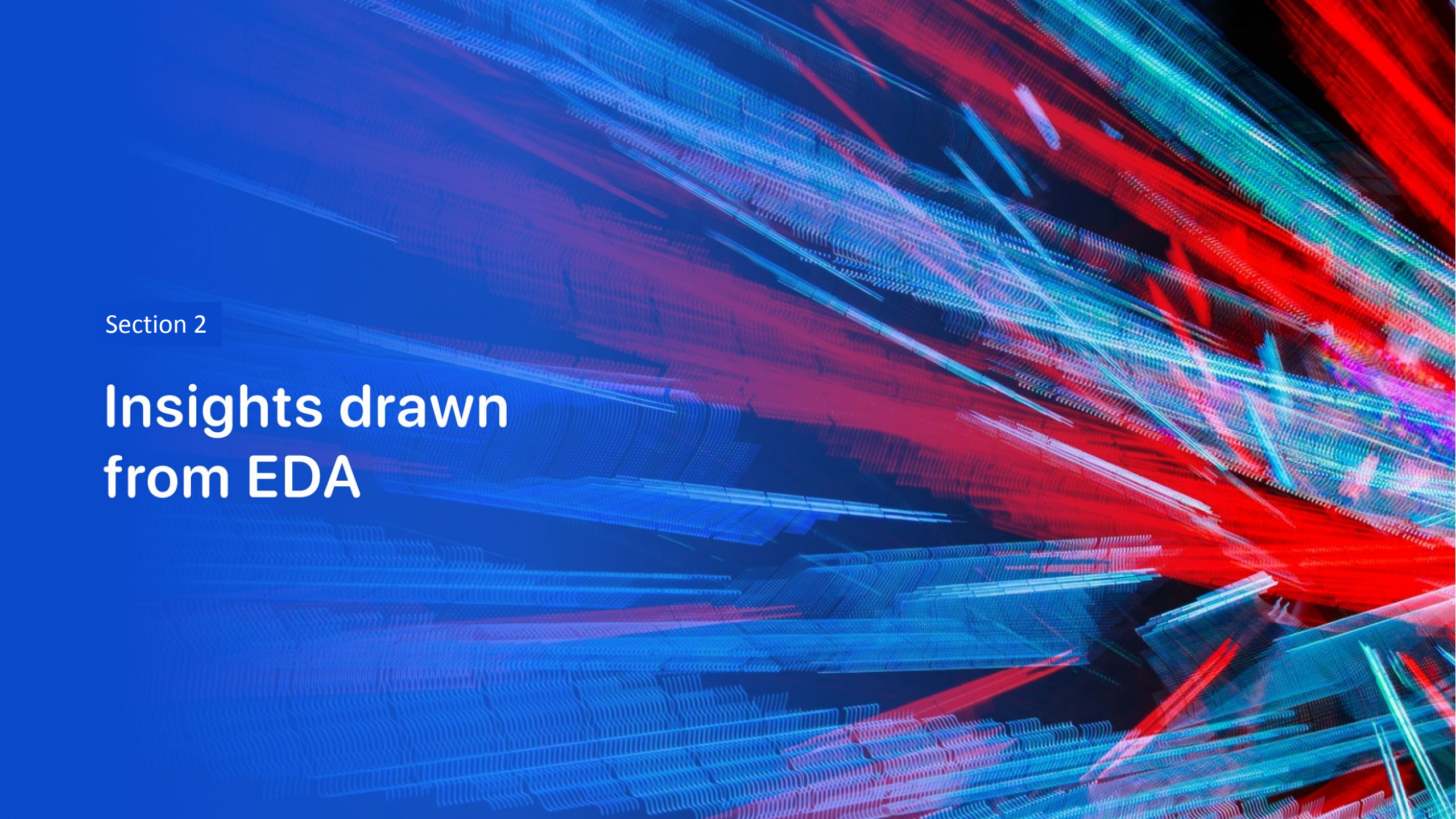
•The model with the best accuracy score wins the best performing model
•In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

Results

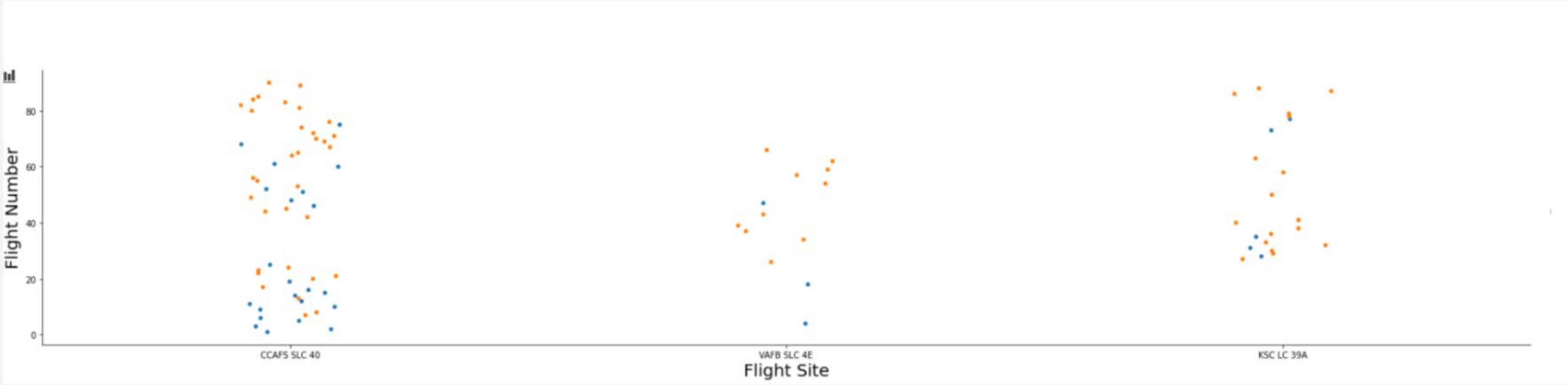
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

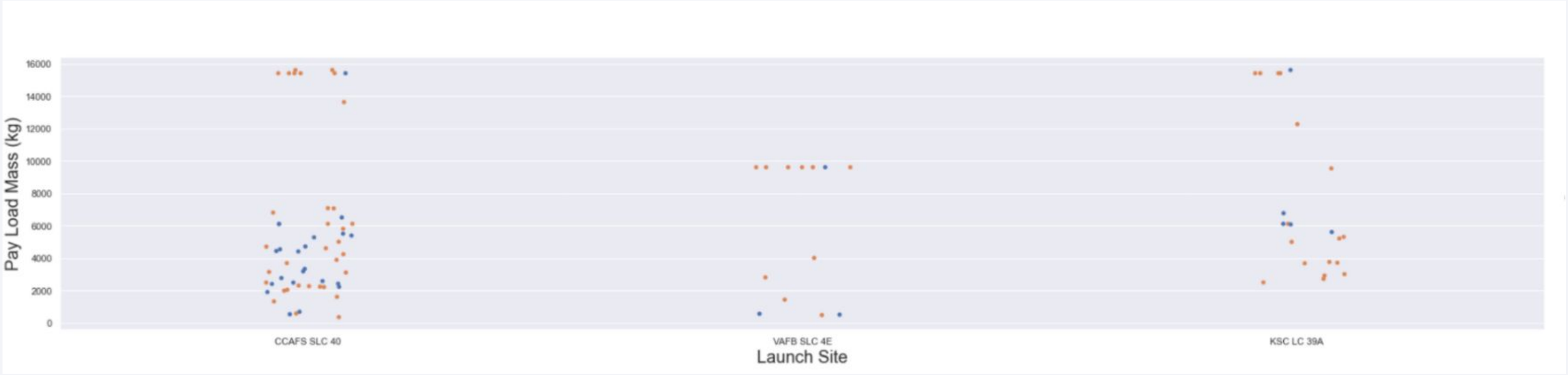


Flight Number vs. Launch Site



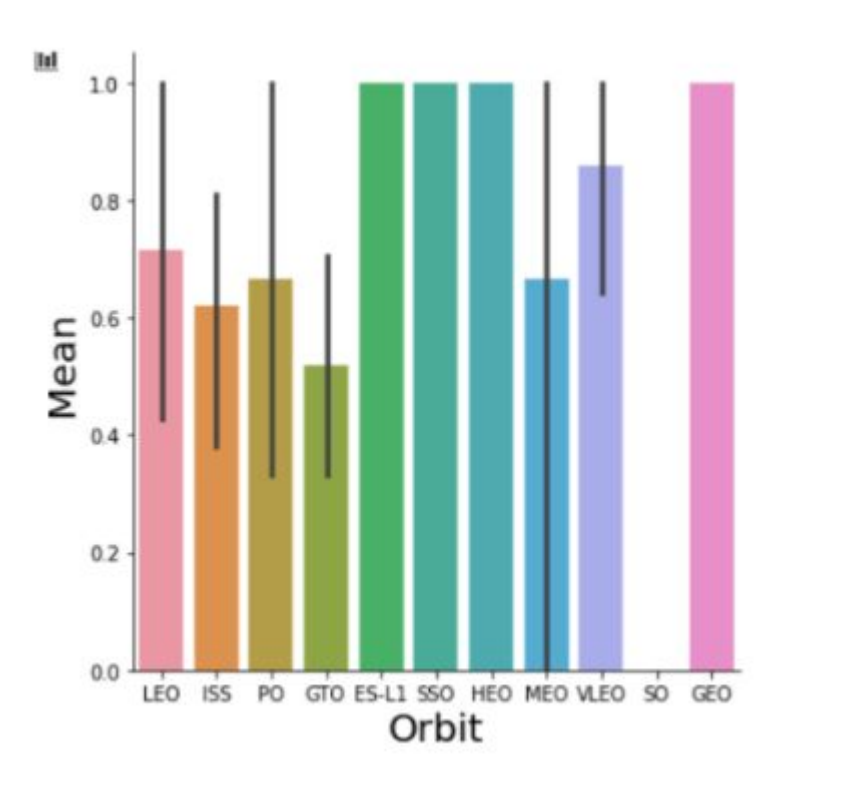
The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site



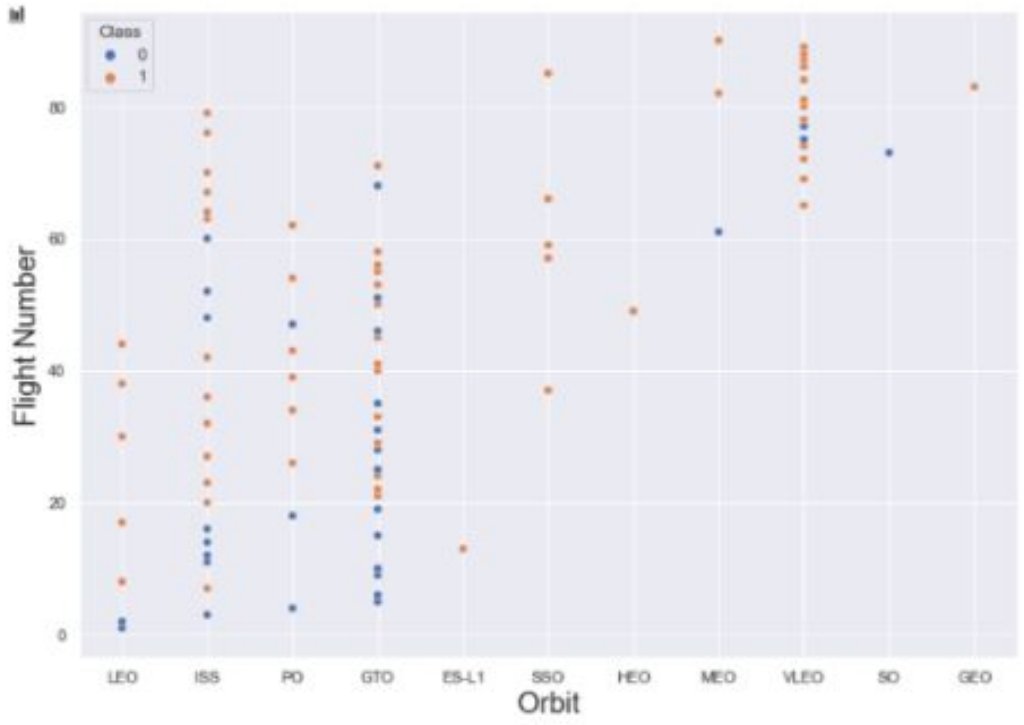
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

Success Rate vs. Orbit Type



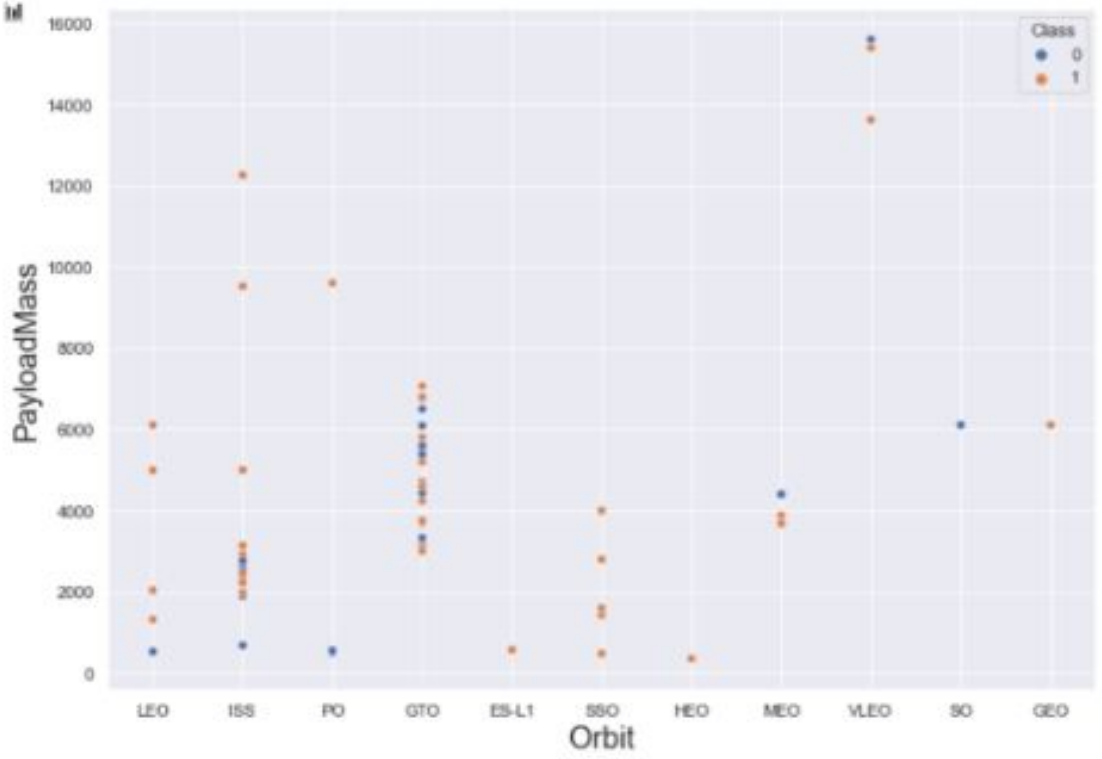
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Flight Number vs. Orbit Type



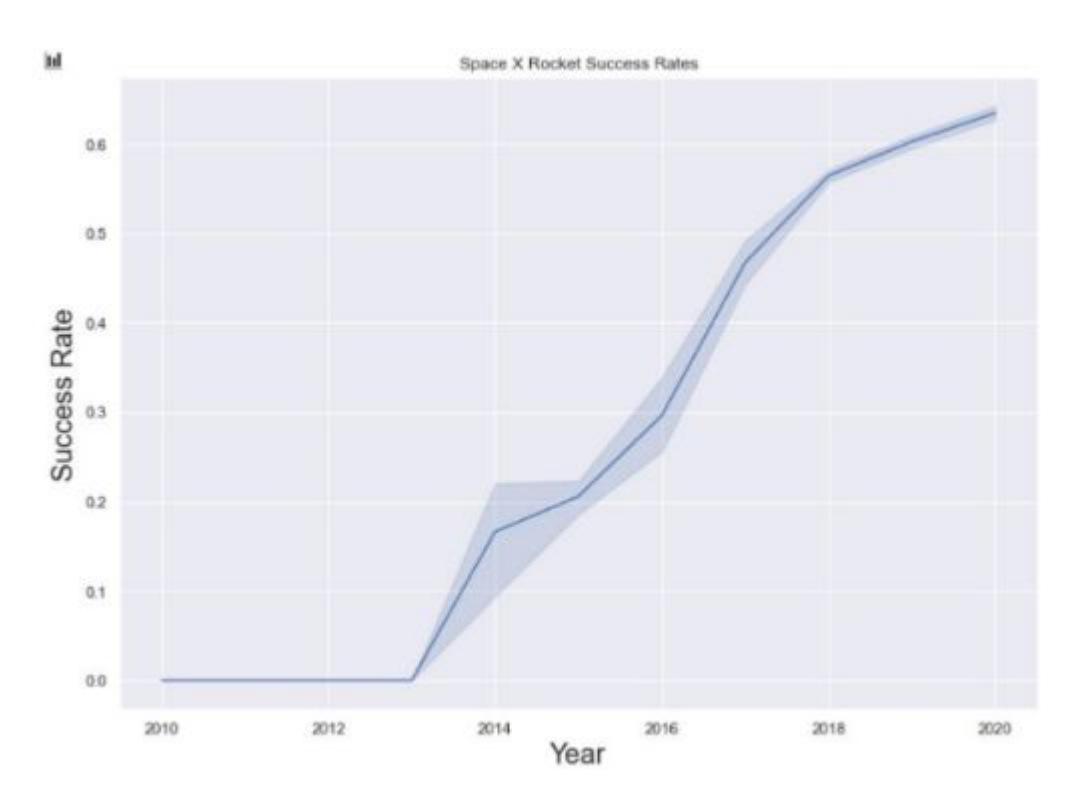
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

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

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Launch Site Names Begin with 'CCA'

```
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

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

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 (CR

Average Payload Mass by F9 v1.1

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

SQL QUERY

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

First Successful Ground Landing Date

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

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

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

SQL QUERY

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

```
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 EXPLAINATION

a much harder query I must say, 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

```
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

```
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

a much more complex query as I had my 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

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

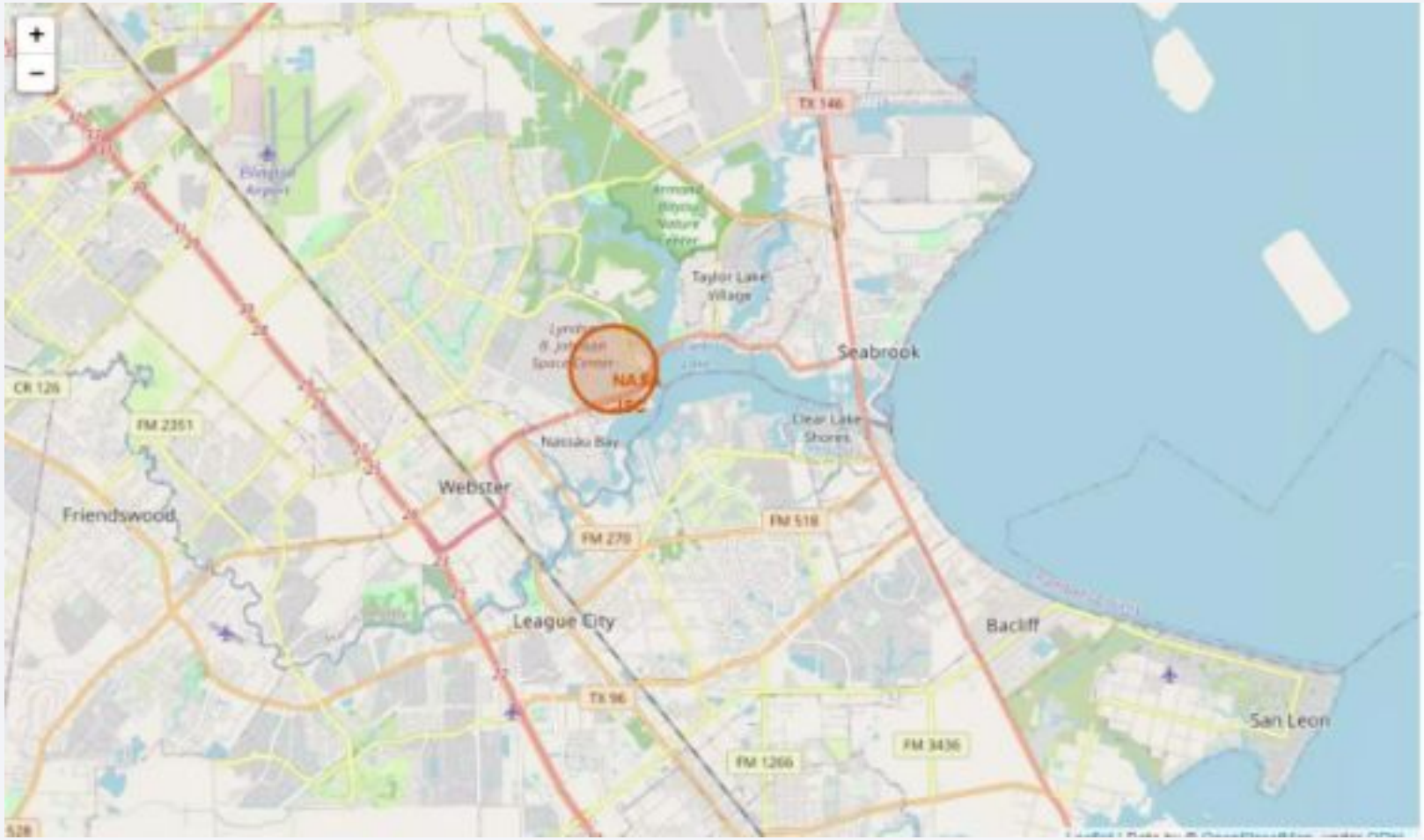
QUERY EXPLAINATION

Function COUNT counts records in column

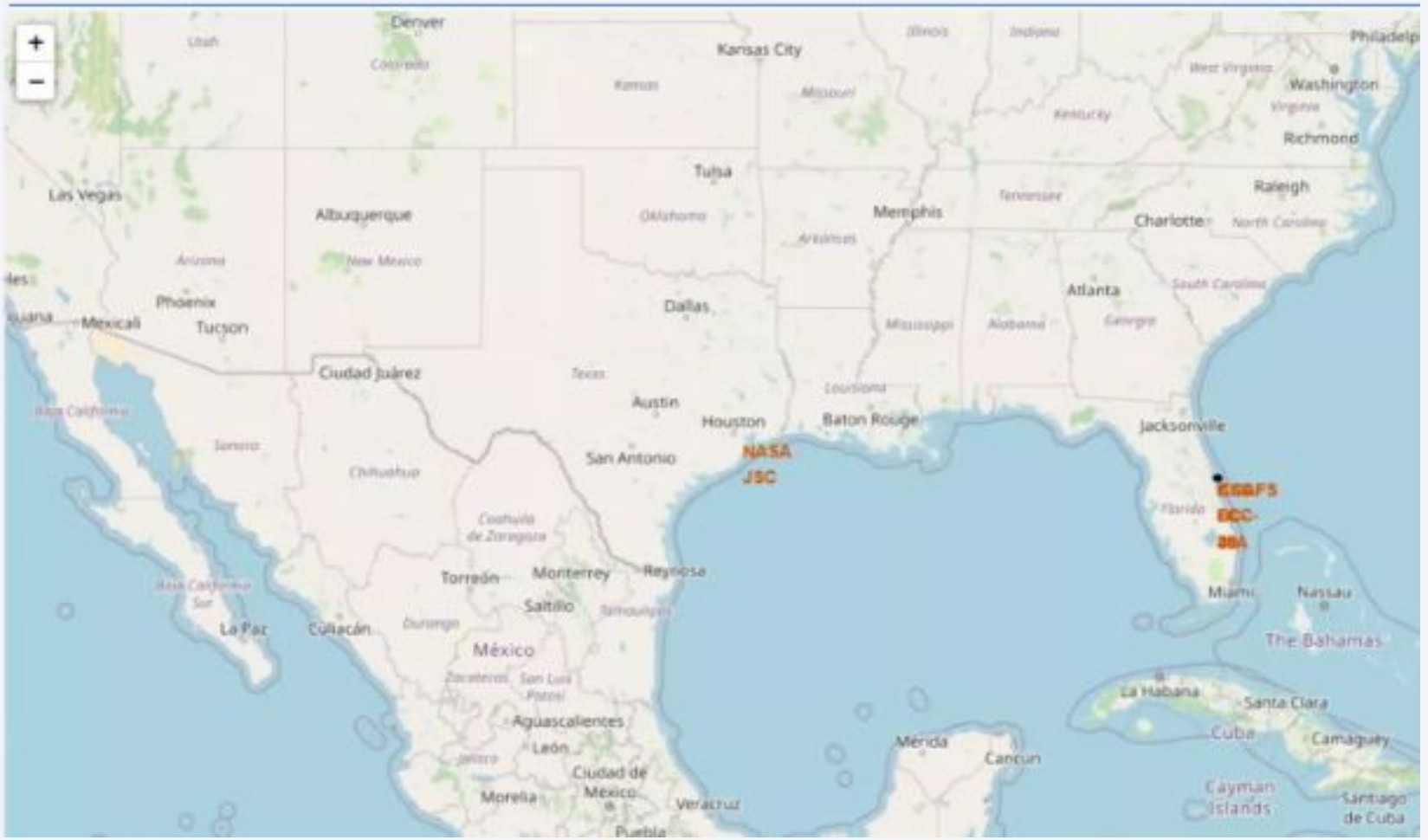
WHERE filters data
LIKE (wildcard)
AND (conditions)
AND (conditions)


Section 3
Launch Sites
Proximities Analysis

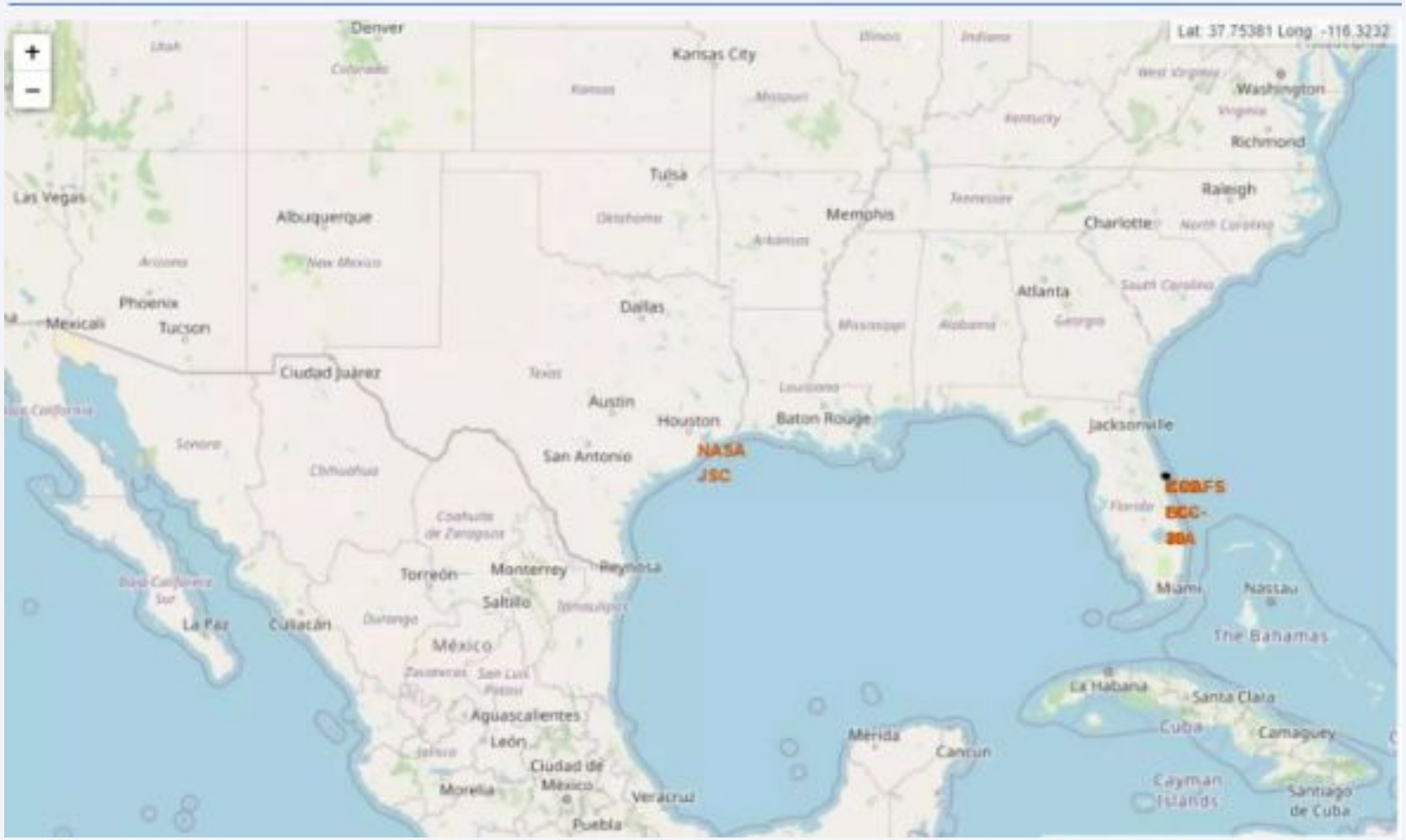
< All launch sites marked on a map>

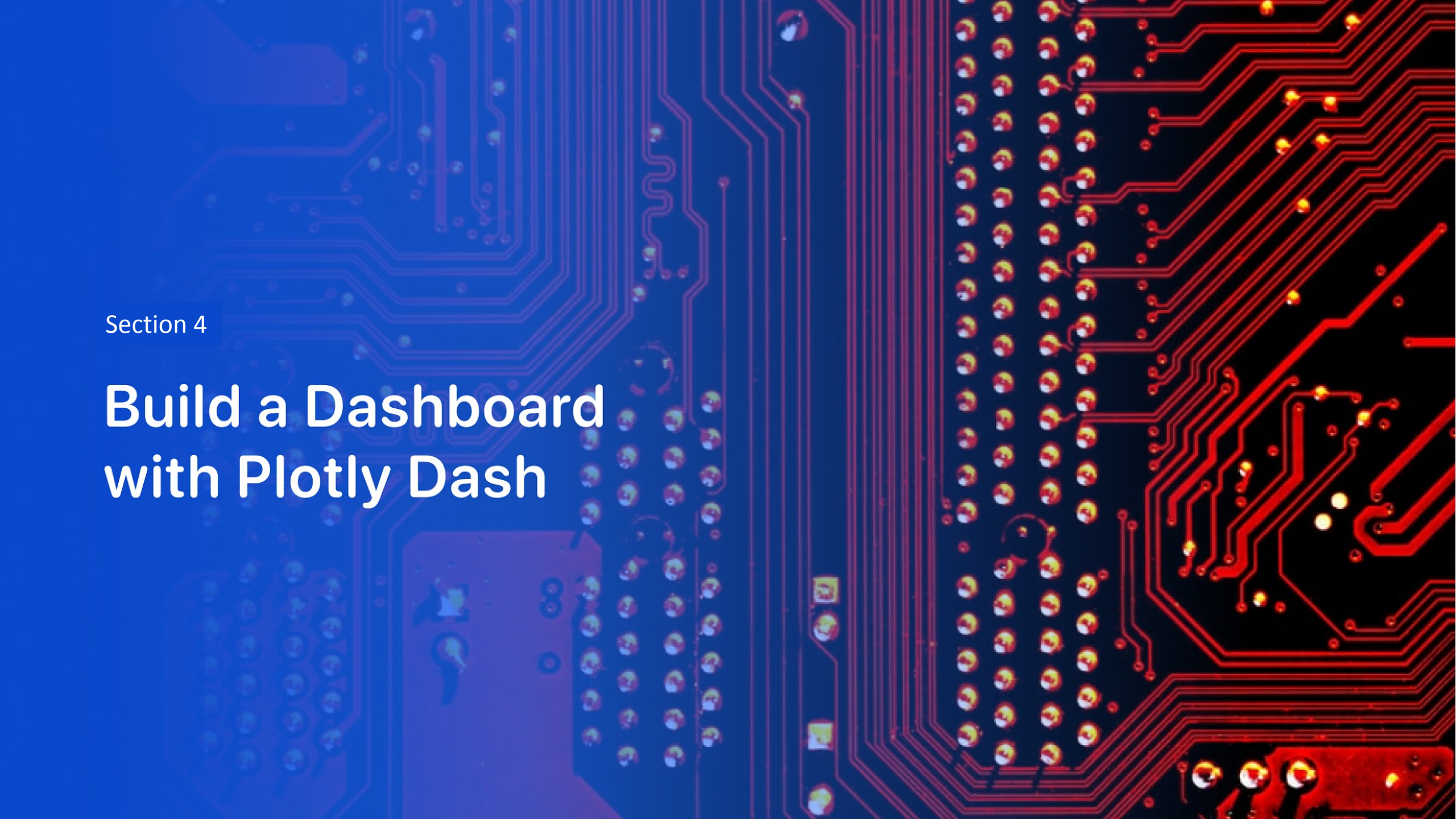


< Success/failed launches marked on the map >



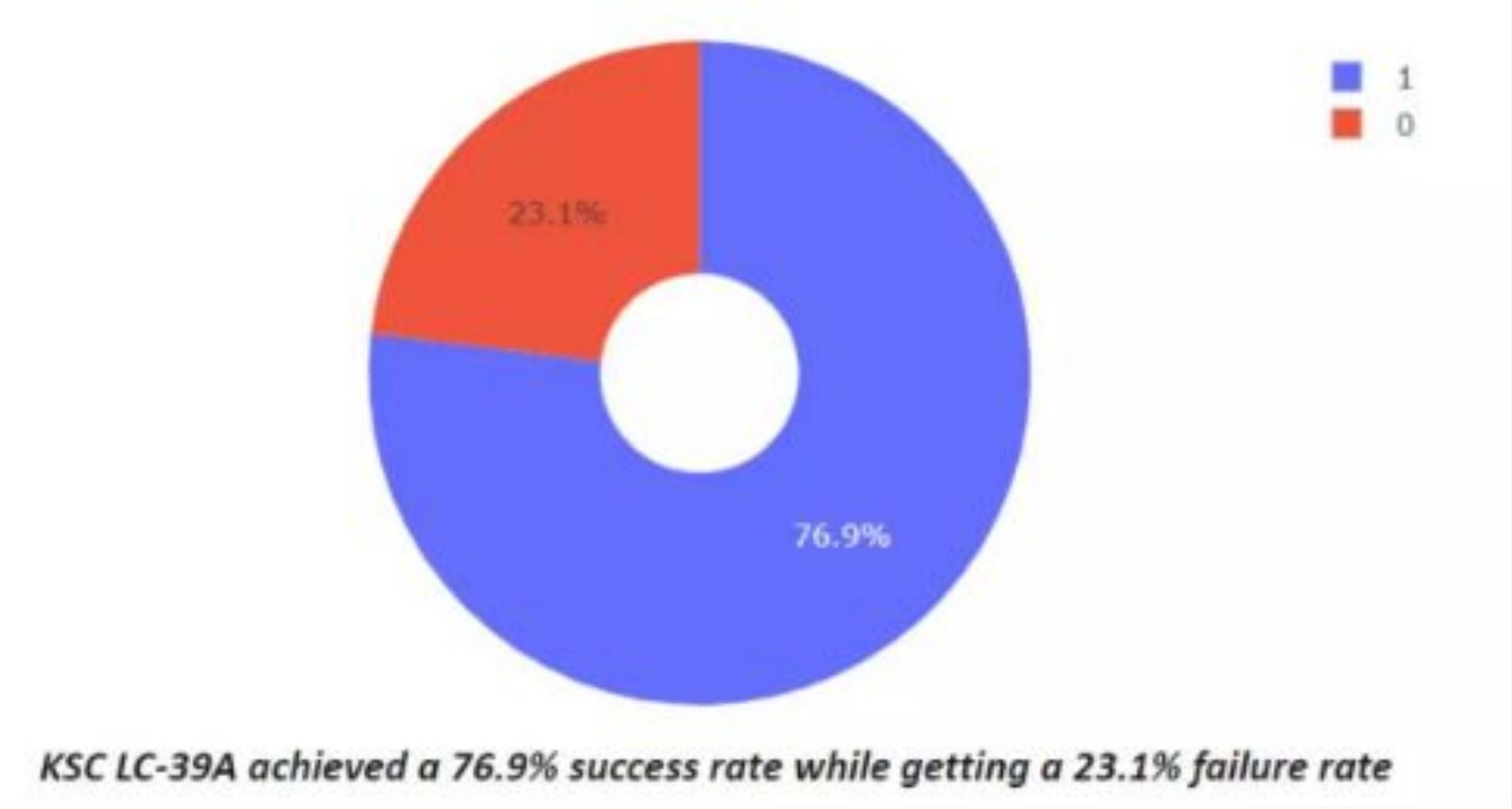
<Distance between a launch site to its proximities>



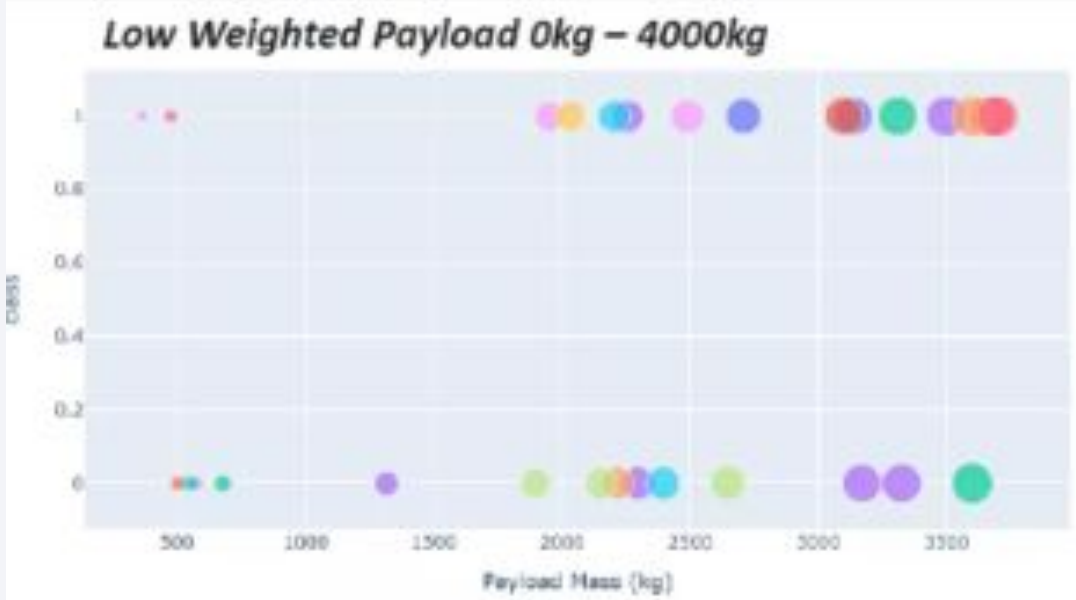

Section 4
Build a Dashboard with Plotly Dash

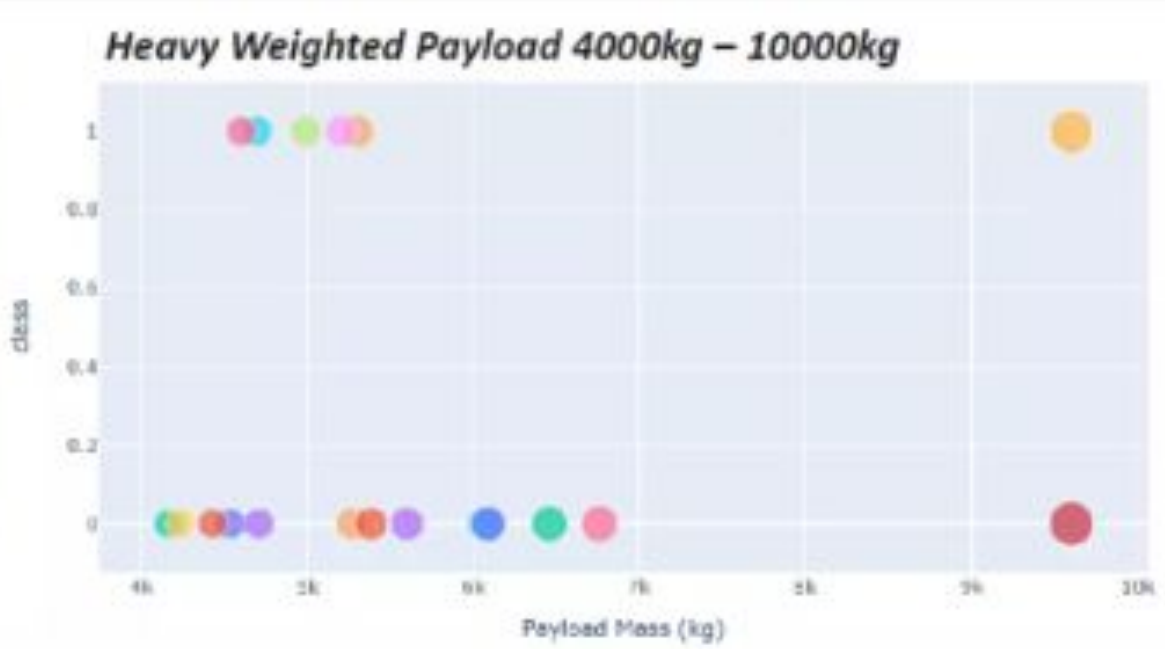
<Total success launches by all sites>

<Success rate by site>



<Payload vs. Launch Outcome scatter plot for all sites>



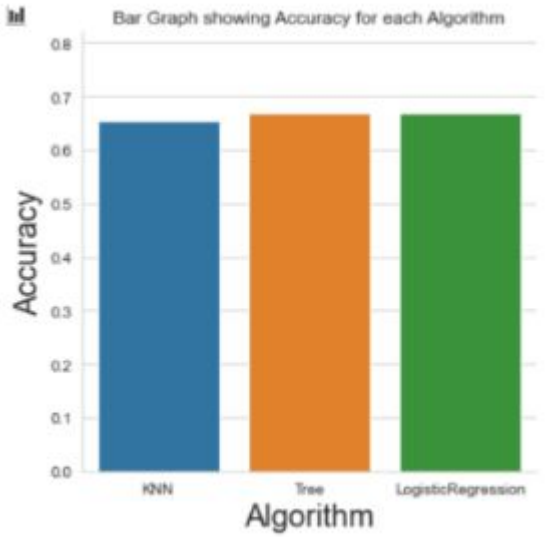


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

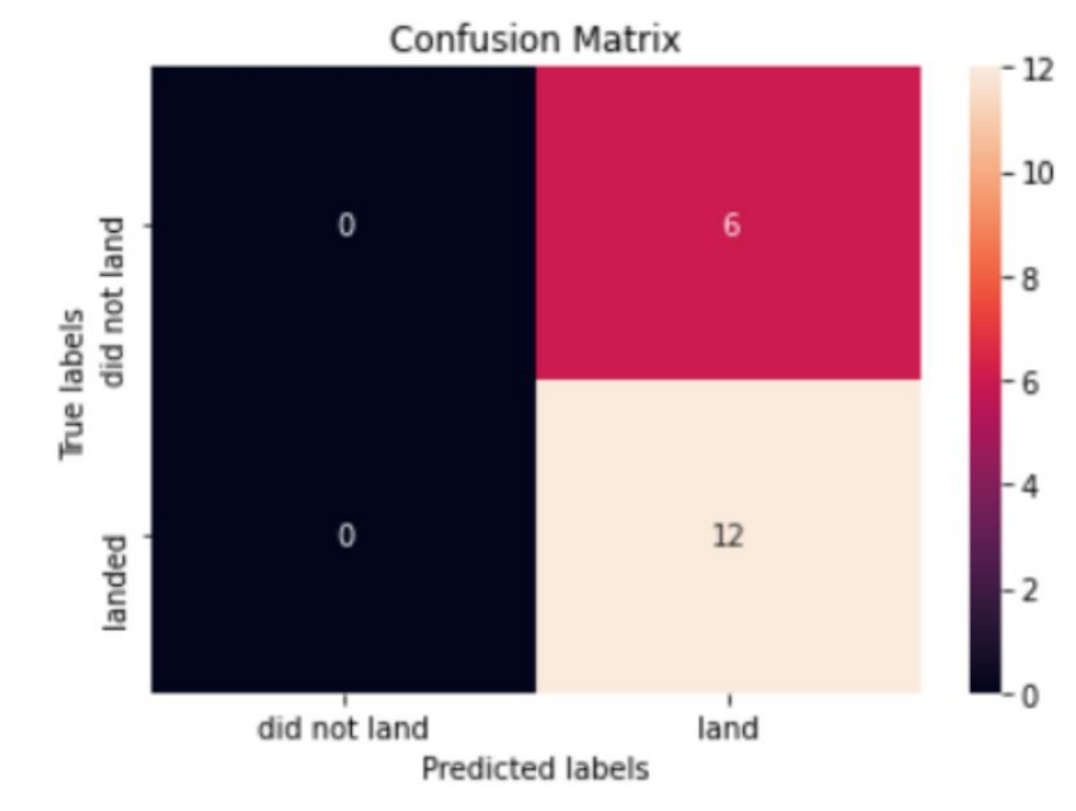


using training data

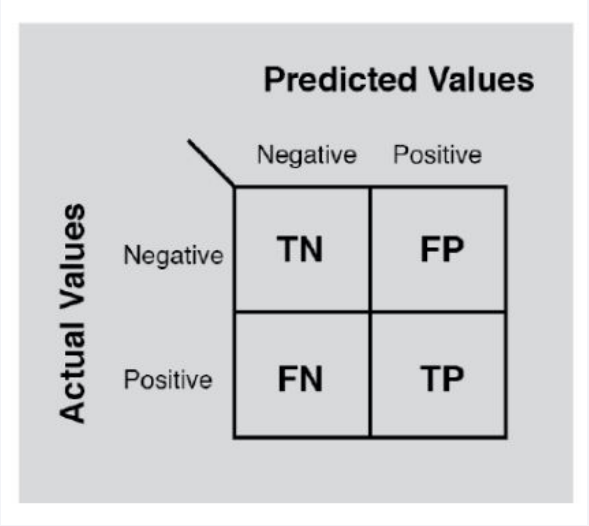
As you can see our accuracy is extremely close but we do have a winner its down to decimal places! using this function

After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 83.33% accuracy on the test data.

Confusion Matrix



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
- ADGGoogleMaps Module (not used but created)
- Module sqlserver (ADGSQLSERVER)
- PythonAnywhere 24/7 dashboard

Thank you!