Winning Space Race with Data Science

Vantuil Junior, B.Arch,
PGDip, PMP®.
08/07/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Building an Interactive Map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis Results
  - Predictive Analytics & Analysis Results

Introduction

- Project background and context

This project aimed to predict land success rates, analyzing the dataset provided in the early stages of SpaceX commissioning. To better understand the scenario, the SpaceX project seeks to launch space missions with a reduced cost of 62 million dollars instead of its competitors, costing over 165 million dollars per event. One of the project's requirements is reusing the first-stage rocket parts to reach its deliverables.

Moreover, by determining the success rate for the landing in each attempt, it's possible to foresee each mission's cost and the associated risks. The outcomes collected in this essay information are helpful for business intelligence, assuming that an alternative enterprise wants to compete against SpaceX for market share.

- Problems you want to find answers

When faced with this business case, some problems showed up for the project team, which are described below:

Which are the relevant factors for the rocket landing in a successful way.

The variables will affect the entire operation and their relationship with determining the success rate.

What are the conditions observed in SpaceX launching, getting the best results and ensuring the best success rate?

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX Rest API
  - (Web Scrapping) from Wikipedia
- Perform data wrangling
  - The hot Encoding data field for Machine Learning and dropping irrelevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting: Scatter Graphs, and 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, and evaluate classification models

Methodology

The following datasets were collected following the processes:

1. The team worked with SpaceX launch data gathered from the SpaceX REST API.
2. This API provided the launch data, including information about the rocket, payload, launch specifications, landing specifications, and landing outcome.
3. The main target is to use this data to predict the probability of each attempt resulting in landing a rocket or not.
4. The SpaceX REST API endpoints, or URL, start with api.spacexdata.com/v4/.
5. Web scraping through Wikipedia using BeautifulSoup, an essential tool for getting data sources related to Falcon 9 Launch details.

SpaceX API

Web Scraping

Data Collection - SpaceX API

Filter DF for Falcon 9 / Clean Data

Use SpaceX REST API

API returns SpaceX data in *.json

Normalize data into flat file as *.csv

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 data frame

```
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 the data frame and export it to flat file (.csv)

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

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

Data Collection - Scrapping

Extract data using beautifulsoup.

Get HTML Response from Wikipedia

Parse HTML table into a List -> dictionary.

Normalize data into flat file as *.csv

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 = []

# Apply find_all() function with `th` element on first_launch_table
first_launch_table.find_all('th')

# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for i in (first_launch_table.find_all('th')):
    name = extract_column_from_header(i)
    if ((name != None) and len(name) > 0):
        column_names.append(name)

# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

column_names
```

Data Collection - Scrapping

5. Creation of a 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

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

7. Converting dictionary to a data frame.

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

8. Data frame to a *.CSV.

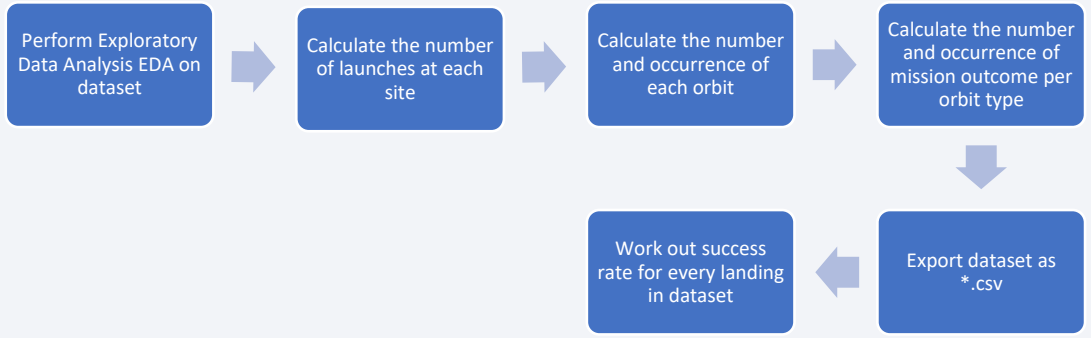
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Introduction.

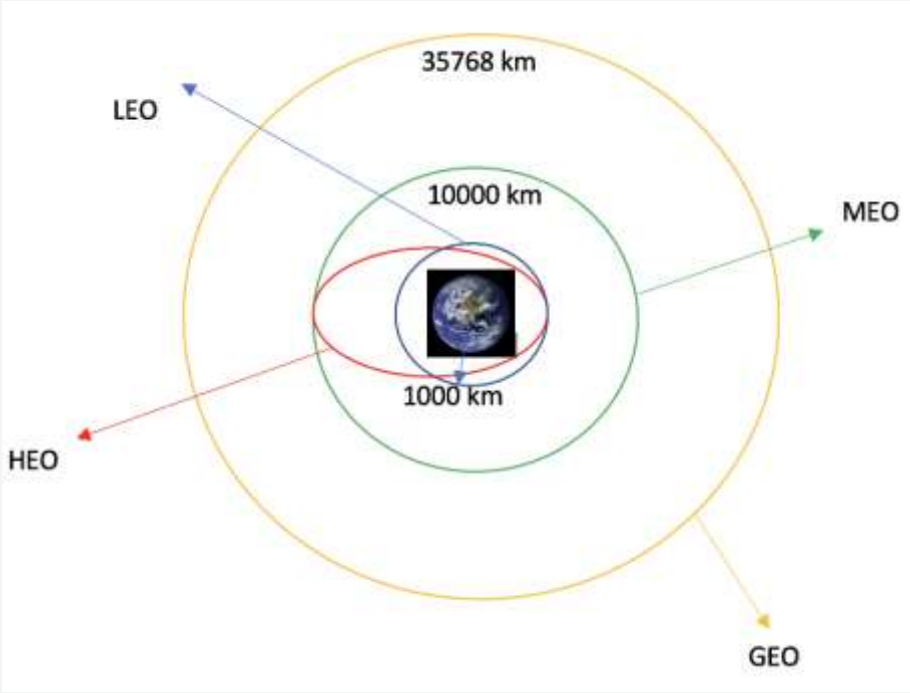
In the data set, several cases occurred where the booster part did not land successfully. In other attempts, a landing failed due to an accident; for example, True Ocean describes that the mission outcome was successfully landed to a specific ocean region, while False Ocean is the opposite, the mission outcome failed due to explosions or errors when aiming the platform. Besides that, True RTLS means the mission outcome was successfully landed on a ground pad, and False RTLS was observed when the mission outcome was unsuccessfully landed on a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship, while False ASDS the opposite. The team worked on converting those outcomes into Training Labels, with 1 representing the booster successfully landed and 0 means it was unsuccessful.

Process

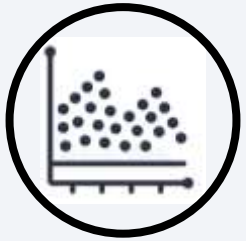


Orbit types aimed at each launching:

Diagram showing the common orbit types SpaceX uses.



EDA with Data Visualization



Scatter Graphs were built providing the following information:

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

As defined, 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.

- Bar Graphs were built providing the following information:

Mean VS. Orbit

A bar diagram makes it easy to compare data sets between different groups at a glance. The goal is to show the relationship between the two axes. Bar charts can also offer significant changes in data over time.

- Line Graphs were built providing the following information:

Success Rate VS. Year

Line graphs are helpful because they clearly show data variables and trends and can help make predictions about the results of data not yet recorded.

EDA with SQL

Performed SQL queries to gather information about the dataset.

For an example of some questions, the data analysts performed SQL queries to get the answers in the dataset:

- Displaying the names of the unique launch sites in the space mission
- Displaying five 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 the drone ship was achieved.
- Listing the names of the boosters which have success in the ground pad and have payload mass more significant than 4000 but less than 6000
- Listing the total number of successful and failed 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 the ground pad, booster versions, and launch_site for the months in the year 2017
- Ranking the count of successful landing_outcomes between 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

The next step was assigned to the data frame launch_outcomes(failures, successes) classes 0 and 1 with Green and Red markers on the map in a MarkerCluster().

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

Aiming to visualize the Launch Data into an interactive map, the data analysts collected the Latitude and Longitude Coordinates for each launch site. Then a circle marker was added around each launch site with a label of the name of the launch site.

Example of some trends in which the Launch Site is situated.

- Are launch s it es near railways? No.
- Are launch s it es near highways? No.
- Are launch s it es near the coastline? Yes.
- Does launch s it es keep a certain distance away from cities? Yes.

Build a Dashboard with Plotly Dash

Handling Python Anywhere to host the website live 24/7 so it's possible to access the data.

- The dashboard is built with Flask and Dash web
- Graphs:
  - Pie Chart showing the total launches by a certain site/all sites
  - Display relative proportions of multiple classes of data.
  - The circle's size can be proportional to the total quantity it represents.

Scatter Graph shows the relationship between Outcome and Payload Mass (Kg) for the different Booster Versions..

- It shows the relationship between two variables.
- It is the best method to demonstrate a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, was 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

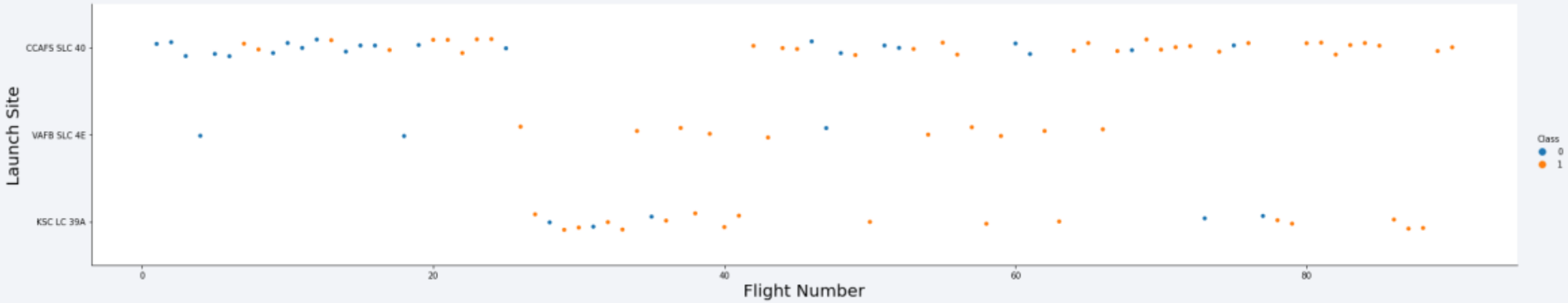
- The model with the best accuracy score wins the best performing model
- In the notebook, there is a dictionary of algorithms with scores at the bottom of the notebook.

Results

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

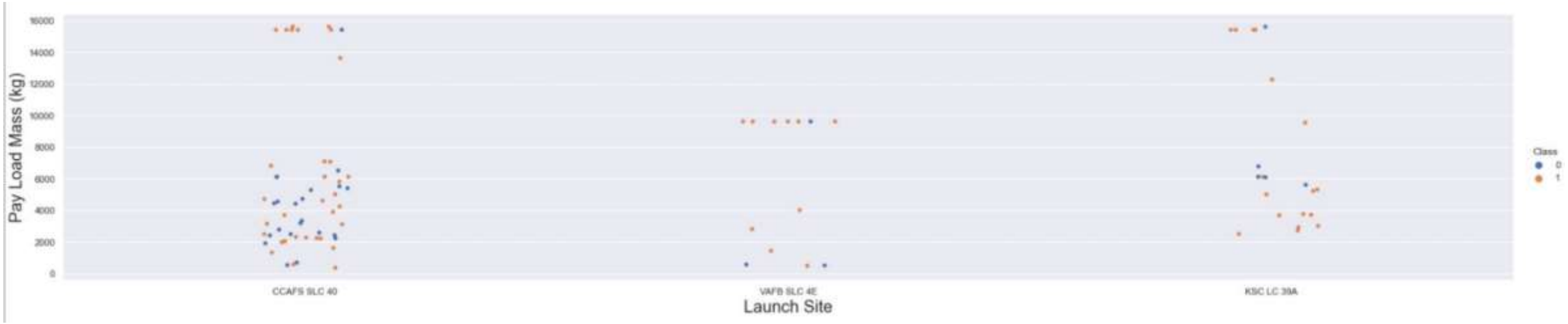
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

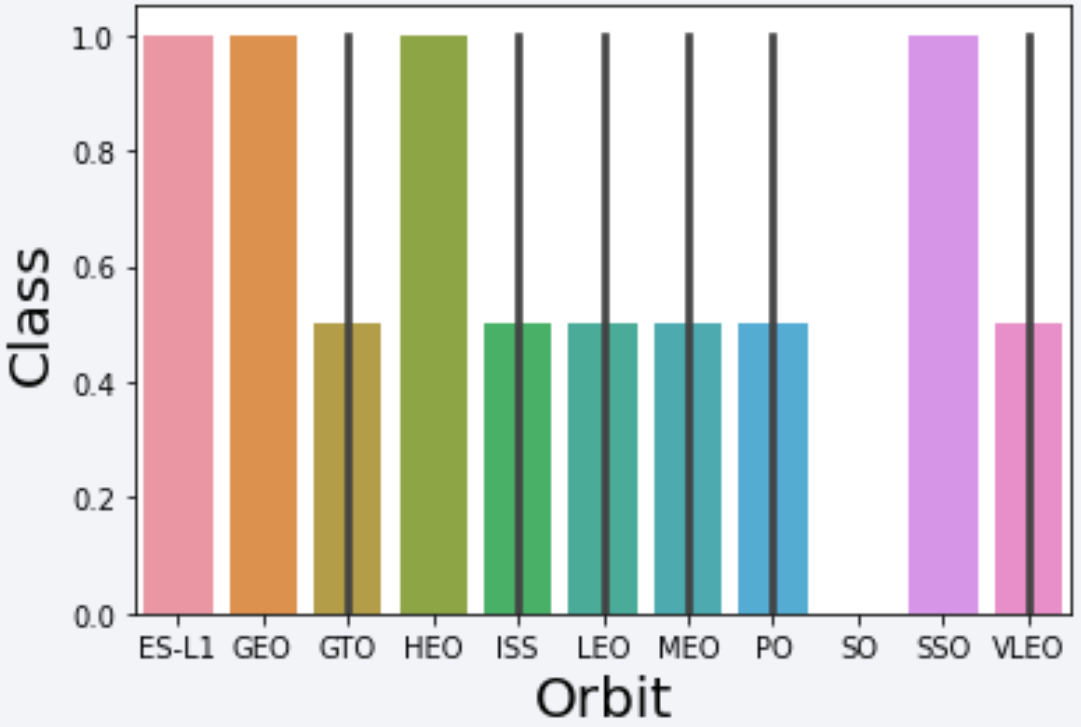
- The graph above demonstrates that the landing site CCAFS SLC 40 had more successful launches than the number of flights.



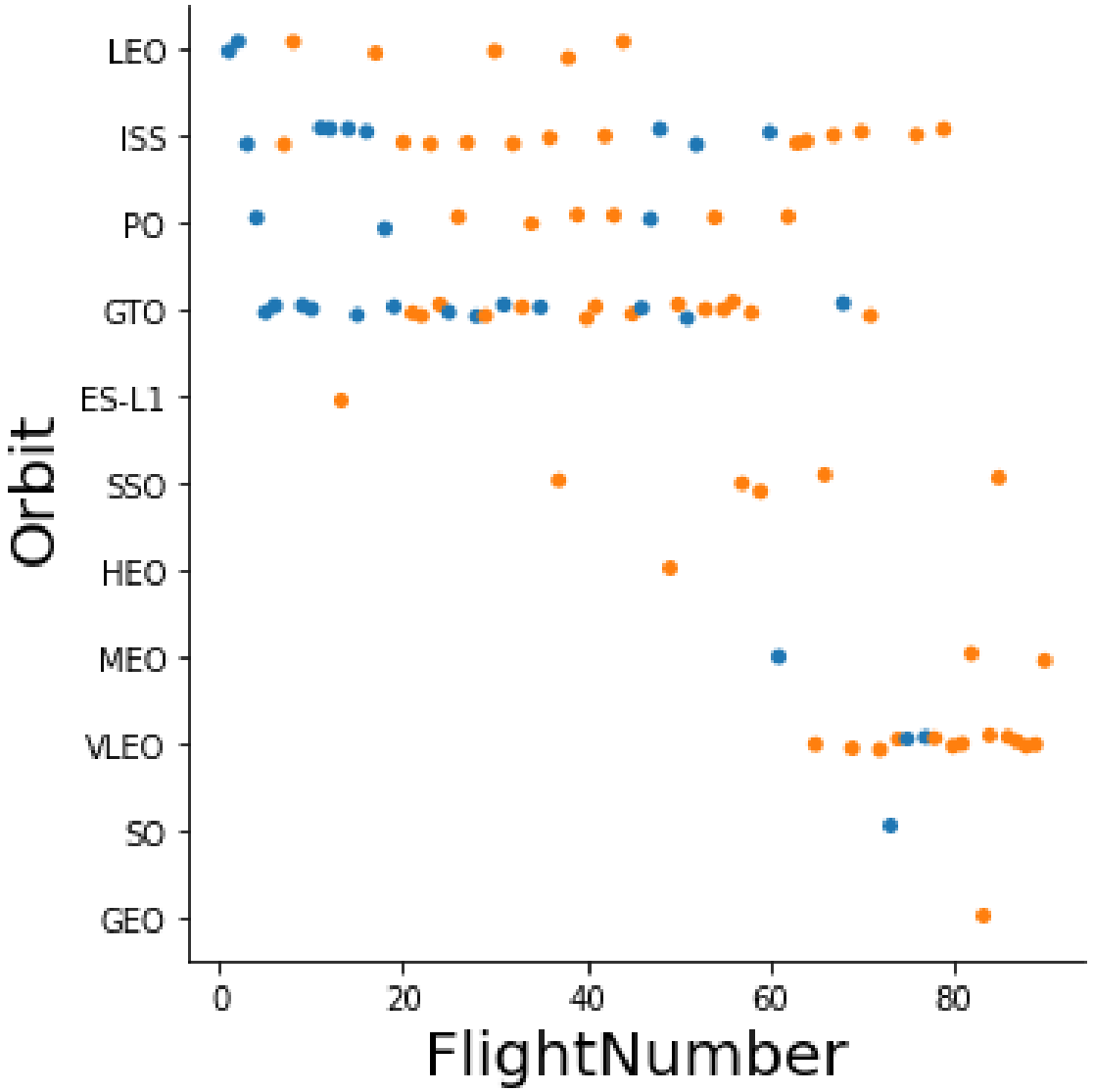
Payload vs. Launch Site

- The greater the payload mass for Launch Site CCAFS SLC 40, the higher the success rate for the Rocket. There is not a pretty clear pattern to be found using this visualization to decide if the Launch Site depends on Pay Load Mass for a successful launch.

Success Rate vs. Orbit Type



Orbit GEO, HEO, SSO and ES-L1 achieved the best Success Rate.

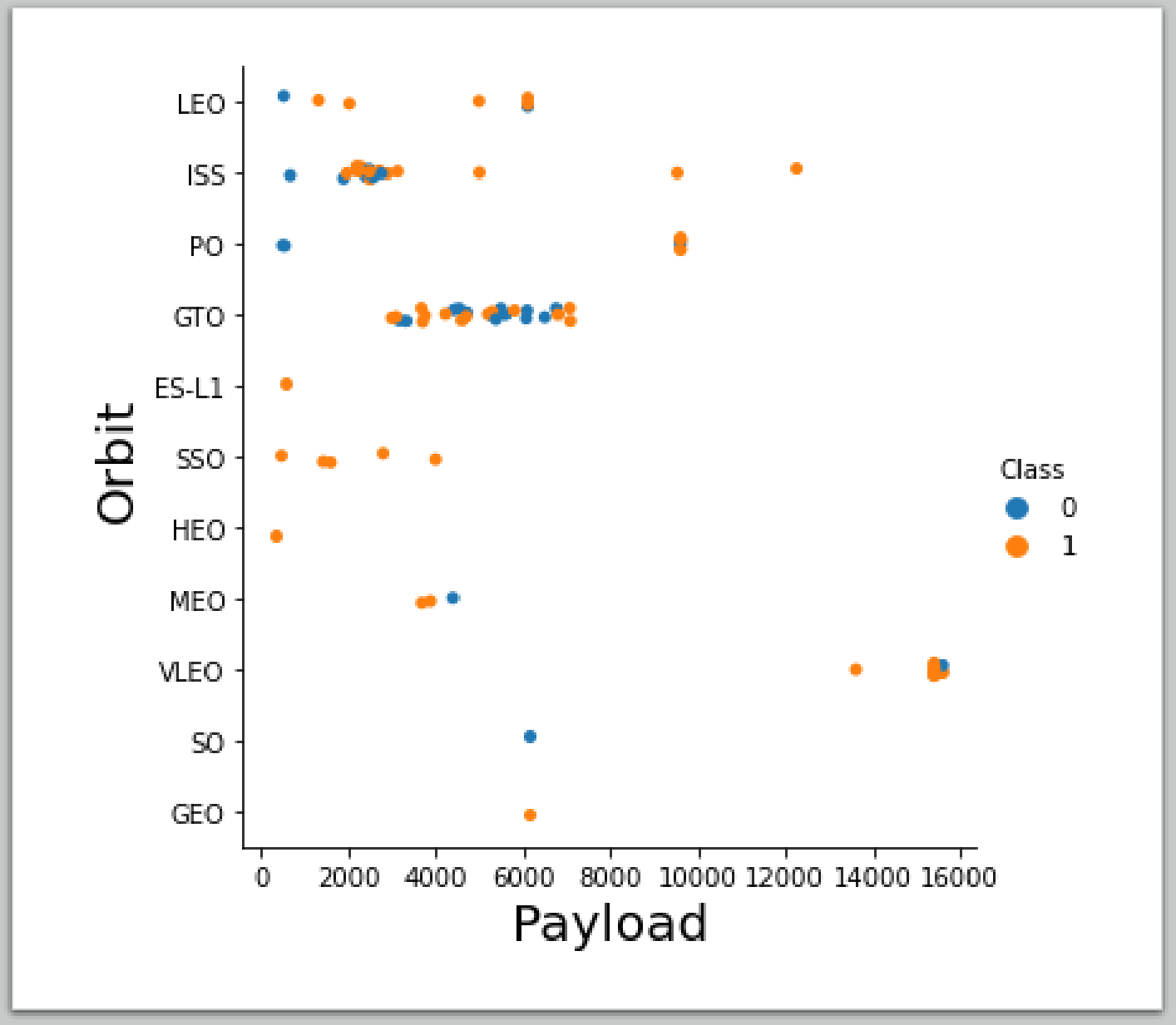


Flight Number vs. Orbit Type

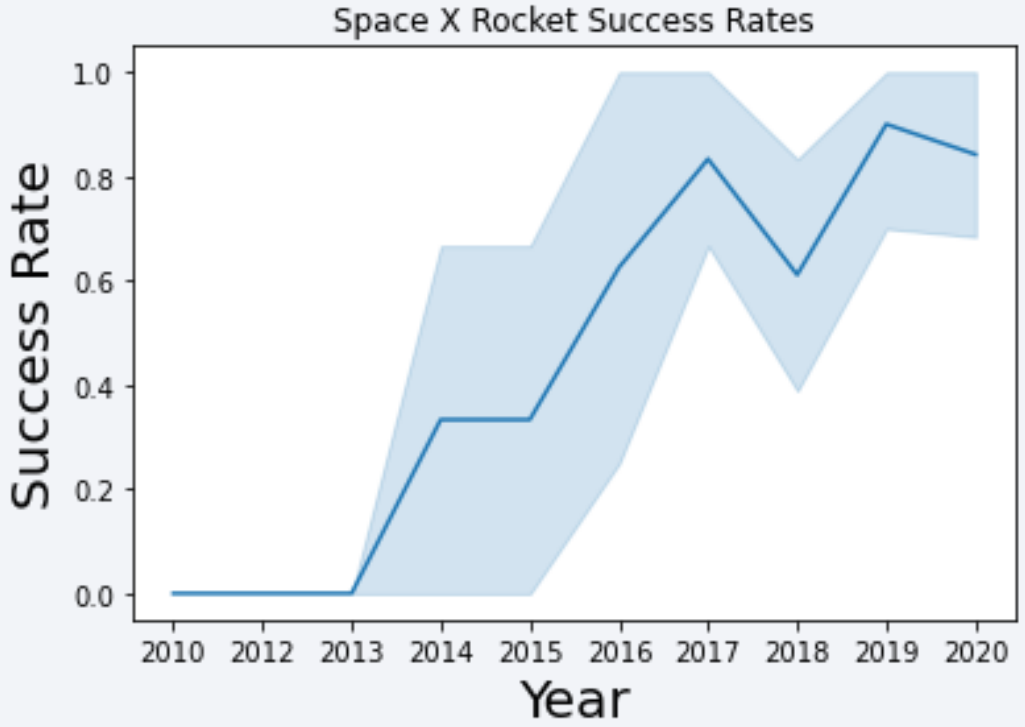
- In the LEO orbit, the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight numbers when in GTO orbit.

Payload vs. Orbit Type

- There is a clear correlation between Heavy payloads having a negative influence on GTO orbits and a positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend



This graph shows that the success rate has constantly started climbing from 2013 to 2020.

All Launch Site Names

- Using the function magic SQL, it was possible to collect all launching sites' names from the data frame.

Display the names of the unique launch sites in the space mission

In [5]:
```
%sql select distinct(LAUNCH_Site) from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [6]:
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[6]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Furthermore, using the same command selected from the SPACEXTBL data frame, it was possible to filter all sites with the initials "CCA" using the separator "LIKE."

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

In [7]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[7]:

sum(PAYLOAD_MASS__KG_)
45596

- One task addressed was regarding displaying the total mass. To solve this question, I handled the select command summing (PAYLOAD_MASS__KG_) column from SPACEXTBL.

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

In [8]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION LIKE '%F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[8]:

avg(PAYLOAD_MASS__KG_)
2928.4

- Unlike the previous task, it now added the command AVG on the same column to collect the average payload mass.

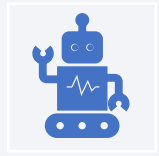
First Successful Ground Landing Date

EXPLANATION:

Using the function MIN works out the minimum date in the

column Date

The WHERE clause filters the dataset only to perform



calculations on Landing_Outcome Success (drone ship).

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)';
```

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- QUERY EXPLANATION:
- 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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND 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

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or \
MISSION_OUTCOME = 'Failure (in flight)'
```

```
 * sqlite:///my_data1.db
Done.
```

Successful_Mission_Outcomes	Failure_Mission_Outcomes
100	1

- QUERY EXPLANATION:
- It was hard to find a satisfactory outcome; it used subqueries here to produce the results. The ‘%foo%’ wildcard shows that the foo phrase is in any part of the string in the records, for example.

Boosters Carried Maximum Payload

- QUERY EXPLANATION:
- Using the word DISTINCT in the query demonstrate that it will only show Unique values in the Booster_Version column from tblSpaceX.
- GROUP BY puts the list in order set to a specific condition.
- DESC means it's arranging the dataset in 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

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

2015 Launch Records

- 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 2017.

SQL 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)) = '2017')
```

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 FT B1029.1	VAFB SLC-4E	Success (drone ship)
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
March	F9 FT B1021.2	KSC LC-39A	Success (drone ship)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1029.2	KSC LC-39A	Success (drone ship)
June	F9 FT B1036.1	VAFB SLC-4E	Success (drone ship)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
August	F9 FT B1038.1	VAFB SLC-4E	Success (drone ship)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
October	F9 B4 B1041.1	VAFB SLC-4E	Success (drone ship)
October	F9 FT B1031.2	KSC LC-39A	Success (drone ship)
October	F9 B4 B1042.1	KSC LC-39A	Success (drone ship)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

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

SQL QUERY

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

QUERY EXPLANATION

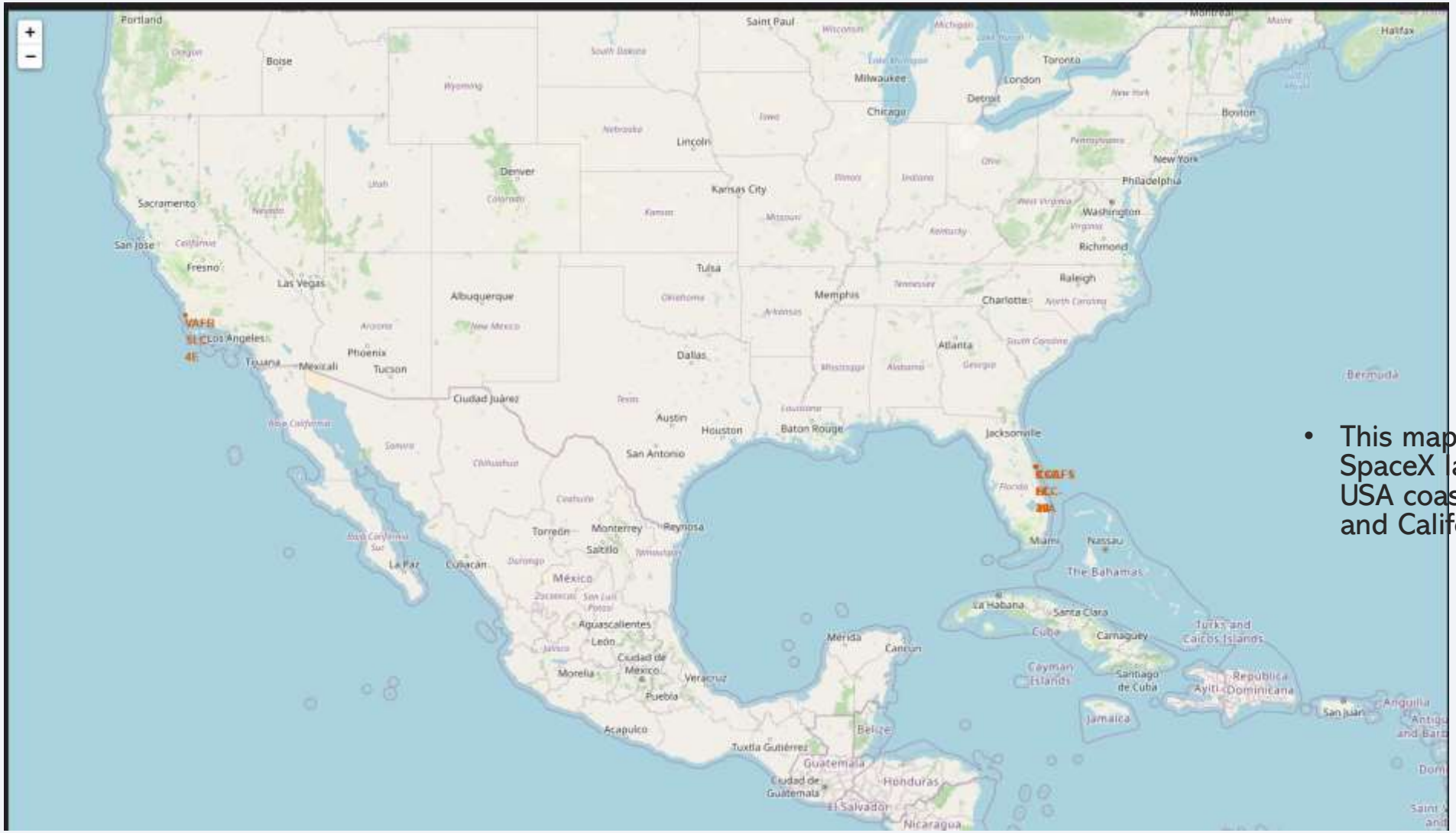
- Function COUNT counts records in column WHERE filters data LIKE (wildcard) AND (conditions) AND (conditions)

	Successful Landing Outcomes Between 2010-06-04 and 2017-03-20
0	34

Section 3

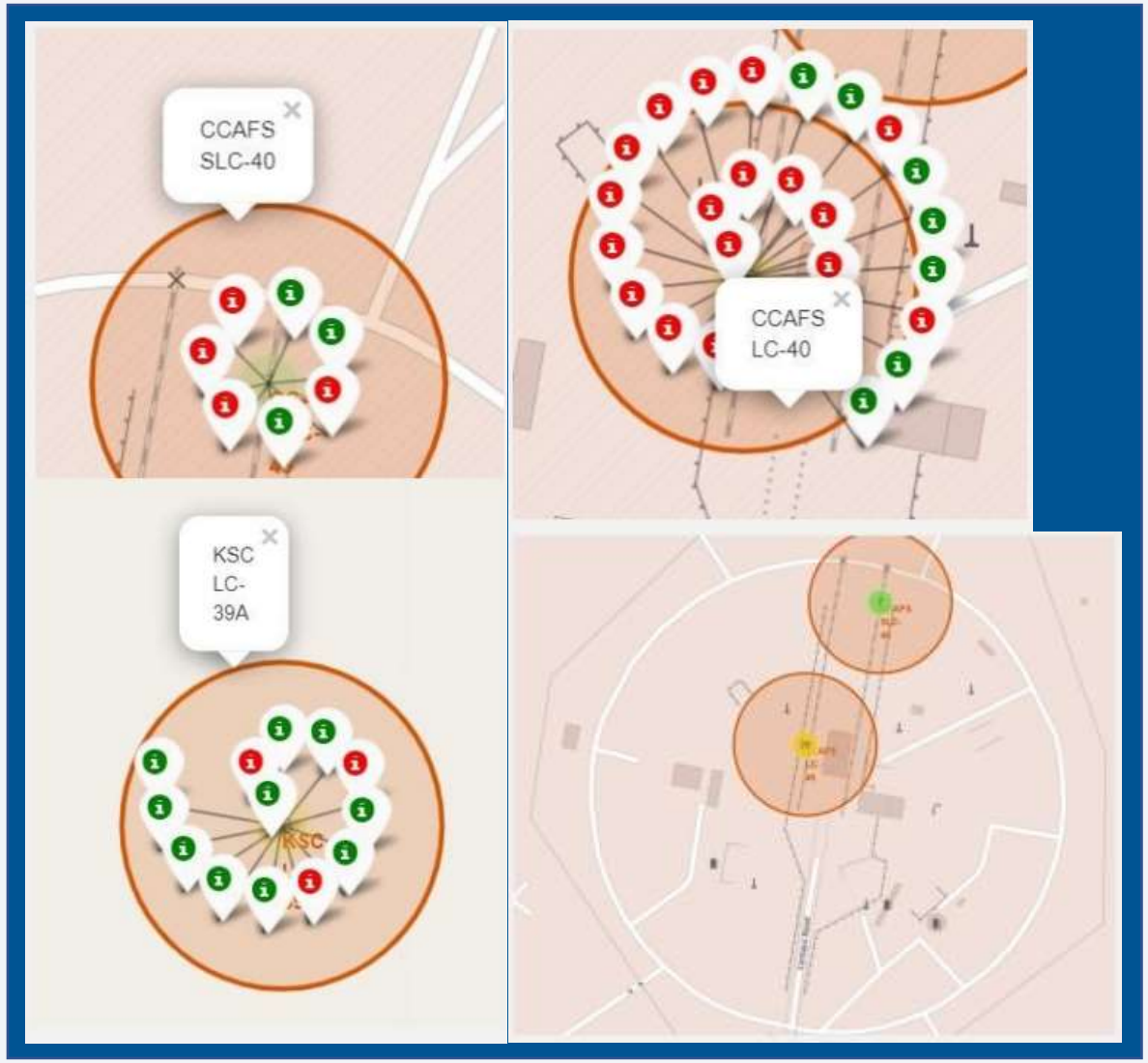
Launch Sites Proximities Analysis

<Mark all launch sites on a map>



- This map emphasizes that the SpaceX launch sites are on the USA coasts, precisely Florida and California states.

<Mark the success/failed launches for each site on the map>

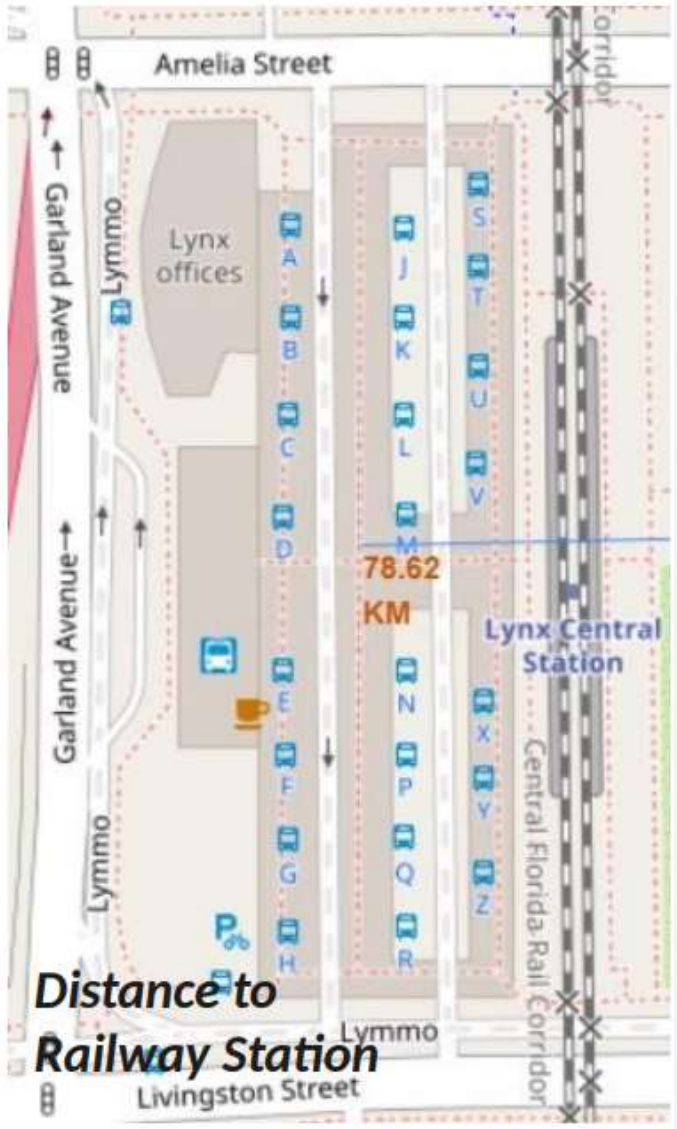


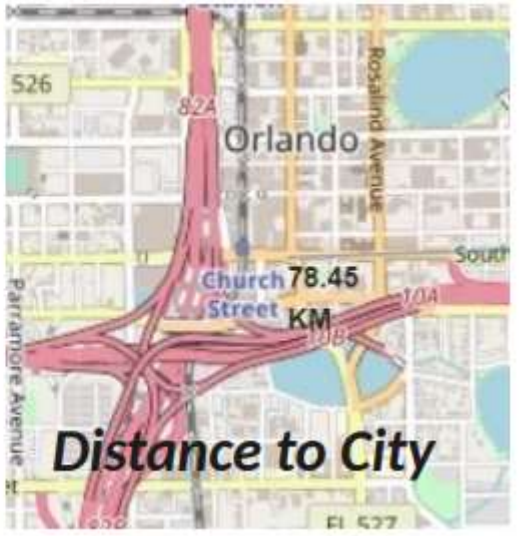
Florida Launch Sites:

Green Marker shows successful Launches and Red Marker shows Failures

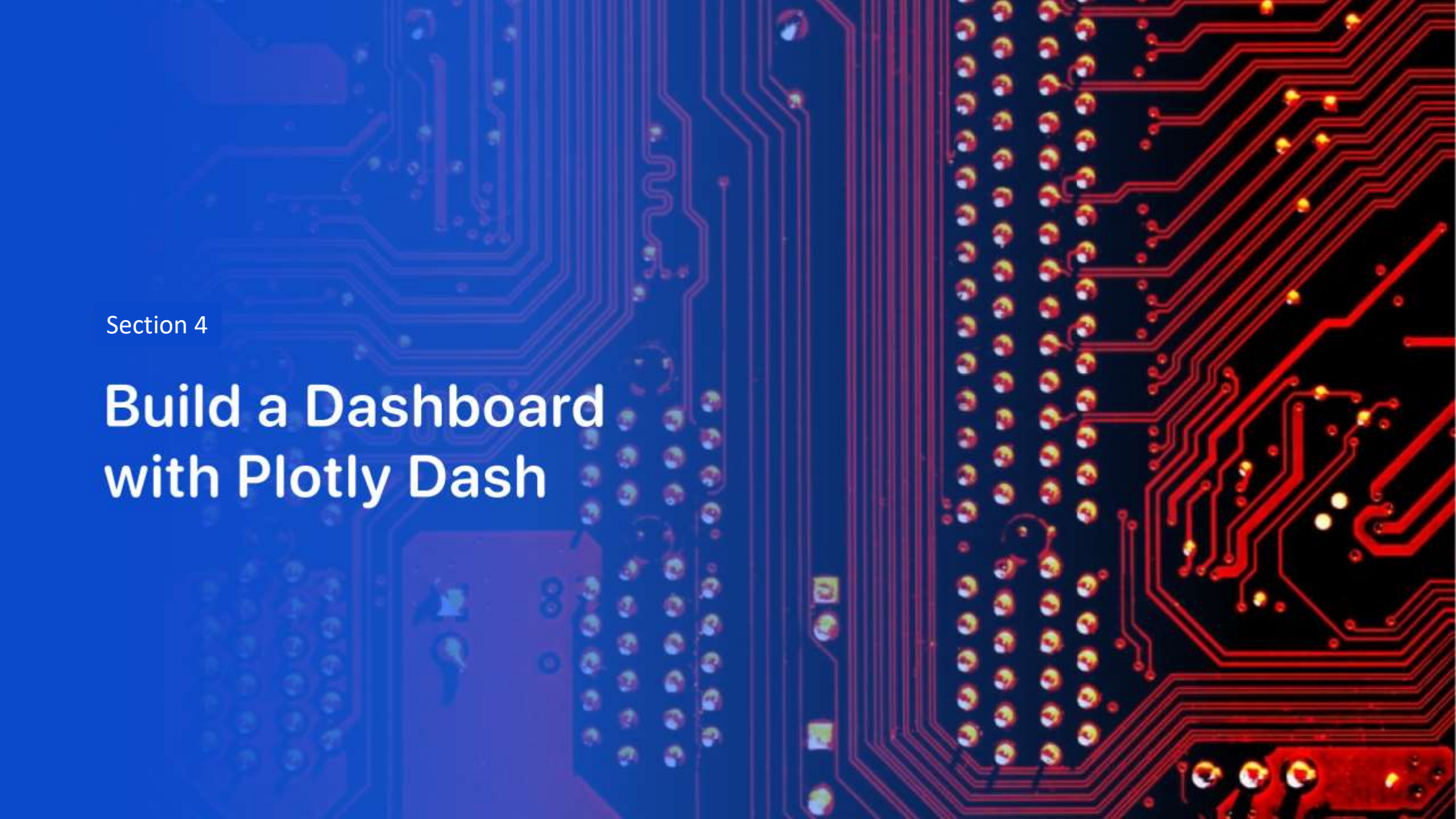
California Launch Sites:

<Calculate the distances between a launch site to its proximities>



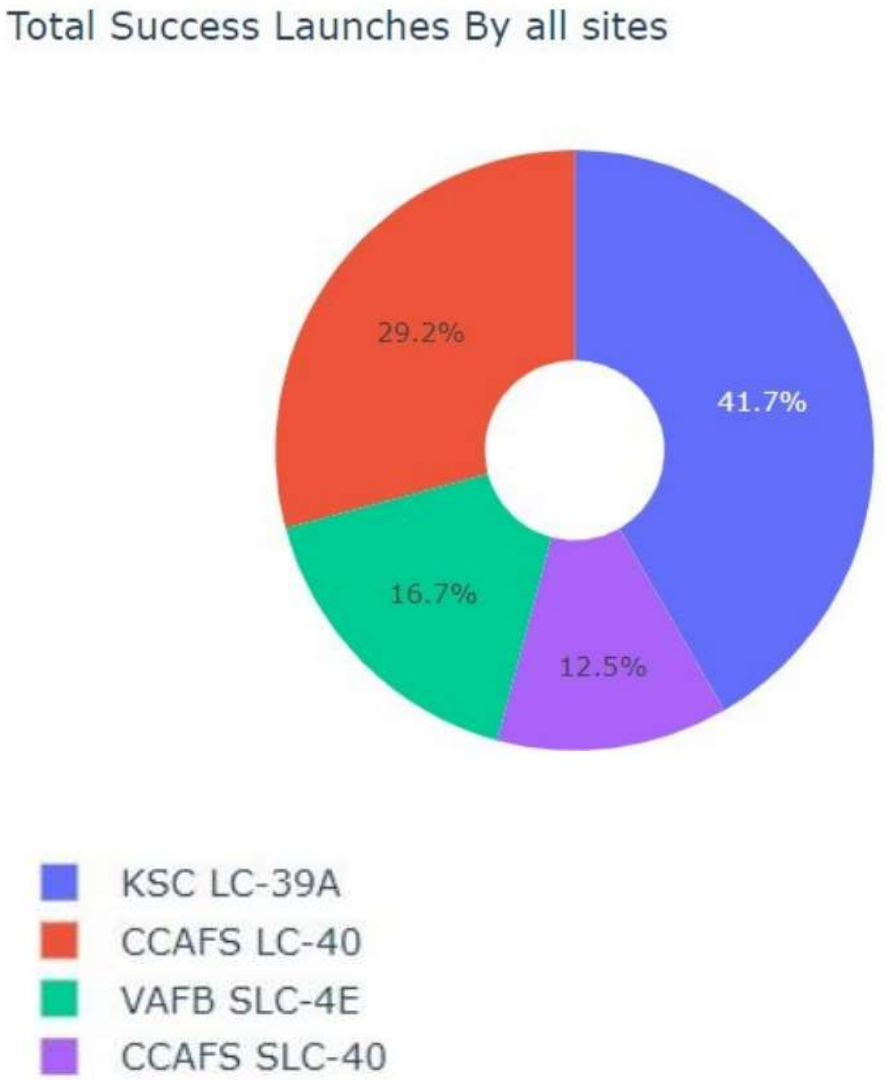


The team worked on the Launch Sites' distance to landmarks to find trends with the Haversine formula using CCAFS-SLC-40 as a reference.

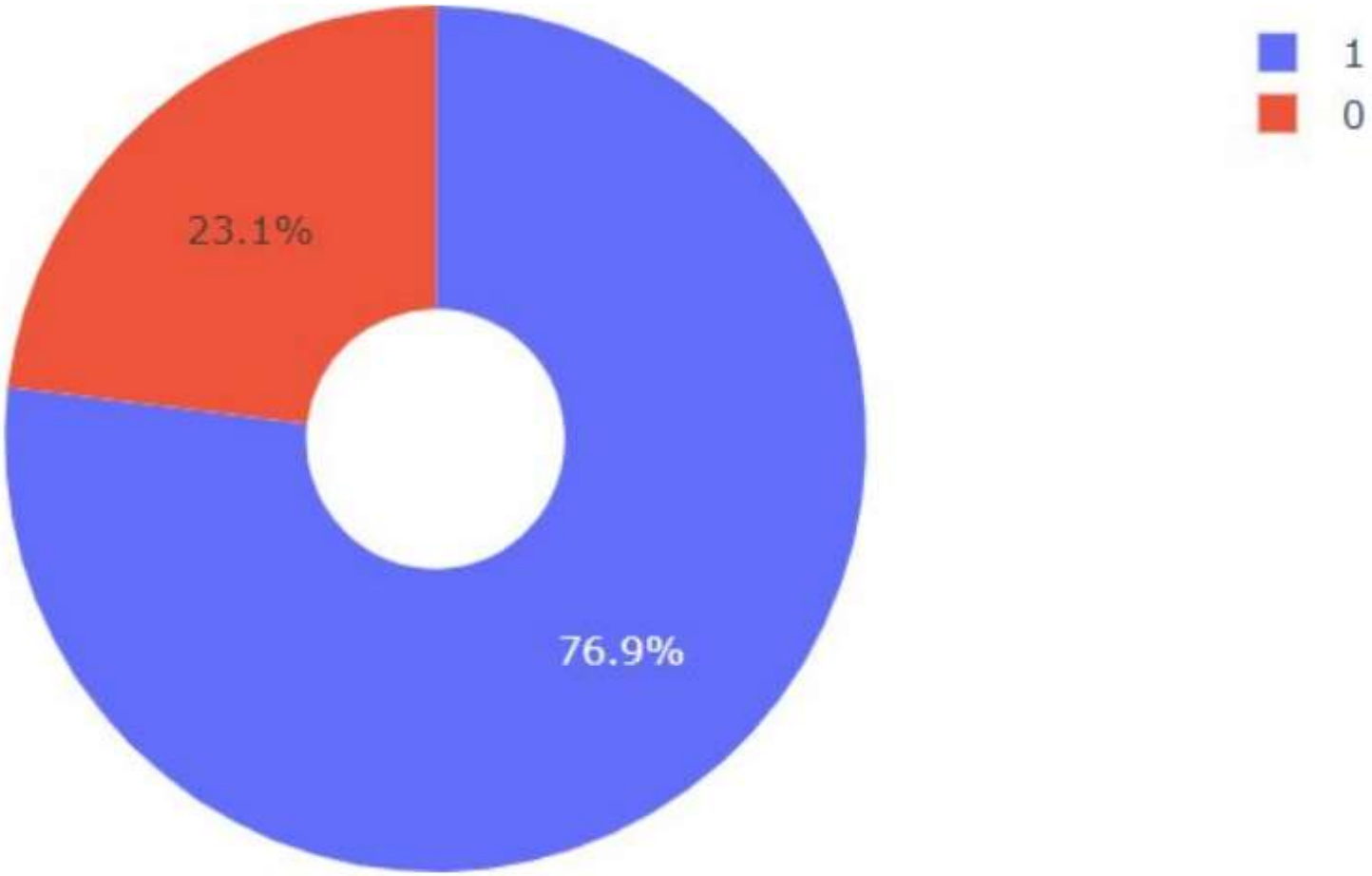

Section 4
Build a Dashboard with Plotly Dash

<Pie chart showing the success percentage achieved by each launch site.>

- The Pie chart demonstrates the successful percentages for launches on all the sites.

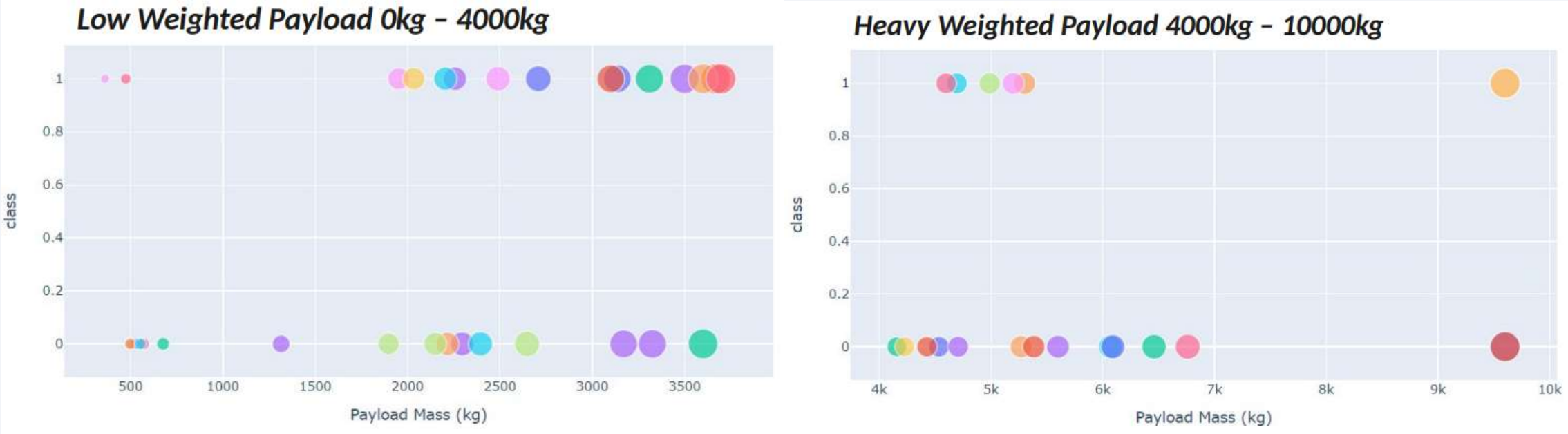


<Pie chart for the launch site with the highest launch success ratio.>



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

<Payload vs. Launch Outcome scatter plot for all sites, with different loads selected in the range slider.>

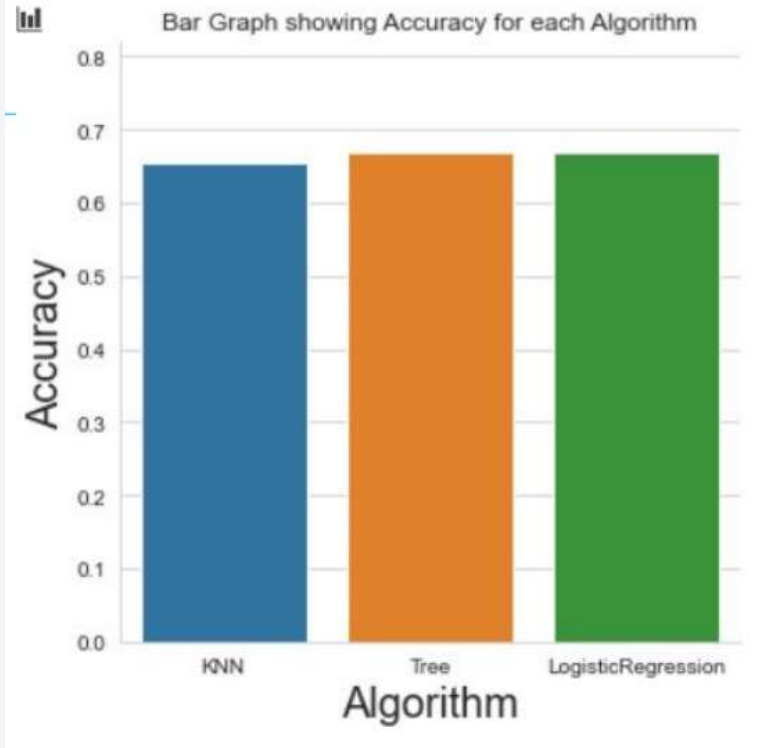


An essential tool to identify the success rates for low weighted payloads is higher than the heavily weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy




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

	Accuracy	Algorithm
0	0.653571	KNN
1	0.667857	Tree
2	0.667857	LogisticRegression

- The tree algorithm suppresses the others showing the highest accuracy.

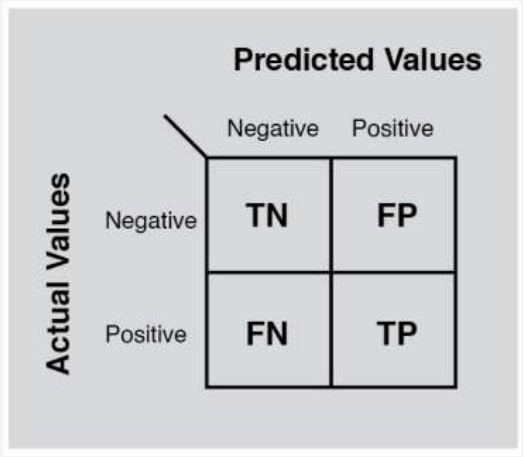
In this part of the report, it was checked the accuracy presented by the algorithms is exceptionally close, but we do have a winner. It's down to decimal places using the function below.

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

```
Best Algorithm is Tree with a score of 0.6678571428571429
Best Params is : {'criterion': 'gini', 'max_depth': 2, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
```

Confusion Matrix

- The logistic regression can distinguish between the different classes by examining the confusion matrix. The significant problem is false positives.



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset.
- • Low weighted payloads perform better than heavier payloads.
- • The success rates for SpaceX launches are 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, and ES-L1 has the best Success Rate.

Thank you!