IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michelle Ng
July 27, 2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data collection
  - Data wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Build an interactive map with Folium
  - Build an Interactive Dashboard with Ploty Dash
  - Predictive analysis using Machine Learning

- **Summary of all results**
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis result

Introduction

- **Project background and context**

- SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars while other providers cost upward of 165 million dollars each.
- The cost is lower due to SpaceX can reuse the first stage. Hence, in order to determine cost of a launch, we need to determine if the first stage will land.
- For this project for SpaceY, we want to determine the price of launch and use the SpaceX information to predict if the Falcon 9 first stage will land successfully.

- **Problems you want to find answers**

- What are the success rate and what attributes contribute to the success rate?
- What are the patterns of the successful launches and what are the proximities of the launches to other sites e.g. highway, railway, city?
- Which model with best accuracy to predict if the first stage of the Falcon 9 lands successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX launch data was collected from the SpaceX REST API api.spacexdata.com/v4/.
  - This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Perform data wrangling
  - Data was processed to show outcome of the first stage whether successfully landed by using 0 (Unsuccessful) and 1 (Successful)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We will build the classification model and train the model by perform Grid Search and calculate the best accuracy (Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors) and output the confusion matrix.

Data Collection

Git Hub URL for Data Collection

1. Request and parse the SpaceX launch data using the url https://api.spacexdata.com/v4/launches/past

2. Convert and normalize the response using json

3. Call API and get info from each launches using ID from columns rocket, payloads, launchpad, and cores

4. Create data frame with the relevant data from step 3 and filter only to include Falcon 9 data

5. Replacing the missing value for Payload Mass with the mean

6. Export to CSV file

Data Collection – SpaceX API

Git Hub URL for Data Collection

1. Request and parse the SpaceX launch data using the url https://api.spacexdata.com/v4/launches/past

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [8]: response = requests.get(spacex_url)
```

2. Convert and normalize the response using json

```
In [12]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

3. Call API and get info from each launches using ID from columns rocket, payloads, launchpad, and cores

```
In [23]: 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

[Git Hub URL for Data Collection]

4. Create data frame with the relevant data from step 3 and filter only to include Falcon 9 data

```
In [30]: # Create a data from launch_dict
         df = pd.DataFrame(launch_dict)
```

```
In [36]: # Hint data['BoosterVersion']!='Falcon 1'
         data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

5. Replacing the missing value for Payload Mass with the mean

```
In [39]: # Calculate the mean value of PayloadMass column
         mean=data_falcon9['PayloadMass'].mean()

         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'].replace(np.nan,mean, inplace=True)
```

6. Export to CSV file

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

Data Collection - Web Scraping

Git Hub URL for Data Collection with Web Scraping

1. Extract a Falcon 9 launch records HTML table from Wikipedia

2. Create a BeautifulSoup object from the HTML response

3. Extract all column/variable names from the HTML table header

4. Create a data frame by parsing the launch HTML tables

5. Export to CSV file

Data Collection - Web Scraping

[Git Hub URL for Data Collection with Web Scraping]

1. Extract a Falcon 9 launch records HTML table from Wikipedia

```
In [5]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [6]: # use requests.get() method with the provided static_url

        # assign the response to a object
        html_data = requests.get(static_url).text
```

2. Create a BeautifulSoup object from the HTML response

```
In [7]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data, 'html.parser')
```

3. Extract all column/variable names from the HTML table header

```
In [22]: launch_dict= dict.fromkeys(column_names)

         # Remove an irrelvant column
         del launch_dict['Date and time ( )']

         # Let's initial the launch_dict with each value to be an empty list
         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'] = []
         # Added some new columns
         launch_dict['Version Booster']=[]
         launch_dict['Booster landing']=[]
         launch_dict['Date']=[]
         launch_dict['Time']=[]
```

Data Collection - Web Scraping

[Git Hub URL for Data Collection with Web Scraping]

4. Create a data frame by parsing the launch HTML tables

```
In [27]: df=pd.DataFrame(launch_dict)
```

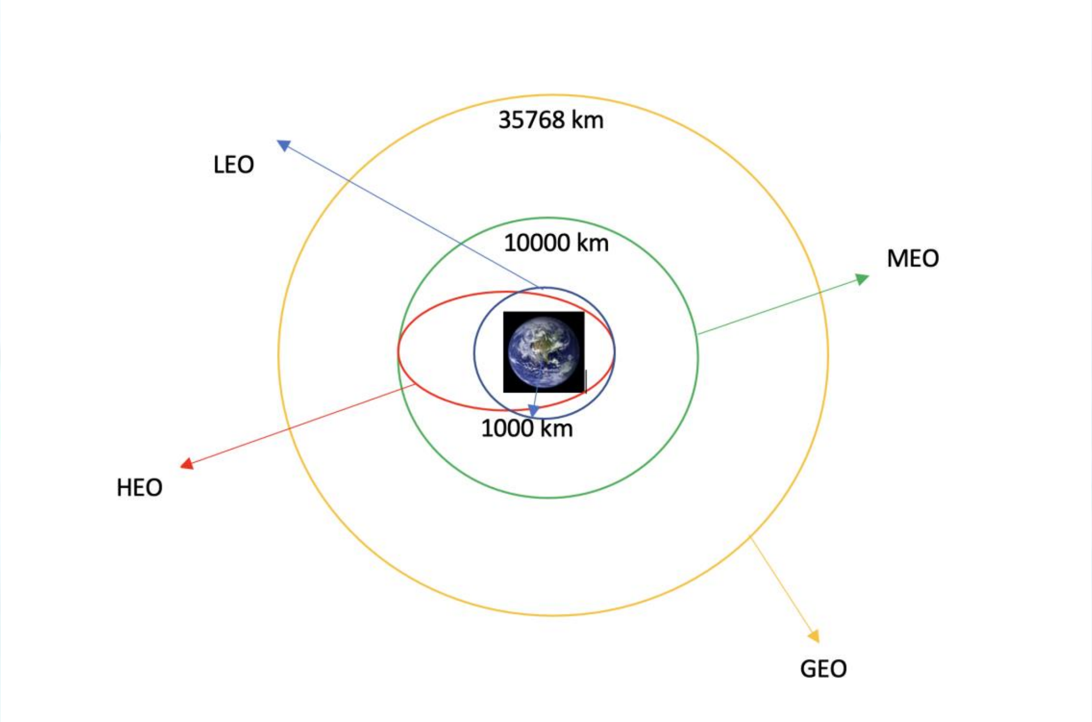
5. Export to CSV file

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

Data Wrangling

Git Hub URL for Data Wrangling

1. Calculate the number of launches on each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column
5. Determine Success Rate based on Outcome
6. Export to a CSV



EDA with SQL

Git Hub URL for Exploratory Data Analysis with SQL

10 SQL Queries that performed to get more information on the landing and its outcome

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA‘
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
10. Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

EDA with Data Visualization

[Git Hub URL for Exploratory Data Analysis with Data Visualization]()

- Scatter Graph
    - Flight Number vs. Payload Mass
    - Flight Number vs Launch Site
    - Payload and Launch Site
    - Flight Number and Orbit type
    - Payload and Orbit type

- Bar Chart
    - Success rate vs orbit type.

- Line Chart
    - Launch Success vs Year

Build an Interactive Map with Folium

[Git Hub URL for Build Interactive Map with Folium]()

- **Mark all launch sites on a map**
  - ❖ Added Marker with Circle, Popup Label and Text Label for initial center location to be NASA Johnson Space Center at Houston, Texas
  - ❖ Added Markers with Circle, Popup Label and Text Label for each site's location on a map using site's latitude and longitude coordinates

- **Mark the success/failed launches for each site on the map**
  - ❖ Use Marker clusters to simplify a map containing many markers having the same coordinate.
  - ❖ Create markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)

- **Calculate the distances between a launch site to its proximities**
  - ❖ Show the distance and draw a Poly Line between a launch site to the selected coastline/highway/railway/city point

Build a Dashboard with Plotly Dash

[Git Hub URL for Build Dashboard with Plotly Dash]

- Add a Launch Site Drop-down Input Component
- Add Pie Chart and a callback function to render the pie chart visualizing launch success counts based on selected site dropdown
- Add a Range Slider to Select Payload (Min 0 (Kg) to Max 10000 (Kg))
- Add Scatter diagram and a callback function to render the scatter diagram visualizing how payload may be correlated with mission outcomes for selected site(s).
- To color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters

Predictive Analysis (Classification)

Git Hub URL for Predictive Analysis

1. Create a NumPy array from the column Class

2. Standardize and transform the data

3. Use the function train_test_split to split the data X and Y into training and test data.

4. Create a logistic regression/SVM/DecisionTree/KNN object then create a GridSearchCV object with cv = 10 to find the best parameters

5. Calculate the accuracy on the test data using the method score

6. Plot the confusion matrix

7. Find the method performs best using accuracy, F1 score and jaccard score

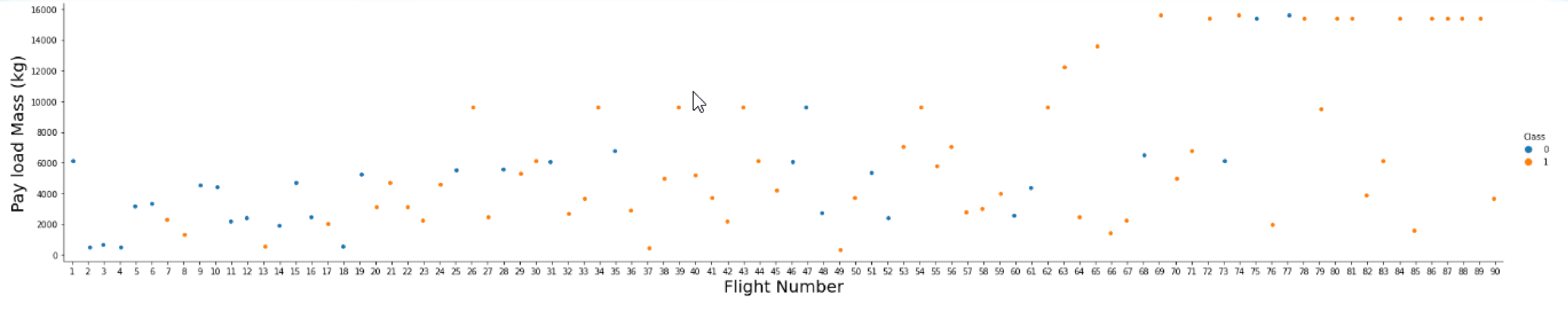
Results

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

Section 2

Insights drawn from EDA

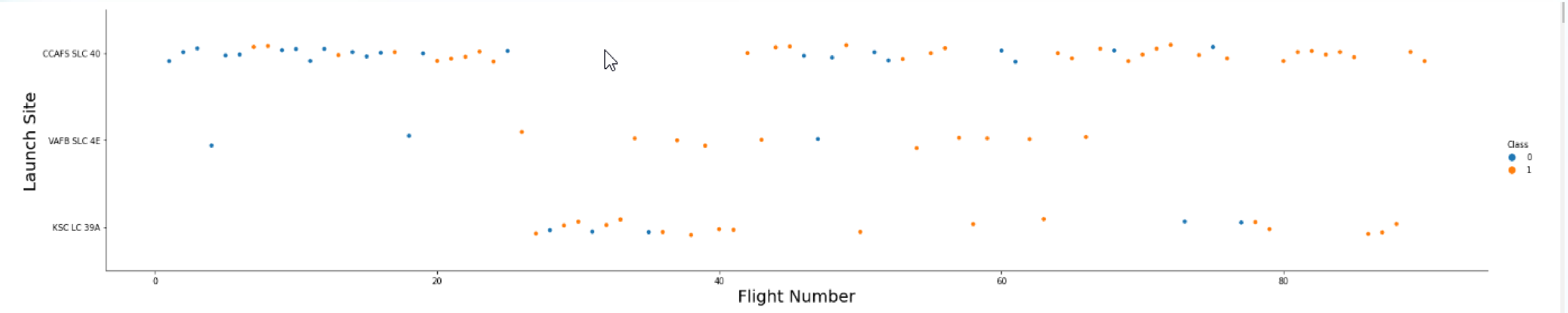
Flight Number vs. Payload Mass



Observation

- As the flight number increases, the first stage is more likely to land successfully.
- The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

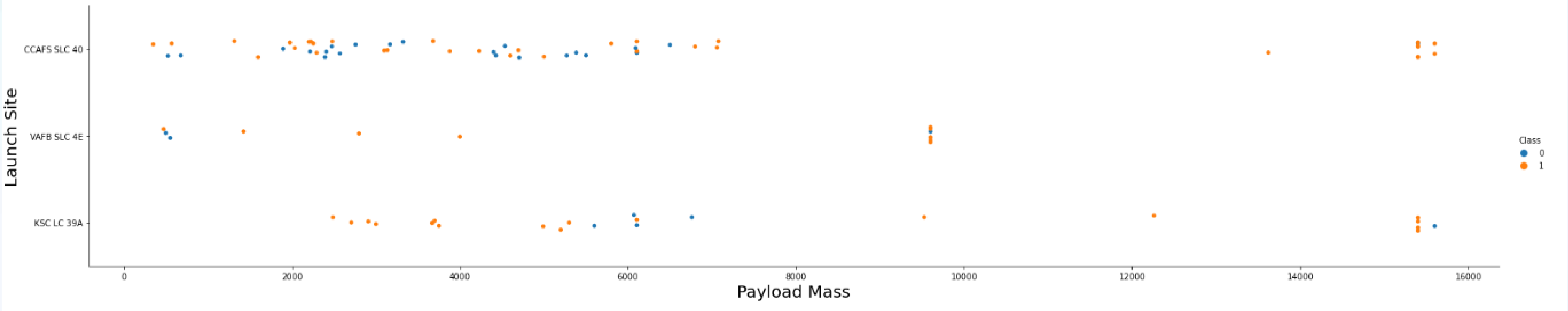
Flight Number vs. Launch Site



Observation

- CCAFS SLC-40 seems to have the most volume of flights.
- The larger the flight amount at a launch site, the greater the success rate at a launch site

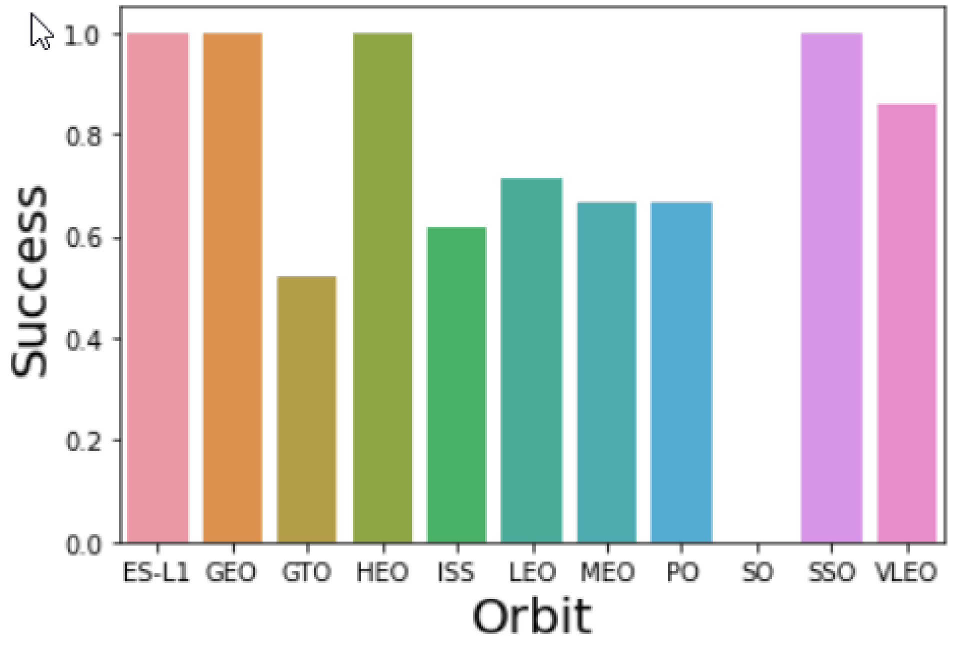
Payload vs. Launch Site



Observation

- For VAFB-SLC launch site, there are no rockets launched for heavy payload mass (greater than 10000).

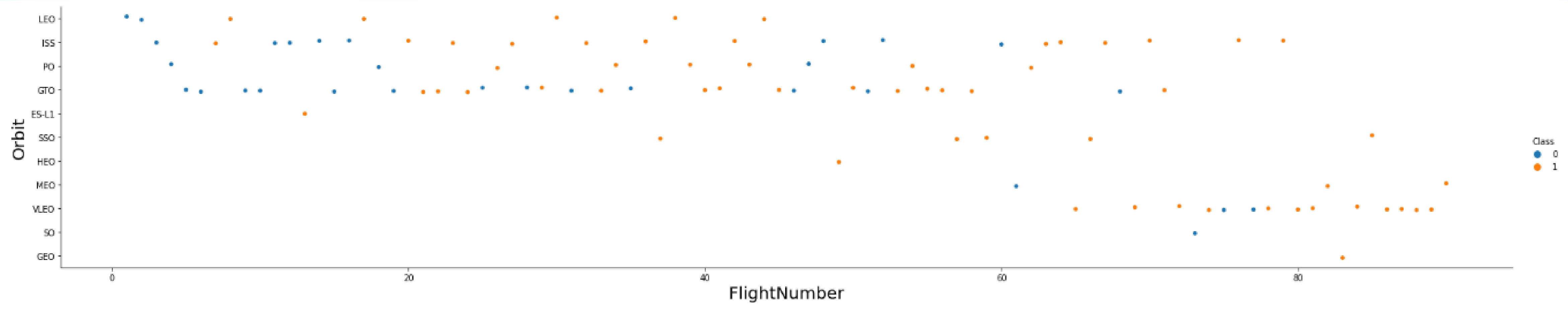
Success Rate vs. Orbit Type



Observation

- Orbit ES-L1, GEO, HEO and SSO have 100% success rate
- Orbit SO has zero success rate
- Others Orbit except the abovementioned has 50% - 70% success rate

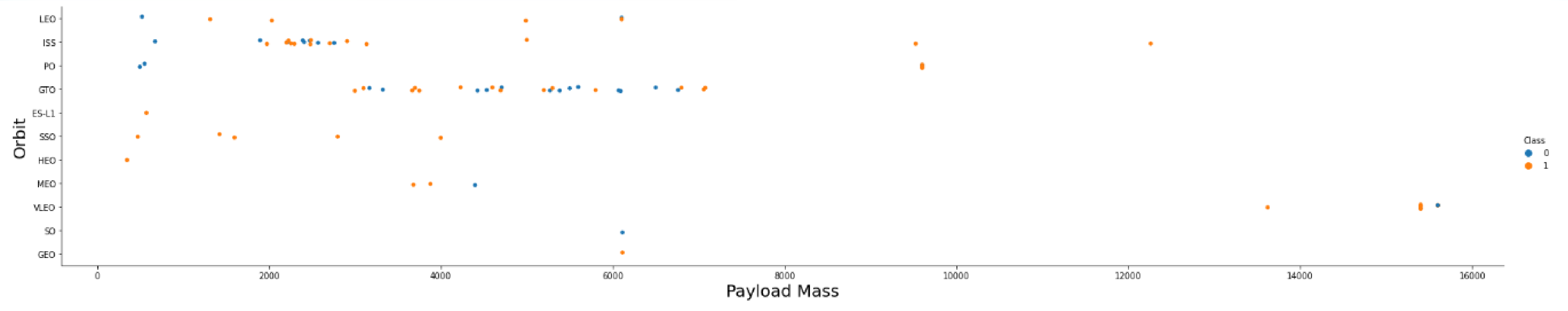
Flight Number vs. Orbit Type



Observation

- LEO orbit the Success appears related to the number of flights
- There seems to be no relationship between flight number when in GTO orbit

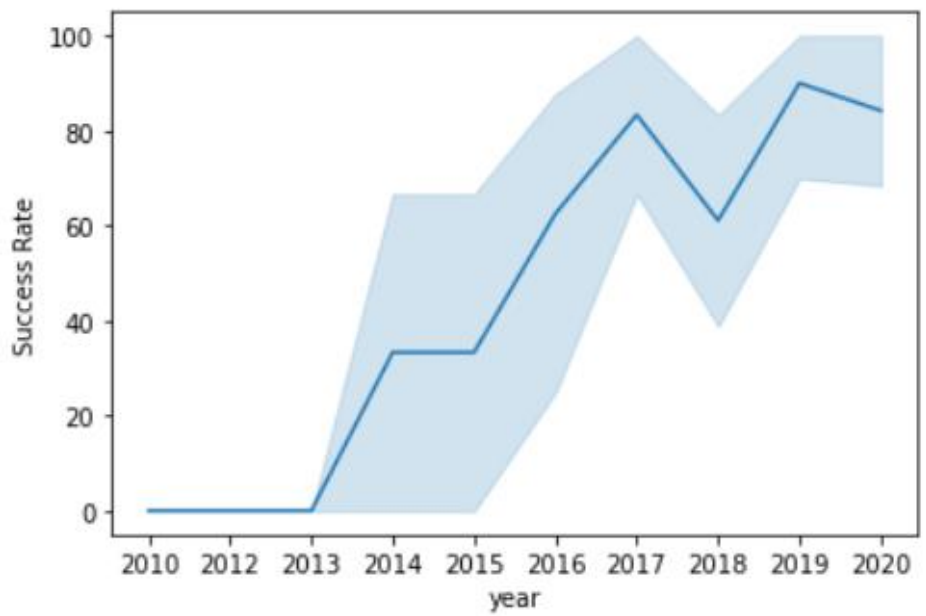
Payload vs. Orbit Type



Observation

- With heavy payloads, the successful landing or positive landing rate are more for Orbit Polar, LEO and ISS
- For Orbit GTO, we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here

Launch Success Yearly Trend



Observation

- The success rate since 2013 kept increasing till 2020

All Launch Site Names

```
[17]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL

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

[17]:

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

Explanation

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

Launch Site Names Begin with 'CCA'

```
[19]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

[19]:

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

Explanation

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

Total Payload Mass

```
[20]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

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

[20]:

SUM(PAYLOAD_MASS__KG_)
45596

Explanation

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

Average Payload Mass by F9 v1.1

```
[22]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1';

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

[22]:

AVG(PAYLOAD_MASS__KG_)
2928.4

Explanation

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

First Successful Ground Landing Date

```
[30]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE [Landing _Outcome] = 'Success (ground pad)';
```

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

[30]:

MIN(DATE)
01-05-2017

Explanation

- List the earliest date (by using Min) when the first successful landing outcome in ground pad was achieved

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[32]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE [Landing _Outcome] = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

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

[32]:

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

Explanation

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
[33]: %sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

[33]:

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Explanation

- List the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
[34]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

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

[34]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Explanation

- List the names of the booster versions which have carried the maximum payload mass

2015 Launch Records

```
[37]: %sql SELECT DISTINCT substr(Date, 4, 2), [Landing _Outcome], BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
      WHERE [Landing _Outcome] = 'Failure (drone ship)'
      AND substr(Date,7,4)='2015';

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

[37]:

substr(Date, 4, 2)	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Explanation

- List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015

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

```
[44]: %sql SELECT [Landing _Outcome],COUNT(*) FROM SPACEXTBL
      WHERE DATE BETWEEN '04-06-2010' and '20-03-2017'
      AND [Landing _Outcome] LIKE '%success%'
      GROUP BY [Landing _Outcome];
```

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

[44]:

Landing _Outcome	COUNT(*)
Success	20
Success (drone ship)	8
Success (ground pad)	6

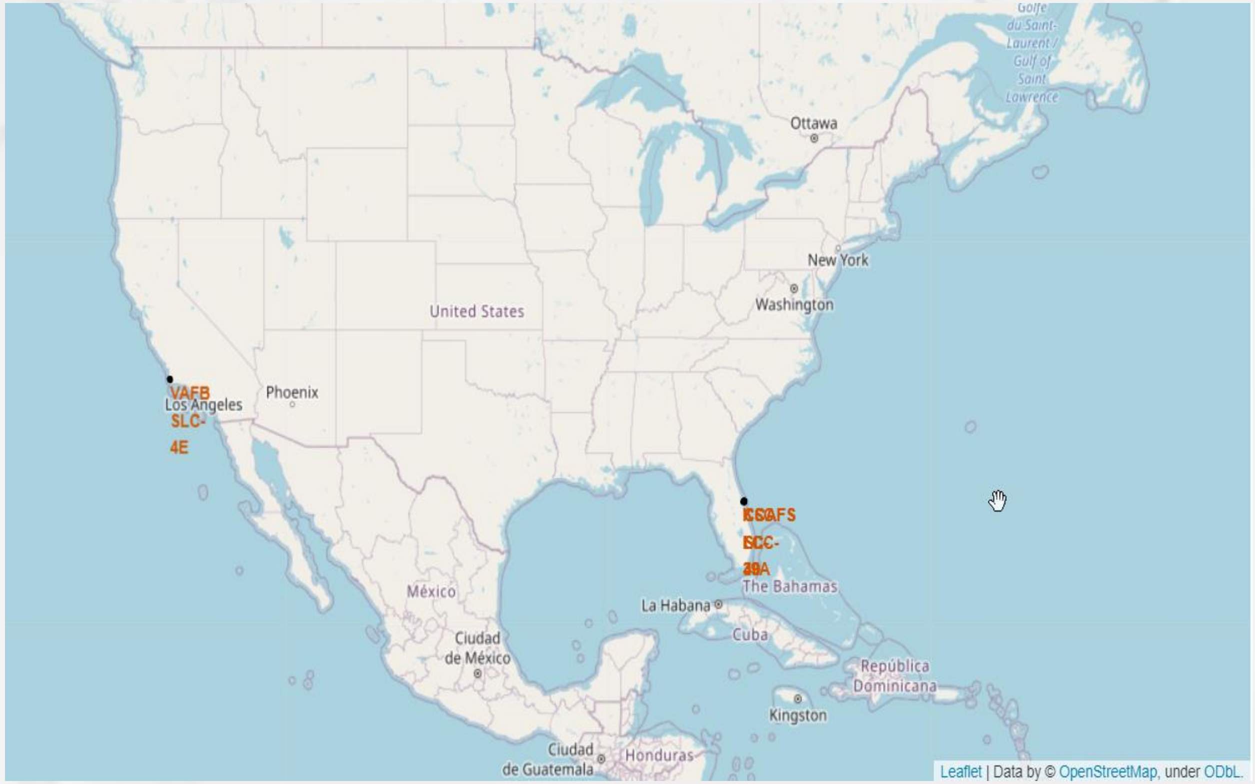
Explanation

- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Section 3

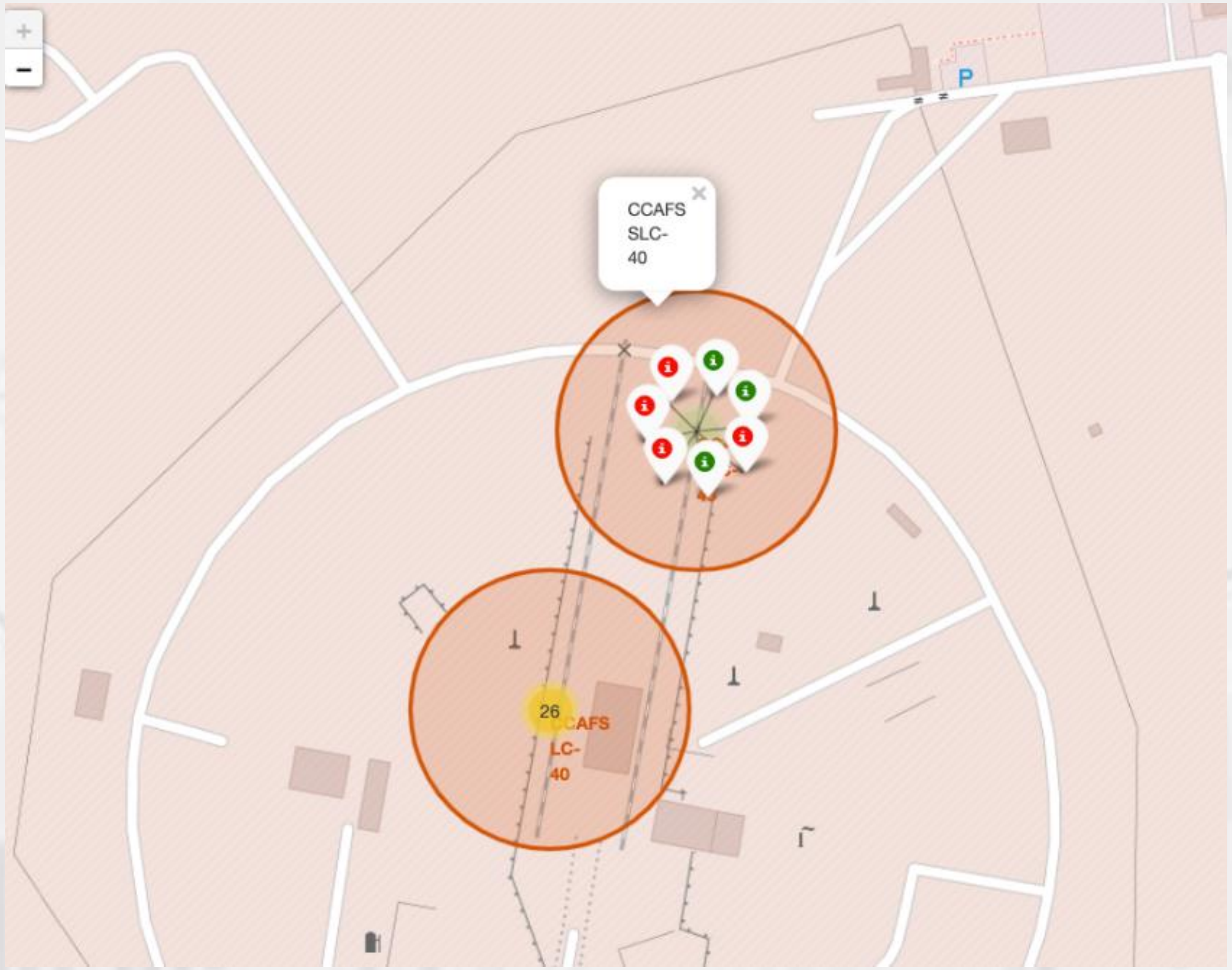
Launch Sites Proximities Analysis

All launch sites on a map



- ❖ Initial center location to be NASA Johnson Space Center at Houston, Texas
- ❖ The map show all launch sites in USmap

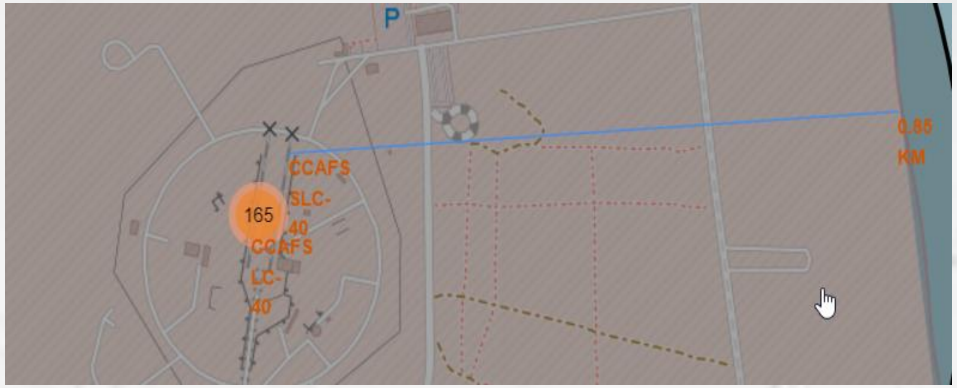
Success/Failed launches for each site on the map



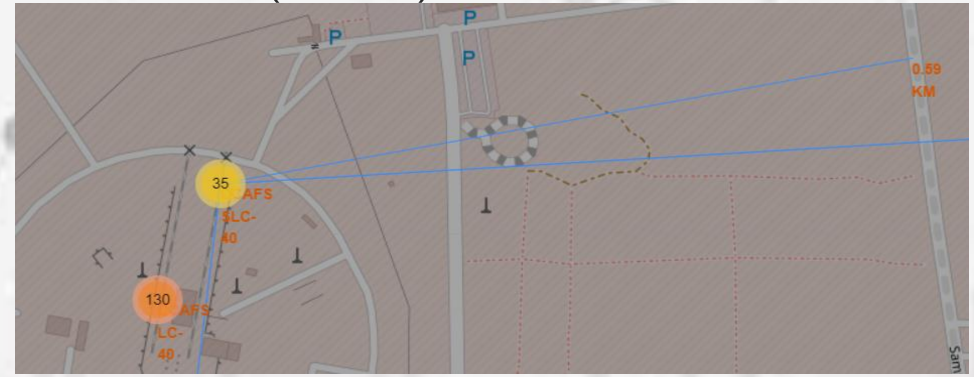
❖Create markers for all launch records.

❖If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)

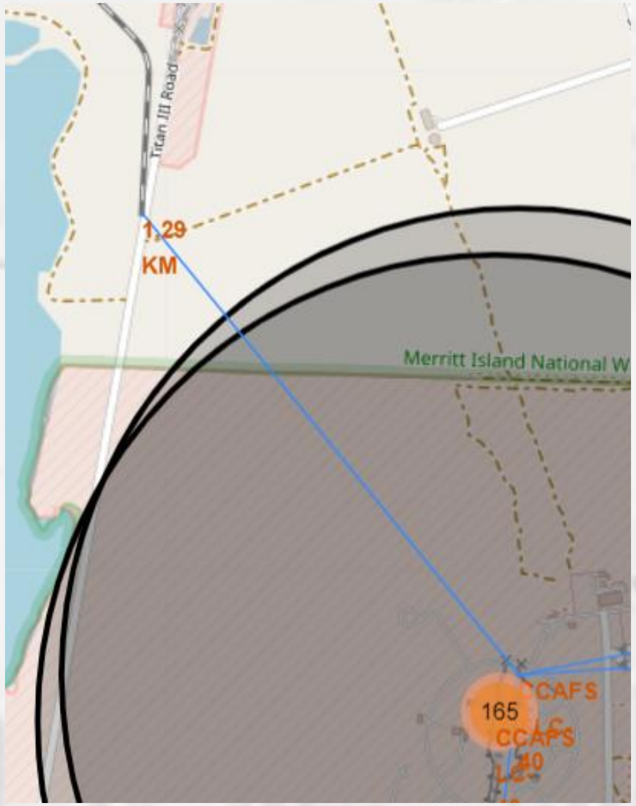
Distances between a launch site to its proximities



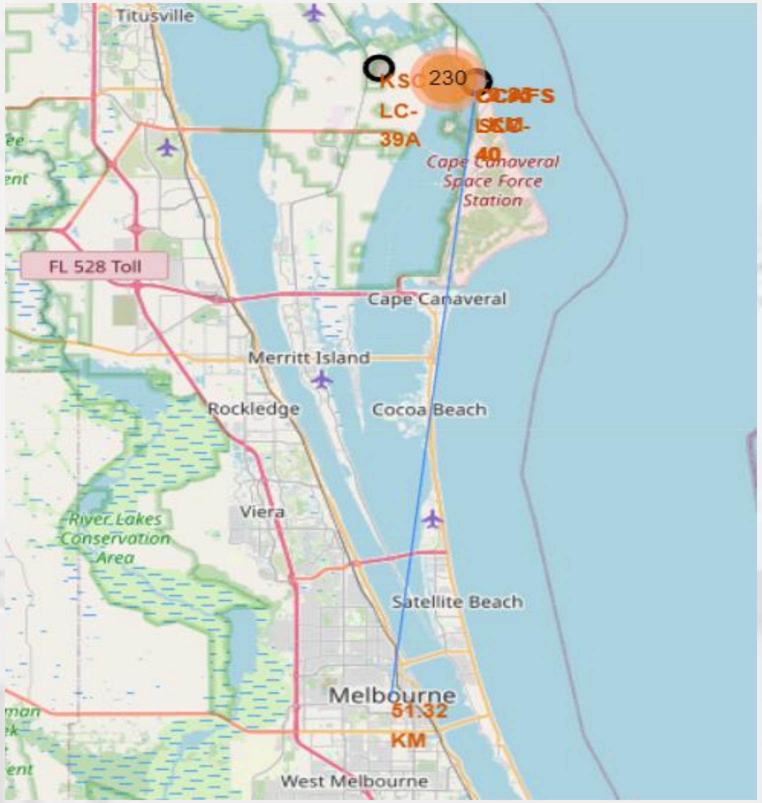
❖ Coastline (0.85 KM)



❖ Highway Samuel C Philips Highway (0.59 KM)



❖ Nasa Railroad (1.29 KM)



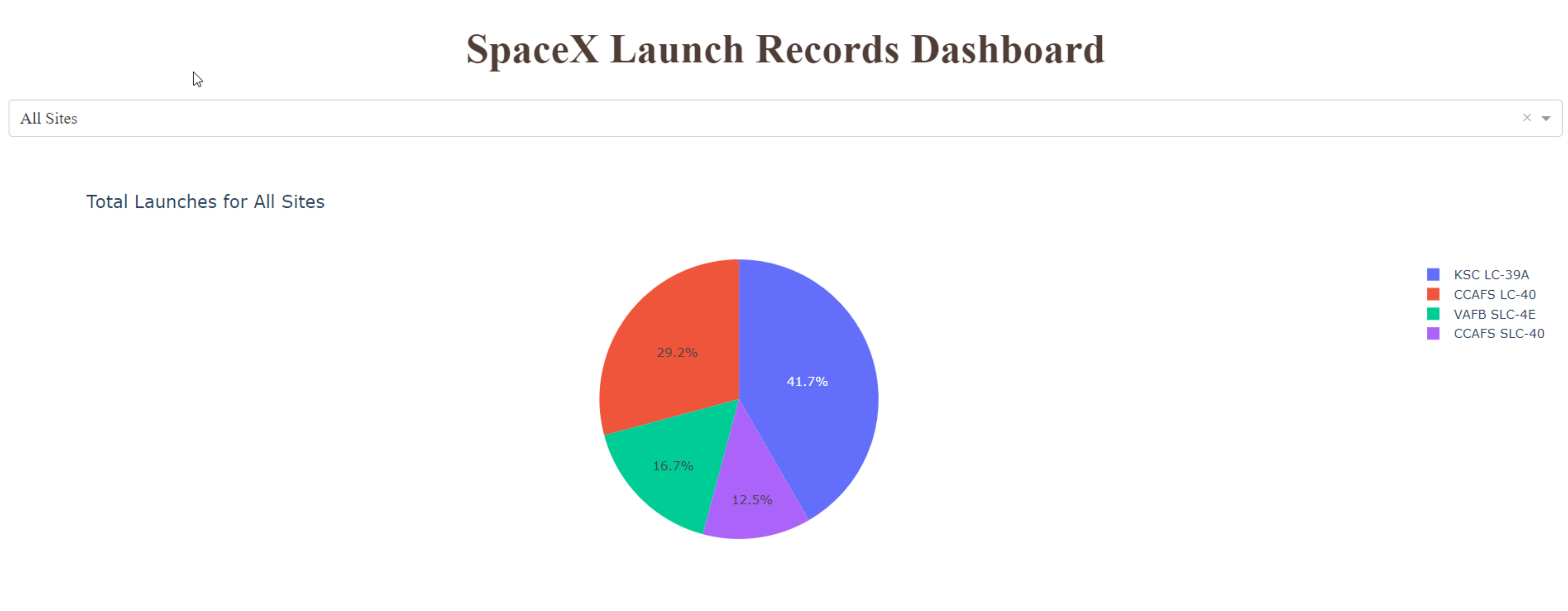
❖ City point Melbourne (51.32 KM)

❖ It show the distance and draw a Poly Line between a launch site e.g. CCAFS SLC-40 to the selected coastline/highway/railway/city point

Section 4

Build a Dashboard with Plotly Dash

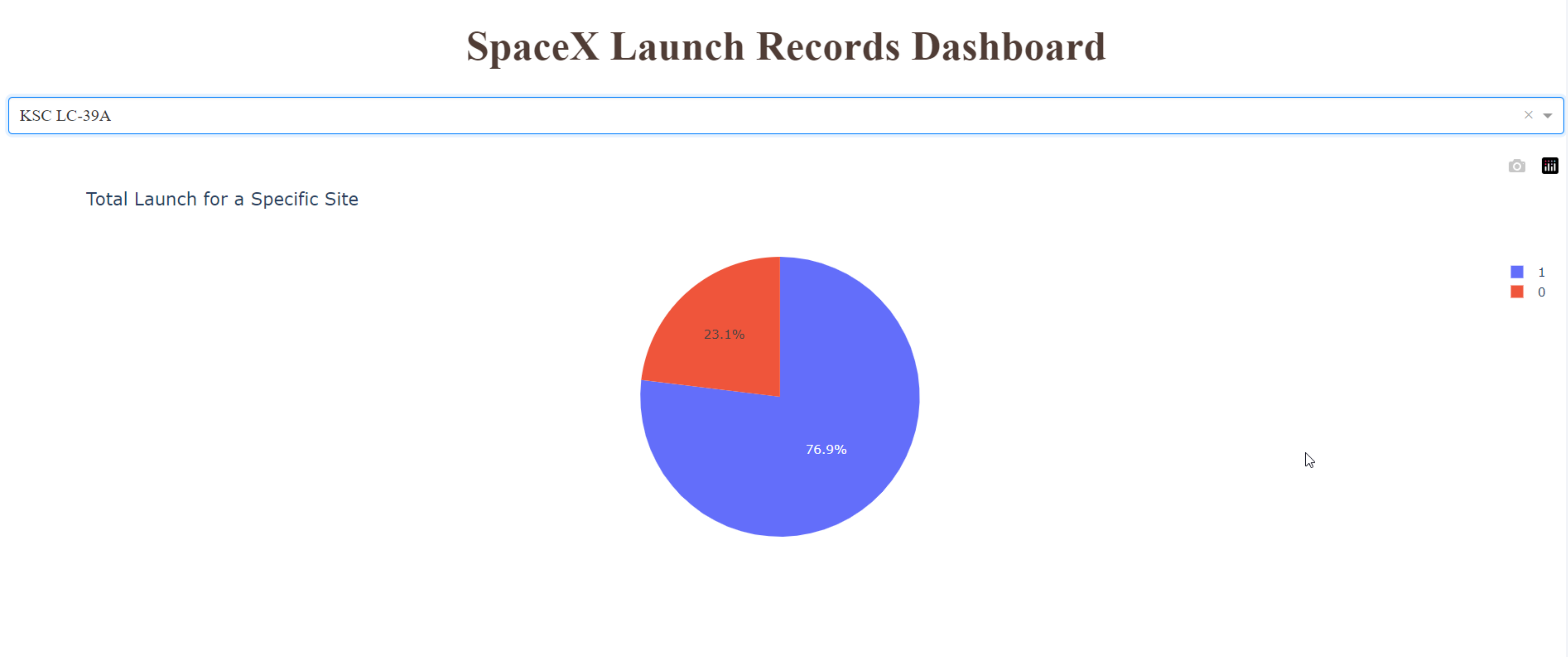
Pie chart showing launch success count for all sites



Observation

- KSC LC-39A has the highest launch success count

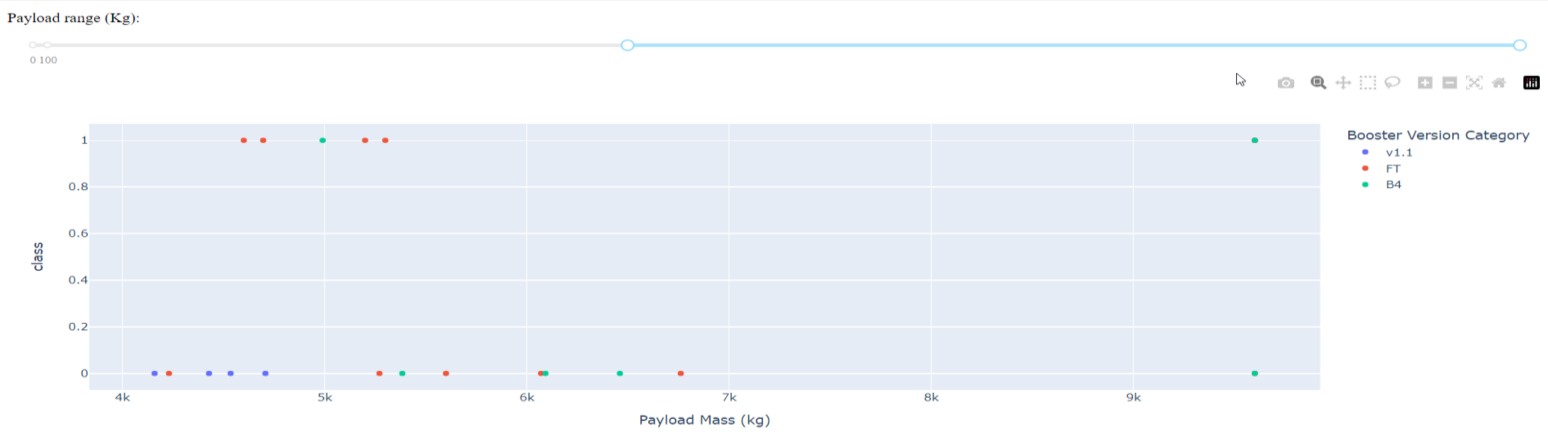
Pie Chart for launch site with highest launch



Observation

- KSC LC-39A 76.9% success rate and 23.1% failure rate

Scatter Plot showing Payload vs. Launch Outcome for all sites



Payload Mass 4000 KG range

Payload Mass 9600 KG range

Observation

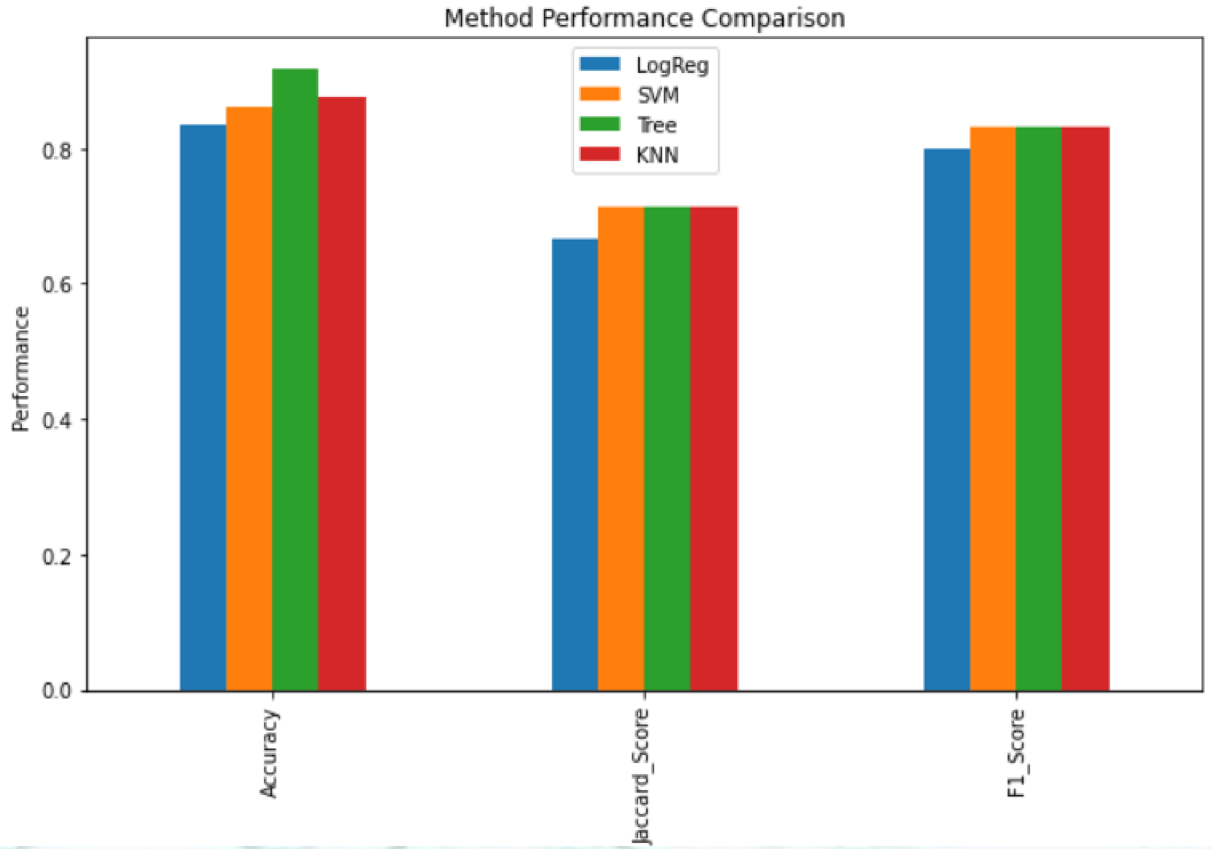
- The heavier the payload mass, the lower success launch outcome

Section 5

Predictive Analysis (Classification)

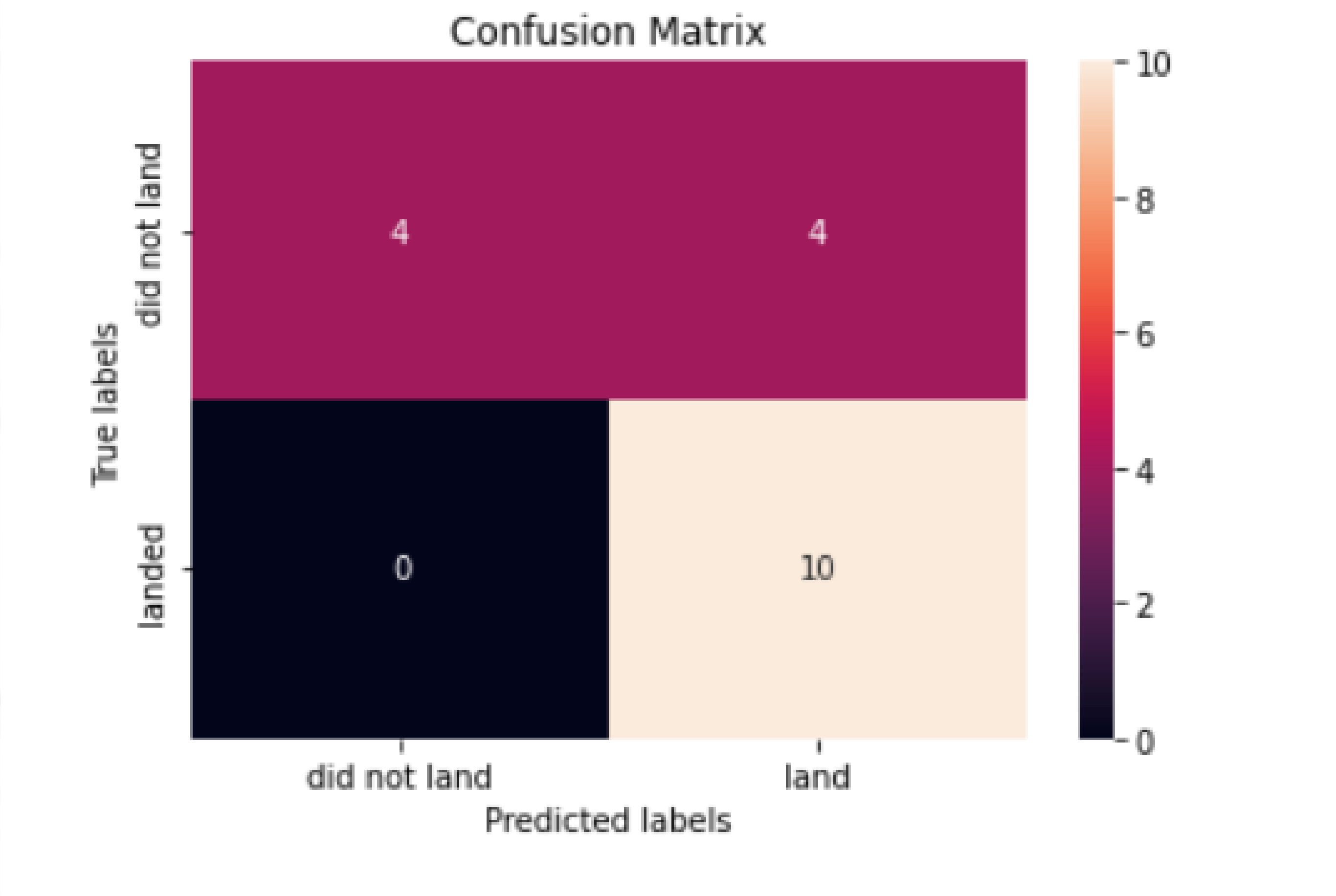
Classification Accuracy

	LogReg	SVM	Tree	KNN
Accuracy	0.835714	0.862500	0.917857	0.876786
Jaccard_Score	0.666667	0.714286	0.714286	0.714286
F1_Score	0.800000	0.833333	0.833333	0.833333

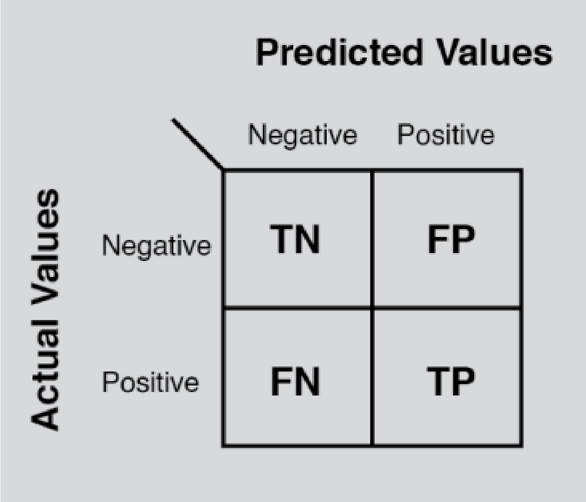


- Refer to Bar chart, it shows the built model accuracy for all built classification models
- Decision Tree model has highest accuracy when compare to other models.

Confusion Matrix



- Examining the confusion matrix, we see that Tree can distinguish between the different classes.
- However, the major problem is false positives e.g. unsuccessful landing marked as successful landing by the classifier



Conclusions

- Decision Tree model has highest accuracy when compare to other models
- KSC LC-39A has the highest launch success count
- The heavier the payload mass, the lower success launch outcome
- Most of the launch sites are near to coastline
- The success rate since 2013 kept increasing till 2020
- Orbit ES-L1, GEO, HEO and SSO have 100% success rate, Orbit SO has zero success rate while others Orbit except the mentioned has 50% - 70% success rate
- The larger the flight amount at a launch site, the greater the success rate at a launch site
- As the flight number increases, the first stage is more likely to land successfully.
- The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Appendix

- Git Hub repository URL https://github.com/ncymic/Applied-Data-Science-Capstone

Thank you!