IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

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 PlotlyDash
- Predictive analysis (Classification)

Summary of all results

- EDA results
- Interactive analytics
- Predictive analysis

Introduction

- Project background and context

  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.

- Problems you want to find answers

  The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully

Section 1

Methodology

Methodology

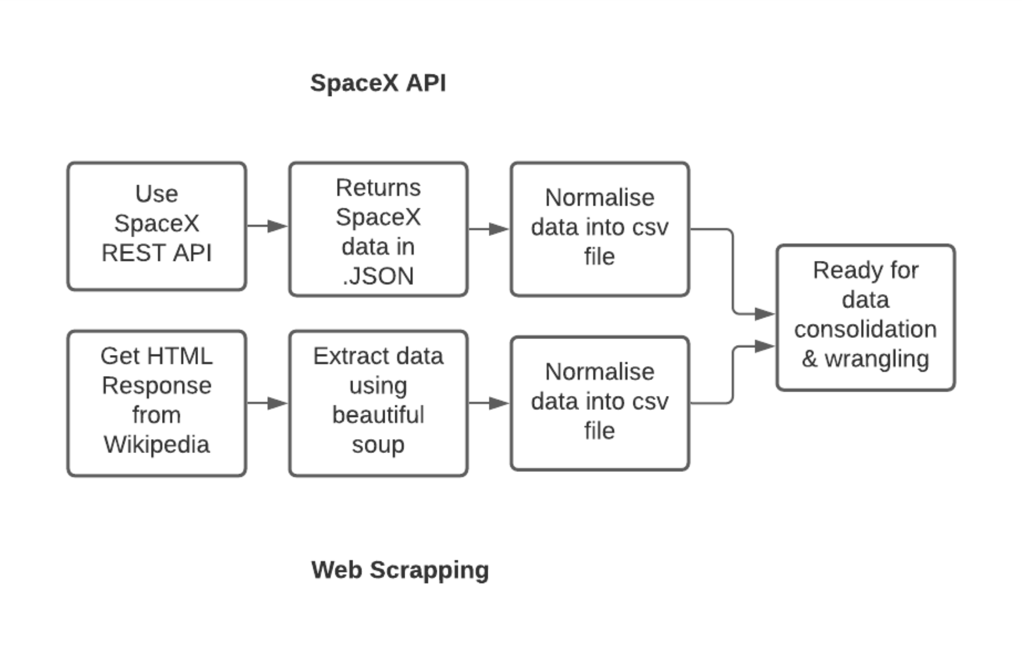
Executive Summary

- Data collection methodology:
  - Space X rest API
  - Webscraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- The following datasets was collected:

  - SpaceX launch data that is gathered from the SpaceX REST API.
  - This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
  - 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

- Data collection with SpaceX REST calls

- https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/01%20Data%20Collection.ipynb

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

- Web Scrapping from Wikipedia

- https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/03%20Webscraping.ipynb

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
```

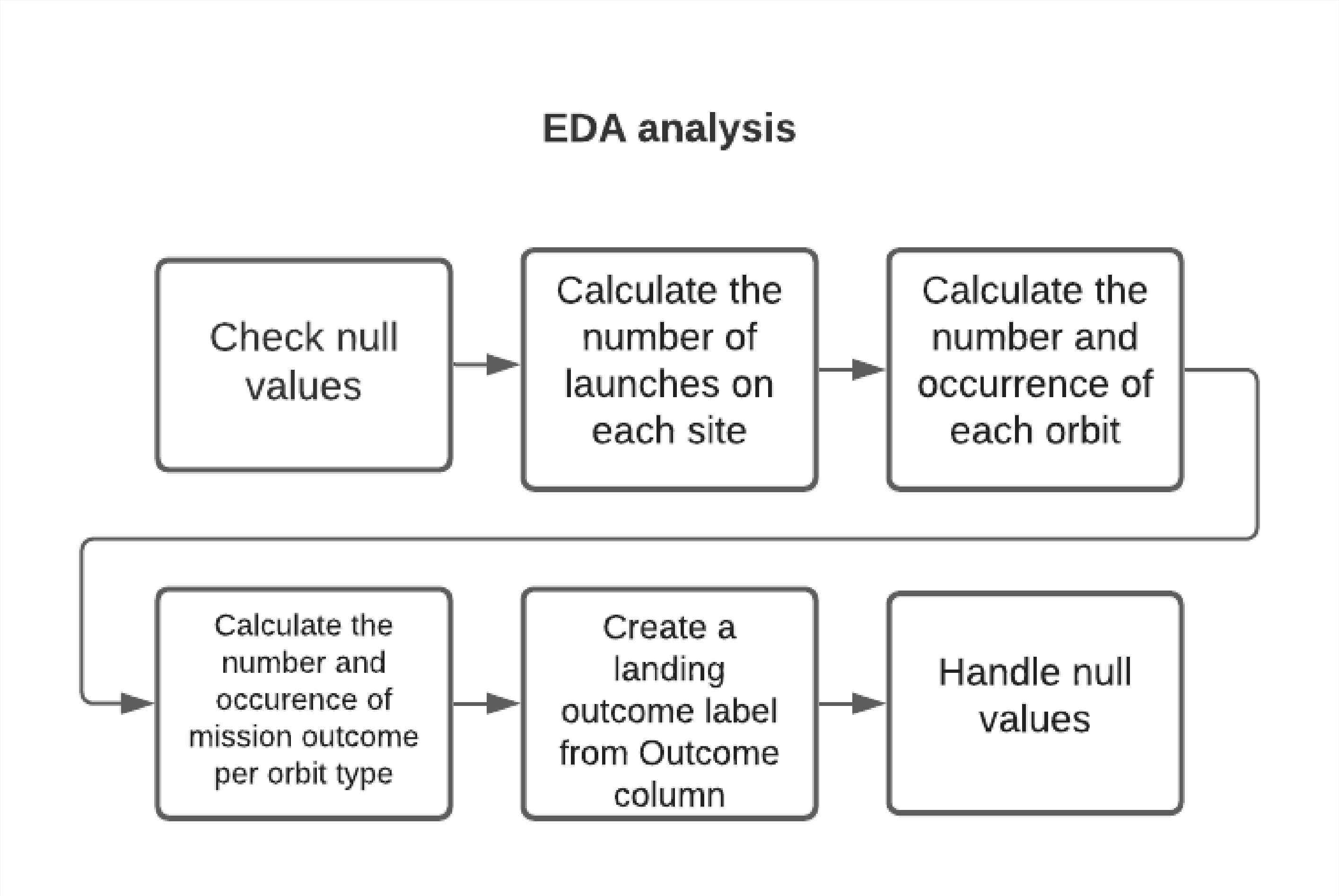
7. Converting dictionary to dataframe

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

8. Dataframe to .CSV

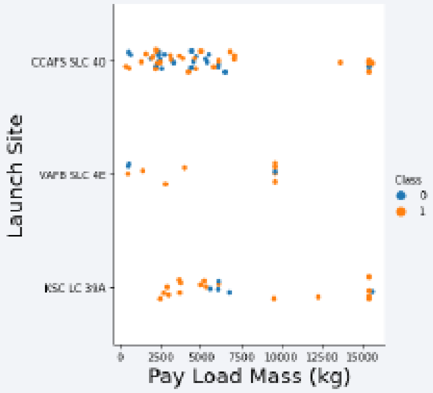
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

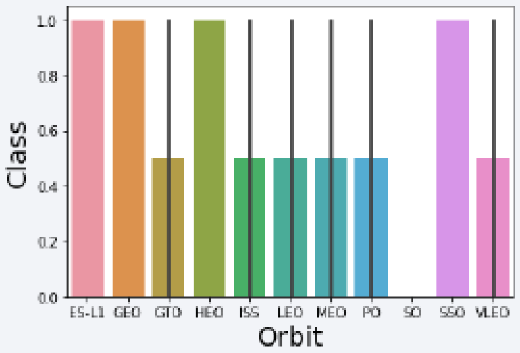
Data Wrangling

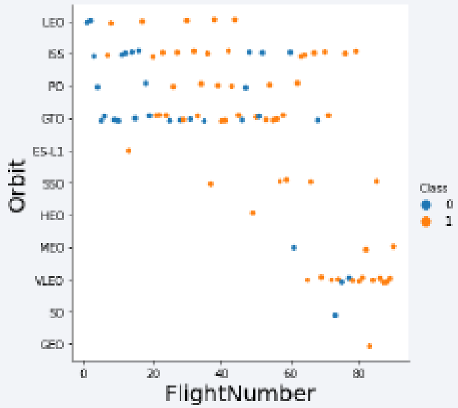


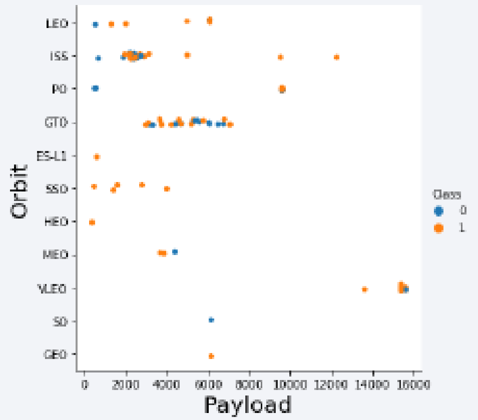
https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/02%20Data%20Wrangling.ipynb

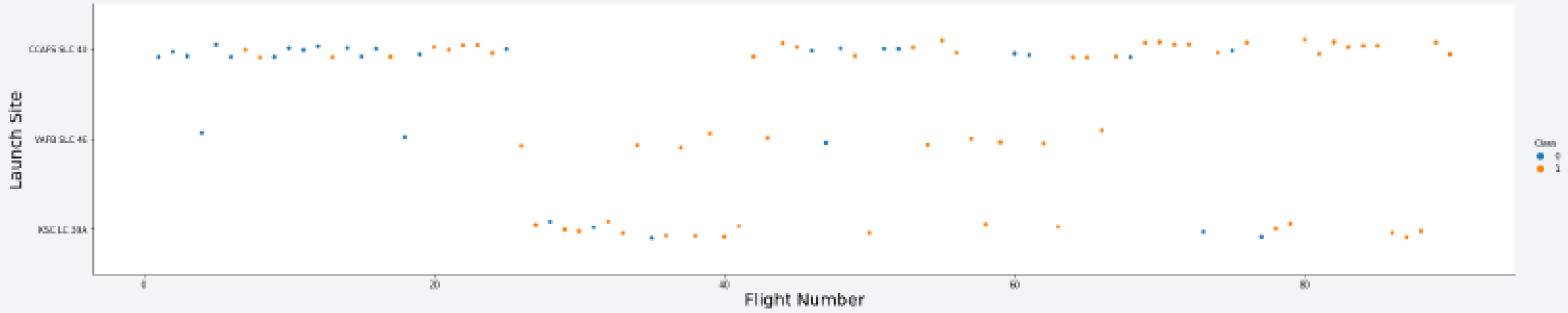
EDA with Data Visualization

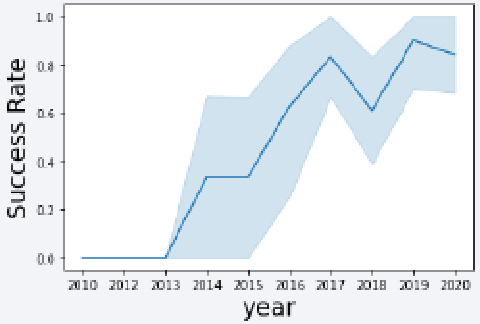












https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/05%20EDA%20Visualisation.ipynb

EDA with SQL

- **SQL queries performed**
  - 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

https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/04%20EDA%20with%20Sql.ipynb

Build an Interactive Map with Folium

Objects created and added to a folium map:

- Markers that show all launch sites on a map
- Markers that show the success/failed launches for each site on the map
- Lines that show the distances between a launch site to its proximities

By adding these objects, following geographical patterns about launch sites are found:

- Are launch sites in close proximity to railways? *Yes*
- Are launch sites in close proximity to coastline? *Yes*
- Do launch sites keep certain distance away from cities? *Yes*

https://github.com/manisangsu/mammoth_falcon/blob/4a52acb9ebaf0e4d9b9a812d556575a25c872bec/06%20Folium.ipynb

Build a Dashboard with Plotly Dash

The dashboard application contains a pie chart and a scatter point chart.

Pie chart

- For showing total success launches by sites
- This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.

Scatter chart

- For showing the relationship between `Outcomes` and `Payload mass (Kg)` by different boosters
- Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg
- This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

https://github.com/manisangsu/mammoth_falcon/blob/081a0f1281004963257b1696c92f9fddec342415/08%20%20%20Ploty%20Dash.py

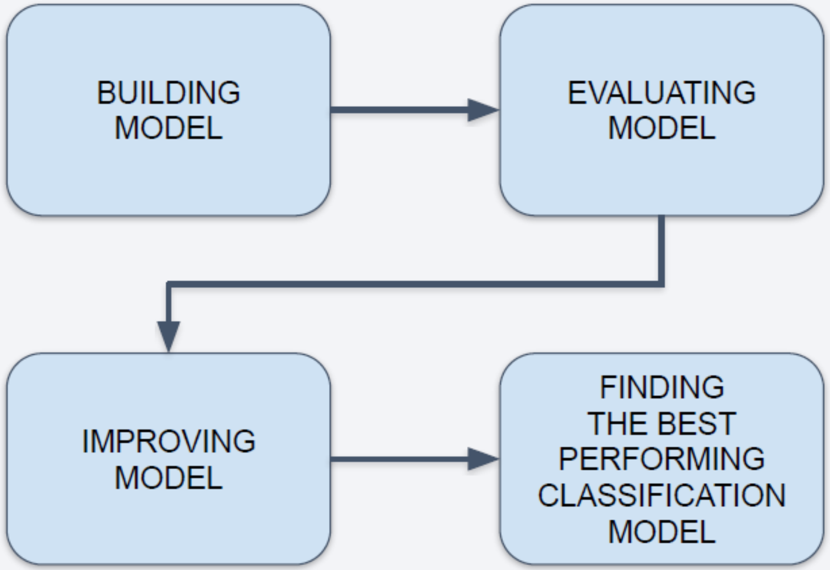
Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels

- Create a column for the class Standardize the data
- Split into training data and test data

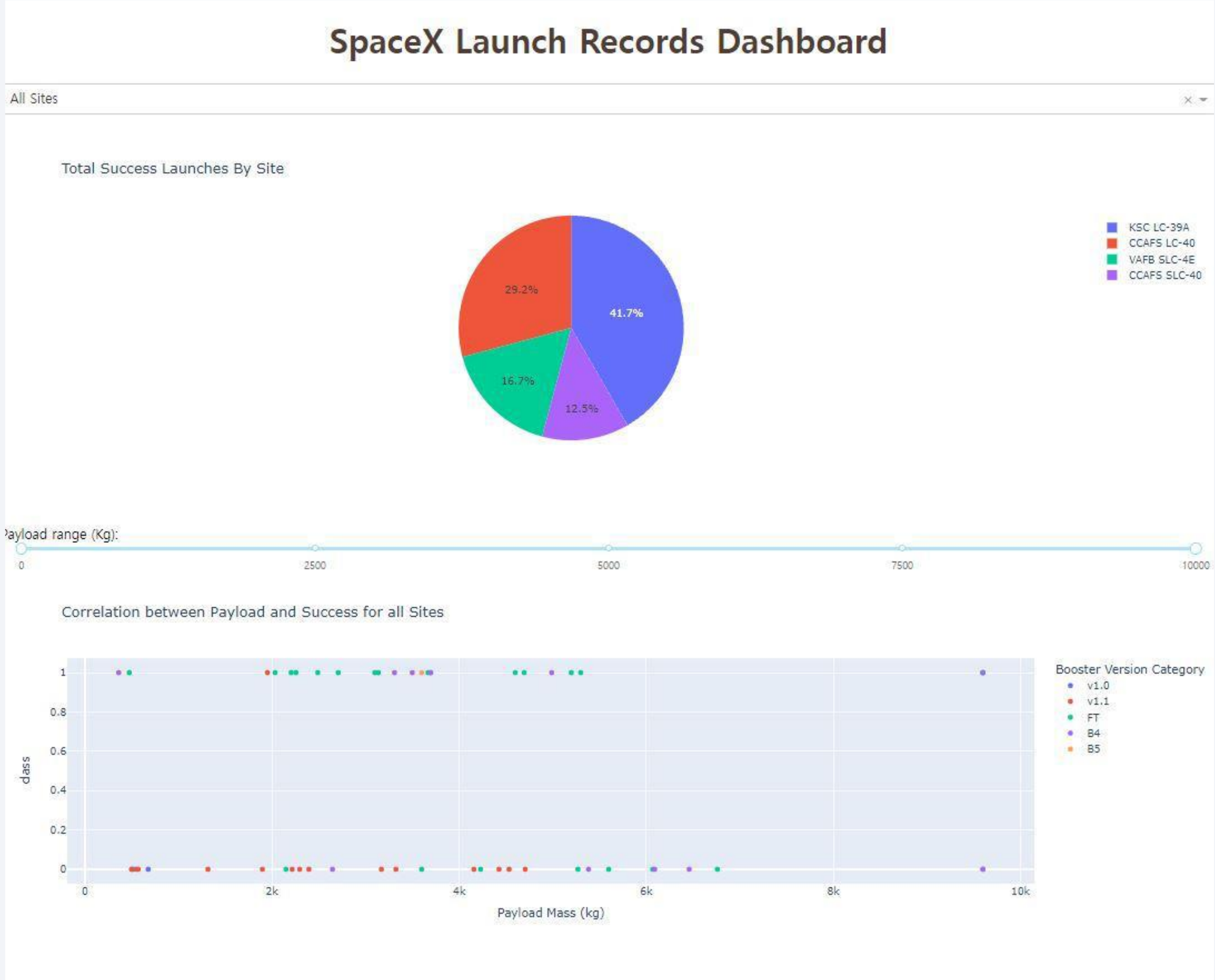
Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

- Find the method performs best using test data



https://github.com/manisangsu/mammoth_falcon/blob/a789450a02f525244b0156efe5afdf20fbad9bbe/7.1%20Machine%20Learning.ipynb

Results

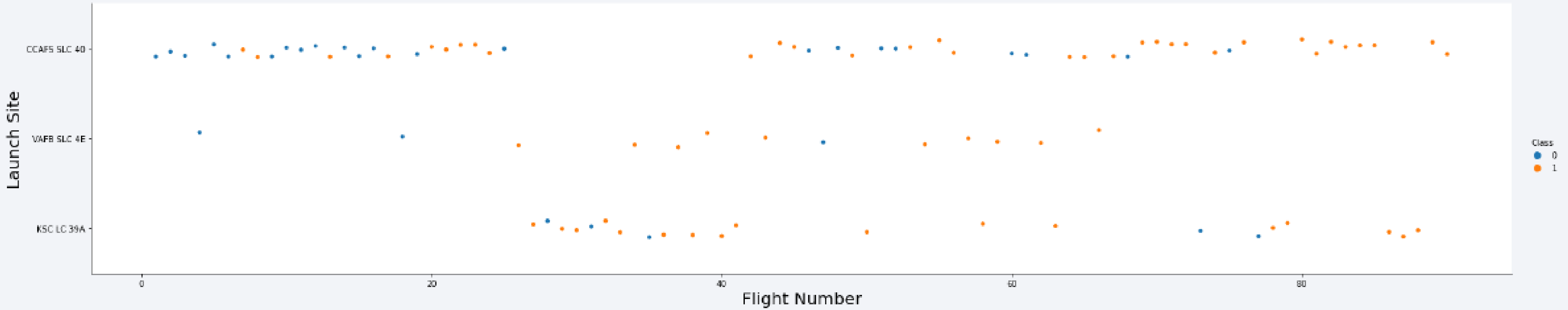


- The left screenshot is a preview of the Dashboard with Plotly Dash.
- The results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and Interactive Dashboard will be shown in the next slides.
- Comparing the accuracy of the four methods, all return the same accuracy of about *83%* for test data.

Section 2

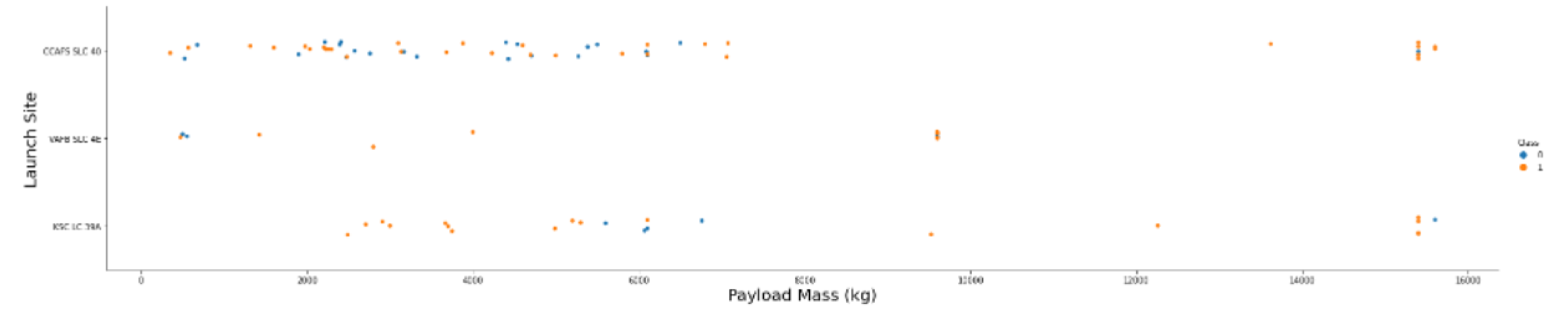
Insights drawn from EDA

Flight Number vs. Launch Site



Launches from the site of CCAFS SLC 40 are significantly higher than launches form other sites.

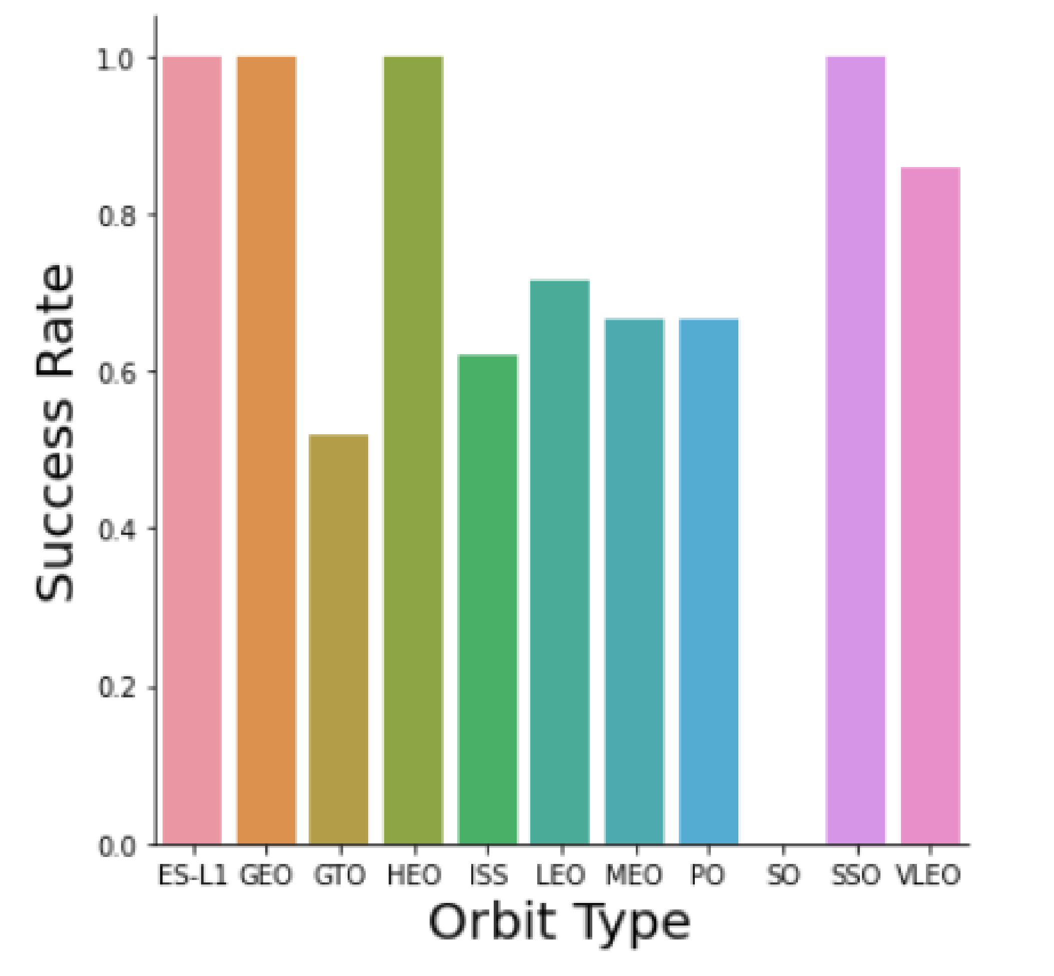
Payload vs. Launch Site



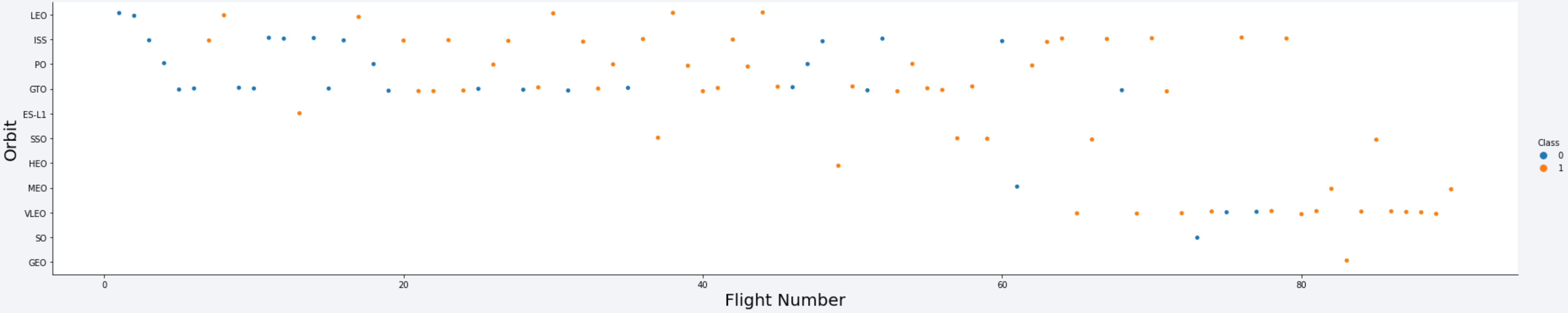
- The majority of IPayLoads with lower Mass have been launched from CCAFS SLC 40.

Success Rate vs. Orbit Type

- The orbit types of ES L1, GEO, HEO, SSO are among the highest success rate.



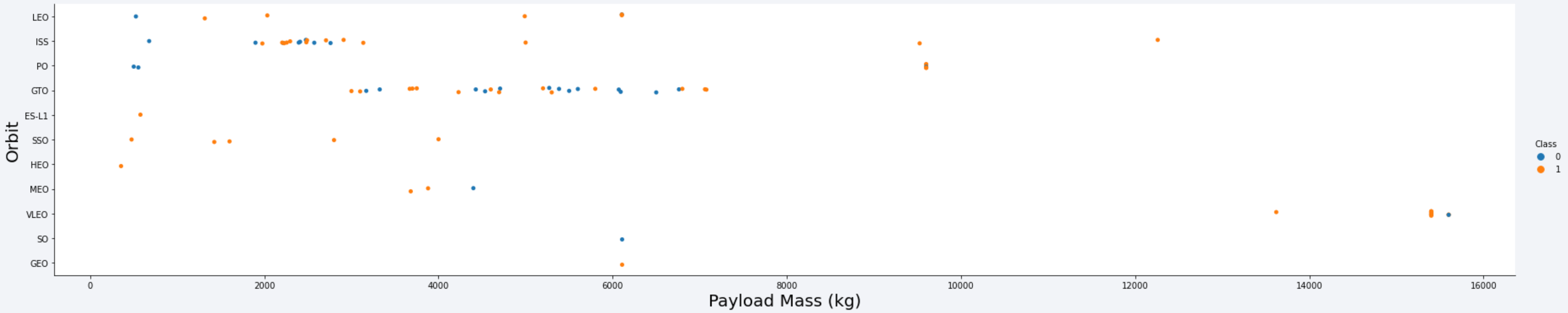
Flight Number vs. Orbit Type



- A trend can be observed of shifting to VLEO launches in recent years.

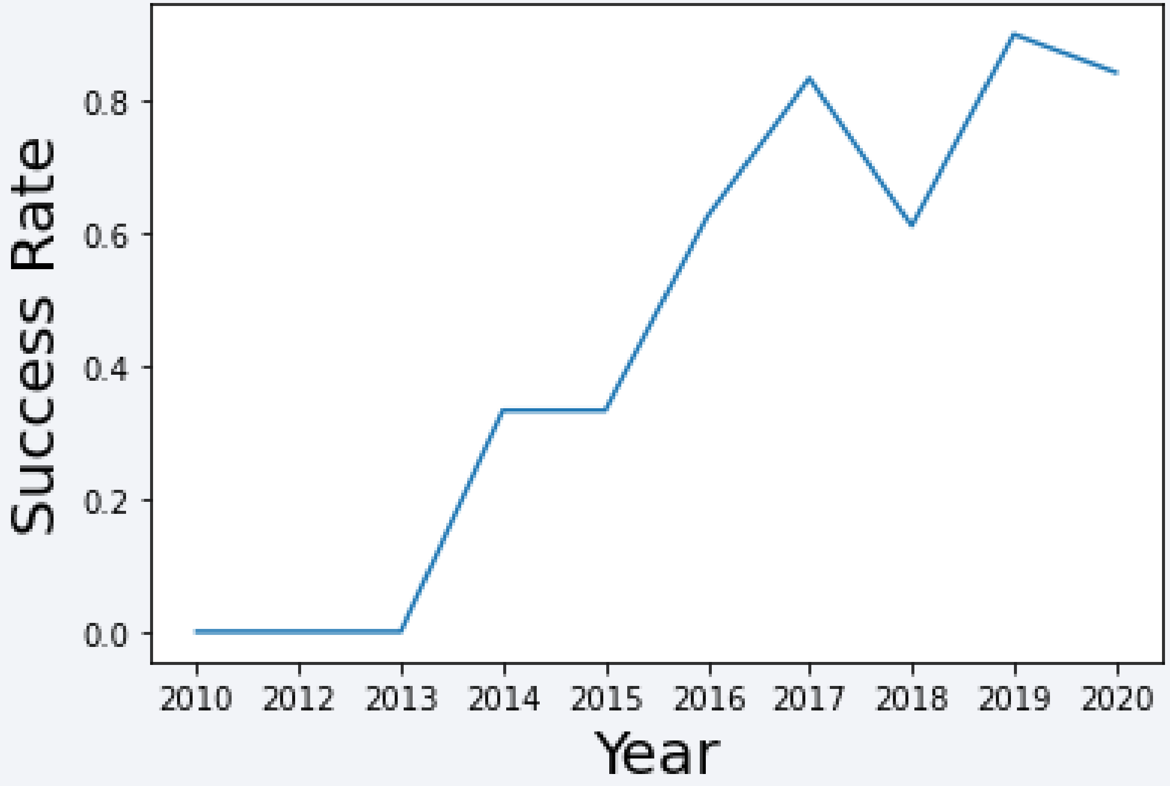
Payload vs. Orbit Type

There are strong correlation between ISS and Payload at the range around 2000, as well as between GTO and the range of 4000-8000.



Launch Success Yearly Trend

Launch success rate has increased significantly since 2013 and has stabilised since 2019, potentially due to advance in technology and lessons learned.



All Launch Site Names

- %sql select distinct(LAUNCH_SITE) from SPACEXTBL
  - The 'distinct' operation on LAUNCH SITE field, display unique values

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

Launch Site Names Begin with 'CCA'

- %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
  - the combination filters for launch site names beginning with CCA , limited to 5 records

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

Total Payload Mass

- %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)’;
  - Calculates the total payload carried by boosters from NASA – filters on NASA as customer and sums the filtered data set by payload

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%’;
    - Calculates the average payload mass carried by booster version F9 v1.1 – filters on the booster version field being = F9 v1.1 and calculates the average payload mass for them

average_payload_mass
2534

First Successful Ground Landing Date

- %sql select min(date) as first_successful_landing from SPACEXTBL where landing__outcome = 'Success (ground pad)’;
  - Finds the dates of the first successful landing outcome on ground pad – filters on a successful landing and then finds the min value among those dates

first_successful_landing
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;
  - Lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

Total Number of Successful and Failure Mission Outcomes

- %sql select mission_outcome, count(*) as total_number from SPACEXTBL group by mission_outcome;
  - The group by function groups data by the mission outcome and count is used to count them by those groups . An overwhelmingly large no of outcomes were success.

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- %sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL);

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

2015 Launch Records

The query filters for a combination of landing outcomes of failure for the year = 2015

```
%%sql select monthname(date) as month, date, booster_version, launch_site, landing__outcome from SPACEXTBL
      where landing__outcome = 'Failure (drone ship)' and year(date)=2015;
```

MONTH	DATE	booster_version	launch_site	landing__outcome
October	10-01-2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)

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

- A ranked count of outcomes between 2010 and 2017

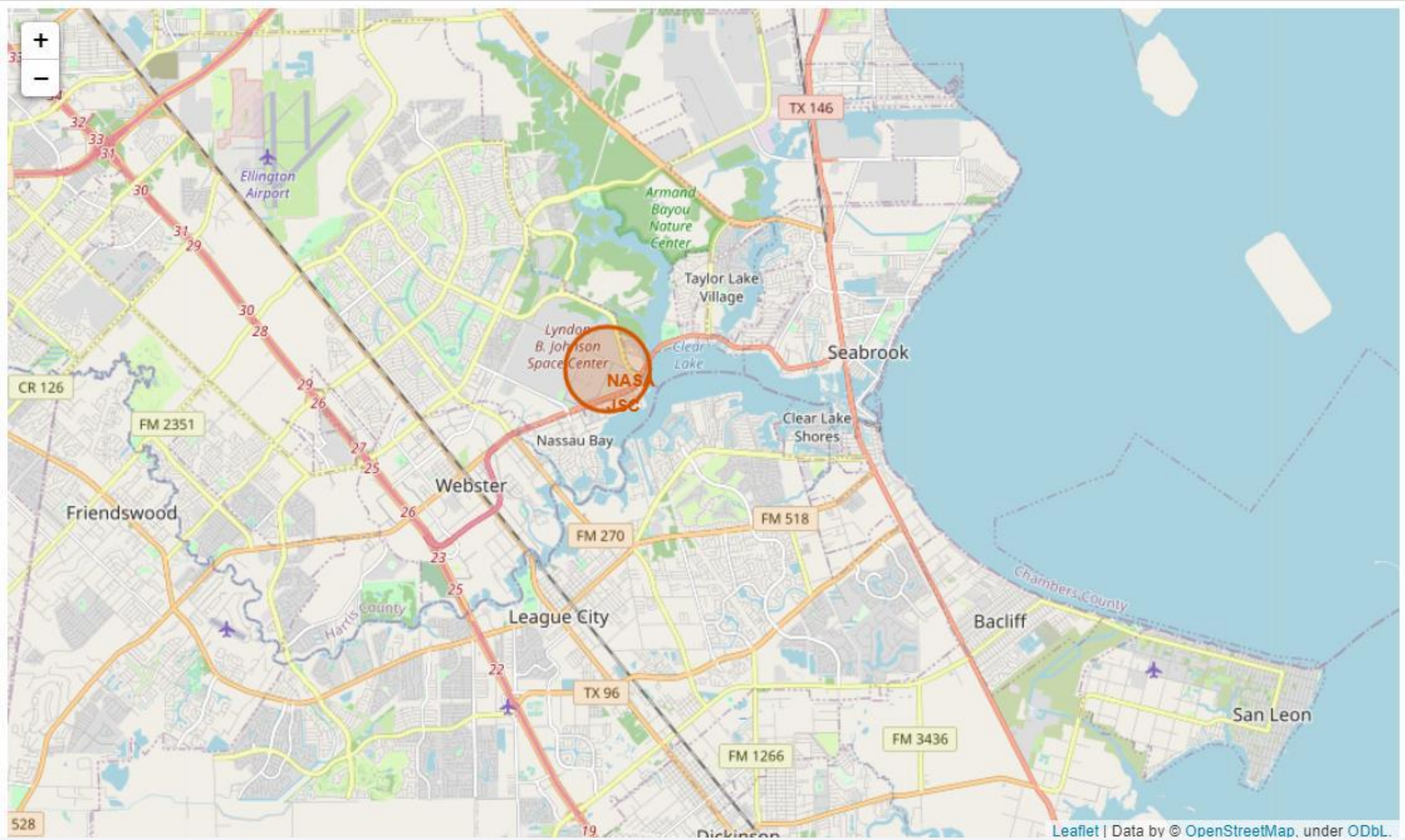
```
%%sql select landing__outcome, count(*) as count_outcomes from SPACEXTBL
      where date between '04-06-2010' and '20-03-2017'
      group by landing__outcome
      order by count_outcomes desc;
```

landing__outcome	count_outcomes
Success	20
No attempt	11
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Controlled (ocean)	3
Failure	3
Failure (parachute)	2

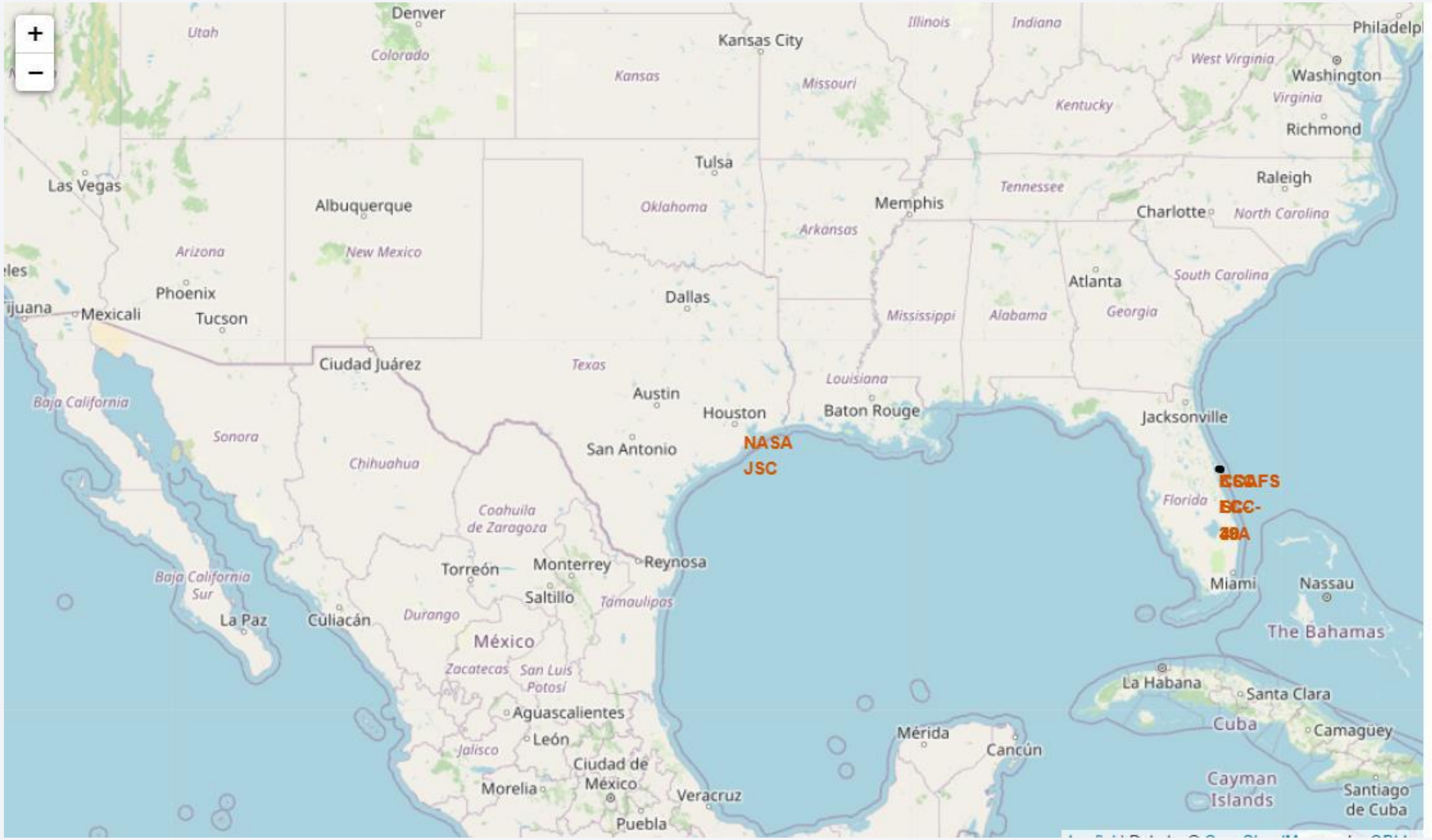
Section 3

Launch Sites Proximities Analysis

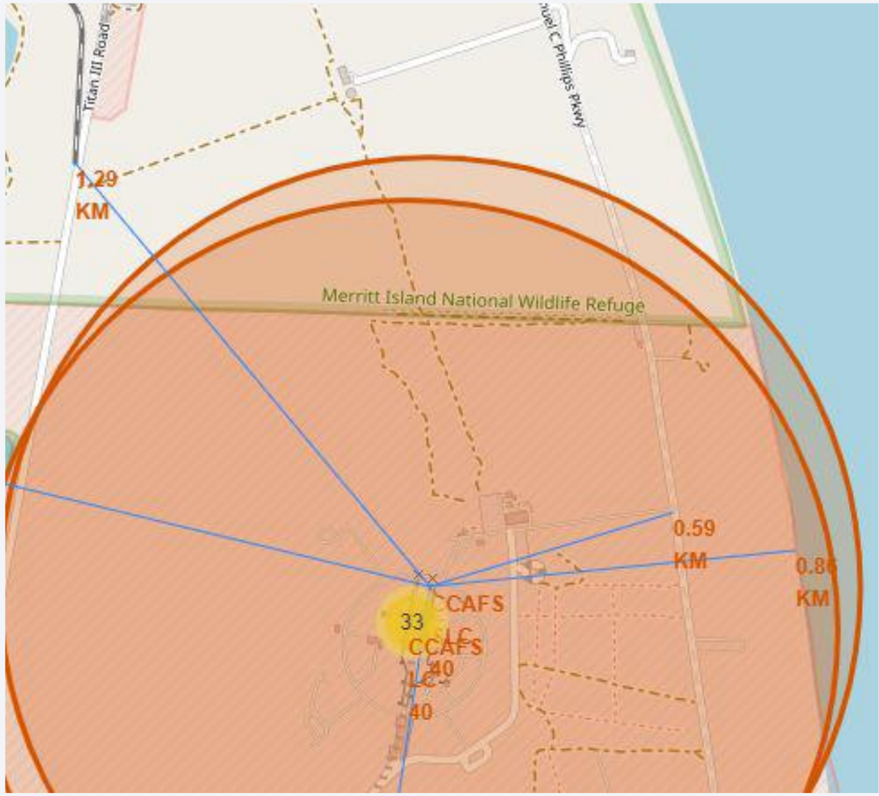
All launch sites marked on a map

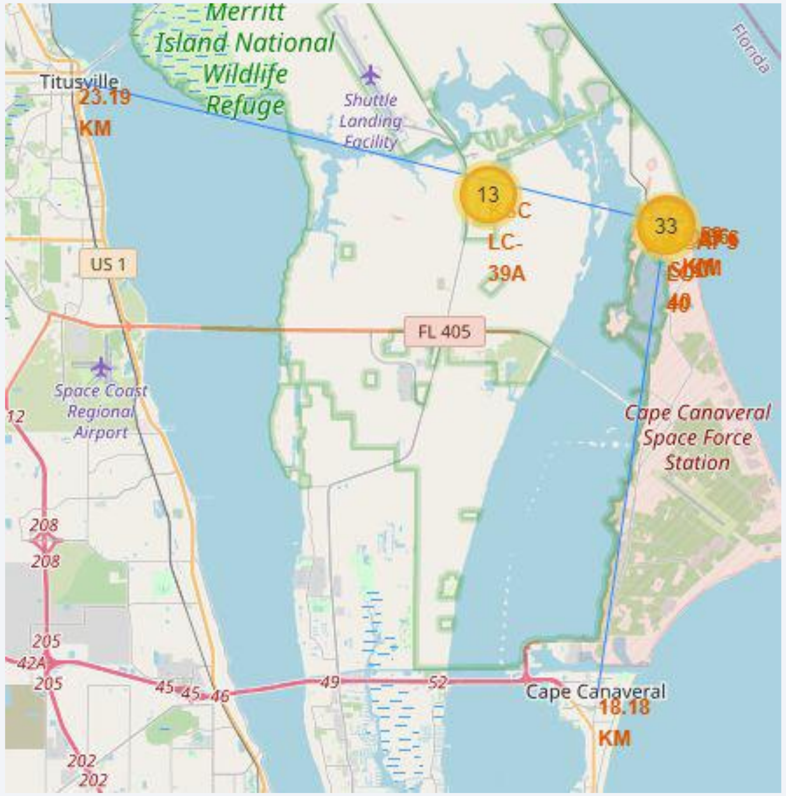


Success/failed launches marked on the map



Distances between a launch site to its proximities





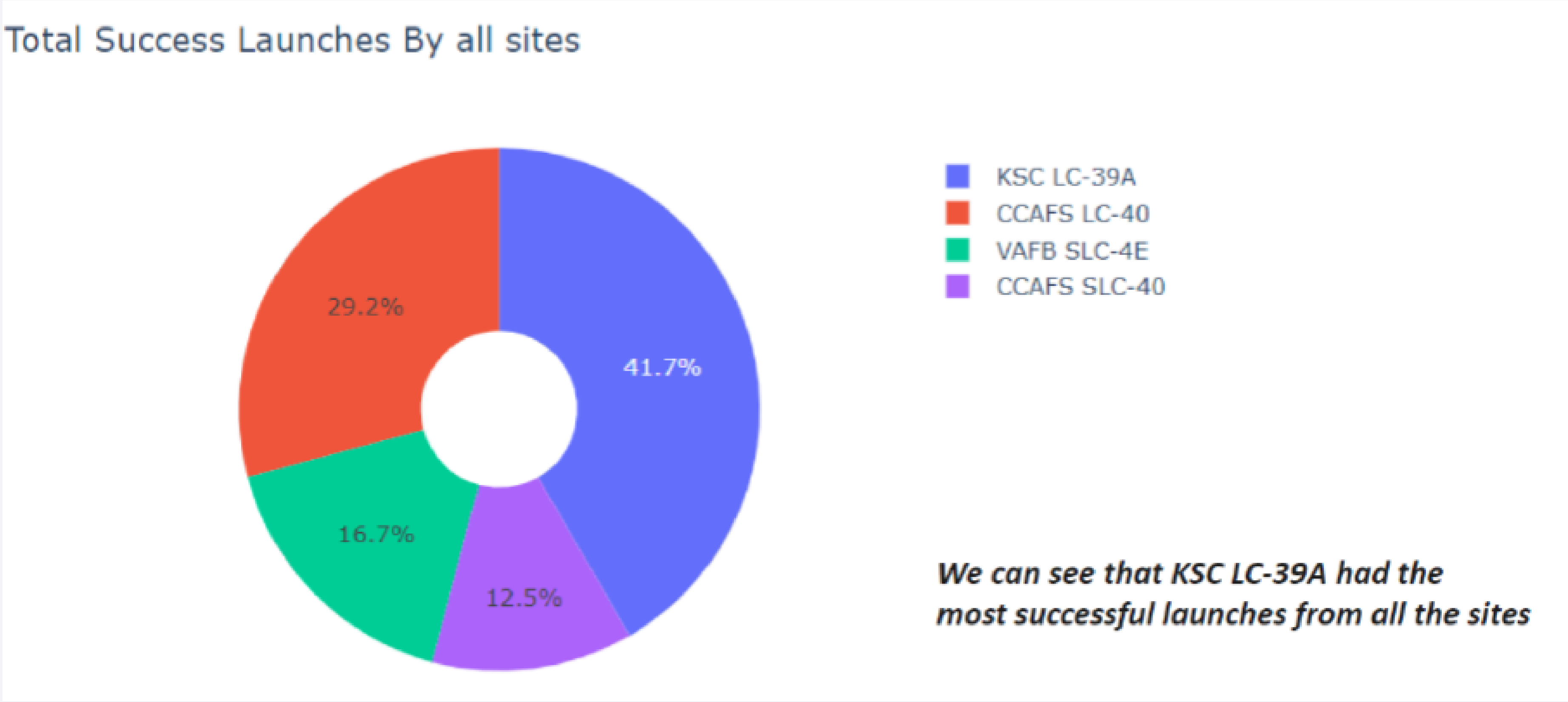
It can be found that the launch site is **close to railways and highways** for transportation of equipment or personnel, and is also **close to coastline**

Launch sites are relatively **far from the cities** so that launch failure does not pose a threat

Section 4

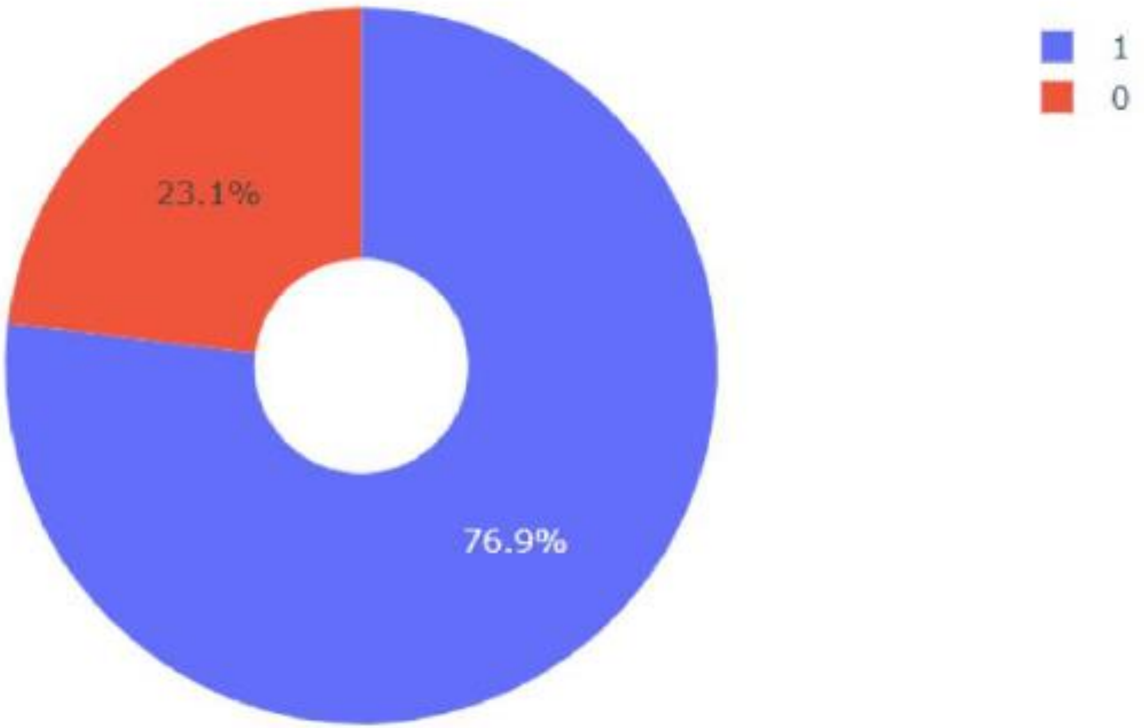
Build a Dashboard with Plotly Dash

Total success launches by all sites



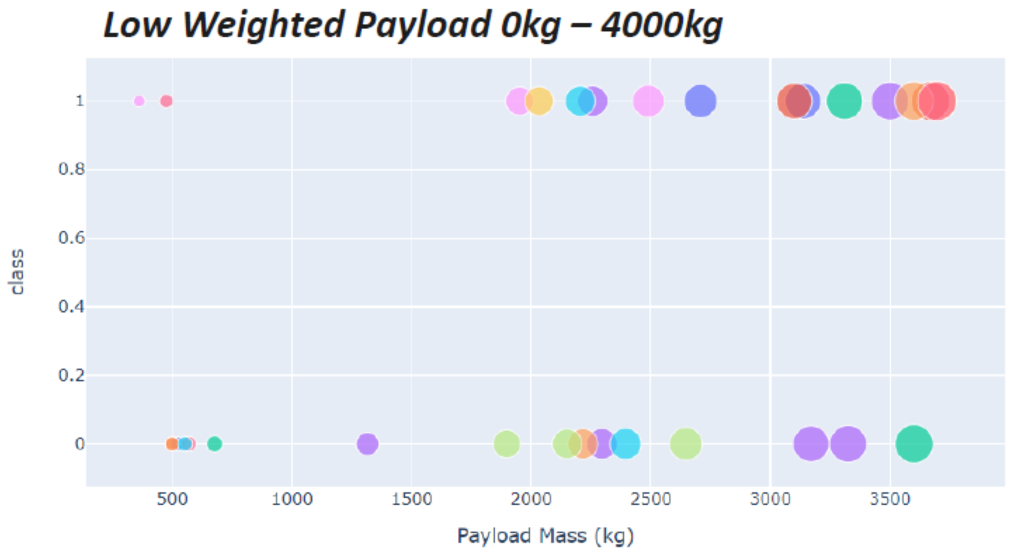
We can see that KSC LC-39A had the most successful launches from all the sites

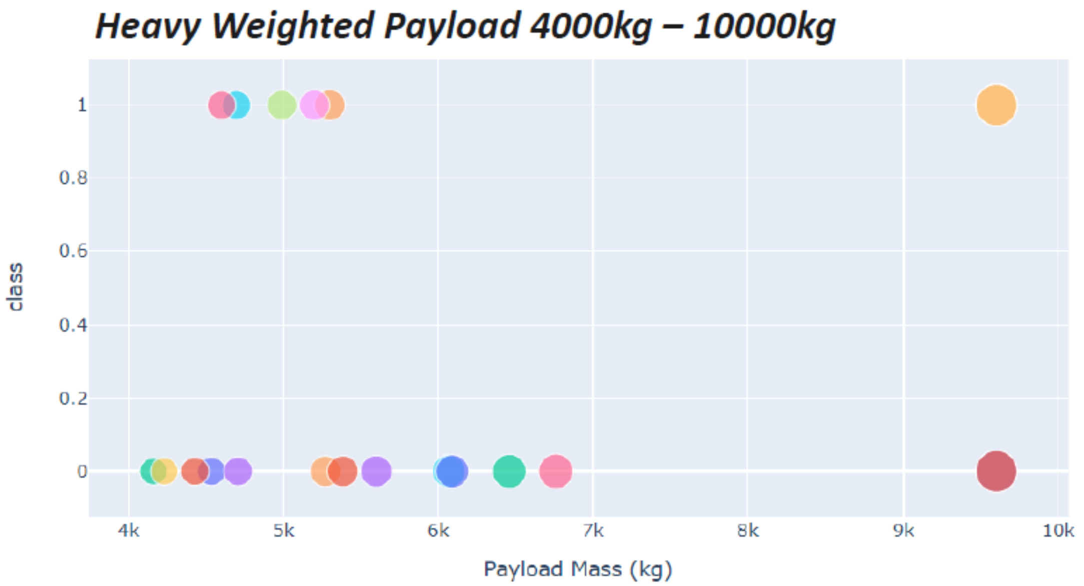
Success rate by site



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

Payload vs launch outcome



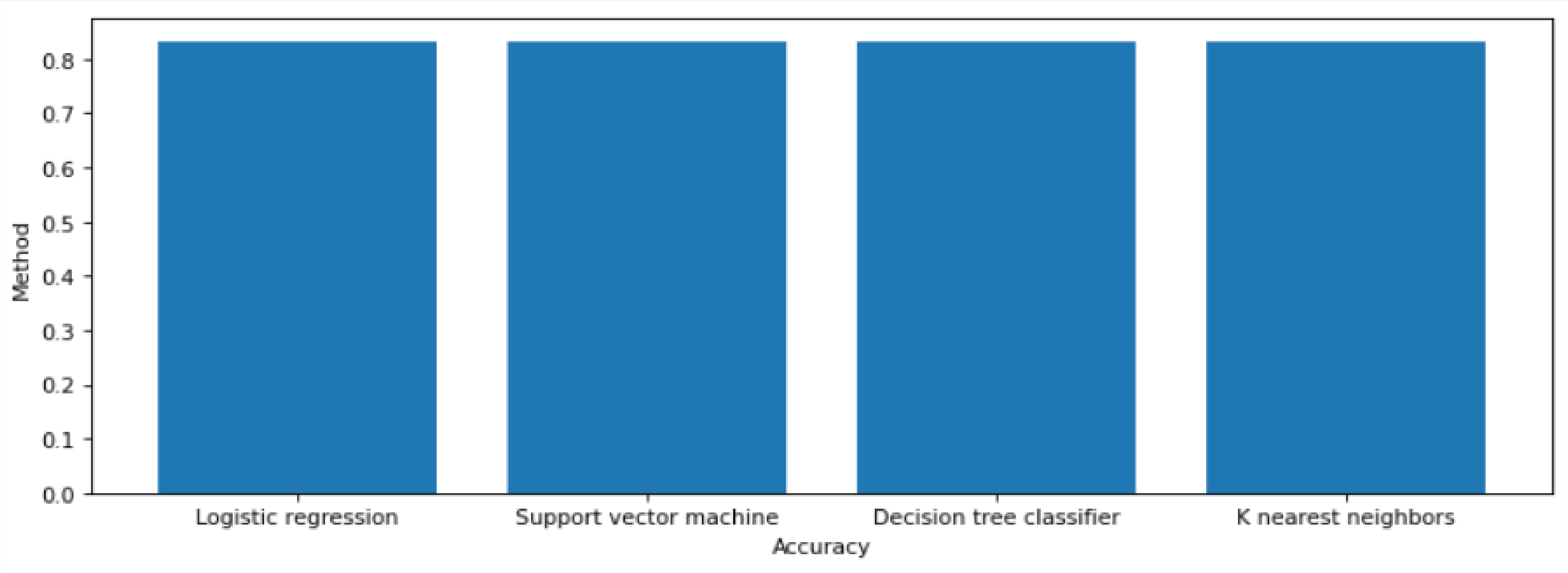


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

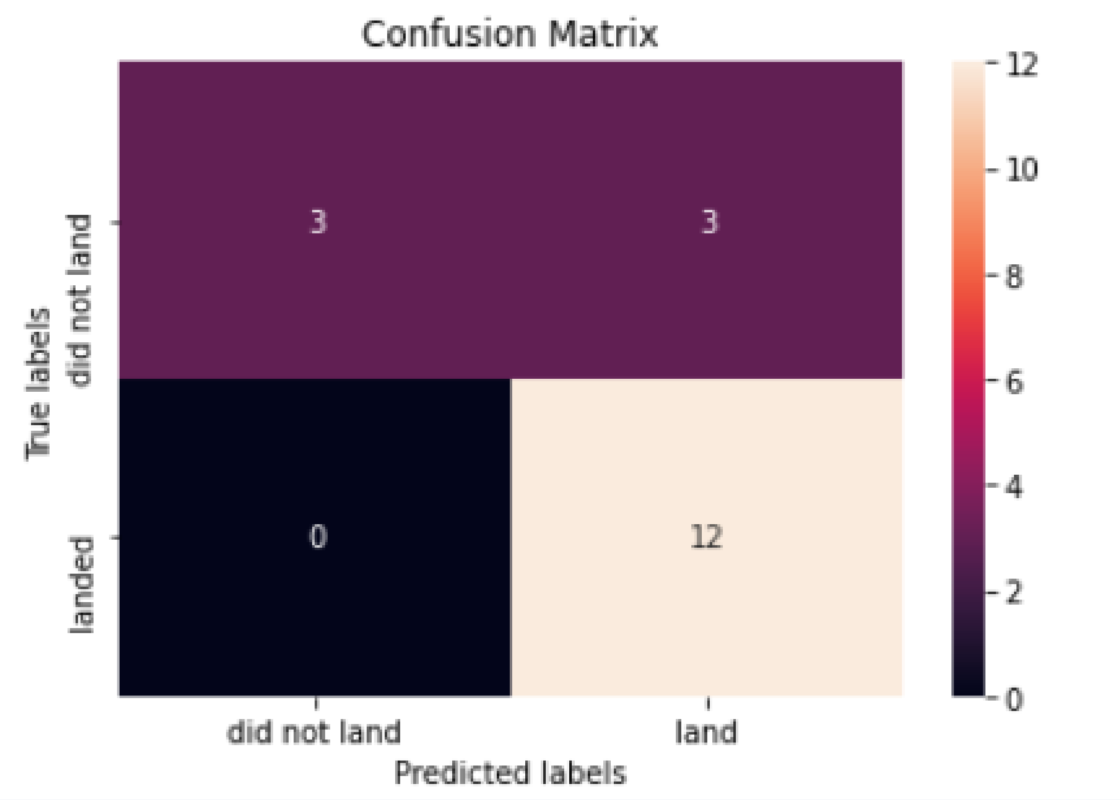
Classification Accuracy



- In the test set, **the accuracy of all models** was virtually the **same** at **83.33%**.
- It should be noted that the test size was small at 18. Therefore, more data is needed to determine the optimal model

Confusion Matrix

- The confusion matrix is the same for all models because all models performed the same for the test set.
- The models predicted 12 successful landings when the true label was successful and 3 failed landings when the true label was failure.
- But there were also 3 predictions that said successful landings when the true label was failure (false positive).
- Overall, these models predict successful landings



Conclusions

- As the number of flights increased, the success rate increased, and recently it has exceeded 80%.
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).
- The launch site is close to railways, highways, and coastline, but far from cities.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- In this dataset, all models have the same accuracy (83.33%), but it seems that more data is needed to determine the optimal model due to the small data size

Appendix

https://github.com/manisangsu/mammoth_falcon

Thank you!