Winning Space Race with Data Science

Utkarsh Gaikwad

23rd July 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 Visualization
  - Building interactive Map with Folium
  - Building dashboard using Plotly and Dash
  - Predictive Analysis (Classification using GridSearchCV)
- Summary of all results
  - EDA Results
  - Interactive Analytics
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX organization markets to sell rocket launches at $62 million while the other competitors require at least $160 million and higher. This is possible because SpaceX reuses the Stage 1 of each rocket Launch.
- Problems you want to find answers
  - The problem here is to predict that stage 1 can land successfully or not so that it can be reused.

Section 1

Methodology

Methodology

Executive Summary

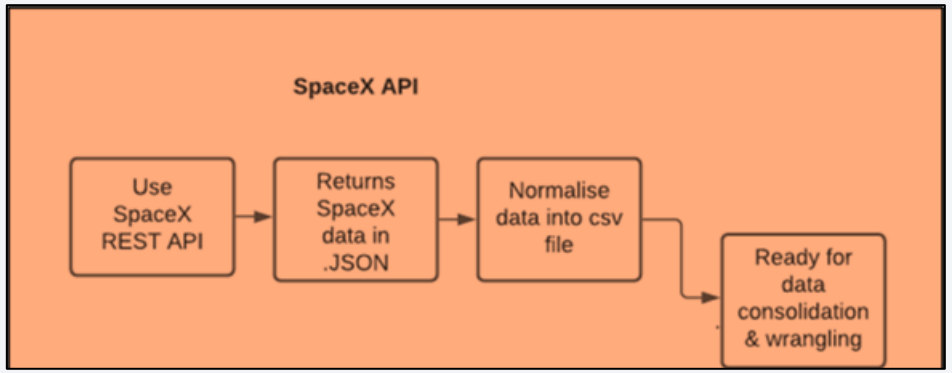
- Data collection methodology:
  - Data was collected by using SpaceX API : https://api.spacexdata.com/v4/rockets/
- Perform data wrangling
  - Missing Values for PayloadMass was replaced by mean values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was collected from Rest API of SPACEX.
- Data was also collected from web scraping of Wikipedia Page on SpaceX

Data Collection – SpaceX API

- • 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/.
- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection – SpaceX API

- Data collection with SpaceX REST calls
- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Getting Response from API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

In [7]:
1 response = requests.get(spacex_url)
```

Getting json from request and normalizing it to Pandas DataFrame

```
In [14]:
1 # Use json_normalize meethod to convert the json result into a dataframe
2 data_json = response.json()
3 data = pd.json_normalize(data_json)
```

Applying Functions to Clean Data

```
In [16]:
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have m
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

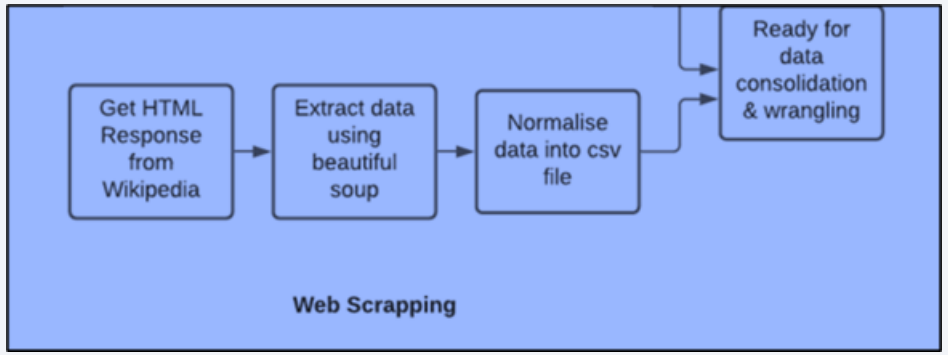
Converting Dictionary to Clean DataFrame

Then, we need to create a Pandas data frame from the dictionary launch_dict.

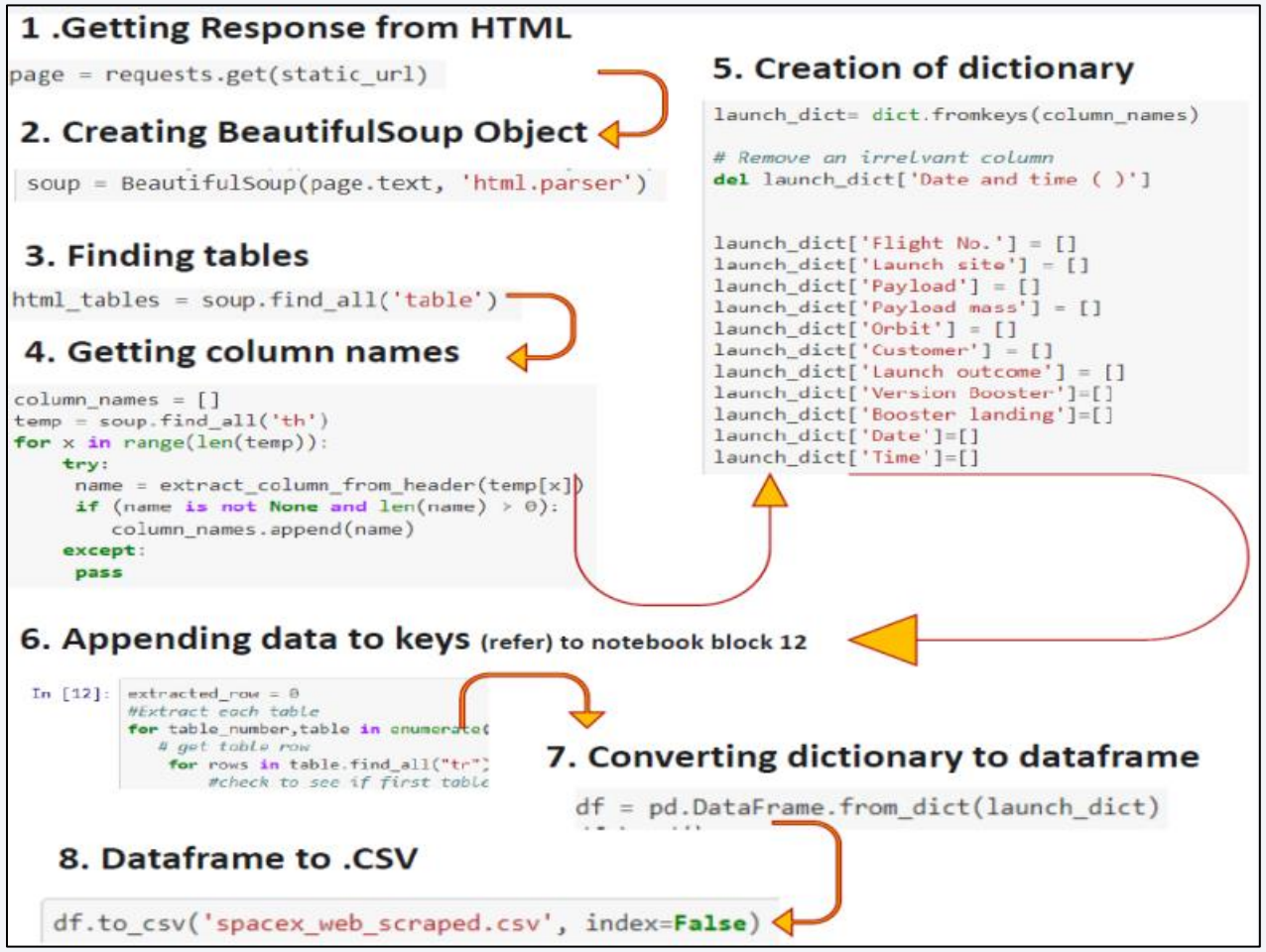
```
In [25]:
1 # Create a data from launch_dict
2 data_falcon = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

- Web Scrapping from Wikipedia



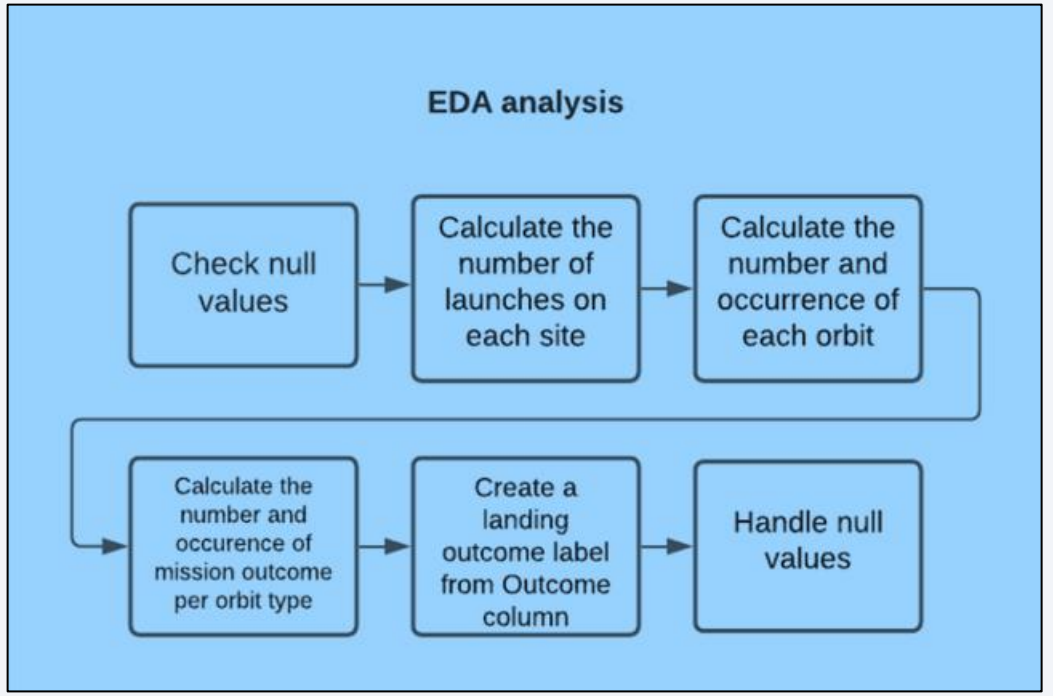
- URL: https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Data Wrangling Process

- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



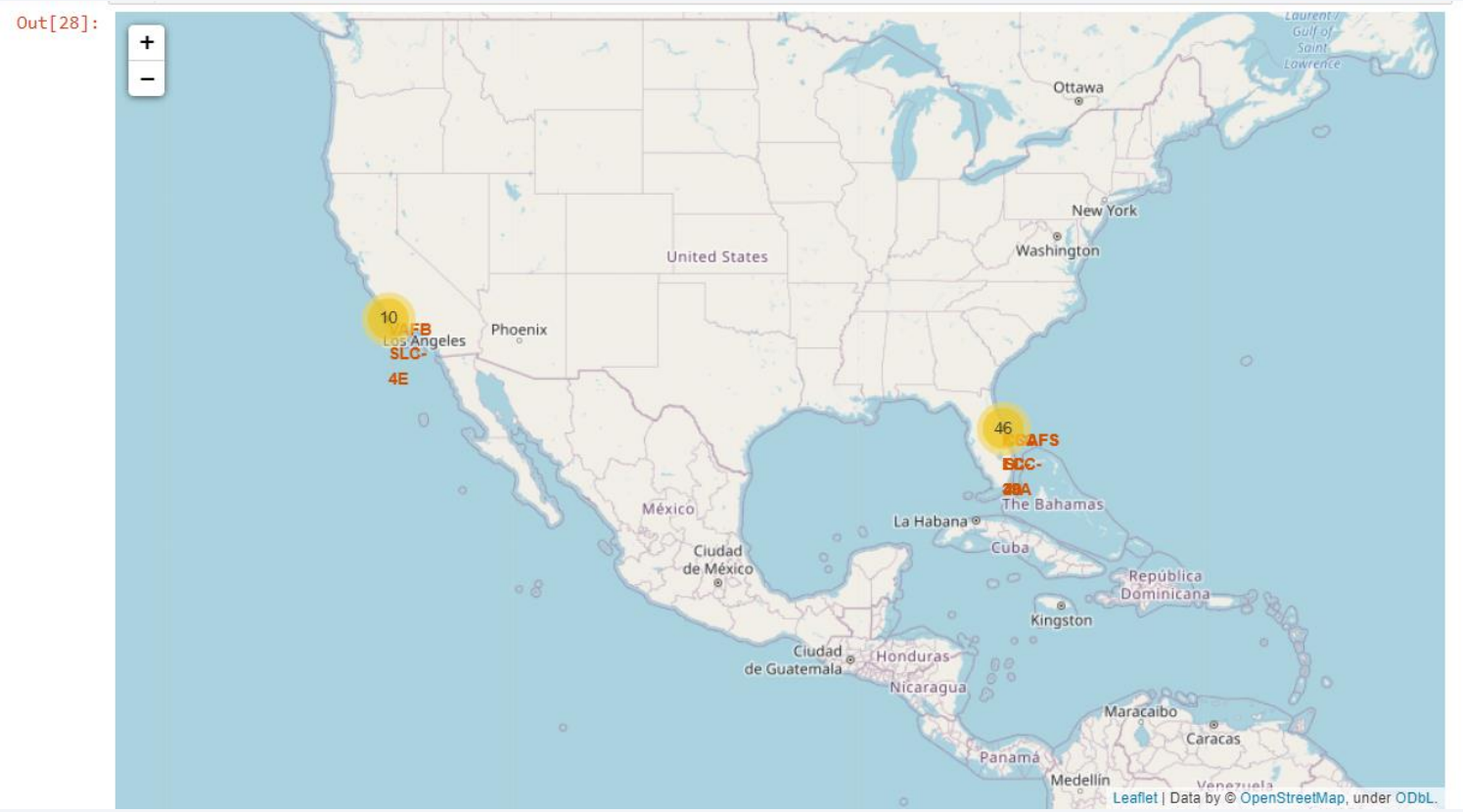
EDA with Data Visualization

- Catplot and bar charts were plotted in the Visualization step
- Flight Number was compared with Payload Mass and Launch Site to check where success rate was good
- Bar plot of Success rate for each orbit type was visualized
- Catplot for FlightNumber and Orbit type was plotted
- Catplot for Payload and Orbit type was plotted
- Line Chart of success rate over various years was plotted
- URL: https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

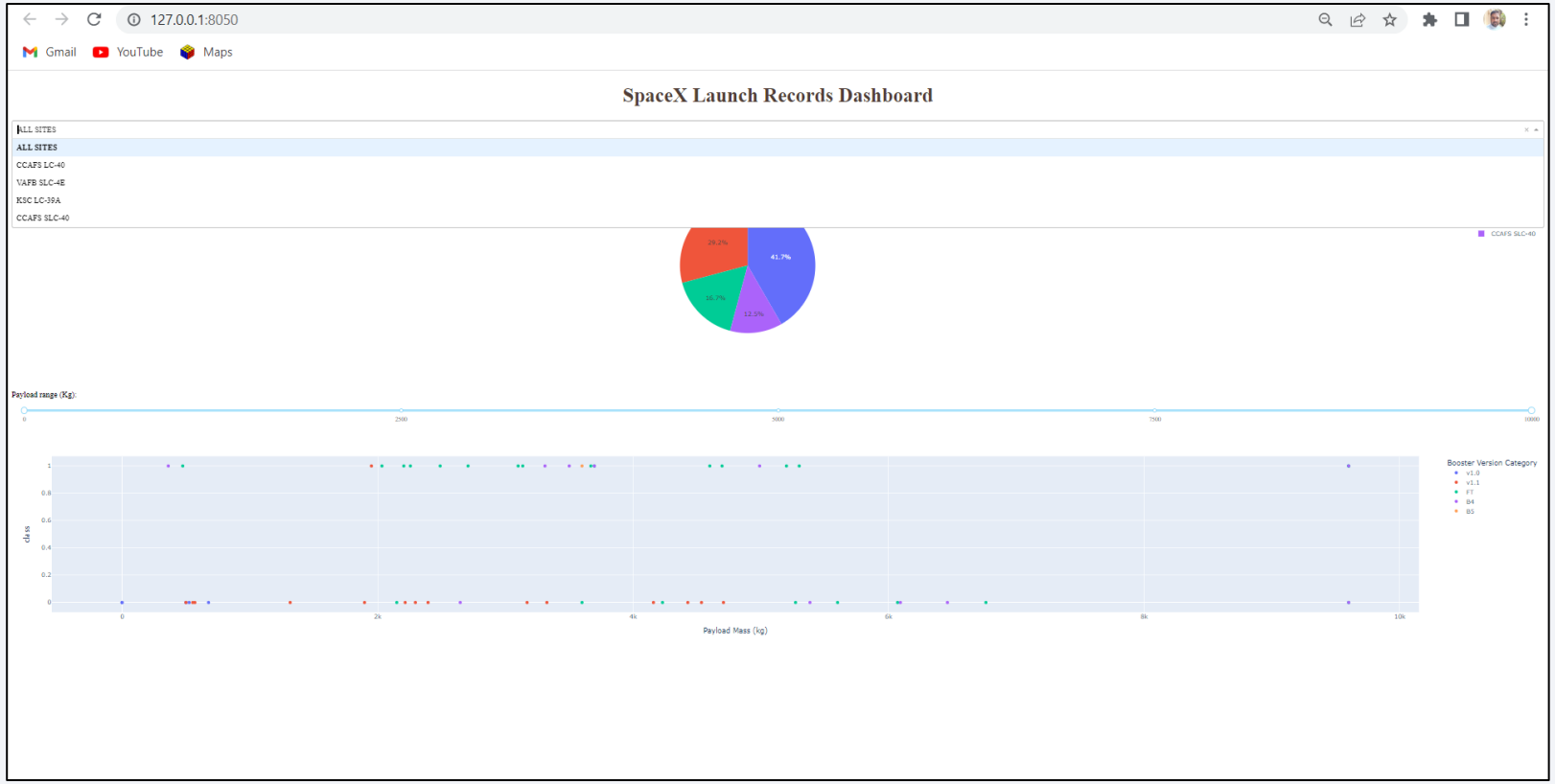
- Below SQL queries were Executed :
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string ‘CCA’
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - 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 the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order
- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium



- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

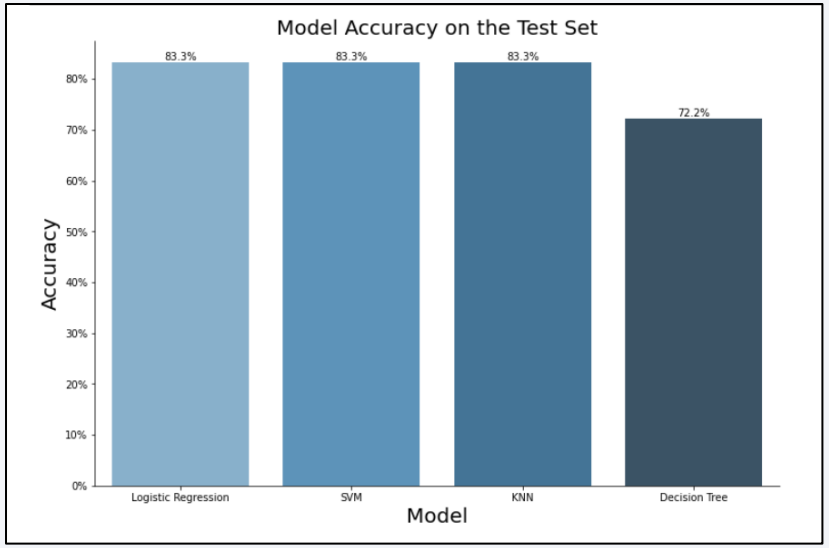
Build a Dashboard with Plotly Dash



- URL: https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/Dashboard%20SpaceX.ipynb

Predictive Analysis (Classification)

- The SVM, KNN, and Logistic Regression model achieved the highest accuracy at 83.3%



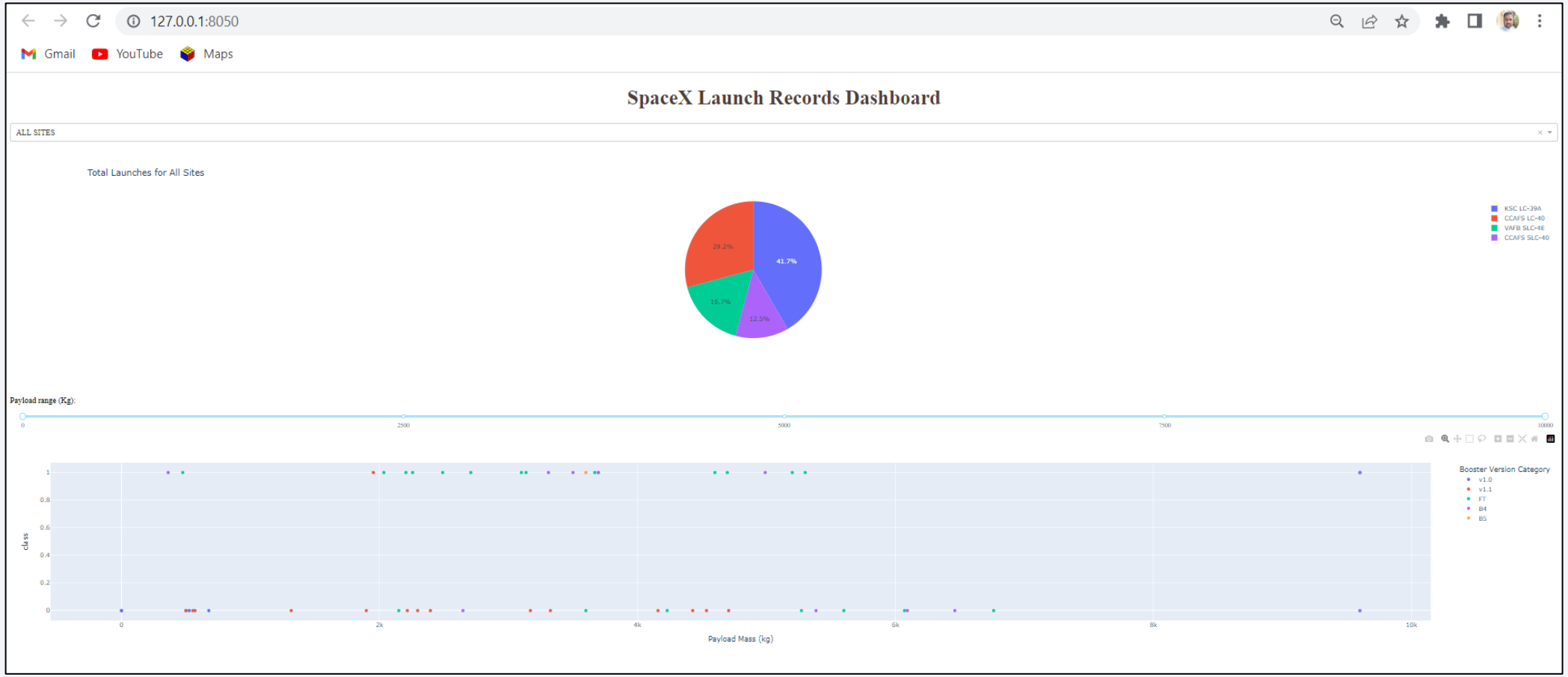
- URL : https://github.com/utkarshg1/IBM-Capestone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

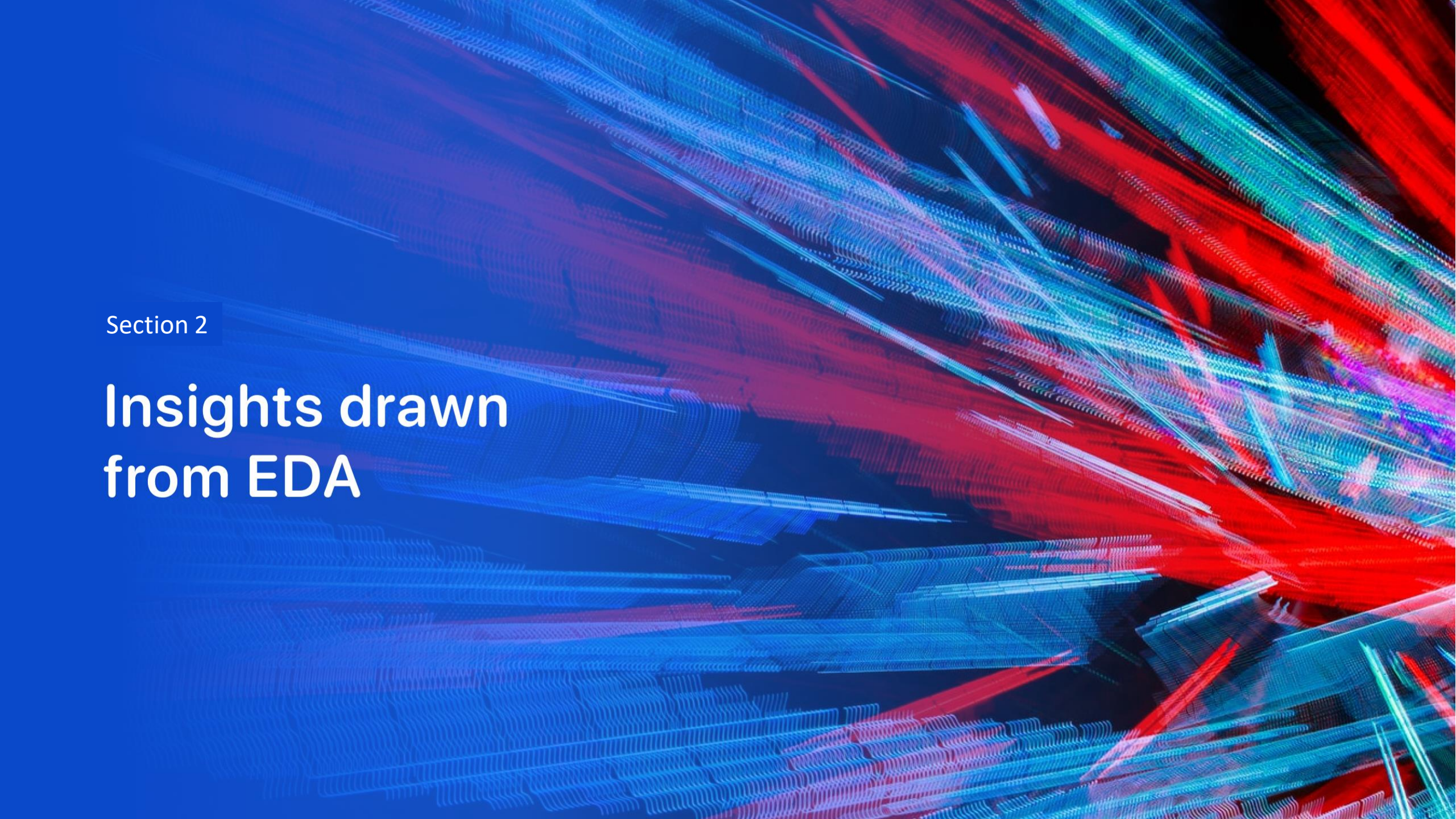
Results

- Exploratory data analysis results
  - KSC LC 39A had the most successful launches from all the sites
  - Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.
  - Low weighted payloads perform better than the heavier payloads
- Predictive analysis results
  - The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset

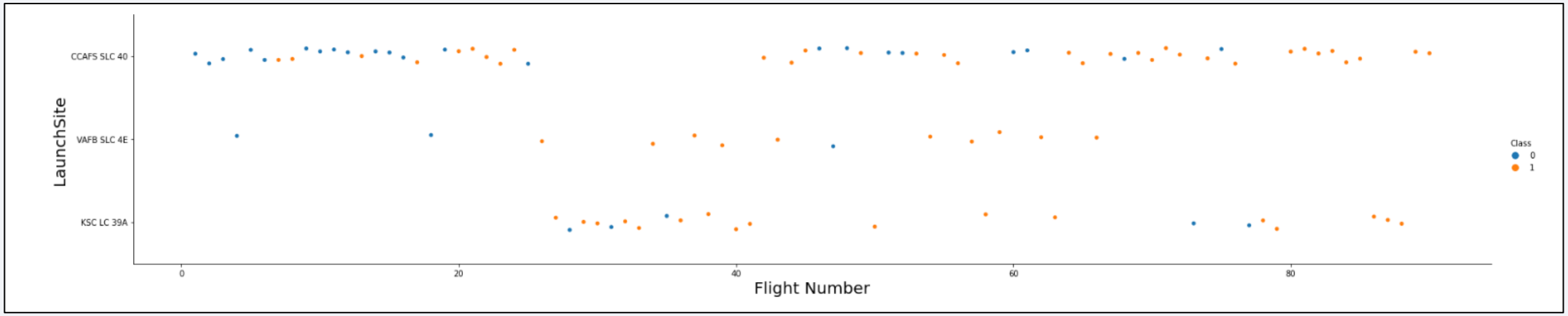
Results

- Dashboard Screenshot



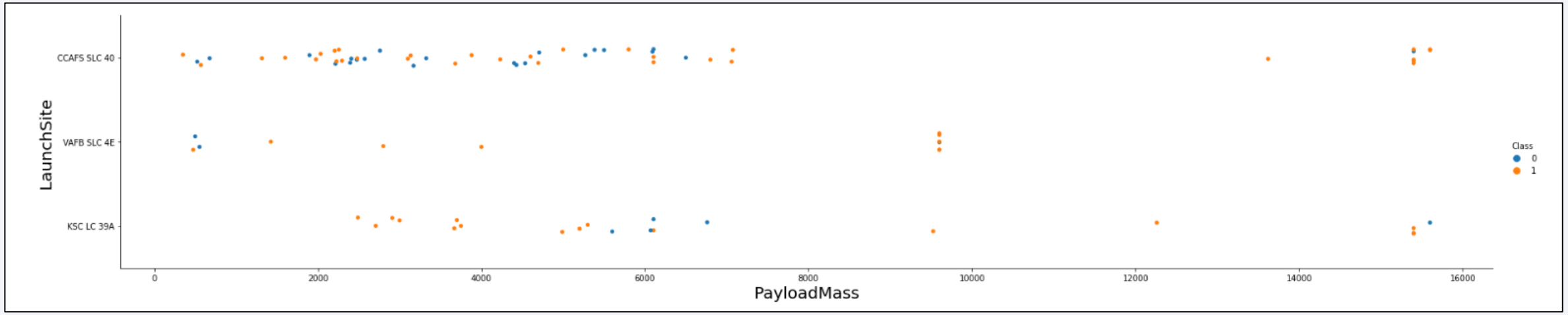

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



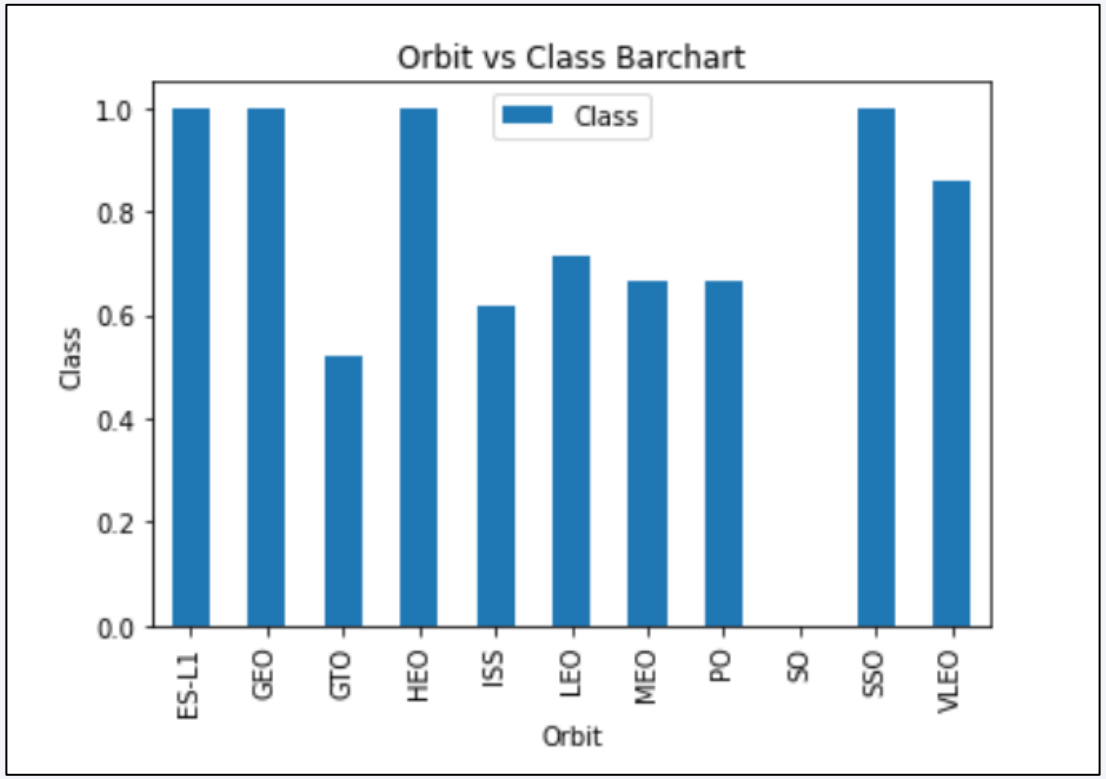
Launches from the site of CCAFS SLC 40 are significantly higher than launches form other sites.

Payload vs. Launch Site



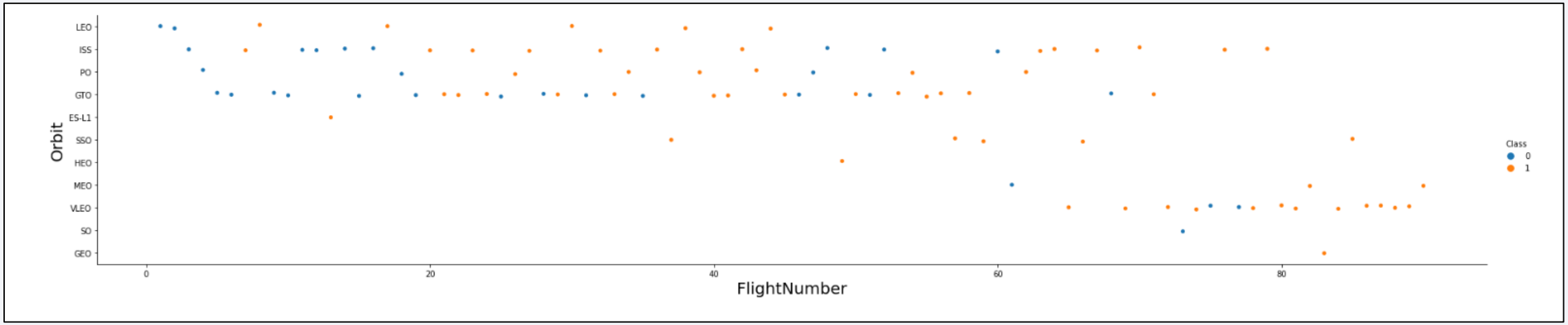
For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



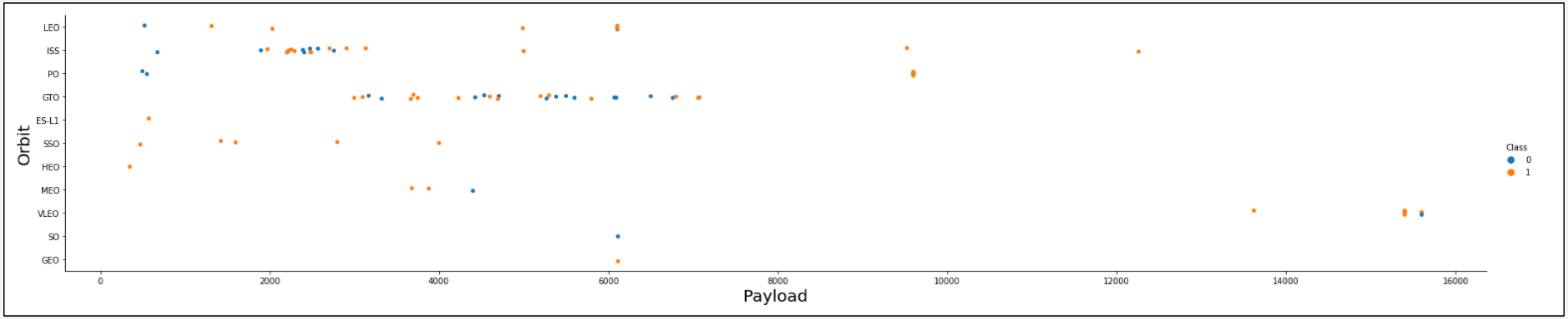
Success rates are highest for Orbits ES-L1, GEO, HEO and SSO
SO orbit has 0 success rate

Flight Number vs. Orbit Type



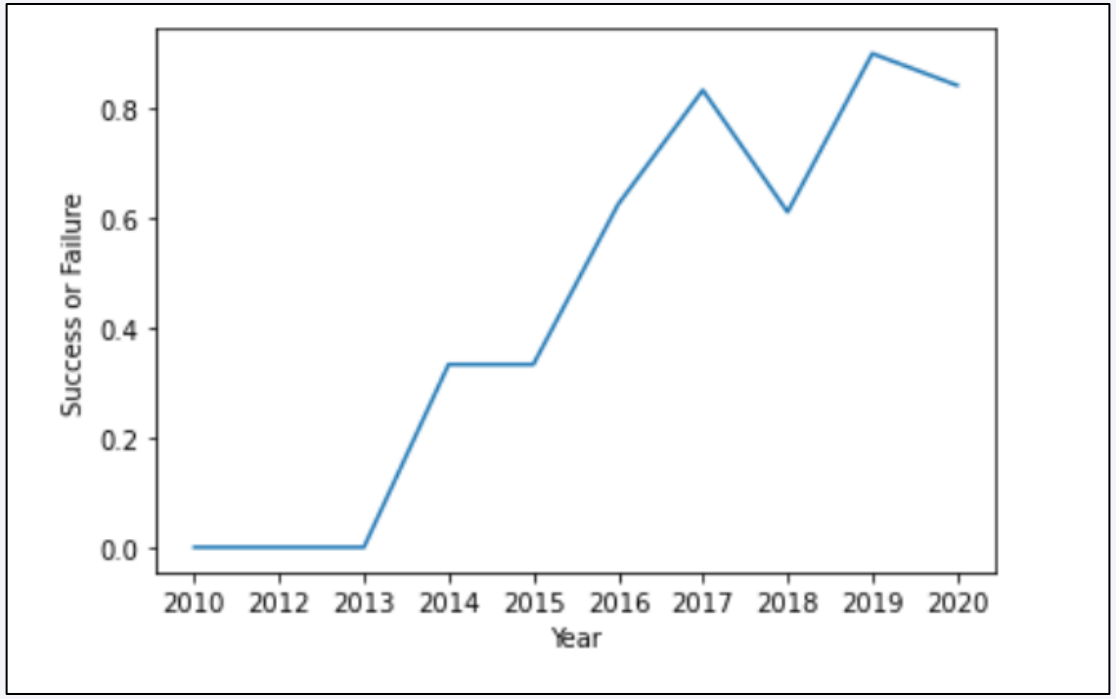
LEO orbit the Success appears related to the number of flights;
on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS

Launch Success Yearly Trend



Success rate since 2013 kept increasing till 2020

All Launch Site Names

- %sql select distinct(Launch_Site) from SPACEXTBL

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

Launch Site Names Begin with 'CCA'

- %sql select * from SPACEXTBL where Launch_Site Like 'CCA%' Limit 5

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)'

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

- %sql select avg(PAYLOAD_MASS__KG_) as AVG_Payload from SPACEXTBL where Booster_Version='F9 v1.1'

AVG_Payload
2928.4

First Successful Ground Landing Date

- %sql select min(date),[Landing _Outcome] from SPACEXTBL/ where [Landing _Outcome]='Success (ground pad)' ;

min(date)	Landing _Outcome
01-05-2017	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql select Booster_Version, PAYLOAD_MASS__KG_, [Landing _Outcome]\
  from SPACEXTBL\
  where (PAYLOAD_MASS__KG_ between 4000 and 6000) and \
  [Landing _Outcome]='Success (drone ship)';

Booster_Version	PAYLOAD_MASS__KG_	Landing _Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- %sql select Mission_Outcome,count(*) as Mission_count \
  from SPACEXTBL \
  group by Mission_Outcome;

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

Boosters Carried Maximum Payload

- %sql select Booster_Version,PAYLOAD_MASS__KG_ \
  from SPACEXTBL \
  where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

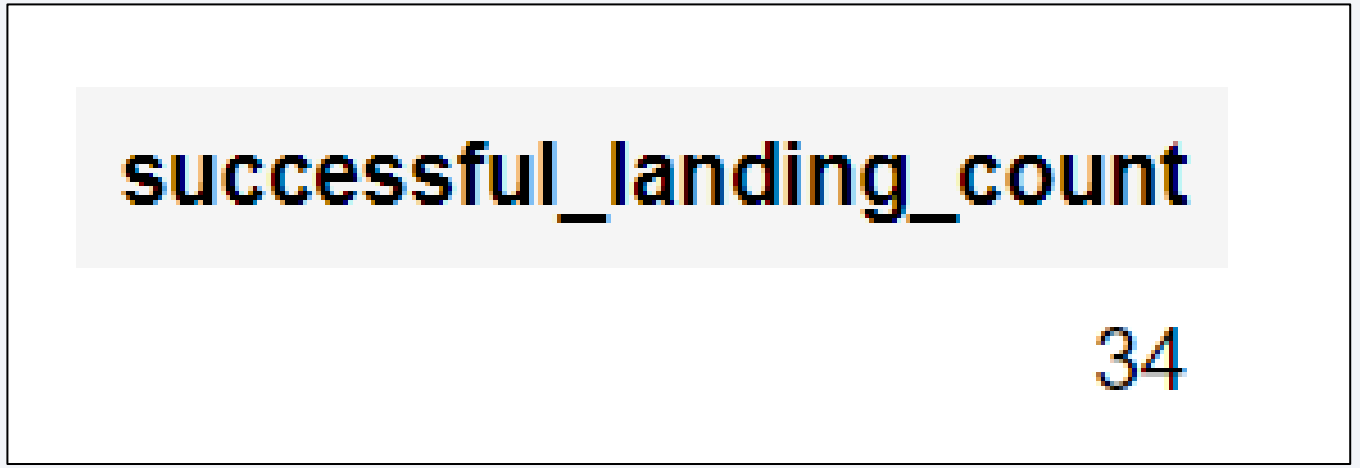
2015 Launch Records

- %sql select DATE, BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
  From SPACEXTBL \
  where [Landing _Outcome] = 'Failure (drone ship)' and \
  substr(Date,7,4)='2015';

Date	Booster_Version	Launch_Site	Landing _Outcome
10-01-2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
14-04-2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- %sql select count(*) as successful_landing_count from SPACEXTBL \
  where [Landing _Outcome] like 'Success%' and \
  Date between '04-06-2010' and '20-03-2017';

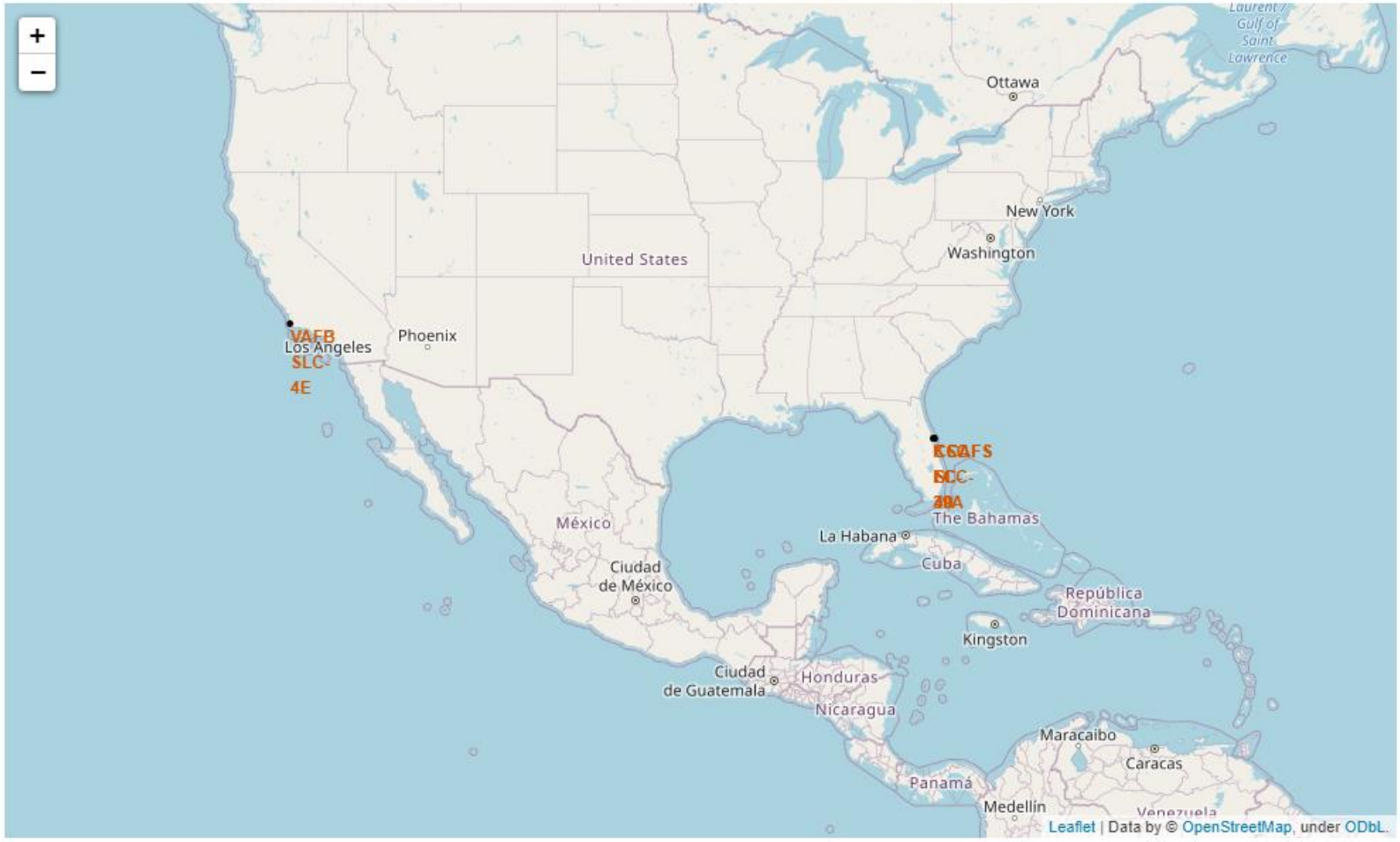


Section 3

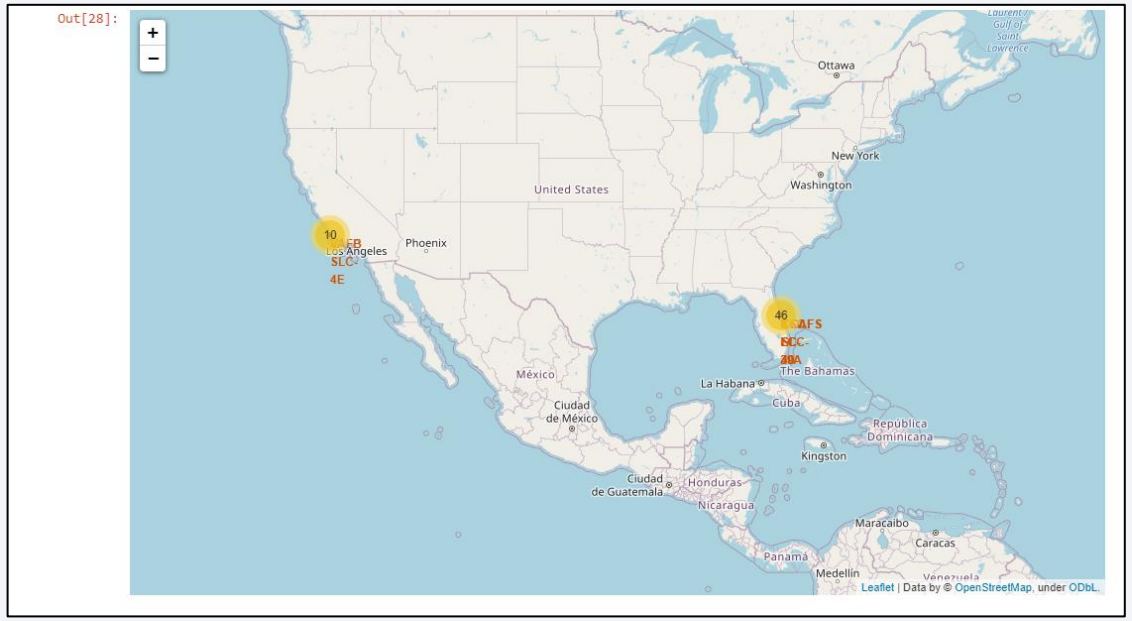
Launch Sites Proximities Analysis

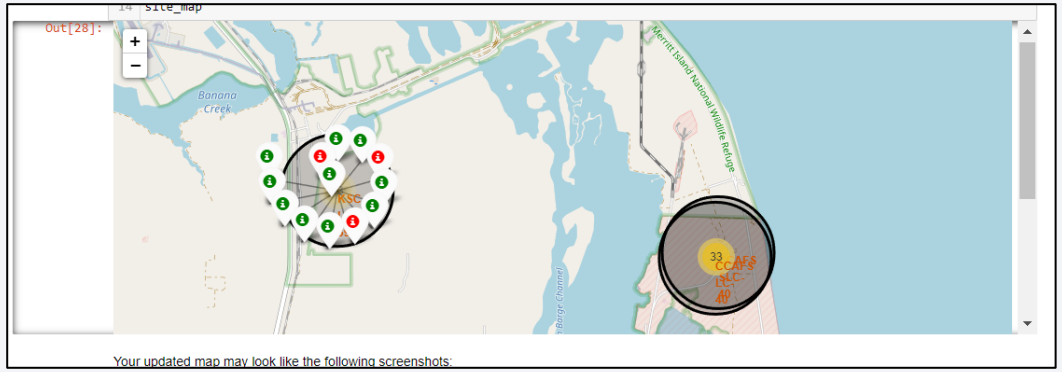
All launch sites marked on a map

Out[23]:

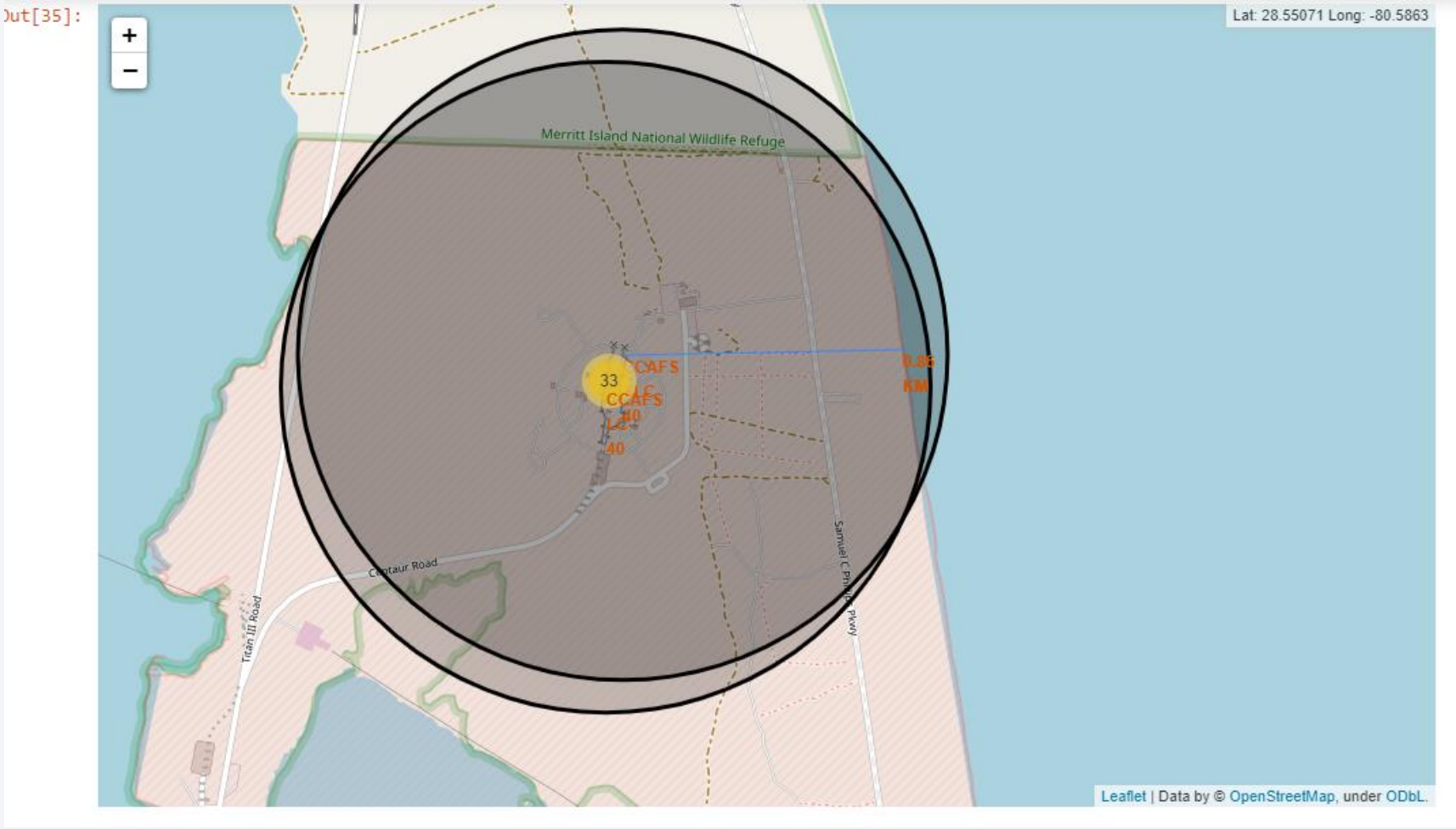


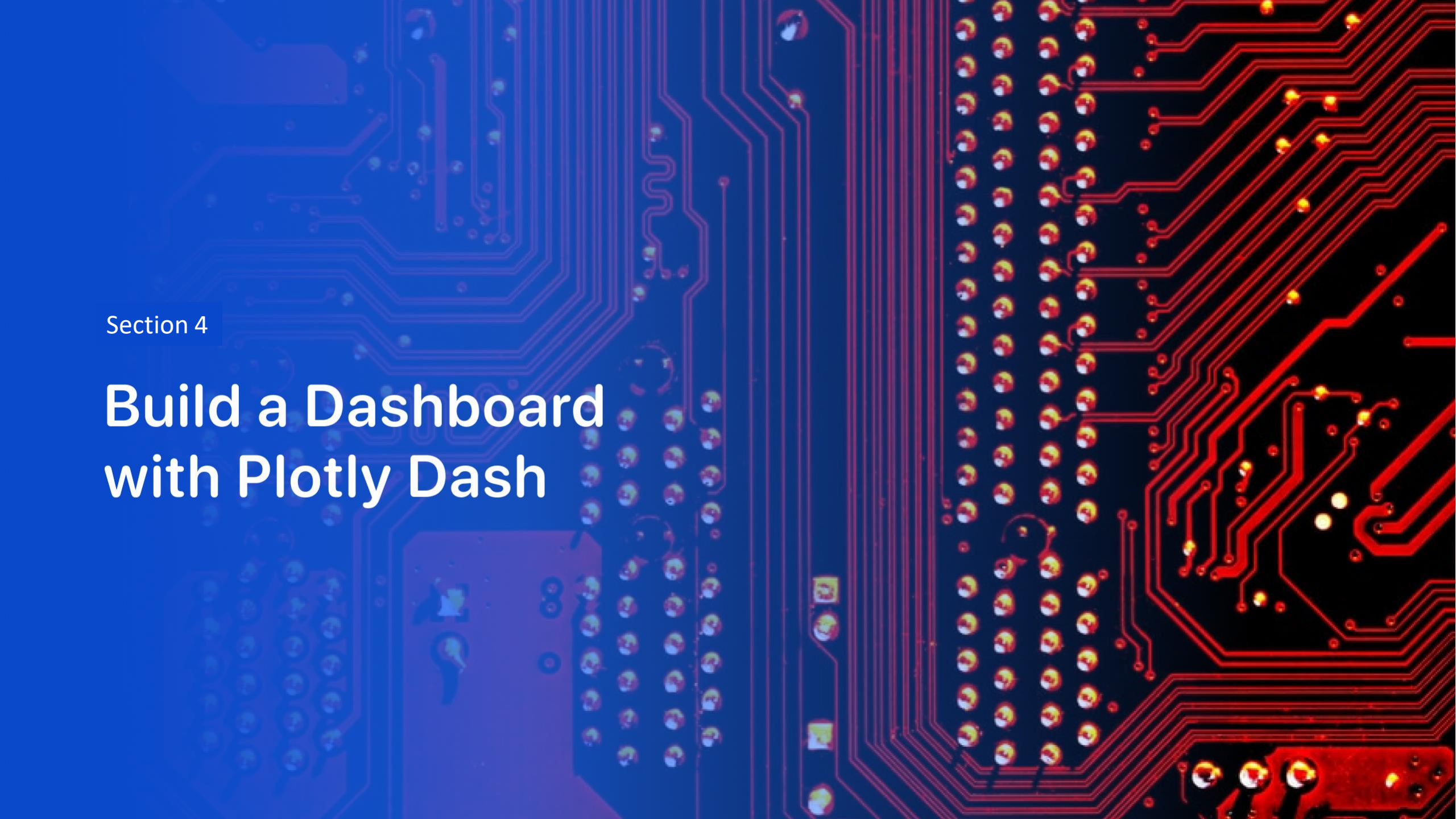
Success and Failure marked for each site



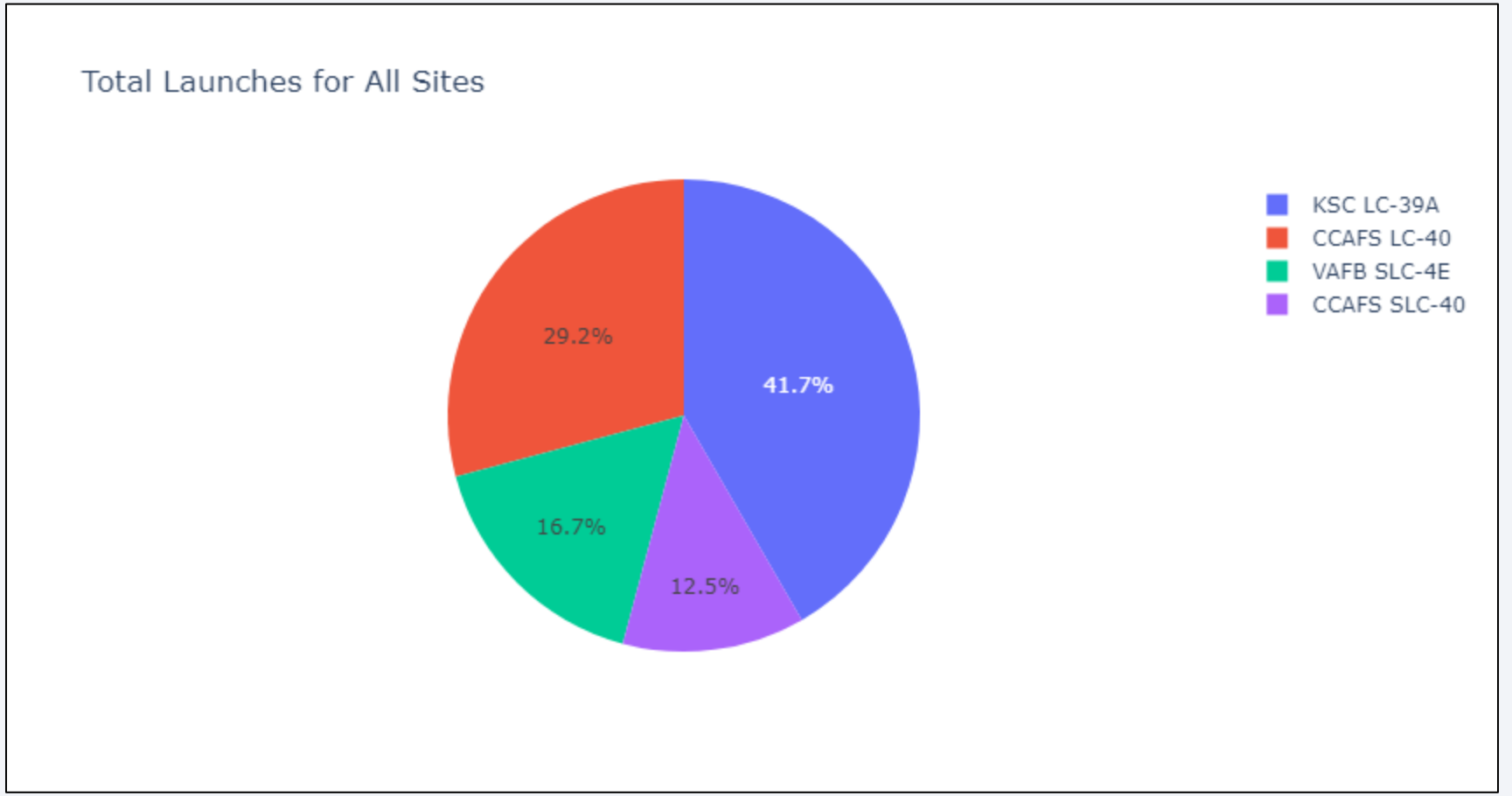


Distance added from coastline



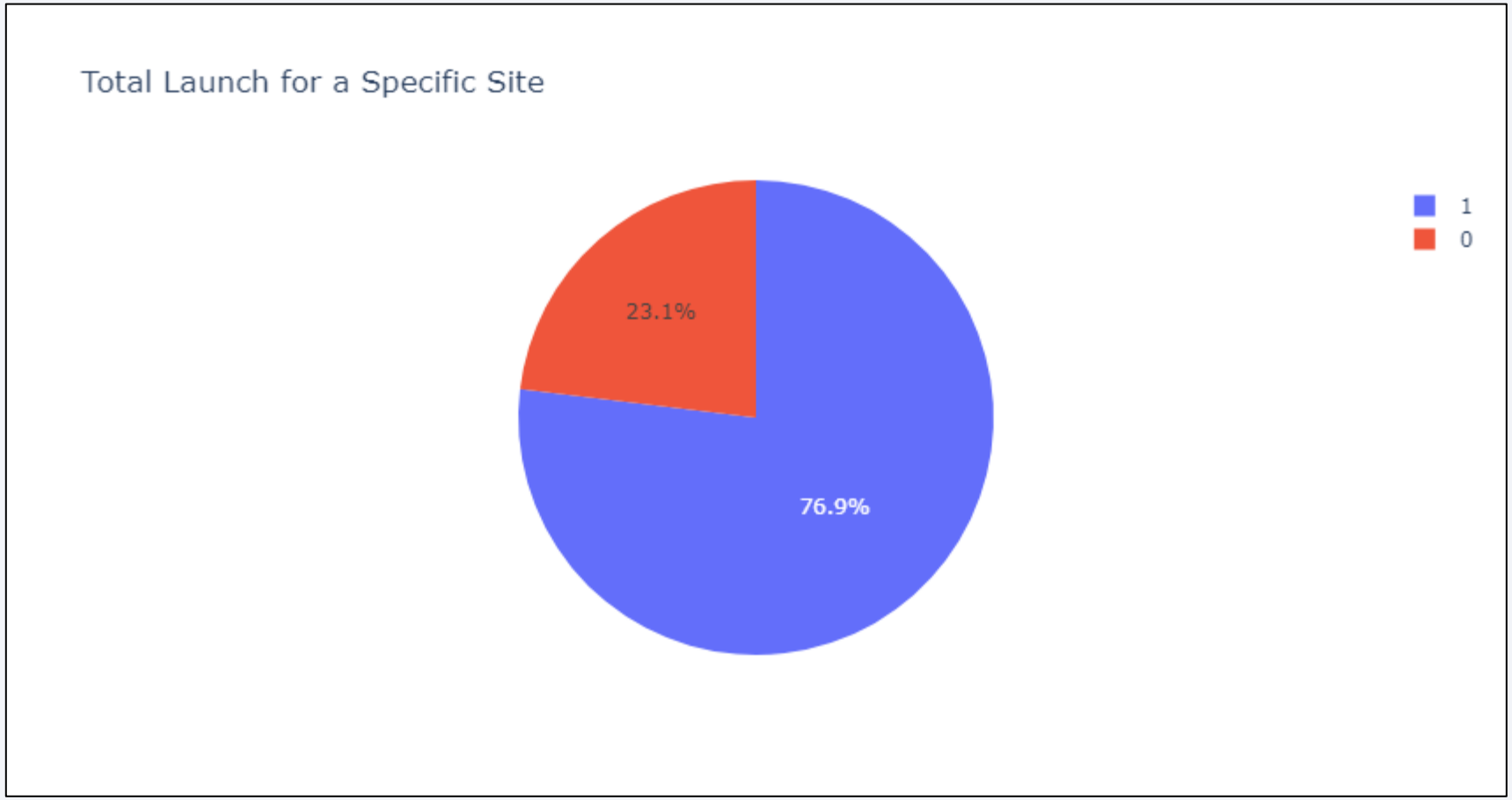

Section 4
Build a Dashboard with Plotly Dash

Total success launches by all sites



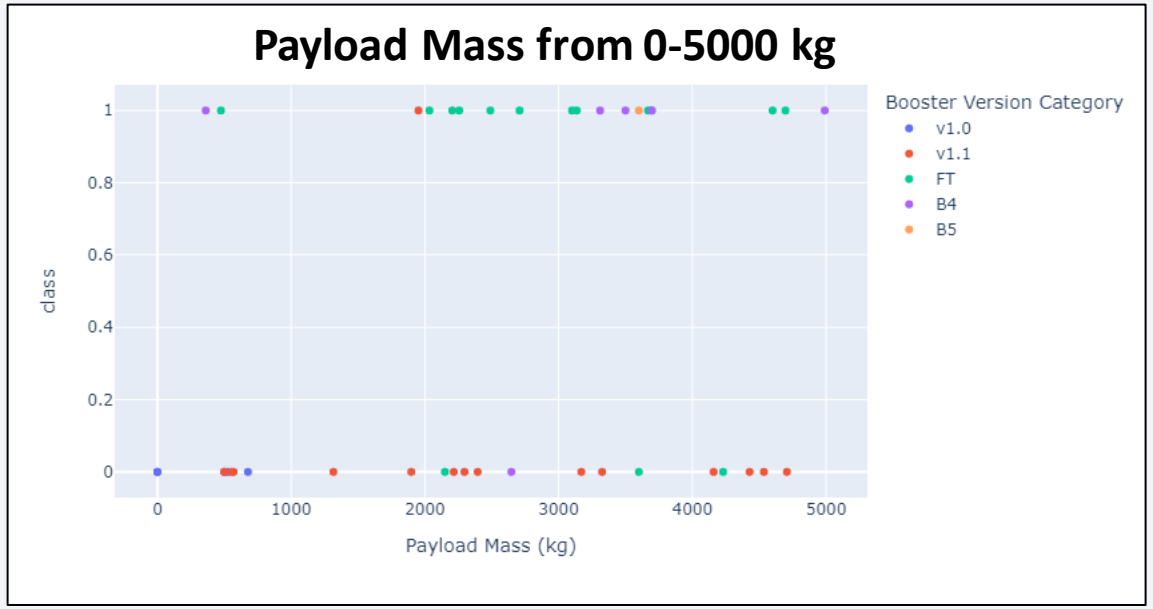
KSC LC-39A has highest successful Launches

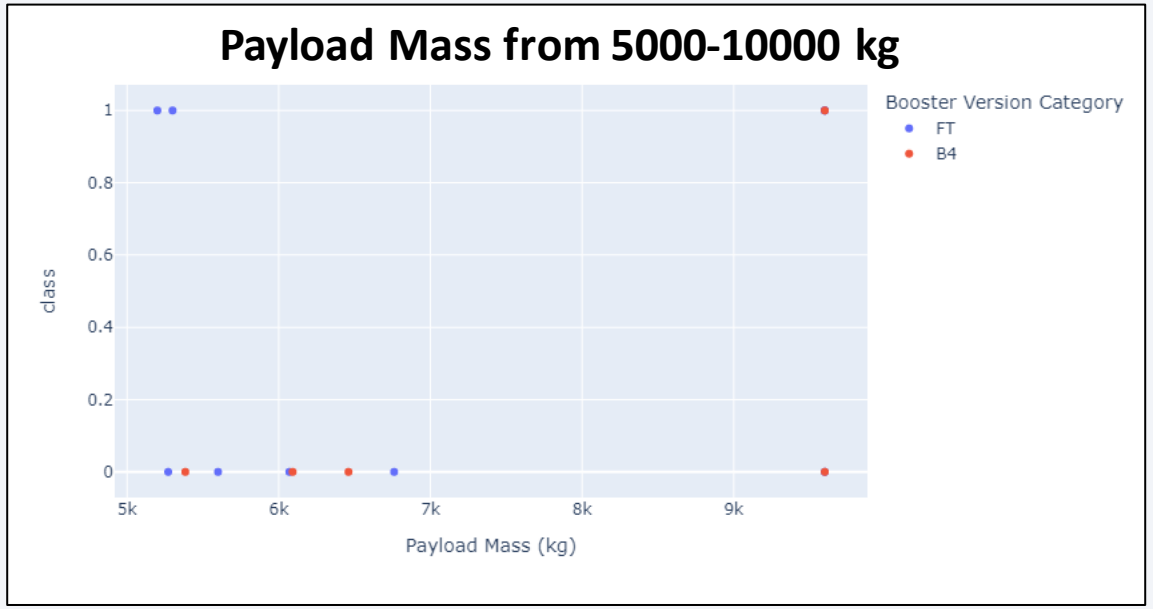
Success rate by site



KSC LC-39A has 76.9% success Rate

Payload vs launch outcome



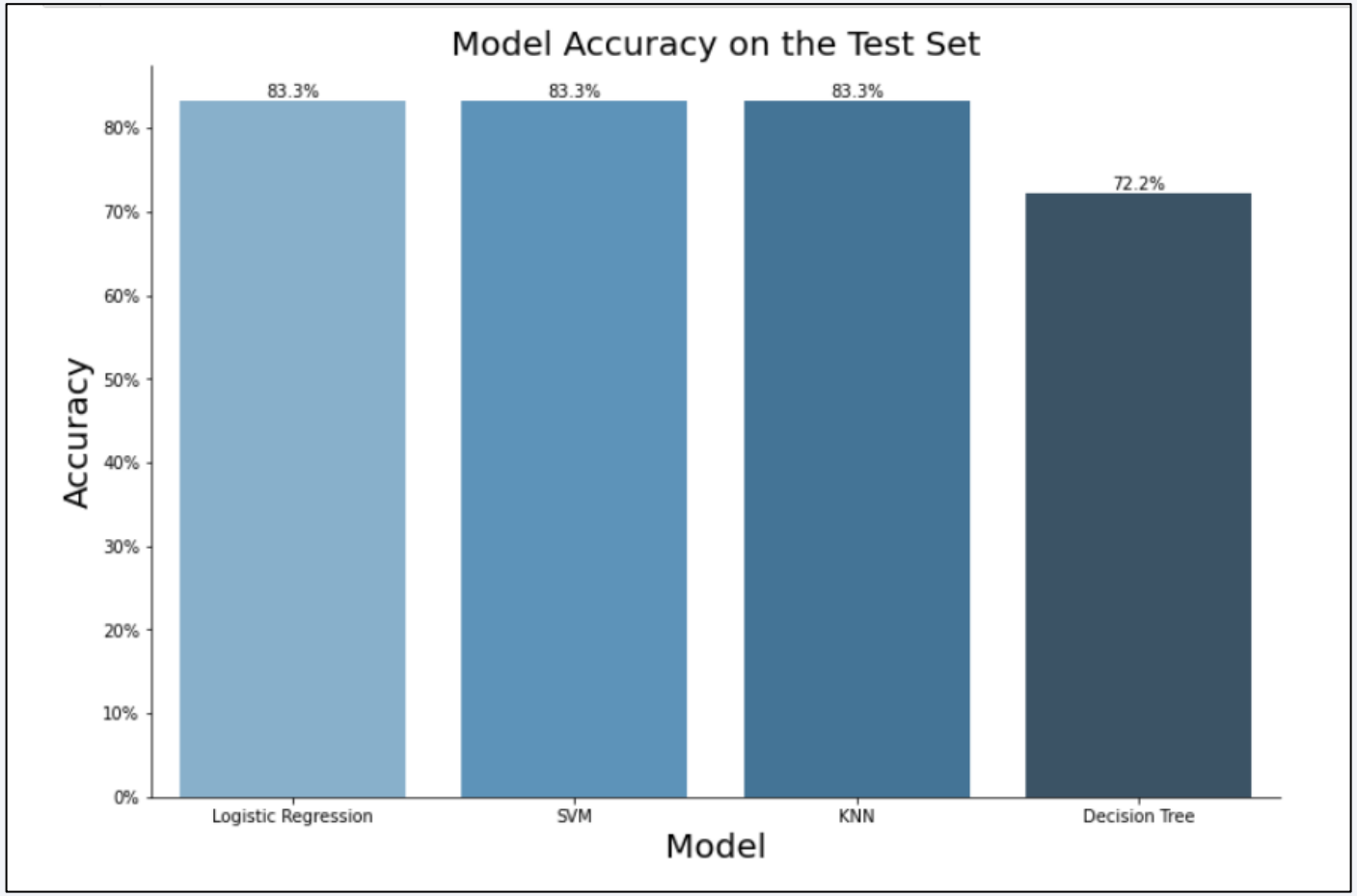


There are less payload mass above 5000 kg compared to below 5000kg

Section 5

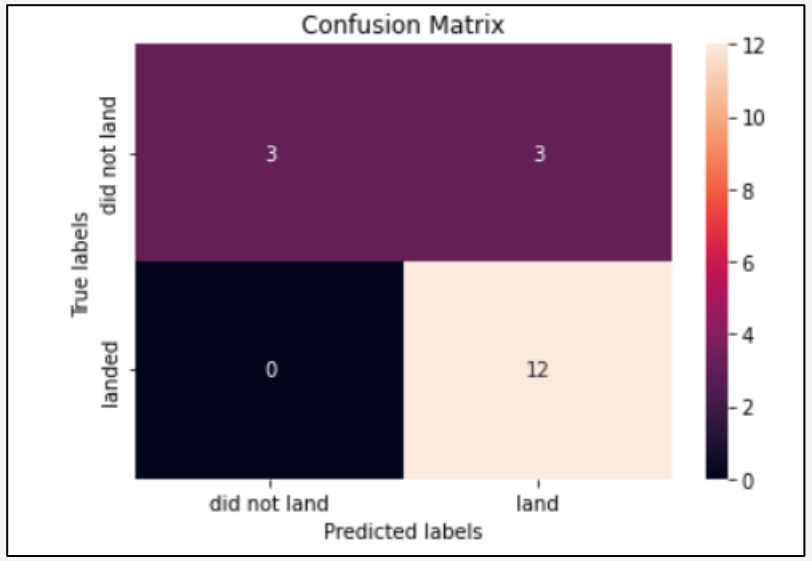
Predictive Analysis (Classification)

Classification Accuracy

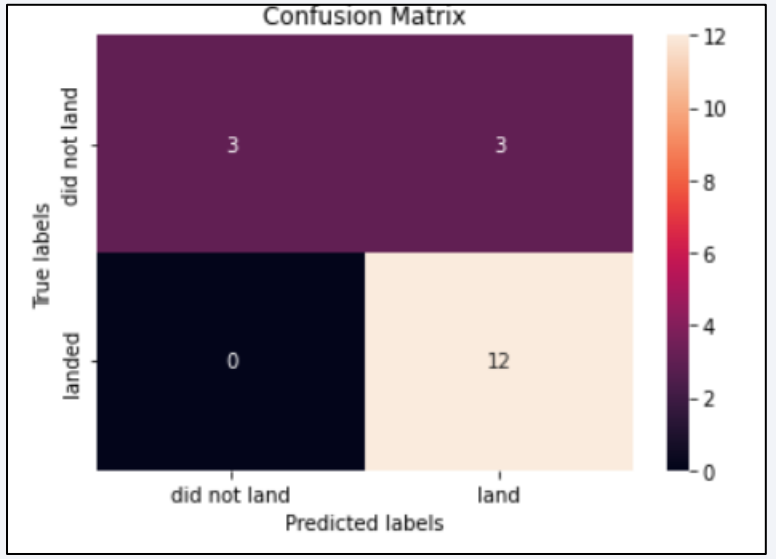


Confusion Matrix

Logistic regression



SVM



Conclusions

- KSC LC 39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- Low weighted payloads perform better than the heavier payloads.
- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset

Thank you!