Winning Space Race with Data Science

Nirmal Kumar

5th April 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API
  - Data Collection via Web Scrapping
  - Data Wrangling
  - Exploratory Data Analysis with Visualization
  - Exploratory Data Analysis with SQL
  - Visualization by Map-Folium and Plotly
  - Classification Modelling
- Summary of all results
  - Exploratory Data Analysis Results
  - Classification Modelling Results

Introduction

Project background and context

Space X is leading commercial space age by making rocket launches cost effective compared to its competitors. It can do so by reusing the first stage. First stage happens to be quite large and most expensive. If we can determine if the first stage can land, we can determine the cost of launch.

Problems you want to find answers

- Gather information about Space X launch data
- Create machine learning model to determine if the first stage will land successfully and can be reused

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and Web scrapping from Wikipedia
- Perform data wrangling
  - Raw data was converted into meaningful features by creating the success class
- 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 primarily via two methods
- GET request
  - Data was collected using get request to the SpaceX API
  - Response was converted to pandas dataframe
  - Further data was filtered, data wrangling methods were applied, missing values were replaced with averages and saved into csv file for further processing
- Web Scrapping
  - Data was extracted from Wikipedia using request and beautiful soup
  - Data was further captured in pandas dataframe from HTML table

Data Collection – SpaceX API

- Data was collected using get request to the SpaceX API
- Response was converted to pandas dataframe
- Further data was filtered, data wrangling methods were applied, missing values were replaced with averages and saved into csv file for further processing
- GitHub URL SpaceX API calls notebook - CapstoneProject/jupyter-labs-spacex-data-collection-api.ipynb at main · nkr1108/CapstoneProject (github.com)

Step 1: GET request for rocket launch data using API

Step 2: Transform the response into Pandas Data frame using .json_normalize()

Step 3: Take relevant data like Rocket, Payloads, Launchpad, cores, flight number and date from the overall data

Step 4: Filter only Falcon 9 launches

Step 5: Perform basic data wrangling ie identifying and taking care of missing values

Step 6: Saving the data into csv file for further analysis

Data Collection - Scraping

- Data was extracted from Wikipedia using request and beautiful soup
- Data was further captured in pandas dataframe from HTML table
- GitHub URL web scraping notebook - CapstoneProject/jupyter-labs-webscraping.ipynb at main · nkr1108/CapstoneProject (github.com)

Step 1: Use HTTP GET method to request Falcon9 Launch HTML page

Step 2: Create a BeautifulSoup object from the HTML response

Step 3: Extract Column Names and Column Data from Table Header and Table Data

Step 4: Convert the data into Pandas Dataframe and save it into csv file for further analysis

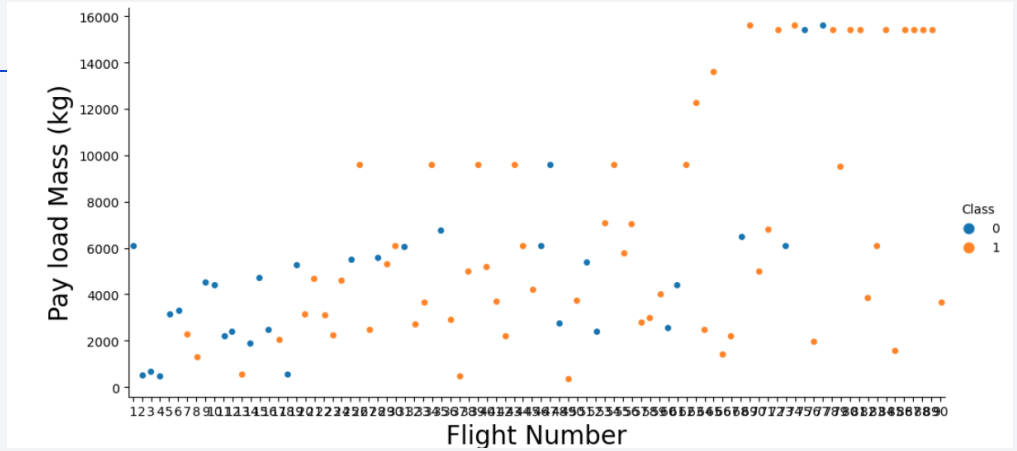
Data Wrangling

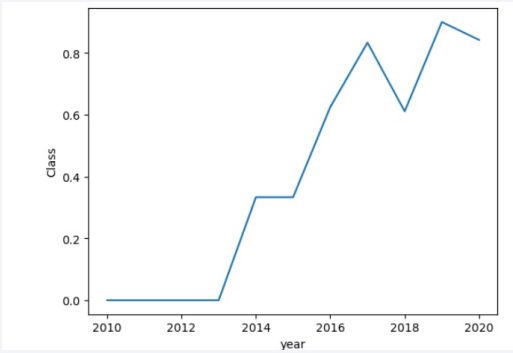
- As part of Data Wrangling following activities were performed
  - Basic Data Analysis was done which includes identifying %age of missing values
  - Number of launches per site was calculated
  - Number of occurrence of each orbit was calculated
  - Missing values were replaced with average value
  - Create Landing outcome label from Outcome Column
  - Success rate was calculated using the Landing outcome
- GitHub URL data wrangling related notebook - CapstoneProject/labs-jupyter-spacex-Data wrangling.ipynb at main · nkr1108/CapstoneProject (github.com)

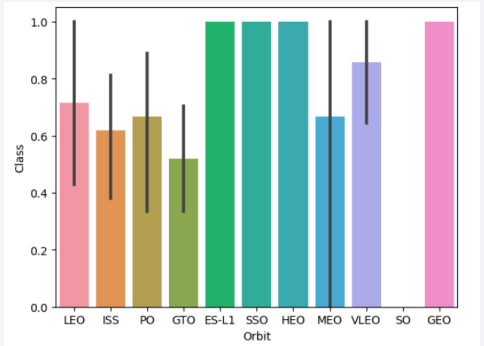
EDA with Data Visualization

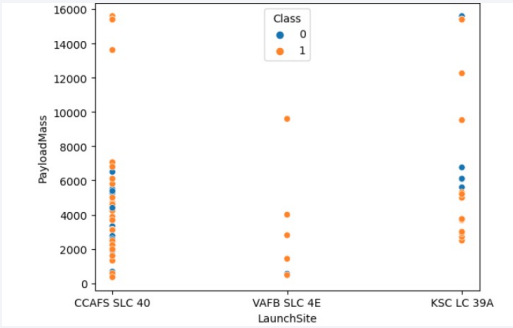
- Different Scatter plots were created between Payload vs Flight Number, Flight Number vs Launch Site, Payload Mass vs Launch Site
- Relationship between success rate of each orbit type
- Visualization of relationship between Flight Number and Orbit Type
- Visualization of relationship between Payload and Orbit Type
- Yearly trend of success rate
- GitHub URL EDA with data visualization notebook

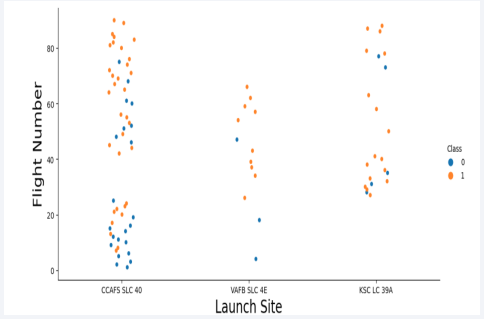
  - [CapstoneProject/jupyter-labs-eda-dataviz.ipynb at main · nkr1108/CapstoneProject (github.com)]











EDA with SQL

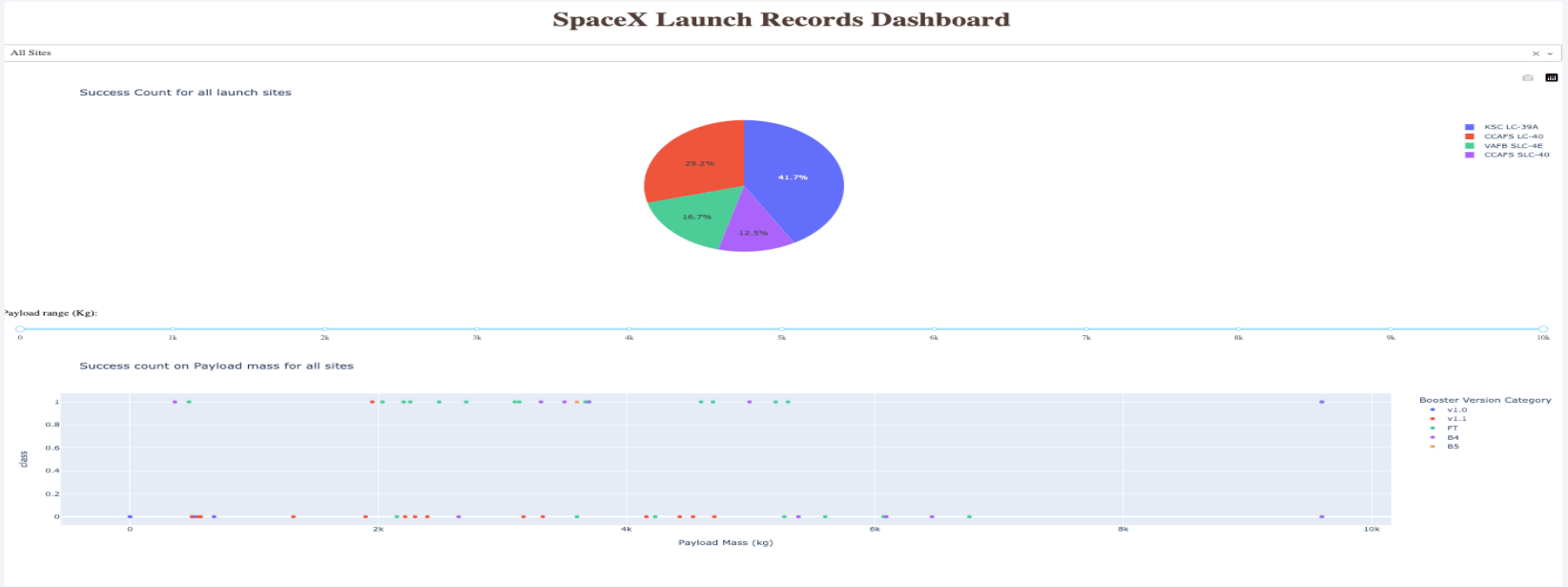
- Using SQLite created table from the CSV file and performed various exploratory data analysis
  - Getting Unique Launch Sites
  - Total Payload carried by boosters by NASA
  - Average payload by booster version F9 v1.1
  - Total number of successful and failure mission outcomes
- GitHub URL of EDA with SQL notebook - CapstoneProject/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · nkr1108/CapstoneProject (github.com)

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as circles and markers to show success and failure of launches at various sites
- Used color-labeled marker clusters for both successful and failed sites launches
- Calculated distances between a launch site to its proximities
- GitHub URL interactive map with Folium map - CapstoneProject/lab_jupyter_launch_site_location.ipynb at main · nkr1108/CapstoneProject (github.com)

Build a Dashboard with Plotly Dash

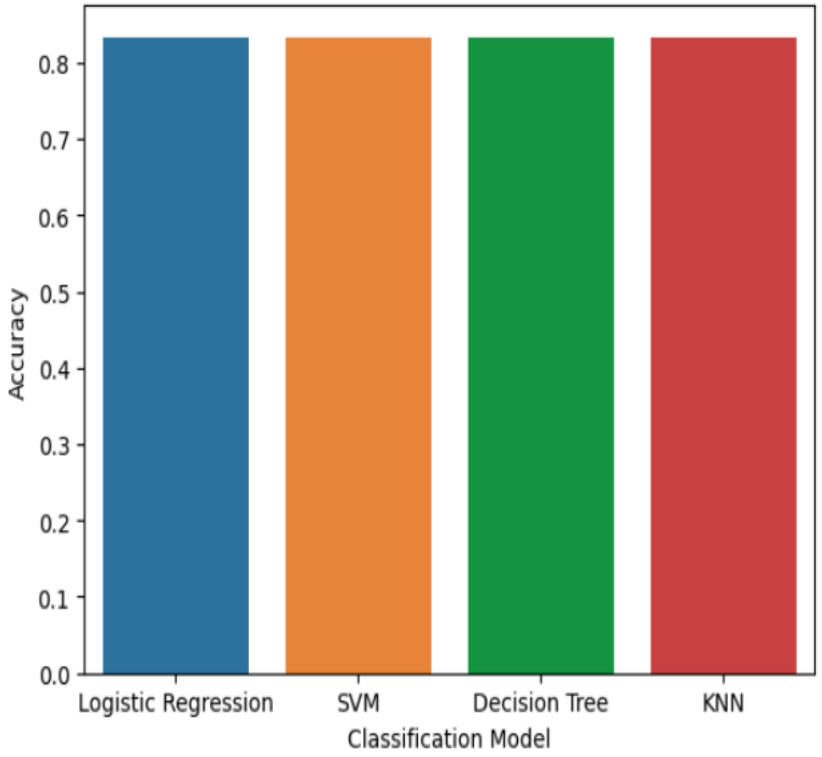
- Built interactive dashboard with plotly
- Pie chart and scatter chart was created including slider as shown in the picture



- GitHub URL Plotly Dash lab - [CapstoneProject/spacex_dash_app.py at main · nkr1108/CapstoneProject (github.com)]

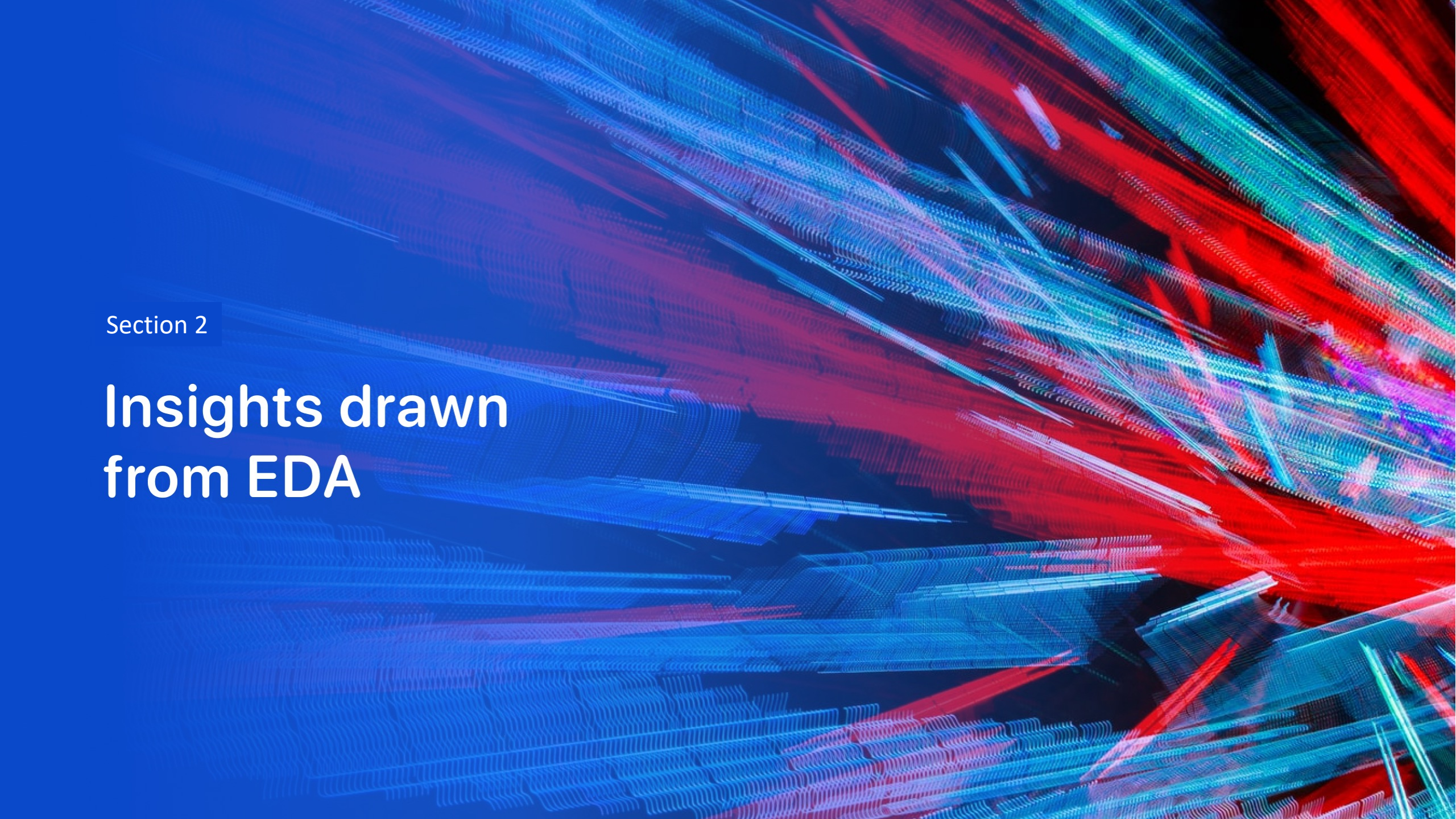
Predictive Analysis (Classification)

- Data was loaded into pandas dataframe
- Data was split into train and test and transformed using standard scalar
- Various models like Logistic Regression, SVM, Decision Tree, KNN was used to find best hyperparameter using GridSearchCV
- Accuracy was used as the performance metric
- GitHub URL predictive analysis lab - [CapstoneProject/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · nkr1108/CapstoneProject (github.com)]



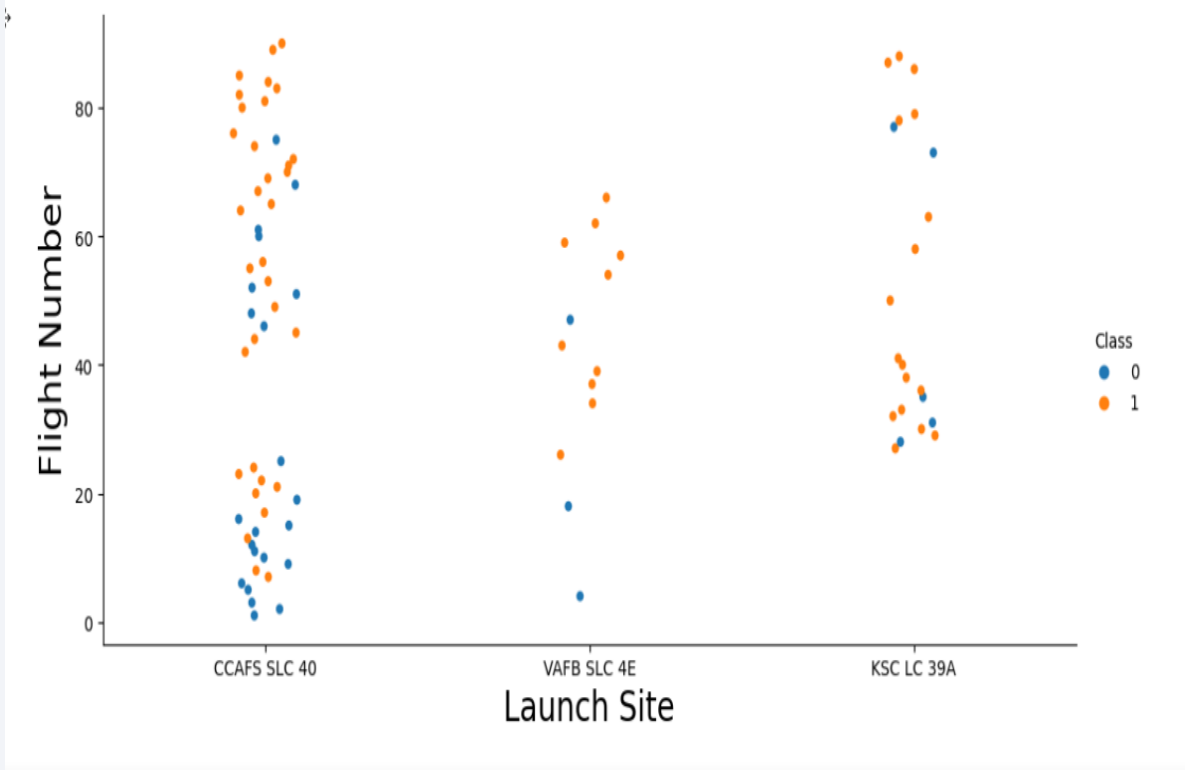
Results

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


Section 2
Insights drawn from EDA

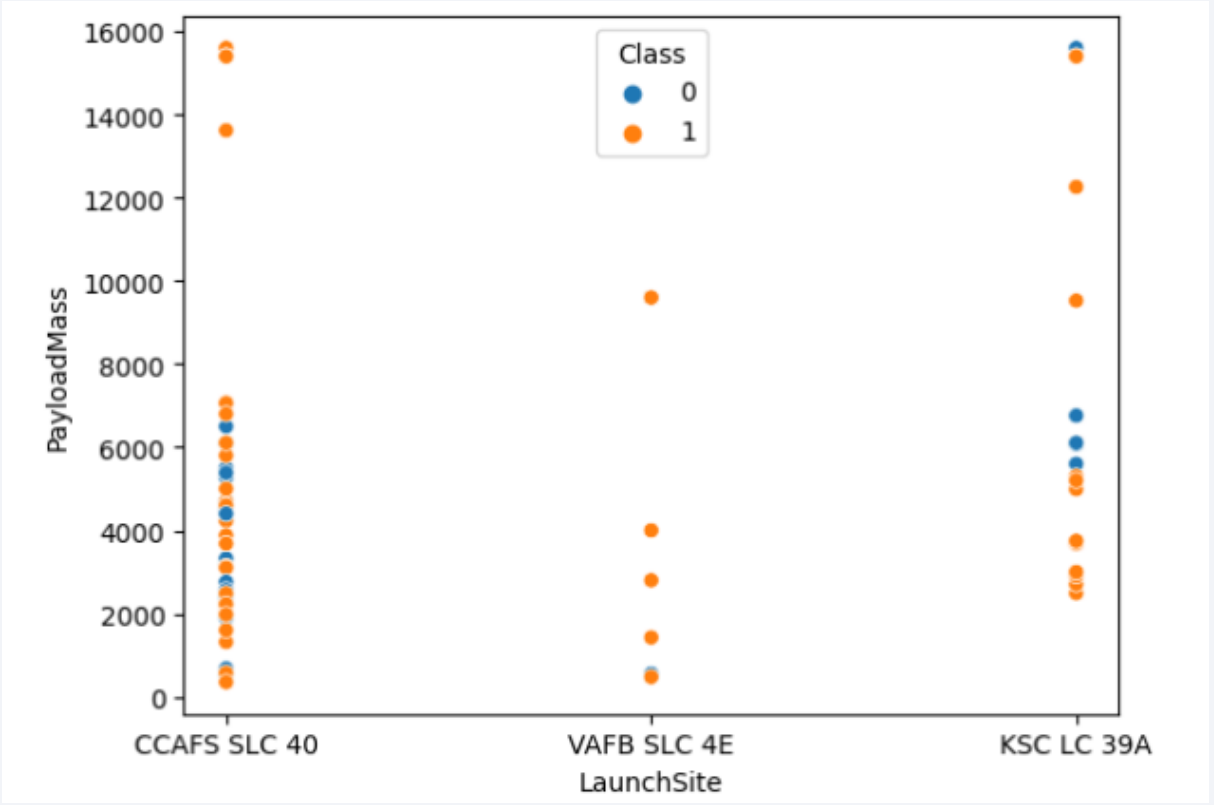
Flight Number vs. Launch Site – Scatter Plot

- Higher Flight Number have more success class
- VAFB SLC 4E have very few record of launches
- Most of the launches have been with CCAFS SLC 40 launch site
- There is no relation between Flight number and Launch Site



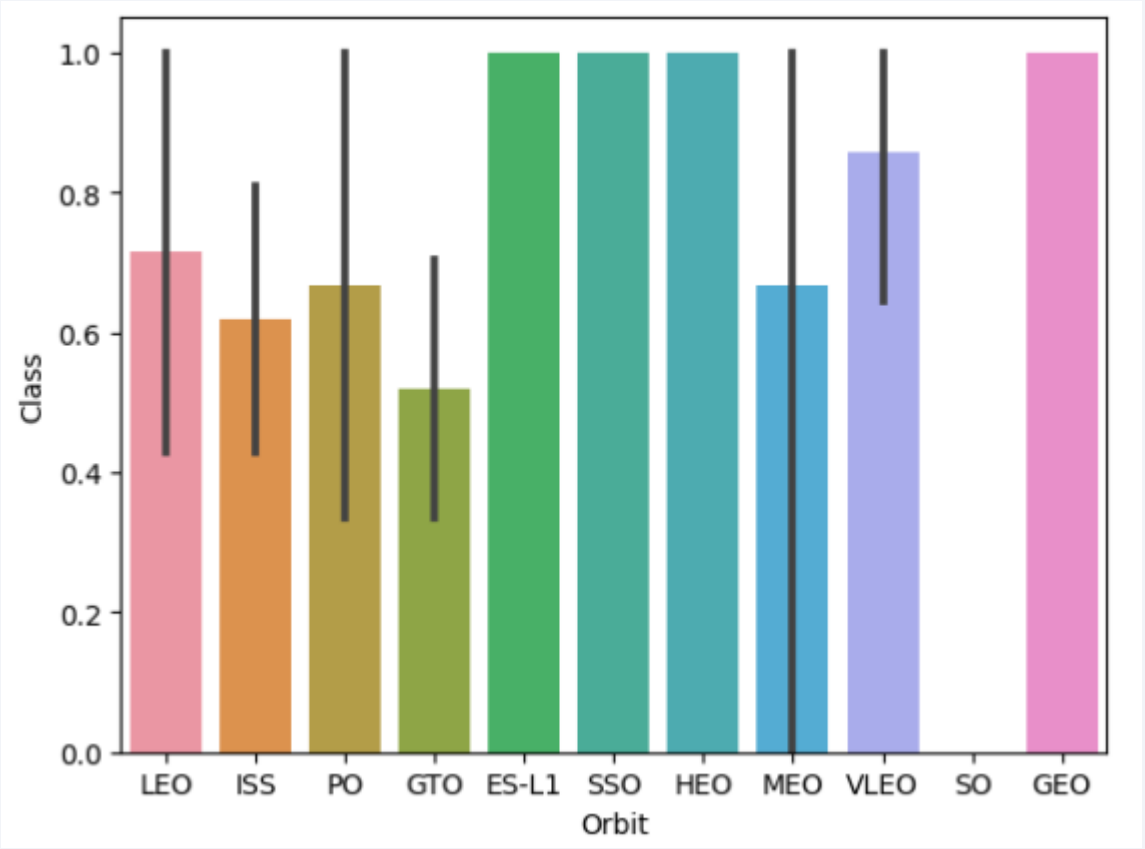
Payload vs. Launch Site

- CCAFS SLC 40 and KSC LC 39A have higher Payload mass compared to VAFB SLC 4E
- There is no relation between Payload Mass and Launch Site



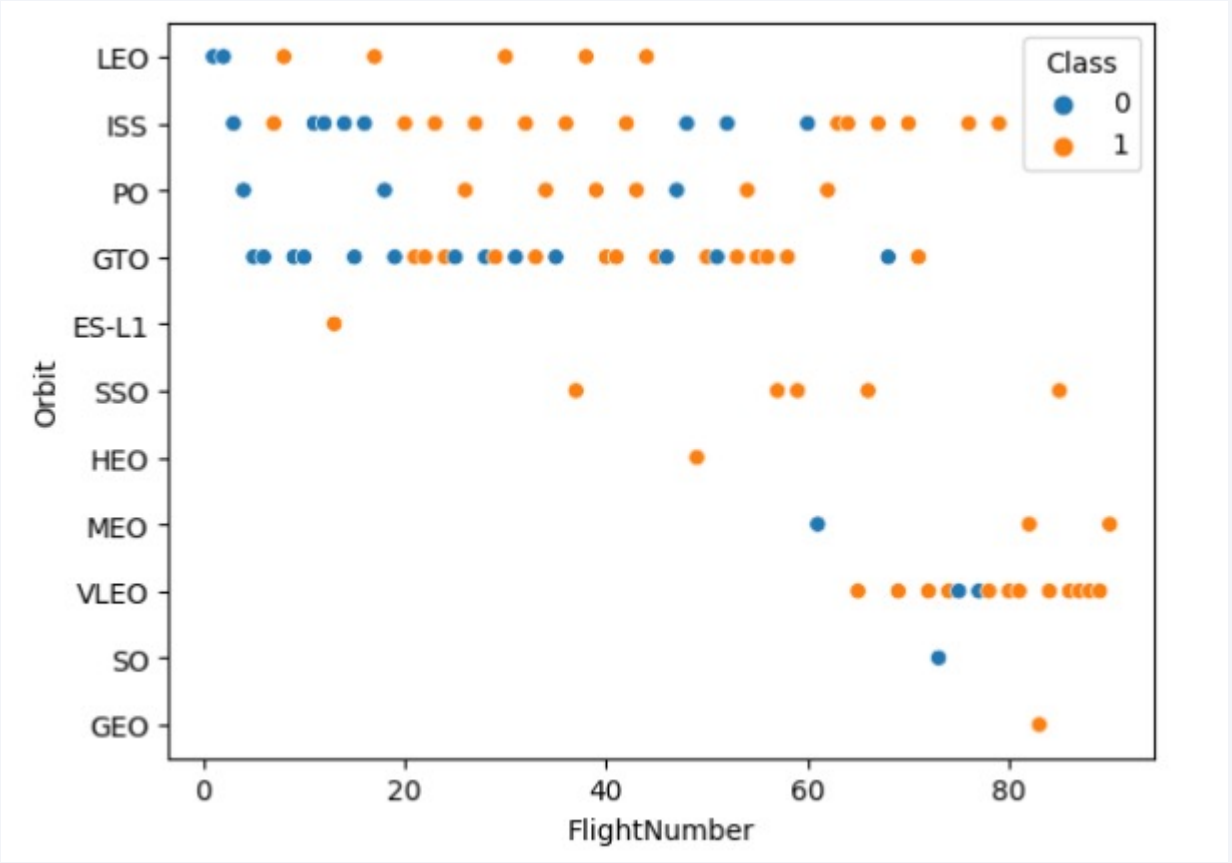
Success Rate vs. Orbit Type

- ES-L1, SSO, HEO, GEO have highest success rate
- GTO have lowest success rate
- SO do not have any data



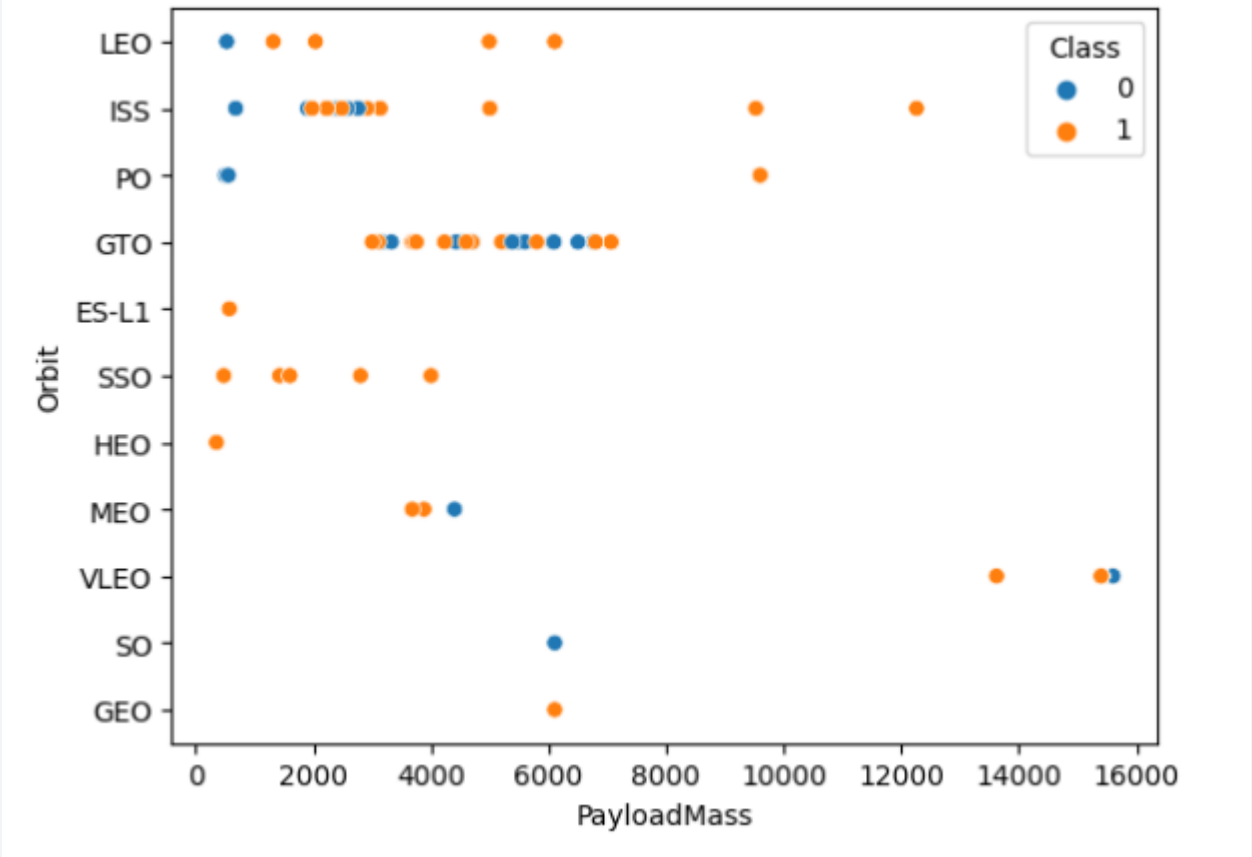
Flight Number vs. Orbit Type

- In the LEO orbit, Higher Flight number have success
- There is no such correlation in GTO orbit
- SSO and HEO do not have any failure



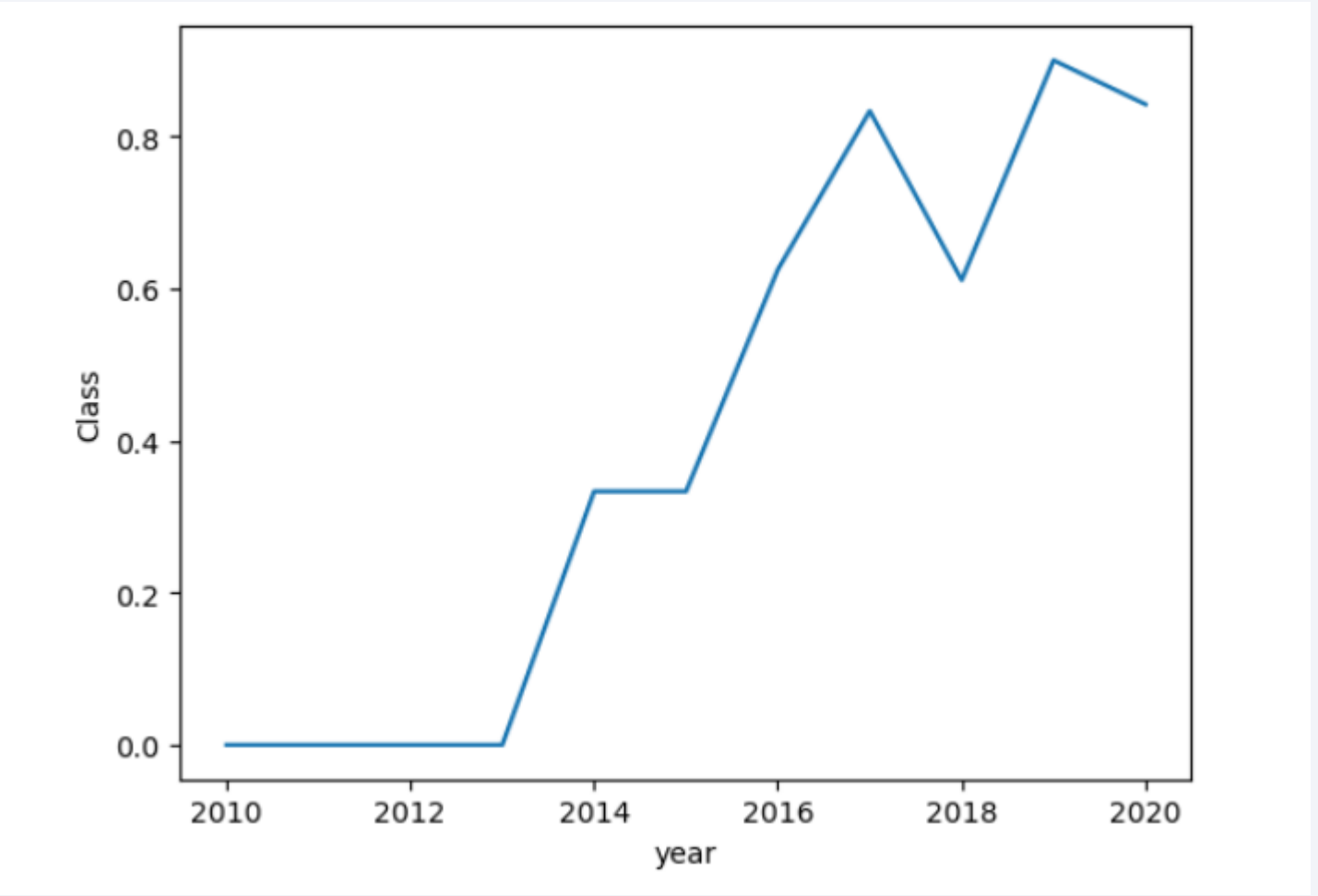
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- Success Rate after 2012 has been increasing



All Launch Site Names

- Find the names of the unique launch sites
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
- Query Results – Use of DISTINCT keyword provides unique values

```
%%sql
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%%sql
SELECT * FROM SPACEXTBL where launch_site like "CCA%" limit 5
```

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

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

- Calculate the total payload carried by boosters from NASA

```
%%sql
select sum(PAYLOAD_MASS__KG_) FROM spacextbl where Customer = "NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
sum(PAYLOAD_MASS__KG_)
45596
```

Average Payload Mass by F9 v1.1

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

```
%%sql
select avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = "F9 v1.1"
```

```
 * sqlite:///my_data1.db
Done.
avg(PAYLOAD_MASS__KG_)
2928.4
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql select min(date) from spacextbl where "Landing _Outcome" like "%ground%"
```

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

min(date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
[ ] %%sql
    select Booster_Version from spacextbl where Mission_Outcome = "Success" and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%%sql
select count(*) from spacextbl where Mission_Outcome like "%Success%"
```

```
 * sqlite:///my_data1.db
Done.
count(*)
100
```

```
[ ] %%sql
select count(*) from spacextbl where Mission_Outcome like "%Failure%"
```

```
 * sqlite:///my_data1.db
Done.
count(*)
1
```

Boosters Carried Maximum Payload

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

```
%%sql
select Booster_Version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

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

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

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
[16] %sql SELECT substr(DATE,4,2),"Landing _Outcome", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE substr(Date,7,4) = "2015" and "Landing _Outcome" like "%failure%"
```

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

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

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

+ Code

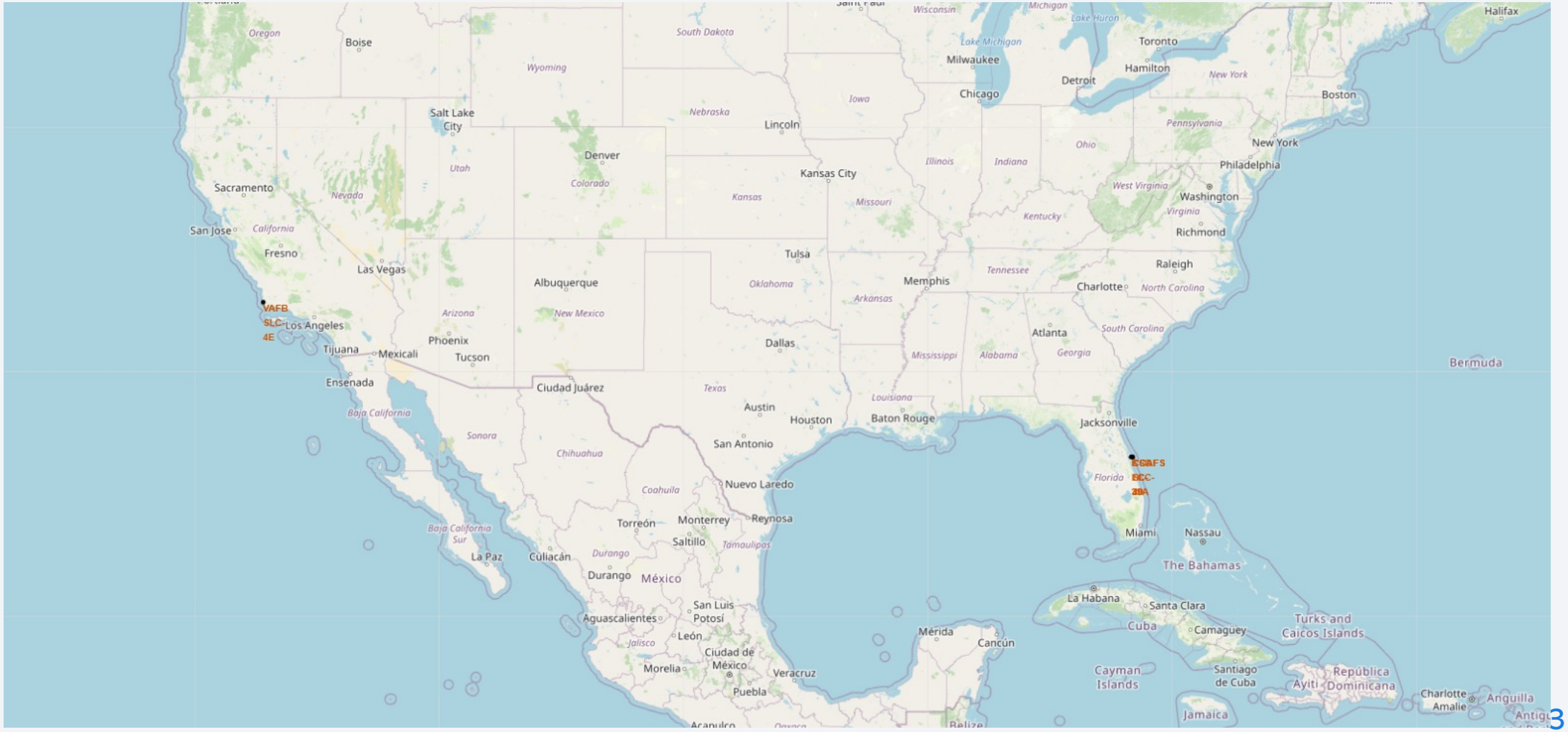
```
%%sql
select "Landing _Outcome", count(*) from spacextbl where date between '04-06-2010' and '20-03-2017'
```

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

Landing _Outcome	count(*)
Failure (parachute)	57

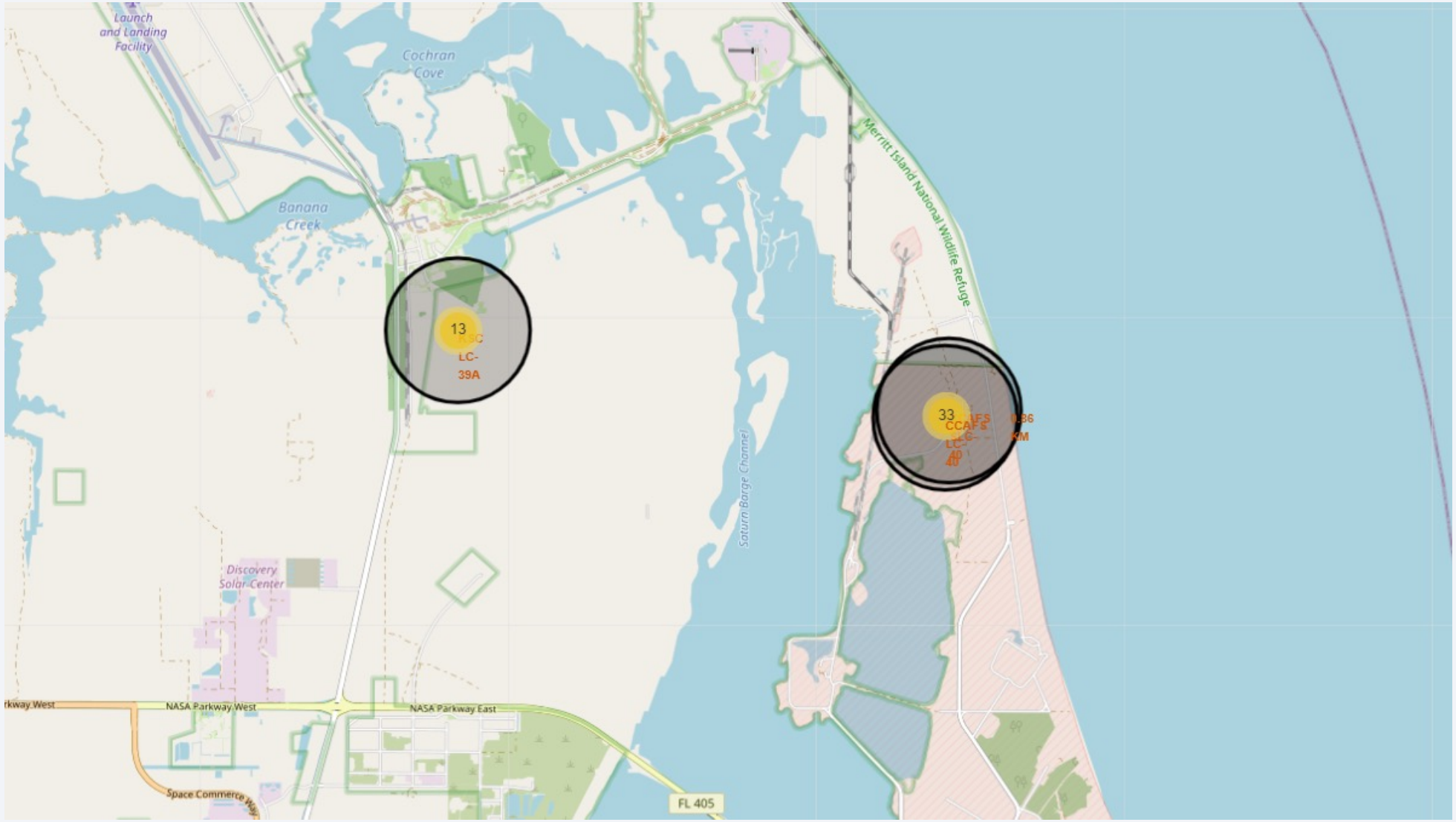

Section 3
Launch Sites
Proximities Analysis

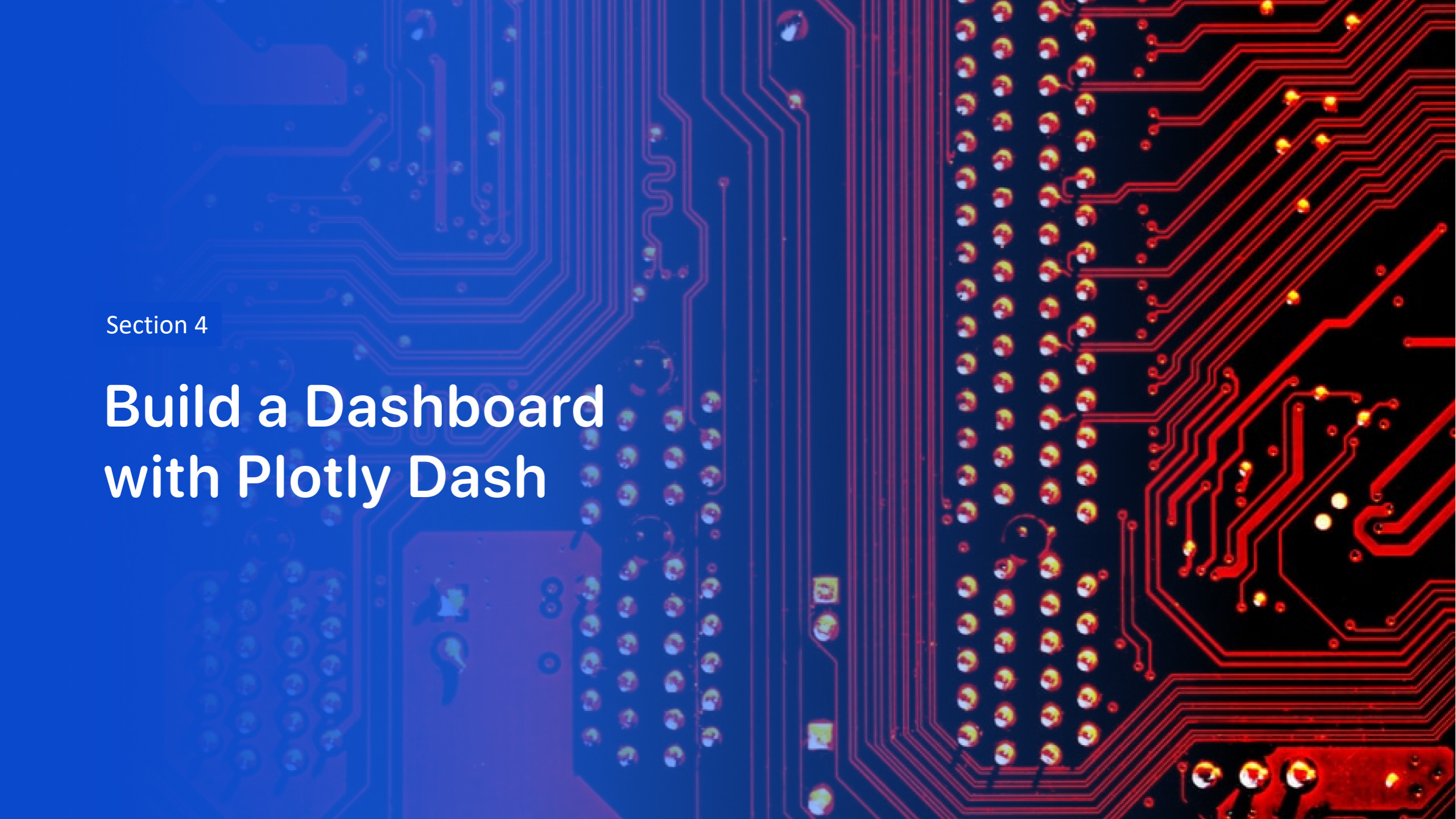
All Launch Sites



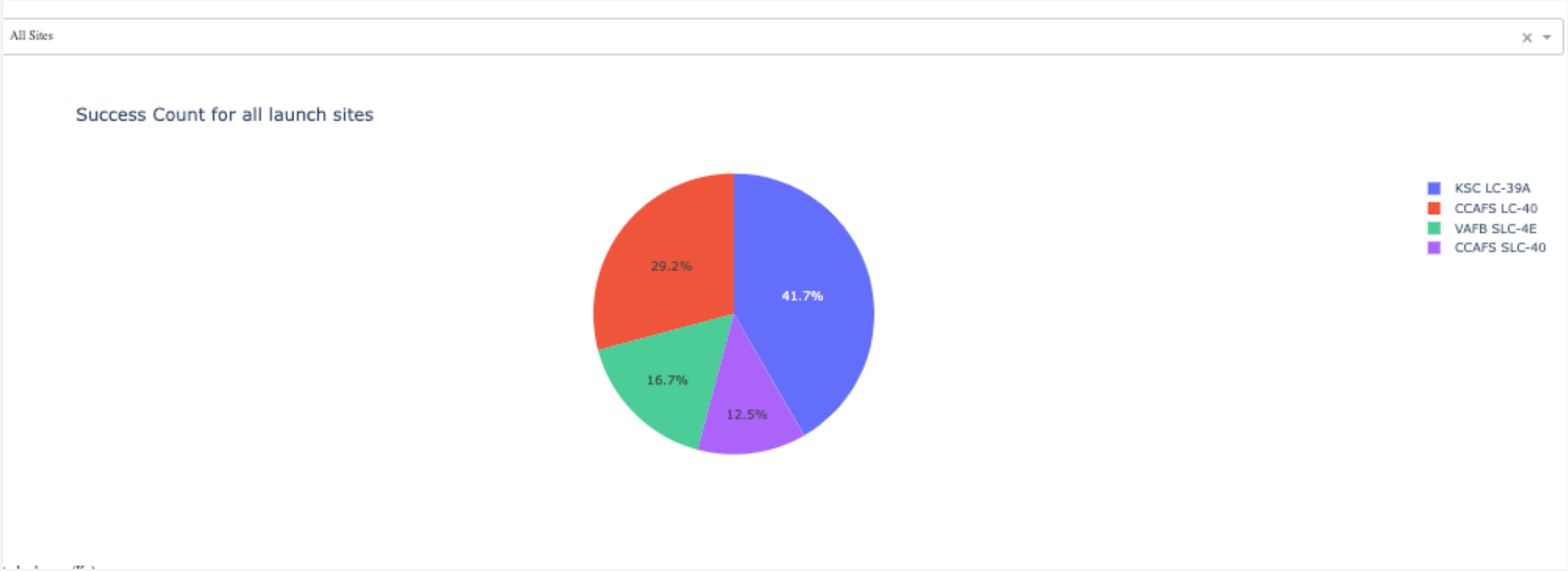
Success/Failure Map

Launch Sites

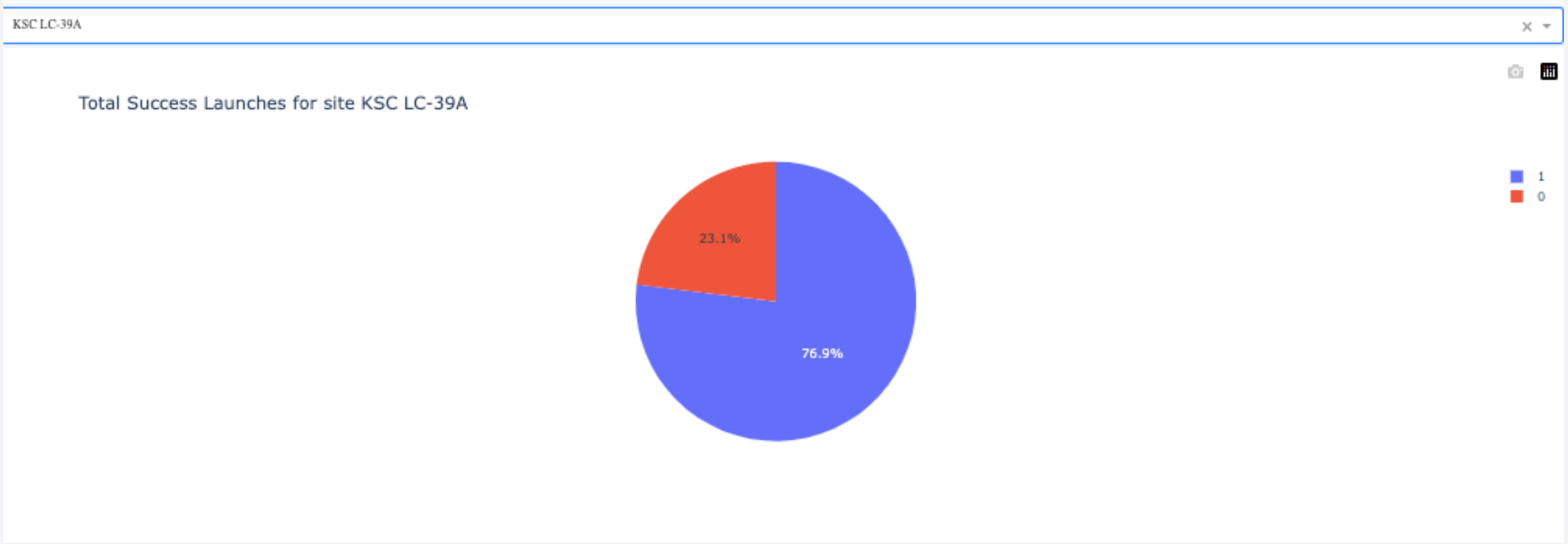



Section 4
Build a Dashboard with Plotly Dash

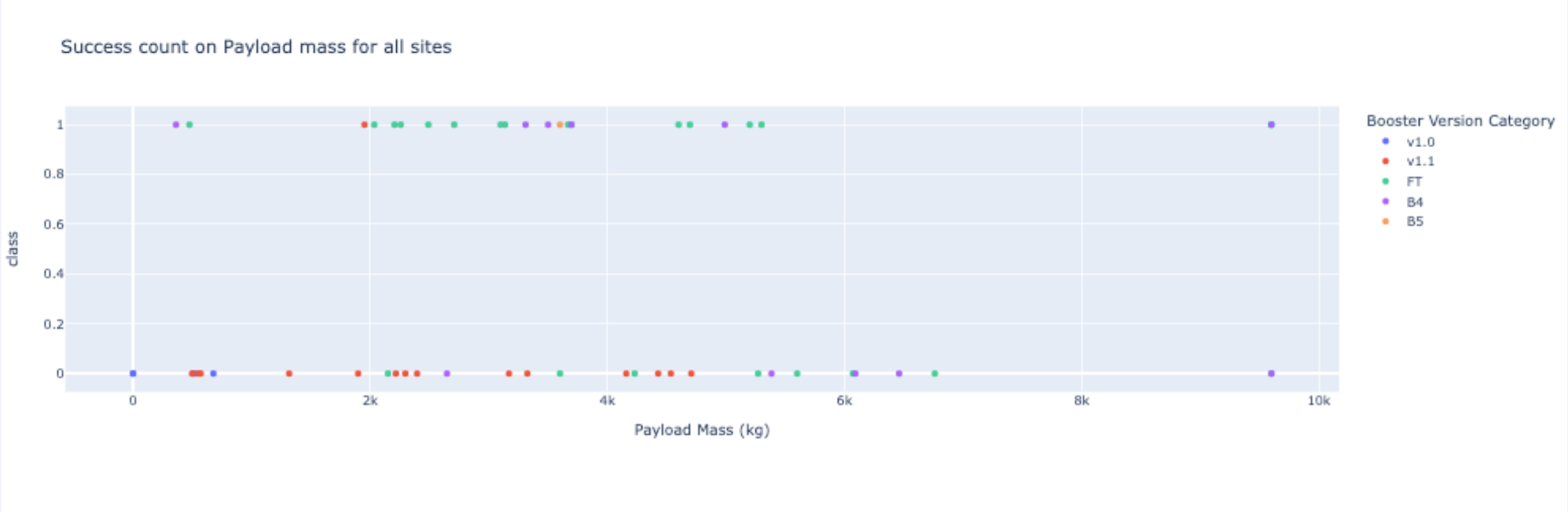
Success Count for all Launch Sites – Pie chart



Pie Chart for Highest Launch Success Ratio



Payload vs Launch Outcome



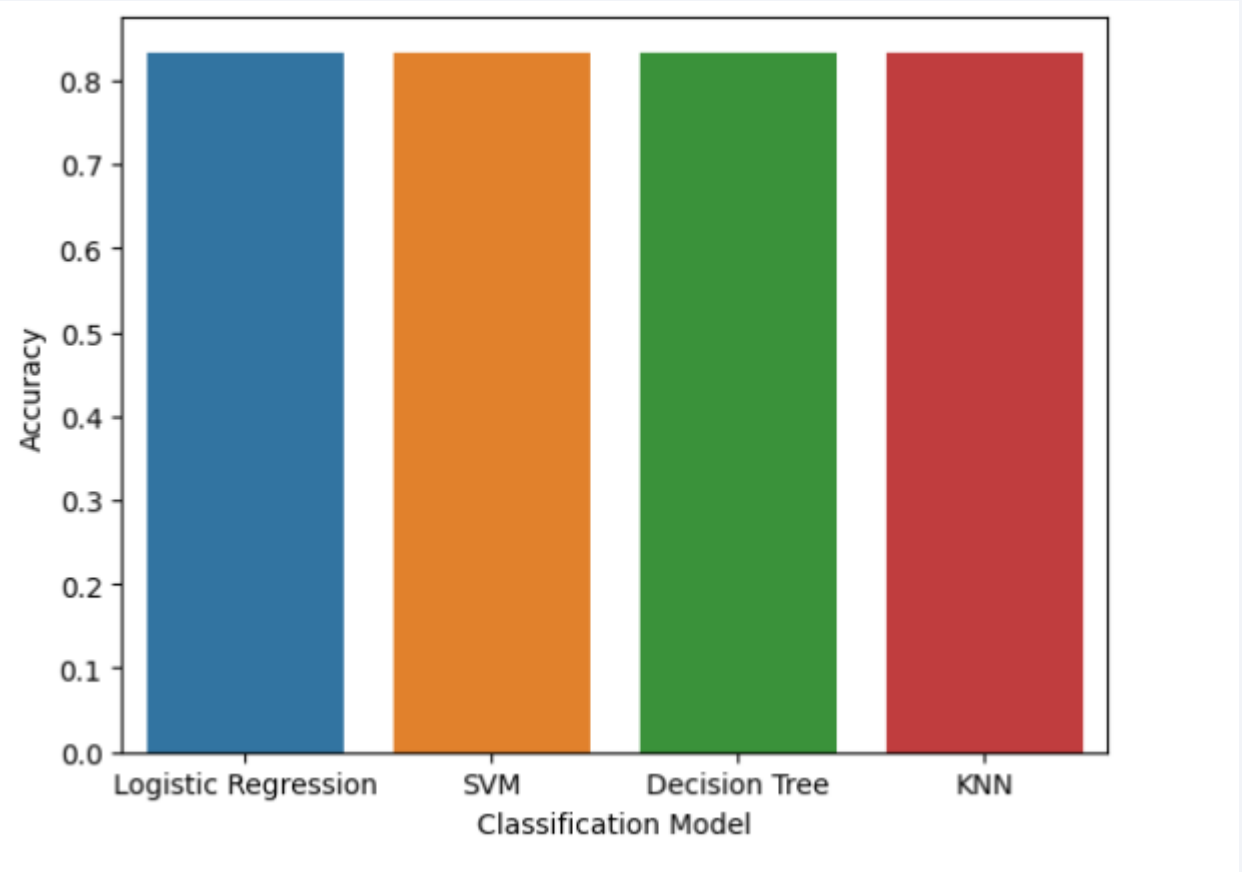
Booster FT has highest success

Section 5

Predictive Analysis (Classification)

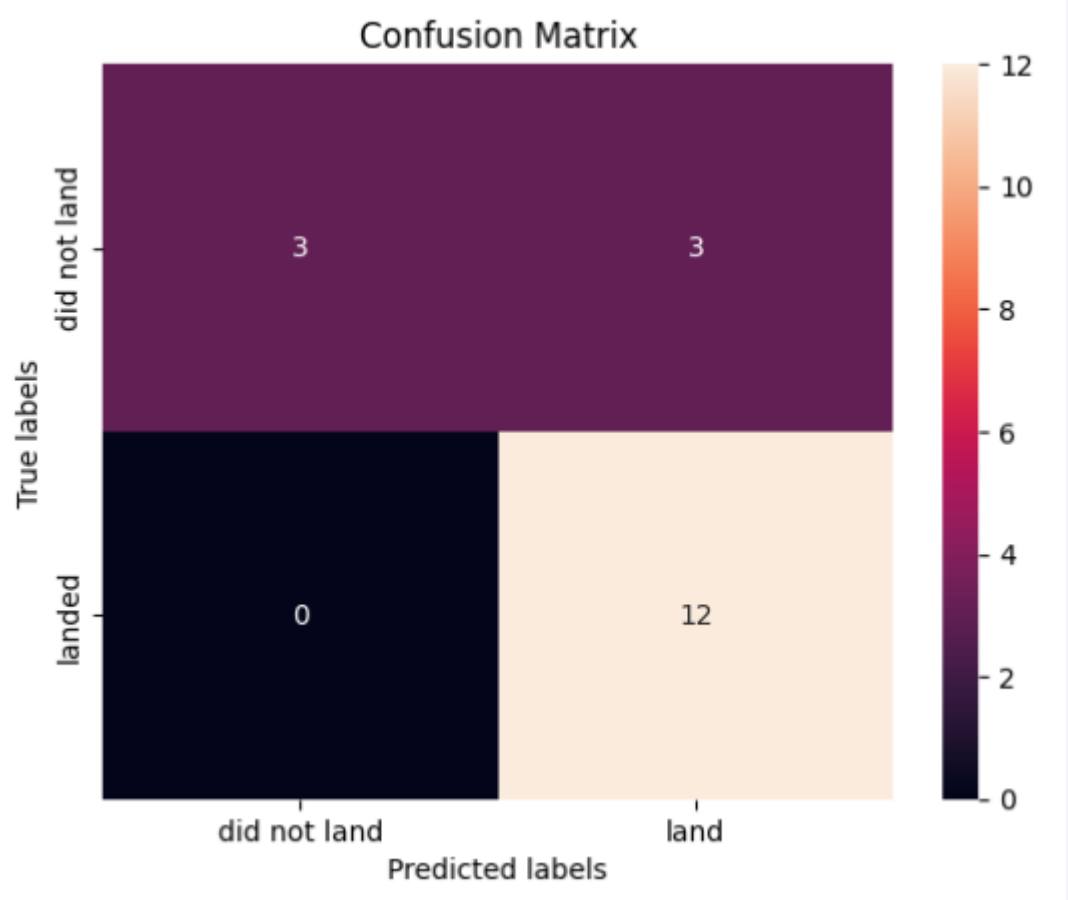
Classification Accuracy

- Given the data we have, all the classification model came with same score as shown in the barchart



Confusion Matrix

- Based on the given data and best model
  - Model has 80% recall score
  - True Positive is 12
  - True Negative is 3



Conclusions

- Model is able to predict the landing of rocket with 83% accuracy
- ES-L1, SSO, HEO, GEO have highest success rate
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- Success Rate after 2012 has been increasing

Appendix

- All the relevant notebooks, different csv files generated and analysis powerpoint is saved under [nkr1108/CapstoneProject: IBM Data Science - Capstone Project (github.com)]

Thank you!