IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ranjeet Singh
14/10/2024

Outline

Executive Summary

- Summary of methodologies
  - Data Collection using API and Web Scrapping
  - Data Wrangling
  - EDA using SQL & Pandas)
  - Data Visualization
  - Interactive Geo Maps using Folium
  - Dynamic Dashboard using Plotly
  - Model Building, Hyperparameter tuning, & Model Evaluation
  - Finding Best Model
- Summary of all results
  - Results of EDA, Folium, Plotly
  - Results of Predictive Modeling

Introduction

- Project background and context
  - SpaceX offers Falcon 9 rocket launches for $62 million, compared to over $165 million from other providers, thanks to its reusable first stage. Predicting whether the first stage will land helps estimate launch costs, which is valuable for competitors bidding against SpaceX. Here, we aim to predict the success of Falcon 9's first stage landing.
- Problems you want to find answers
  - Identify the key factors contributing to a successful landing
  - How successful landing depend on individual input features
  - Predict optimal input features for successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through SpaceX API and Web Scrapping from Wikipedia
- Perform data wrangling
  - Handling Missing Value, One Hot Encoding for categorical features
- 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

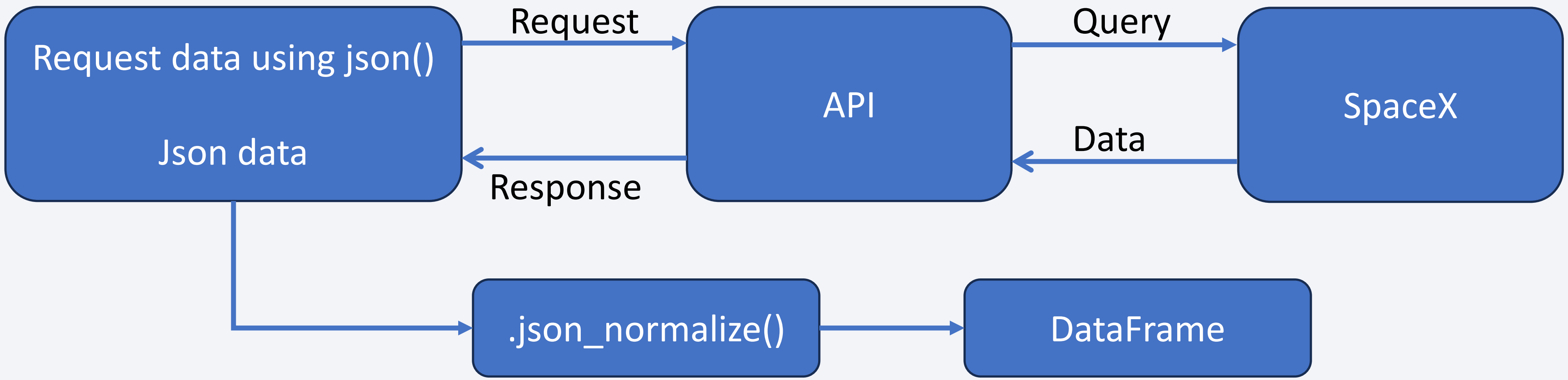
- Data was collected in 2 ways.

- **SpaceX API:** We sent request through API . We received the response as a JSON using .json() function and further we converted that to a Pandas Dataframe with the help of .json_normalize()

- **Web Scrapping:** We used BeatifulSoup Library to scrape data from Wikipedia. We received the HTML response and after parsing the response , we converted that into Pandas Dataframe. In the next slides , we will see in details.

Data Collection – SpaceX API

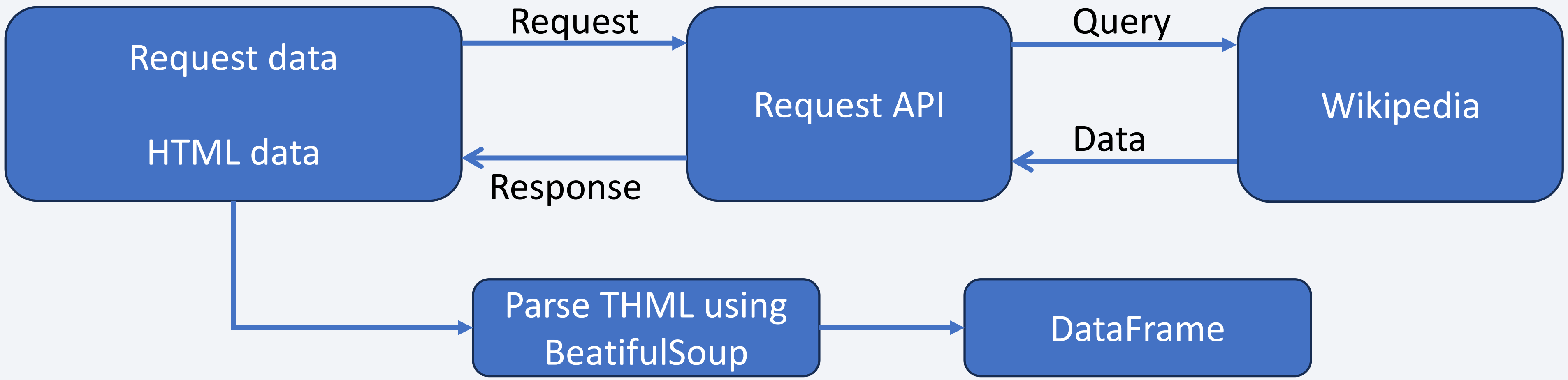
- Data collection using SpaceX API



- GitHub Notebook: SpaceX API Data Collection

Data Collection - Scraping

- Process involving data collection using web scrapping



- GitHub URL Web Scraping Notebook

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

EDA with Data Visualization

- We analyzed the relationships between various features, such as flight number, payload mass, launch site, succuss rate by orbit type, and year trend of successful launches.
- Charts:
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload Mass vs Orbit type

- Jupyter Notebook: EDA Data Visulization

EDA with SQL

- We used SQL for EDA to uncover key insights, including:
    - List of unique launch sites
    - Total payload carried by boosters
    - Number of successful and failed launch
    - And so on …

    - Jupyter Notebook: [EDA using SQL]

Build an Interactive Map with Folium

- We marked:
- All launch sites using marker and circles
- Marked successful and failure events at each sites.
- Distance between the launch sites and the coastline/railroad/cities.

- Jupyter Notebook: [Folium Interactive Map]

Build a Dashboard with Plotly Dash

- Dropdown list for Launch Site selection
  - User can choose all sites or a specific launch site
- Pie Chart for launch success or failure
- Scatter chart for relation between Success and Payload Mass
- Added Slider for Payload Mass Range
  - User can choose variable payload mass range

- GitHub URL:Plotly Dashboard

Predictive Analysis (Classification)

- The process involves following steps:

Data Standardize
- StandardScalar

→

Train-Test Split
- Training data size= 80%

→

Model Building
- LogisticRegression
- Support Vector Classification
- Decision Tree
- KNeighborsClassifier

→

Model Optimization
- GridSearchCV

→

Model Evaluation
- Accuracy Score
- Confusion Matrix

→

Best Model

- Jupyter Notebook: [Machine Learning Prediction]

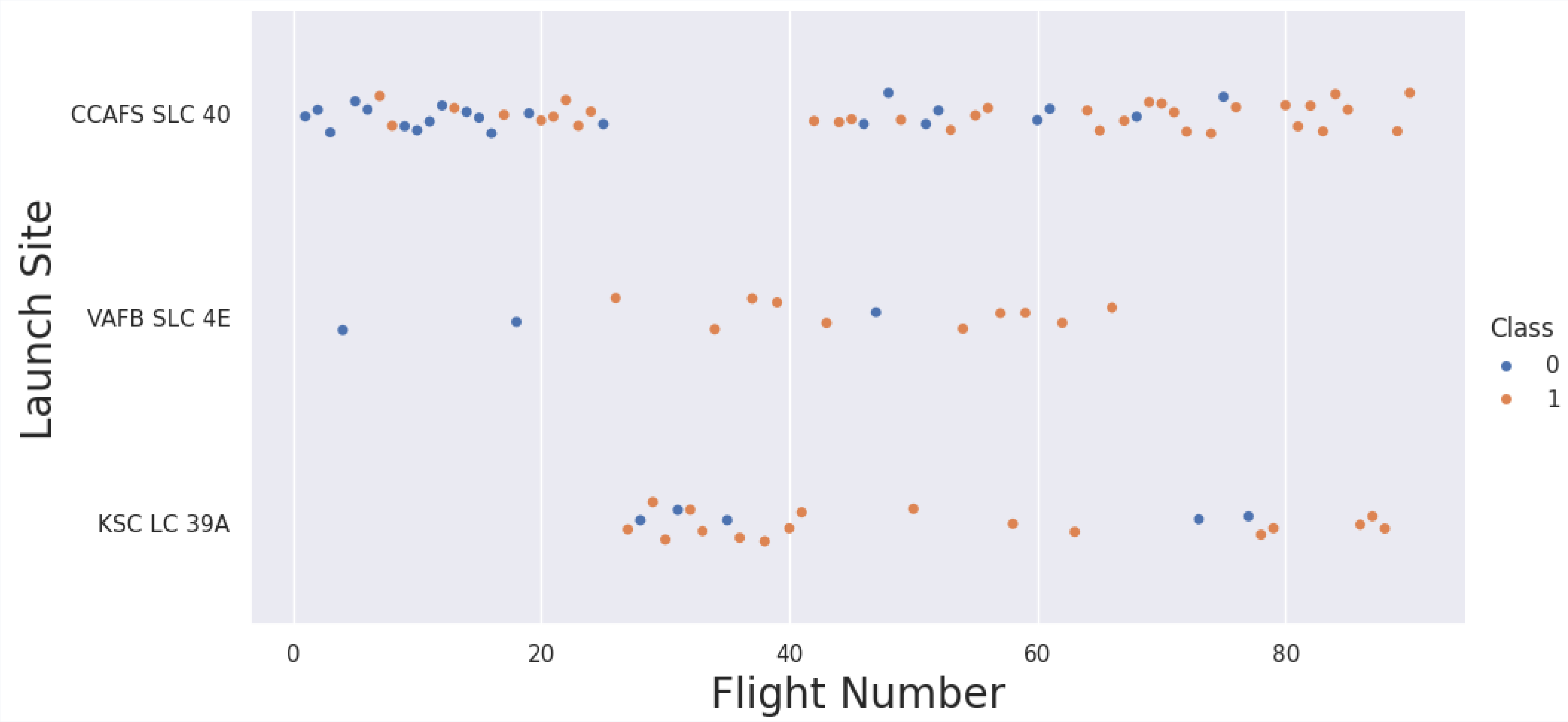
Results

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

Section 2

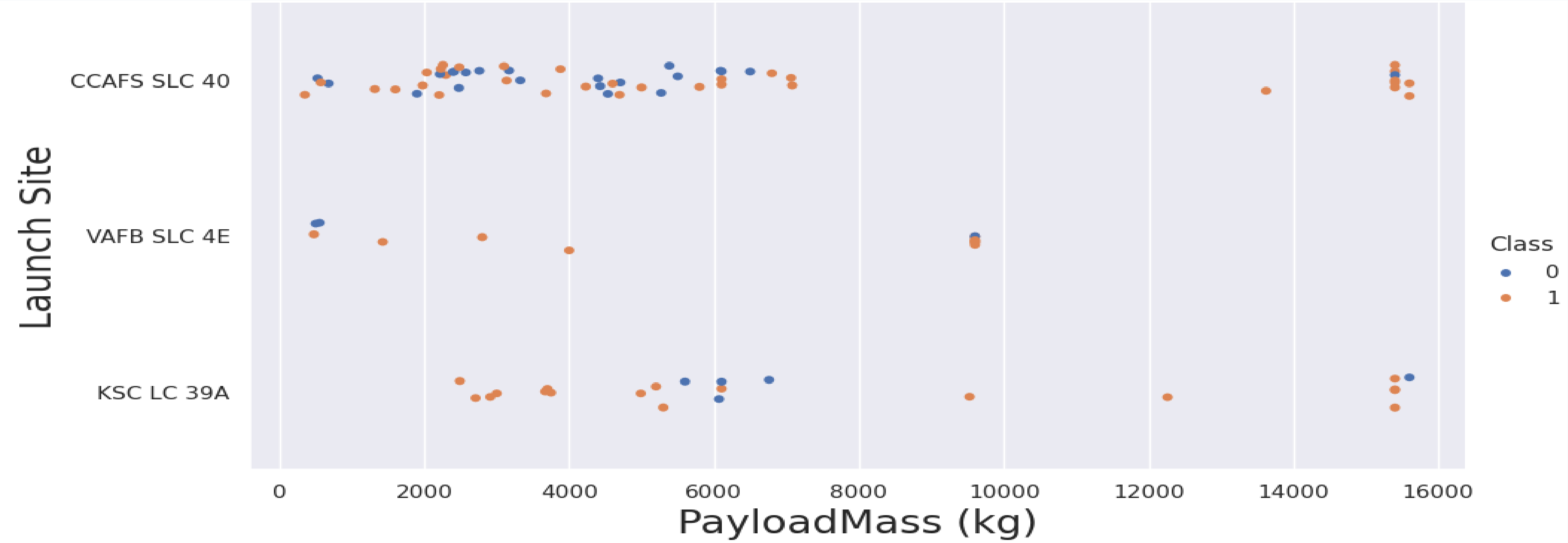
Insights drawn from EDA

Flight Number vs. Launch Site



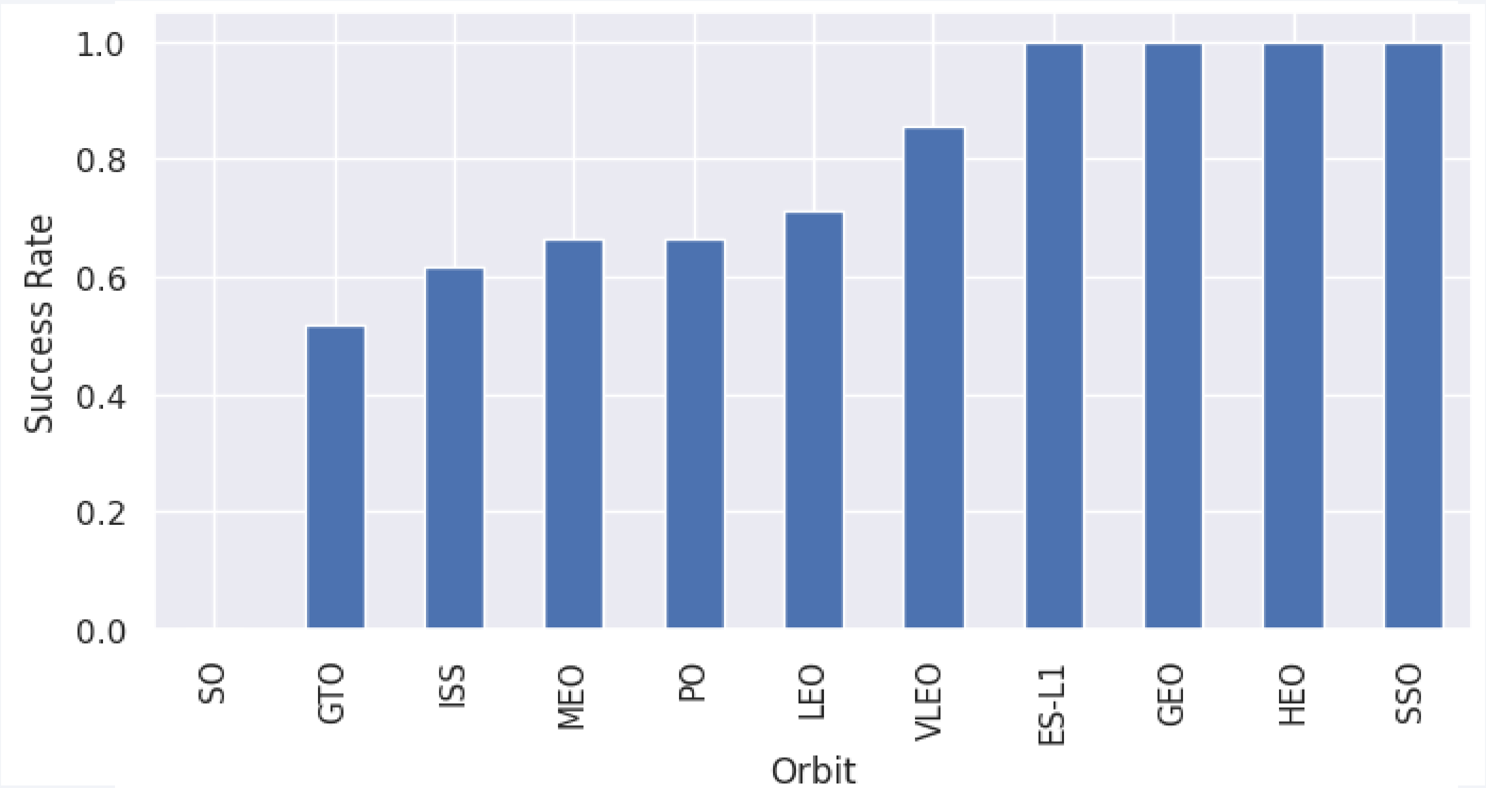
- Initially success rate were low, with increase in more launch, success rate increased significantly.

Payload vs. Launch Site



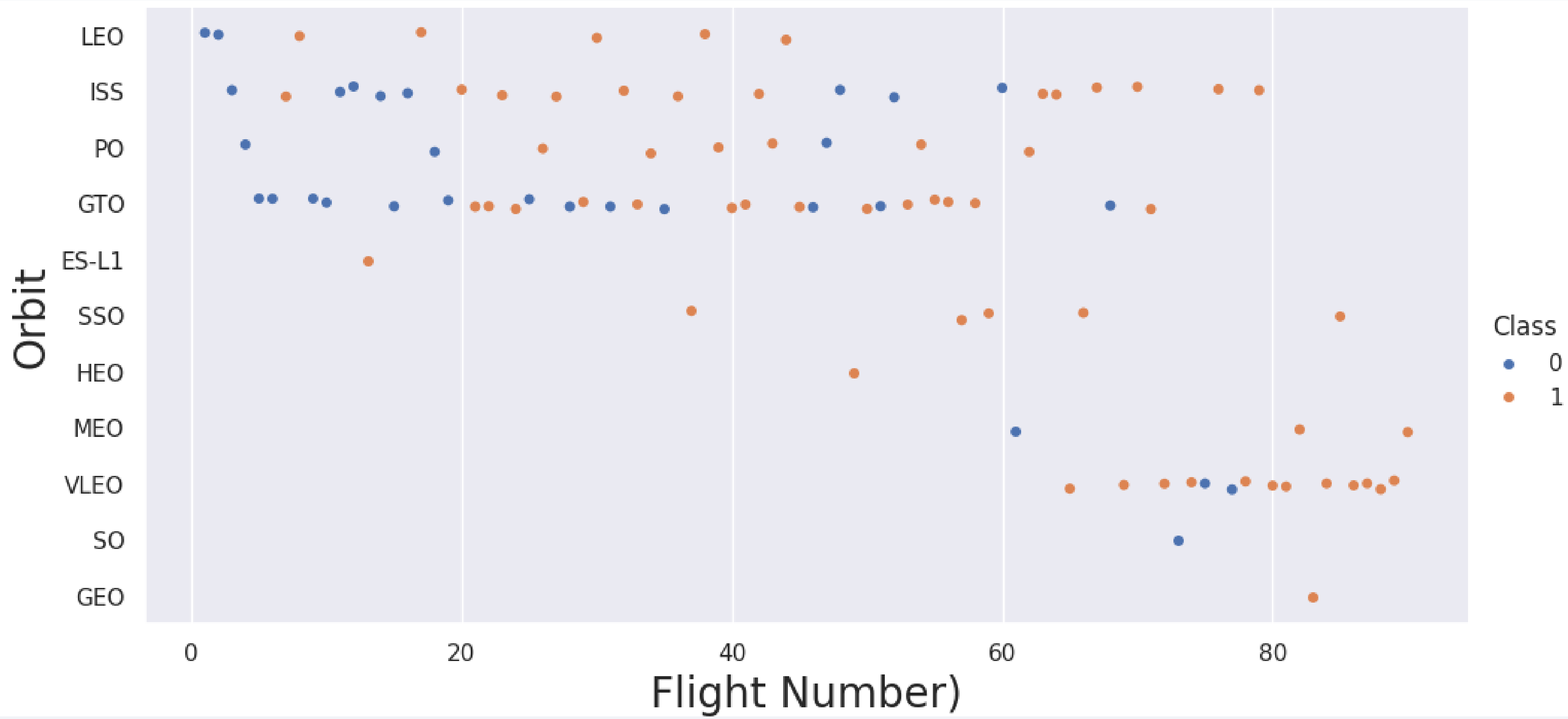
- CCAFS SLC 40 has high succuss with higher payload mass

Success Rate vs. Orbit Type



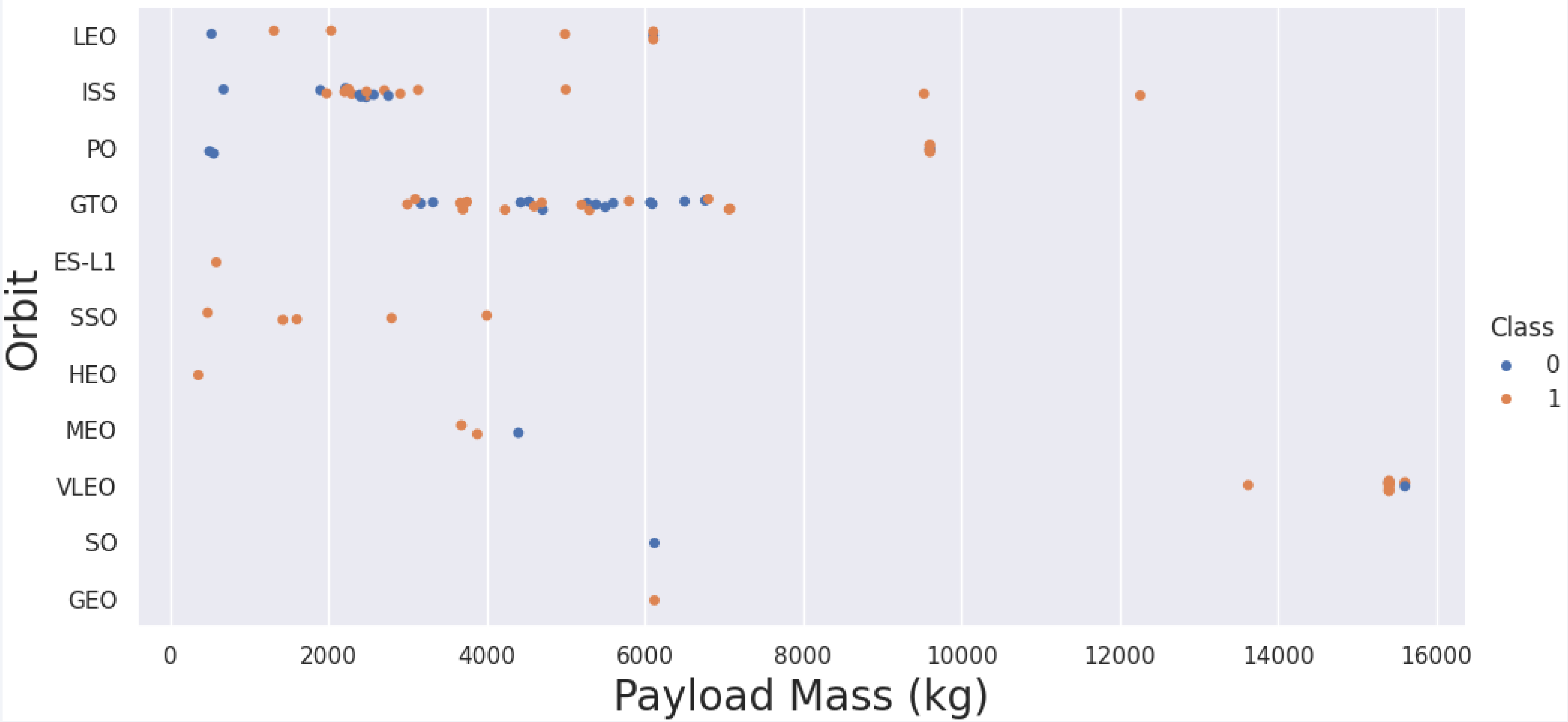
- Orbit ES-L1, GEO, HEO and SSO has highest success rate

Flight Number vs. Orbit Type



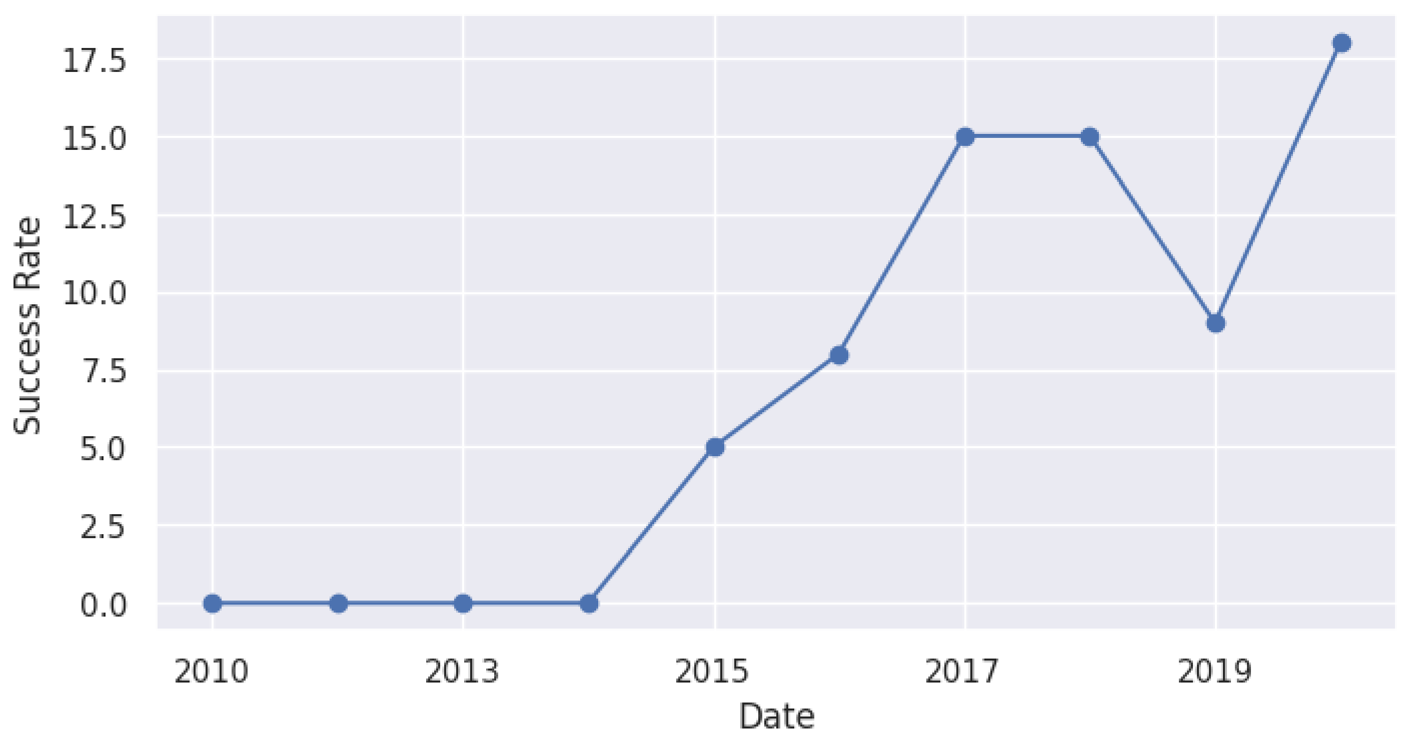
- Most of the events result into failure for lower flight number (< 20)

Payload vs. Orbit Type



- Orbit LEO,ISS and PO has better succuss with higher payload mass.

Launch Success Yearly Trend



- After,2014, there is a sudden spike in success rate

All Launch Site Names

- Four Launch Sites:

```
[11]: %sql select distinct Launch_Site from SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
```

[11]:

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

- Used distinct keyword to find unique Launch Site

Launch Site Names Begin with 'CCA'

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

```
[13]: %sql select * from SPACEXTABLE where Launch_Site like "CCA%" limit 5

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

[13]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	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

```
[13]: %sql select sum(PAYLOAD_MASS__KG_) from  SPACEXTABLE where Customer = "NASA (CRS)"

 * sqlite:///my_data1.db
Done.

[13]: sum(PAYLOAD_MASS__KG_)
                      45596
```

- Used sum() aggregate function for total payload mass and where clause to filter data for NASA only
- Result: 45596Kg

Average Payload Mass by F9 v1.1

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

```
[15]: %sql select avg(PAYLOAD_MASS__KG_) as Average_Payload_Mass from  SPACEXTABLE where Booster_Version like "F9 v1.1"

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

[15]:

Average_Payload_Mass
2928.4

- Used avg() aggregate function for average payload mass and where clause to filter data for booster **F9 v1.1** only
- Result: 2928.4

First Successful Ground Landing Date

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

```
[16]: %sql select min(Date) from SPACEXTABLE  where Landing_Outcome = 'Success (ground pad)'

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

[16]:

min(Date)
2015-12-22

- Used min() function for Date and where clause to filter.
- Result: 2015-12-22

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

```
[18]: %sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and \
          PAYLOAD_MASS__KG_ between 4000 and 6000

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

[18]:

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

- All booster versions are type of F9 FT B10XXXX

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
[20]: %sql select Mission_Outcome, count(Date) as Total from SPACEXTABLE group by Mission_Outcome

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

[20]:

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

- Total Success: 100
- Only 1 Failure

Boosters Carried Maximum Payload

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

```
[21]: %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

[21]:

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

- All booster versions are type of F9 FT B5 B10XX.X

2015 Launch Records

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

```
[23]: %sql select substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site  from SPACEXTABLE where \
          substr(Date,0,5)='2015' and Landing_Outcome = "Failure (drone ship)"

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

[23]:

Month	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

- Both the failure events were at the CCAFS LC-40 site.

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

```
[31]: %%sql select Landing_Outcome, count(*) as Total from SPACEXTABLE
          where Date between '2010-06-04' and '2017-03-20'
          group by Landing_Outcome
          order by Total desc

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

[31]:

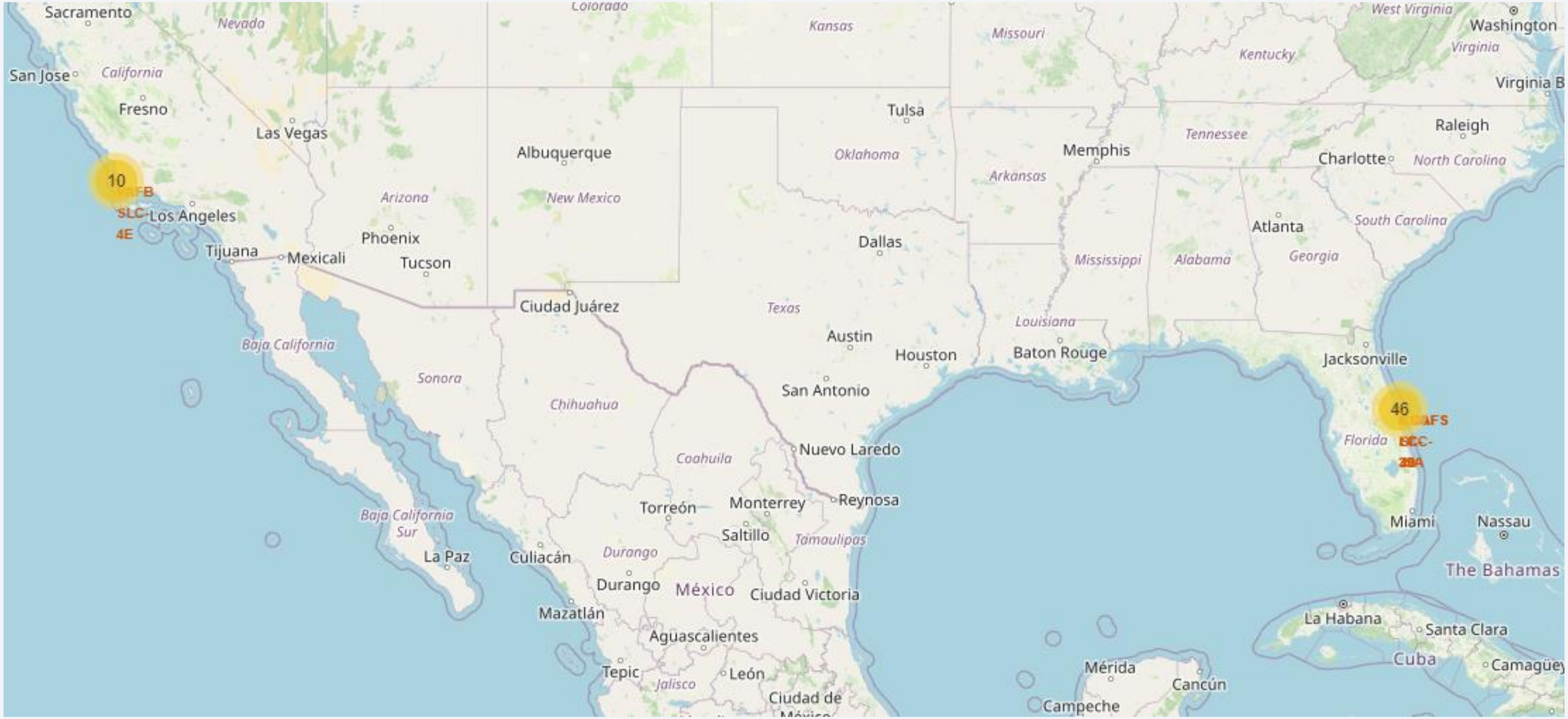
Landing_Outcome	Total
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- No attempt were made for 10 events.

Section 3

Launch Sites Proximities Analysis

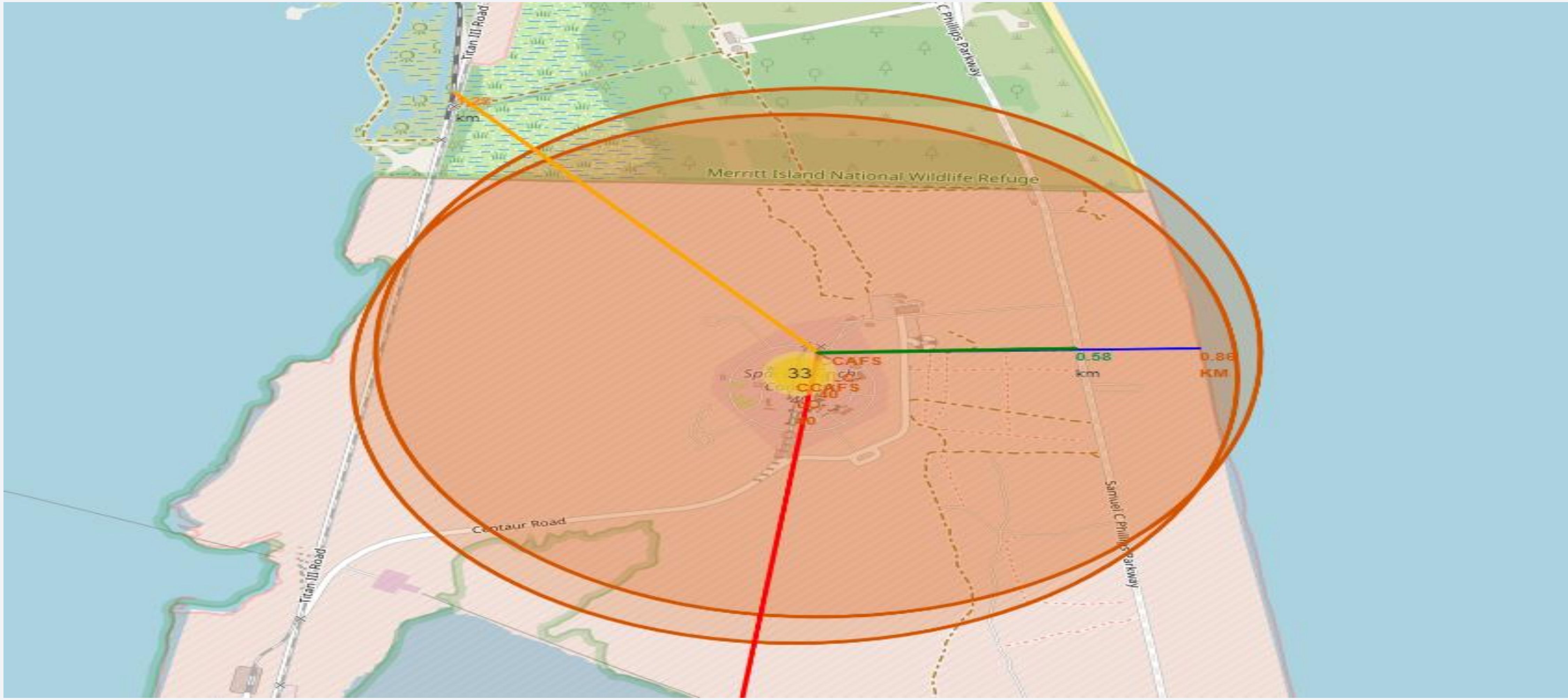
Launch Sites



Successes and Failures Launch Sites

- We can see success/Failures for individual launch sites
- Green Marked are Successes and Reds are Failures

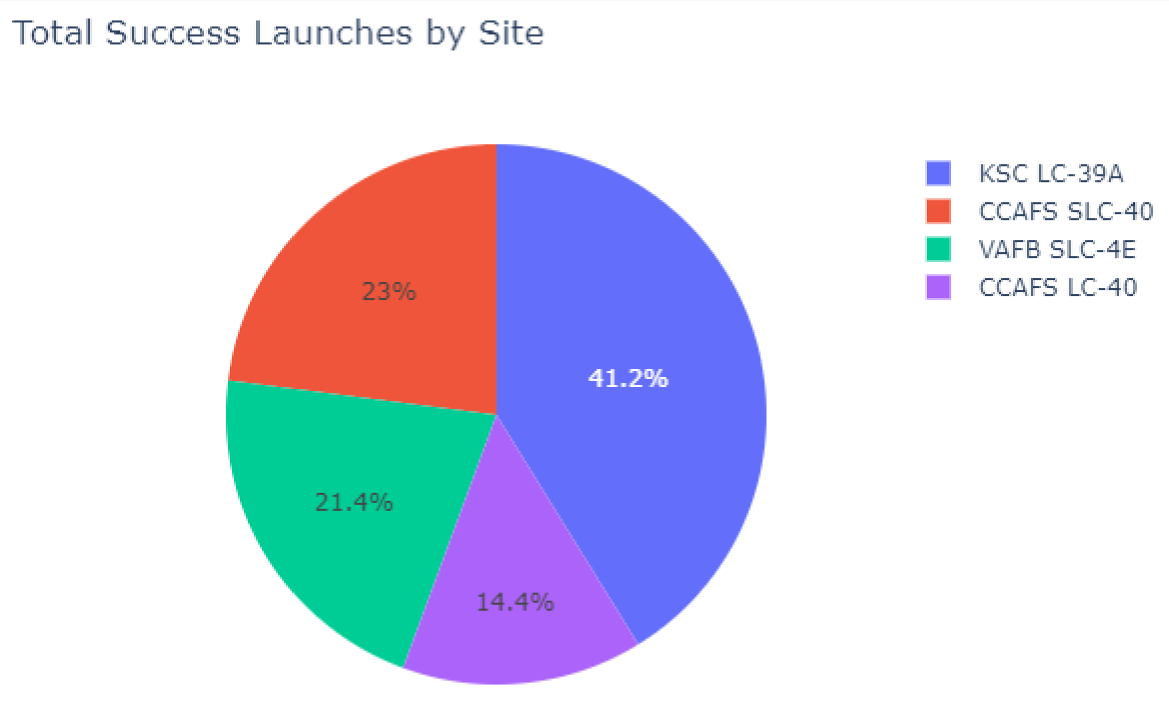
Distance between Launch site and its proximities



Section 4

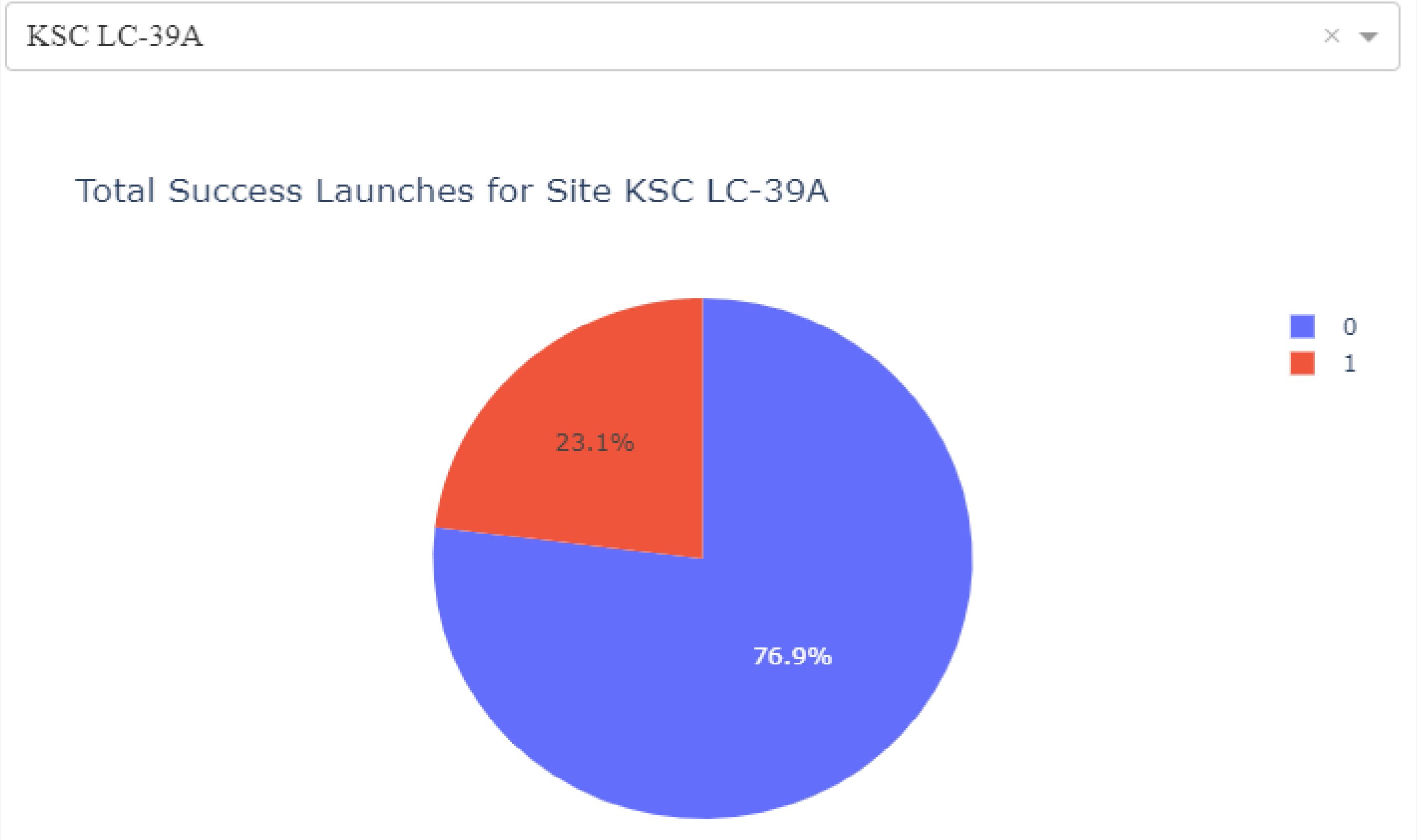
Build a Dashboard with Plotly Dash

Success rate by Launch Site



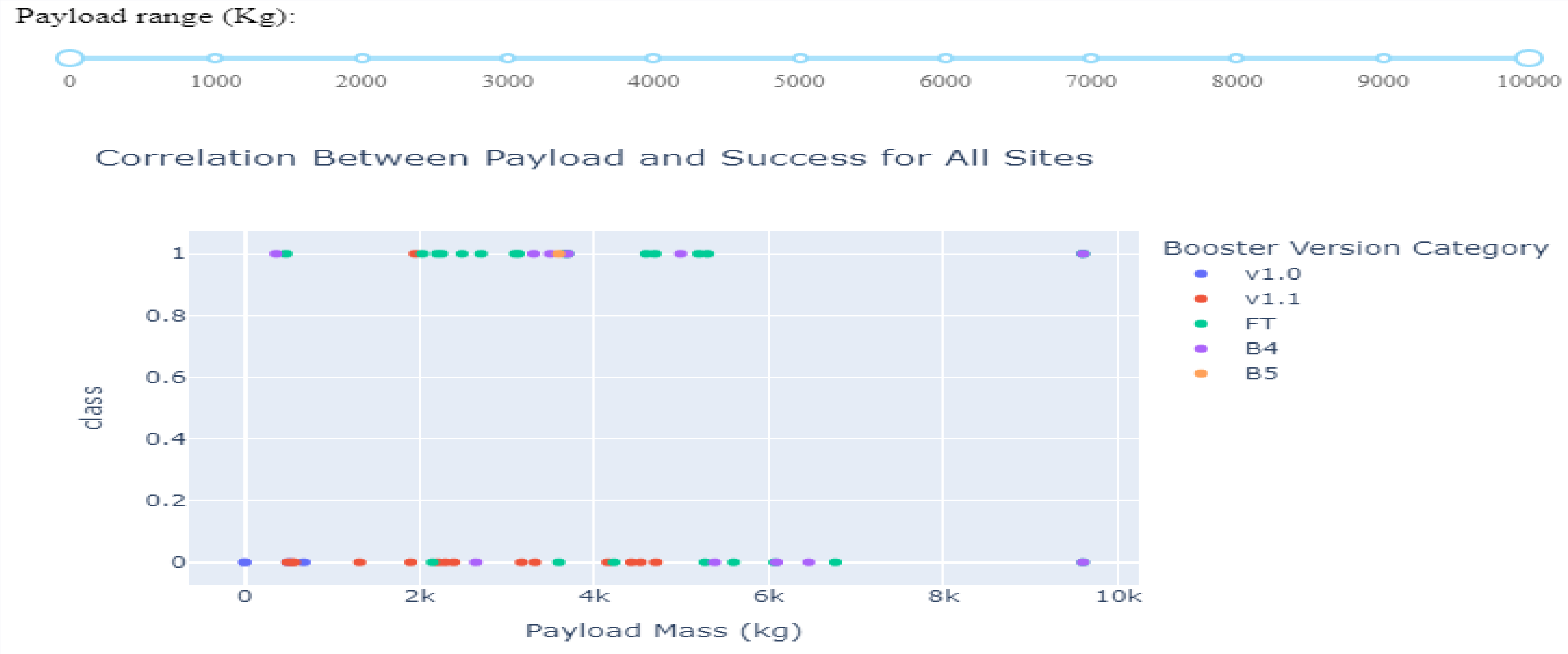
- KSC LC-39A has highest succuss (41.2%)
- CCAFS LC-40 has lowest succuss (14.4%)

Launch Site having highest success



- KSC LC-39A site has highest success rate (76.9%)

Relation Between Payload Mass and Launch Success

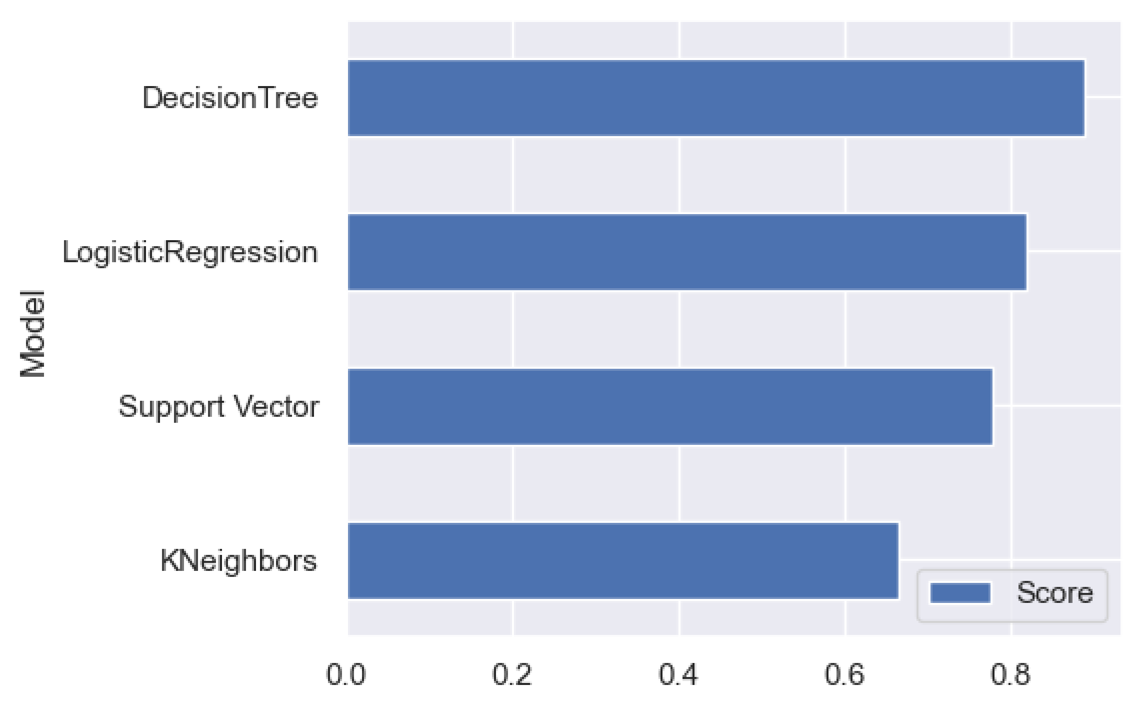


- FT Booster has relatively higher success.
- Payload above 5800Kg usually results into failure

Section 5

Predictive Analysis (Classification)

Classification Accuracy




```python
# Best model SVM  eith accuracy score 0.8333333333333334 on test data
model_df = pd.DataFrame(
    data={ "Model": ["LogisticRegression","Support Vector","DecisionTree","KNeighbors"],
      "Score": [logreg_cv.best_score_,svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_]
      })

model_df.sort_values(by="Score", ascending=False)
```

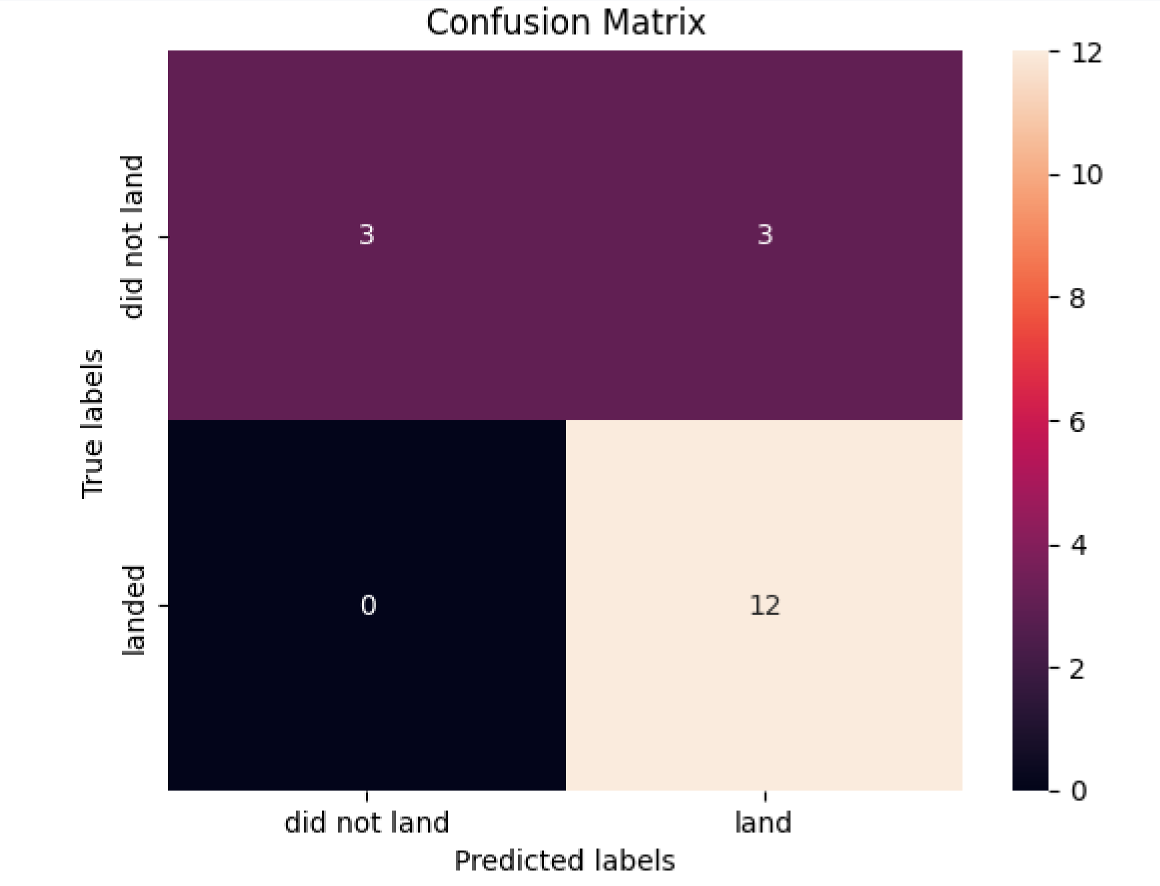
[61] ✓ 0.0s

	Model	Score
2	DecisionTree	0.889286
0	LogisticRegression	0.819643
1	Support Vector	0.777143
3	KNeighbors	0.664286

- DecisionTree has accuracy of 88.9%

Confusion Matrix

- True Positive (TP)=12
- False Negative (FN)=0
- True Negative (TN)=3
- False Positive (FP)=3



Conclusions

- Launch sites with higher success rates are more favored for future missions.
- A significant increase in launch success occurred after 2014.
- The orbits ES-L1, GEO, HEO, SSO, and VELO show the highest success rates.
- KSC LC-39 holds the record for the most successful launches
- Decision Tree classifier proved to be best ML algorithm for the dataset
- We achieved an accuracy of 88.9%

Appendix

Project GitHub URL: All Project Files

Thank you!