IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ryan Ellis
9/25/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with web scraping and SpaceX Api
  - EDA with visualizations and dashboards
  - Predictive Analysis with Machine Learning
- Summary of all results
  - Exploratory Data Analysis Results
  - Predictive Analysis Results

Introduction

- **Background:**
  - 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. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- **Desirables:**
  - Best way to estimate successful landing.
  - Best place to launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Use of SpaceX API (https://api.spacexdata.com/v4/rockets/)
  - Webscraping from Wikipedia
- Perform data wrangling
  - One hot encoded data for landing zones
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

1. Request data from the SpaceX API and Convert into Pandas Dataframe
2. Set up data for list
   Filter data into list
3. Make another Pandas Dataframe

GitHub:
https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%201/01%20Data%20Collection%20API%20Lab.ipynb

```
response = requests.get(spacex_url)
```

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

```
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}
```

```
# Create a data from launch_dict
rocket_data = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

- Webscraped Wiki with Beautiful Soup to get Falcon 9 data
- parsed into Pandas Dataframe

- GitHub:https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%201/02%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

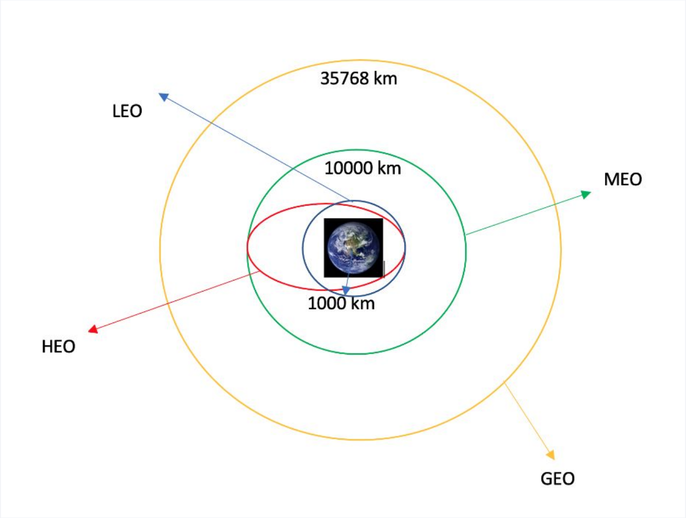
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html')
```

Data Wrangling

- Preformed EDA
- Calculated number of launch sites
- Calculated the number and occurrence of each orbit
- Calculated the number and occurence of mission outcome per orbit type
- Created a landing outcome label from Outcome column
- Exported data to CSV
- GitHub: https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%201/03%20Data%20Wrangling.ipynb



EDA with Data Visualization

Visualized the following:

- Relationship between Flight Number and Launch Site
- Relationship between Payload and Launch Site
- Relationship between success rate of each orbit type
- Relationship between FlightNumber and Orbit type
- Relationship between Payload and Orbit type
- Launch success yearly trend

- GitHub:
  https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%202/05%20EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

1. Display Names
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 succesful landing outcome in ground pad was acheived.
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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub:

https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%202/04%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Used circles to identify launch sites
- Used markers as launch success or failure
  - Green: Success
  - Red: Failure
- Used line to determine distance to coast

- GitHub: https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%203/06%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Built dashboard with Plotly Dash
- Made Pie Charts to show the total successful launches count for all sites
- Made slider with payload range
- Made scatter plot based on slider range
- GitHub: https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%203/07%20spacex_dash_app.py

Predictive Analysis (Classification)

Building the model Steps:

1. Perform exploratory Data Analysis and determine Training Labels
2. create a column for the class
3. Standardize the data
4. Split into training data and test data
5. Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

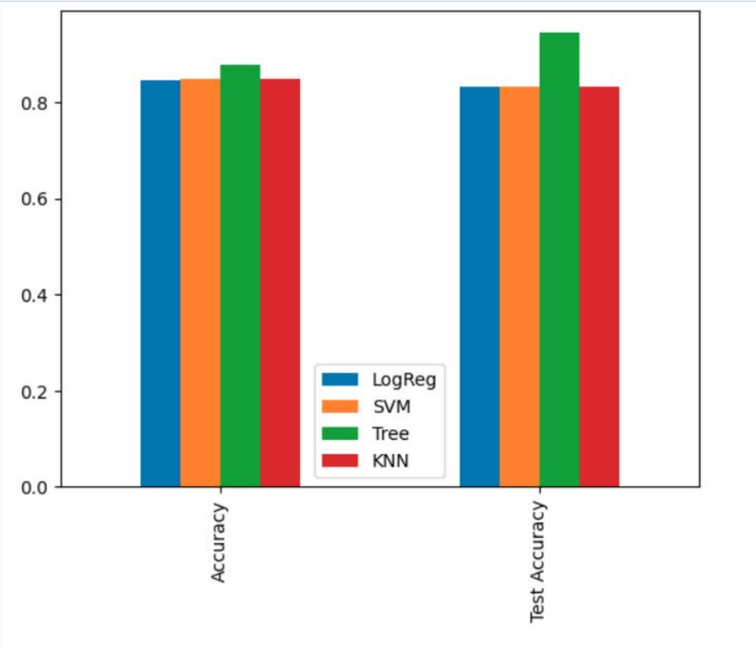
Finding the method performs best using test data

- Used and recorded accuracy of each model

GitHub:
https://github.com/Codeblockz/Ellis_Ryan_AppliedDataScirenceCapstone/blob/main/Week%203/08%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

Predictive Analysis Results



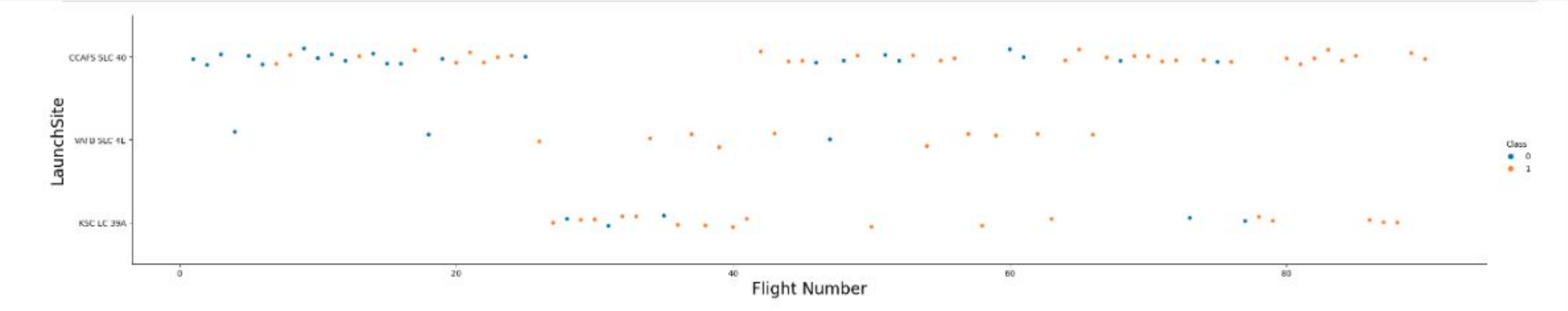
	Accuracy	Test Accuracy
LogReg	0.84643	0.8333
SVM	0.84821	0.8333
KNN	0.84821	0.8333
Tree	0.87679	0.9444

Section 2

Insights drawn from EDA

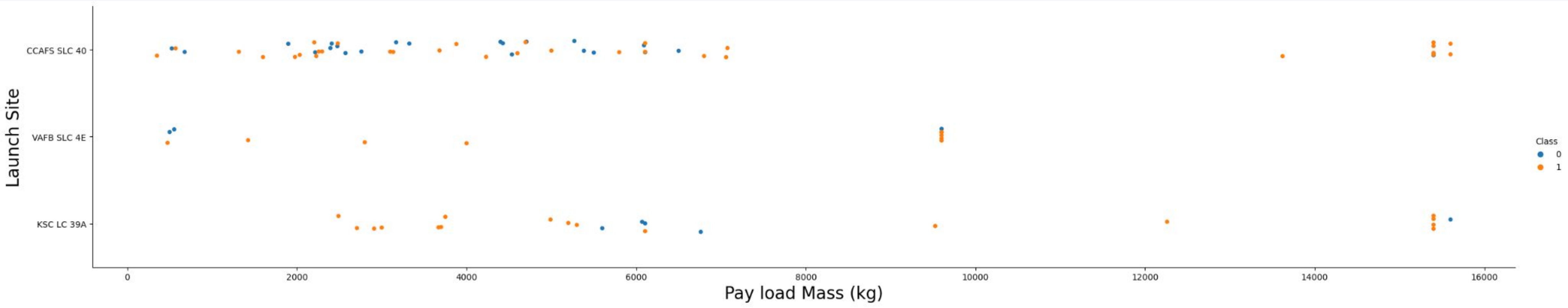
Flight Number vs. Launch Site

- It appears that the larger the flight number, the greater the chance of a successful launch



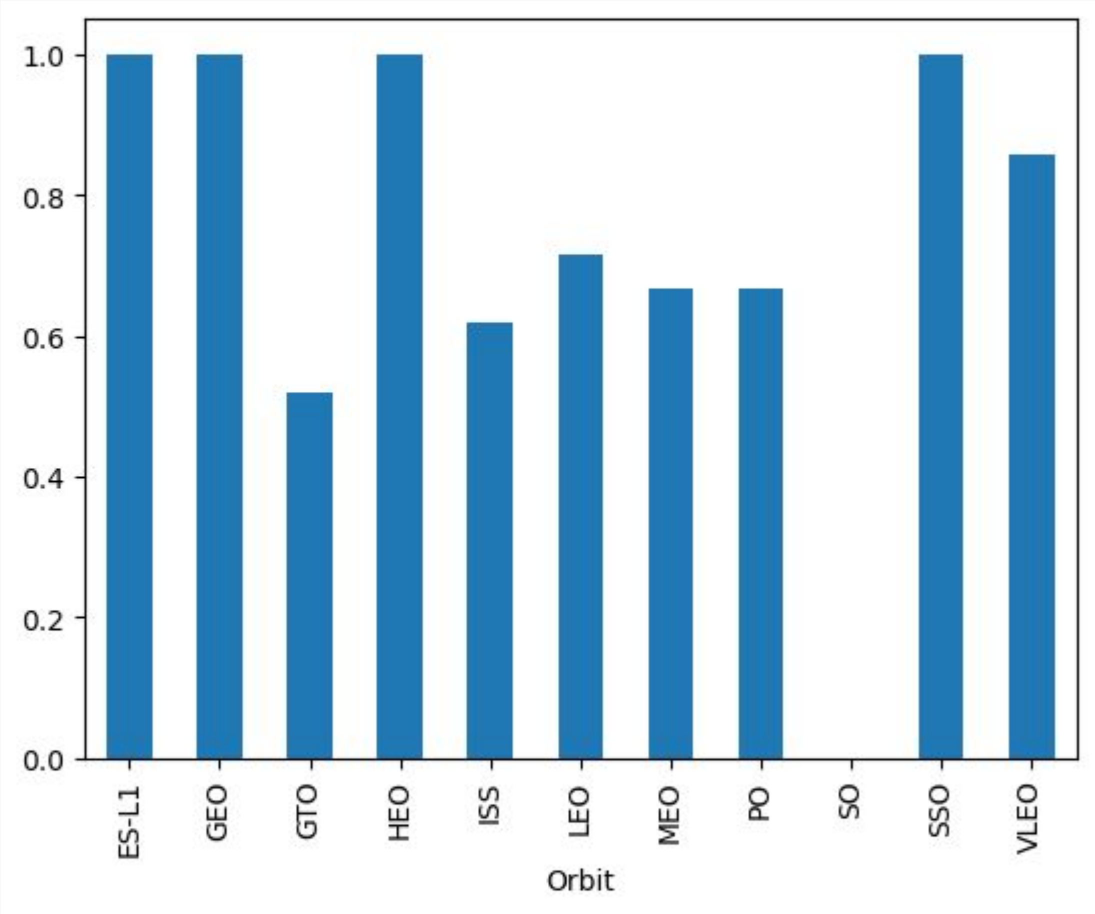
Payload vs. Launch Site

- We found for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



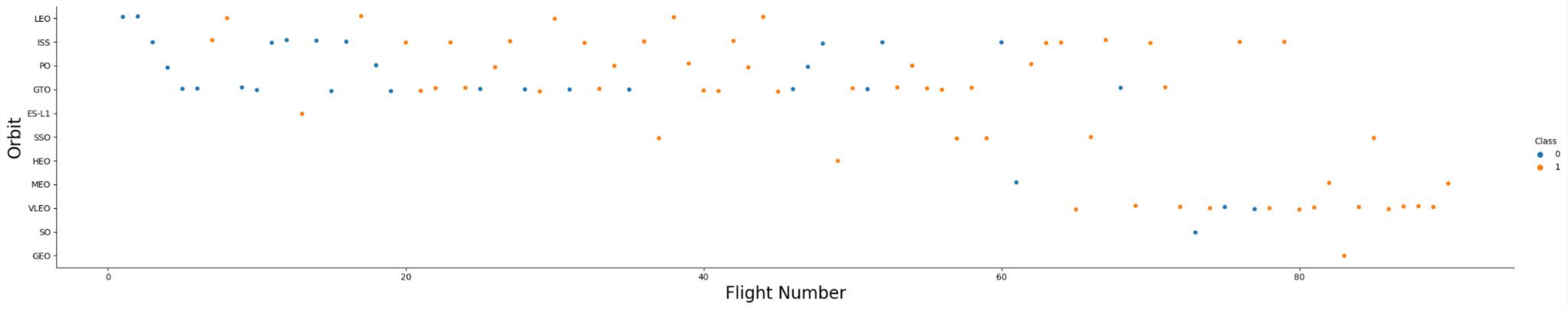
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, and VELO have the highest success rate



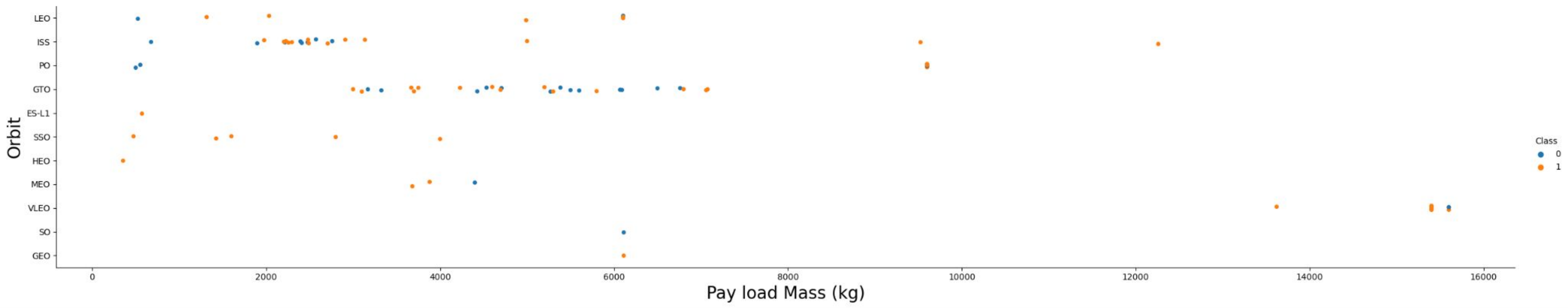
Flight Number vs. Orbit Type

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



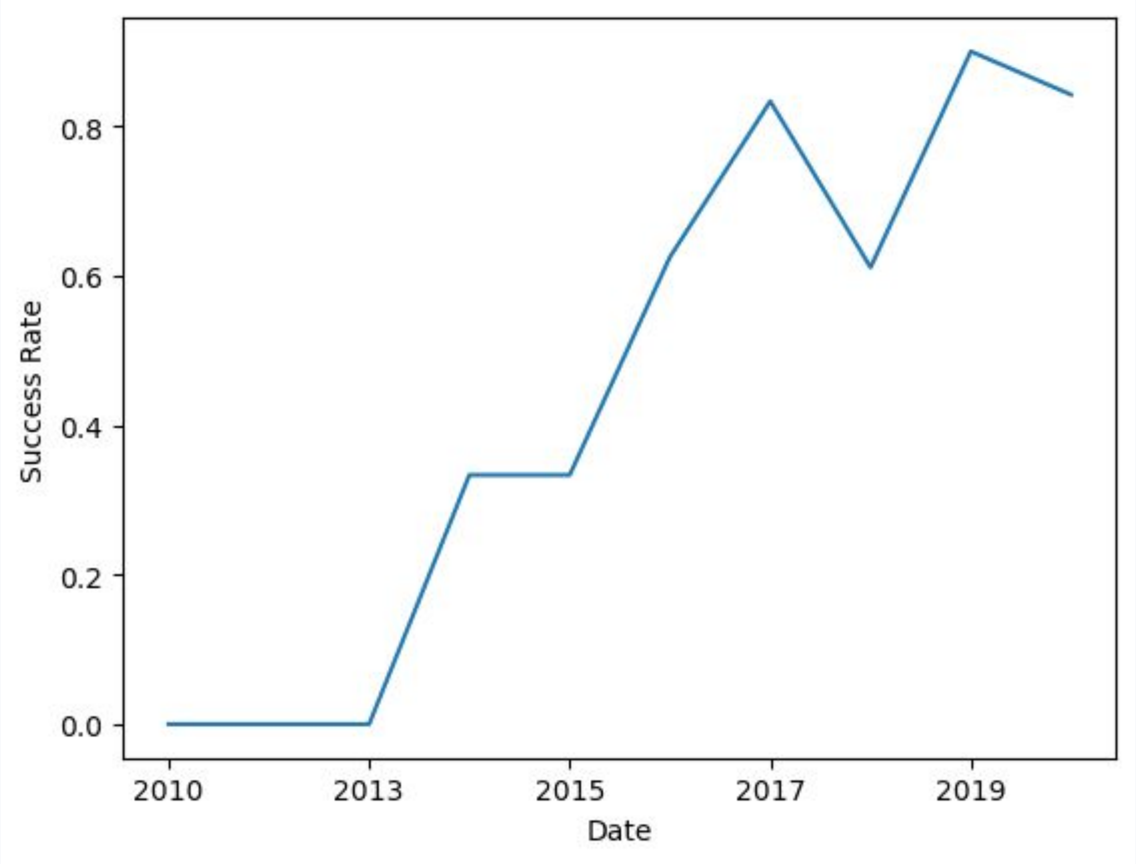
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- Over time, the success rate has increased



All Launch Site Names

- Used DISTINCT to find the unique launch sites

```
In [8]:  %sql select distinct Launch_Site from SPACEXTABLE;

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

Out[8]:

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

Launch Site Names Begin with 'CCA'

- Used LIMIT to limit the result to 5

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

In [9]:
```
%sql select * from SPACEXTABLE where launch_site like 'CCA%' limit 5;
```

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

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Used SUM to calculate the payload mass

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTABLE where customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- Used AVG to calculate the average
- Used LIKE to differentiate booster_Version

```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTABLE where booster_Version like '%F9 v1.1%';
```

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

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

- Used MIN to find the first date
- Used LIKE to determine a successful landing

```
%sql select min(date)  from SPACEXTABLE where Landing_Outcome like '%Success%';
```

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

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Used BETWEEN to restrain results between 4000 and 6000

```
%sql select booster_version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and payload_mass__kg_ between 40
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Used COUNT to show total number
- Used GROUP BY to differentiate the results

```
%sql select mission_outcome, count(*) as Total_Number from SPACEXTABLE group by mission_outcome;
```

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

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

Boosters Carried Maximum Payload

- Used a nested query to filter query

```
%sql select booster_version from SPACEXTABLE where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTABLE);
```

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

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql select  substr(Date, 4, 2) as MONTH, booster_version, launch_site, landing_outcome from SPACEXTABLE where landing_outc
```

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

MONTH	Booster_Version	Launch_Site	Landing_Outcome

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

- Used a combination of WHERE, GROUP BY, and COUNT to order results of query

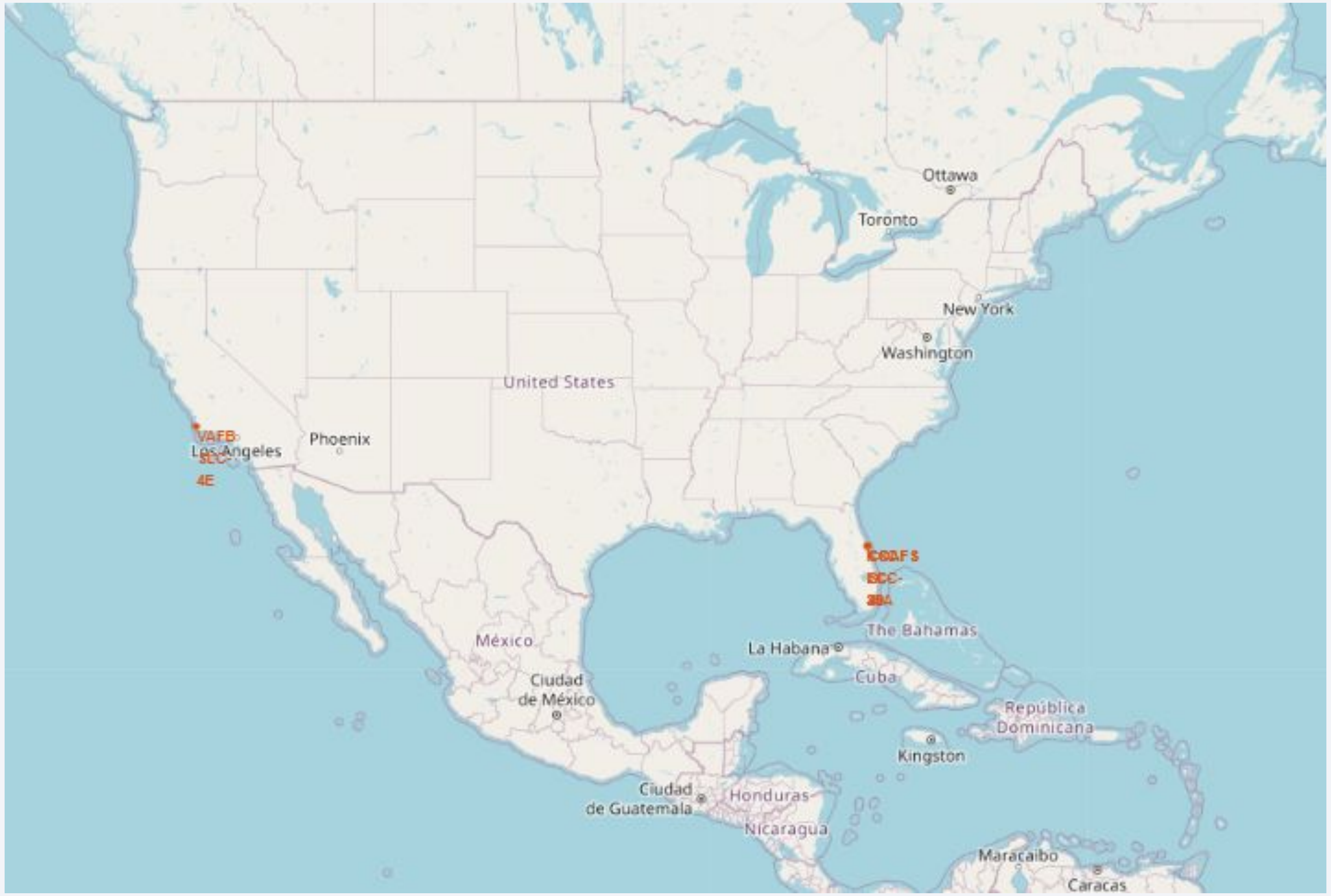
```
%%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) FROM SPACEXTABLE
WHERE DATE between '2010-04-06' and '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

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

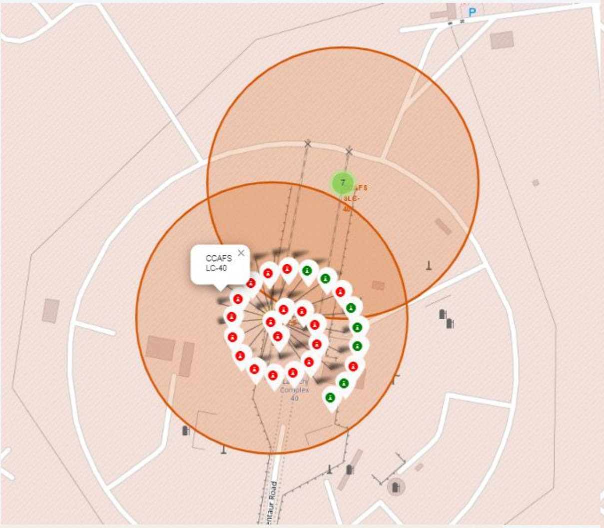
Section 3

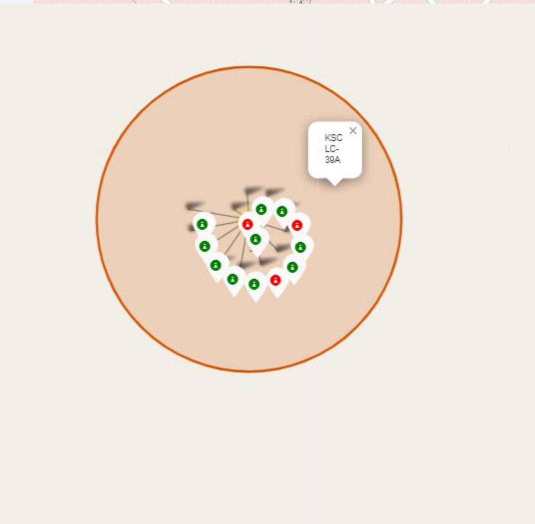
Launch Sites Proximities Analysis

All Launch Sites



Markers Showing Launch Sites





Florida Launch Sites

- Green Markers show success
- Red Markers show faulire

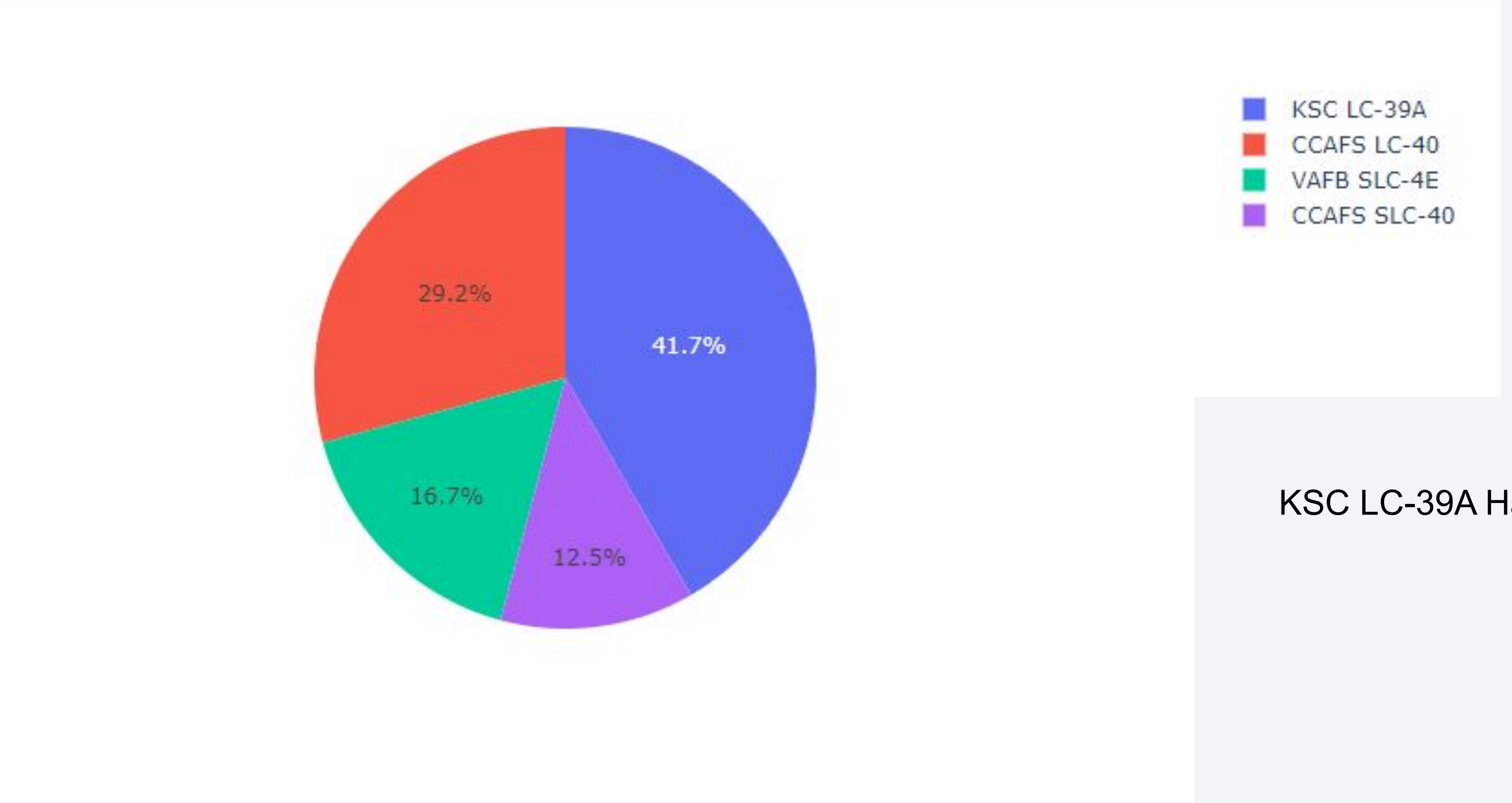
Distance to Landmarks

Distance to Road: 0.66 KM
Distance to Coast: 0.95 KM
Distance to Train: .95 KM

Section 4

Build a Dashboard with Plotly Dash

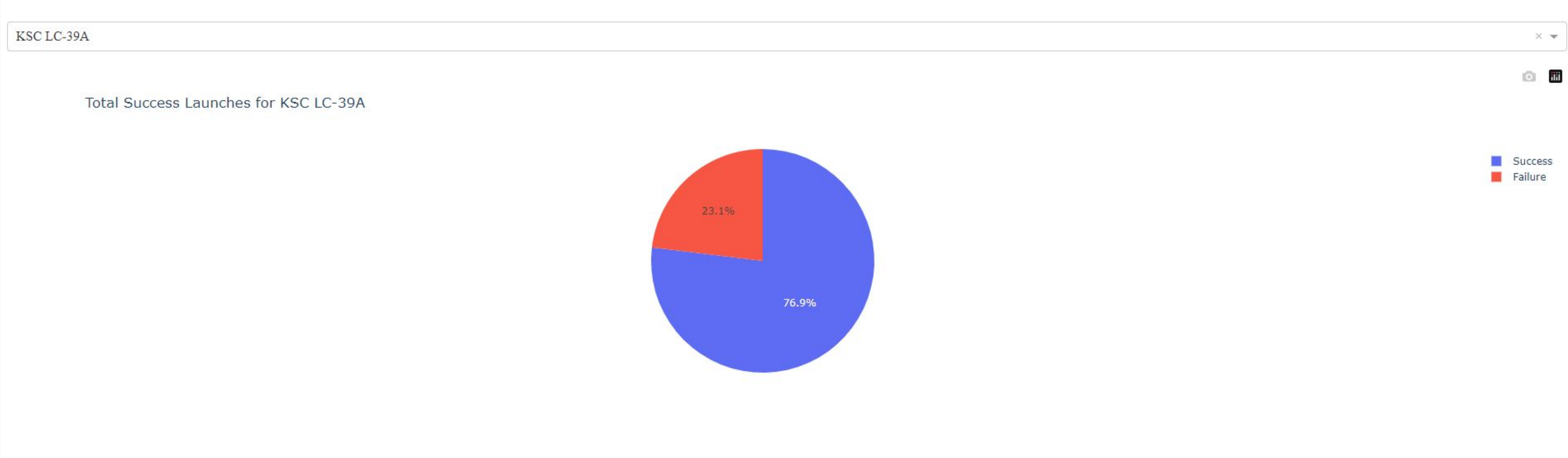
Total Success Launches By Site



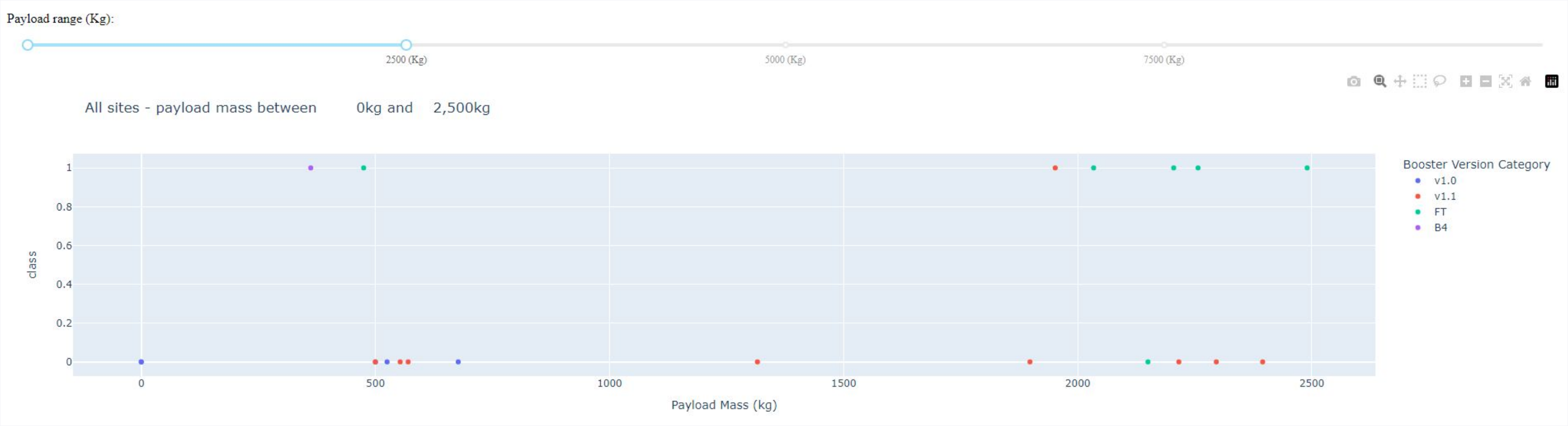
KSC LC-39A Has the most success

KSC LC-39A Success Rate

Success Rate: 76.9



Payload vs Launch Outcome for all sites

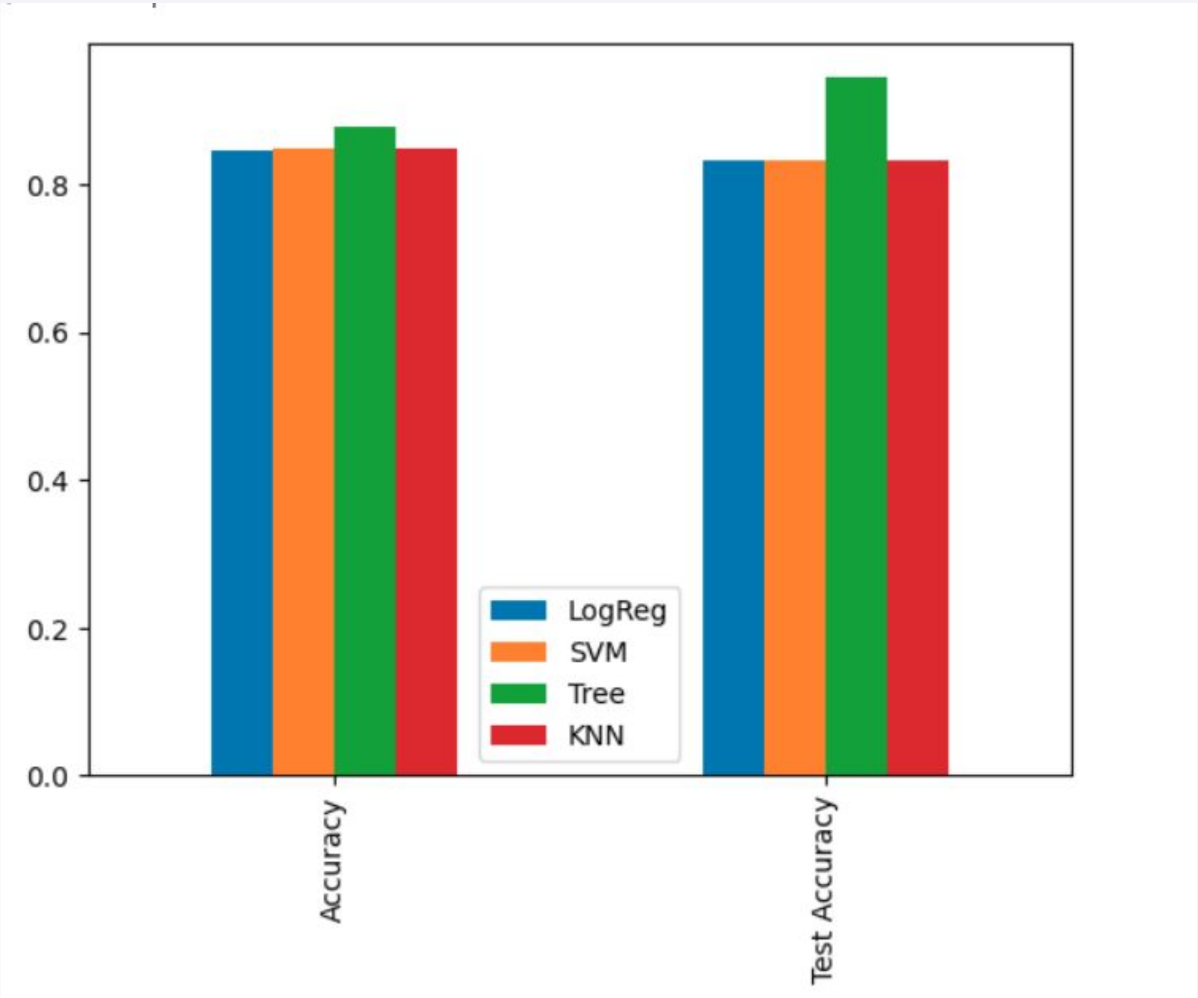


For the rockets with a lower payload, there was more success.

Section 5

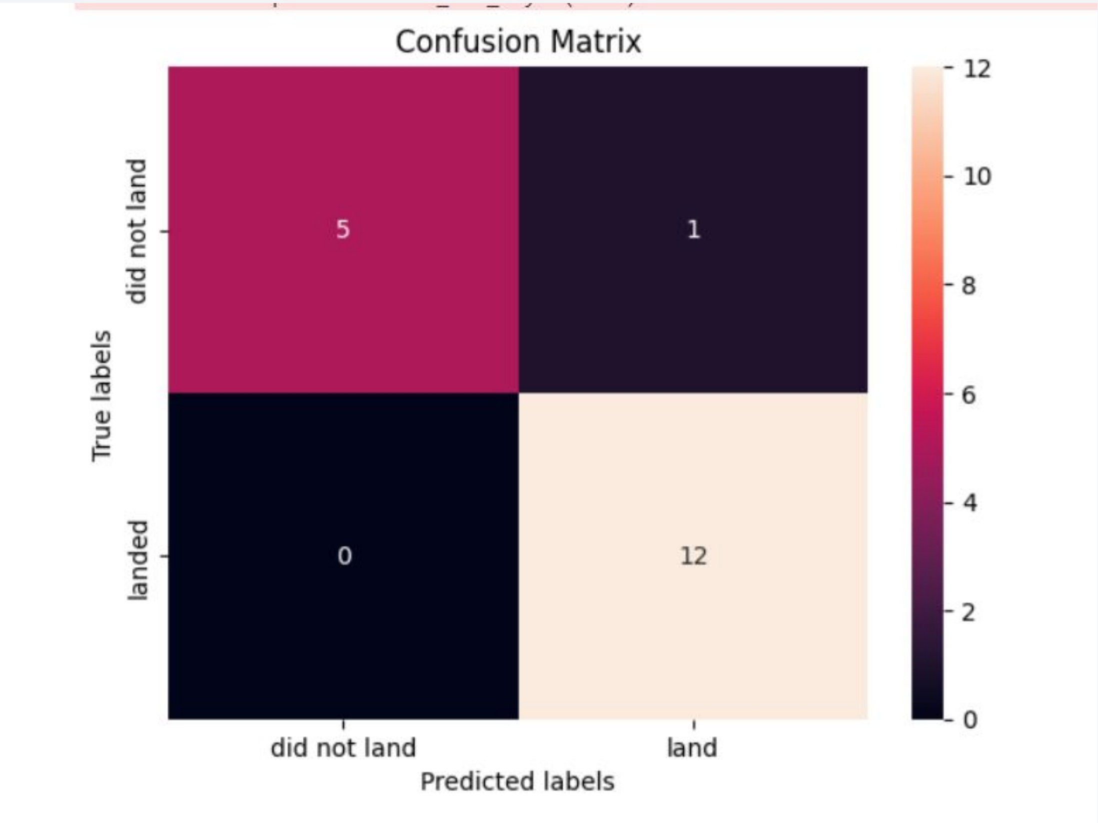
Predictive Analysis (Classification)

Classification Accuracy



The Decision Tree classifier performed the best compared to the other models

Confusion Matrix



Shows the accuracy of the model

Conclusions

- More flights = better success rate at launch site
- The success rate increased by year
- ES-L1, GEO, HEO, SSO, and VELO have the highest success rate
- Low payload mass showed better results than a high payload mas

Appendix

- IBM Data Science Professional Certificate

Thank you!