IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martin Blindheimsvik
27-April-2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with SQL
  - EDA with Visualization
  - Data visualization using Folium
  - Building a dashboard with Plotly Dash
  - Predictive analysis
- 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. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.
- Problems you want to find answers
  - We want to predict whether the first stage of Falcon 9 will land successfully.

Section 1

Methodology

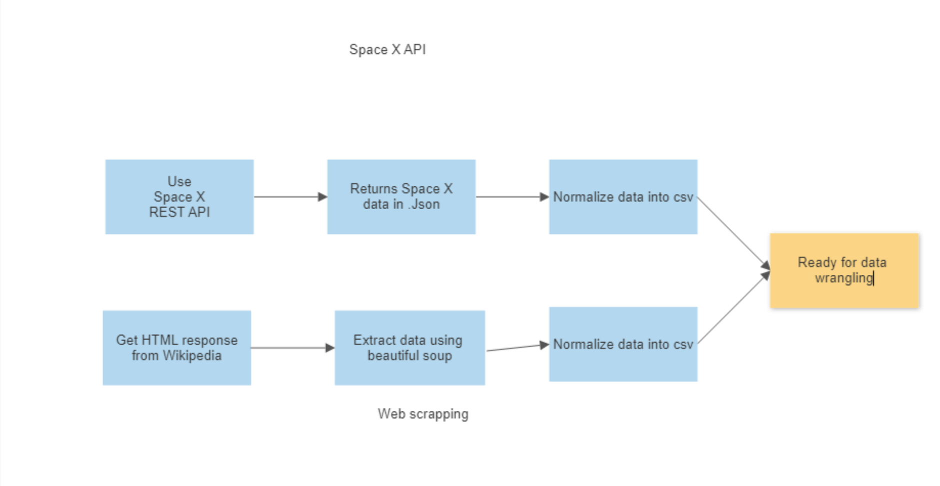
Methodology

Executive Summary

- Data collection methodology:
  - Space X Rest api
  - Web scraping from wikipedia
- Perform data wrangling
  - One hot encoding was used to standardize the data so it could be used for training machine learning models. Pandas and numpy was also used to examine and clean null values and relevant/irrelevant variables
- 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, SVM, Decision tree, KNN models were fitted and evaluted

Data Collection

- Space X launch data was collect from the Space X launch REST API.
  - The dataset contains information regarding launches, rockets used, payload, launch specifications, landing specifications and outcomes.
- Falcon 9 data was collected using Web scraping
  - The dataset was scraped from Wikipedia using Beautful soup



Data Collection – SpaceX API

- Data collection using SpaceX REST calls

1. Getting response from API

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

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

2. Converting API to .JSON

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

3. Applying functions to clean data

```
# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(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}
```

- [SpaceX API Notebook]

Data Collection - Scraping

- Web scraping for wikipedia

1. Getting response from HTML

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

2. Creating Beautfulsoup object

```
soup = BeautifulSoup(response.content, 'html.parser')
```

3. Finding tables

```
html_tables = soup.find_all('table')
```

4. Getting column names

```
column_names = []
tables = first_launch_table.find_all('th')
print(tables[1])

for th in tables:
    name = extract_column_from_header(th)
    if name is not None:
        if len(name) > 0:
            column_names.append(name)
```

5. Creation of dictionary to hold data

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. We then append the data to keys and convert the dictionary to a dataframe

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

7. Lastly write dataframe to a csv file

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

[Web scraping - Notebook]()

Data Wrangling

1. Check null values and examine variable types
2. Calculate the number of launches on each site
3. Calculate the number of occurrences of each orbit
4. Calculate the number and occurrence of mission outcome per orbit type
5. Create a landing outcome variable from Outcome column
6. Examine the succesrate and write our dataframe to a csv file.

[Data wrangling - Notebook]

EDA with Data Visualization

1. Scatterplots
   - These were used to examine potential relationships between variables

2. Bar chart
   - Bar charts were used to visualize occurrence
   - We also used them to see if there was a relationship between orbit type and success rate
3. Line chart
   - A line chart was used to check how success rate varied as a function of year

EDA with Data Visualization - Notebook

EDA with SQL

- SQL queries performed included
  - Displaying the names of the unique launch sites
  - Displaying 5 records where launch sites began with the string 'CCA'
  - 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 dates when the first successful landing outcome in ground pad was achieved
  - Displaying names of the boosters which had success in drone ship and had payload mass between 4000-6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster versions that carried the highest payload mass
  - Listed the records displaying month names, failure landing outcomes in drone ship, booster versions, and launch site for the months in 2015
  - Ranked the count of successful landing outcomes between 04-06-2010 and 20-03-2017

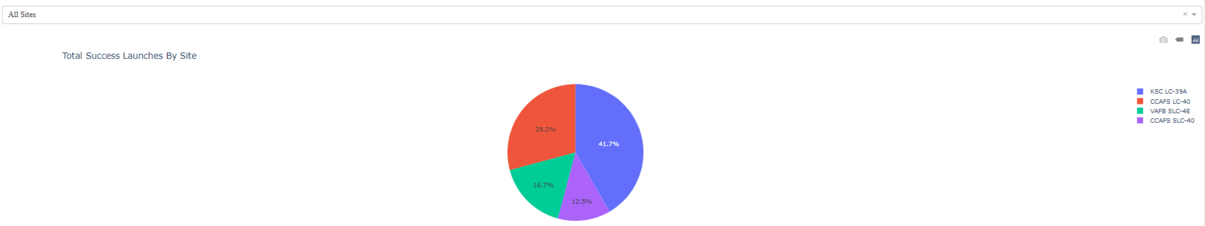
[EDA with SQL - Notebook]

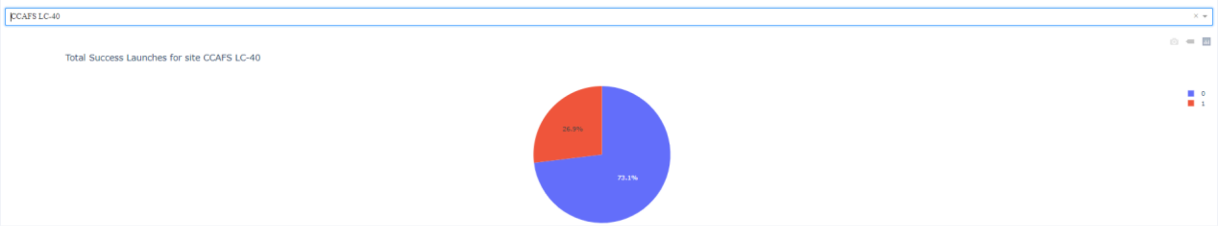
Build an Interactive Map with Folium

- All launch sites were marked on the map along with corresponding success/failure launches for each site and the distance from the sites to nearby infrastructure. We added the closest highway, railroad and city to our selected launch site. This was done because its important to account for when doing a launch.

[Folium - Notebook]()

Build a Dashboard with Plotly Dash





- A pie chart for all launch sites was added to look at and compare the success rates for the various launch sites.
- A Pie chart was added to examine individual failures and successes of a launch site
- Lastly, a scatter plot was added to examine the correlation between payload and success rate

Plotly Dash Github

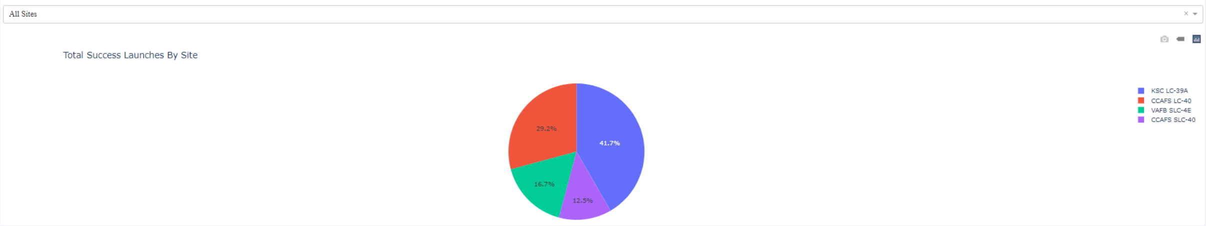
Predictive Analysis (Classification)

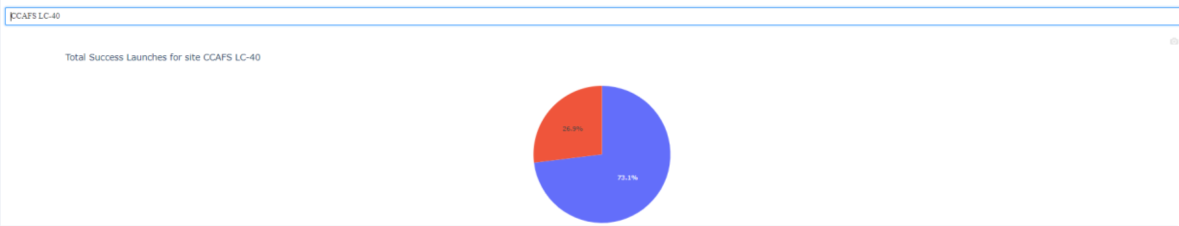
- We first standardized the data using StandardScaler. It was then split using the train_test_split method with 20% of the data being used as the test set.
- A logistic regression, SVM, Decision tree, and KNN model were fitted to the training data. The models were tested on the 20% test set.
- GridSearchCV was used to perform cross validating to find the best parameters for each model.
- All four models had the same test accurracy of 83.3%.

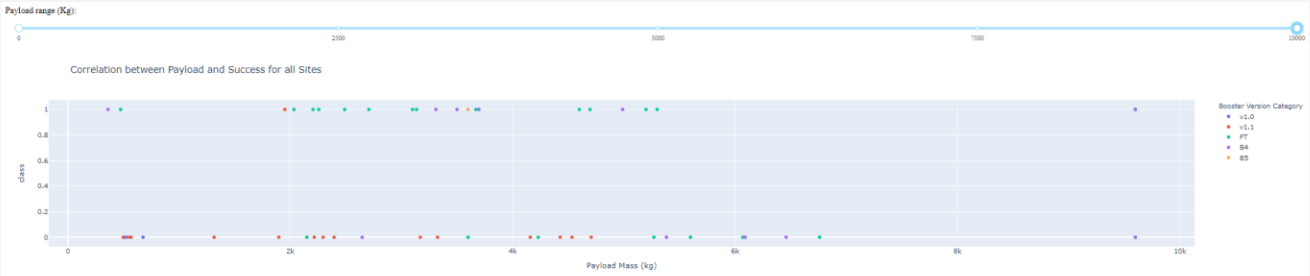
[Predictive Analysis - Notebook]

Results

- Success rate in general has increased over the years.
- Orbits ES-L1, GEO, HEO and SSO have the highest success rates at 100%, while GTO gas the lowest non-zero success rate at 50%. Orbit SO had a 0% success rate.
- The success rate is higher for lower payloads.





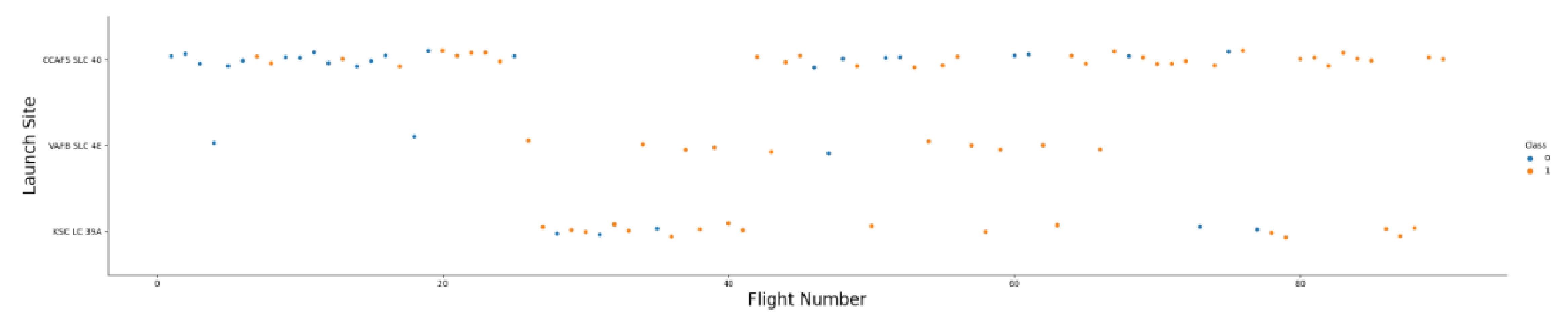


- The LR, SVM, Decision Tree, and KNN models all performed the same for our data.

Section 2

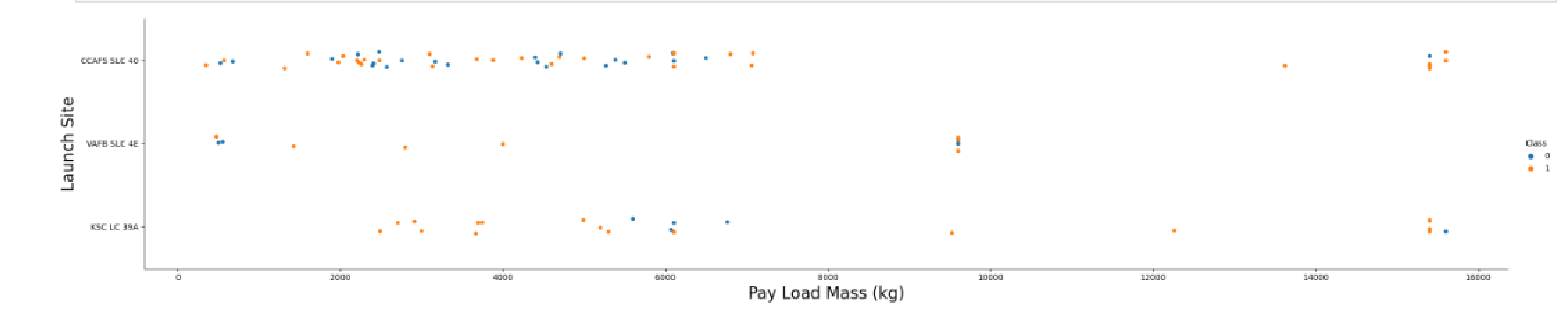
Insights drawn from EDA

Flight Number vs. Launch Site



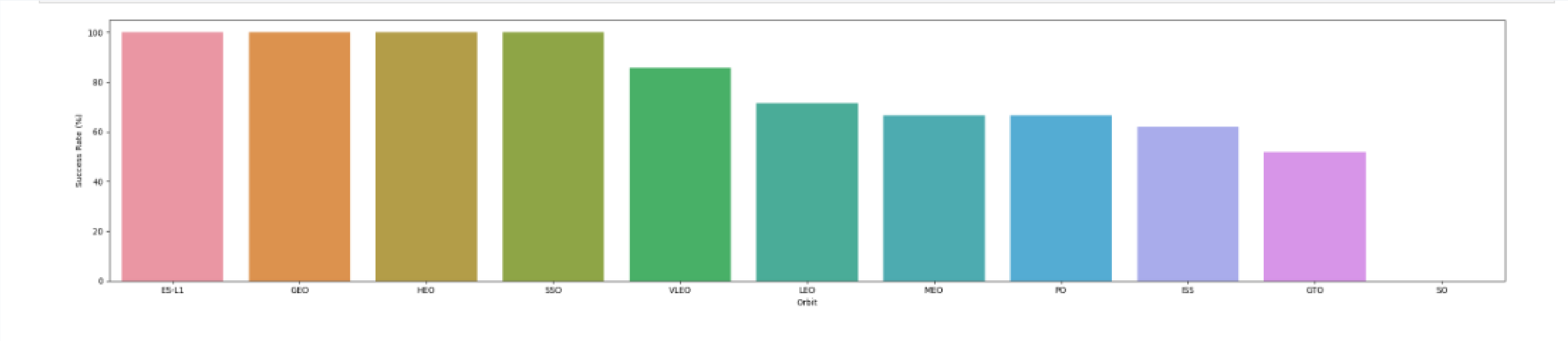
- There are significantly more launches from CCAFS SLC 40 than other sites. Success rate also increases as a function of flight number for all sites.

Payload vs. Launch Site



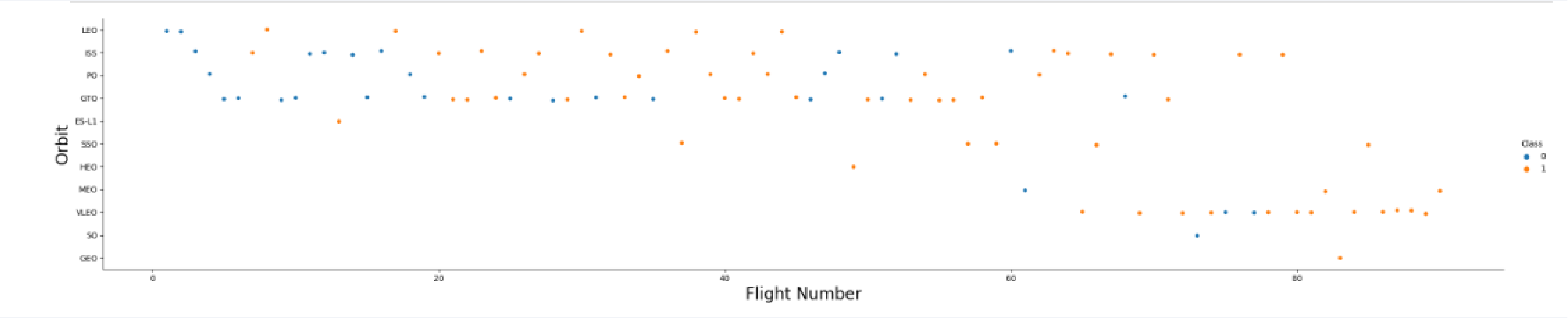
- Most launches have Pay load less than 7000 kg.
- There are very few launches with payloads between 7000 and 15000 kg

Success Rate vs. Orbit Type



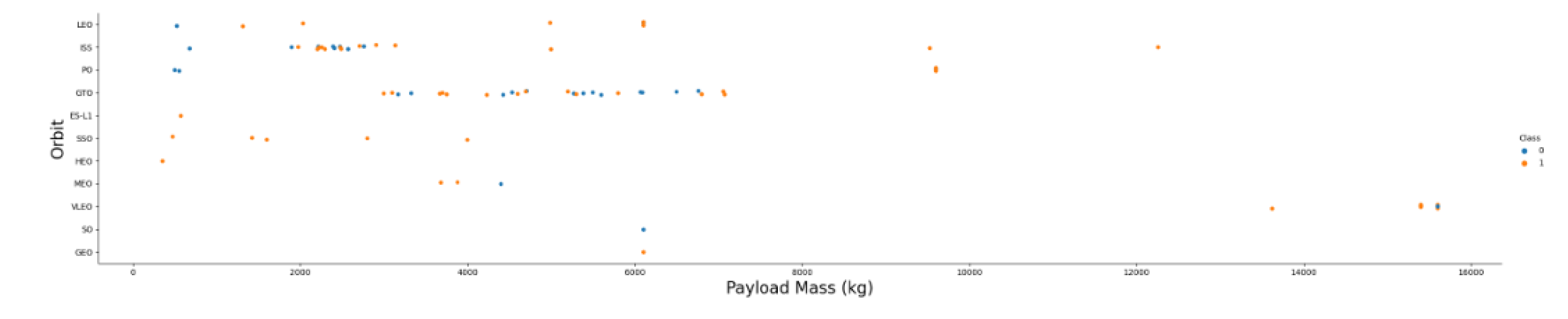
- Orbits ES-L1, GEO, HEO and SSO have the highest success rates at 100%, while GTO gas the lowest non-zero success rate at 50%. Orbit SO had a 0% success rate.

Flight Number vs. Orbit Type



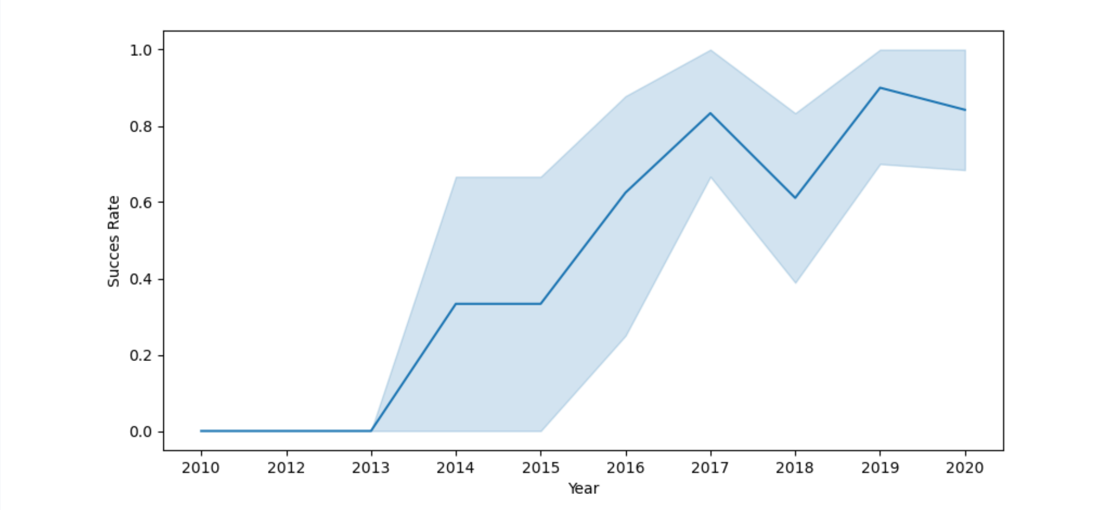
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- there is a strong correlation for ISS, Polar and LEO with payloads. They have more successful landings with higher payloads than other orbits.

Launch Success Yearly Trend



- Launch success rate has increased significantly on average since 2013. The rate increased rapidly until 2017 and has tapered off and stabilized since.

All Launch Site Names

- **%sql** select distinct LAUNCH_SITE from SPACEXTBL;
- Query displays the unique values of the Launch_Site variable from SPACEX table

Out[10]:

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;
- Query displays 5 records based on Launch_Site values starting with 'CCA'

Out[16]:

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_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
- Displays total payload mass carried by boosters launched by NASA (CRS)

Out[22]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- **%sql** select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION like 'F9 v1.1%'
- Displays the average payload mass carried by booster version F9 v1.1

Out[28]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- **%sql** select min(DATE) from SPACEXTBL where "Landing _Outcome" = 'Success (ground pad)';
- Query lists the date when the first successful landing was achieved

min(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- **%sql** select distinct BOOSTER_VERSION from SPACEXTBL where (PAYLOAD_MASS__KG_ between 4000 and 6000) and ("LANDING _OUTCOME" = 'Success (drone ship)');

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(*) from SPACEXTBL group by MISSION_OUTCOME;
- Lists total number of successful and failure mission outcomes

Mission_Outcome	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) order by 1;
- Query checks the max payload and gets all booster versions with that value

Out[44]:

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

2015 Launch Records

- %sql select substr(DATE, 4, 2) as Month, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where ("LANDING _OUTCOME" = 'Failure (drone ship)') and substr(Date,7,4)='2015';
- Lists months, failure_landing outcomes in drone ship, booster versions and launch sites for the year 2015

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

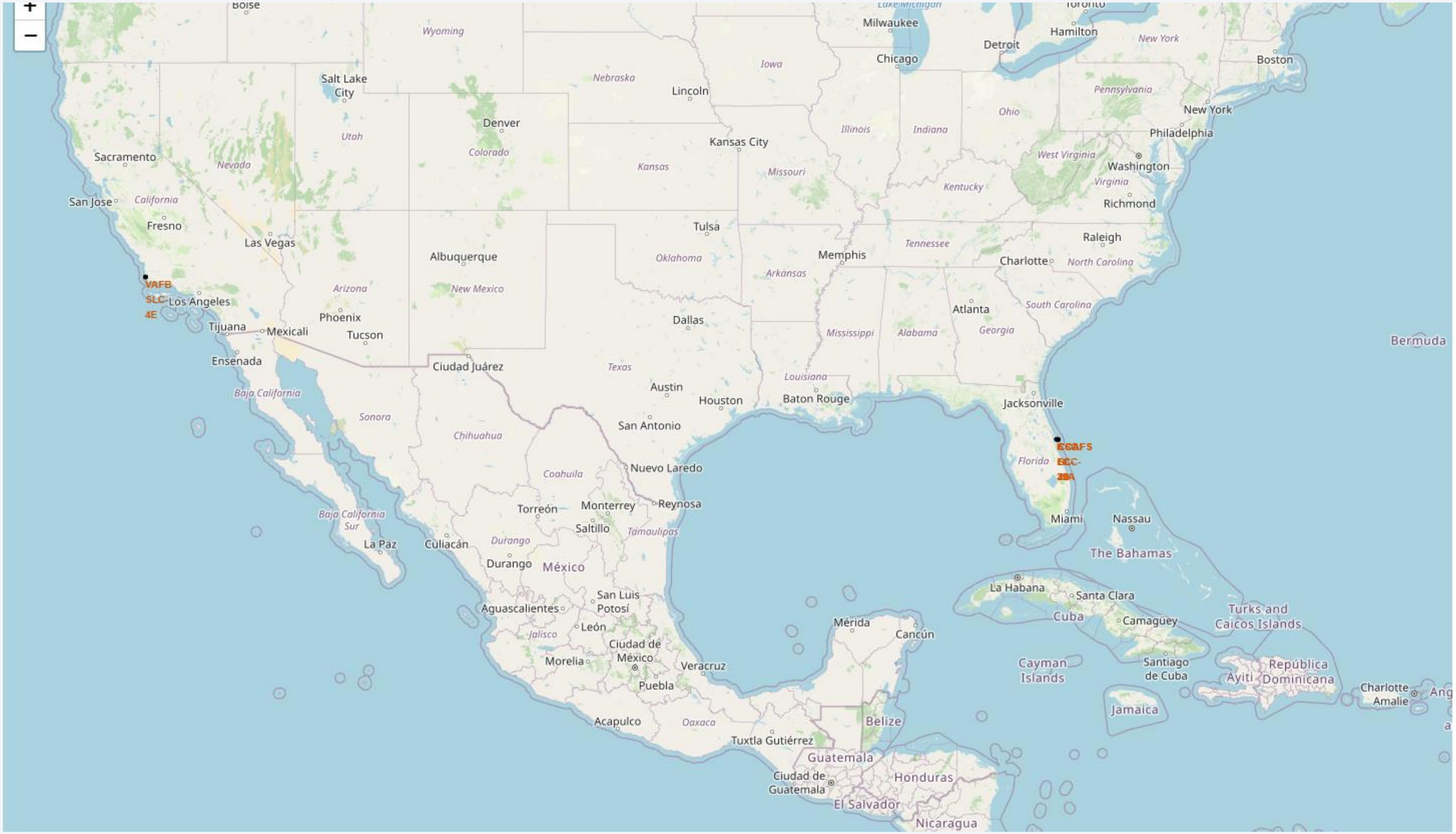
- %sql select "LANDING_OUTCOME", count(*) as qty from SPACEXTBL where ("LANDING_OUTCOME" like '%Success%') and (substr(DATE, 4, 2) between'04-06-2010' and '20-03-2017') group by "LANDING_OUTCOME" order by qty DESC;
- Query selects landing outcomes between 2010-06-04 and 2017-03-20 and orders them in descending order

Landing_Outcome	qty
Success	31
Success (drone ship)	10
Success (ground pad)	7

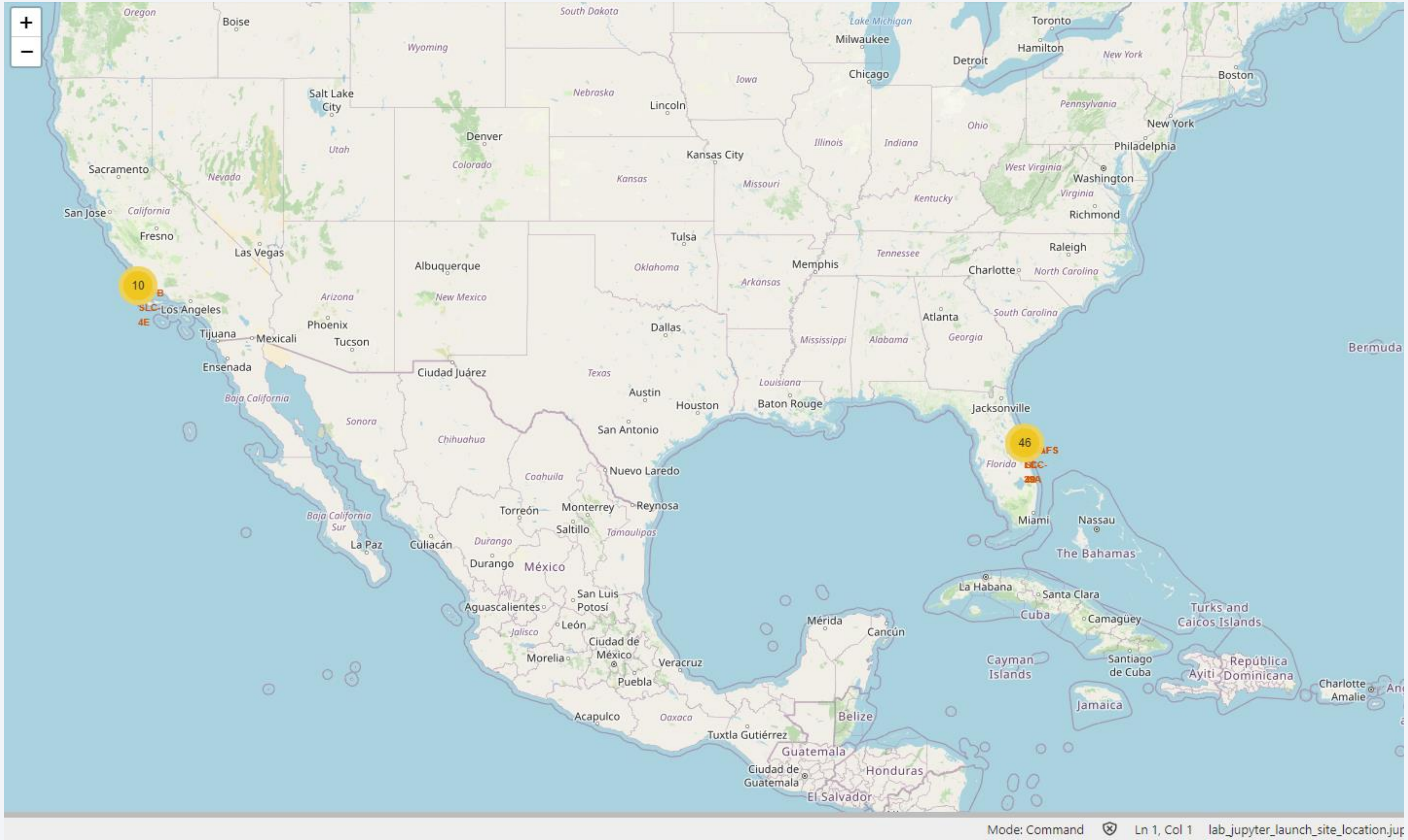
Section 3

Launch Sites Proximities Analysis

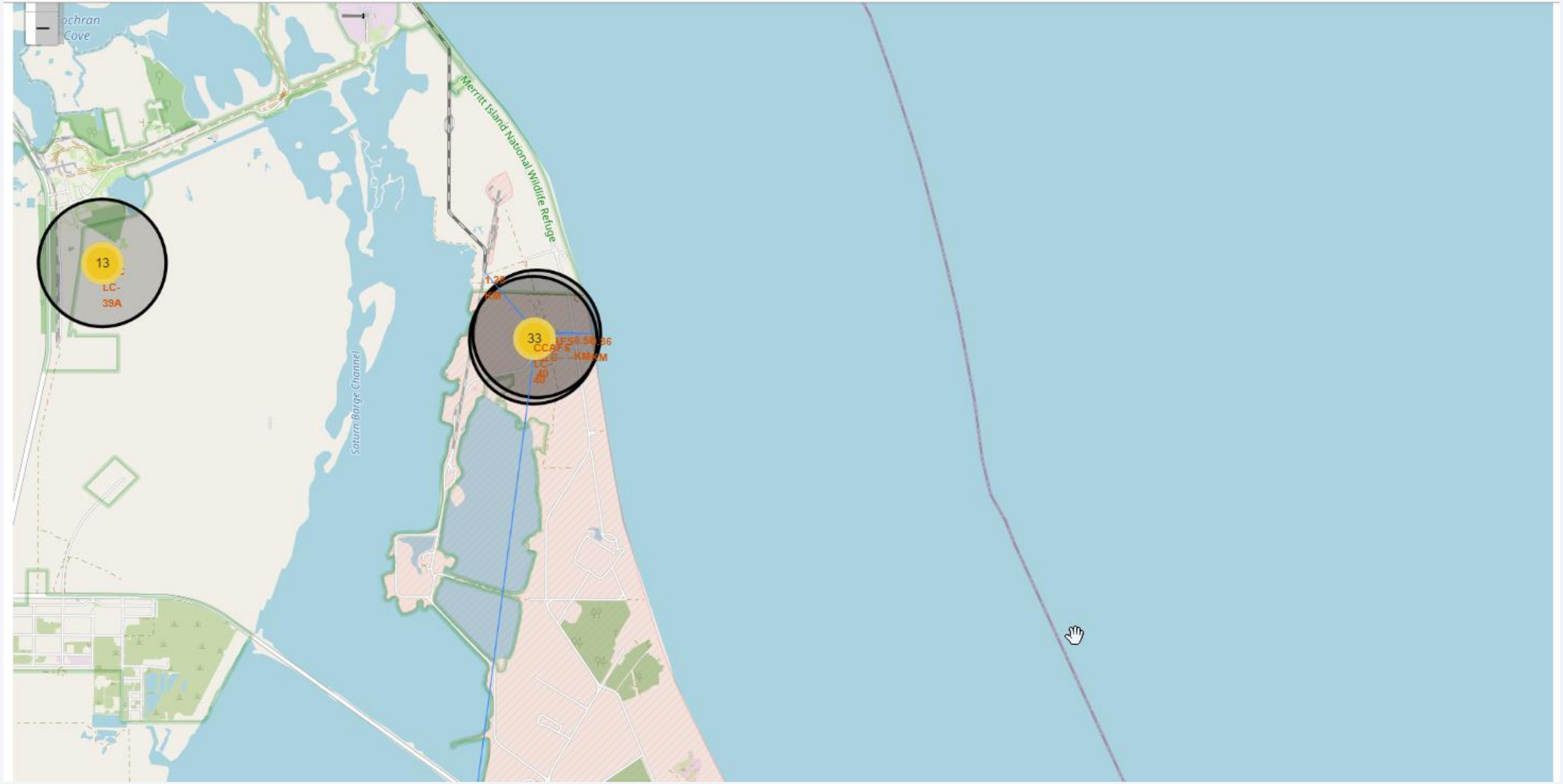
All launch sites marked on a map



All launch sites with success/failures marked



Selected launch site with nearby railway, highway and city

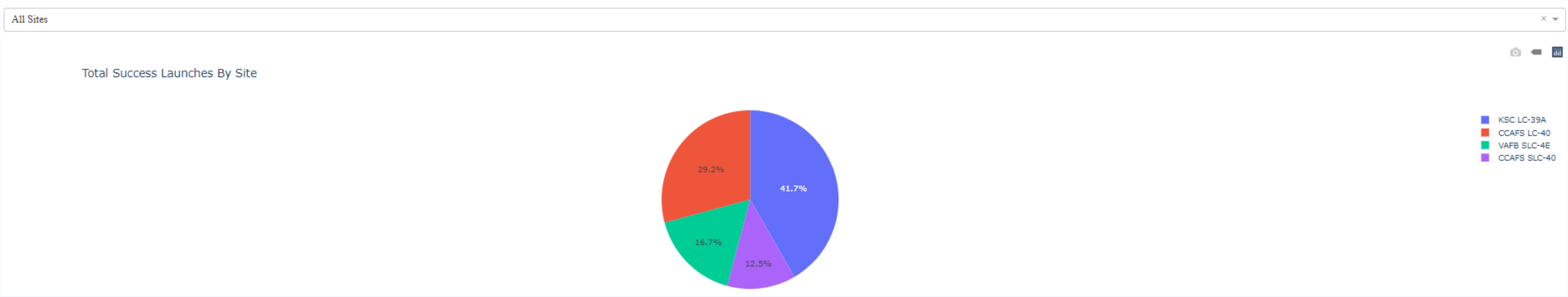


Section 4

Build a Dashboard with Plotly Dash

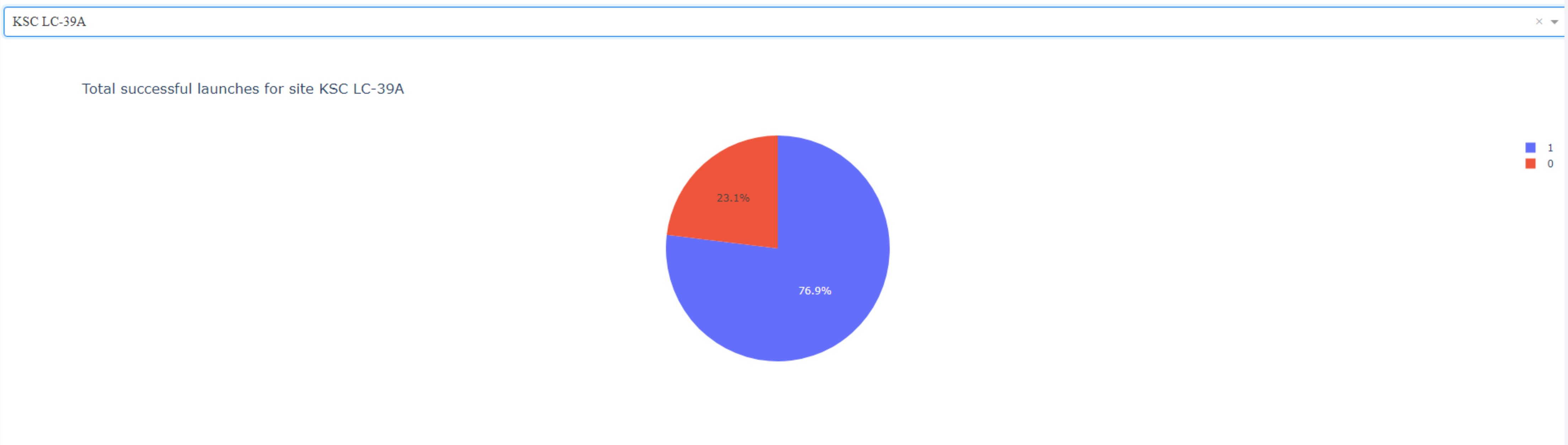
Total success launches by all sites

- Total successful launches by sites



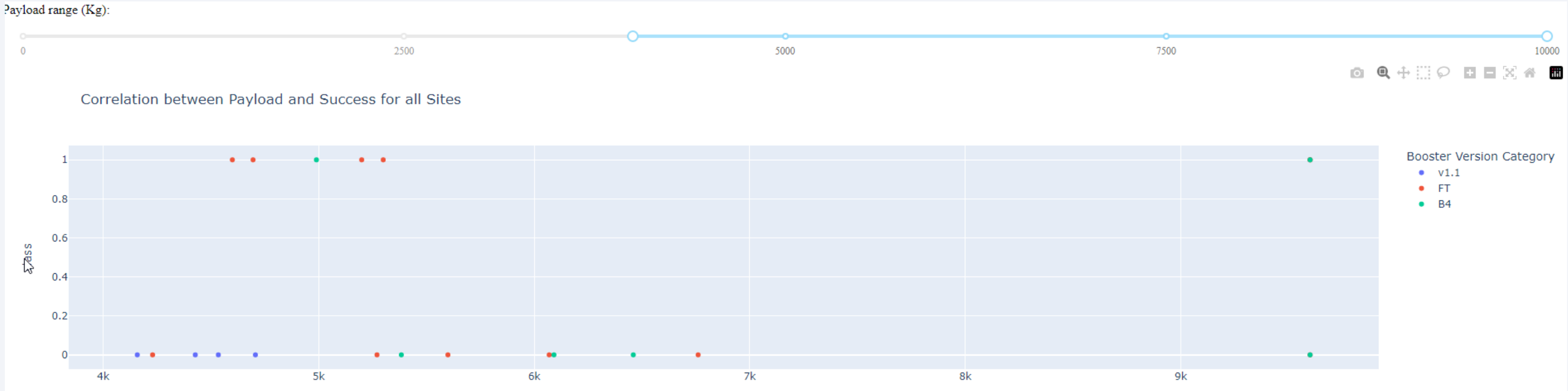
Success rate by site

- KSC LC-39A is the Launch site with the highest success rate, the rate is 76.9%



Payload vs launch outcome

- We can see that the success rate is higher for lower payloads



Section 5

Predictive Analysis (Classification)

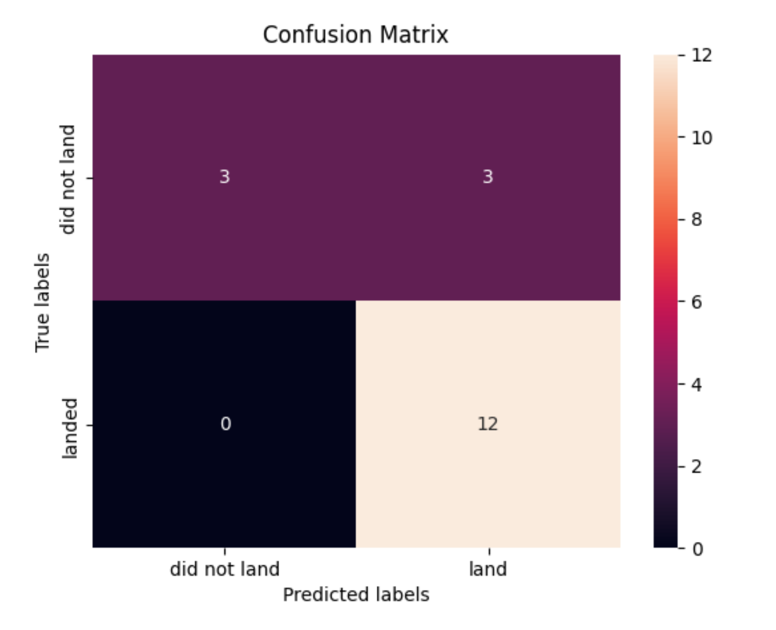
Classification Accuracy

]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

- The confusion matrix for the KNN shows that 3 that were labeled did not land were wrongly predicted as landed. The rest were predicted correctly.



Conclusions

- All ML models performed the same
- Low weighted payloads were more likely to land successfully than ones that weighed more
- The successful launches of Space X increased rapidly until 2017 and then the rate of increase stablized.
- KSC LC 39A was the most successful launch site
- Orbit GEO, HEO, SSO, ES L1 had the best success rates

Thank you!