IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chad Diao
9/16/2022

Outline

Executive Summary

- ## Summary of Methodologies
  - Data collection via API & web scraping
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Interactive map with Folium
  - Building dashboard with Plotly Dash
  - Machine Learning Prediction
- ## Summary of all results
  - Predictive analysis results using four models (machine learning lab)
  - Logistic Regression (logreg)
  - Support Vector Machine (svm)
  - Decision Tree Classifier (tree)
  - K Nearest Neighbors (knn)
  - All models gave similar findings, with an accuracy rate of 83.33%.

Introduction

Background:

- SpaceX is a revolutionary American spacecraft manufacturer that is able to launch their Falcon 9 rocket for a cost of 62 million dollars, with other companies costing up to 165 million dollars. This drastic difference in cost is mostly due to the reusing of the first stage of the launch. If we can determine if the first stage will land successfully, we can determine the cost of the launch. The goal of this project is to predict the probability of a successful landing outcome of the first stage of the rocket.

Problems:

- What are the factors that contribute to the landing outcome?
- What are the relationships between each of the variables?
- What are the ideal conditions to achieve the best landing success rate?

SpaceX Falcon 9 Rocket Landing (Wikipedia)

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding, removing irrelevant columns, adding "Class" column to classify whether a landing was successful or not
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models with GridSearchCV

Data Collection

- Data sets were collected in a process involving both API requests from SpaceX API and web scraping data from tables in Wikipedia using BeautifulSoup.
- Rest API: using the get request and decoding the Json response content into a pandas dataframe.
- Web Scraping: using BeautifulSoup to extract data from HTML table and converting it to a pandas dataframe.
- The following slides will display the flowchart of data collection from API and web scraping.

Data Collection – SpaceX API

Perform get request on SpaceX Rest API → API returns data as JSON file → Clean data and fill in missing values → Create dictionary to cast to DataFrame

- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/Data%20Collection%20API.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

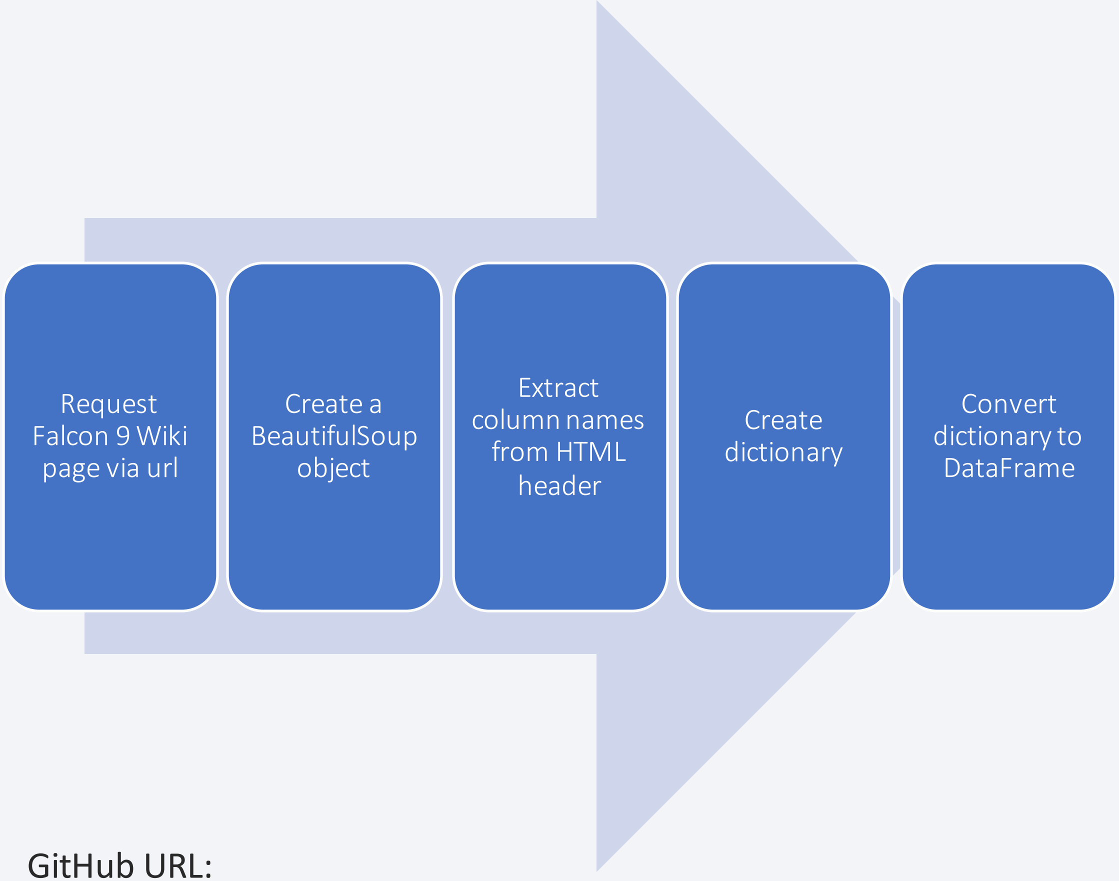
```
response = requests.get(spacex_url)
```

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

```
# 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 ro
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 t
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)]
```

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

Data Collection - Scraping




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

```python
soup = BeautifulSoup(data, 'html5lib')
```

```python
column_names = []

# Apply find_all() function with `th` element on
# Iterate each th element and apply the provided
# Append the Non-empty column name (`if name is

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

```python
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with eac
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']=[]
```

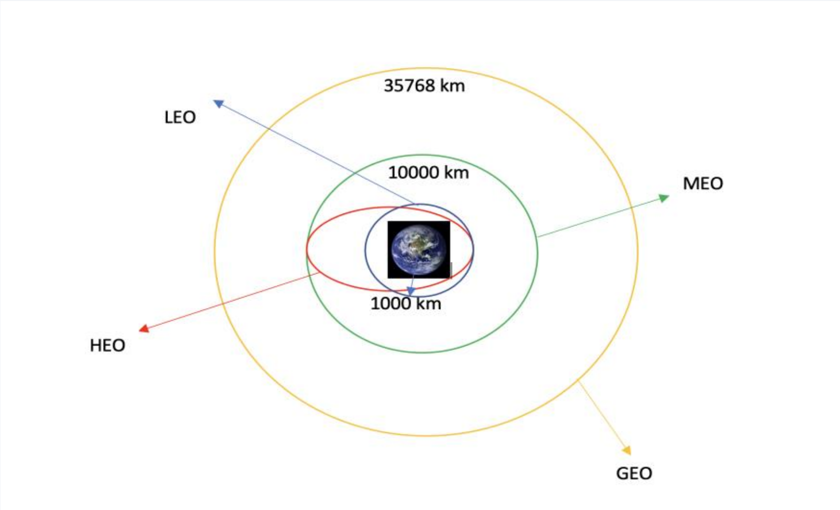
- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

- The data set has several cases where the launcher did not successfully land. We converted these outcomes to Training Labels, with 1 meaning the booster successfully landed and 0 means it was unsuccessful.

- Each launch has a dedicated orbit.

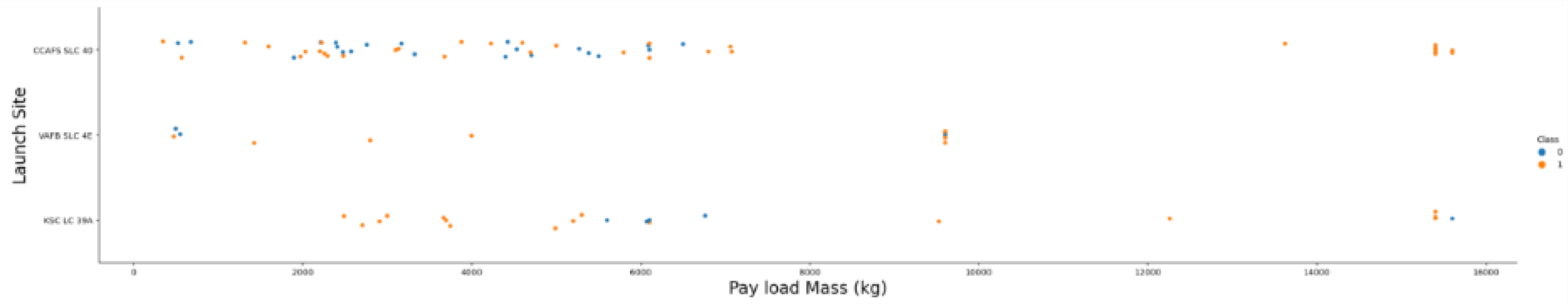


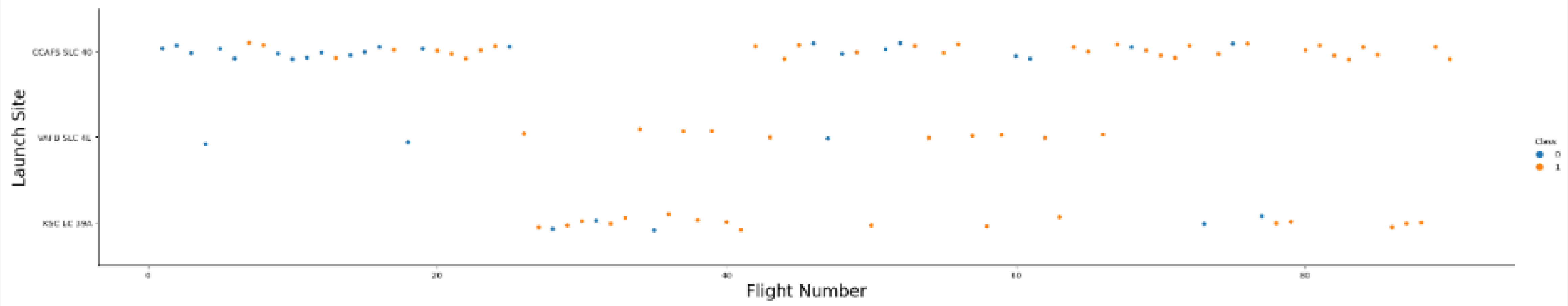
Exploratory Data Analysis EDA → Calculate the number of launches on each site → Calculate the number and occurrence of each orbit → Create a landing outcome label from Outcome column → Calculate success rate for all landings in the dataset

- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/Data%20wrangling.ipynb

EDA with Data Visualization

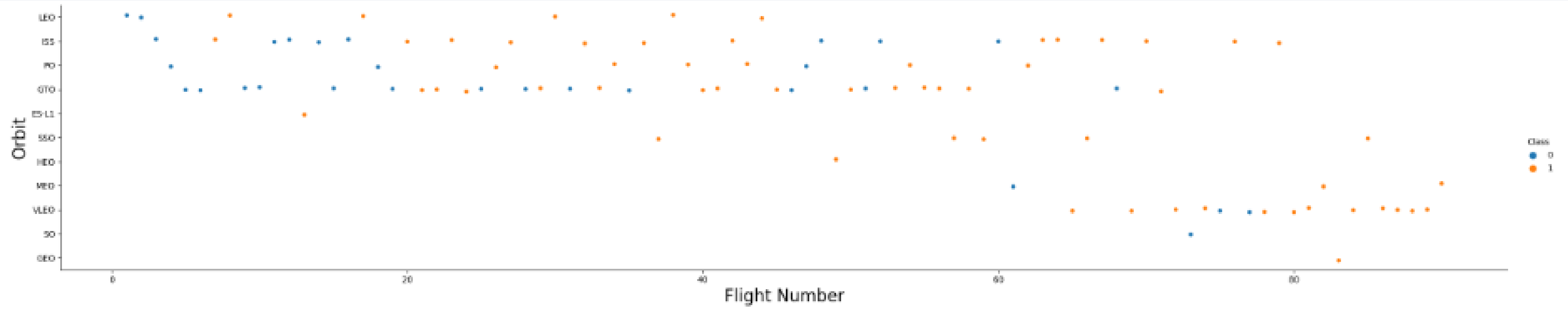


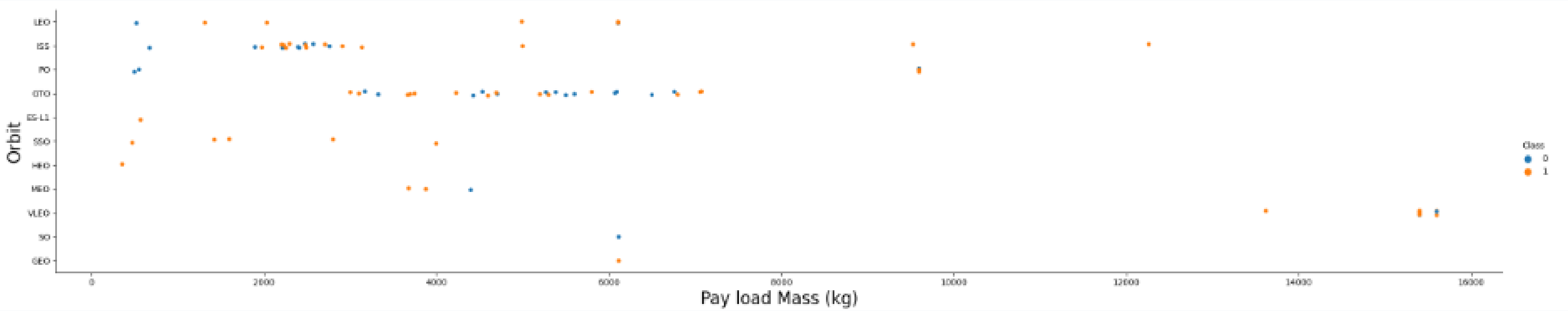


- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization (cont.)

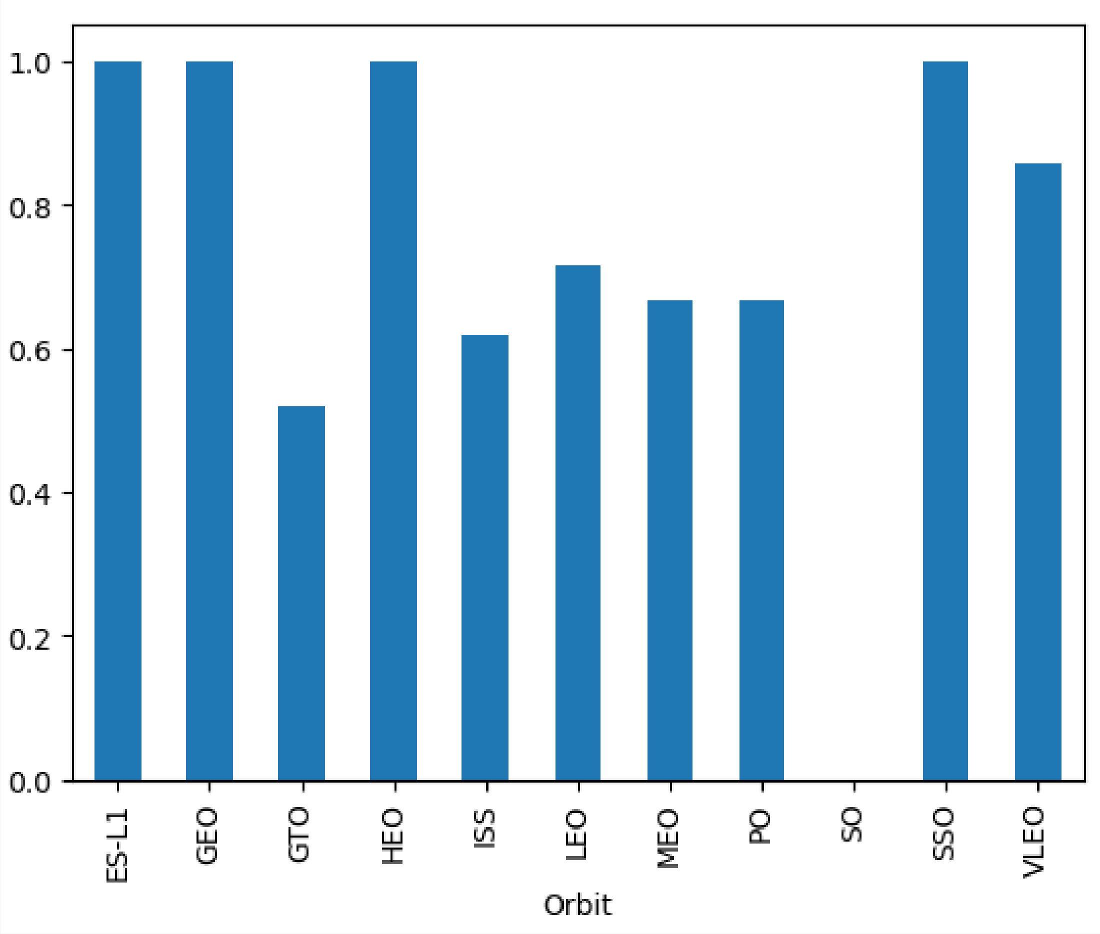


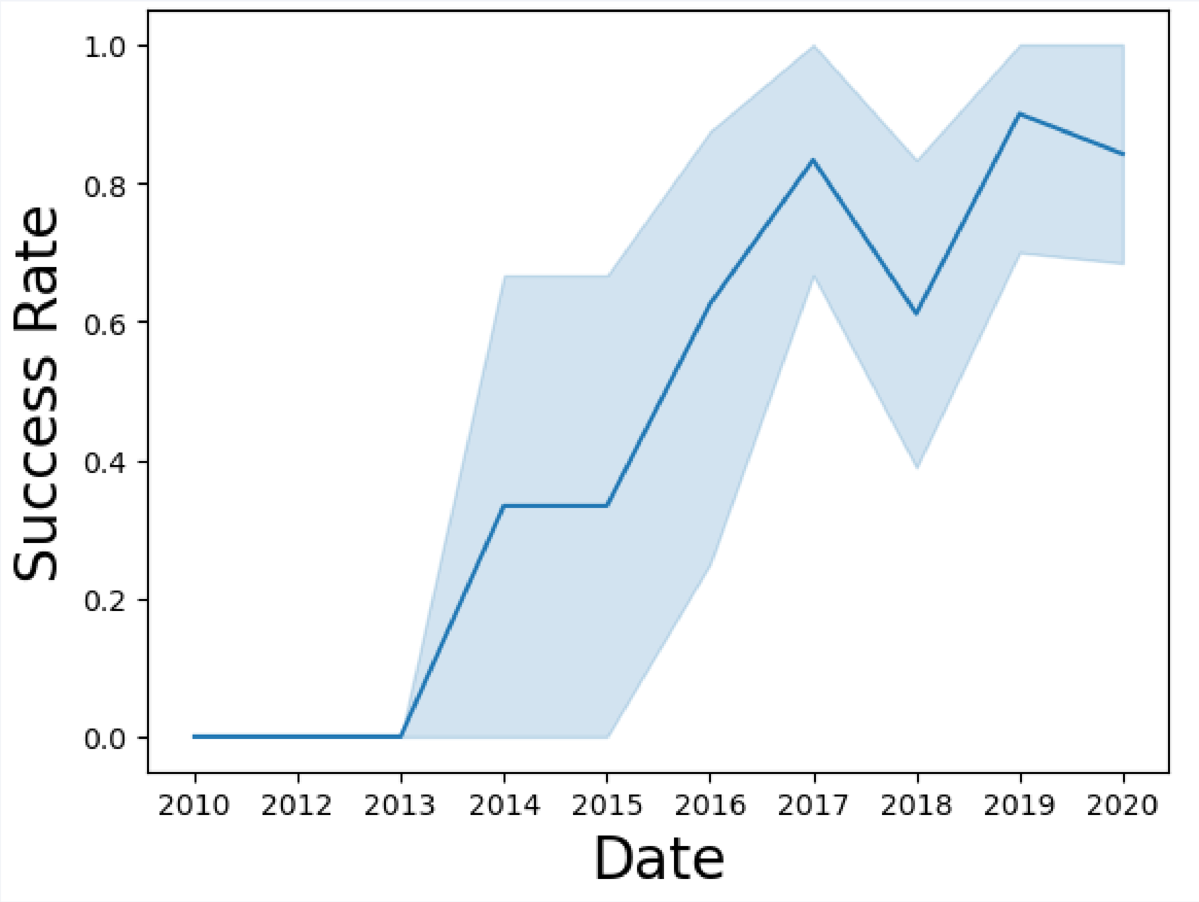


- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization (cont.)





- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-dataviz.ipynb

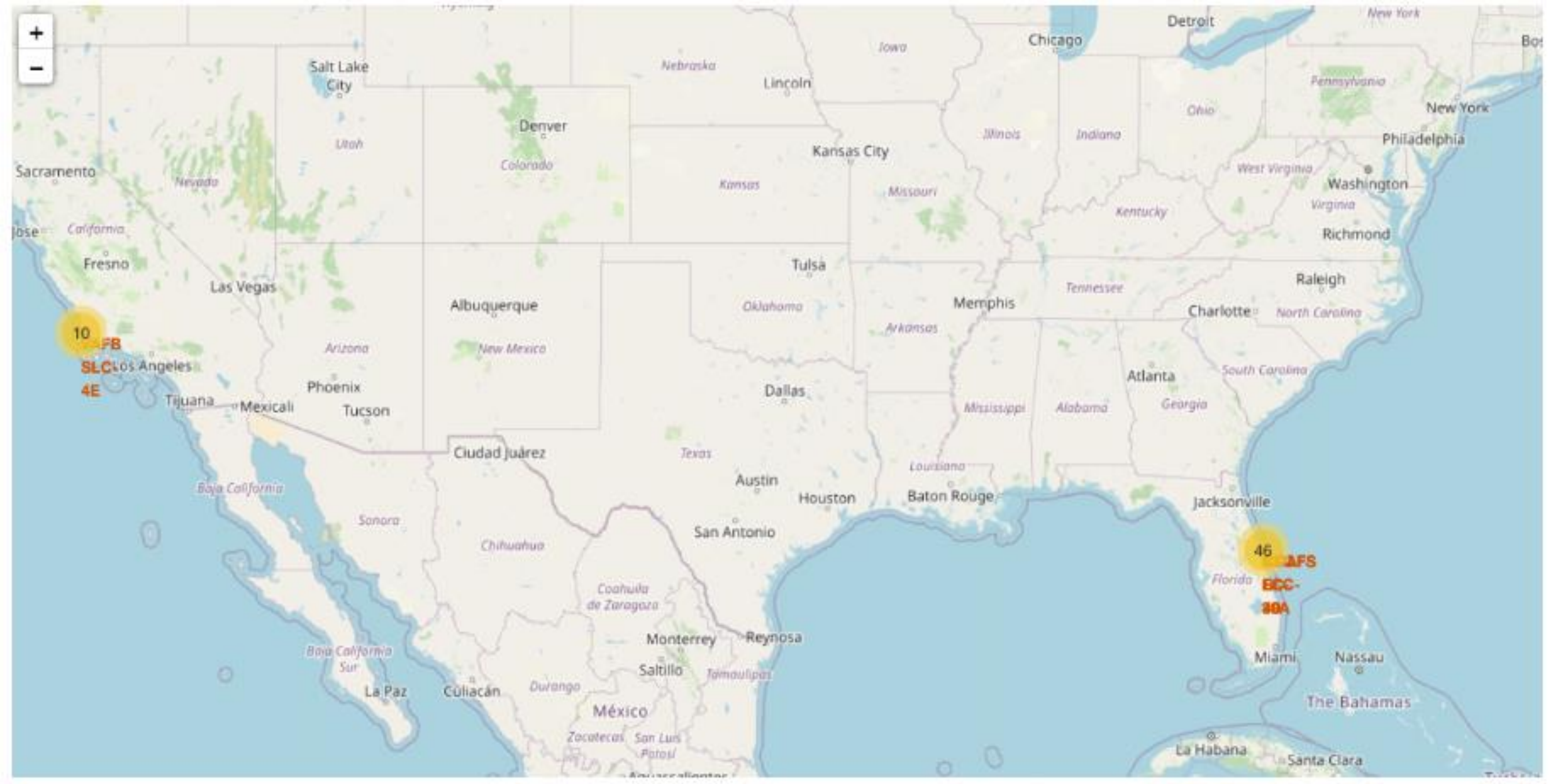
EDA with SQL

SQL queries were performed to gather information about the dataset.

- Task 1: Displaying the names of the unique launch sites in the space mission.
- Task 2: Displaying 5 records where launch sites begin with the string 'CCA'
- Task 3: Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Task 4: Displaying average payload mass carried by booster version F9 v1.1
- Task 5: Listing the date when the first successful landing outcome in ground pad was achieved.
- Task 6: Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Task 7: Listing the total number of successful and failure mission outcomes
- Task 8: Listing the names of the booster_versions which have carried the maximum payload mass
- Task 9: Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Task 10: Ranking the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- GitHub URL: https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium



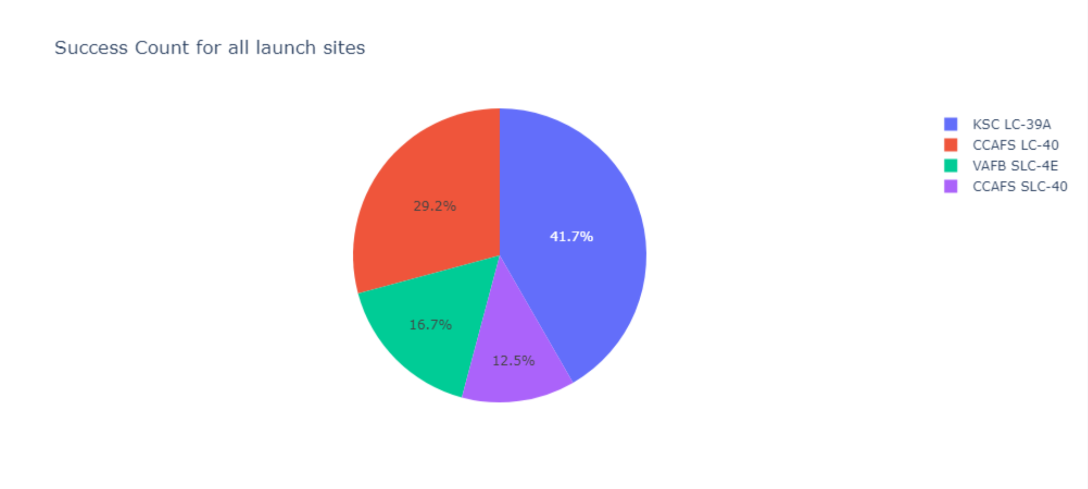
We added these objects to calculate the distances from different landmarks to discover various trends about the location of the launches.

To visualize the launch sites into an interactive map, we created a circle marker with a label at each launch site.

- GitHub URL

Build a Dashboard with Plotly Dash

- Built an interactive dashboard allowing the user to change different inputs.



KSC LC-39A had the most successful launches out of all sites

Interactive slider to allow the user to adjust payload range on the graph

Predictive Analysis (Classification)

Building The Model

- Load the data into dataframes
- Split the datasets into training and test data
- Set parameters for GridSearchCV object and fit into dataset

Evaluating The Model

- Look over the accuracy for every model using method "score"
- Plot the confusion matrix

Finding The Best Model

- The best performing model is the one with the best accuracy score.

- GitHub URL:

https://github.com/chaddiao/Applied-Data-Science-Capstone/blob/master/predictive%20analysis.ipynb

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

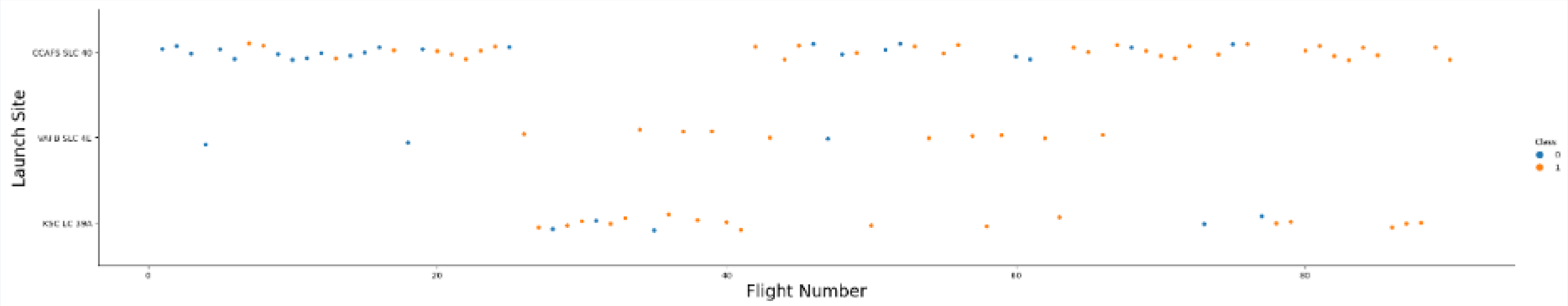
PREDICTIVE ANALYSIS RESULTS

Results

Section 2

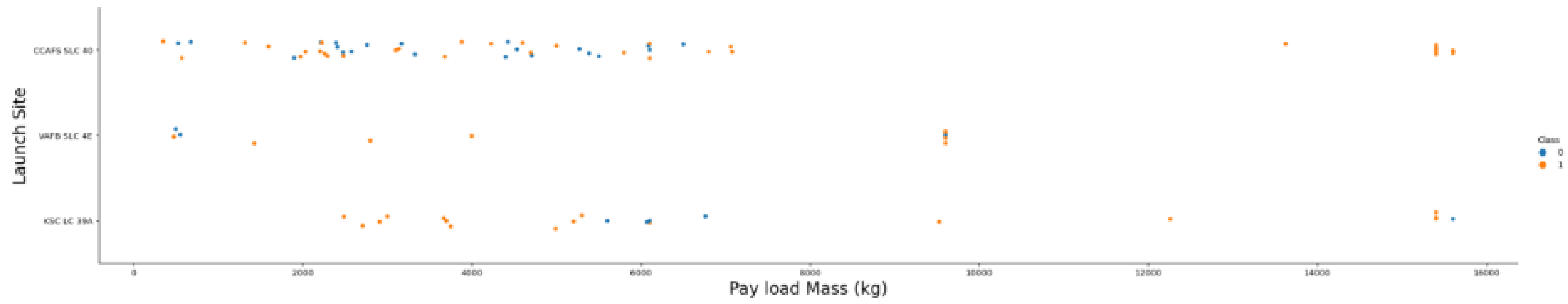
Insights drawn from EDA

Flight Number vs. Launch Site



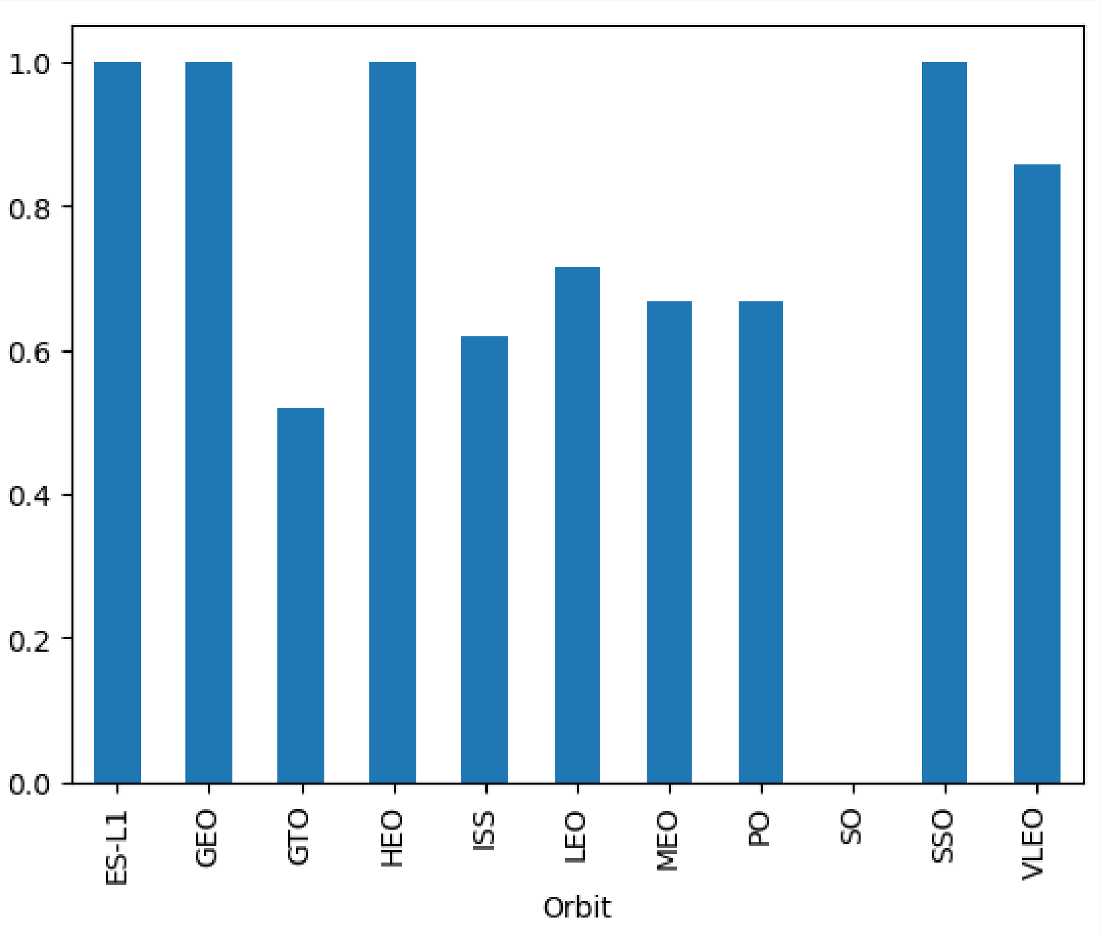
- This scatter plot shows that in general, the higher the flight number, the more success rate a launch will have.
- However, this trend seems nonexistent in site CCAFS SLC 40.

Payload vs. Launch Site



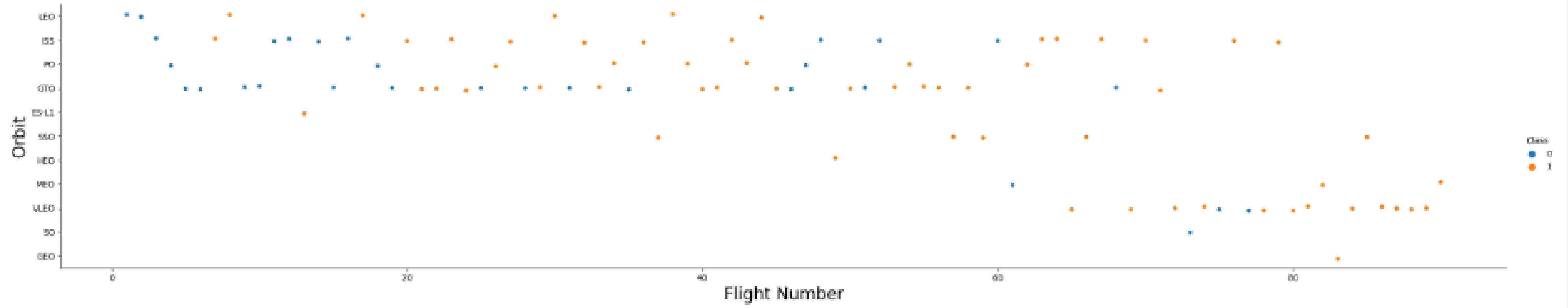
- Majority of payloads with mass 0kg – 6000kg were launched from the CCAFS SLC 40 site.
- Once the payload mass is over 8000kg, the success rate is greatly increased.

Success Rate vs. Orbit Type



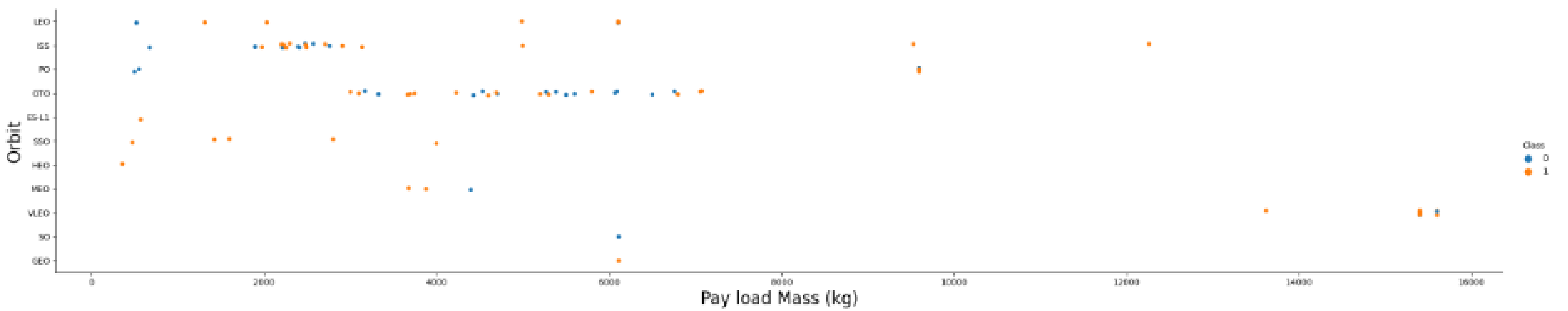
- Orbit types of ES-L1, GEO, HEO, and SSO have the highest success rates.

Flight Number vs. Orbit Type



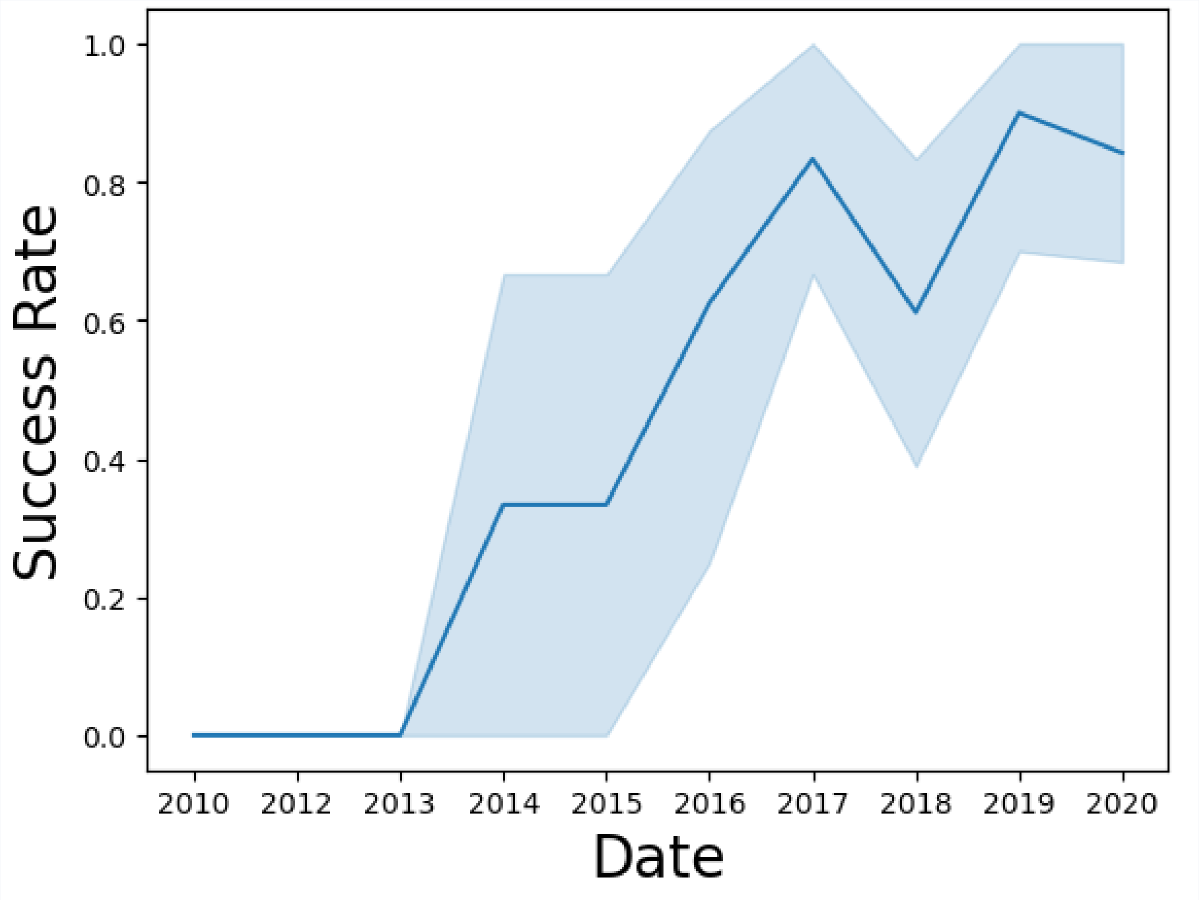
- In the LEO and VLEO orbits, success seems to be correlated to the number of flights, with more flights equaling high success.
- In the GTO orbit there seems to be no correlation between number of flights and success rate.

Payload vs. Orbit Type



- There is a heavy correlation between payload and the ISS orbit at around 2000kg-3000kg.
- On GTO orbits, higher payloads seem to be correlated with less success.

Launch Success Yearly Trend



- The success rate has been increasing steadily since 2013, all the way to 2017 likely due to technological advancements.

All Launch Site Names

- Used "Distinct" to display all unique launch sites from the SpaceX data table.

```
%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

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

Launch Site Names Begin with 'CCA'

- Used the query to display 5 records where launch sites begin with the string '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

- Using the function SUM(), all values in the column "PAYLOAD_MASS_KG_" were added.
- Total payload carried was 45596 KG.

```
%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

- Using the function AVG(), all values in the column "PAYLOAD_MASS_KG_" were averaged out.
- Average payload carried by booster version F9 v1.1 was 2928.4 KG.

```
%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

- Using the function MIN(), the earliest date was outputted.
- The date of the first successful landing outcome was 22nd of December, 2015.

```
%%sql
SELECT min(DATE)
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Used WHERE clause to filter the dataset with successful drone ship landings.

- Used AND clause to further filter payload mass to be in between 4000kg and 6000kg.

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237
ases.appdomain.cloud:32731/bludb
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 LIKE "%" to filter whether Mission_Outcome was successful or failure.

- 100 successes were found.
- 1 failure was found.

```
%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

- The booster that carried the maximum payload was determined using the WHERE clause and MAX() function.

```
%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

- Used WHERE clause and LIKE "%" to find booster versions, and their launch sites for the year 2015. Used AND to filter out only failed landing outcomes.

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.
databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- Used GROUP BY clause to group the landing outcomes in descending order with ORDER BY clause.

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

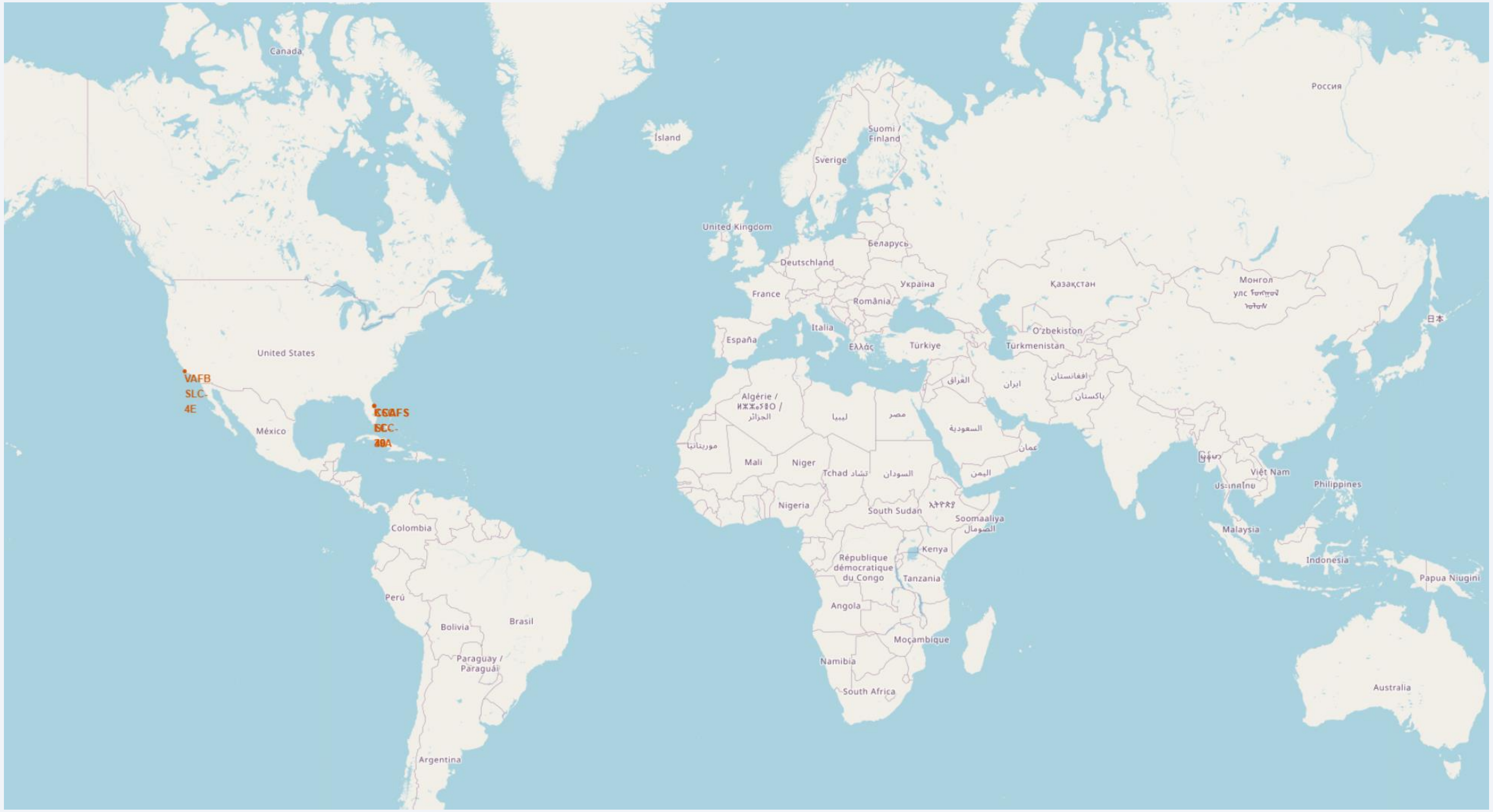
```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.c
loud:32731/bludb
Done.
```

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

Section 3

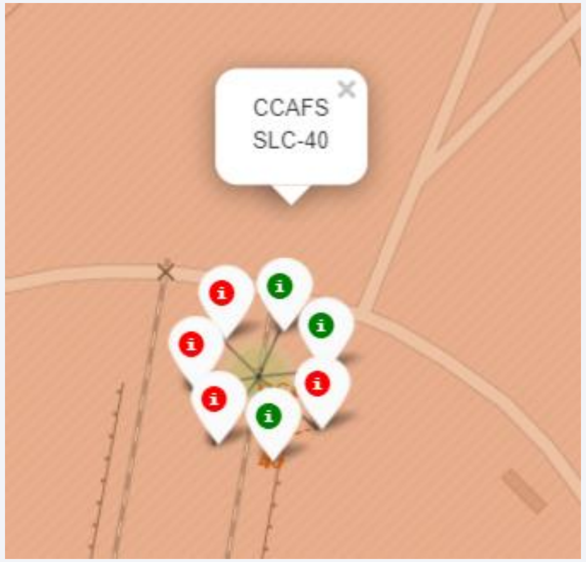
Launch Sites Proximities Analysis

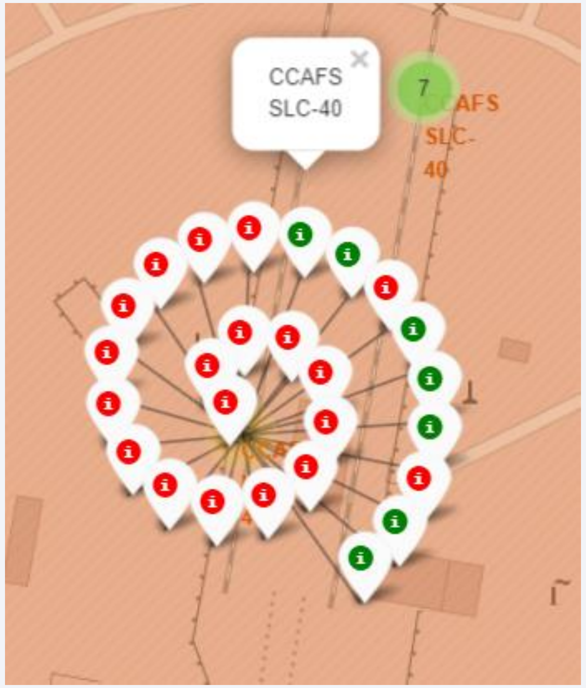
Global Launch Sites

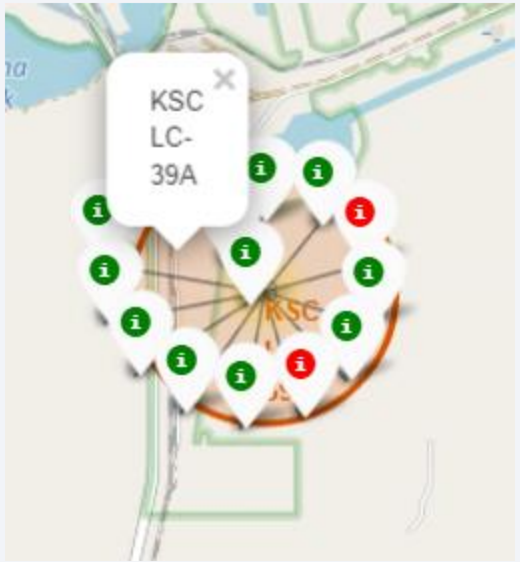


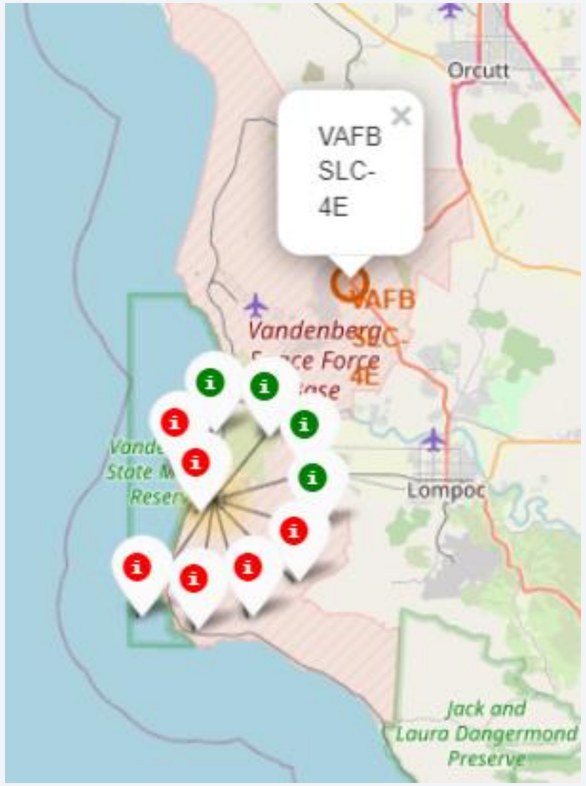
- Launch sites are in California and Florida

Landing Outcomes w/ Color Labelled Markers









- Green markers show successful landing outcomes.
- Red markers show failures.

Launch Site Distances to Landmarks

- Distance to coast = 0.9km

- Distance to major city = 78.45km

- Distance to closest highway = 29.21km
- Distance to closest railway station = 78.62km

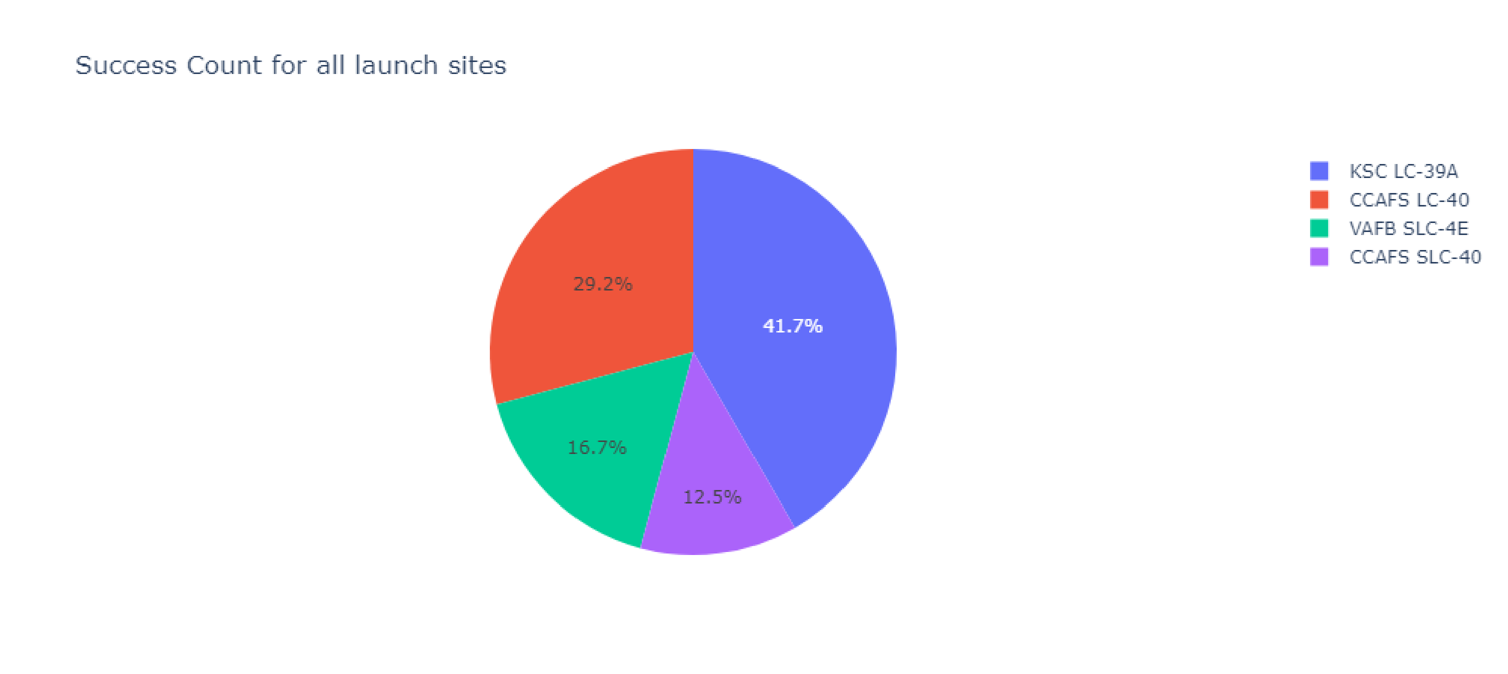
- In conclusion, launch sites are in close proximity to the coast, but keep their distance from major cities, highways, and railways.

Section 4

Build a Dashboard with Plotly Dash

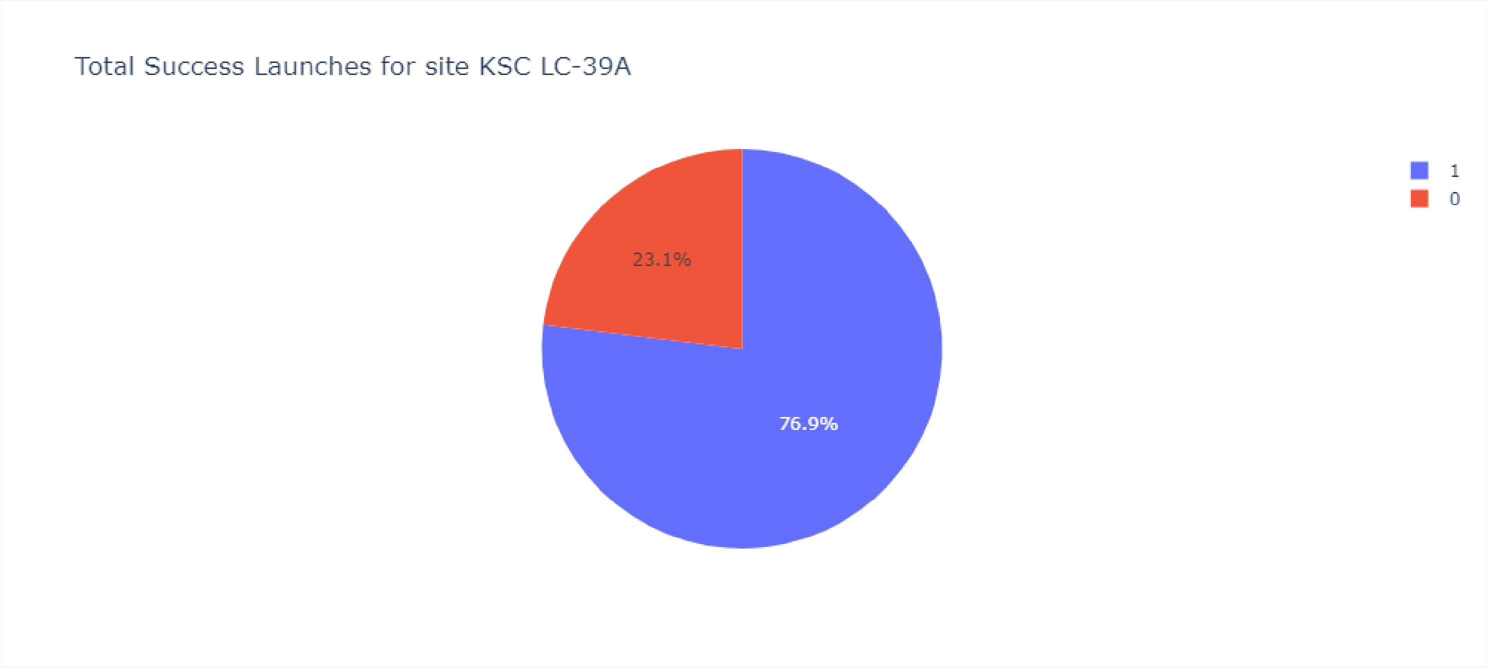
Launch Success Count For All Sites

- KSC LC-39A had the most successful launches out of all the sites.



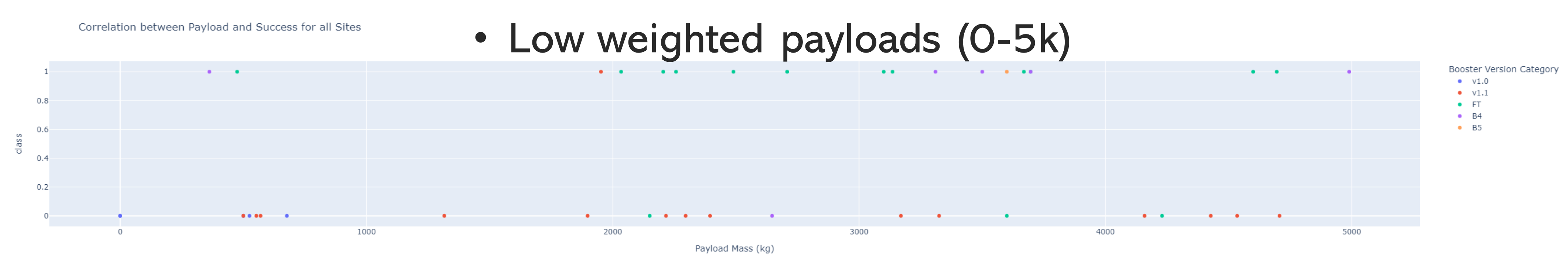
KSC LC-39A Success Ratio

- We can look further into the KSC LC-39A site, and see that 76.9% of total launches from that site were successful, while 23.1% were failures.

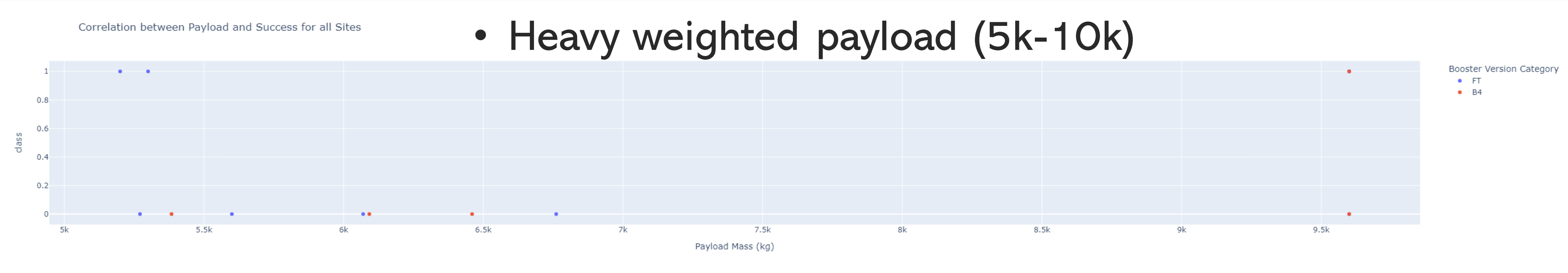


Payload v.s. Launch Outcomes

- Low weighted payloads (0-5k)



- Heavy weighted payload (5k-10k)



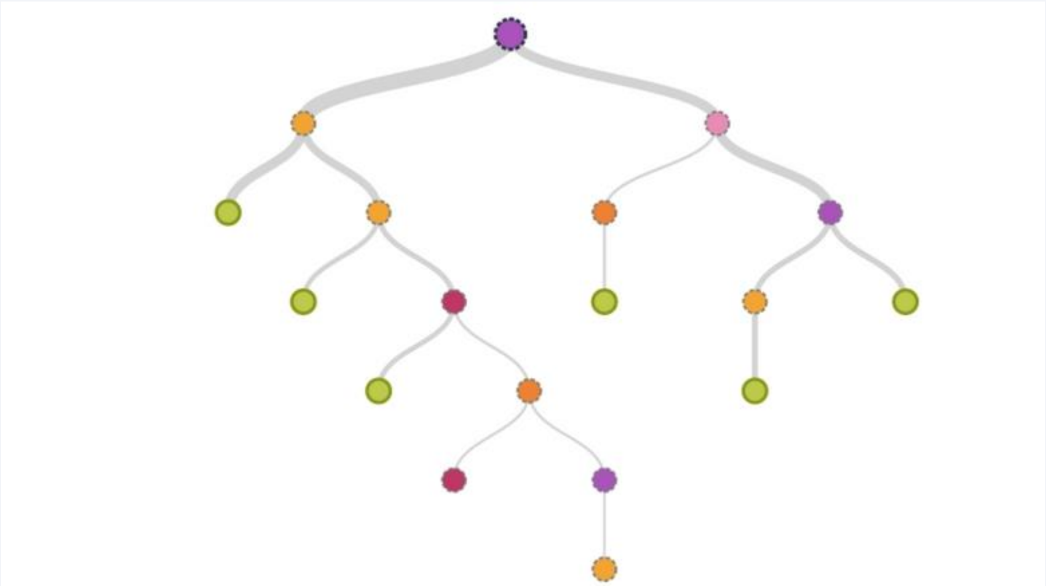
- Success rate is higher for lower weighted payloads (0-5k) than heavier weighted payloads (5k-10k).

Section 5

Predictive Analysis (Classification)

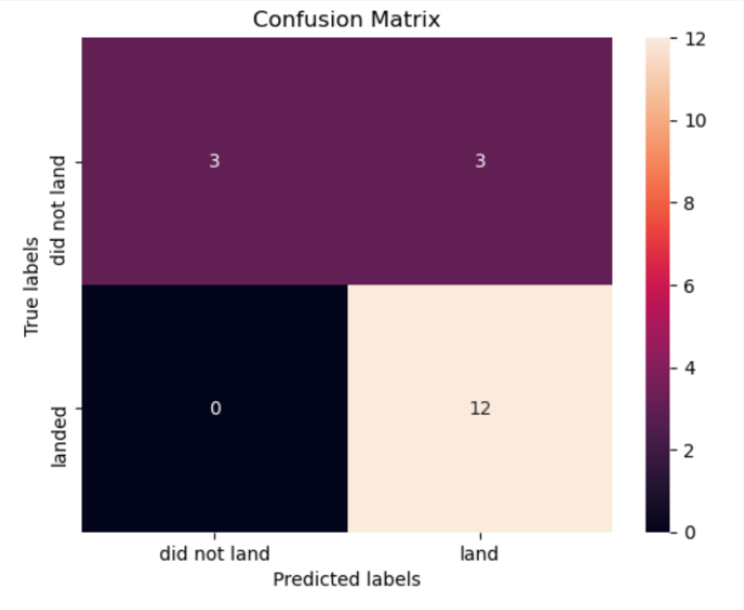
Classification Accuracy

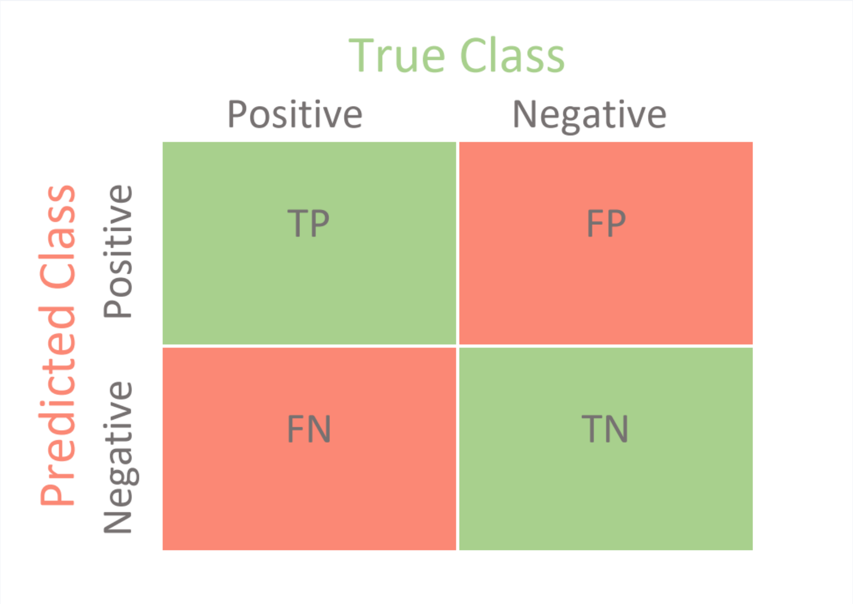
- Tree algorithm was identified as the best performing algorithm with the highest classification accuracy of 83.33% on the test data.



Confusion Matrix

- The confusion matrix shows that the classifier can distinguish between classes. The problem is that sometimes it marks an unsuccessful landing as successful by the classifier (false positive).





Conclusions

- Tree Classifier Algorithm is the best machine learning approach for this dataset.
- KSC LC-39A has the best success ratio out of all sites, with 76.9% of launches being successful.
- Lower weighted payloads have a higher success rate for all sites.
- The success rate has been increasing steadily since 2013 to 2017 likely due to technological advancements.
- Orbit types of ES-L1, GEO, HEO, and SSO have the highest success rates.

Thank you!