IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

John Leeman
November 2, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data collection using API, Web Scraping and Data Wrangling
  - ➢ Exploratory Data Analysis with SQL
  - ➢ Data Visualization with Folium
  - ➢ Machine Learning Predictive Analysis

- Summary of all results
  - ➢ Data Analysis results
  - ➢ Screenshots of Analytic Visualizations
  - ➢ Predictive Analysis results

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. We want to determine if the first stage will land, we can determine the cost of a launch. This information can then be used to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - ➢ What factors into a successful rocket landing?
  - ➢ What interactions of these factors help determine a successful landing?
  - ➢ What are the best operating conditions necessary to predict a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using SpaceX API and web scrapping Wikipedia
- Perform data wrangling
    - Data was processed using 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 using the response from the SpaceX Rest API

- The response content was decoded as JSON using .json() function call and turn it into a pandas dataframe using .json_normalize().

    - The data was then cleaned and missing values were accounted for.

- Web scraping was done against Wikipedia for Falcon 9 launch records using BeautifulSoup to extract launch records. The html was parsed and saved to a Pandas Dataframe which was used for analysis.

Data Collection – SpaceX API

- The SpaceX API was used to collect data. It was then cleaned and formed.
- Json_normalize() was used to convert data to a dataframe to be used in analysis.
- GitHub URL:
  - [CapstoneRepo/Applied Data Science Capstone Notebook.ipynb at main · jleeman22/CapstoneRepo (github.com)]

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
myDataFrame = pd.json_normalize(response.json())
```

In [33]:
```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = myDataFrame[['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 rows that have multiple payloads in a single rocket.
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 the feature.
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)]
data.head()
```

In [47]:
```
# Calculate the mean value of PayloadMass column
payload_mean_value=data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value=payload_mean_value, inplace=True)
result_df = data_falcon9
result_df.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

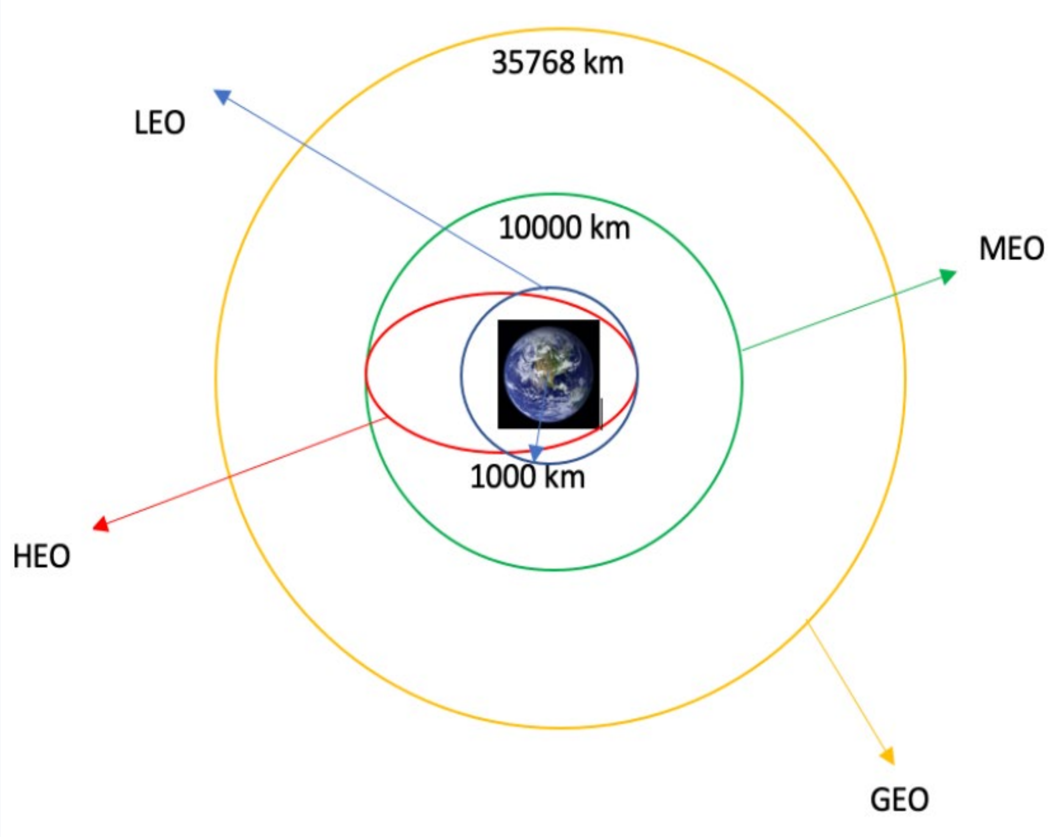
- Request Falcon9 launch data from Wikipedia URL
- Use BeautifulSoup on the response text
- Extract the data from the html
- [CapstoneRepo/Capstone Web Scrapping.ipynb at main · jleeman22/CapstoneRepo (github.com)]()

```
In [9]: # use requests.get() method with the provided static_url
        # assign the response to a object
        x = requests.get(static_url).text

In [10]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         from bs4 import BeautifulSoup
         soup = BeautifulSoup(x, 'html.parser')

In [18]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
            # get table row
             for rows in table.find_all("tr"):
                 #check to see if first table heading is as number corresponding to launch a number
                 if rows.th:
                     if rows.th.string:
                         flight_number=rows.th.string.strip()
                         flag=flight_number.isdigit()
                 else:
                     flag=False
                 #get table element
                 row=rows.find_all('td')
                 #if it is number save cells in a dictonary
                 if flag:
                     extracted_row += 1
                     # Flight Number value
                     # TODO: Append the flight_number into launch_dict with key `Flight No.`
                     #print(flight_number)
                     launch_dict['Flight No.'].append(flight_number)
                     datatimelist=date_time(row[0])
```

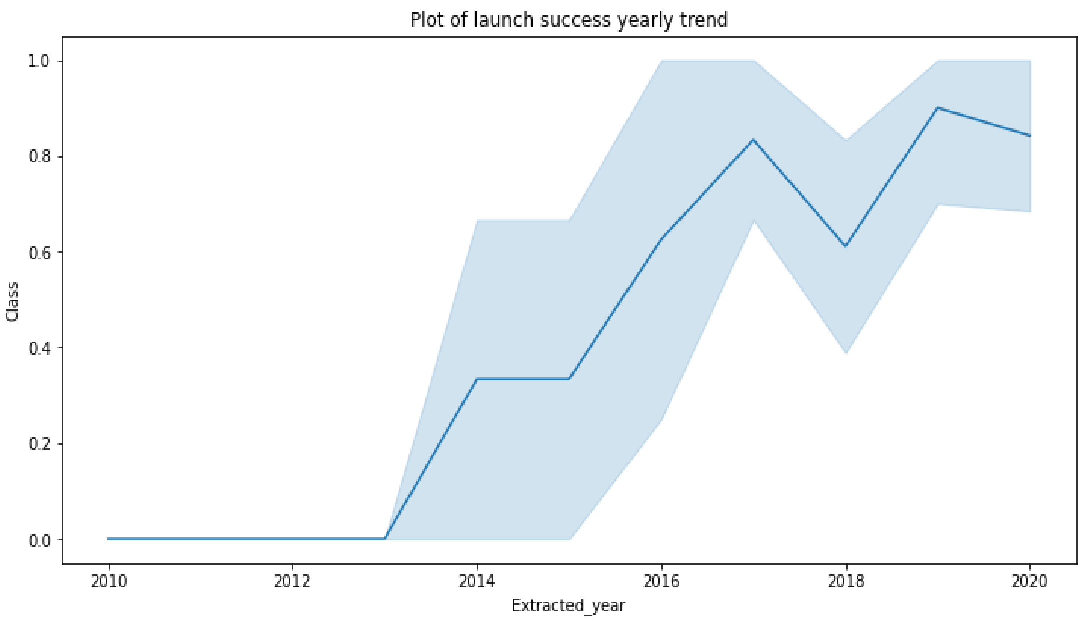
Data Wrangling

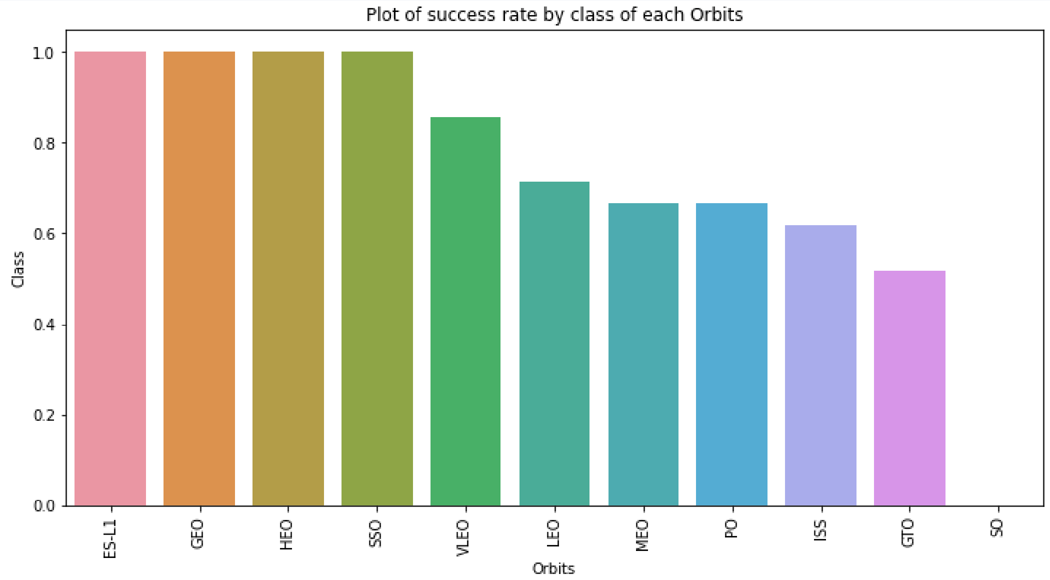


- Data wrangling is the process of cleaning data to facilitate it's use in Exploratory Data Analysis (EDA).
- The number of launches at each site was calculated, as well as the number and occurrences of each orbit type.
- We then created a landing outcome label from outcome column for additional analysis and machine learning. The results were exported to a CSV file.
- [CapstoneRepo/Data Wrangling EDA.ipynb at main · jleeman22/CapstoneRepo (github.com)]

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly





- [CapstoneRepo/EDA with Visualization lab.ipynb at main · jleeman22/CapstoneRepo (github.com)]

11

EDA with SQL

- Used EDA with SQL to get a better understanding of the dataset. Queries were written to find:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS).
  - The average payload mass carried by booster version F9 v1.1.
  - The total number of successful and failure mission outcomes.
  - The failed landing outcomes in drone ship, their booster version and launch site names.
  - The ranks of the landing outcome counts or successes for given periods of time.
- CapstoneRepo/EDA with SQL Lab.ipynb at main · jleeman22/CapstoneRepo (github.com)

Build an Interactive Map with Folium

- We wanted to visualize the launch data in an interactive folium map with marked launch sites, mapped objects such as markers, circles, lines to indicate the success or failure of launches.
- We assigned the dataframe launch outcomes (failure or success) to class 0 and 1 (0 for failure 1 for success) with red and green markers on the map in MapCluster().
- We calculated the distances between a launch site to its proximities to find answers to some questions:
  - Are launch sites near railways, highways and coastlines?
  - Are launch sites close to nearby cities?
- CapstoneRepo/Visual Analytics with Folium lab.ipynb at main · jleeman22/CapstoneRepo (github.com)

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash which allowed users to select different inputs.
- Plotted pie charts showing the total launches by a certain sites.
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

- GitHub URL:CapstoneRepo/EDA with Visualization lab.ipynb at main · jleeman22/CapstoneRepo (github.com)

Predictive Analysis (Classification)

- Building the model
  - Loaded the data using numpy and pandas
  - Transformed the data and split it into training and testing
  - Set parameters and algorithms to tune using GridSearchCV
- Evaluate the model
  - Check accuracy of each model
  - Get tuned hyperparameters for each type of algorithms
- Improve the model
  - Use feature engineering and algorithm tuning
- Find the best performing classification model
- GitHub URL:CapstoneRepo/Machine Learning Prediction lab.ipynb at main · jleeman22/CapstoneRepo (github.com)

Results

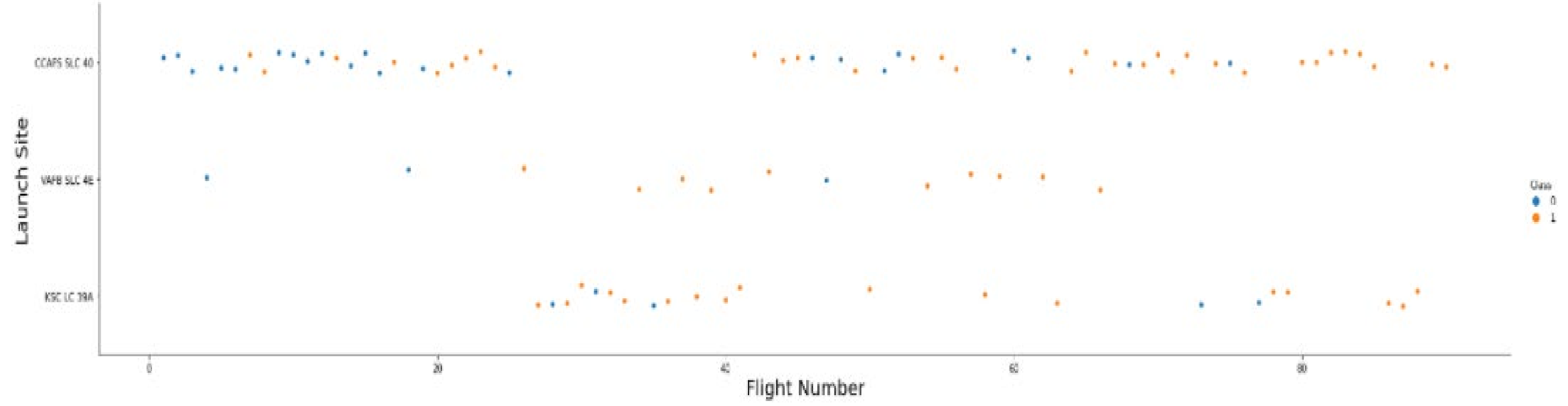
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

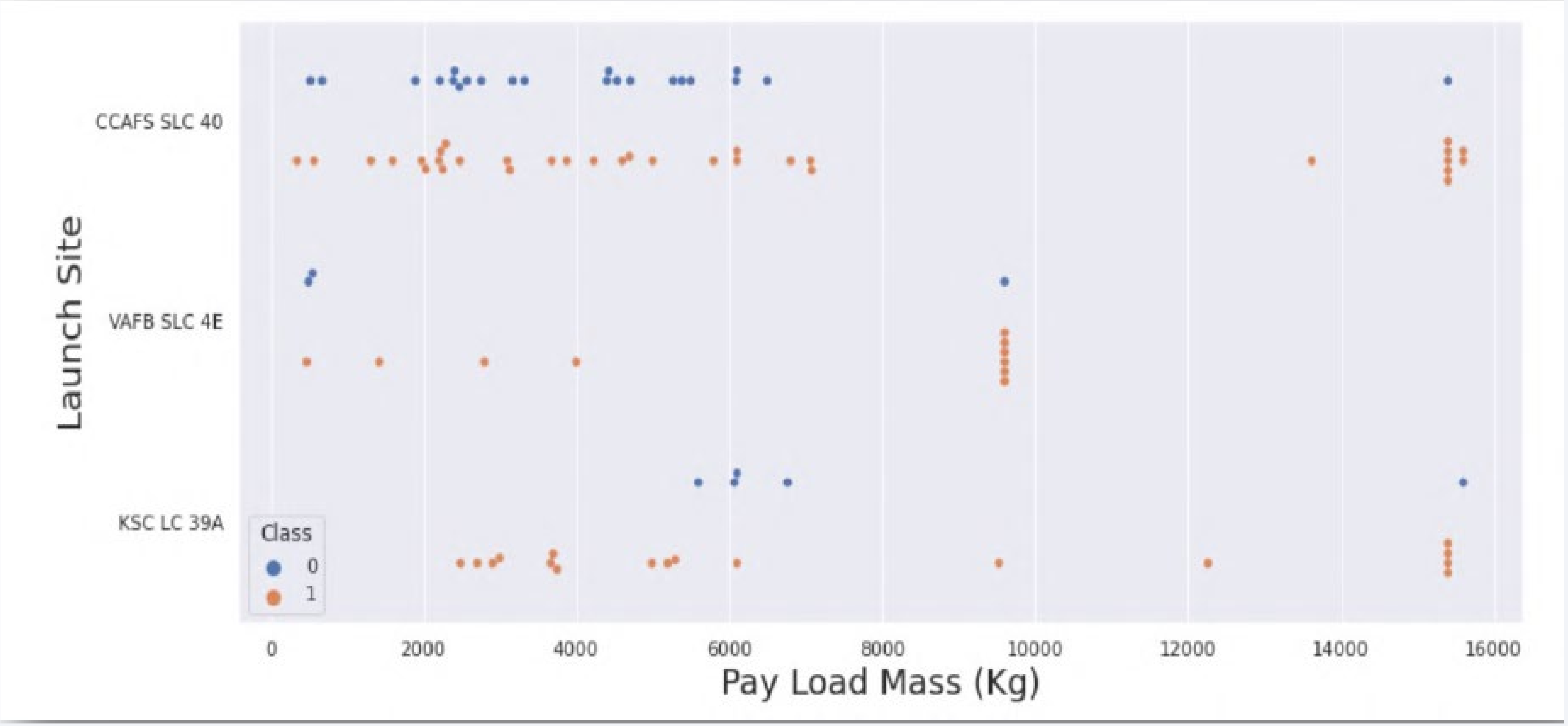
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate will be.



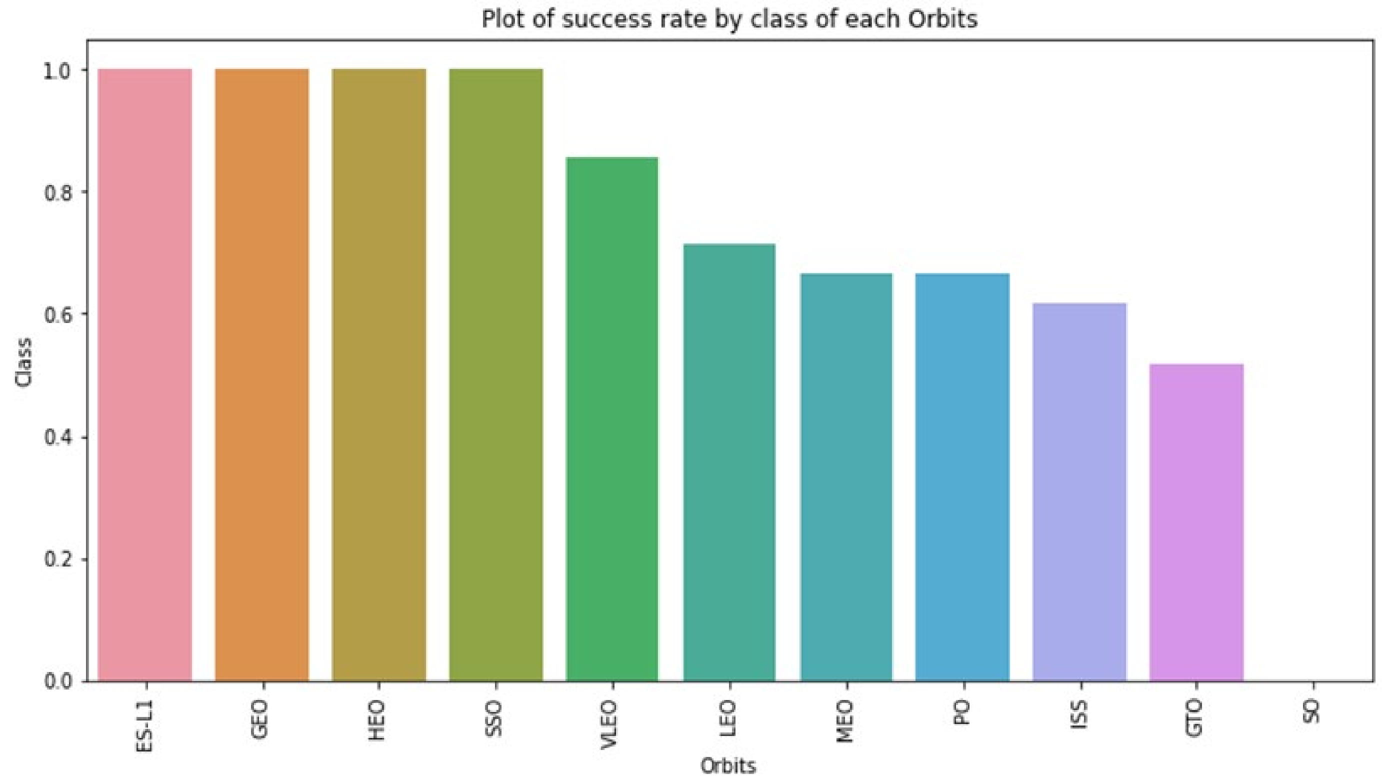
Payload vs. Launch Site

- The scatter plot shows that if the payload is greater, the rate of success increases drastically.



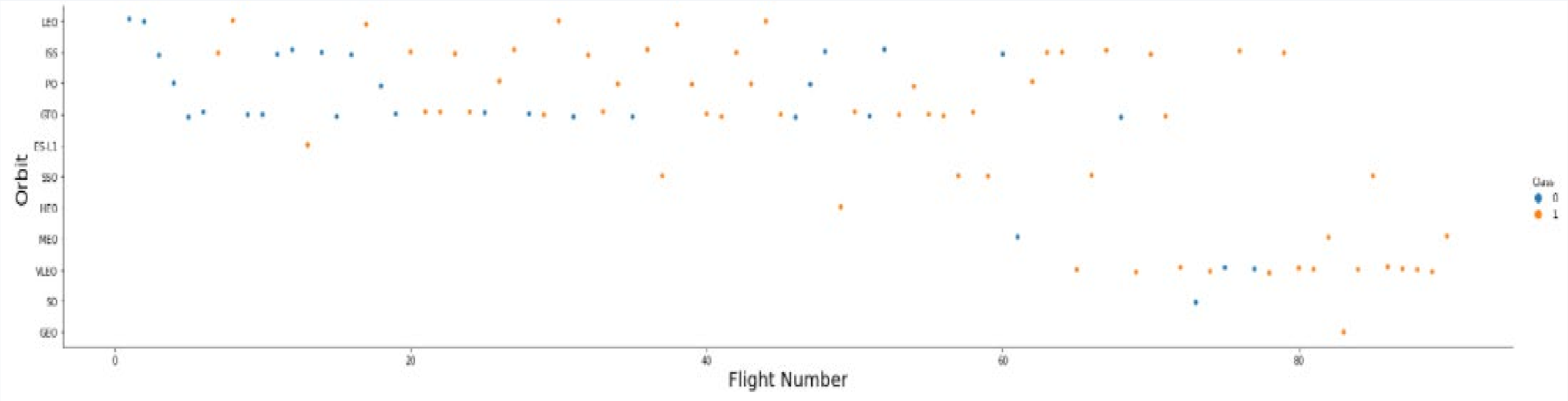
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



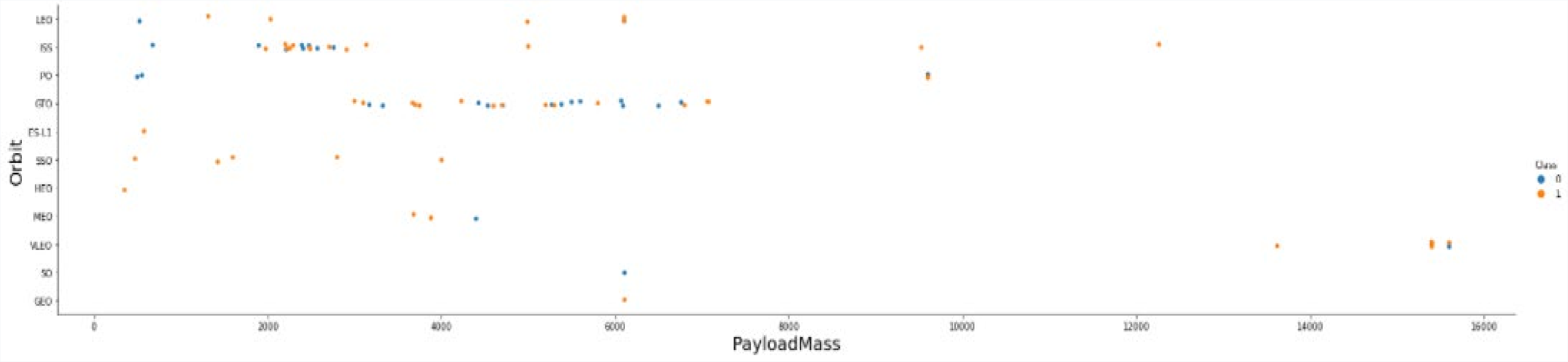
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



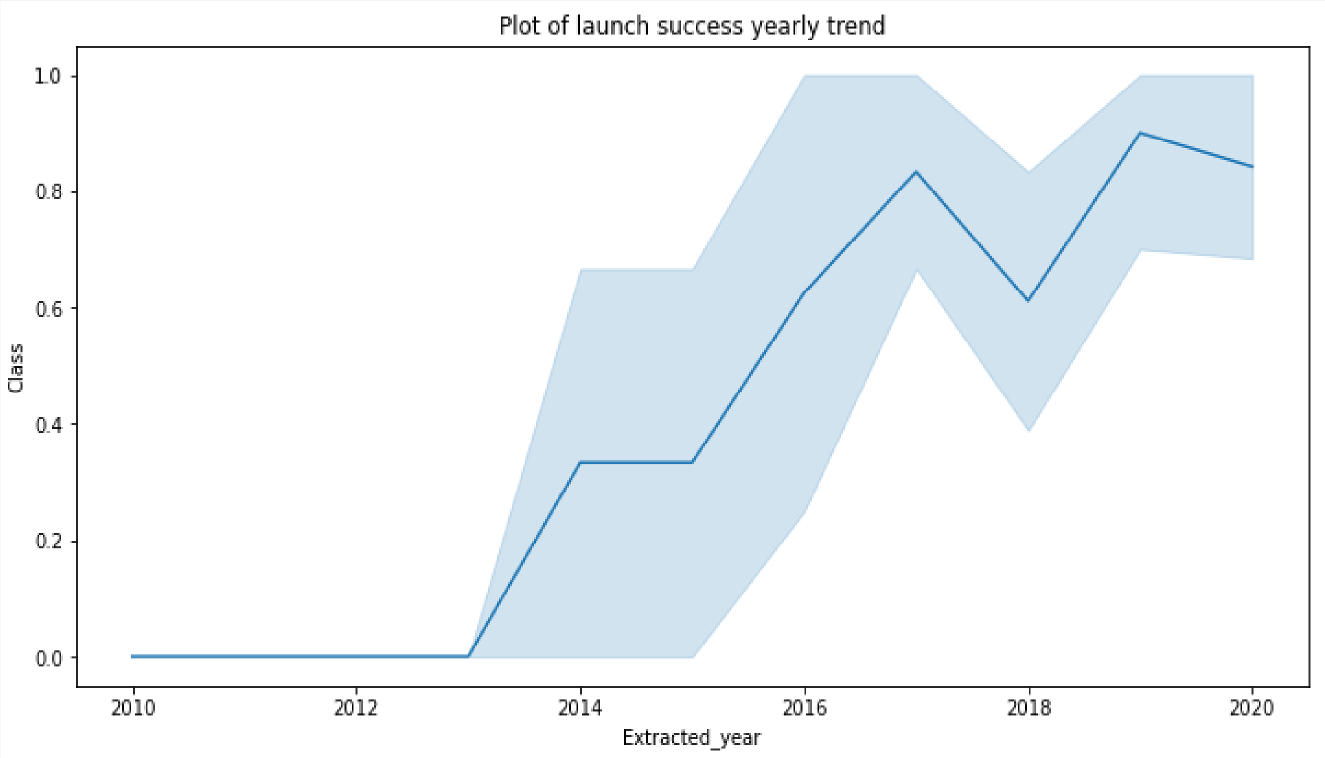
Payload vs. Orbit Type

- We see that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- Observing the plot, we see that success rate from 2013 kept on increasing until 2020



All Launch Site Names

- Use the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACE_X_TBL
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA

```
%sql SELECT * FROM SPACE_X_TBL where LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACE_X_TBL WHERE CUSTOMER = 'NASA (CRS)'
```

1
45596

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACE_X_TBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

1
2928

First Successful Ground Landing Date

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

```
%sql SELECT MIN(DATE) FROM SPACE_X_TBL WHERE Landing__Outcome = 'Success (ground pad)'
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- 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 SPACE_X_TBL WHERE Landing__Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) FROM SPACE_X_TBL WHERE MISSION_OUTCOME = 'Success' OR  MISSION_OUTCOME = 'Failure (in flight)'
```

1
100

Boosters Carried Maximum Payload

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

```
%sql SELECT BOOSTER_VERSION FROM SPACE_X_TBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACE_X_TBL) ORDER BY BOOSTER_VERSION
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

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

```
%sql SELECT Landing__Outcome, BOOSTER_VERSION, Launch_Site FROM SPACE_X_TBL WHERE Landing__Outcome = 'Failure (drone ship)' AND (DATE BETWEEN '2015-01
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
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

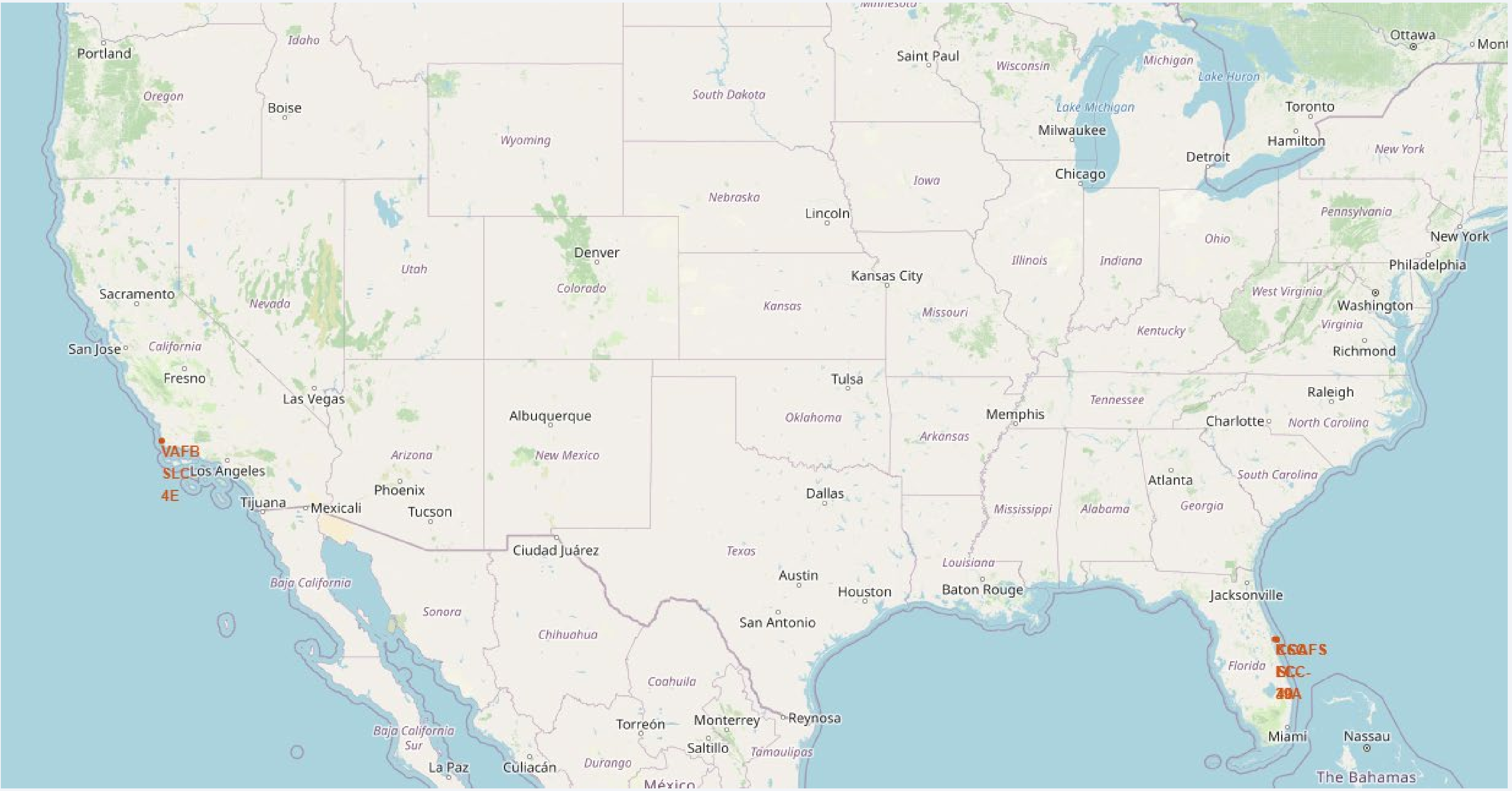
```
%sql SELECT Landing__Outcome, COUNT(Landing__Outcome) FROM SPACE_X_TBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing__Outcome ORDE
```

landing__outcome	2
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

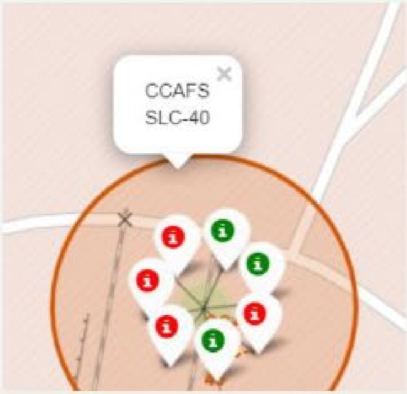
Launch Sites Proximities Analysis

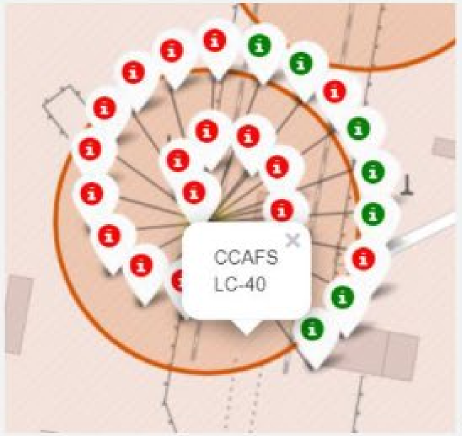
SpaceX Launch Site Locations

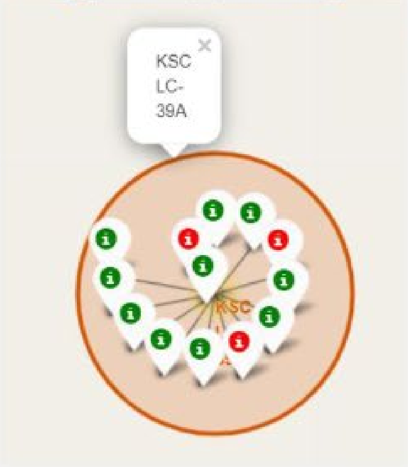


The map shows that all of the SpaceX launch sites are located on the coasts in the United States.

Markers Showing Launch Sites with Colored Labels







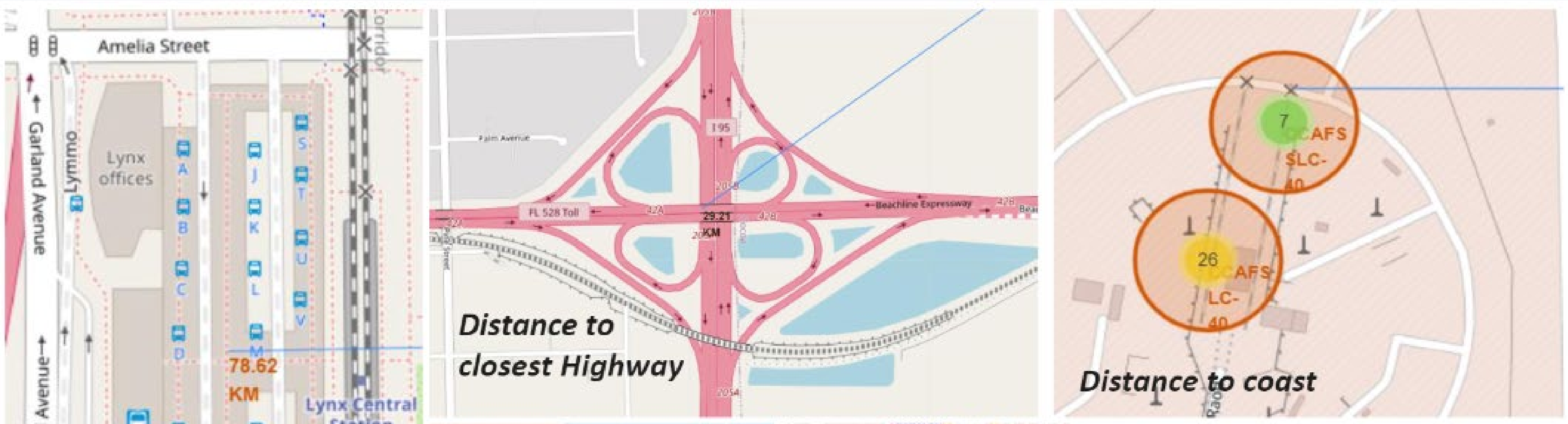
Florida Launch Sites

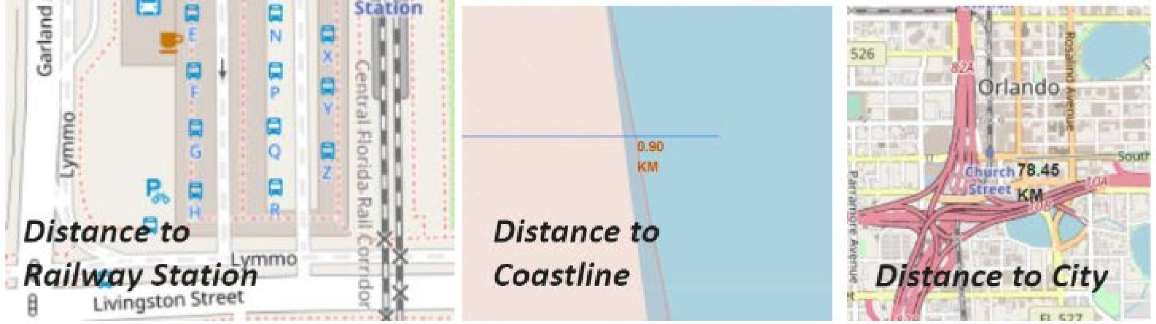
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site’s Distance to Landmarks



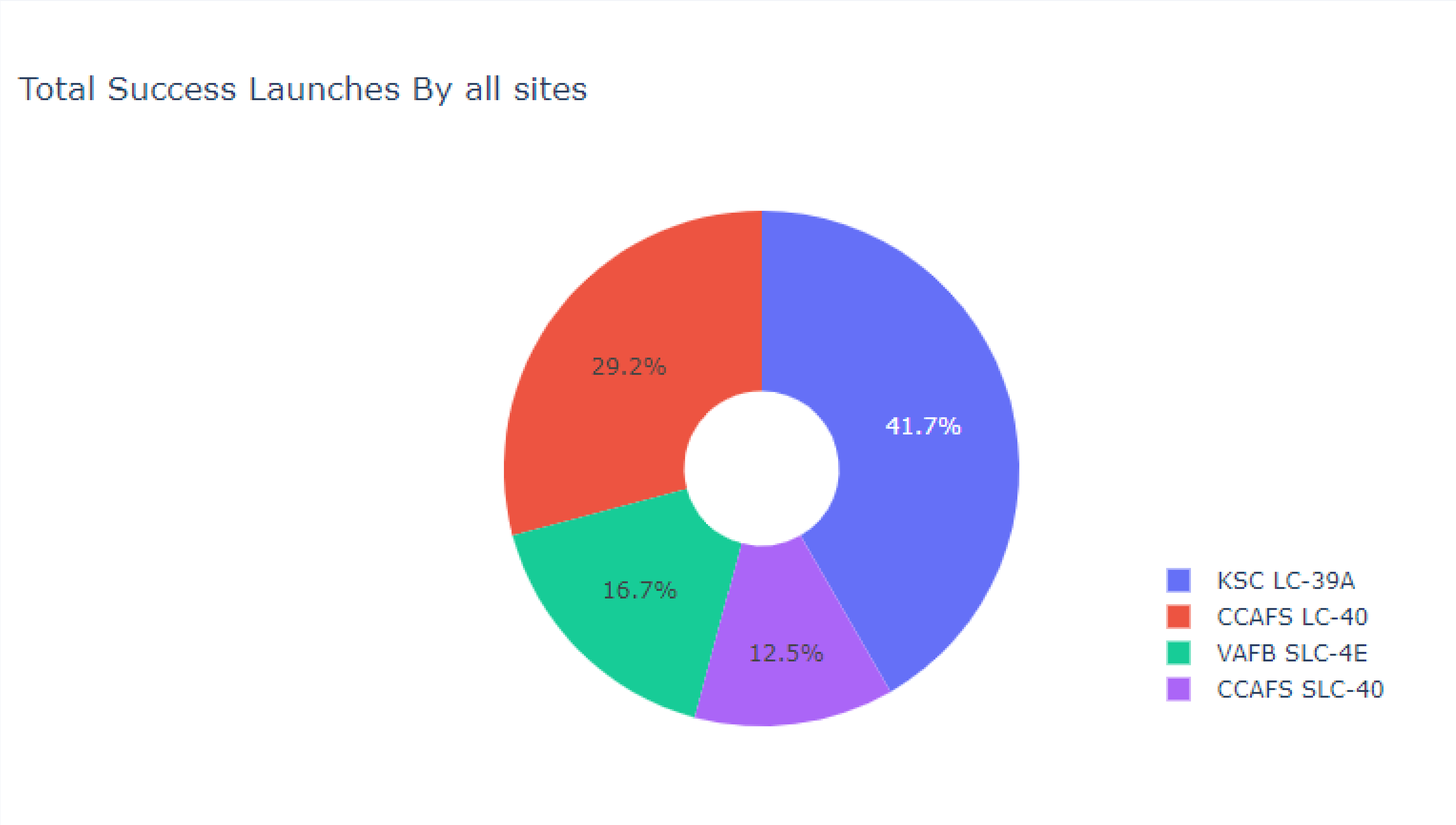


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

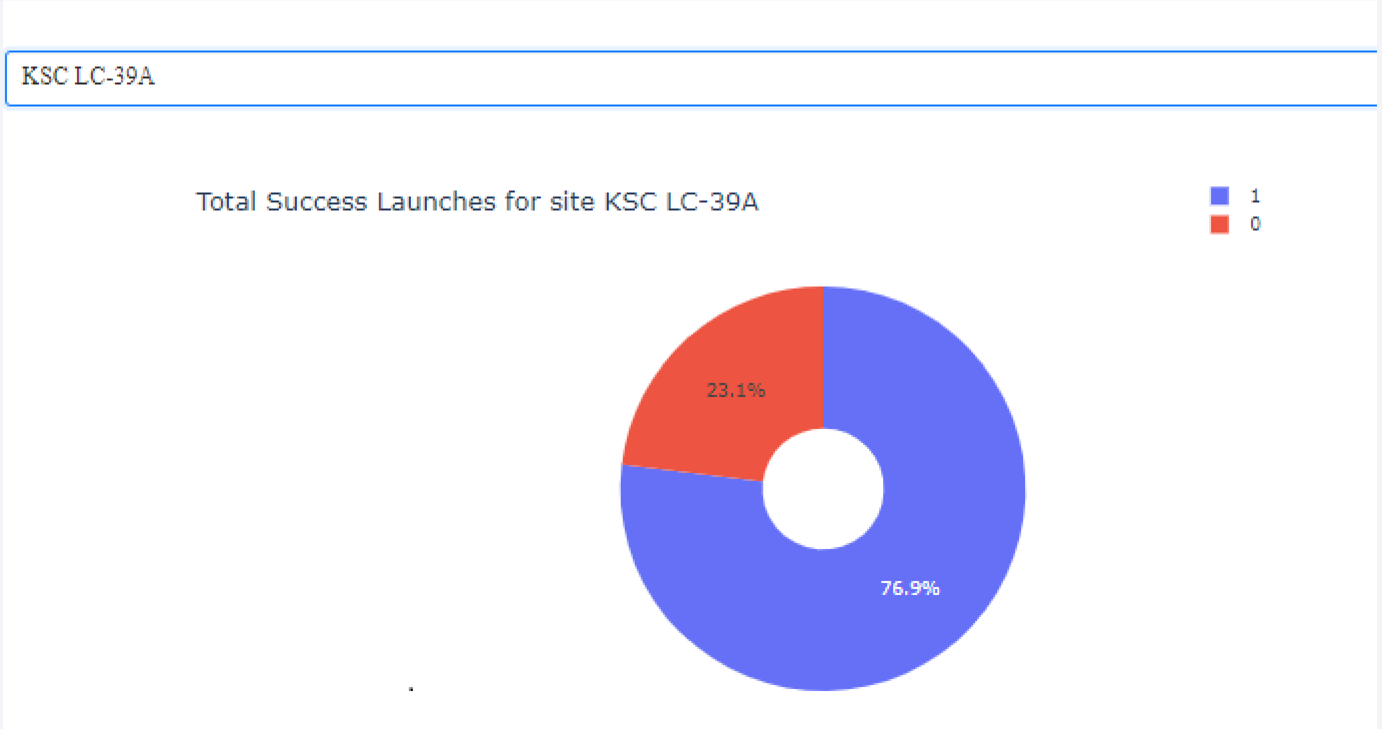
Build a Dashboard with Plotly Dash

Success Percentage of Each Launch Site



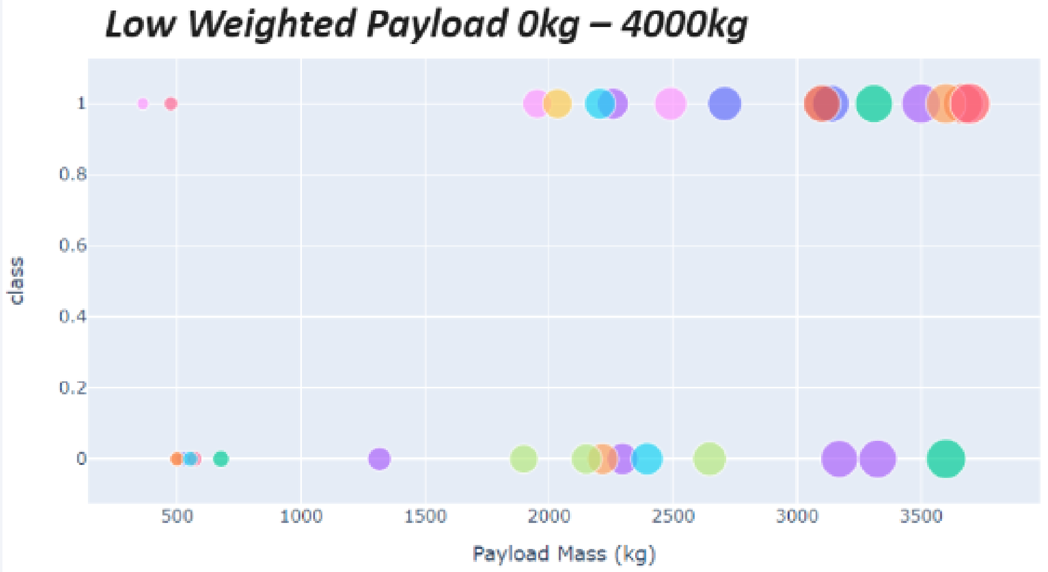
We see that KSC LC-39A had the most successful launches of all of the sites.

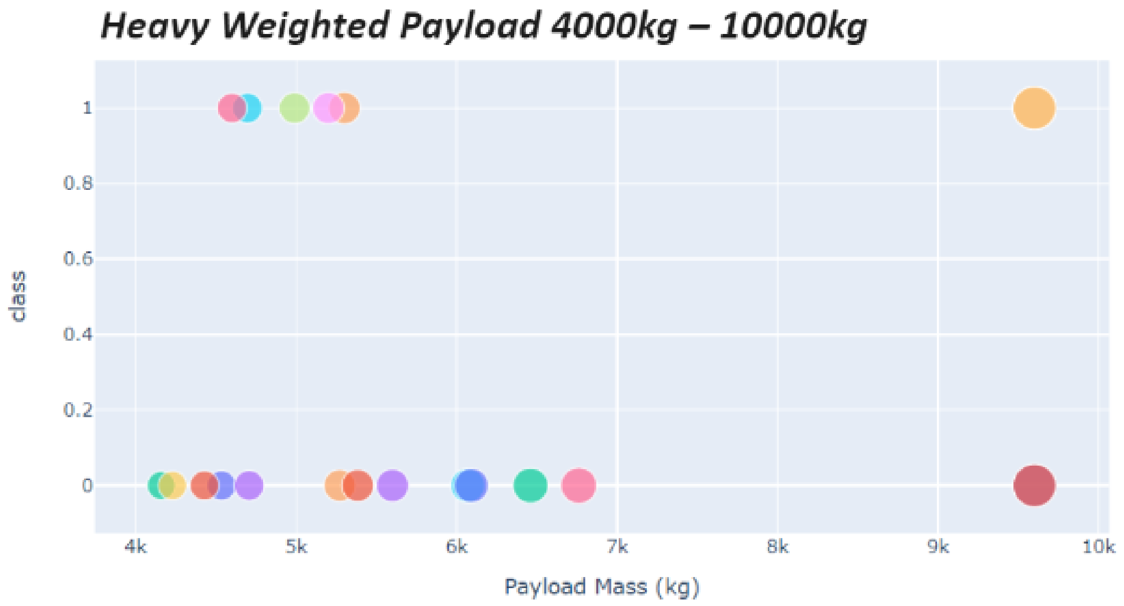
Launch Site KSC-39A has the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate and a 23.1% failure rate

Payload vs. Launch Outcome Scatter Plot





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

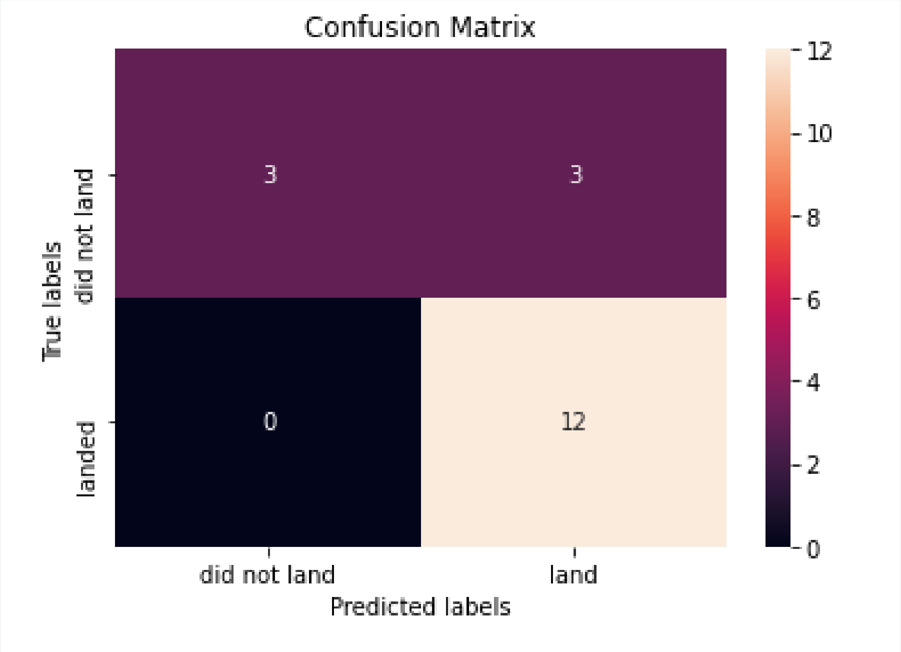
- The decision tree classifier is the model with the highest classification accuracy

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix



➢ The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

We can conclude that:

- The larger the amount of flights at a launch site, the greater the success rate at a launch site.
- Launch success rate for SpaceX launches increased from 2013 until 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites: 76.9%.
- The Decision tree classifier was identified as the best machine learning algorithm for this task/dataset.

Thank you!