Winning Space Race with Data Science

Ray Liang
6/3/24

Outline

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

Executive Summary

Summary of Methodologies

- Predict success of Falcon 9 first stage landing through web scraping API
- Understand factors contributing to successful landings through data transformation
- Estimate Overall Cost of Launch
- Implement Machine Learning Techniques

Summary of all Results

- Develop Predictive model for likelihood of success
- Exploratory Data Analysis and Predictive Analysis Results

Introduction

Project background and context

- SpaceX lists its Falcon 9 rocket launches on its website with a price tag of $62 million, while competitors charge over $165 million per launch, largely due to SpaceX's ability to recover and reuse the first stage. Ascertaining whether the first stage will successfully land can help predict the total launch cost, which is valuable data for any rival firm considering a competitive bid against SpaceX.

Problems you want to find answers

- Optimal methods for predicting first-stage landing success.
- Where are the Ideal launch locations.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Through SpaceX API and Web-scraping 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
  - Initially, we loaded the dataset and divided it into features (X) and targets (Y). Following this, we normalized the feature set (X) to prepare it for analysis. For the classification task, we applied various algorithms, and for each technique, we fine-tuned the parameters utilizing GridSearchCV to optimize model performance.

Data Collection

Describe how data sets were collected.

You need to present your data collection process use key phrases and flowcharts

- Data was gathered through a GET request to the SpaceX API.
- The response content was subsequently decoded into JSON format by invoking the .json() function, and then transformed into a pandas dataframe via .json_normalize().
- We proceeded to cleanse the dataset, inspect for any missing values, and impute these gaps as required.
- Additionally, we conducted web scraping of Falcon 9 launch records from Wikipedia using BeautifulSoup.
- Our goal was to extract the launch records in the form of an HTML table, parse this table, and then transcribe it into a pandas dataframe to facilitate subsequent analysis.

Data Collection – SpaceX API

1. Retrieve data from the SpaceX API and transform it into a Pandas dataframe.
2. Prepare the data by filtering and organizing it into a list.
3. Create an additional Pandas dataframe from the processed list data.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/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

- Launch data for SpaceX can be sourced from Wikipedia as well.
- Following a specific flowchart, data is acquired from Wikipedia and subsequently stored for use.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/02%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

1. Get request for rocket launch data using API

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

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

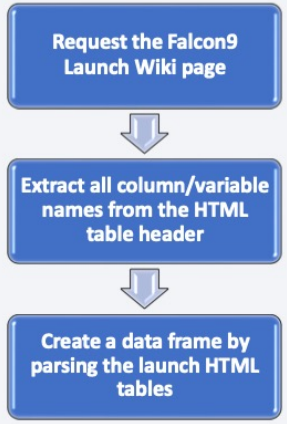
2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

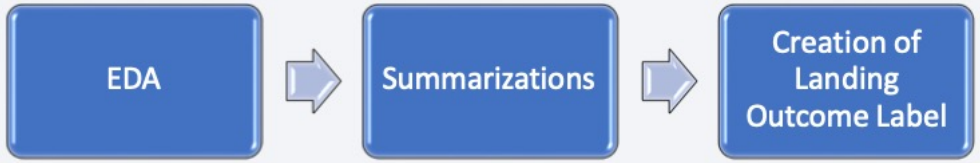
3. We then performed data cleaning and filling in the missing values

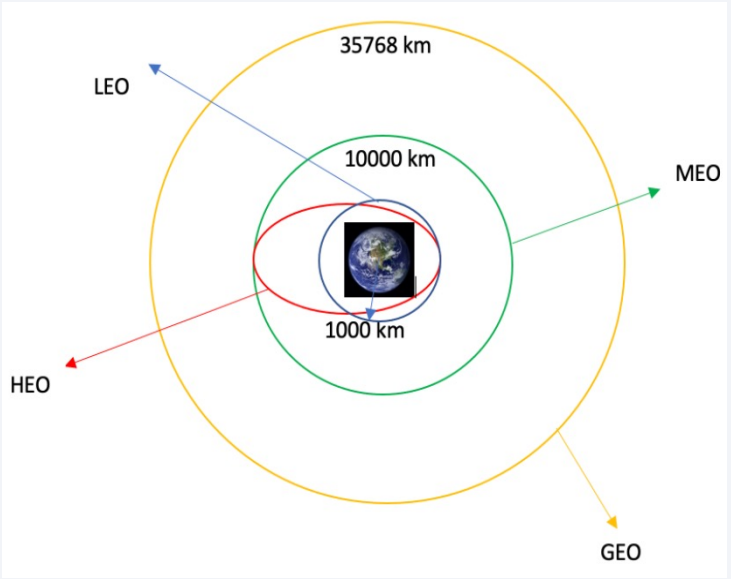
In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```



Data Wrangling

- Conducted Exploratory Data Analysis (EDA).
- Determined the total number of launch sites.
- Computed the count and frequency of each orbit type.
- Calculated the count and distribution of mission outcomes for each orbit type.
- Generated a landing outcome label from the 'Outcome' column.
- Exported the dataset to a CSV file.





https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/03%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Analyzed the correlation between Flight Number and Launch Site.
- Examined the connection between Payload mass and chosen Launch Site.
- Investigated the success rate associated with each orbit type.
- Explored the relationship between Flight Number and Orbit type.
- Studied the link between Payload mass and Orbit type.
- Charted the trend of launch success over successive years.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/05%20EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

1. List unique space mission launch sites.
2. Identify top 5 'CCA' launch sites.
3. Calculate total payload by NASA CRS missions.
4. Average payload by Falcon 9 v1.1 boosters.
5. Date of first successful ground pad landing.
6. Boosters with successful drone ship landings carrying 4,000-6,000 kg payloads.
7. Count of successful vs. failed missions.
8. Booster versions with max payload.
9. 2015 failed drone ship landings with booster versions and launch sites.
10. Rank landing outcomes from 2010-06-04 to 2017-03-20.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/04%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Plotted all launch sites on a Folium map and added map features like markers, circles, and lines to indicate launch successes or failures at each site.
- Categorized launch outcomes as 0 for failure and 1 for success.
- Used color-coded marker clusters to visualize launch sites with higher success rates.
- Measured distances from launch sites to nearby railways, highways, and coastlines, and analyzed their proximity to cities.

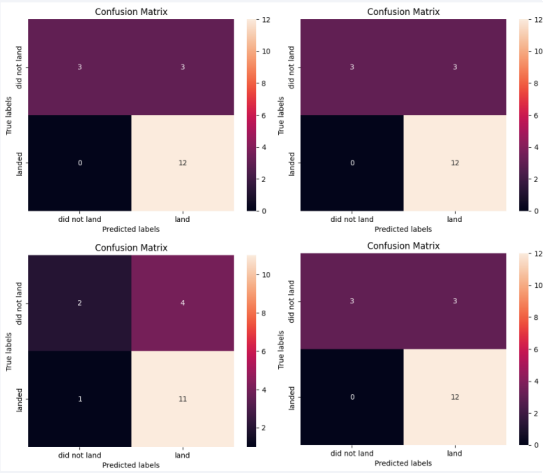
https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/06%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Created a dashboard using Plotly Dash.
- Displayed Pie Charts to represent the count of successful launches across all sites.
- Incorporated a slider to select a range of payload masses.
- Generated a scatter plot that updates according to the selected payload range from the slider.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/07%20spacex_dash_app.py

Predictive Analysis (Classification)

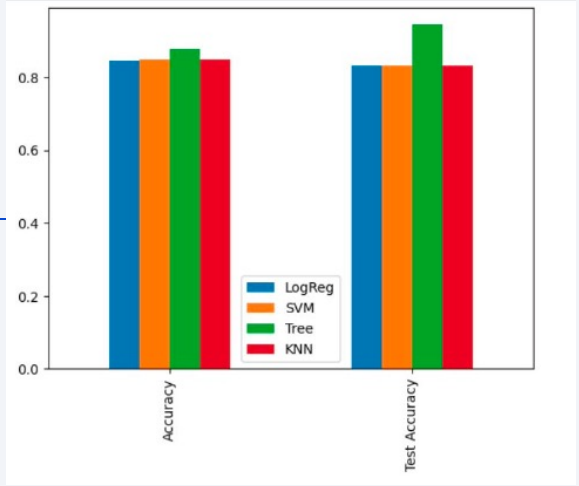


- Conducted Exploratory Data Analysis (EDA) and identified training labels.
- Added a column to denote the class.
- Normalized the dataset.
- Divided the dataset into training and testing subsets.
- Searched for the optimal hyperparameters for SVM, Classification Trees, and Logistic Regression models.
- Evaluated and compared the accuracy of each model using the test data.

https://github.com/raysliang/Applied-Data-Science-Capstone/blob/main/08%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

- SpaceX operates from four distinct launch sites.
- Initial launches were conducted for SpaceX and NASA.
- The F9 v1.1 booster's average payload is 2,928 kg.
- The first successful landing occurred in 2015, five years after SpaceX's inaugural launch.
- Numerous Falcon 9 boosters have successfully landed on drone ships, often carrying payloads above the average.
- Mission outcomes have nearly a 100% success rate.
- In 2015, two booster versions, F9 v1.1 B1012 and F9 v1.1 B1015, failed to land on drone ships.
- Landing outcomes have improved over the years.



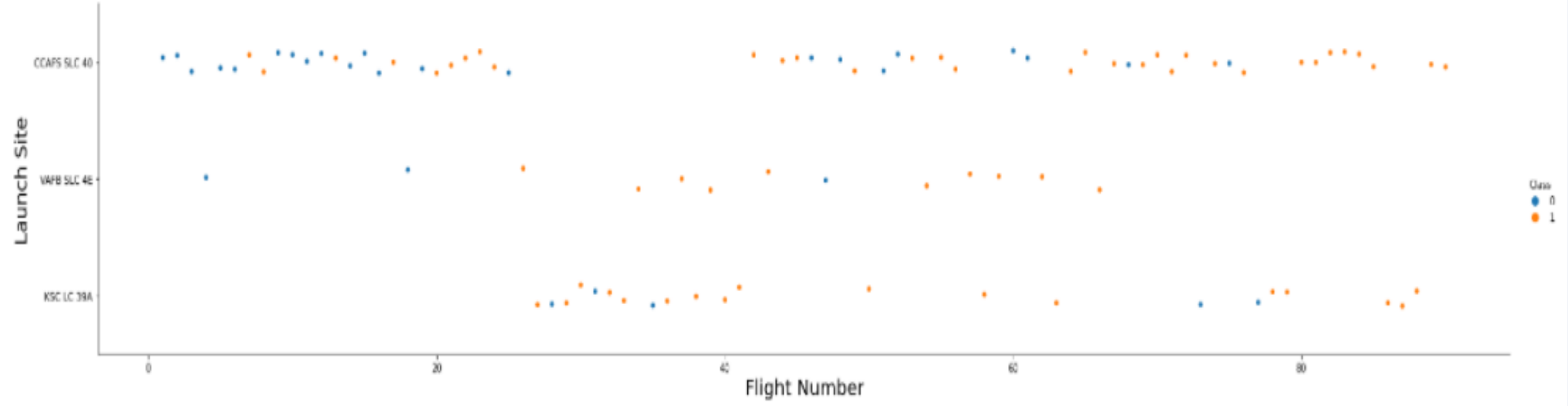
	Accuracy	Test Accuracy
LogReg	0.84643	0.8333
SVM	0.84821	0.8333
KNN	0.84821	0.8333
Tree	0.87679	0.9444

Section 2

Insights drawn from EDA

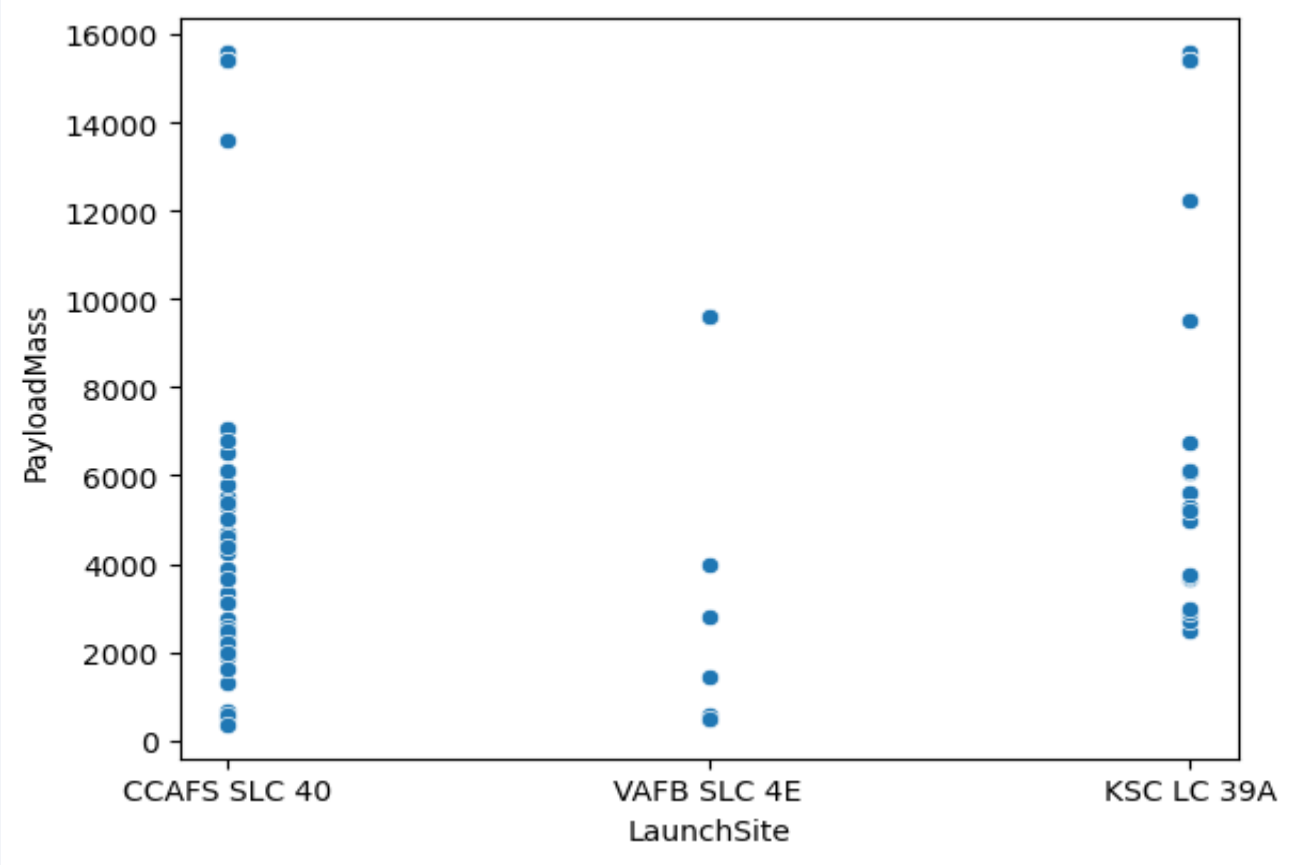
Flight Number vs. Launch Site

The analysis of the plot indicates that a higher number of flights at a launch site correlates with an increased success rate at that site.



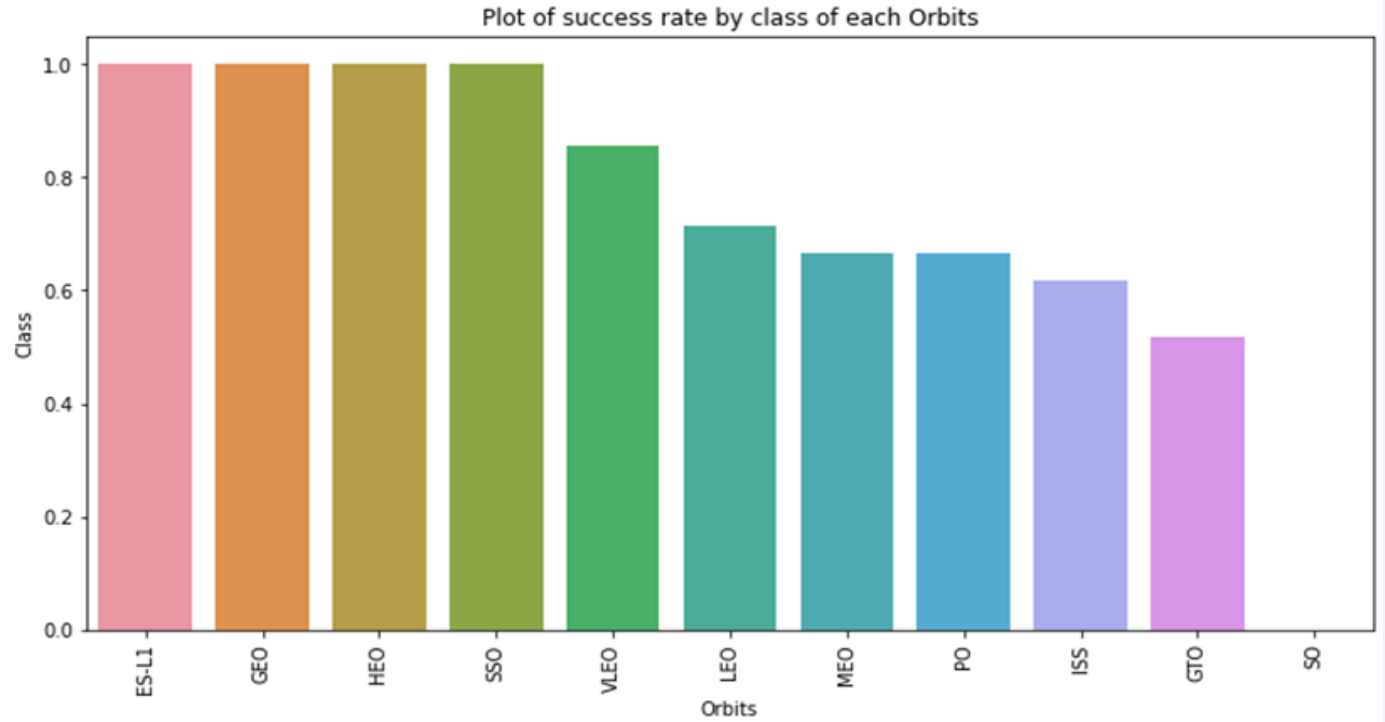
Payload vs. Launch Site

- The CCAFS launch site has conducted the most launches, accommodating a wide range of payload masses.



Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations
- The highest SR orbits are ES-L1,SSO,and HEO

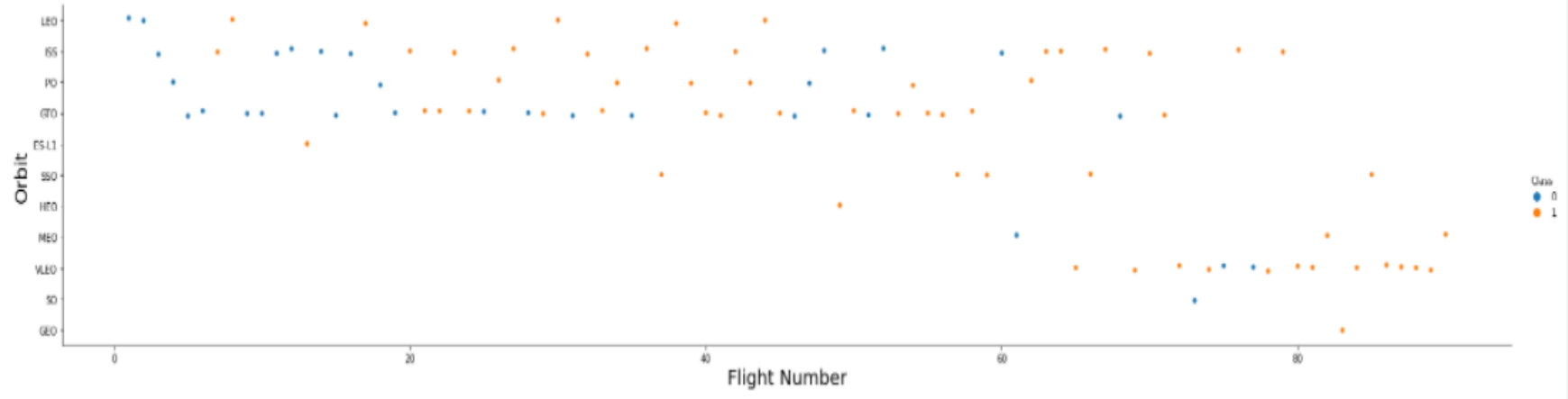


Flight Number vs. Orbit Type

Show a scatter point of Flight number vs. Orbit type

Show the screenshot of the scatter plot with explanations

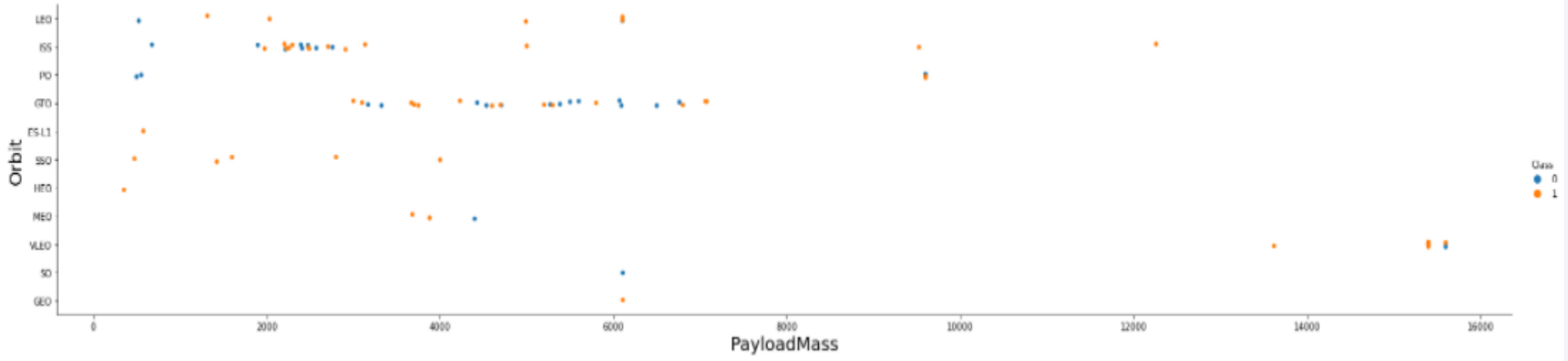
The ISS is high in Success rate



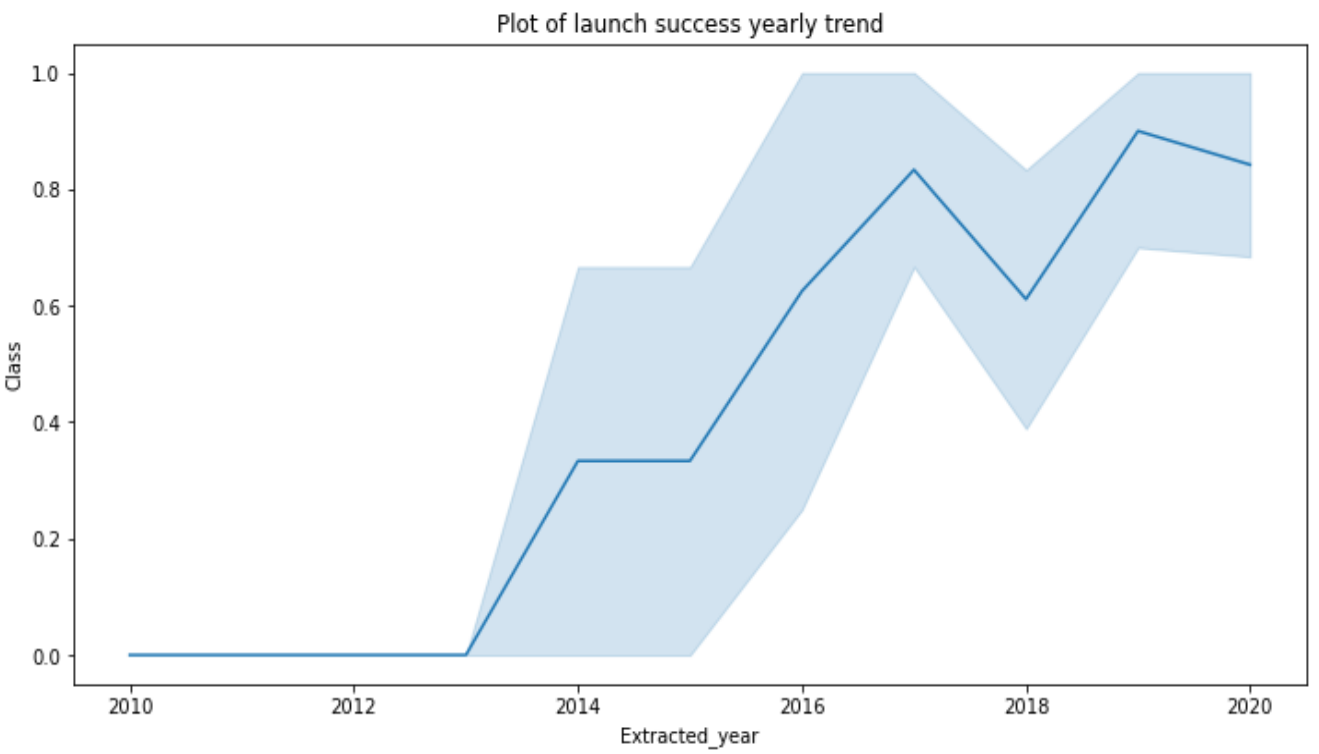
Payload vs. Orbit Type

Show a scatter point of payload vs. orbit type

Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend



- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations
- The success rate start to increase dramatically and the highest where in 2019

All Launch Site Names

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

- Find the names of the unique launch sites
- Present your query result with a short explanation here
- We used Distinct command in sql to find the unique names in launch sites

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

We used the following command (%sql select * from spacex where launch_site like '%CCA%' limit 5) and the result as follow

Out[10]:

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

- Calculate the total payload carried by boosters from NASA
- We used the command sum to calculate the total payload mass

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

Display average payload mass carried by booster version F9 v1.1

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

In [14]:

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

- Performed an SQL query to find the dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

In [15]:
```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

- Performed an SQL query to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [16]:
```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Performed an SQL query to calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

Performed an SQL query to list the names of the booster which have carried the maximum payload mass

2015 Launch Records

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

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Performed an SQL query to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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))

In [19]:

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

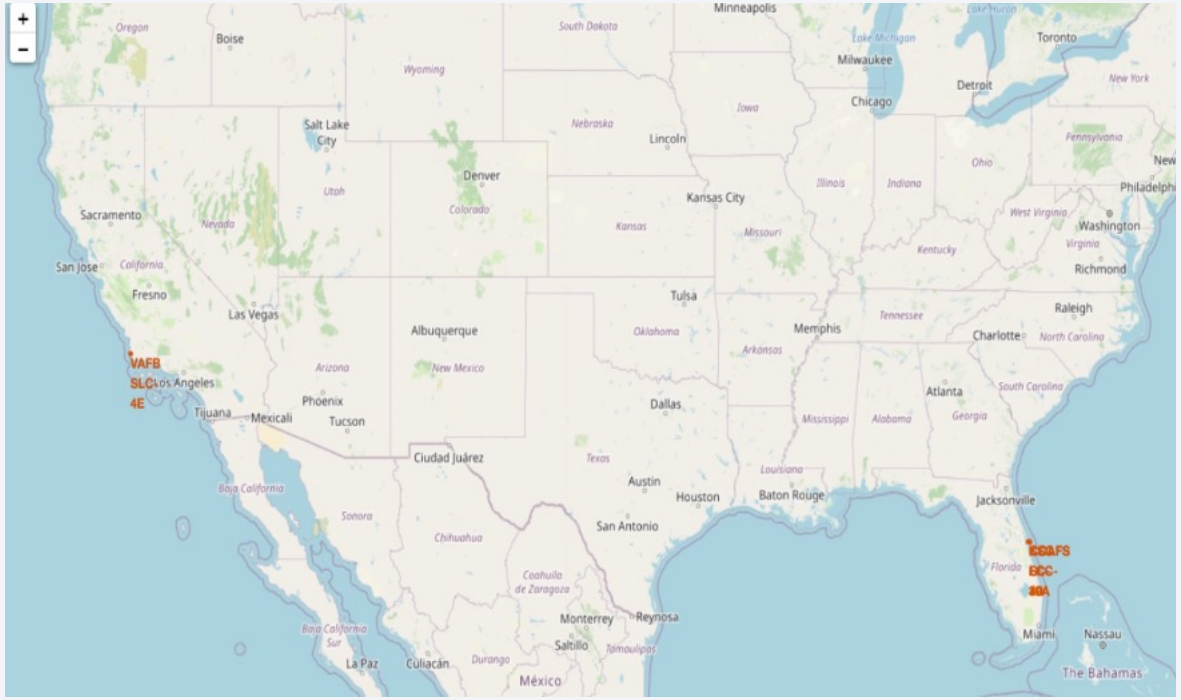
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Performed an SQL query to 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

Section 3

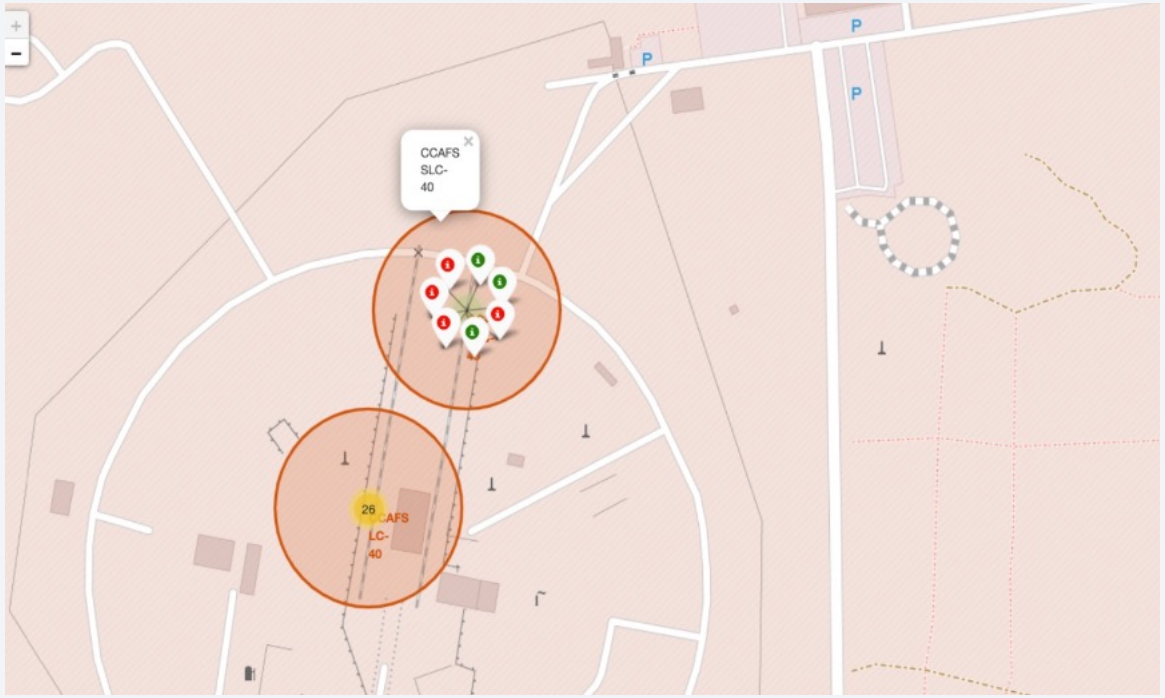
Launch Sites Proximities Analysis

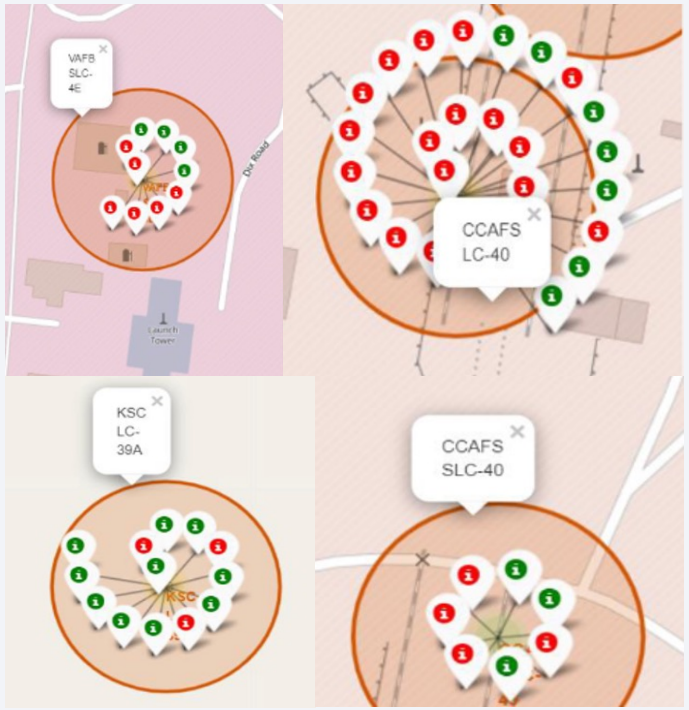
All Launch Sites on Map



- The launch sites are labelled by a marker with their names on the map.

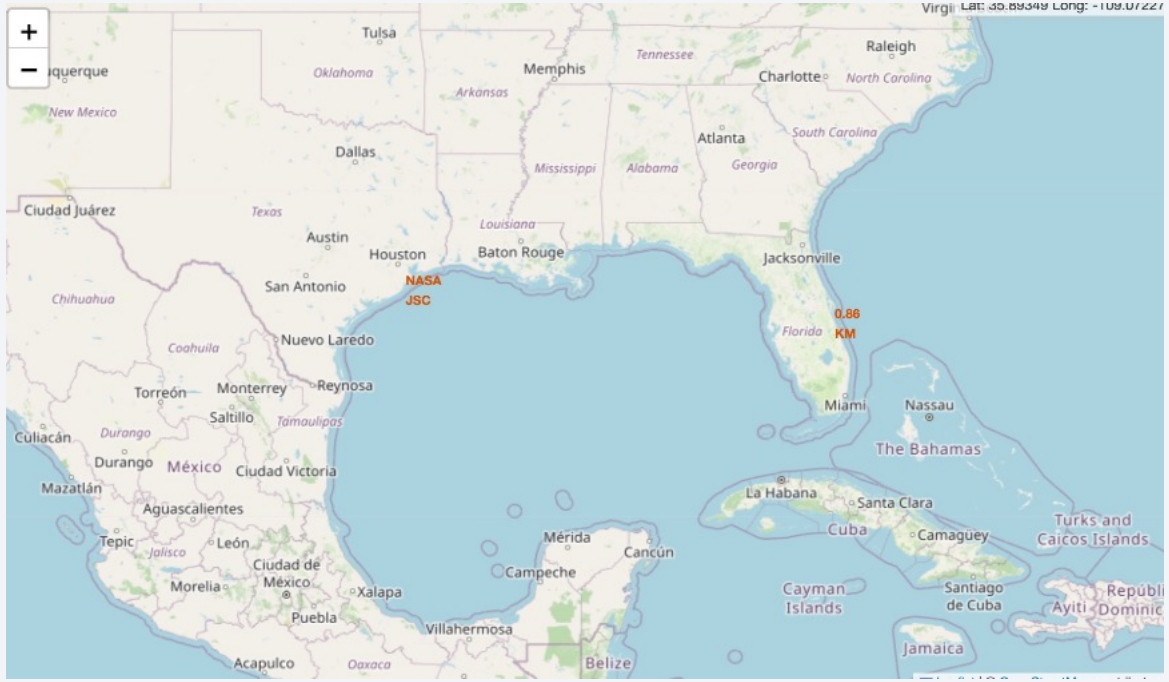
Markers showing Launch Sites



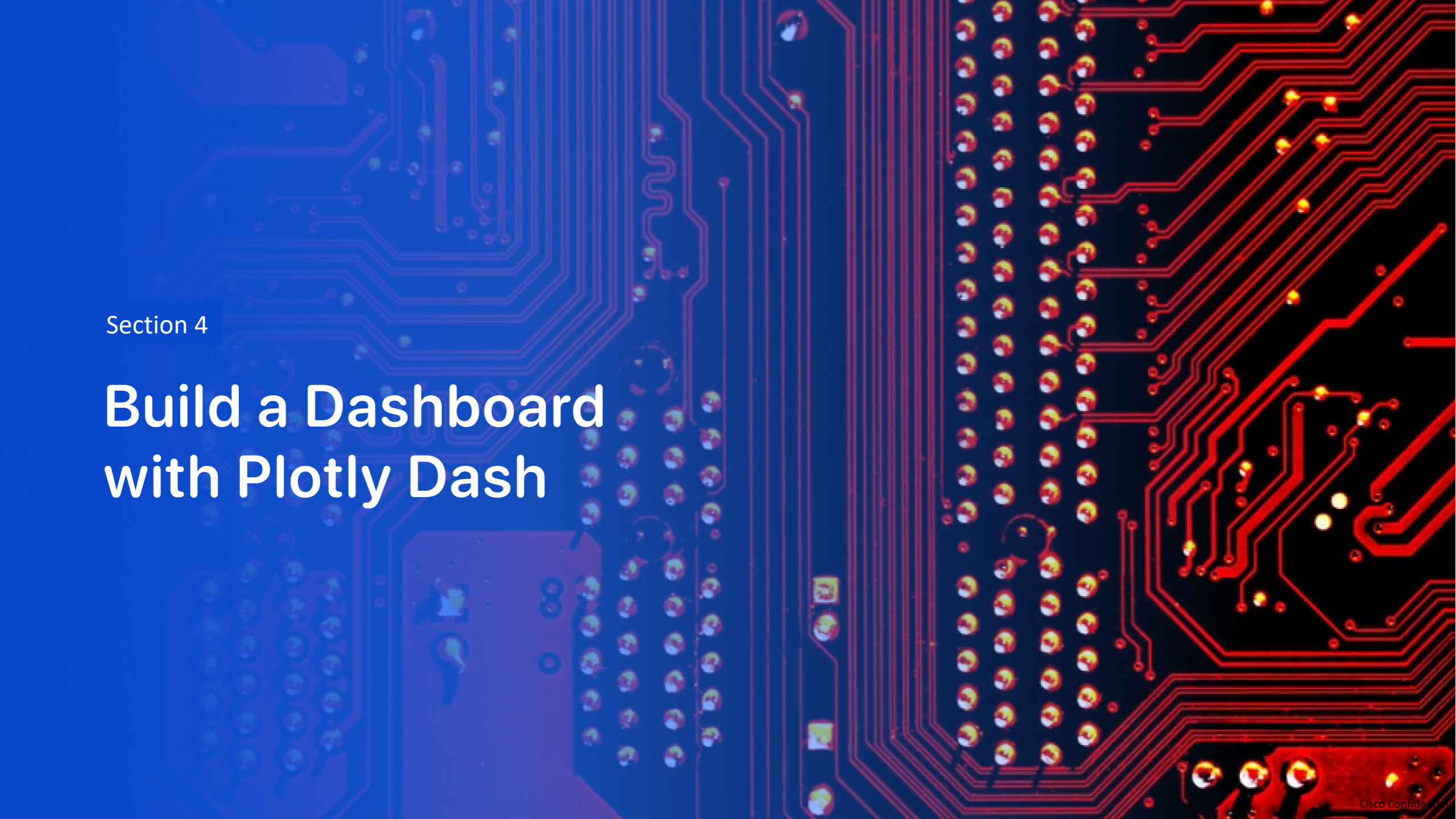


- The launch records are grouped in clusters on the map, then labelled by green markers for successful launches, and red markers for unsuccessful ones.

Distinace between Launch Sites Promixity



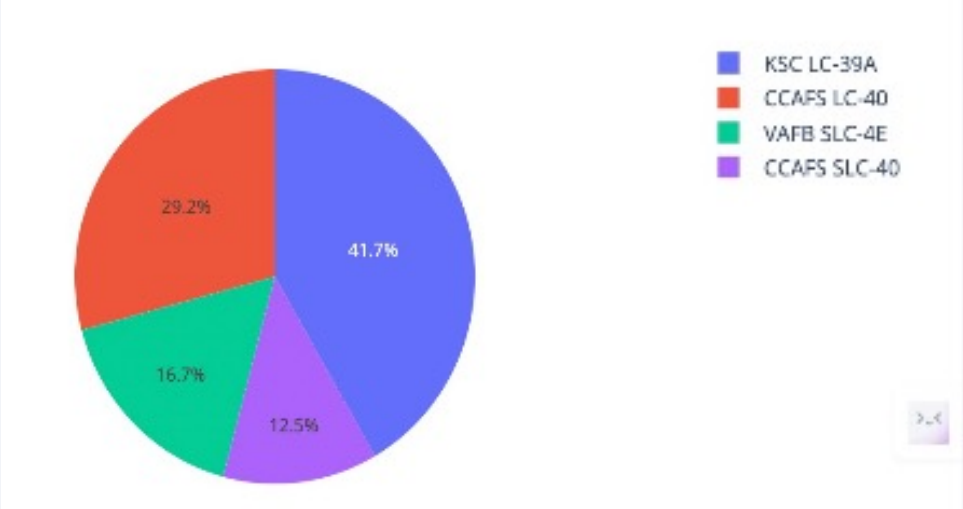
- The closest coastline from NASA JSC is marked as a point using MousePosition and the distance between the coastline point and the launch site, which is approximately 0.86 km.



Section 4

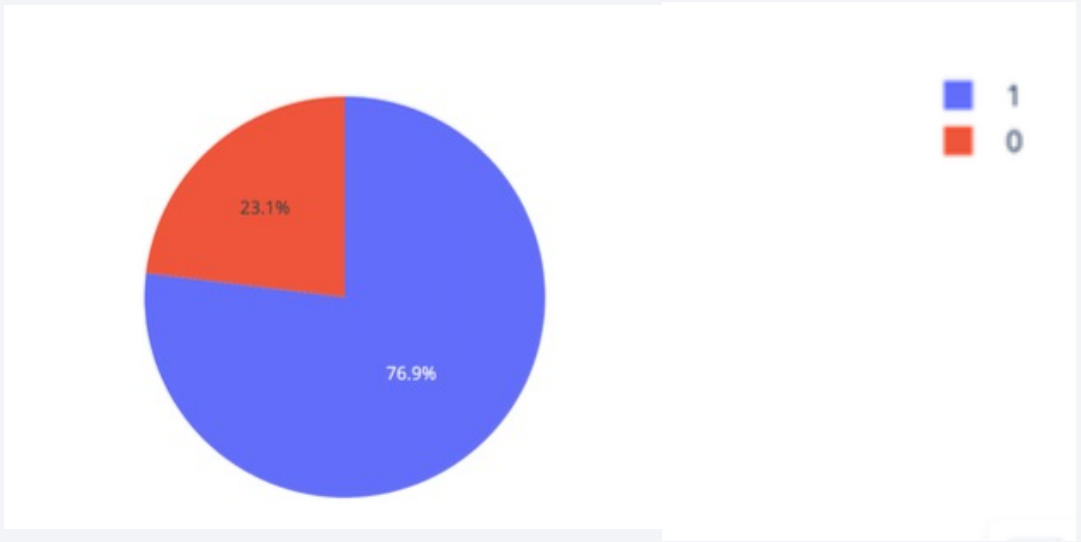
Build a Dashboard with Plotly Dash

Total Success Launches



KSCLC39A has the highest amount of success launches with 41.7% from the entire record

Successful Launch site Ratio



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

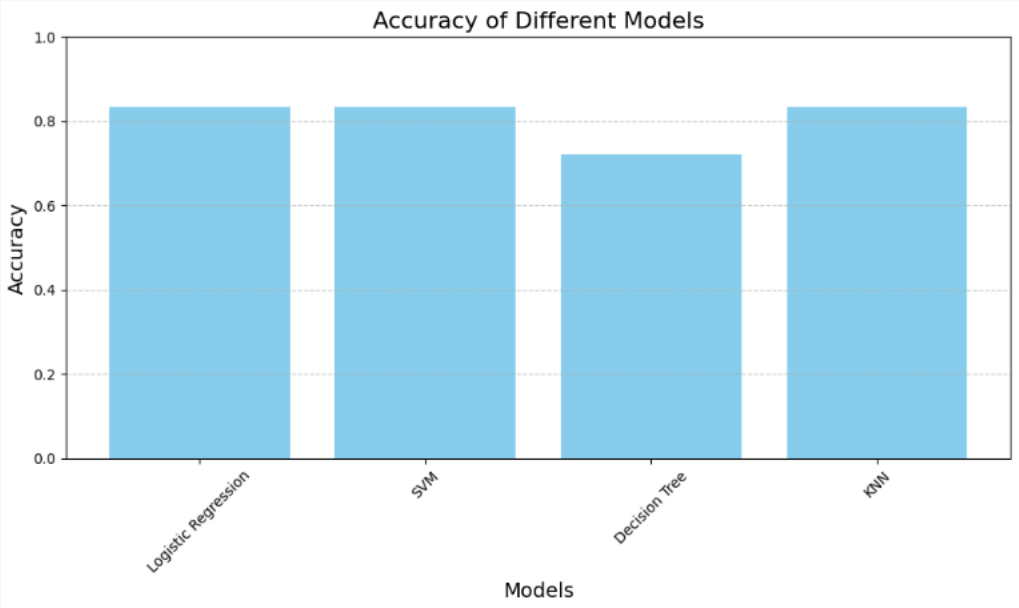
Payload vs Launch Outcome

Booster version FT (green spots) has the highest success launches, followed by B4 (purple spots) with the second highest success launches, among all booster versions.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

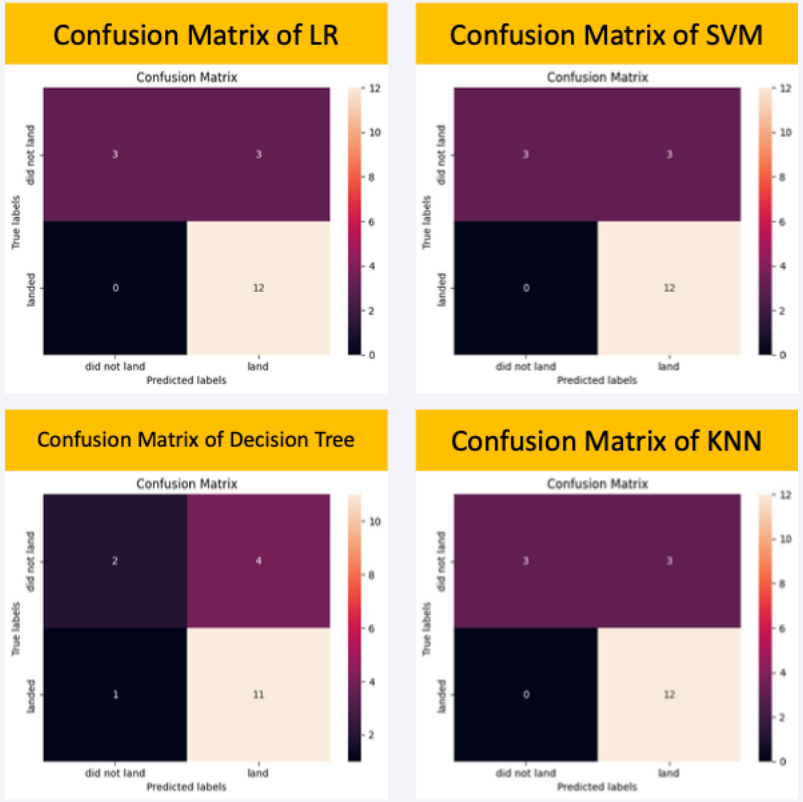



```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[
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_feat
```

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

Confusion Matrix



- LR (Logistic Regression), SVM (Support Vector Machine), and KNN (K-Nearest Neighbors) models accurately predicted all 12 successful landings with no errors.
- The Decision Tree model correctly identified 11 successful landings but misclassified one as a failure or non-landing.
- All three models, LR, SVM, and KNN, share the same accuracy rate of 83.33%, which is reflected in identical confusion matrices.

Conclusions

- LR, SVM, and KNN are the best models for predicting this dataset's launch outcomes.
- Smaller payloads are associated with higher success rates than larger ones.
- SpaceX's launch success rate improves with experience, indicating a trend of increasing proficiency in launches.
- Launch Complex 39A boasts the most successful launches relative to other sites.
- GEO, HEO, SSO, and ES L1 orbits have the highest success frequencies.
- KSC LC-39A has the most successful launches

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

https://github.com/raysliang/Applied-Data-Science-Capstone

Thank you!