IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yutong Gao
02/06/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics
  - Predictive Analytics result

Introduction

Background

SpaceX offers Falcon 9 rocket launches on its website at a significantly lower cost of 62 million dollars compared to other providers, which charge upwards of 165 million dollars per launch. The primary reason for this cost disparity is SpaceX's ability to reuse the first stage of the rocket. Consequently, accurately predicting whether the first stage will successfully land is crucial in estimating the cost of a launch. Such insight becomes invaluable when other companies seek to compete with SpaceX in bidding for rocket launches. Therefore, the objective of this project is to develop a machine learning pipeline capable of predicting the likelihood of a successful first stage landing. Issues:

- Identifying the key factors influencing the successful landing of the rocket.
- Analyzing the interplay among different features affecting the success rate of the first stage landing.
- Determining the operational conditions necessary for a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

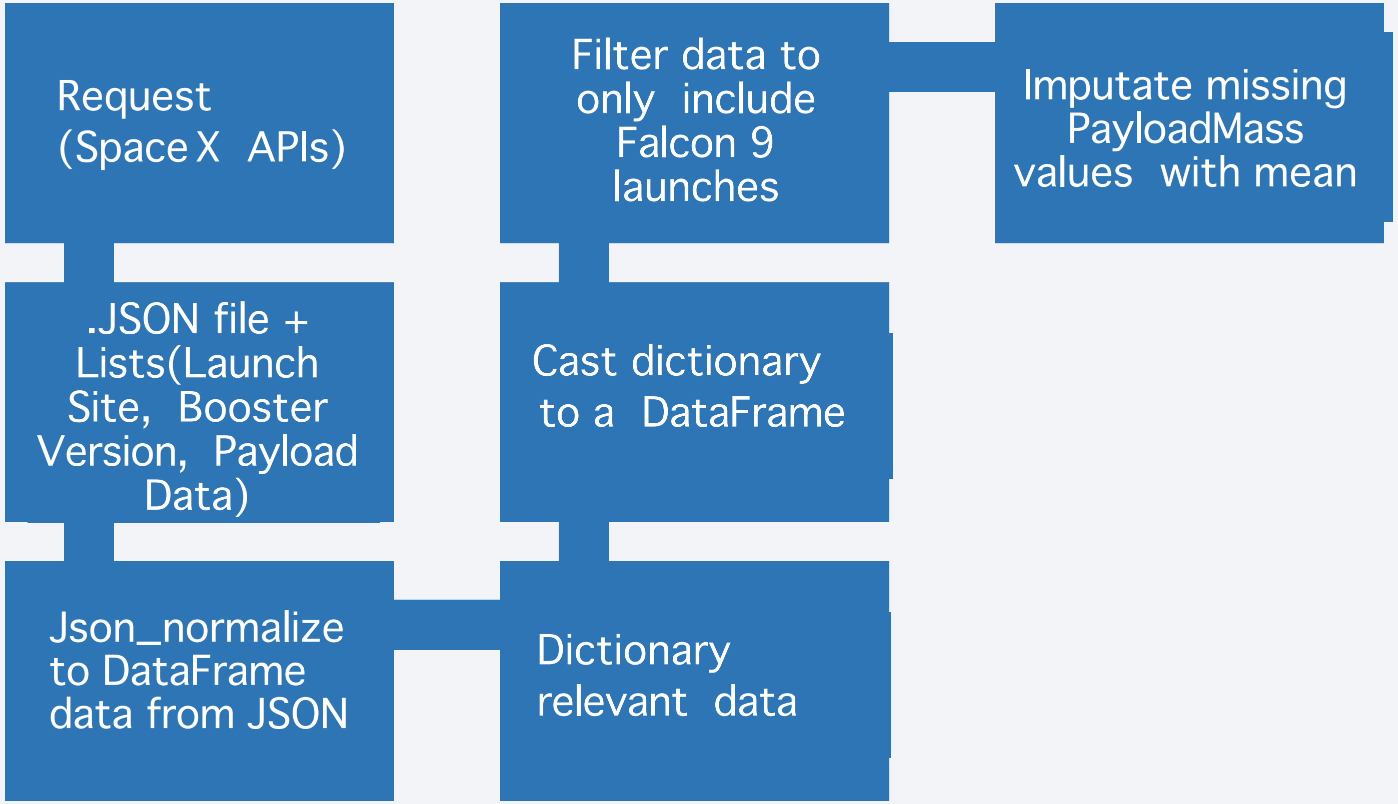
- Data collection methodology:
  - Using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Data processing and One-hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, and evaluate classification models

Data Collection

- The data collection process involved multiple methods. Initially, we utilized GET requests to access the SpaceX API. Subsequently, we decoded the response content into JSON format using the .json() function call and transformed it into a Pandas dataframe using .json_normalize(). Following this, data cleaning procedures were implemented, including the identification and handling of missing values. Additionally, web scraping techniques were employed to extract Falcon 9 launch records from Wikipedia using BeautifulSoup. The goal was to retrieve launch records presented as HTML tables, parse the data, and convert it into a Pandas dataframe to facilitate further analysis.

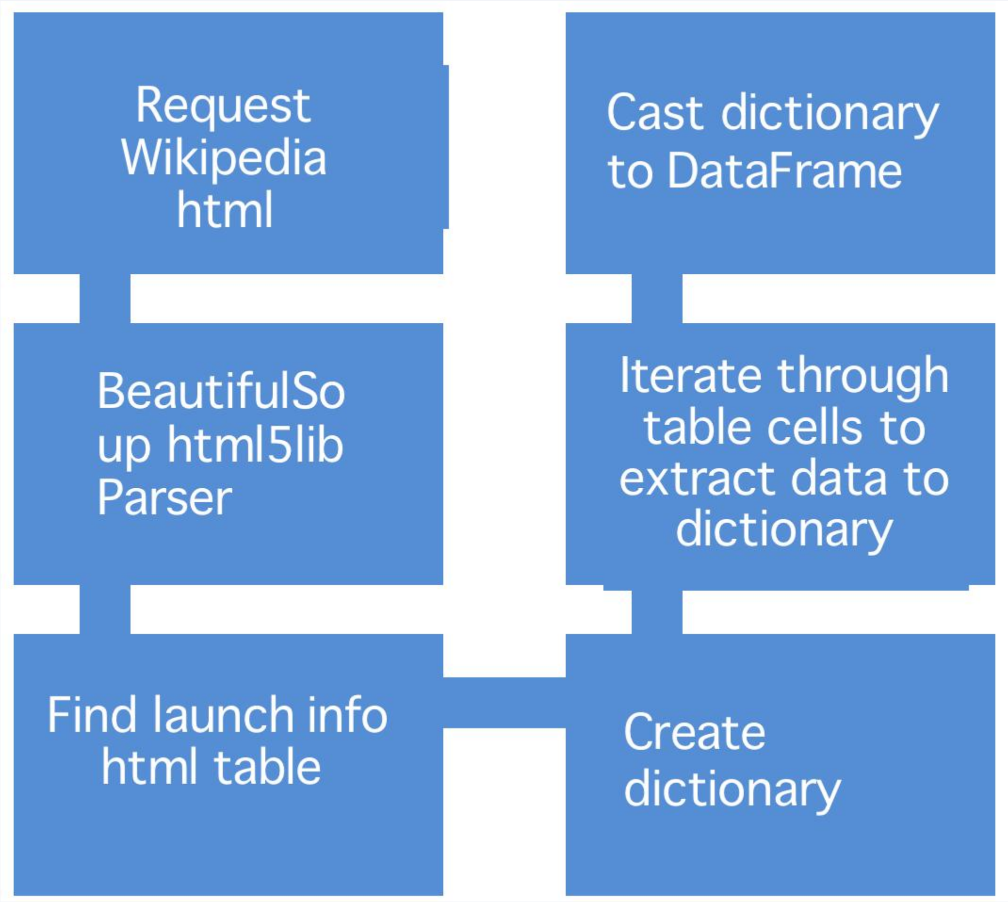
Data Collection – SpaceX API

- We utilized a GET request to the SpaceX API to gather data, conducted data cleaning, and performed basic data wrangling and formatting tasks. For further details, please refer to the notebook available at the following link: Data Collection API Notebook.



Data Collection - Scraping

- We employed web scraping techniques using BeautifulSoup to extract Falcon 9 launch records. Subsequently, we parsed the table and transformed it into a Pandas dataframe. For further details, please refer to the notebook available at the following link: Data Collection with Web Scraping Notebook.



Data Wrangling

- We conducted exploratory data analysis (EDA) to establish the training labels. This involved calculating the frequency of launches at each site and identifying the distribution of orbits. Furthermore, we derived a landing outcome label from the outcome column and exported the results to a CSV file. For more detailed information, please refer to the notebook available at the following link: Data Wrangling Notebook.

EDA with Data Visualization

We conducted data exploration by visualizing various relationships within the dataset. These visualizations included:

- Exploring the relationship between flight number and launch site.
- Analyzing the relationship between payload and launch site.
- Investigating the success rate of each orbit type.
- Examining the relationship between flight number and orbit type.
- Identifying the yearly trend in launch success.
- For more detailed insights, please refer to the exploratory visualizations in the provided notebook: Data Exploration Notebook.

EDA with SQL

In the provided notebook, we loaded the SpaceX dataset into a PostgreSQL database directly from Jupyter Notebook. We then applied exploratory data analysis (EDA) using SQL queries to gain insights from the data. Some of the queries we executed include:

- Finding the names of unique launch sites in the space mission.
- Calculating the total payload mass carried by boosters launched by NASA (CRS).
- Determining the average payload mass carried by booster version F9 v1.1.
- Obtaining the total number of successful and failed mission outcomes.
- Identifying the failed landing outcomes on drone ships, along with their respective booster versions and launch site names.
- For more details and the SQL queries used, please refer to the notebook available at the following link: EDA with SQL Notebook.

Build an Interactive Map with Folium

In our analysis, we marked all launch sites on a Folium map and added map objects such as markers, circles, and lines to indicate the success or failure of launches for each site. We assigned the feature launch outcomes (failure or success) to class 0 and 1, with 0 representing failure and 1 representing success.

Using color-labeled marker clusters, we identified launch sites with relatively high success rates. Additionally, we calculated the distances between each launch site and its proximities, answering questions such as:

- Are launch sites near railways, highways, and coastlines?

- Do launch sites maintain a certain distance from cities?

For more detailed insights and visualizations, please refer to the notebook available at the following link: [Analysis and Visualization Notebook](https://github.com/chuksoo/IBM-Data-Science-Capstone-SpaceX/blob/main/Analysis%20and%20Visualization.ipynb).

Build a Dashboard with Plotly Dash

In our project, we developed an interactive dashboard using Plotly Dash. The dashboard includes:

- Pie charts displaying the total launches from different launch sites.
- Scatter plots illustrating the relationship between launch outcome and payload mass (in kilograms) for different booster versions.

To explore the interactive dashboard and view the visualizations, please refer to the notebook available at the following link: Plotly Dash Dashboard Notebook.

Predictive Analysis (Classification)

In our project, we followed a comprehensive approach for building machine learning models:

- Data loading and preprocessing: We utilized numpy and pandas to load and transform the data. After preprocessing, we split the data into training and testing sets.
- Model building and hyperparameter tuning: We constructed various machine learning models and fine-tuned their hyperparameters using GridSearchCV to optimize performance.
- Model evaluation: We used accuracy as the metric to evaluate the performance of our models. Additionally, we focused on feature engineering and algorithm tuning to enhance model performance.
- Identification of the best-performing model: Through rigorous experimentation and evaluation, we identified the classification model that achieved the highest accuracy.

For detailed implementation and results, please refer to the notebook available at the following link: Machine Learning Prediction Notebook.

Results

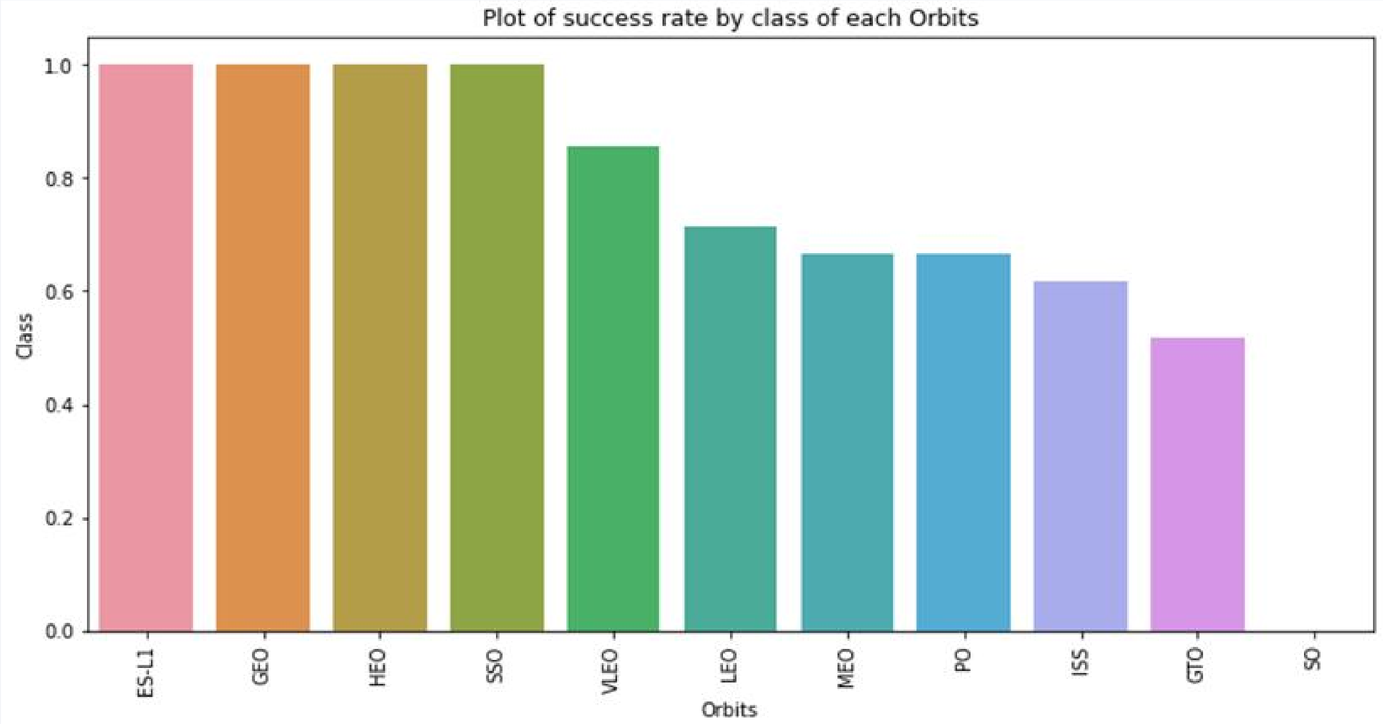
- Exploratory data analysis results
- Interactive analytics demo
- Predictive analysis results

Section 2

Insights drawn from EDA

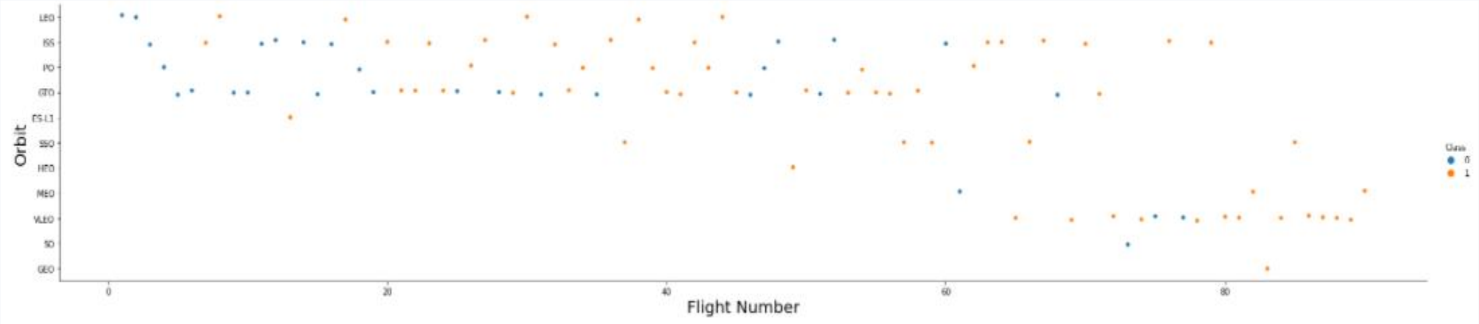
Success Rate vs. Orbit Type

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

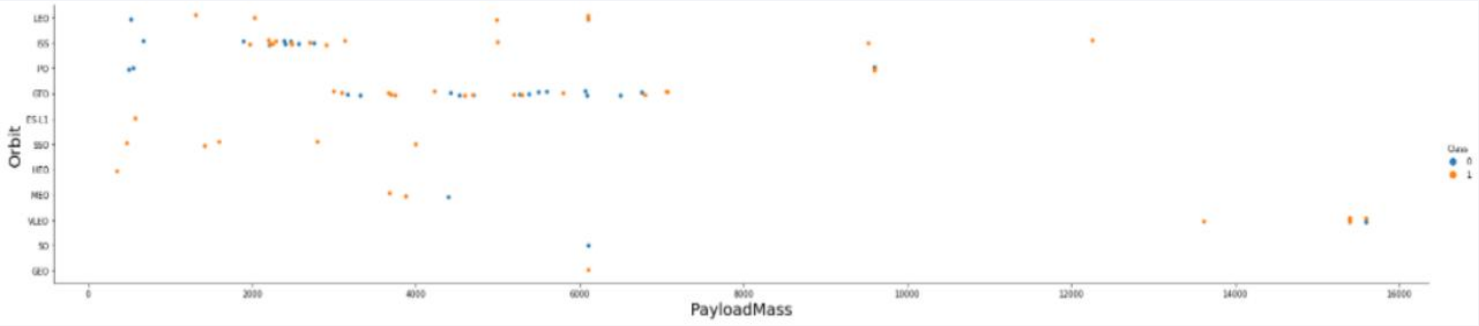


Flight Number & Payload vs. Orbit Type

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.

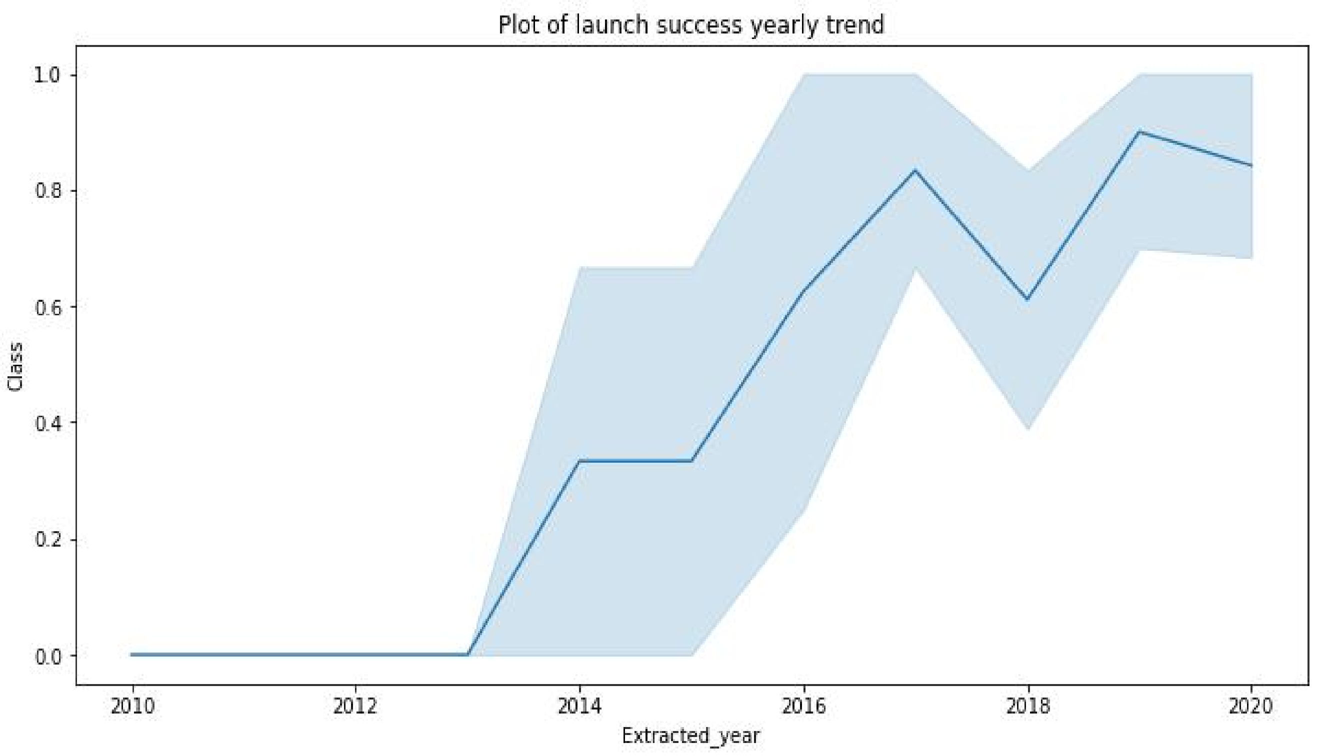


We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



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

Launch Site Names Begin with 'CCA'

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

In [11]:

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	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 & Average 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

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

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

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

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)

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]:

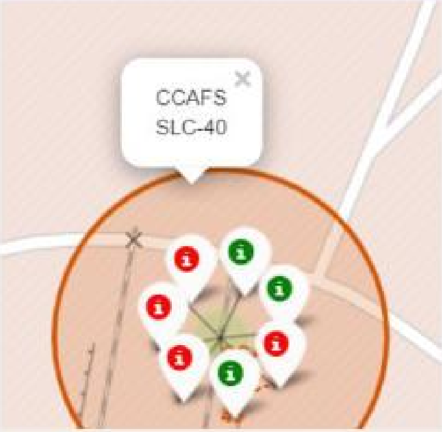
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

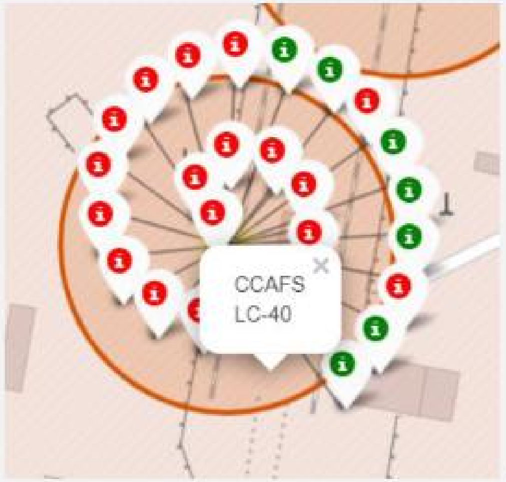
Section 4

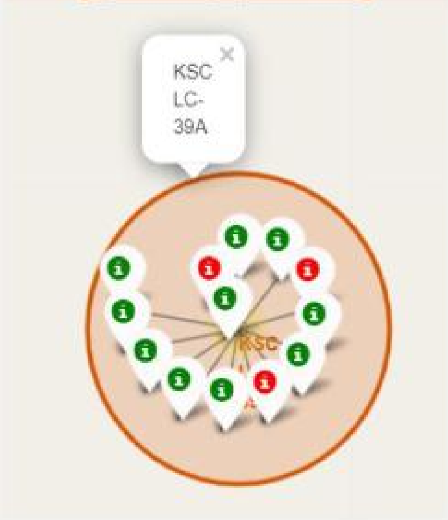
Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels







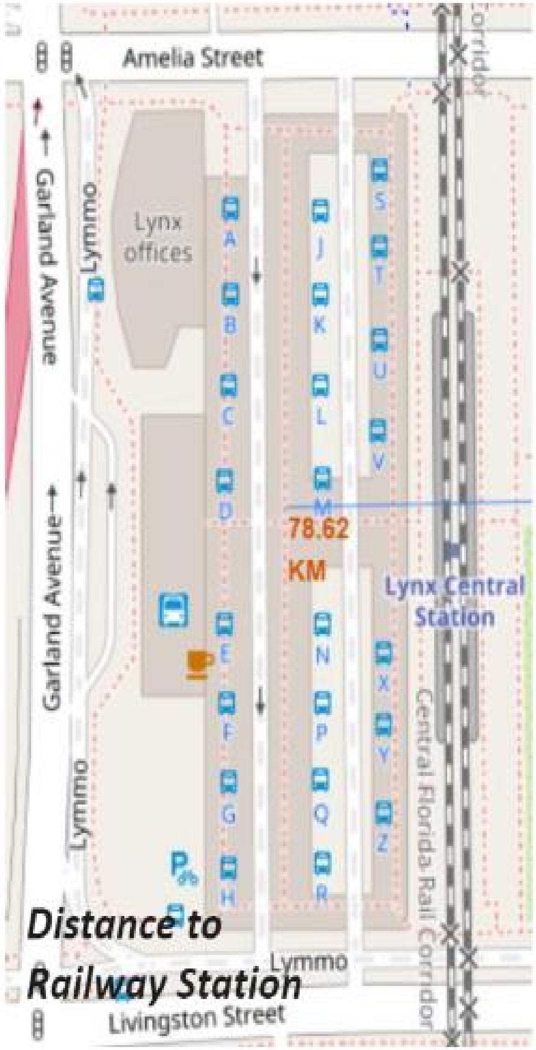
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

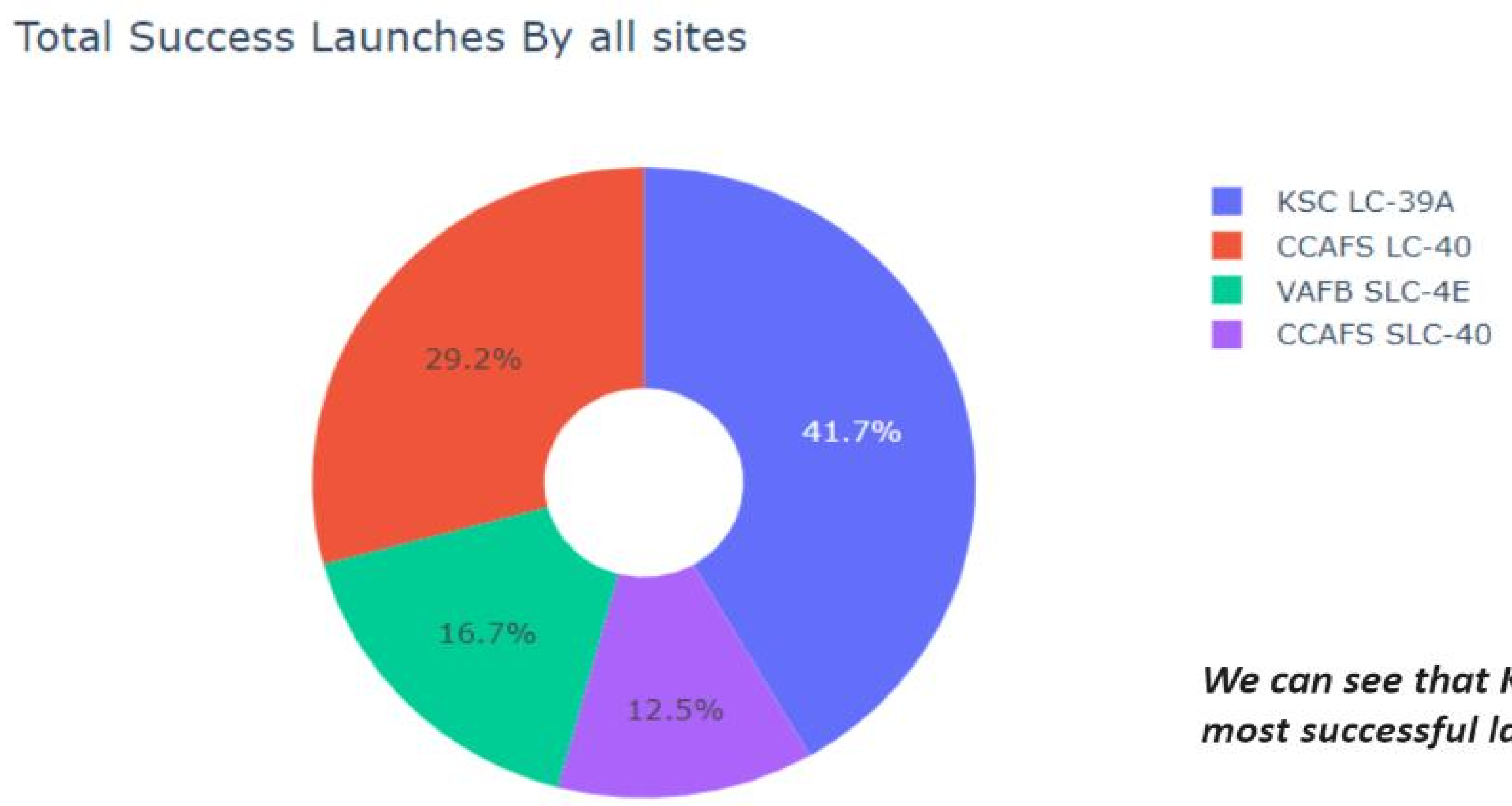


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 5

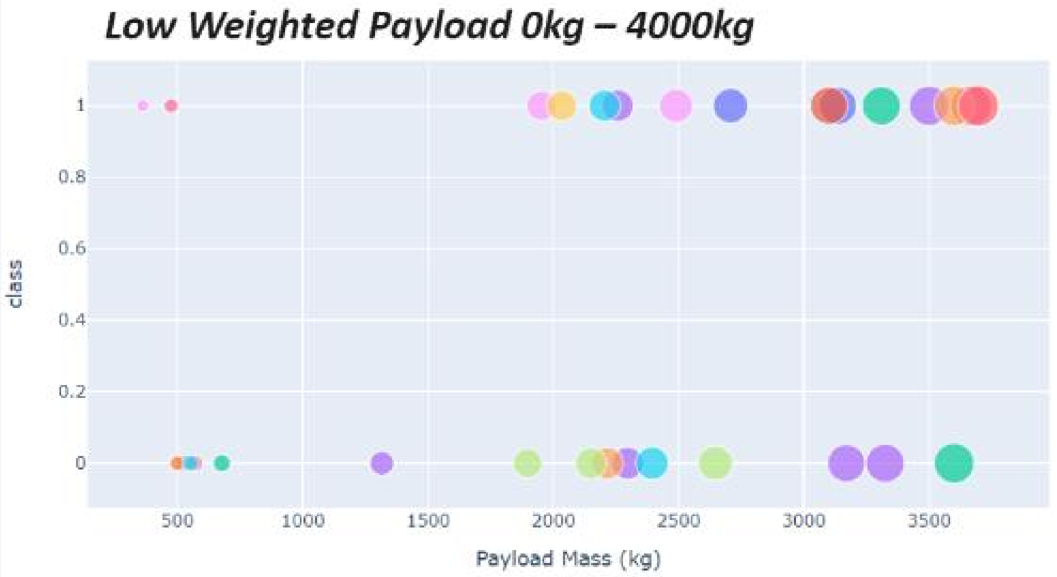
Build a Dashboard with Plotly Dash

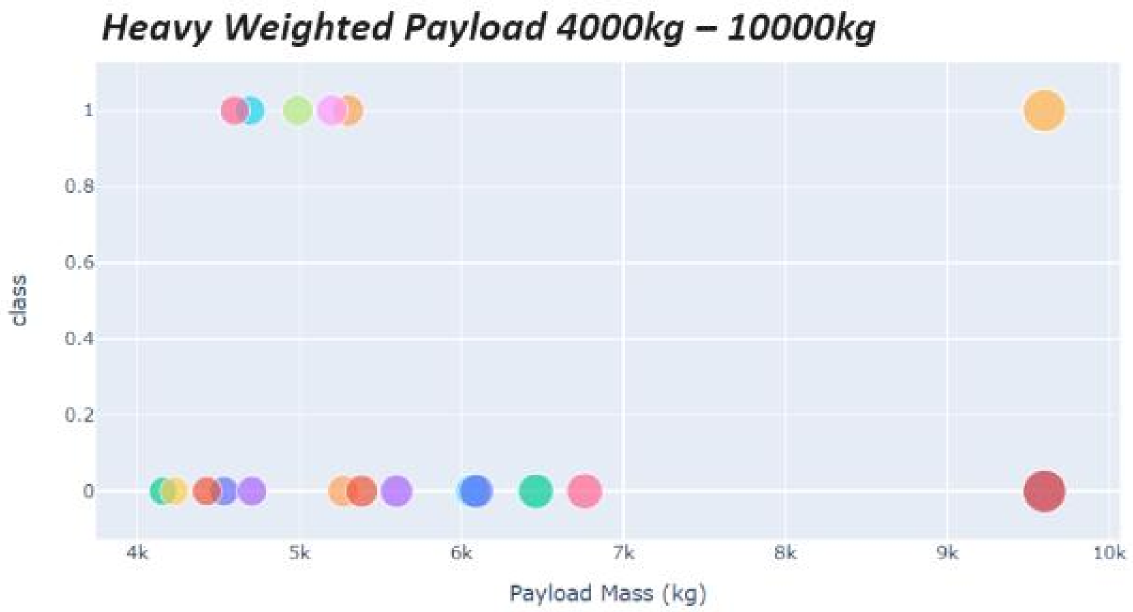
Pie chart showing the success percentage achieved by each launch site



We can see that KSC LC-39A had the most successful launches from all the sites

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





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

Section 6

Predictive Analysis (Classification)

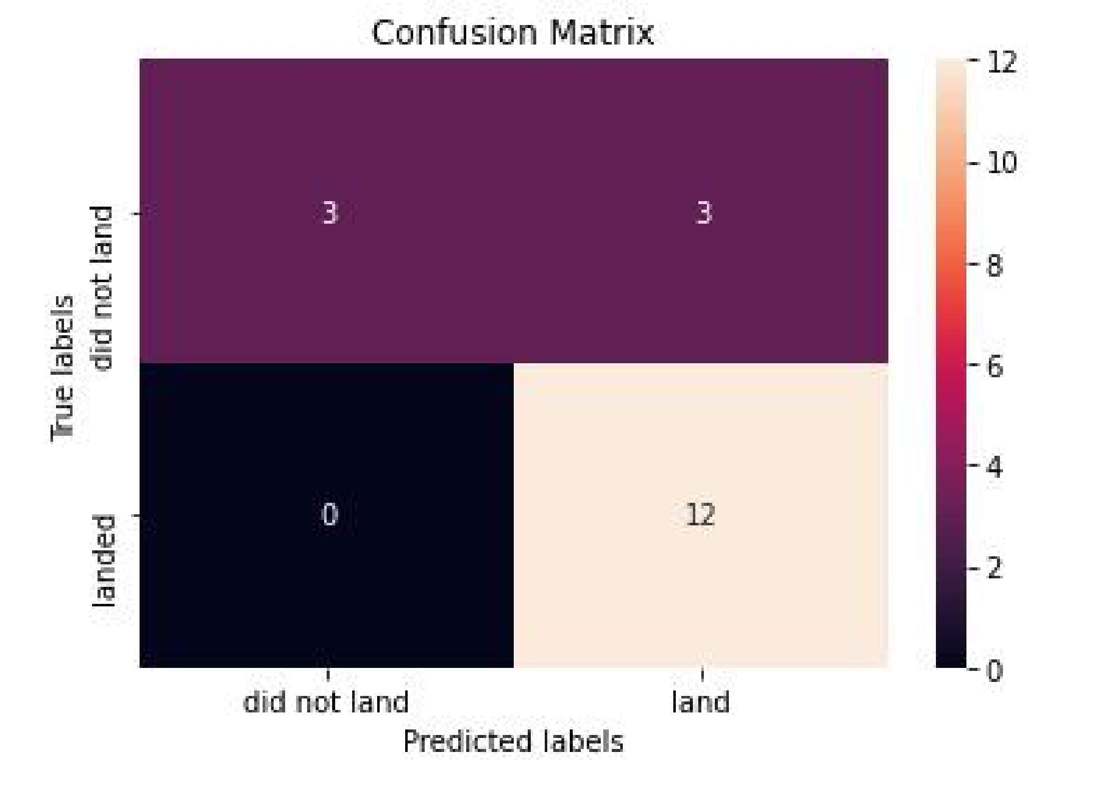
Classification Accuracy

```
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[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



Conclusions

Based on our analysis and findings:

- There is a positive correlation between the number of flights conducted at a launch site and the success rate of launches at that site.
- The launch success rate began to increase in 2013 and continued to improve until 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.
- KSC LC-39A emerged as the launch site with the highest number of successful launches.
- The Decision Tree Classifier demonstrated superior performance as the best machine learning algorithm for this task, based on the evaluation metrics employed.

These conclusions provide valuable insights into the factors influencing launch success rates and the effectiveness of different machine learning algorithms in predicting launch outcomes.

Thank you!