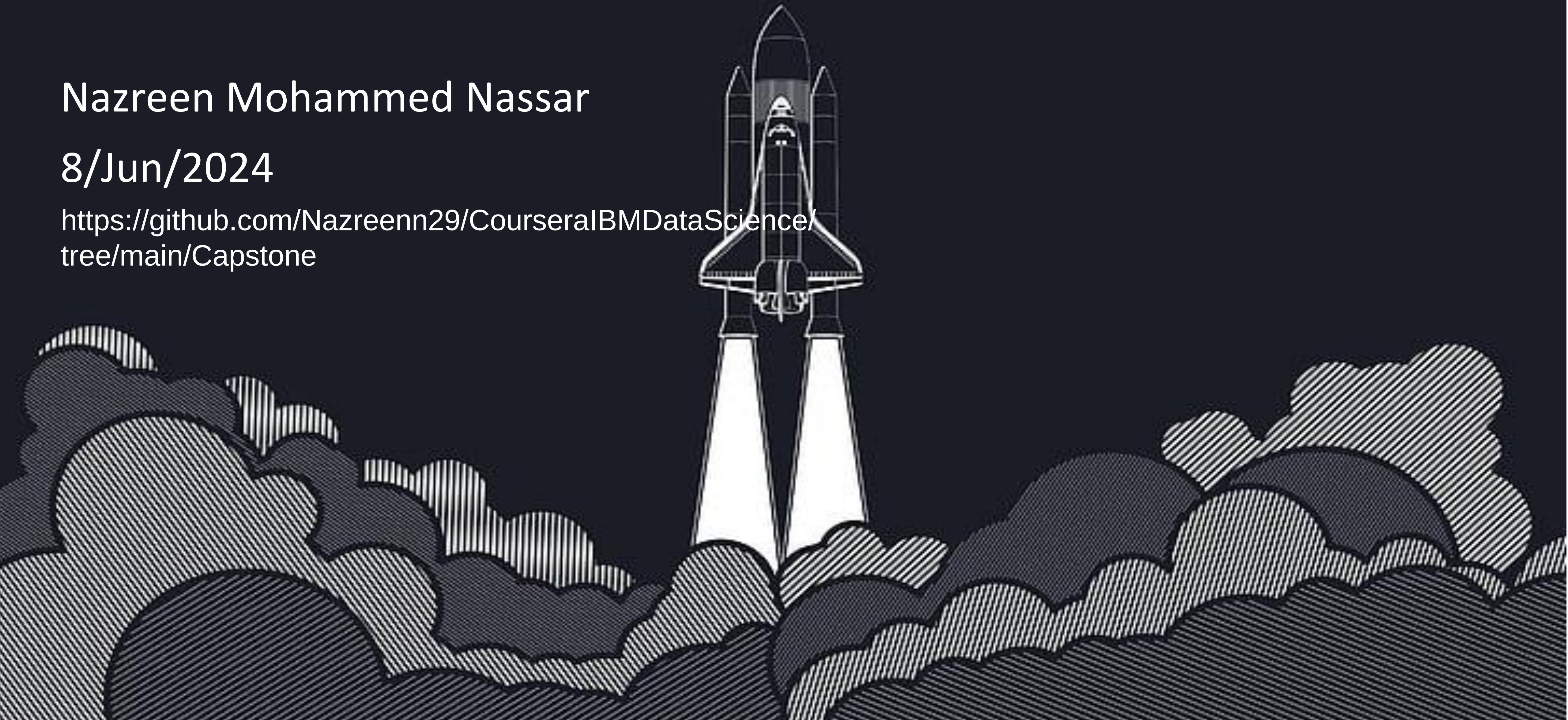
Data Science Supremacy in Space

Nazreen Mohammed Nassar

8/Jun/2024

https://github.com/Nazreenn29/CourseraIBMDataScience/tree/main/Capstone



Project Blueprint

- Executive Summary
- Introduction
- Methodology
- Results
  - EDA with Visualization
  - EDA with SQL
  - Interactive Maps with Folium
  - Plotly Dash Dashboard
  - Predictive Analytics
- Conclusion

Executive Summary

The objective of this capstone project was to identify the factors contributing to the successful landing of the first stage of a SpaceX Falcon 9 rocket. Using data sourced from the public SpaceX API and SpaceX Wikipedia page, the analysis aimed to differentiate between successful and unsuccessful launches. This involved a comprehensive exploration and transformation of the data, followed by the application of machine learning models to predict landing outcomes.

Summary of Methodologies

- **Data Collection**:Utilized the SpaceX REST API and web scraping techniques to gather relevant data on rocket launches.
- **Data Wrangling**:Created a success/fail outcome variable to classify the landing results.
- **Data Exploration**:Used data visualization techniques to analyze factors such as payload, launch site, flight number, and yearly trends.Calculated key statistics with SQL, including total payload, payload range for successful launches, and the total number of successful and failed outcomes.
- **Geographical Analysis**:Examined launch site success rates and their proximity to geographical markers.Visualized the launch sites with the highest success rates and identified successful payload ranges.
- **Model Building**:Developed models to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree, and K-nearest neighbor (KNN) algorithms.

Result

Exploratory Data Analysis:

- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate

Visualization/Analytics:

- Most launch sites are near the equator, and all are close to the coast

Predictive Analytics:

- All models performed similarly on the test set. The decision tree model slightly outperformed

Introduction

Background

SpaceX, at the forefront of space exploration, endeavors to democratize space travel by making it accessible to all. Among its notable achievements are missions to the International Space Station, deployment of a satellite constellation for global internet coverage, and successful manned space missions. The key to SpaceX's cost-effective approach, with launches priced at $62 million each, lies in its innovative reuse of the Falcon 9 rocket's first stage. In contrast, competitors, lacking this reusability feature, incur significantly higher costs, upwards of $165 million per launch. By accurately predicting the landing success of the first stage, the pricing of launches can be determined. Leveraging public data and machine learning models, we aim to forecast the likelihood of first-stage reusability, whether for SpaceX or its competitors, thus shaping the future economics of space exploration.

Examination of Influential Factors: Investigating the impact of payload mass, launch site, flight frequency, and orbital trajectories on the success of first-stage landings.

Temporal Analysis of Success Rates: Examining the trend in successful first-stage landings over time to understand the trajectory of technological advancement and operational efficiency.

Optimization of Predictive Modeling: Developing and refining predictive models for binary classification to accurately forecast the success of first-stage landings, utilizing machine learning techniques and historical data.

METHODOLOGY

Methodology

Steps:

- **Data Collection:**Gather data using SpaceX's REST API and web scraping techniques.

- **Data Wrangling:**Prepare the data for analysis and modeling by filtering, handling missing values, and applying one-hot encoding.

- **Exploratory Data Analysis (EDA):**Explore the data using SQL and data visualization techniques to uncover insights and trends.

- **Data Visualization:**Visualize the data with tools like Folium and Plotly Dash to create interactive and informative visual representations.

- **Model Building:**Develop classification models to predict landing outcomes. Fine-tune and evaluate these models to identify the best-performing model and optimal parameters.

Data Collection

The data collection process involved a combination of API requests from the SpaceX public API and web scraping data from a table on SpaceX's Wikipedia page. The following slide will present a flowchart detailing the data collection process from the API, and the subsequent slide will illustrate the flowchart for data collection via web scraping.

- SpaceX API Data Columns:FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

- Wikipedia Web Scrape Data Columns:Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection –SpaceX API

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Steps:

Request Data from SpaceX API: Initiate a request to the SpaceX API to retrieve comprehensive data on rocket launches.

Decode API Response: Decode the API response using the .json() method and convert it into a dataframe using the .json_normalize() function for easier manipulation and analysis.

Fetch Detailed Launch Information: Utilize custom functions to request detailed information about the launches from the SpaceX API, enriching the dataset with additional launch details.

Create Dictionary from Data: Organize the retrieved data into a dictionary structure to facilitate data manipulation and transformation.

Convert Dictionary to Dataframe: Transform the dictionary into a dataframe, ensuring the data is structured in a tabular format suitable for analysis.

Filter Dataframe for Falcon 9 Launches: Filter the dataframe to include only launches conducted by the Falcon 9 rocket, focusing the analysis on relevant data.

Handle Missing Values: Address missing values in the Payload Mass column by replacing them with the calculated mean, ensuring completeness of the dataset.

Export Data to CSV File: Export the cleaned and processed data to a CSV file, facilitating further analysis and modeling.

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 Collection –Web Scraping

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/jupyter-labs-webscraping%20(1).ipynb

Retrieve Data from SpaceX API: Make a request to the SpaceX API to obtain detailed data on rocket launches. This data will include various parameters such as launch dates, payload details, and landing outcomes.

Parse and Normalize API Response: Decode the API response using the .json() method to convert it from JSON format into a Python dictionary. Then, use the .json_normalize() function to transform this data into a pandas dataframe, which allows for easier data manipulation and analysis.

Collect Additional Launch Information: Develop and employ custom functions to make additional API requests, fetching more granular details about each launch. This step enriches the dataset by adding information such as mission objectives, payload specifics, and technical details

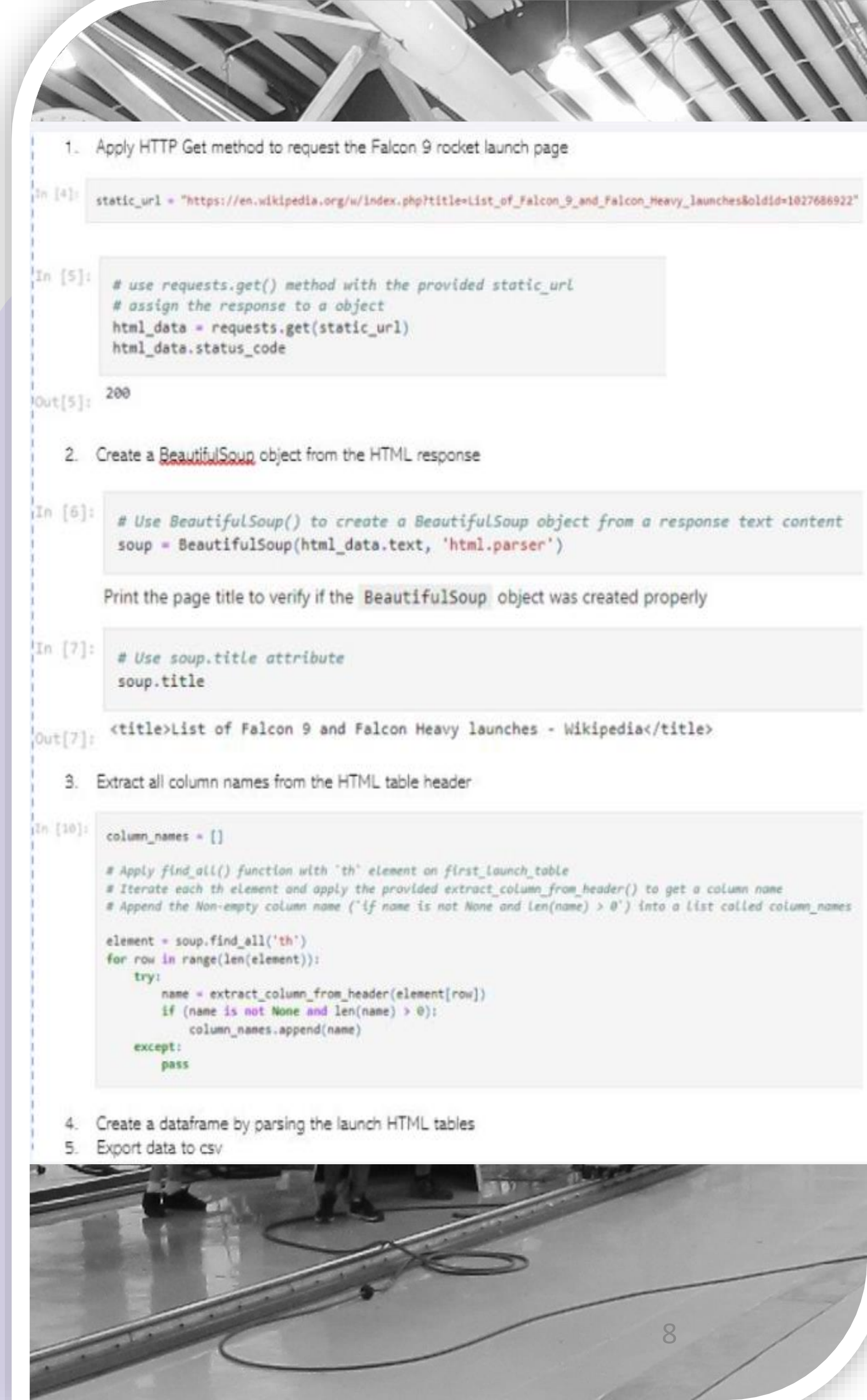
Organize Data into a Dictionary: Structure the enriched data into a well-organized dictionary. This step ensures that the data is systematically arranged, making subsequent transformations more straightforward.

Convert Dictionary to Dataframe: Transform the dictionary into a pandas dataframe. This conversion is crucial for performing data analysis and visualization, as dataframes provide a flexible and powerful structure for handling large datasets.

Filter for Falcon 9 Launches: Apply a filter to the dataframe to isolate only those launches conducted by the Falcon 9 rocket. Focusing on Falcon 9 launches ensures the analysis is relevant to SpaceX's reusable rocket technology.

Handle Missing Payload Mass Values: Address missing values in the Payload Mass column by calculating the mean of the available data and substituting missing entries with this mean value. This step ensures that the dataset remains comprehensive and avoids issues during analysis and modeling.

Export Processed Data to CSV File: Export the cleaned and processed dataframe to a CSV file. This step allows for easy sharing, storage, and further analysis of the dataset. CSV files are widely used and compatible with many data analysis tools and software.

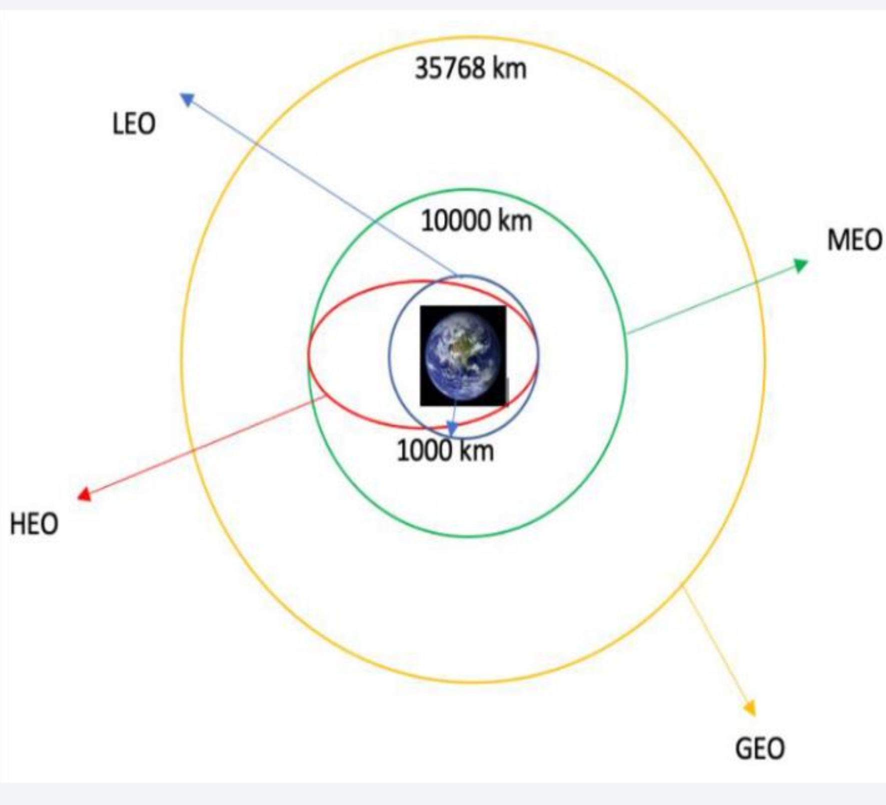


Data Wrangling

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

Steps

- **Perform EDA** and determine data labels
- **Calculate:**
  - The number of launches for each site.
  - The frequency of each orbit type.
  - The frequency of mission outcomes per orbit type.
- **Create Binary Landing Outcome Column:**
  - Add a column representing the binary landing outcome (dependent variable).
- **Export Data to CSV File:**
  - Save the processed data to a CSV file.
- **Landing Outcomes:**
  - **True Ocean:** Successful landing in a specific region of the ocean.
  - **False Ocean:** Unsuccessful landing in a specific region of the ocean.
  - **True RTLS (Return to Launch Site):** Successful landing on a ground pad.
  - **False RTLS:** Unsuccessful landing on a ground pad.
  - **True ASDS (Autonomous Spaceport Drone Ship):** Successful landing on a drone ship.
  - **False ASDS:** Unsuccessful landing on a drone ship.
- Outcomes are converted into binary format: 1 for a successful landing and 0 for an unsuccessful landing.



EDA with Visualization

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/jupyter-labs-eda-dataviz%20(1).ipynb

Plots Used:

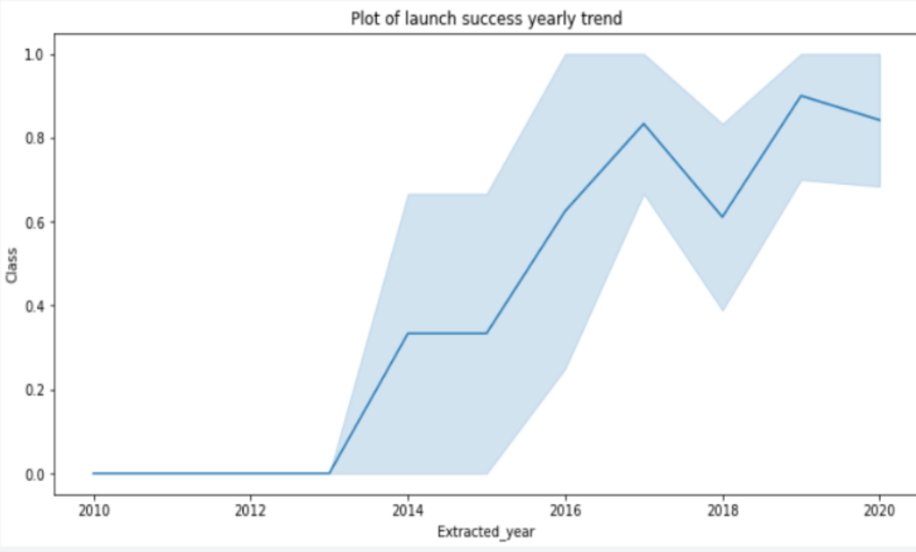
Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend

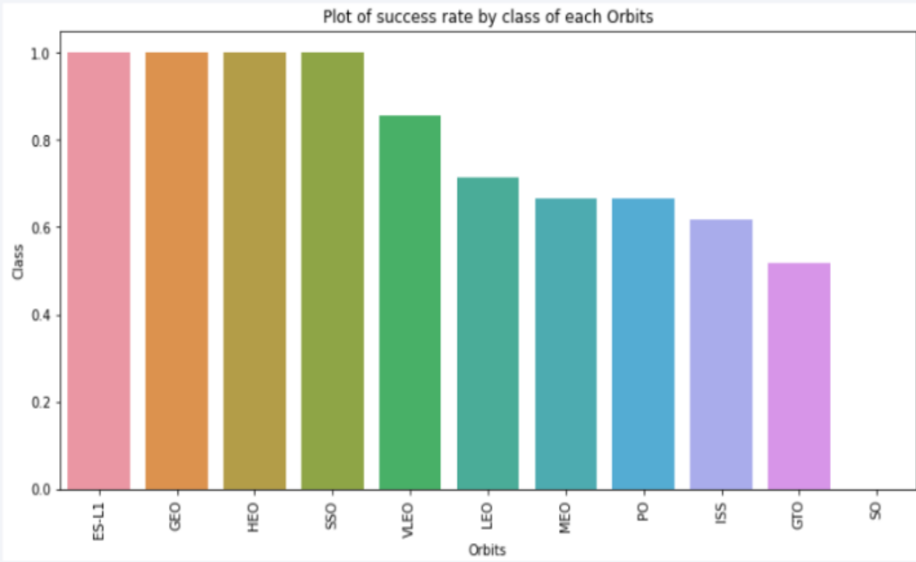
Charts

Flight Number vs. Payload,Flight Number vs. Launch Site,Payload Mass (kg) vs. Launch Site,Payload Mass (kg) vs. Orbit type

Analysis

- **View relationship** by using **scatter plots**. The variables could be useful for machine learning if a relationship exists
- Scatter plots, line charts, and bar plots were used to compare relationships between variables todecide if a relationship exists so that they could be used in training the machine learning model
- **Show comparisons** among discrete categories with **bar charts**. Bar charts show the relationships among the categories and a measured value.





EDA with SQL

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

Display:

Names of unique launch sites,5 records where launch site begins with ‘CCA’,Total payload mass carried by boosters launched by NASA (CRS),Average payload mass carried by booster version F9 v1.1.

List:

Date of first successful landing on ground pad,Names of boosters which had success landing on drone ship and have payload mass greater than 4,000 but less than 6,000,Total number of successful and failed missions,Names of booster versions which have carried the max payload,Failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015Count of landing outcomes between 2010-06-04 and 2017-03-20 (desc)

Build an interactive map with Folium

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/Week3_Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

Markers Indicating Launch Sites

- Added **blue circle** at **NASA Johnson Space Center's coordinate** with a **popup label** showing its name using its latitude and longitude coordinates
- Added **red circles** at **all launch sites coordinates** with a **popup label** showing its name using its name using its latitude and longitude coordinates

Colored Markers of Launch Outcomes

- Added **colored markers** of **successful** (**green**) and **unsuccessful** (**red**) **launches** at each launch site to show which launch sites have high success rates

Distances Between a Launch Site to Proximities

- Added **colored lines** to **show distance between** launch site **CCAFS SLC- 40 and** its proximity to the **nearest coastline, railway, highway, and city**

Build a Dashboard with Plotly

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/spacex_dash_app.py

Dropdown List with Launch Sites

- Allow user to select all launch sites or a certain launch site

Pie Chart Showing Successful Launches

- Allow user to see successful and unsuccessful launches as a percent of the total

Slider of Payload Mass Range

- Allow user to select payload mass range

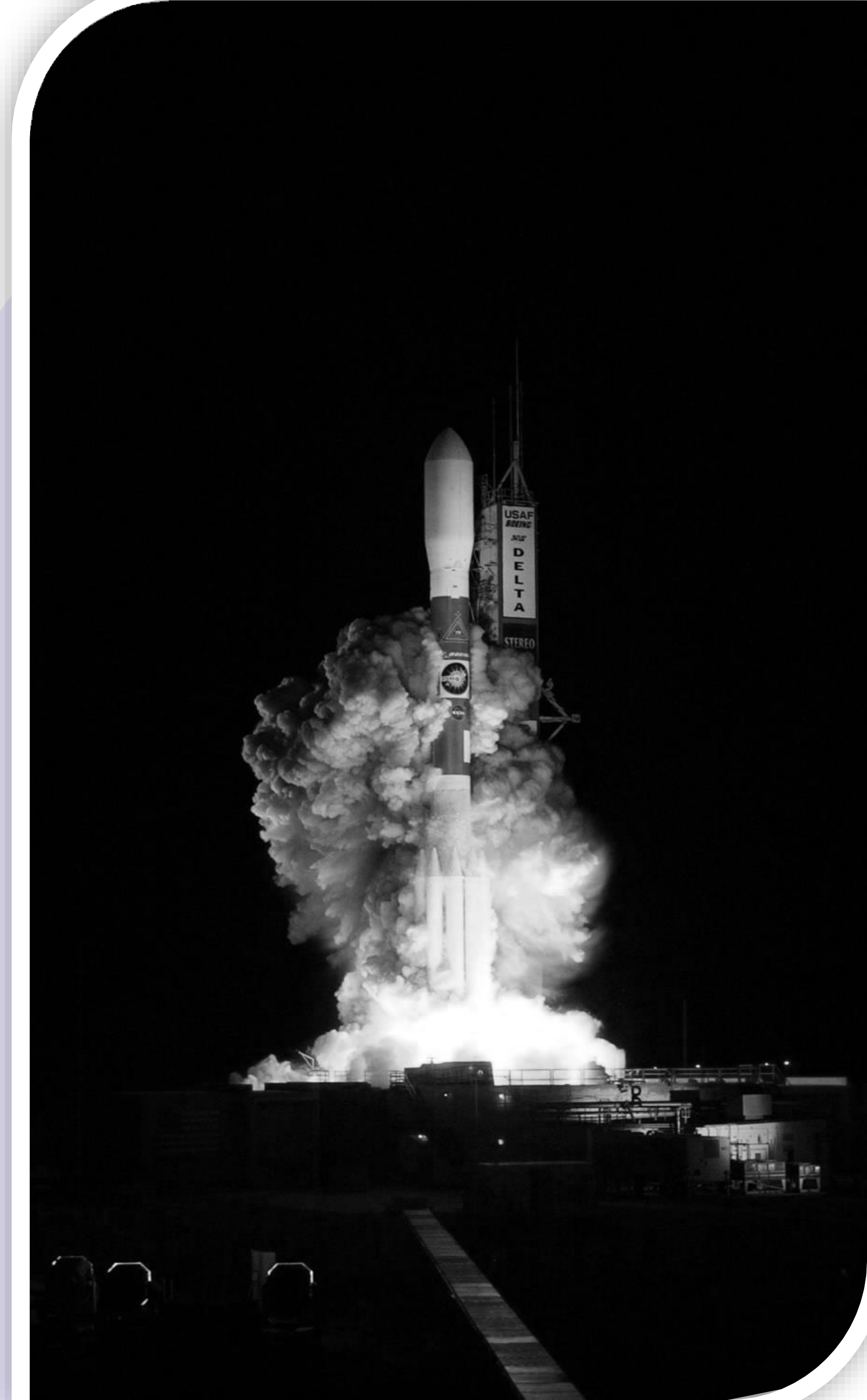
Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

- Allow user to see the correlation between Payload and Launch Success

Predictive Analytics

https://github.com/Nazreenn29/CourseraIBMDataScience/blob/main/Capstone/Machine%20Learning%20Prediction.ipynb

- **Create** NumPy array from the Class column
- **Standardize** the data with StandardScaler. Fit and transform the data.
- **Split** the data using train_test_split
- **Create** a GridSearchCV object with cv=10 for parameter optimization
- **Apply** GridSearchCV on different algorithms: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), K-Nearest Neighbor (KNeighborsClassifier())
- **Calculate** accuracy on the test data using .score() for all models
- **Assess** the confusion matrix for all models
- **Identify** the best model using Accuracy



Results Summary

Exploratory Data Analysis

- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

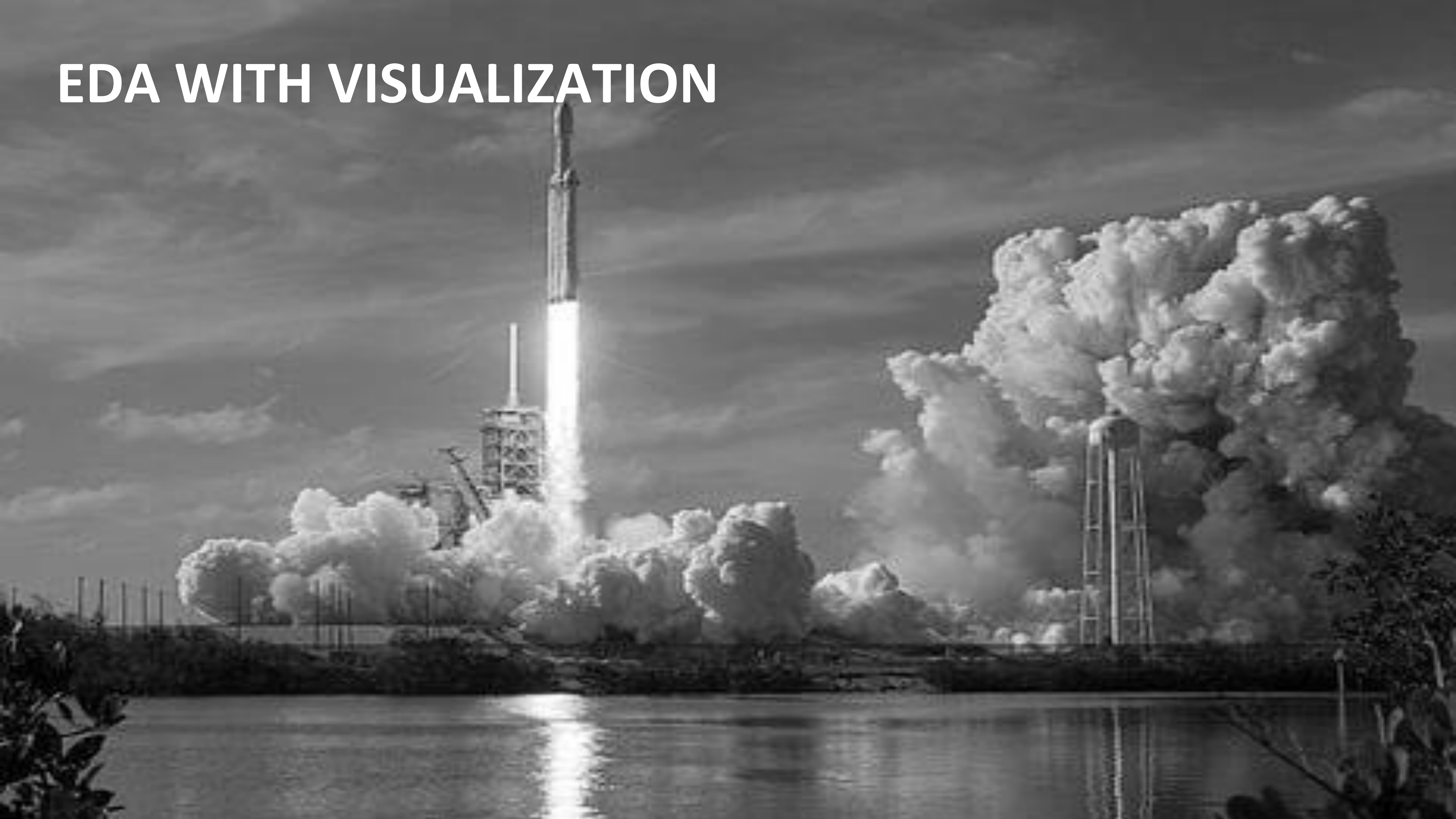
Visual Analytics

- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities

Predictive Analytics

- Decision Tree model is the best predictive model for the dataset

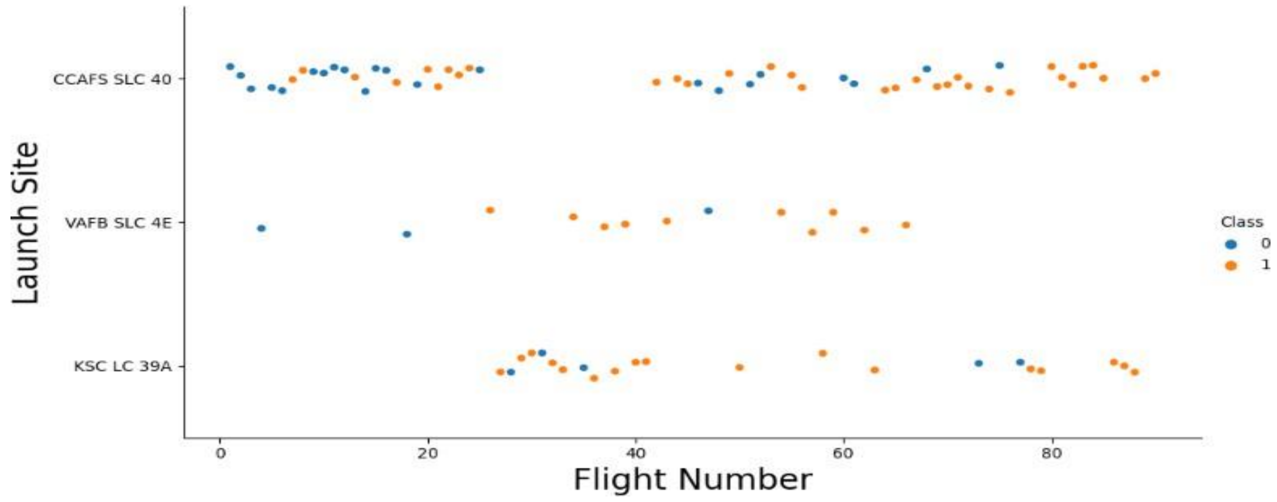
EDA WITH VISUALIZATION



Flight Number vs. Launch Site

Exploratory Data Analysis

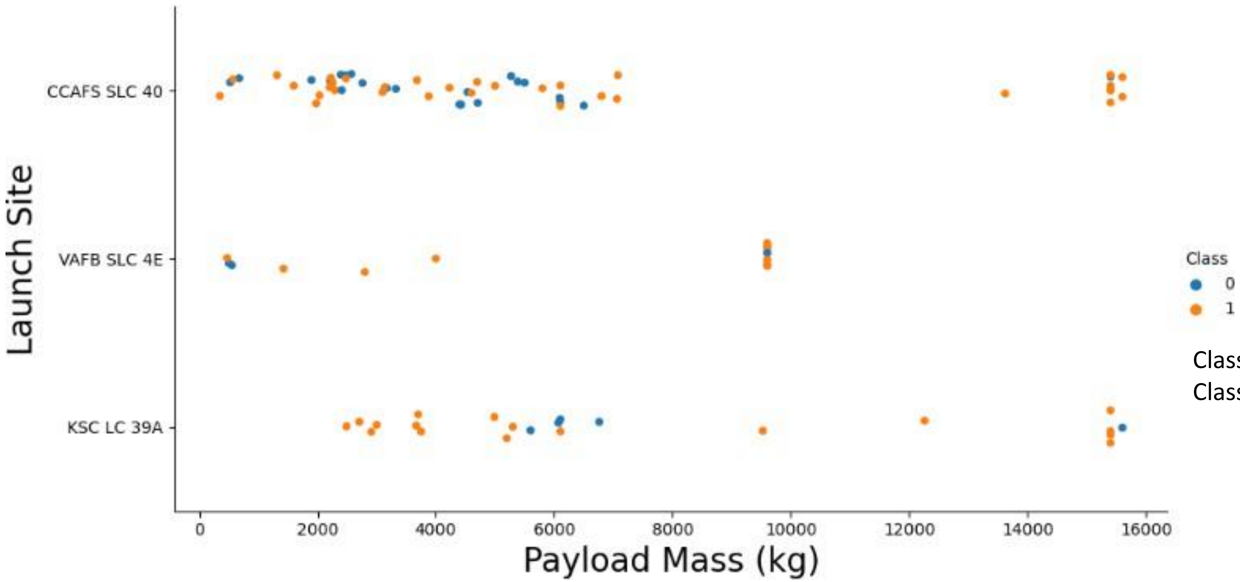
- **Earlier flights** had a **lower success rate** (**blue = fail**)
- **Later flights** had a **higher success rate** (**orange = success**)
- Around half of launches were from CCAFS SLC 40 launch site
- VAFB SLC 4E and KSC LC 39A have higher success rates
- We can infer that new launches have a higher success rate



Payload vs. Launch Site

Exploratory Data Analysis

- Typically, the **higher** the **payload mass** (kg), the **higher** the **success rate**
- Most launces with a payload greater than 7,000 kg were successful
- KSC LC 39A has a 100% success rate for launches less than 5,500 kg
- VAFB SKC 4E has not launched anything greater than ~10,000 kg

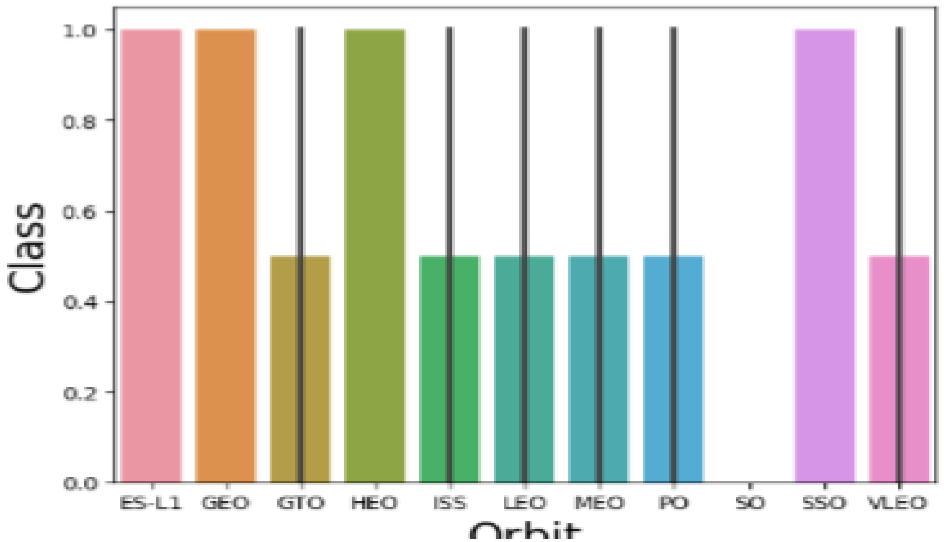


Class 0 = Fail
Class 1 = Success

Success Rate by Orbit

Exploratory Data Analysis

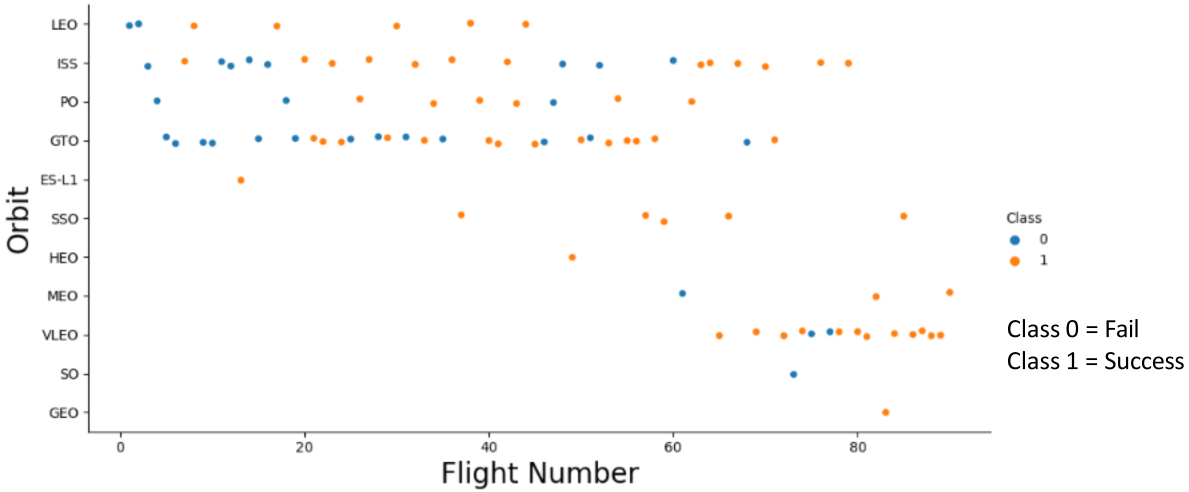
- **100% Success Rate**: ES-L1, GEO, HEO and SSO
- **50%-80% Success Rate**: GTO, ISS, LEO, MEO, PO
- **0% Success Rate**: SO



Flight Number vs. Orbit

Exploratory Data Analysis

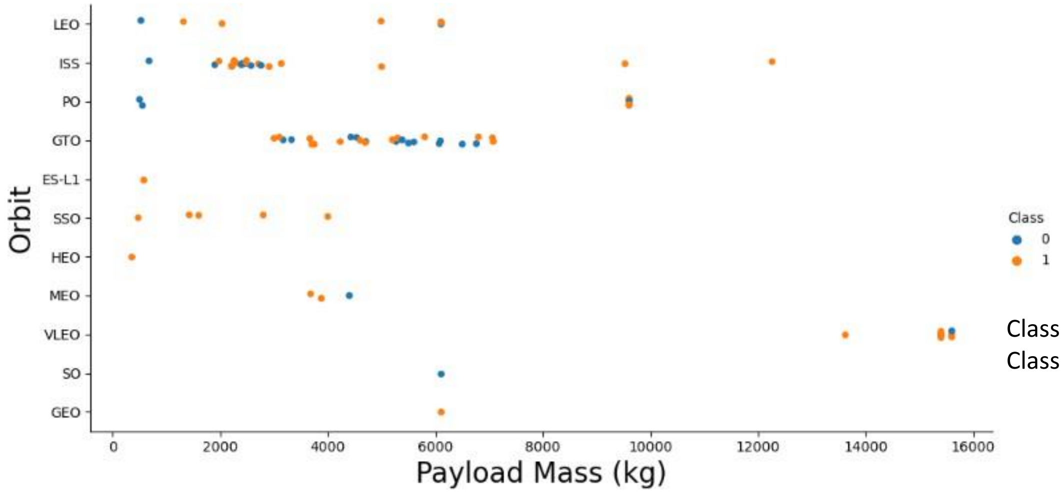
- The success rate typically increases with the number of flights for each orbit
- This relationship is highly apparent for the LEO orbit
- The GTO orbit, however, does not follow this trend



Payload vs. Orbit

Exploratory Data Analysis

- Heavy payloads are better with LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads

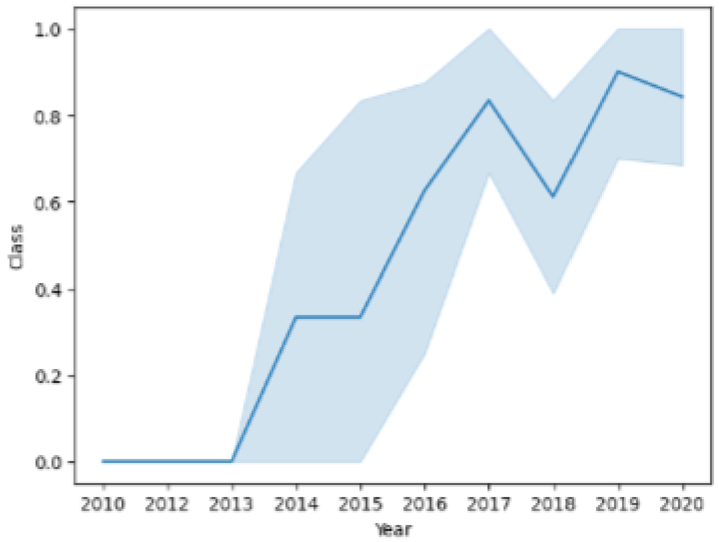


Class 0 = Fail
Class 1 = Success

Launch Success over Time

Exploratory Data Analysis

- The success rate improved from 2013-2017 and 2018-2019
- The success rate decreased from 2017-2018 and from 2019-2020
- Overall, the success rate has improved since 2013



EDA WITH SQL

All Launch Site Names

Query unique launch site names from database.

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same launch site with data entry errors.
CCAFS LC-40 was the previous name. Likely only 3 unique launch_site values: CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

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 Beginning with `CCA

First five entries in database with Launch Site name beginning 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 Payload Mass from NASA

This query sums the total payload mass in kg where NASA was the customer.

CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

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

This query calculates the average payload mass or launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of our payload mass range

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

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

t[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Pad Landing

This query returns the first successful ground pad landing date.

First ground pad landing wasn't until the end of 2015.

Successful landings in general appear starting 2014.

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

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload Between 4000 and 6000

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

]:

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

Total Number of Each Mission Outcome

This query returns a count of each mission outcome.

```
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:
```

	failureoutcome
0	1

Boosters that Carried Maximum Payload

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

	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

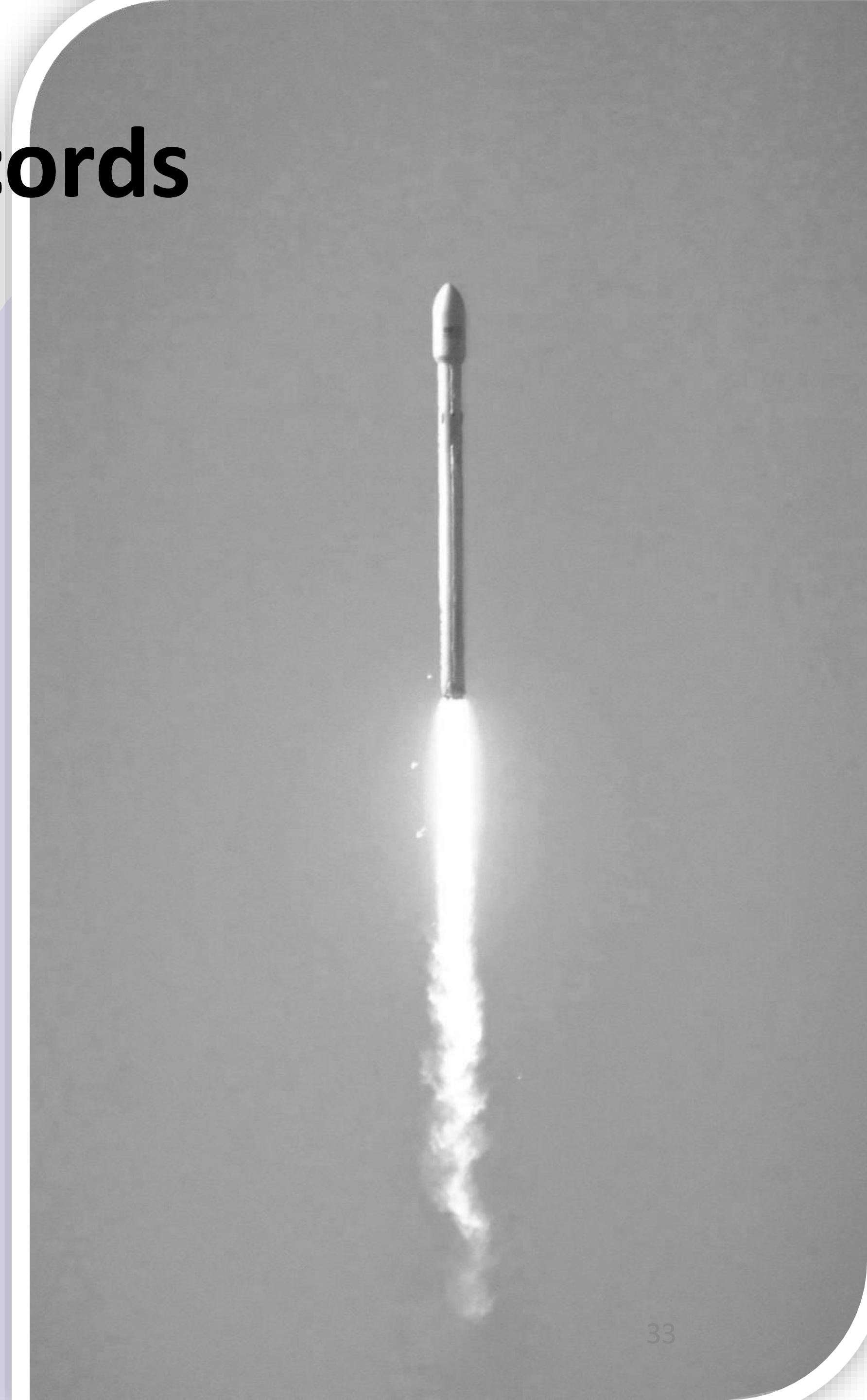
2015 Failed Drone Ship Landing Records

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

```
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)
```

	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)



Ranking Counts of Successful Landings Between 2010-06-04 and 2017-03-20

```
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)
```

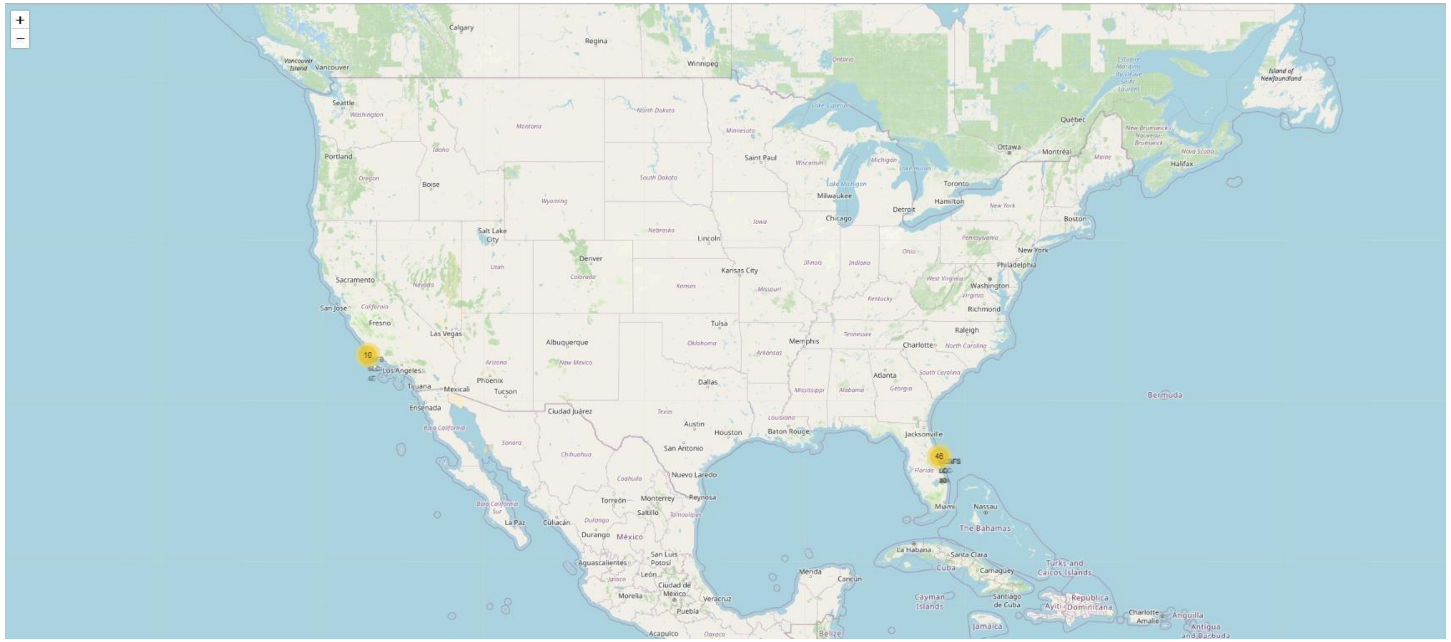
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

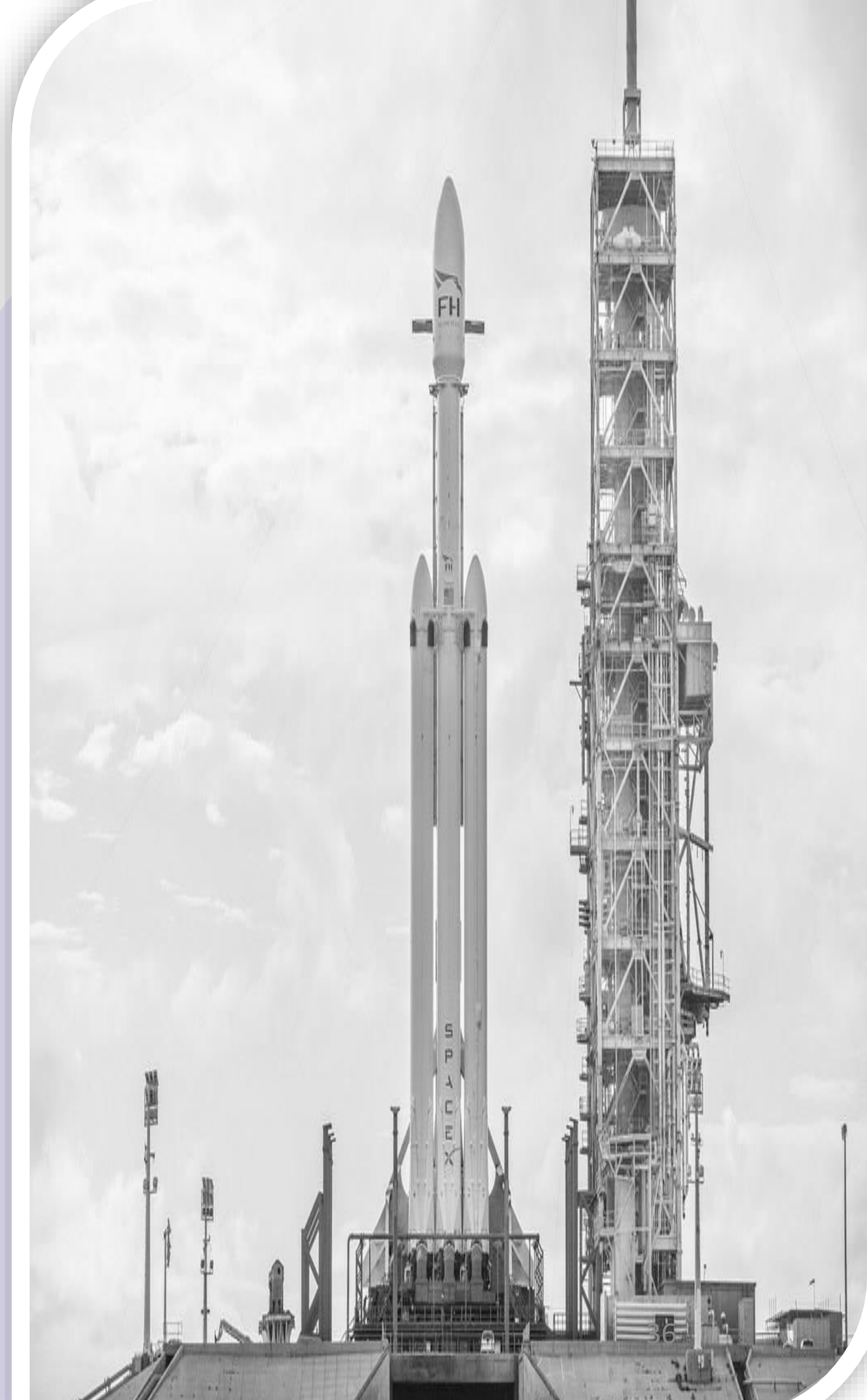
Interactive Mapwith Folium

Launch Sites

With Markers

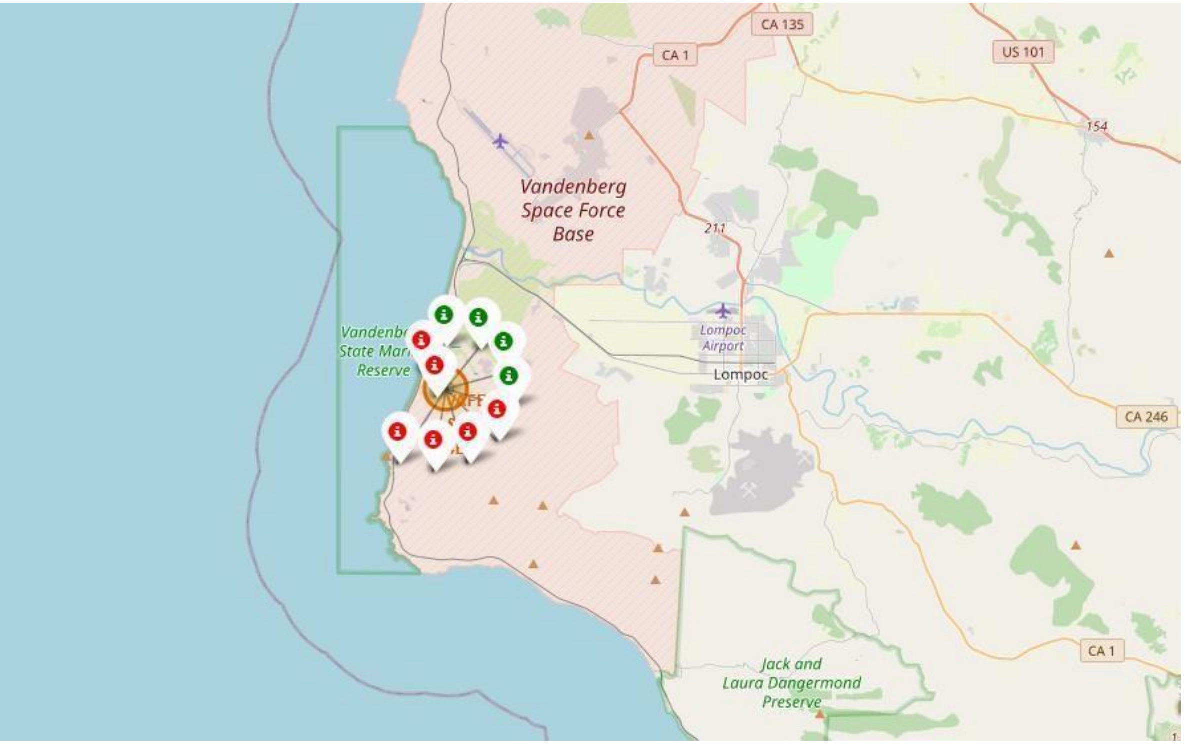
- **Near Equator**: Launching rockets from sites close to the equator provides significant advantages. The proximity to the equator facilitates launches to equatorial orbits and benefits from Earth's rotational speed for prograde orbits. Rockets launched from these locations receive a natural boost from the Earth's rotation, reducing the need for additional fuel and boosters, ultimately saving on launch costs.





Launch Outcomes

Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

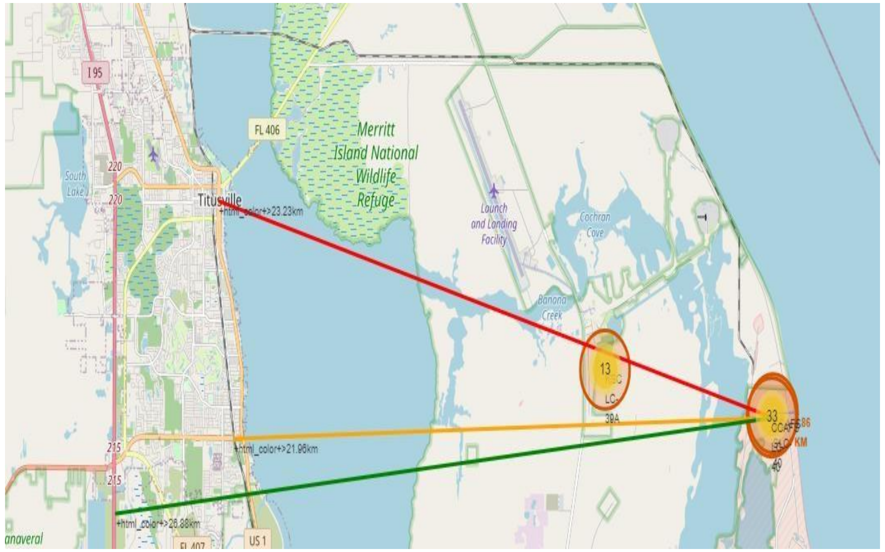


Distance to Proximities

Coasts: Launch sites located near coastlines help ensure that spent rocket stages or failed launches fall safely into the ocean, minimizing the risk to people and property.

Safety/Security: An exclusion zone around the launch site is necessary to keep unauthorized personnel away and ensure the safety of the surrounding areas.

Transportation/Infrastructure and Cities: The launch site should be positioned far enough from populated areas to avoid potential damage from failed launches, yet close enough to transportation infrastructure—such as roads, railways, and docks—to facilitate the transport of personnel and materials essential for launch operations.




```
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, closest_highway[0], closest_highway[1])
print('distance_highway =',distance_highway, ' km')
distance_railroad = calculate_distance(launch_site_lat, launch_site_lon, closest_railroad[0], closest_railroad[1])
print('distance_railroad =',distance_railroad, ' km')
distance_city = calculate_distance(launch_site_lat, launch_site_lon, closest_city[0], closest_city[1])
print('distance_city =',distance_city, ' km')
```

```
distance_highway = 0.5834695366934144  km
distance_railroad = 1.2845344718142522  km
distance_city = 51.43416999517233  km
```

Dashboard with Plotly

Launch Success by Site

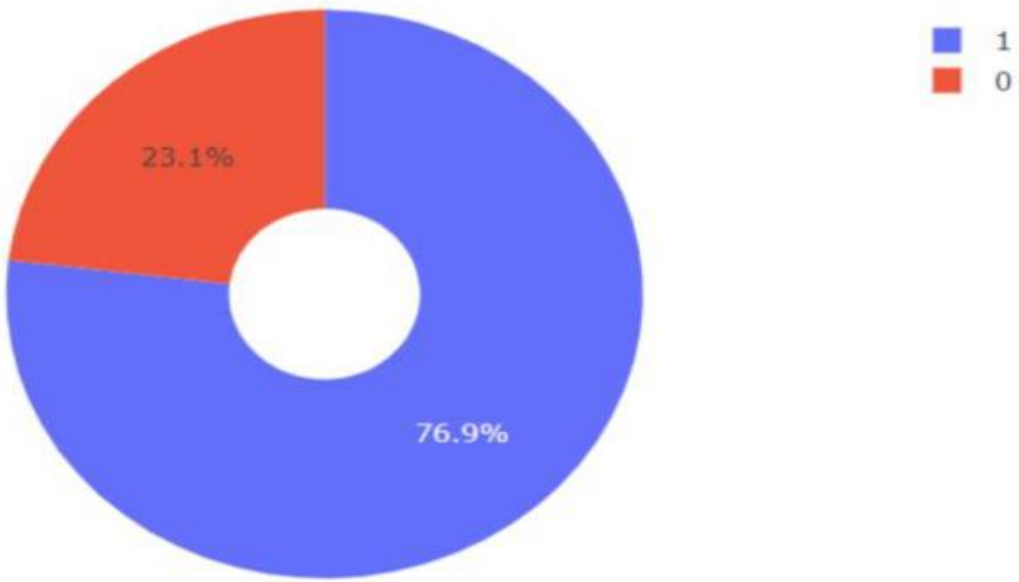
Success as Percent of Total

- **KSC LC-39A** has the **most successful launches** amongst launch sites (**41.2%**)

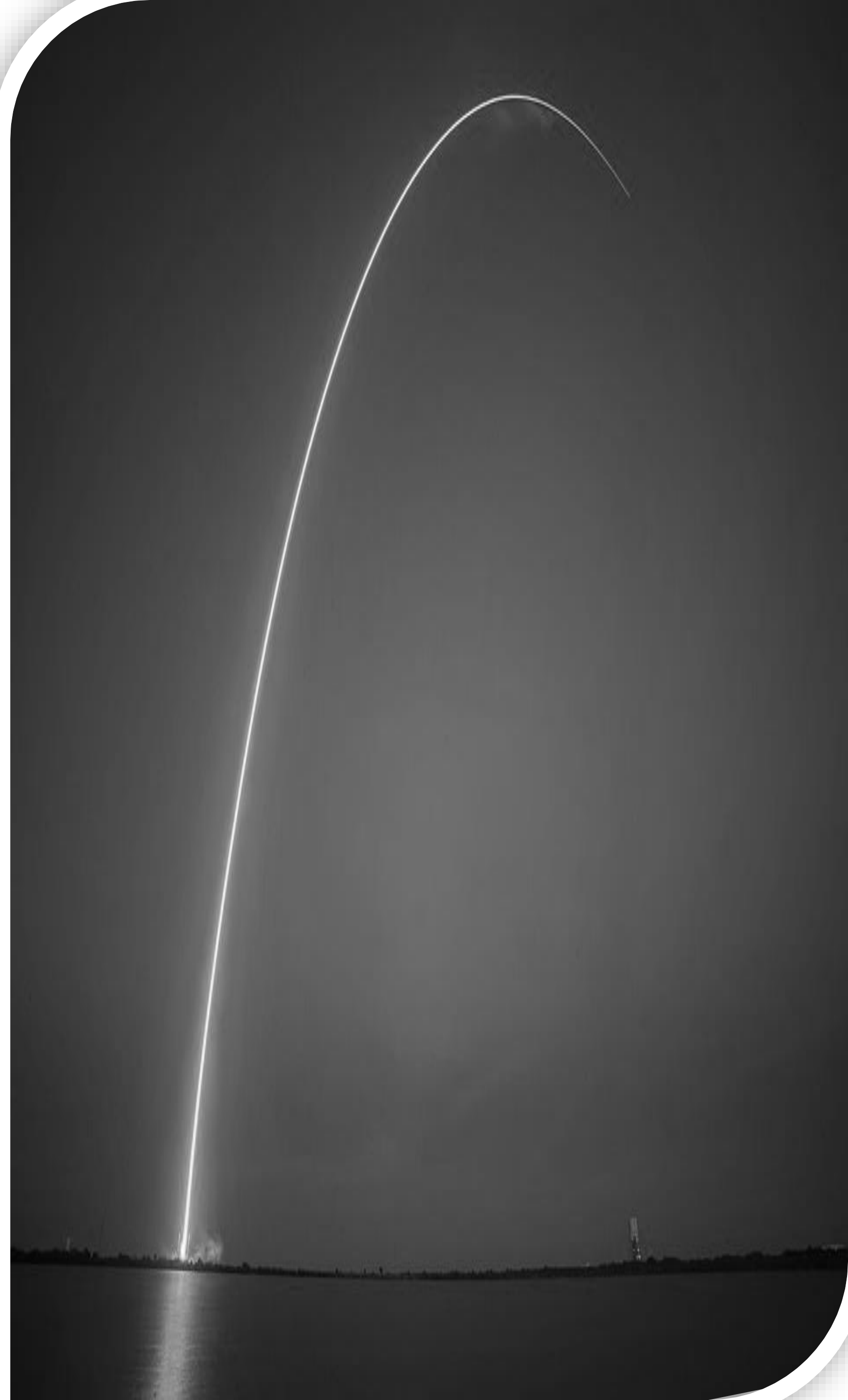
Launch Success (KSC LC-29A)

Success as Percent of Total

- **KSC LC-39A** has the **highest success rate** amongst launch sites (**76.9%**)
- 10 successful launches and 3 failed launches



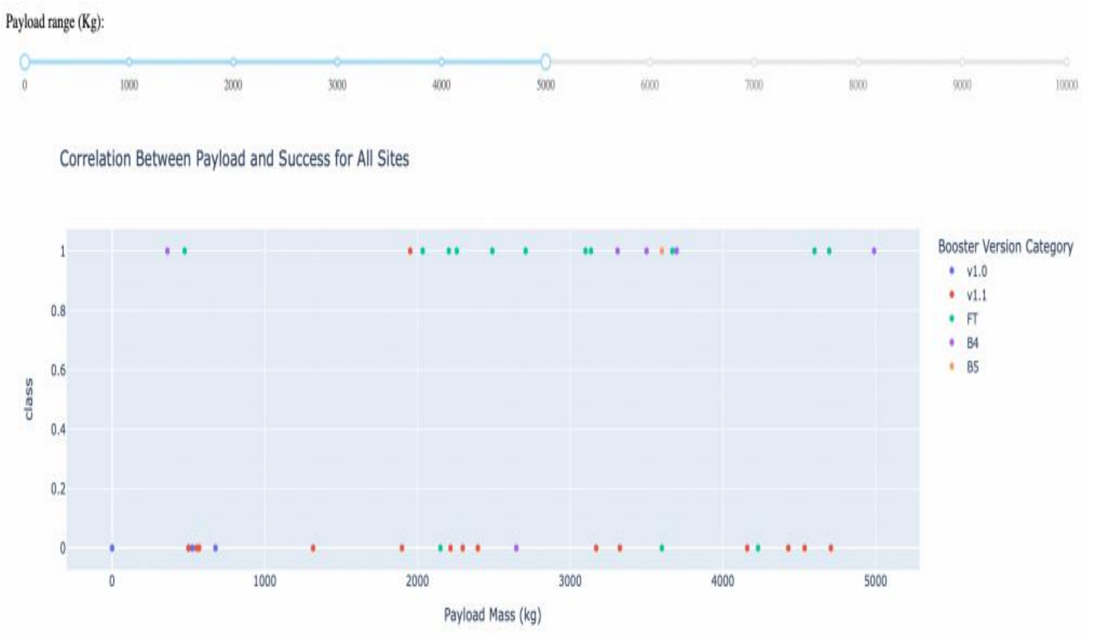
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

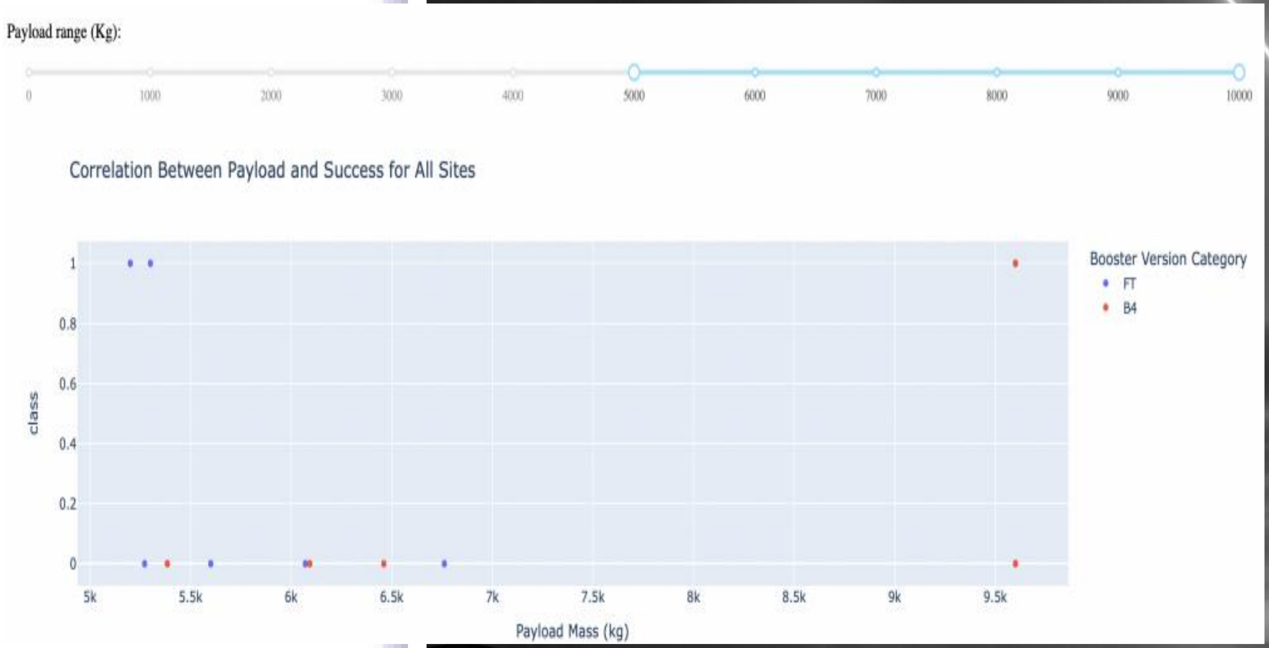


Payload Mass and Success

By Booster Version

- **Payloads between 2,000 kg** and **5,000 kg** have the **highest success rate**
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome





Predictive Analytics

Classification

Accuracy

- **All** the **models** performed at about the same level and had the **same scores** and **accuracy**. This is likely due to the **small dataset**. The **Decision Tree model slightly outperformed** the rest when looking at .best_score_
- .best_score_ is the average of all cv folds for a single combination of the parameters

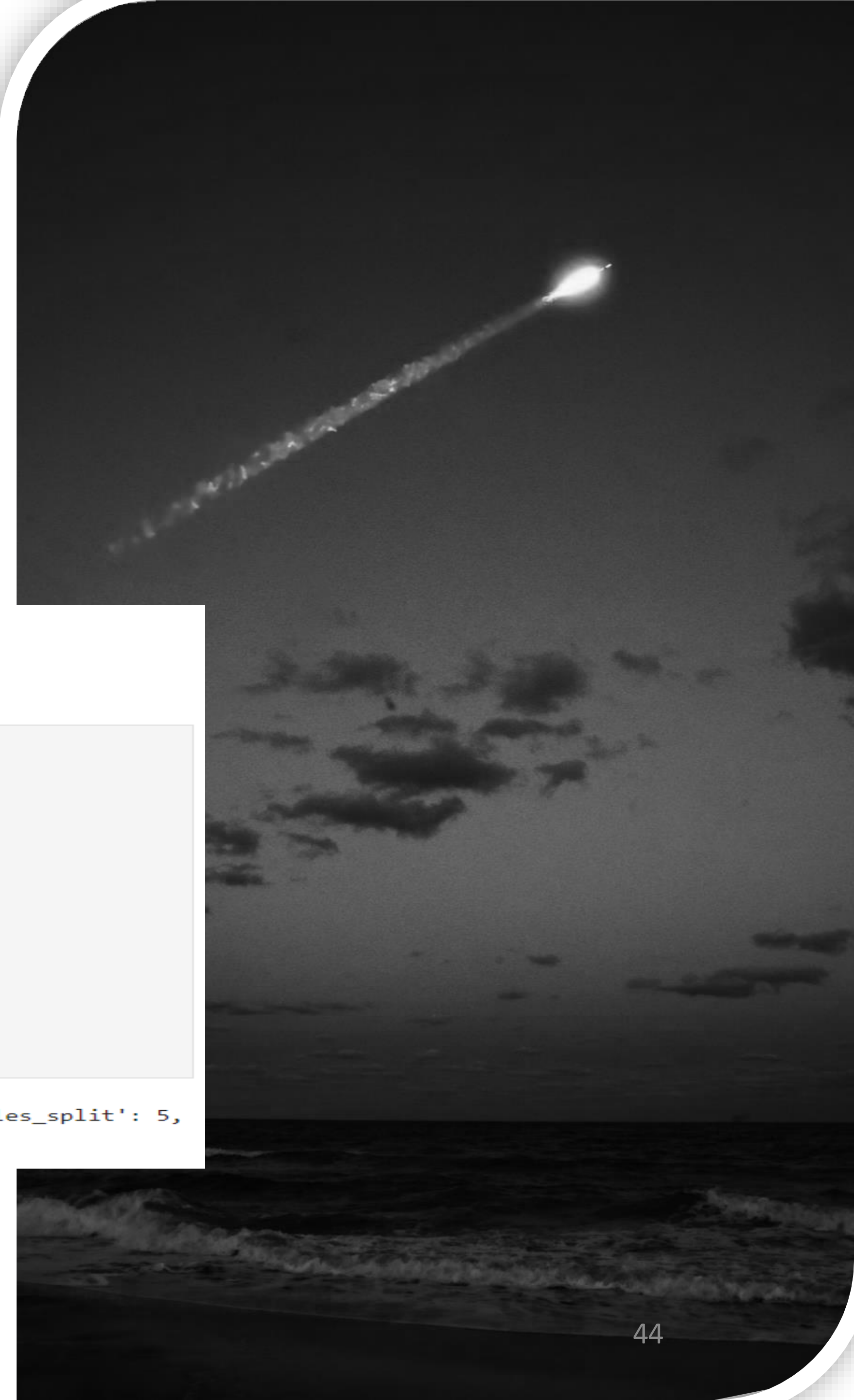
TASK 12

Find the method performs best:

```
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_)
```

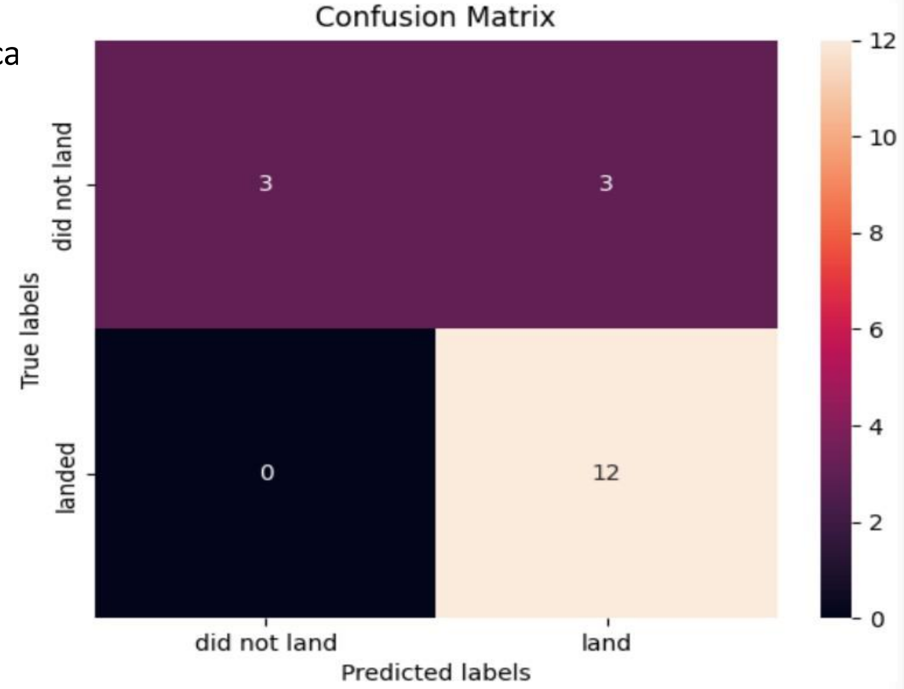
```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'random'}
```



Confusion Matrices

Performance Summary

- A confusion matrix summarizes the performance of a classification algorithm
- All the confusion matrices were identica
- Confusion Matrix Outputs:
  - 12 True positive
  - 3 True negative
  - **3 False positive**
  - 0 False Negative
- **Precision** = TP / (TP + FP)
  - 12 / 15 = .80
- **Recall** = TP / (TP + FN)
  - 12 / 12 = 1
- **F1 Score** = 2 * (Precision * Recall) / (Precision + Recall)
  - 2 * (.8 * 1) / (.8 + 1) = .89
- **Accuracy** = (TP + TN) / (TP + TN + FP + FN) = .833



Conclusion

Research Findings
Model Performance: The models showed similar performance on the test set, with the decision tree model slightly outperforming the others.
Equator: Most launch sites are situated near the equator to take advantage of the Earth's rotational speed, providing a natural boost that reduces the need for additional fuel and boosters.
Coast: All launch sites are located near coastlines.
Launch Success: The success rate of launches has increased over time.
KSC LC-39A: This launch site has the highest success rate, achieving a 100% success rate for launches with payloads less than 5,500 kg.
Orbits: ES-L1, GEO, HEO, and SSO orbits have a 100% success rate.
Payload Mass: Across all launch sites, higher payload masses (kg) are associated with higher success rates.

Appendix

GitHub repository url:

https://github.com/Nazreenn29/CourseraIBMDataScience/tree/main/Capstone

Instructors:

Instructors: Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors