



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamed Goma
8th September 2025



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 in screenshots
 - Predictive Analytics result

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches at a cost of \$62 million, while other providers charge upwards of \$165 million per launch. A major reason for SpaceX's lower cost is its ability to reuse the first stage of the rocket. Therefore, being able to predict whether the first stage will land successfully is directly linked to estimating the overall cost of a launch. This information can also be valuable to competitors looking to bid against SpaceX. The objective of this project is to build a machine learning pipeline capable of predicting the likelihood of a successful first-stage landing.

Problems you want to find answers

- What factors influence whether the rocket's first stage lands successfully?
- How do different features interact to affect landing success rates?
- What operating conditions are necessary to maximize the chances of a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
 - One-hot encoding was applied to 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

- The dataset was compiled using multiple methods. First, data was retrieved from the SpaceX API via GET requests. The response content was decoded into JSON format using the `.json()` function and then converted into a pandas DataFrame with `json_normalize()`. Next, the dataset underwent a data cleaning process, where missing values were identified and filled where appropriate. Additionally, web scraping was performed on Wikipedia's Falcon 9 launch records using BeautifulSoup. The launch history tables were extracted from the HTML, parsed, and transformed into pandas DataFrames for further analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```


Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

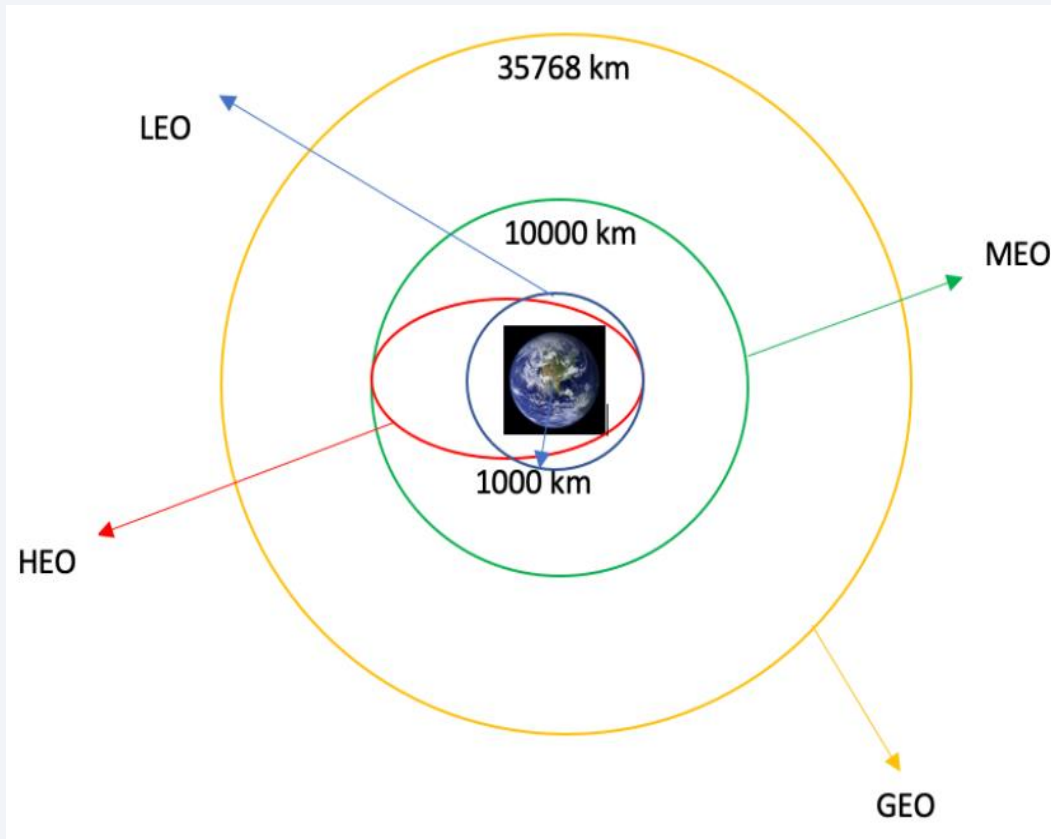
```
response.status_code
```

```
200
```

```
# Use json_normalize meethod to convert the json result into a dataframe
jlist = requests.get(static_json_url).json()
df2 = pd.json_normalize(jlist)
df2.head()
```

```
df = pd.read_json(static_json_url)
```

Data Wrangling

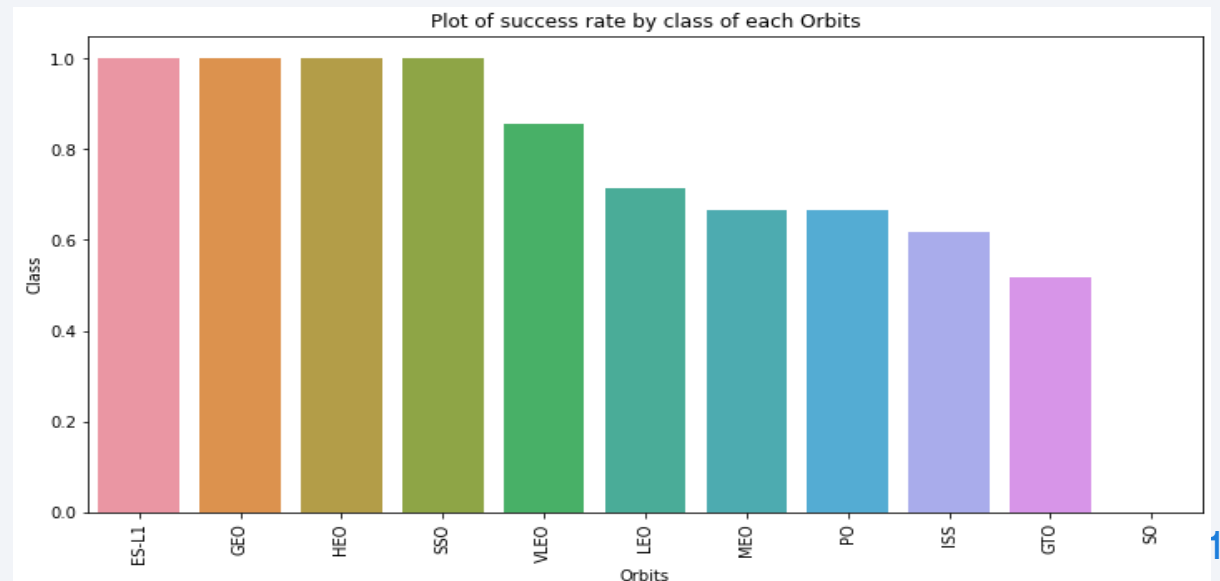
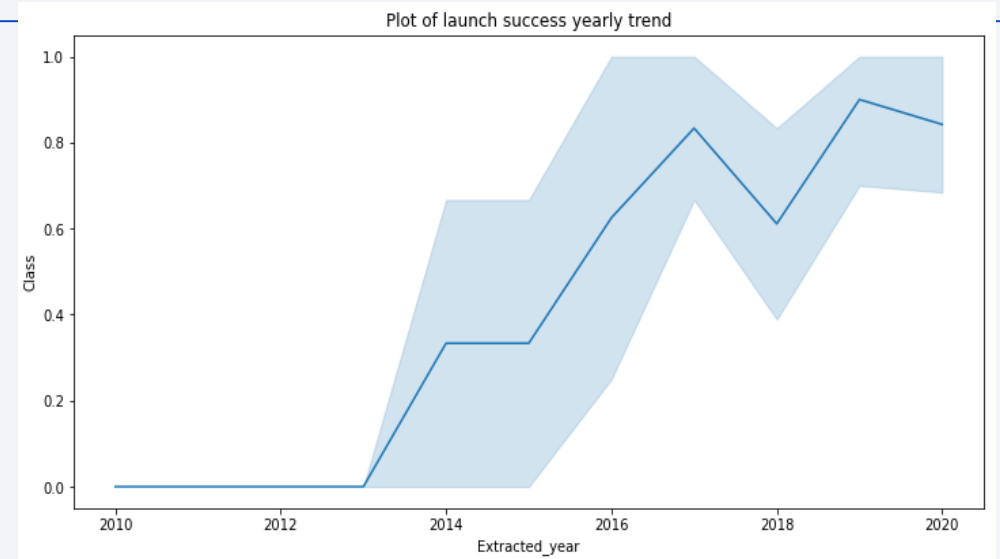


- We conducted exploratory data analysis (EDA) to better understand the dataset and define the training labels. This included analyzing the number of launches per site as well as the frequency and distribution of different orbits. From the outcome column, we derived a landing outcome label to serve as the target variable. The processed results were then exported to a CSV file for subsequent modeling.

EDA with Data Visualization

We explored the dataset by creating visualizations to examine key relationships, including:

- Flight number vs. launch site
- Payload vs. launch site
- Success rate by orbit type
- Flight number vs. orbit type
- Yearly trends in launch success



EDA with SQL

The SpaceX dataset was loaded into a PostgreSQL database directly from within the Jupyter Notebook environment. We then performed exploratory data analysis (EDA) using SQL queries to extract insights, such as:

- Identifying the unique launch sites used in space missions
- Calculating the total payload mass carried by boosters launched under NASA (CRS) missions
- Determining the average payload mass for the booster version F9 v1.1
- Counting the total number of successful and failed mission outcomes
- Retrieving details of failed drone ship landings, including the associated booster versions and launch site names

Build an Interactive Map with Folium

We visualized the launch sites using Folium maps, adding markers, circles, and lines to indicate the success or failure of launches at each location. Launch outcomes were encoded as binary classes: **0** for failure and **1** for success.

By leveraging color-coded marker clusters, we identified which launch sites exhibited higher success rates. Additionally, we computed the distances from each launch site to nearby infrastructure and landmarks, addressing questions such as:

- Are launch sites located near railways, highways, or coastlines?
- Do launch sites maintain a minimum distance from urban areas?

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

Results

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

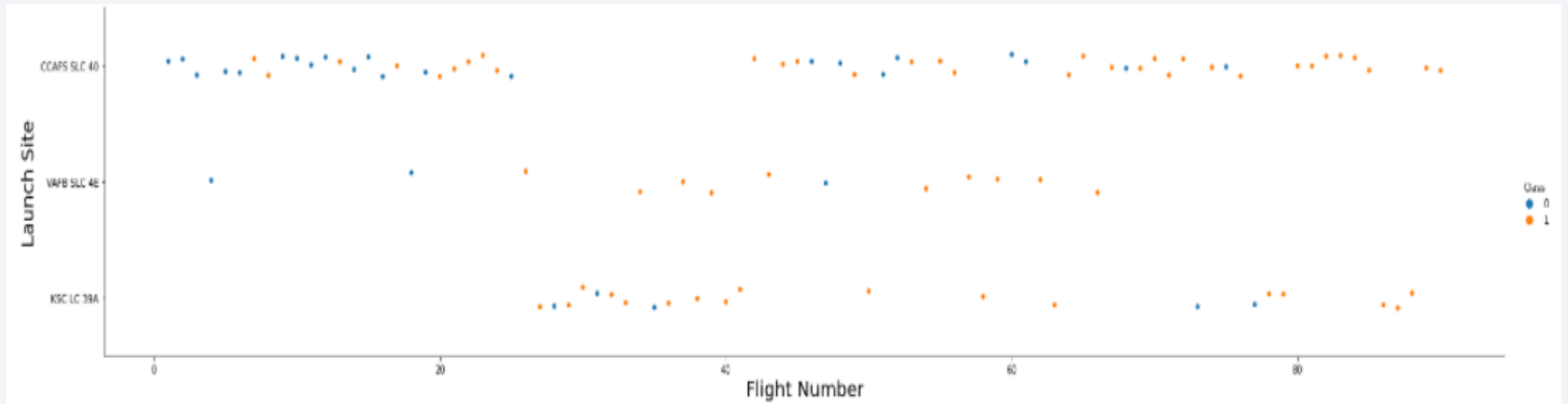
The background of the slide is a complex, abstract composition. It features a dark blue base color on the left, which transitions into a vibrant, multi-colored area on the right. This transition is achieved through a series of diagonal, overlapping bands and streaks in shades of red, teal, and light blue. A fine, grid-like pattern is visible throughout the image, particularly in the teal and red areas, giving it a digital or data-driven appearance. The overall effect is one of dynamic movement and high-tech aesthetics.

Section 2

Insights drawn from EDA

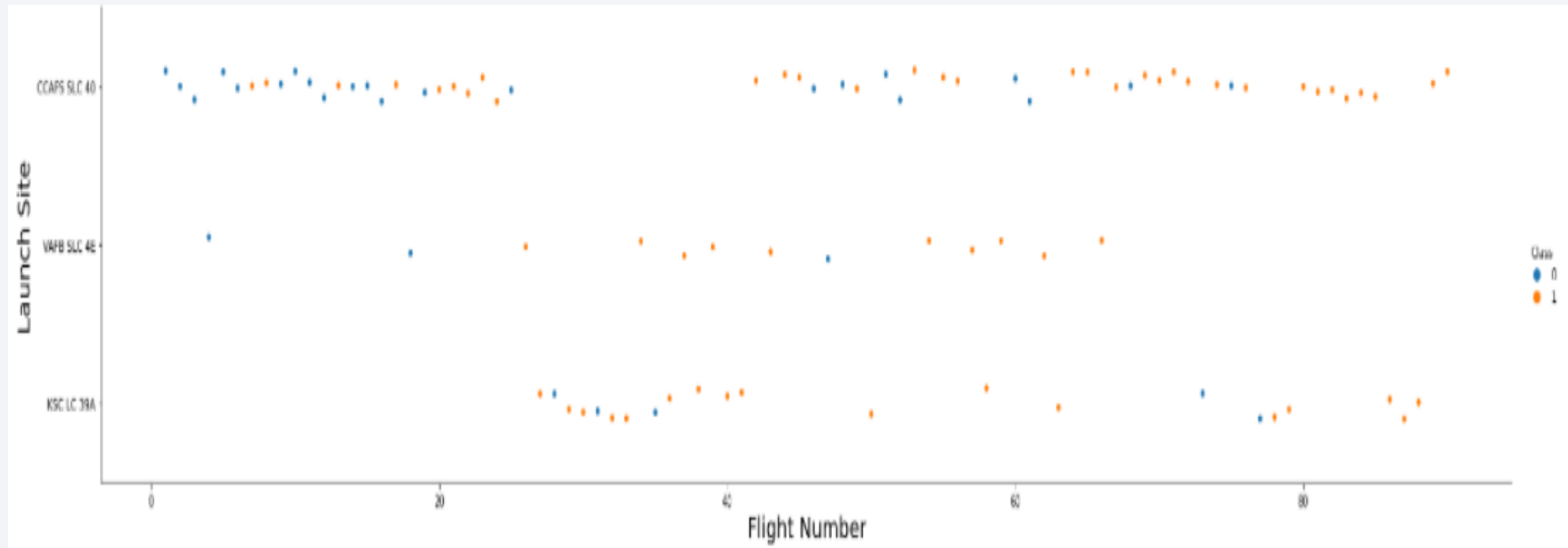
Flight Number vs. Launch Site

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



Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



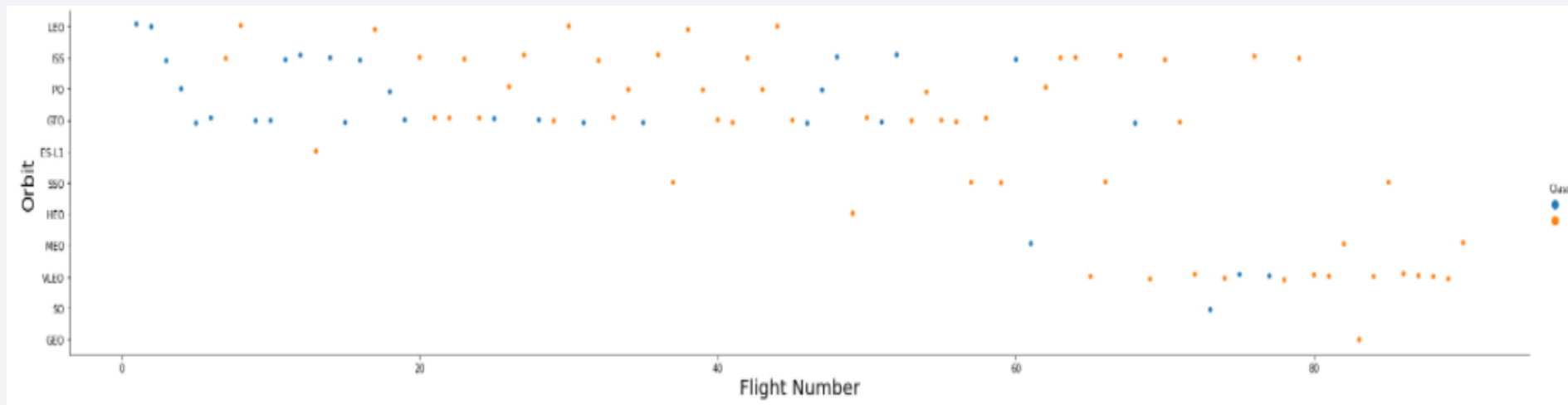
Success Rate vs. Orbit Type

- 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 illustrates the relationship between flight number and orbit type. In the case of LEO orbits, success appears to increase with the number of flights. However, for GTO orbits, no clear relationship is observed between flight number and launch success.



Payload vs. Orbit Type

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



Launch Success Yearly Trend

- The plot shows that the success rate consistently improved from 2013 to 2020.



All Launch Site Names

- We used 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

```
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

- We used the query above to display 5 records where launch sites begin with 'CCA'

Total Payload Mass

- We calculated the total payload carried by boosters from NASA resulting in 45596 using the query below

```
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

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
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

- We applied a **WHERE** to filter boosters that successfully landed on a drone ship and used an **AND** condition to further restrict the results to landings with a payload mass between **4,000 and 6,000**.

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

- We used the wildcard operator (%) in the **WHERE** clause to filter records where the **MissionOutcome** indicated either a success or a failure.

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** aggregate function.

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

2015 Launch Records

- We combined **the WHERE, LIKE, AND, and BETWEEN** clauses to filter for failed drone ship landings in 2015, retrieving the corresponding booster versions and launch site names.

```
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

- We queried the dataset to select landing outcomes and their counts, applying a **WHERE** clause to filter records between 2010-03-20 and 2010-06-04. The results were then grouped by landing outcome using **GROUP BY** and sorted in descending order with **ORDER BY**.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky with stars and a view of the Earth's surface from space. The Earth's surface is mostly dark, with a dense network of yellow and orange lights representing city lights at night. The lights are concentrated in the lower right portion of the image, following the curve of the Earth. The upper portion of the image shows the dark blue sky with a few stars.

Section 4

Launch Sites Proximities Analysis

All launch sites global map markers



launch sites map SpaceX in Florida & California

Markers showing launch sites with color labels



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

The background of the slide is a close-up, artistic photograph of a printed circuit board (PCB). The board is dark, and the intricate circuitry is highlighted with a vibrant red glow. Numerous small, circular components, likely solder joints or micro-components, are visible along the traces, some of which are also glowing. The lighting creates a sense of depth and technological sophistication.

Section 5

Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

Total Success Launches By all sites



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

Pie chart showing the Launch site with the highest launch success ratio



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

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

- The decision tree classifier model got 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 of the decision tree classifier indicates that it can differentiate between the classes. However, the main issue lies in the false positives cases where unsuccessful landings are incorrectly classified as successful ones.



Conclusions

- In conclusion, the analysis shows that launch sites with a **higher** number of flights tend to achieve **greater success rates**. The overall launch success rate steadily increased from 2013 to 2020. Among the different orbits, ES-L1, GEO, HEO, SSO, and **VLEO demonstrated the highest success rates**, while KSC LC-39A recorded the greatest number of successful launches. Furthermore, the **Decision Tree Classifier** proved to be the most effective machine learning algorithm for this task.

Thank you!

