IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Neel Patel
10/17/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Analytics Demo
  - Predictive Analysis Results

Introduction

- Project background and context

The era of commercial space travel has arrived, with companies like SpaceX leading the way in making space exploration more affordable. SpaceX's Falcon 9 rocket is priced at $62 million per launch, significantly lower than competitors, whose costs often exceed $165 million. This cost efficiency is largely due to SpaceX's ability to reuse the first stage of the Falcon 9.

This project aims to predict the likelihood of a successful landing for the Falcon 9's first stage. Understanding this outcome is essential for determining launch costs and for potential competitors looking to bid against SpaceX.

- Problems you want to find answers
  - What correlations exist between rocket variables and the successful landing rate of the Falcon 9's first stage?
  - What conditions optimize the chances of achieving the highest successful landing rate?

Section 1

Methodology

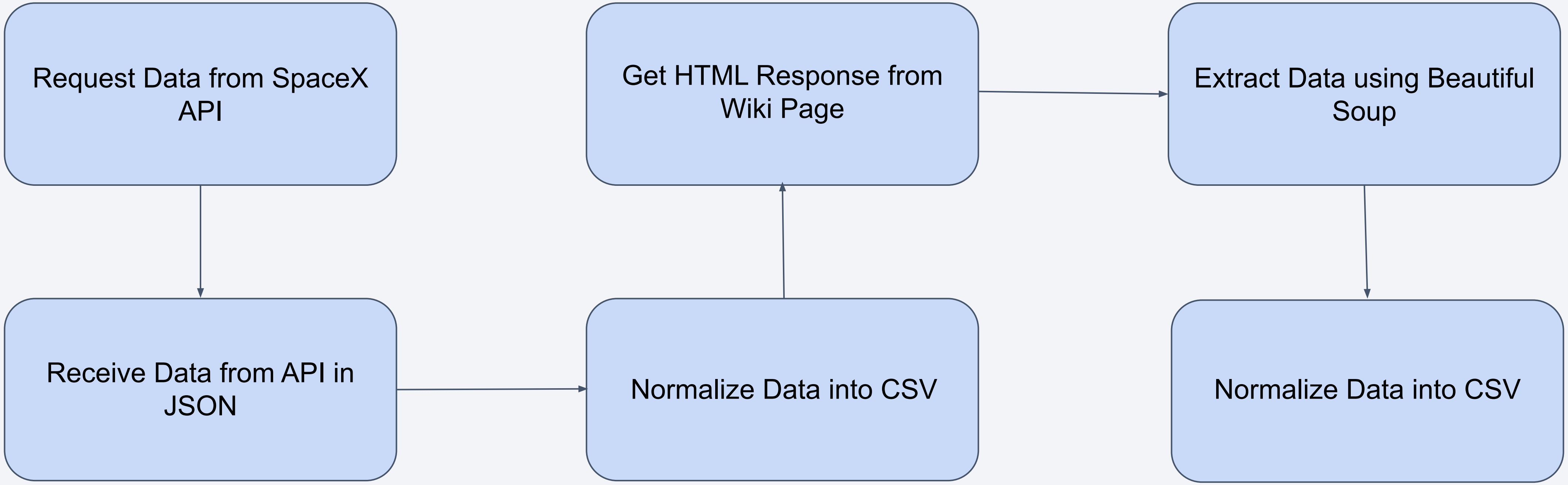
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Convert the outcomes into training groups with boosters landed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

- The data collection process involves gathering information through two primary methods: **API requests** to the SpaceX API and **web scraping** from a Wikipedia page containing launch records for SpaceX, Falcon 9, and Falcon Heavy.



Data Collection – SpaceX API

1.Getting response from HTML

```
html_data = requests.get(static_url).text
```

2.Creating a BeautifulSoup object

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3.Finding all tables and assigning the result to a list

```
html_tables = soup.find_all('table')
```

4.Extracting column name one by one

```
column_names = []
```

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

5.Creating an empty dictionary with keys

```
launch_dict= dict.fromkeys(column_names)
```

```
# Remove an irrelvant column
del launch_dict['Date and time ( )']
```

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6.Filling up the `launch_dict` with launch records
(Too long to put in here, so please refer to the notebook)

7.Creating a Dataframe and exporting it to a CSV

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Collection - Scraping

1.Getting response from HTML

```
html_data = requests.get(static_url).text
```

2.Creating a BeautifulSoup object

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3.Finding all tables and assigning the result to a list

```
html_tables = soup.find_all('table')
```

4.Extracting column name one by one

```
column_names = []
```

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

5.Creating an empty dictionary with keys

```
launch_dict= dict.fromkeys(column_names)
```

```
# Remove an irrelvant column
del launch_dict['Date and time ( )']
```

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6.Filling up the `launch_dict` with launch records

(Too long to put in here, so please refer to the notebook)

7.Creating a Dataframe and exporting it to a CSV

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

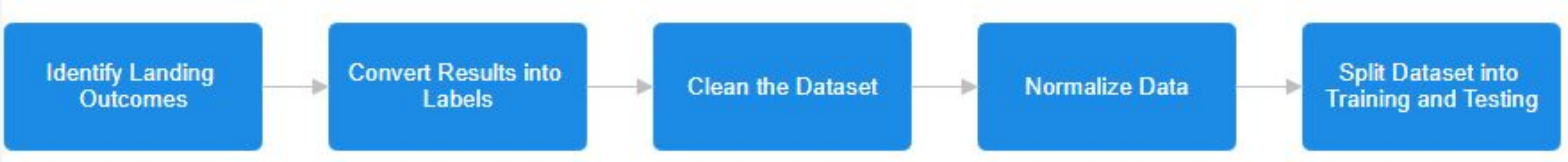
Categorize each landing attempt based on the results:

- ○ **True Ocean**: Successfully landed in the ocean.
- ○ **False Ocean**: Failed to land in the ocean.
- ○ **True RTLS**: Successfully landed on the ground pad.
- ○ **False RTLS**: Failed to land on the ground pad.
- ○ **True ASDS**: Successfully landed on the drone ship.
- ○ **False ASDS**: Failed to land on the drone ship.

Convert Landing Results into Training Labels

- Assign binary labels for training:
  - `1` = Successful Landing
  - `0` = Failed Landing

Clean the Dataset
Normalize Data
Split Dataset



EDA with Data Visualization

Summary of Plotted Charts

1. Scatter Charts

- **Flight Number vs. Launch Site**
- **Payload vs. Launch Site**
- **Flight Number vs. Orbit Type**
- **Payload vs. Orbit Type**

Purpose:
Scatter charts were used to visualize the correlation between two numerical variables. They provide insights into how one variable may affect another and reveal patterns or trends within large datasets. These charts help in understanding relationships, such as how different launch sites or orbit types influence flight numbers and payloads.

2. Bar Chart

- **Orbit Type vs. Success Rate**

Purpose:
A bar chart was employed to compare success rates across different orbit types. It allows for easy visualization and comparison of discrete categories at a glance. By representing one axis with categories (orbit types) and the other with success rates, this chart effectively illustrates the relationship between the type of orbit and the associated success rate of launches.

3. Line Chart

- **Year vs. Success Rate**

Purpose:
A line chart was utilized to show the trend of success rates over the years. This chart effectively conveys data variables and trends, making it easier to observe changes over time. It helps in predicting potential future outcomes based on historical data, allowing for a clearer understanding of how launch success has evolved.

EDA with SQL

Task 1: Display unique launch sites

- Query: `SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;`

Task 2: Display 5 records where launch sites begin with 'CCA'

- Query: `SELECT * FROM "SPACEXTABLE" WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;`

Task 3: Display total payload mass carried by NASA (CRS)

- Query: `SELECT SUM("Payload_Mass_kg") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Launch_Provider" = 'NASA (CRS)';`

Task 4: Display average payload mass for booster version F9 v1.1

- Query: `SELECT AVG("Payload_Mass_kg") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';`

Task 5: List the date of the first successful landing outcome on ground pad

- Query: `SELECT MIN("Date") AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';`

Task 6: List booster names with successful drone ship landings and payload mass > 4000 but < 6000

- Query: `SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass_kg" > 4000 AND "Payload_Mass_kg" < 6000;`

Task 7: Count of successful and failure mission outcomes

- Query: `SELECT "Landing_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY "Landing_Outcome";`

Task 8: List booster versions with maximum payload mass using subquery

- Query: `SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass_kg" = (SELECT MAX("Payload_Mass_kg") FROM SPACEXTABLE);`

Task 9: Display month names, failure landing outcomes in drone ship, booster versions, and launch site for 2015
Task 10: Rank landing outcomes count between 2010-06-04 and 2017-03-20

- Query: `SELECT "Landing_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Total_Count DESC;`

Build an Interactive Map with Folium

Map Objects Created with Folium

- **Markers for Launch Sites:**
    - Indicate all launch sites on the map for easy identification.
- **Markers for Launch Outcomes:**
    - Differentiate between successful and failed launches at each site, allowing quick assessment of performance.
- **Lines Showing Distances:**
    - Visualize distances from launch sites to nearby infrastructure (railways, highways, coastlines) for spatial analysis.

Rationale for Adding Objects

- **Geographical Pattern Analysis:**
    - **Proximity to Railways and Highways:** Assess logistical advantages for transportation.
    - **Proximity to Coastline:** Consider safety and operational factors.
    - **Distance from Cities:** Ensure compliance with safety regulations and minimize risks.

These objects facilitate a comprehensive understanding of launch site operations and their contextual relationships with key geographical features.

Build a Dashboard with Plotly Dash

Plots and Interactions in the Dashboard

- **Pie Chart:**
  - **Purpose:** Displays the total successful launches by launch sites.
  - **Interactivity:** Users can select to view the overall successful landing distribution or focus on the success rate of individual launch sites.
- **Scatter Chart:**
  - **Purpose:** Illustrates the relationship between launch outcomes and payload mass (in kg) for different boosters.
  - **Inputs:**
    - Selection for all sites or an individual site.
    - A slider to adjust the payload mass between 0 and 10,000 kg.
  - **Benefits:** Helps analyze how success rates correlate with launch locations, payload mass, and booster categories.

Rationale for Adding Plots and Interactions

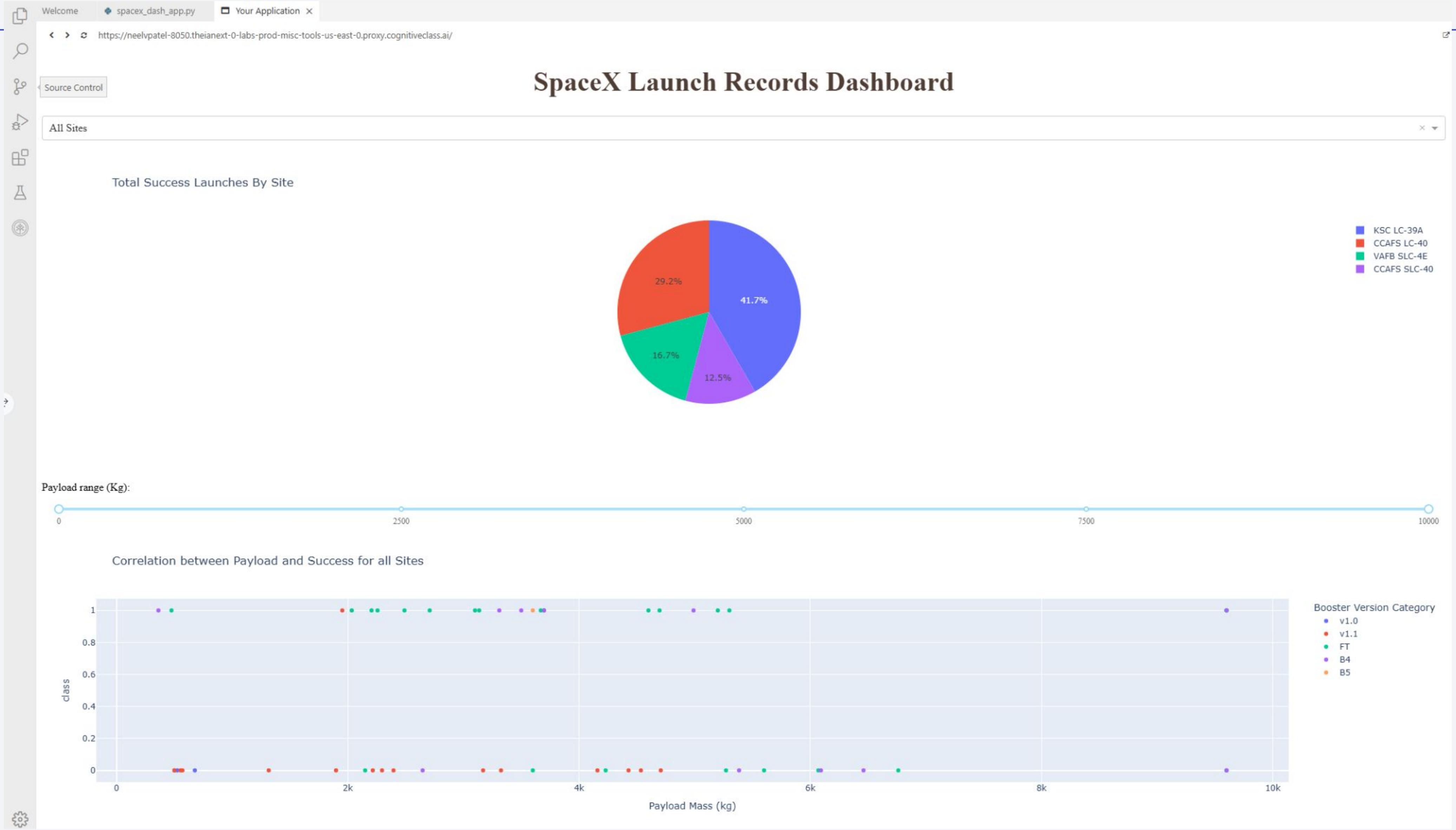
- **Insightful Analysis:** The pie chart provides a clear overview of success rates across sites, while the scatter chart enables detailed examination of factors affecting launch outcomes.
- **User Engagement:** Interactive elements allow users to explore data dynamically, facilitating deeper insights into launch performance and operational trends.

Predictive Analysis (Classification)

Model Development Process

1. **Exploratory Data Analysis (EDA)**
    - Analyze data distributions and correlations.
    - Create class labels (1 for success, 0 for failure).
2. **Data Preparation**
    - **Standardize Data:** Normalize features.
    - **Split Data:** Divide into training and test sets.
3. **Model Selection and Hyperparameter Tuning**
    - Test models: SVM, Classification Trees, Logistic Regression.
    - Optimize hyperparameters using Grid or Random Search.
4. **Model Evaluation**
    - Evaluate performance on test data.
    - Compare accuracy, precision, recall, and F1-score.
5. **Model Improvement**
    - Iterate improvements based on performance metrics.

Results



Welcome
spacex_dash_app.py
Your Application
https://neelvpatel-8050.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/
Source Control
SpaceX Launch Records Dashboard
All Sites
Total Success Launches By Site
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
0
2500
5000
7500
10000
Correlation between Payload and Success for all Sites
class
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

Section 2

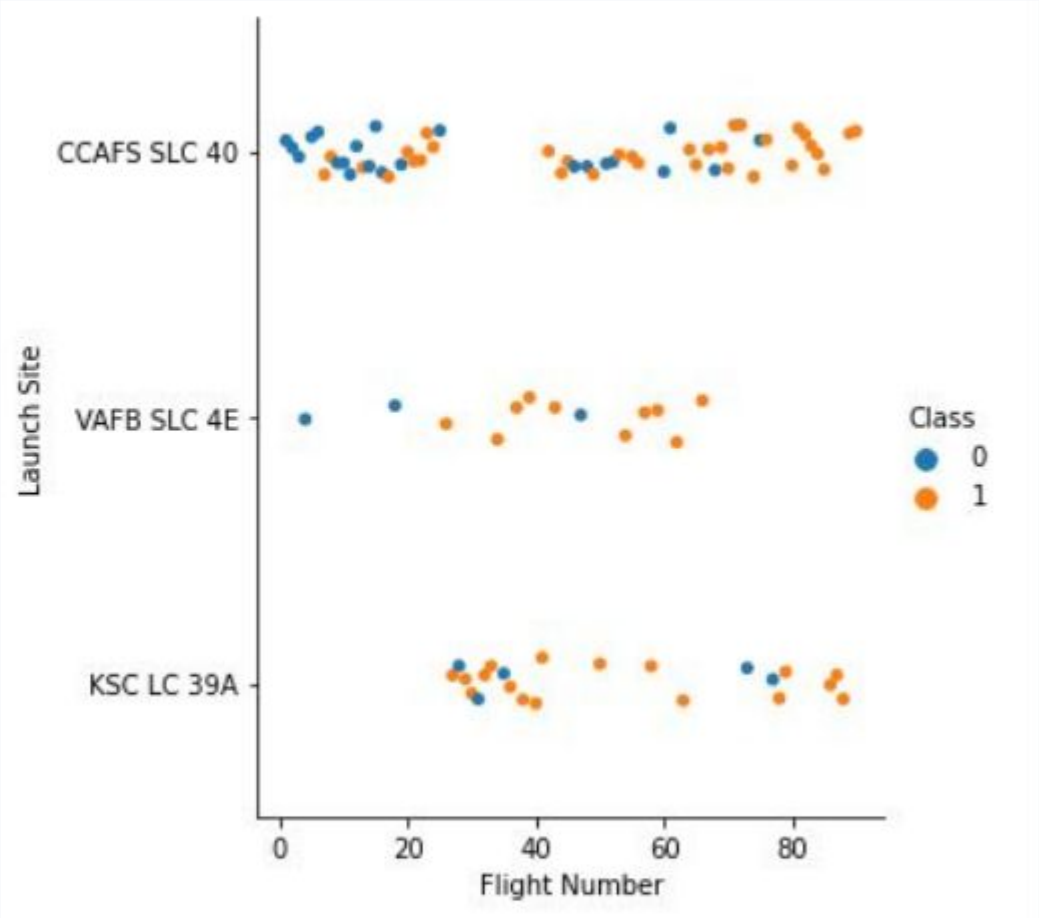
Insights drawn from EDA

Flight Number vs. Launch Site

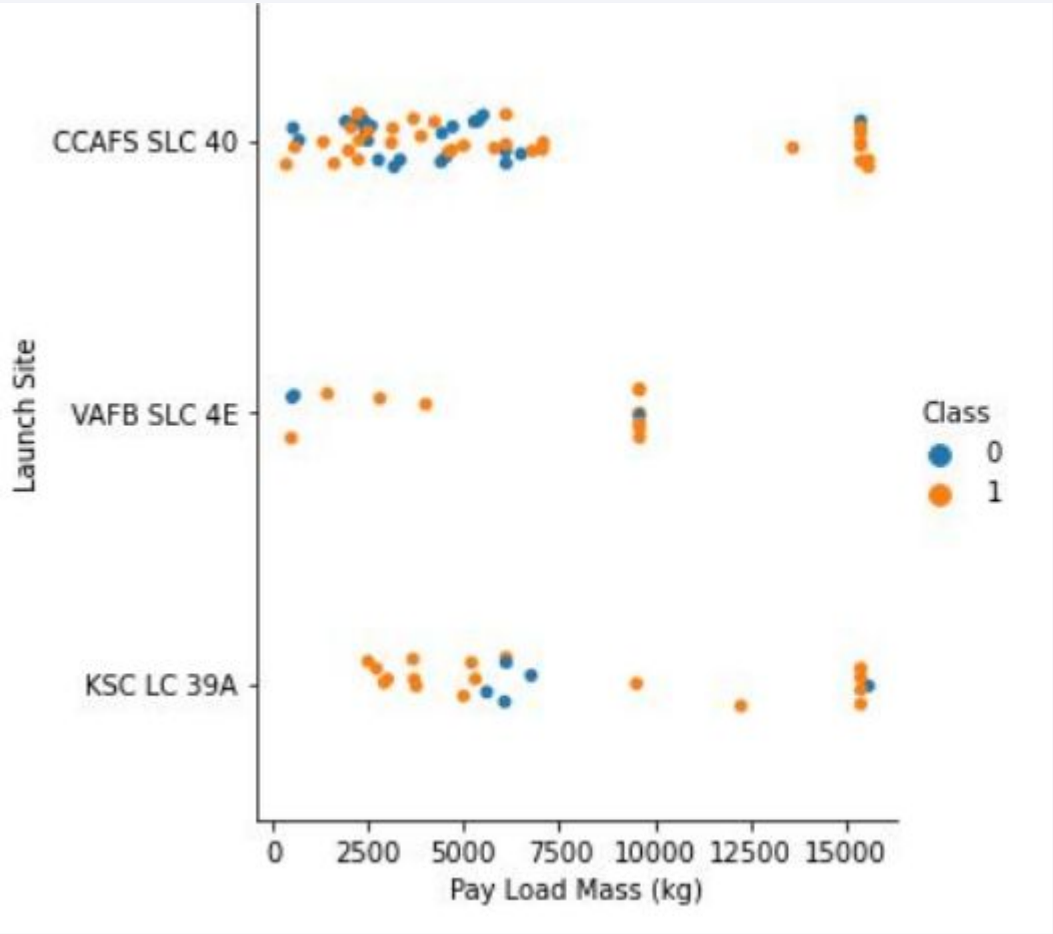
This scatter plot shows Flight Number versus Launch Site.

- **Class 0 (blue)**: Represents unsuccessful launches.
- **Class 1 (orange)**: Represents successful launches.

The plot illustrates that as flight numbers increase, the success rate also improves, especially after the 20th flight, marking a significant improvement in launch success.



Payload vs. Launch Site



Here's a scatter plot of Payload versus Launch Site:

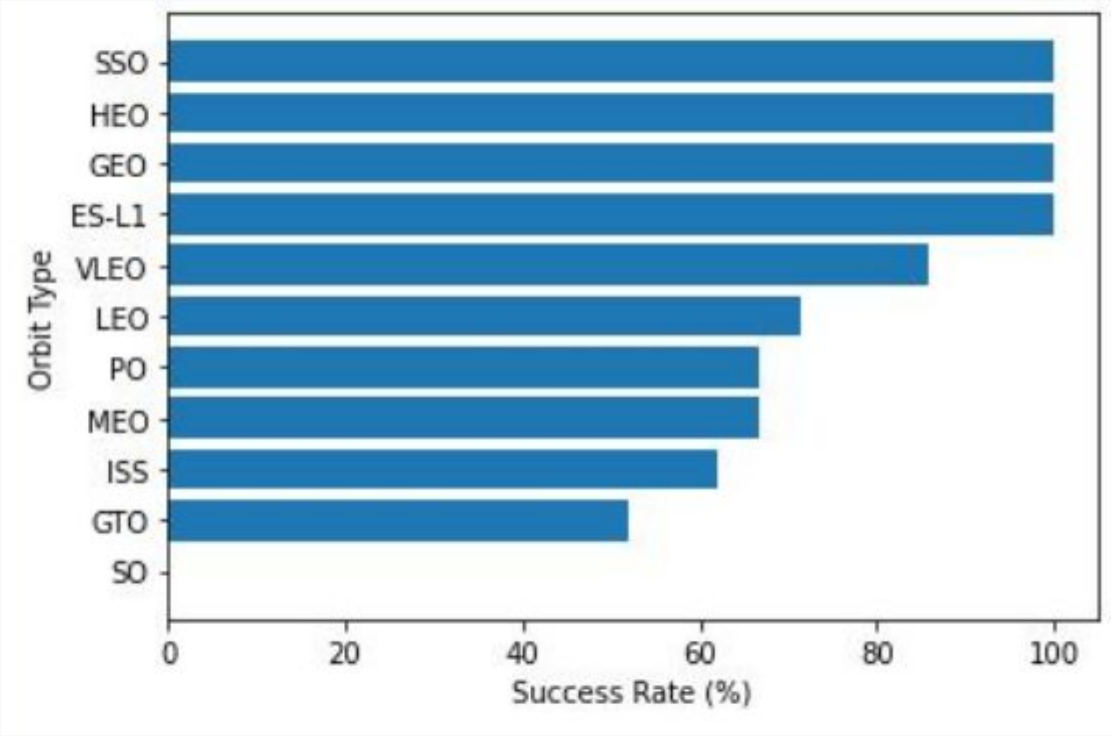
- **Class 0 (blue)**: Represents unsuccessful launches.
- **Class 1 (orange)**: Represents successful launches.

This plot suggests that while larger payload masses might have a higher success rate, it's challenging to draw definitive conclusions, as there isn't a clear pattern linking payload mass directly to successful launches.

Success Rate vs. Orbit Type

Orbit types SSO, HEO, GEO, and ES-L1 show the highest success rates at 100%.

- In contrast, the GTO orbit type has a success rate of only 50%, making it the lowest, aside from the SO type, which failed in its only attempt.



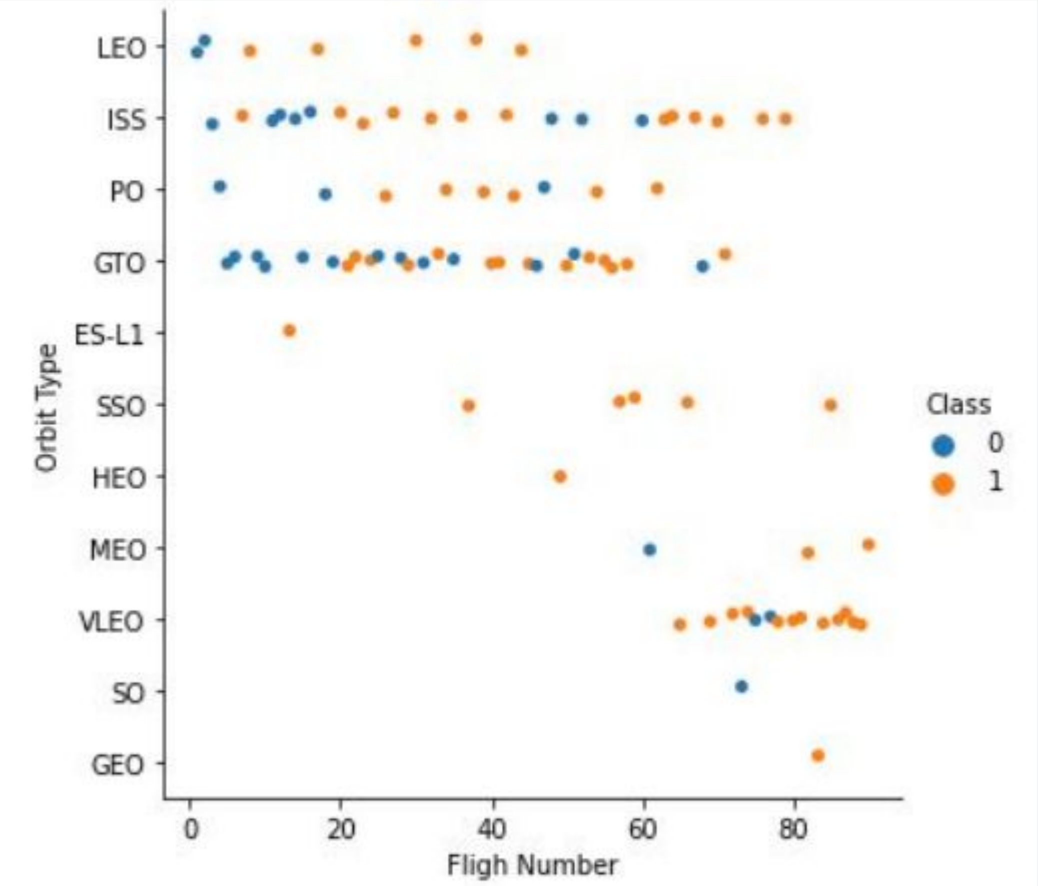
Flight Number vs. Orbit Type

Class 0 (blue) indicates unsuccessful launches, while **Class 1** (orange) represents successful launches.

The scatter plot suggests that, in most cases, **launch success correlates with the flight number**—as the flight number increases, the success rate tends to improve.

However, for **GTO orbit**, there appears to be no clear relationship between flight numbers and the success rate.

SpaceX initially focused on **LEO** with moderate success but has increasingly relied on **VLEO** for recent launches, where the success rate is notably high.

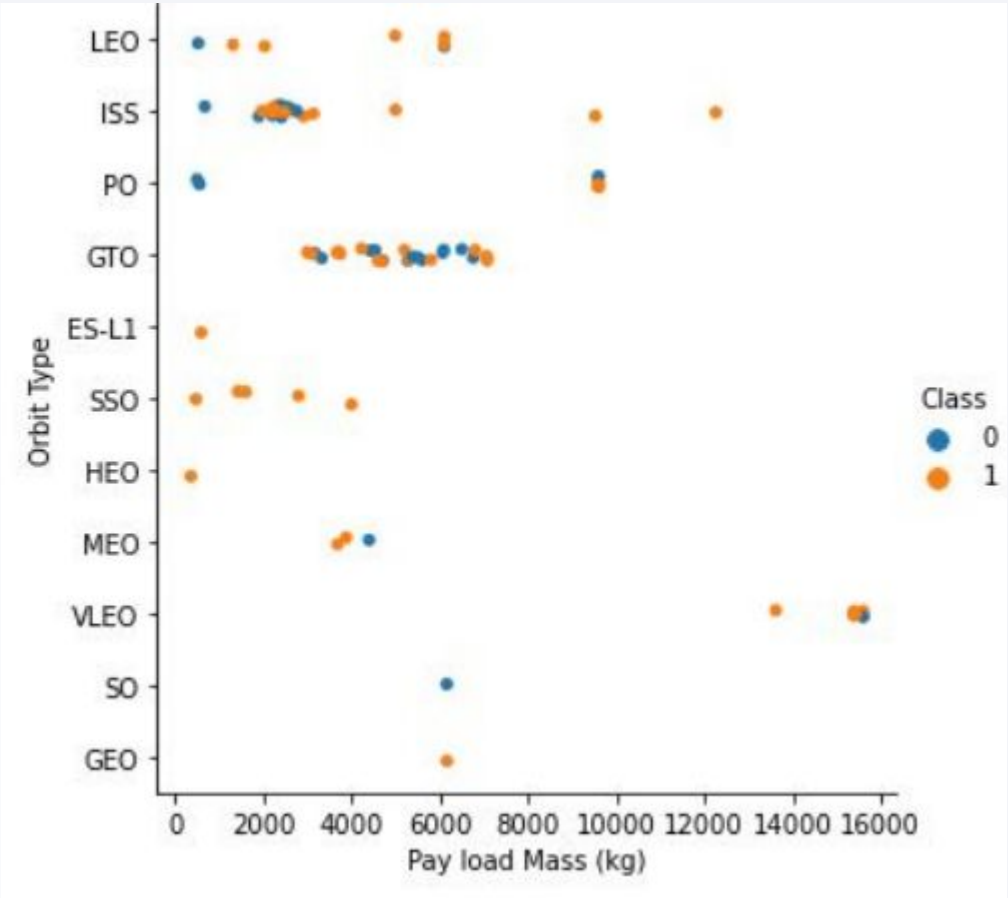


Payload vs. Orbit Type

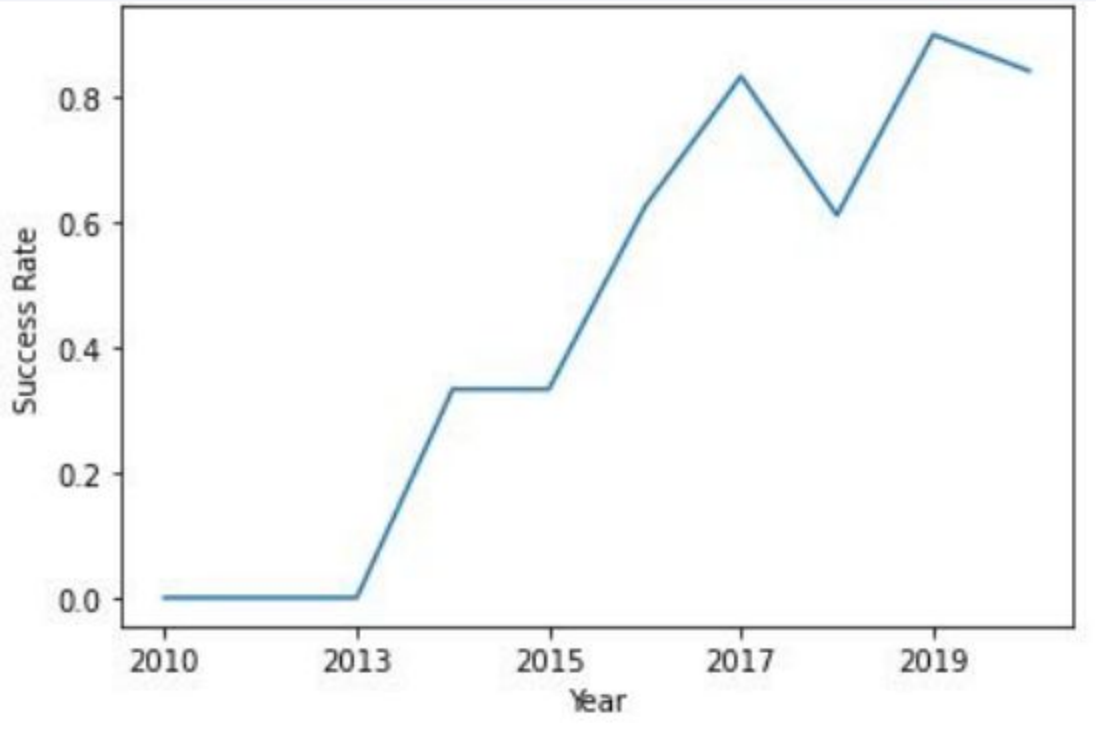
Class 0 (blue) shows unsuccessful launches, while **Class 1** (orange) indicates successful launches.

The scatter plot reveals that **heavy payloads** are generally associated with higher success rates, particularly in **LEO** and **ISS** orbits.

In contrast, for **GTO**, it is difficult to distinguish between successful and unsuccessful landings as the data points are clustered closely together, showing no clear pattern.



Launch Success Yearly Trend



- Since **2013**, the **success rate** steadily increased up until **2017**.
- In **2018**, there was a slight decline in the rate.
- Currently, the success rate is approximately **80%**.

All Launch Site Names

The SQL `DISTINCT` clause was used to retrieve only unique values from the `Launch_Site` column in the SpaceX table.

- **There are four unique launch sites**:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

The Query:

```
SELECT DISTINCT LAUNCH_SITE

FROM SPACEXTBL;
```

Launch Site Names Begin with 'CCA'

The SQL query retrieves records from the SpaceX table, displaying only five records due to the inclusion of the `LIMIT 5` clause. This limit ensures that the result set does not exceed five entries.

To filter for launch site names that start with `CCA`, the query employs the `LIKE` operator in conjunction with the percent sign (`%`). This allows for matching any characters that may follow the `CCA` prefix in the `Launch_Site` field.

The Query:

```
SELECT * FROM SPACEXTBL

WHERE LAUNCH_SITE LIKE ‘CCA%’

LIMIT 5
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

The SQL query utilizes the `SUM()` function to compute the total mass of payloads (measured in kilograms) carried by boosters for NASA. The `WHERE` clause filters the dataset to ensure that only records where the customer is 'NASA (CRS)' are included in the calculation.

The resulting total payload mass is **45,596 kg**.

The Query:

```
SELECT SUM(PAYLOAD_MASS_KG_) AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = ‘NASA (CRS)’
```

Average Payload Mass by F9 v1.1

The SQL query employs the `AVG()` function to determine the average payload mass (in kilograms) for launches using the F9 v1.1 booster version. The `WHERE` clause filters the dataset to include only those records where the booster version is specified as 'F9 v1.1'.

The resulting average payload mass is **2,928 kg**.

The Query:

```
SELECT AVG(PAYLOAD_MASS_KG_) AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = ‘F9 v1.1’
```

First Successful Ground Landing Date

The SQL query utilizes the `MIN()` function to identify the earliest date in the `DATE` column. The `WHERE` clause filters the dataset to include only those records where the landing outcome is classified as 'Success (ground pad)'.

The resulting date of the first successful landing on a ground pad is **December 22, 2015**.

The Query:

```
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING_OUTCOME = ‘Success (ground pad)’
```

Successful Drone Ship Landing with Payload between 4000 and 6000

The SQL query retrieves the names of boosters from the dataset, applying filters in the `WHERE` clause to ensure that only records with a landing outcome of 'Success (drone ship)' are included. Additionally, it uses the `AND` operator to further filter the results, displaying only those records where the payload mass is between 4,000 kg and 6,000 kg.

The Query:

SELECT BOOSTER_VERSION

FROM SPACEXTBL

WHERE LANDING_OUTCOME = ‘Success (drone ship)’ and (PAYLOAD_MASS_KG_ BETWEEN 4000 AND 6000)

- Result

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

Total Number of Successful and Failure Mission Outcomes

The SQL query uses the `COUNT()` function to tally the total number of missions in the dataset. By employing the `GROUP BY` statement, the query groups the results based on the `Mission_outcome` field, allowing for a summary of the total count for each outcome category (success and failure).

According to the results, SpaceX has successfully completed nearly **99%** of its missions.

The Query:

```
SELECT MISSION_OUTCOME,
    COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

- Result

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

The SQL query utilizes a subquery to first determine the maximum payload mass using the `MAX()` function. The outer query then filters the dataset to find the names of the boosters that have achieved this maximum payload mass.

According to the results, the booster version **F9 B5 B10xx.x** carried the maximum payload.

The Query:

```
SELECT DISTINCT BOOSTER_VERSION,
PAYLOAD_MASS_KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS_KG = (
    SELECT MAX(PAYLOAD_MASS_KG_)
    FROM SPACEXTBL)
```

- Result

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

The SQL query filters the dataset to retrieve records where the `Landing_outcome` is 'Failure (drone ship)'. The `WHERE` clause also includes an additional condition to specify that the records must be from the year 2015. The `AND` operator is used to combine these conditions.

According to the results, there were **two landing failures on drone ships** in 2015.

The Query:

```
SELECT LANDING_OUTCOME,
        BOOSTER_VERSION,
        LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING_OUTCOME = ‘Failure (drone ship)’ AND YEAR(DATE) = ‘2015’
```

- Result

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

The SQL query filters the dataset using the `WHERE` clause to include only records where the date falls between June 4, 2010, and March 20, 2017. The query then utilizes the `COUNT()` function to tally the occurrences of each landing outcome. The `GROUP BY` clause groups the results by the `Landing_outcome`, while the `ORDER BY` clause sorts the counts in descending order using the `DESC` keyword.

According to the results, the number of successes and failures between June 4, 2010, and March 20, 2017, was similar.

The Query:

```
SELECT LANDING_OUTCOME,
    COUNT(LANDING_OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY total_number DESC
```

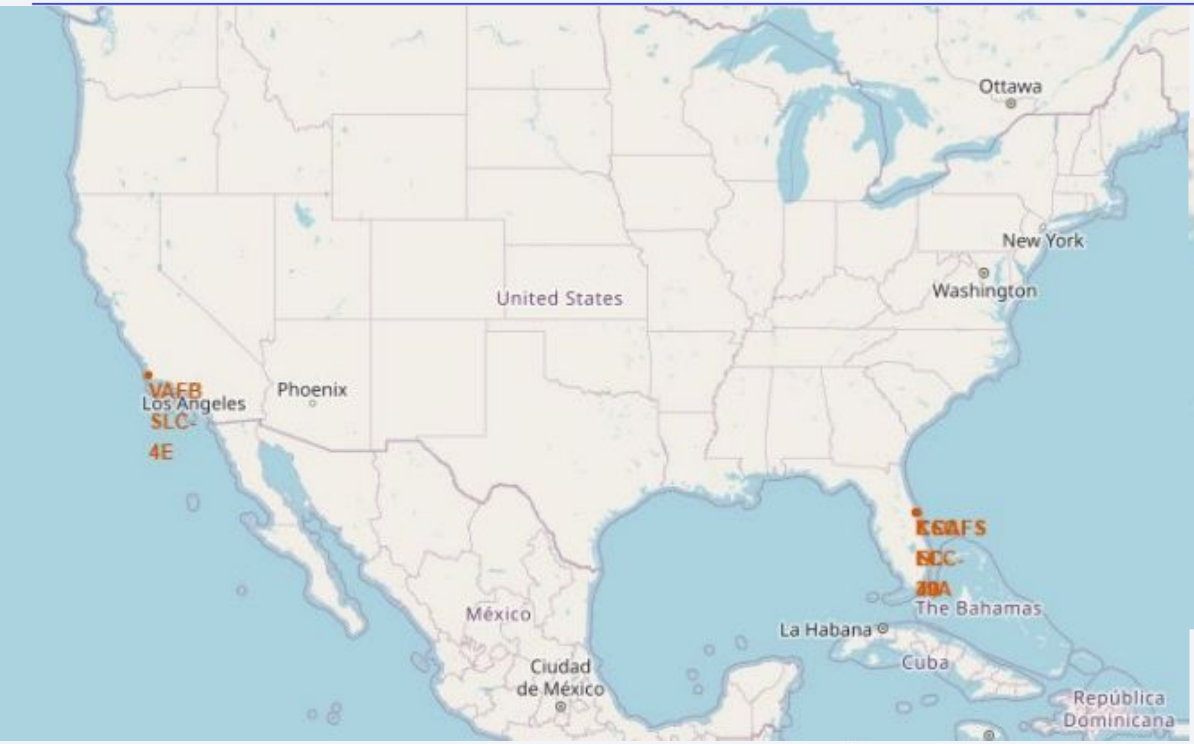
- Result

landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

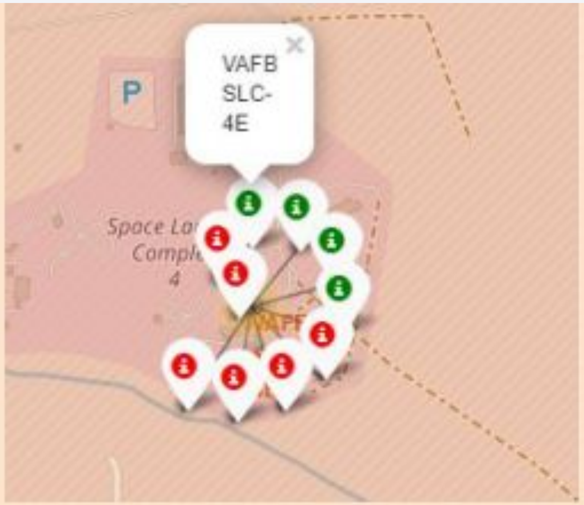
Launch Sites Proximities Analysis

Launch Site Locations



- **Launch Sites Overview**: The left map displays all SpaceX launch sites, while the right map confirms that all these sites are located within the United States. This highlights SpaceX's operational focus on domestic launches.
- **Geographic Distribution**: Both maps illustrate that all launch sites are situated near the coast. This coastal positioning is significant for several reasons:
  - **Safety**: Launching from coastal areas reduces the risk to populated regions in the event of a launch failure, as rockets can ascend over open water.
  - **Range**: Coastal sites allow for a broader range of launch trajectories, enabling missions to various orbits and destinations without overflying land.

<Folium Map Screenshot 2>



Launch site in California

Successful Landing is in Green
Failed Landing is in Red

Launch sites in Florida

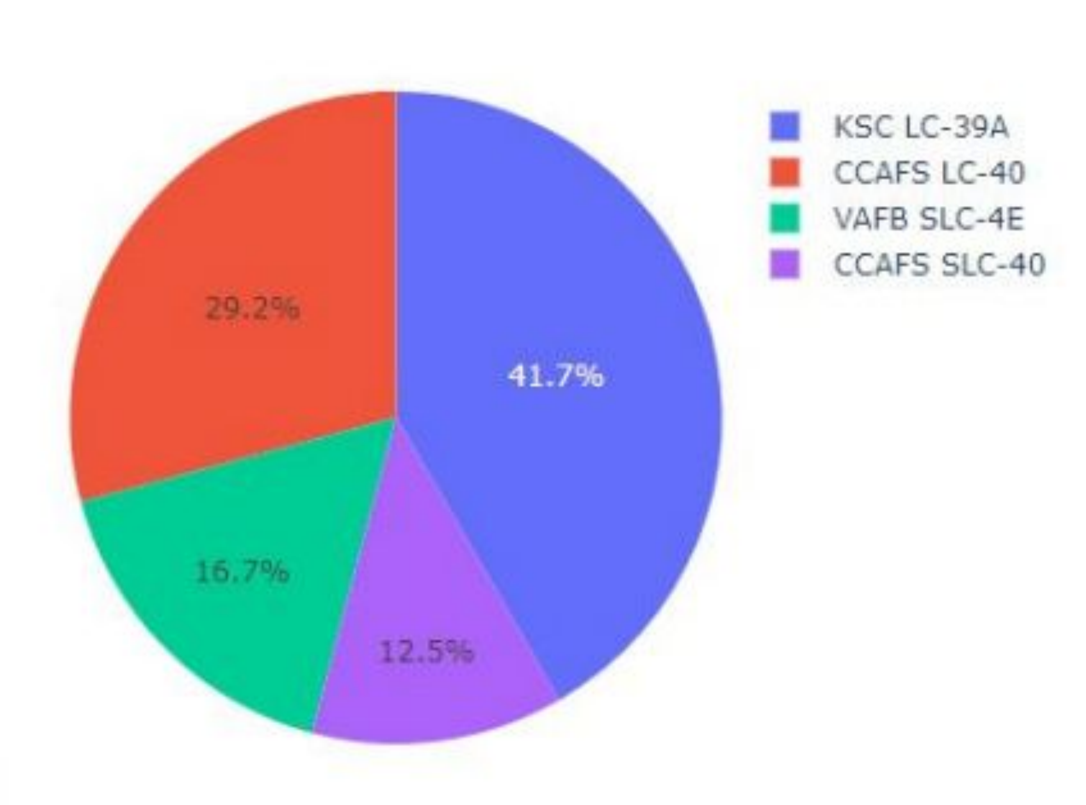
Section 4

Build a Dashboard with Plotly Dash

Total Success Launches By all sites

Explanation of Important Elements and Findings:

- **KSLC-39A Success**: The screenshot shows that **KSLC-39A** has the highest number of successful launches among all SpaceX sites, indicating its reliability.
- **VAFB SLC-4E Performance**: **VAFB SLC-4E** records the fewest successful launches, which may be due to:
  - **Small Data Sample**: Limited launches can skew results.
  - **Geographic Challenges**: Being the only California site, VAFB may face unique difficulties, such as weather and airspace restrictions, that affect launch success compared to east coast sites.
- **Future Implications**: Analyzing launch site performance can guide future operational strategies to enhance success rates.



Launch Site with Best Launch Success Ratio

Explanation of Important Elements and Findings:

- **KSLC-39A Performance**: The screenshot indicates that **KSLC-39A** has the highest landing success rate among SpaceX launch sites, achieving **10 successful landings** (76.9%) and **3 failures** (23.1%).
- **Success Rate Implications**: This high success rate highlights KSLC-39A's reliability as a launch site, suggesting effective operational practices and favorable conditions contributing to successful landings.

Payload and Launch Outcome Scatter Plot

Launch Success Rates by Payload Weight: The figures indicate that the **launch success rate (class 1)** is significantly higher for **low-weight payloads (0-5000 kg)** compared to **heavy-weight payloads (5000-10000 kg)**.

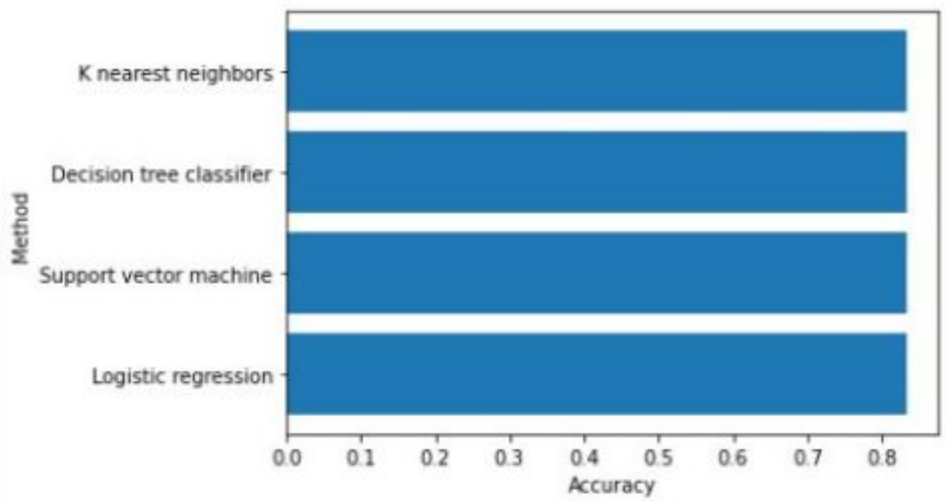
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Findings: All models achieved an accuracy of **83.33%** on the test set. However, the small test size of **18** samples may not provide a reliable measure of model performance.

Conclusion: To accurately determine the optimal model, a larger dataset is needed.

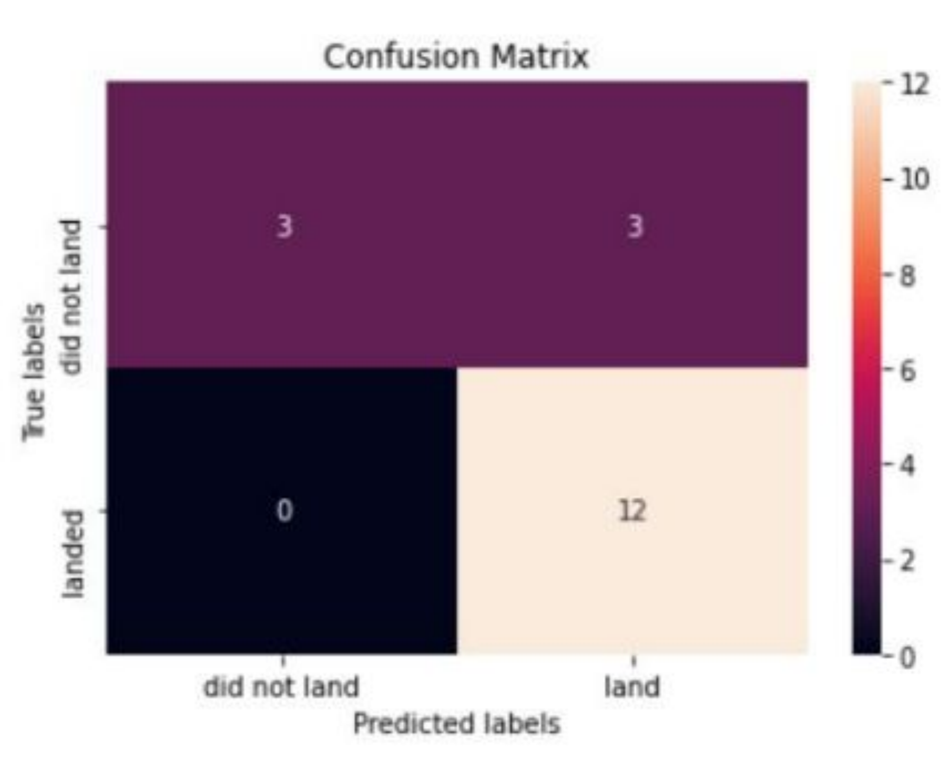


	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

Confusion Matrix

The confusion matrix for the best-performing model, which is identical across all models due to their equal performance, reveals the following outcomes:

- **True Positives (TP)**: **12** successful landings were correctly predicted.
- **True Negatives (TN)**: **0** failed landings were correctly predicted (indicating no failures were identified correctly).
- **False Positives (FP)**: **3** instances were incorrectly predicted as successful landings when they were actually failures.
- **False Negatives (FN)**: **3** failed landings were incorrectly predicted as successful landings.



Summary:

- The models effectively predict **successful landings**, as evidenced by the high number of true positives.
- However, the presence of **false positives** indicates that the models struggle with identifying failed landings.

Overall, while the models demonstrate strong performance in predicting successful outcomes, there is a need for improvement in accurately detecting failures.

Conclusions

- **Model Accuracy and Data Needs**: Despite all models achieving an accuracy of **83.33%**, the small dataset size (18 samples) limits the reliability of these results. More comprehensive data is needed to accurately assess and determine the optimal model.
- **Challenges with Failure Predictions**: The presence of **false positives** (3 predictions of successful landings that were actually failures) indicates a limitation in the models' ability to accurately detect failed landings. This suggests that further refinement may be needed to enhance the models' performance in this area.
- **Increasing Success Rate**: As the number of flights has increased, the overall success rate has improved, now exceeding **80%.** This trend reflects enhancements in launch technology and operational procedures.
- **KSLC-39A Performance**: **KSLC-39A** stands out as the site with the highest number of successful launches and the best success rate, demonstrating its effectiveness as a launch facility.
- **Payload Weight Impact**: The success rate for **low-weight payloads (0-5000 kg)** is higher than that for **heavy-weight payloads (5000-10000 kg)**, suggesting that lighter payloads may face fewer challenges during launch.

Thank you!