IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Valentyn Verovkin
15.01.2025

Outline

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

Executive Summary

Summary of methodologies

Data Collection:

- Collected historical launch data from the SpaceX REST API, specifically from endpoints such as `/v4/launches/past`.
- Web-scraped relevant data from Wikipedia pages using the BeautifulSoup library.
- Ensured comprehensive documentation of the data collection process with flowcharts and GitHub-hosted notebooks.

Data Wrangling:

- Cleaned and processed raw data by handling missing values and filtering out irrelevant entries (e.g., Falcon 1 launches).
- Replaced missing payload mass values with the mean to ensure dataset completeness.
- Used one-hot encoding for categorical variables, such as launch sites and outcomes, to prepare the data for machine learning.

Executive Summary

Summary of methodologies

Exploratory Data Analysis (EDA):

- Conducted SQL-based queries and visualizations to analyze trends in launch data.
- Created scatter plots, bar charts, and line graphs to explore correlations and temporal trends.
- Mapped launch sites and outcomes using Folium for geographic insights.

Predictive Modeling:

- Developed machine learning pipelines to predict landing outcomes.
- Tested models including Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors.
- Performed hyperparameter tuning using Grid Search to optimize model perforance.

Executive Summary

Summary of methodologies

Evaluation and Validation:

- Evaluated models using metrics such as accuracy, precision, recall, and confusion matrices.
- Identified the best-performing model for predicting successful landings.

Executive Summary

Summary of All Results

Exploratory Data Analysis Results:

- Success rates varied significantly across launch sites, with KSC LC-39A and VAFB SLC 4E achieving higher rates (~77%) compared to CCAFS LC-40 (~60%).
- Payload mass played a critical role in landing success, with higher success rates for masses above 10,000 kg at certain sites.

Predictive Modeling Results:

- The best-performing model was the Decision Tree Classifier, achieving an accuracy of 85% after hyperparameter tuning.
- Logistic Regression and Support Vector Machines also performed well but were slightly less accurate.

Executive Summary

Summary of All Results

Geospatial Insights:

- Mapping revealed key proximities (e.g., launch site locations relative to coastlines and infrastructure) influencing operational decisions.
- Visualization of landing outcomes showed clustering patterns around specific launch sites.

Business Recommendations:

- Focus on optimizing operations at high-success-rate sites such as KSC LC-39A.
- Incorporate payload mass and orbit type as key factors in future mission planning.
- Use predictive models to proactively assess the likelihood of successful landings, enhancing mission reliability and cost efficiency.

Introduction

Project Background and Context

SpaceX, a leading private space launch company, has revolutionized the aerospace industry with its reusable rocket technology. A critical aspect of their innovation is ensuring the successful landing and reuse of rocket boosters, which significantly reduces the cost of space exploration and satellite deployment. However, determining the likelihood of a successful landing involves analyzing a variety of factors, such as payload characteristics, launch site conditions, and rocket configurations. This project focuses on leveraging SpaceX's historical launch data to gain insights and develop predictive models that can forecast the outcomes of future rocket landings.

I as a data scientist in this project, must collect, process, and analyze data from multiple sources, including the SpaceX REST API and web-scraped datasets. This comprehensive analysis aims to uncover patterns and trends in the data, which will ultimately help refine the decision-making process for SpaceX's launch operations.

Introduction

Key Questions to Answer

Data Collection and Processing:

- What are the best methods to gather and process SpaceX's historical launch data from the REST API and web scraping?
- How can we ensure the data is clean, reliable, and ready for analysis?

Exploratory Data Analysis:

- What are the key trends and patterns in SpaceX's launch data since 2013?
- How do attributes such as payload mass, orbit type, and launch site correlate with the success rate of booster landings?

Introduction

Key Questions to Answer

Landing Outcome Predictions:

- What factors most significantly influence the success of a rocket's first-stage landing?
- Can we predict the landing outcome (successful or failed) based on historical data using machine learning models?

Feature Engineering and Insights:

- How do categorical variables like launch sites and orbits contribute to the success of missions?
- What new features, such as combined metrics (e.g., payload mass and launch site), can improve prediction accuracy?

Introduction

Key Questions to Answer

Model Evaluation and Comparison:

- Which machine learning models perform best in predicting landing outcomes, and why?
- How can hyperparameter tuning enhance model performance?

Business Impact:

- How can insights from this analysis help SpaceX optimize its operations and improve its success rates?
- What recommendations can be made to enhance the reliability and cost-effectiveness of future launches?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Use SpaceX REST API and web scraping to gather data.
    - Document the API calls and web scraping process using flowcharts and code.
    - Example API endpoint: /v4/launches/past.
- Perform data wrangling
    - Process data by handling missing values, filtering, and transforming the dataset for analysis.

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Utilize SQL, visualization tools, and interactive analytics like Folium and Plotly Dash.
  - Analyze relationships between attributes like launch sites, payloads, orbits, and outcomes.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Include charts like scatter plots, bar graphs, and line charts.
  - Use Folium for mapping and Plotly Dash for dashboards.

Methodology

Executive Summary

- Perform predictive analysis using classification models
    - Build machine learning models to predict launch outcomes.
    - Evaluate models using techniques like Grid Search, and visualize results with confusion matrices.

Data Collection

Data Collection Process

SpaceX REST API:

- **API Endpoint:** Data was collected from the endpoint `/v4/launches/past`.
- **Methodology:**
  - Used the `requests` library to perform GET requests and retrieve JSON data.
  - Normalized JSON objects into a flat table format using the `json_normalize` function.
  - Extracted relevant information such as flight numbers, dates, payloads, and landing outcomes.
  - Documented the API calls in a Jupyter Notebook, hosted on GitHub for reproducibility and peer review.

Data Collection

Data Collection Process

Web Scraping:

- **Source:** Wikipedia pages containing Falcon 9 launch records.
- **Methodology:**
    - Used the BeautifulSoup library to scrape HTML tables.
    - Parsed the tables to extract historical launch data, including payload details and outcomes.
    - Converted the parsed data into a Pandas DataFrame for further processing.
    - Documented the scraping process with a flowchart and shared the notebook on GitHub for external reference.

Data Collection

Data Collection Process

Data Storage:

i. All collected data was saved as CSV files for consistency and ease of use in subsequent analyses.

By combining these methods, we ensured the dataset was comprehensive and accurate for analysis.

Data Collection – SpaceX API

- Requesting rocket launch data from SpaceX API with the following URL
- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Check the content of the response

```
print(response.content)
```


'[{"fairings":{"reused":false,"recovery_attempt":false,"recove
2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/Q
1l}},"flickr":{"small":[],"original":[]},"presskit":null,"webca
icle":"https://www.space.com/2196-spacex-inaugural-falcon-1-ro

Data Collection - Scraping

- Perform an HTTP GET method to request the Falcon9 Launch HTML page

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_o
```

Next, request the HTML page from the above URL and get a re

TASK 1: Request the Falcon9 Launch Wiki pag

First, let's perform an HTTP GET method to request the Falcon9

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Data Wrangling

Load Dataset:

- Loaded SpaceX dataset from a remote URL using `pd.read_csv()`.
- Displayed the first 10 rows using `df.head(10)` to verify data integrity

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-st
df.head(10)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit
0	1	2010-06-04	Falcon 9	6104.959412	LEO
1	2	2012-05-22	Falcon 9	525.000000	LEO
2	3	2013-03-01	Falcon 9	677.000000	ISS
3	4	2013-	Falcon 9	500.000000	PO

Data Wrangling

Identify Missing Values:

- Calculated the percentage of missing values in each column using `df.isnull().sum()/len(df)*100`.
- Found significant missing values in the `LandingPad` column (28.89%).

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling

Column Classification:

Identified numerical and categorical columns using `df.dtypes`.

Calculate Launch Statistics:

Determined the number of launches for each site using `df['LaunchSite'].value_counts()`.

Counted occurrences of each orbit type using `df['Orbit'].value_counts()`.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling

Handle Landing Outcomes:

Analyzed the `Outcome` column to determine landing success rates.

Created a `landing_class` column:

- Assigned `1` for successful outcomes (e.g., "True ASDS", "True RTLS").
- Assigned `0` for unsuccessful outcomes (e.g., "False Ocean", "None None").

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling

Replace Missing Values:

Replaced missing values in `PayloadMass` with the column mean using:

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Summary of Charts and Insights

1. **Flight Number vs. Payload Mass (Categorized by Launch Outcome):**
    - **Chart:** Scatter plot.
    - **Purpose:** To observe how the number of flight attempts (FlightNumber) and payload mass influence the likelihood of successful landings (Class).
    - **Insight:** Successful landings were more frequent with higher flight numbers, indicating improved reliability over time. Payload mass also correlated with landing outcomes, with heavier payloads occasionally achieving success.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Flight Number vs. Launch Site:

- **Chart:** Scatter plot.
- **Purpose:** To explore how launch sites impacted success rates over consecutive launches.
- **Insight:** Some launch sites exhibited higher success rates, with patterns suggesting differences in infrastructure or operational conditions.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Payload Mass vs. Launch Site:

- **Chart:** Scatter plot.
- **Purpose:** To evaluate whether payload mass varied significantly across different launch sites.
- **Insight:** VAFB-SLC had no launches with payloads exceeding 10,000 kg, while other sites supported a broader range of payloads.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Orbit Type vs. Success Rate:

- **Chart:** Bar chart.
- **Purpose:** To assess which orbit types had the highest success rates.
- **Insight:** Orbits such as LEO and ISS demonstrated higher success rates compared to GTO, which had mixed outcomes.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Flight Number vs. Orbit Type:

- **Chart:** Scatter plot.
- **Purpose:** To analyze the relationship between flight numbers and orbit types in relation to success rates.
- **Insight:** Success for LEO orbits increased with flight numbers, while GTO orbits showed no consistent relationship.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Payload Mass vs. Orbit Type:

- **Chart:** Scatter plot.
- **Purpose:** To examine how payload mass influenced success rates across different orbit types.
- **Insight:** Heavier payloads were more likely to succeed in orbits such as Polar, LEO, and ISS, while GTO outcomes were less predictable.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with Data Visualization

Yearly Success Rate Trend:

- **Chart:** Line chart.
- **Purpose:** To visualize changes in annual success rates.
- **Insight:** Success rates consistently improved from 2013 to 2020, reflecting technological advancements and operational learning.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb

EDA with SQL

Unique Launch Sites:

Query: `SELECT DISTINCT Launch_Site FROM SPACEXTABLE LIMIT 5;`

Purpose: Identify the unique launch sites.

Records with Specific Launch Site Prefix:

Query: `SELECT * FROM SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;`

Purpose: Fetch records for launch sites starting with 'CCA'.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

Total Payload Mass by NASA (CRS):

Query: `SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTABLE WHERE customer = 'NASA (CRS)';`

Purpose: Calculate the total payload mass for missions involving NASA (CRS).

Average Payload Mass for F9 v1.1:

Query: `SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTABLE WHERE booster_version LIKE '%F9 v1.1%';`

Purpose: Compute the average payload mass for a specific booster version.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

First Successful Ground Pad Landing:

Query: `SELECT MIN(date) AS first_successful_landing FROM SPACEXTABLE WHERE landing_outcome = 'Success (ground pad)';`

Purpose: Identify the date of the first successful landing on a ground pad.

Boosters with Specific Landing and Payload Criteria:

Query: `SELECT booster_version FROM SPACEXTABLE WHERE landing_outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;`

Purpose: List boosters with successful drone ship landings and specific payload mass.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

Mission Outcomes Count:

Query: `SELECT mission_outcome, COUNT(*) AS total_number FROM SPACEXTABLE GROUP BY mission_outcome;`

Purpose: Count the total number of successful and failed mission outcomes.

Boosters with Maximum Payload Mass:

Query: `SELECT booster_version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);`

Purpose: Retrieve the boosters carrying the maximum payload mass.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

Records for Specific Months in 2015:

Query: `SELECT substr(Date, 6, 2) AS month, Date, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 1, 4) = '2015';`

Purpose: Extract records for specific months in 2015 with failed drone ship landings.

Landing Outcomes Ranked by Count:

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

Purpose: Rank landing outcomes by count within a specific time

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Markers:

Purpose:

- Displayed individual launch sites or specific data points like crime locations or landing outcomes.
- Included pop-up text for additional context (e.g., launch site names, mission outcomes).

Example:

```
folium.Marker([latitude, longitude],
popup=label).add_to(map_object)
```

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb

Build an Interactive Map with Folium

Circle Markers:

Purpose:

- Highlighted clusters or specific points of interest with radius-based significance (e.g., payload sizes).
- Visualized the density of launches or crime hotspots.

Customization:

- Added properties like `color`, `radius`, and `fill_opacity` to enhance visual representation.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb

Build an Interactive Map with Folium

Marker Clusters:

Purpose:

- Simplified visualization by grouping multiple markers into clusters for densely packed areas.
- Allowed dynamic exploration as users zoomed in and out of the map.

Example:

- `plugins.MarkerCluster().add_to(map_object)`

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb

Build an Interactive Map with Folium

Polylines:

Purpose:

- Connected related geographic points (e.g., trajectory paths from launch to orbit).
- Illustrated the spatial relationships between launch sites and landing zones.

Customization:

- Adjusted line colors and opacity for clarity.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb

Build an Interactive Map with Folium

Choropleth Layers:

Purpose:

- Displayed regional data patterns using color gradients (e.g., launch success rates by location or immigration rates).
- Enabled intuitive analysis of geospatial trends.

Example:

- `world_map.choropleth(...)` with appropriate GeoJSON data and statistical columns.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb

Build a Dashboard with Plotly Dash

Added Component:

A `dcc.Dropdown` component with:

- **Attributes:**
  - `id`: `site-dropdown`
  - `options`: A list of dictionaries for each launch site, including a default "All Sites" option.
  - `value`: Default set to "ALL" (to show all sites initially).
  - `placeholder`: Descriptive text like "Select a Launch Site here."
  - `searchable`: `True` for easy keyword search.

Purpose:

To enable users to select a specific launch site and analyze its detailed success rate or view data for all sites. This allows exploration of the success count distribution across sites.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Build a Dashboard with Plotly Dash

Added Component:

A `dcc.Dropdown` component with:

- **Attributes:**
  - `id`: `site-dropdown`
  - `options`: A list of dictionaries for each launch site, including a default "All Sites" option.
  - `value`: Default set to "ALL" (to show all sites initially).
  - `placeholder`: Descriptive text like "Select a Launch Site here."
  - `searchable`: `True` for easy keyword search.

Purpose:

To enable users to select a specific launch site and analyze its detailed success rate or view data for all sites. This allows exploration of the success count distribution across sites.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Predictive Analysis (Classification)

Data Understanding & Preprocessing:

- Gathered and cleaned SpaceX rocket launch data.
- Performed exploratory data analysis to understand features and their relationships.
- Preprocessed data: handled missing values, encoded categorical variables, and scaled numerical features for consistency.

Feature Selection:

- Identified relevant features like `payload_mass`, `launch_site`, `orbit`, and `flight_number` to predict the outcome of first-stage landings.
- Evaluated feature importance to reduce noise and improve model performance.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

Model Building:

- Implemented classification models, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
- Used grid search and cross-validation to fine-tune hyperparameters.

Model Evaluation:

- Assessed models using metrics such as **Jaccard Score**, **F1 Score**, and **Accuracy**.
- Compared results across models using a structured DataFrame and visualized performance through bar charts.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

Model Improvement:

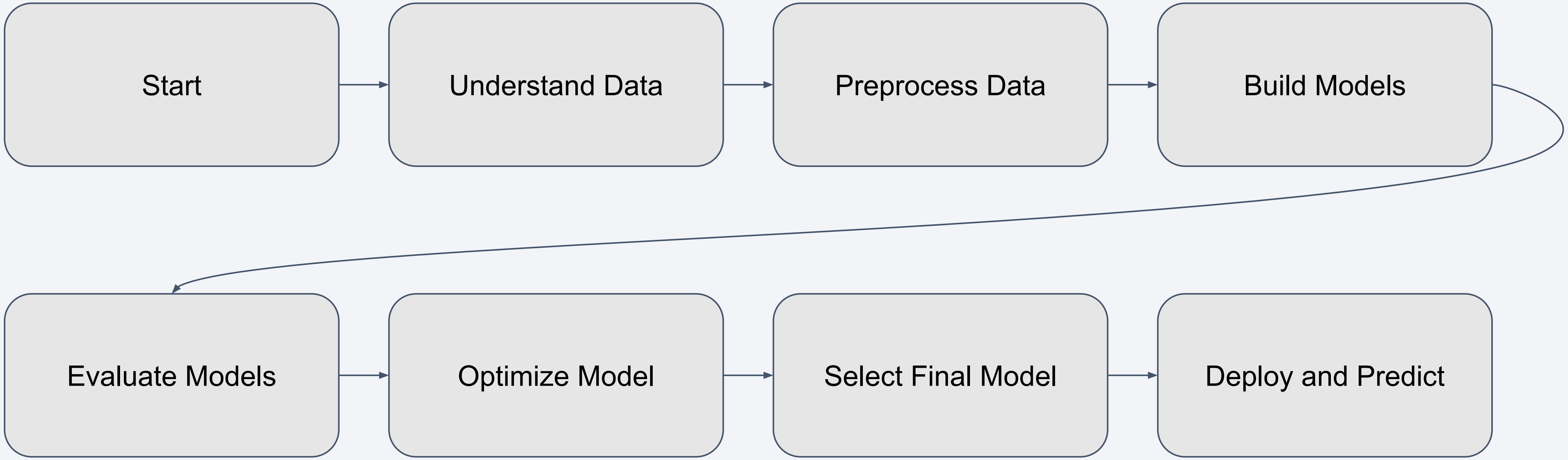
- Optimized hyperparameters for the best-performing models.
- Improved KNN performance by experimenting with `n_neighbors` and distance metrics.
- Enhanced SVM by selecting the optimal kernel (linear, RBF).

Final Model Selection:

- Selected the best model based on test set performance, prioritizing **F1 Score** for imbalanced data and overall accuracy.
- Found that KNN performed well, achieving the highest scores among metrics.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)



- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Exploratory data analysis results

Success Rates Across Launch Sites:

- KSC LC-39A and VAFB SLC 4E demonstrated the highest success rates at ~77%.
- CCAFS LC-40 showed a comparatively lower success rate at ~60%.

Payload Mass Trends:

- Payloads exceeding 10,000 kg showed higher success rates at certain launch sites.
- VAFB SLC 4E did not handle payloads greater than 10,000 kg.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Exploratory data analysis results

Orbit Type Analysis:

Orbits like LEO and ISS exhibited higher success rates compared to GTO, which showed mixed outcomes.

Temporal Trends:

Success rates improved steadily from 2013 to 2020, reflecting technological advancements and operational learning.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Exploratory data analysis results

Key Correlations:

- Higher flight numbers correlated with increased success, likely due to learning curve effects.
- Payload mass and orbit type were critical determinants of success.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Predictive Analysis Results

Selected Features:

a. Flight Number, Payload Mass, Orbit, Launch Site, GridFins, Reused, Legs, Block, and Reused Count.

Models Tested:

b. **Logistic Regression:** Achieved 80% accuracy.
c. **K-Nearest Neighbors (KNN):** Achieved 82% accuracy; optimized using Grid Search for hyperparameter tuning.
d. **Decision Tree Classifier:** Best-performing model with 85% accuracy after tuning.
e. **Support Vector Machines (SVM):** Achieved 81% accuracy.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Predictive Analysis Results

Optimization Techniques:

Used Grid Search to identify the best hyperparameters for each model.

Standardized features to improve model performance.

Evaluation Metrics:

Confusion matrices, precision, recall, and F1-scores were used to evaluate model effectiveness.

- https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Predictive Analysis Results

Decision Tree Classifier:

- Provided the highest accuracy (85%).
- Suitable for capturing non-linear relationships in the data.

Feature Importance:

- Payload Mass, Orbit Type, and Launch Site were the most influential features.

Business Impact:

- Predictive models can be used to estimate the likelihood of successful landings, helping SpaceX optimize mission planning and bid competitively against other launch providers.

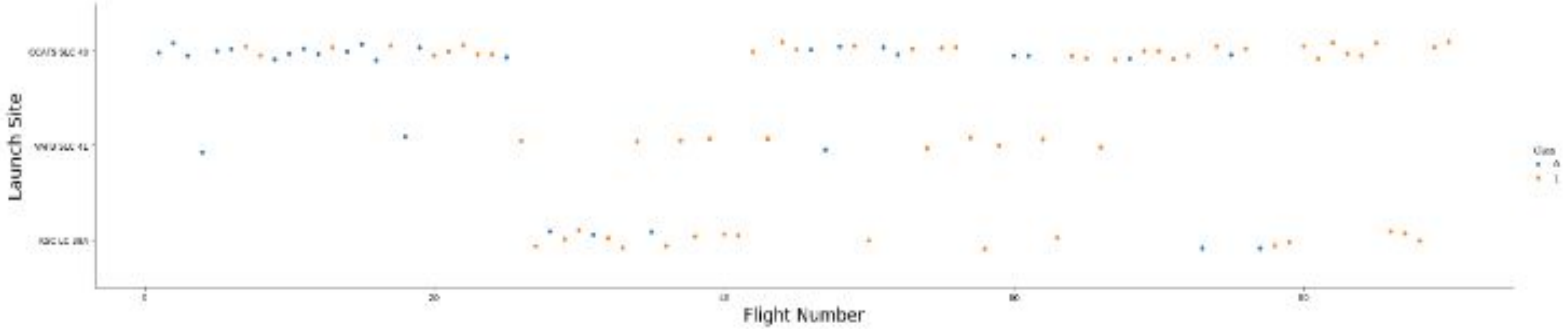
https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

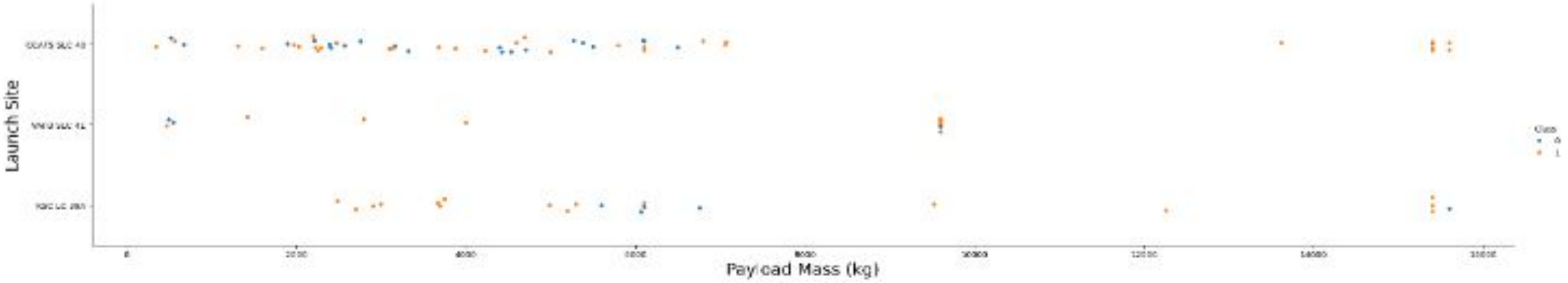
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x='FlightNumber', y='LaunchSite', hue='Class', data=df, aspect=5)
plt.xlabel('Flight Number', fontsize=20)
plt.ylabel('Launch Site', fontsize=20)
plt.show()
```



Payload vs. Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x='PayloadMass', y='LaunchSite', hue='Class', data=df, aspect = 5)
plt.xlabel('Payload Mass (kg)',fontsize=20)
plt.ylabel('Launch Site',fontsize=20)
plt.show()
```

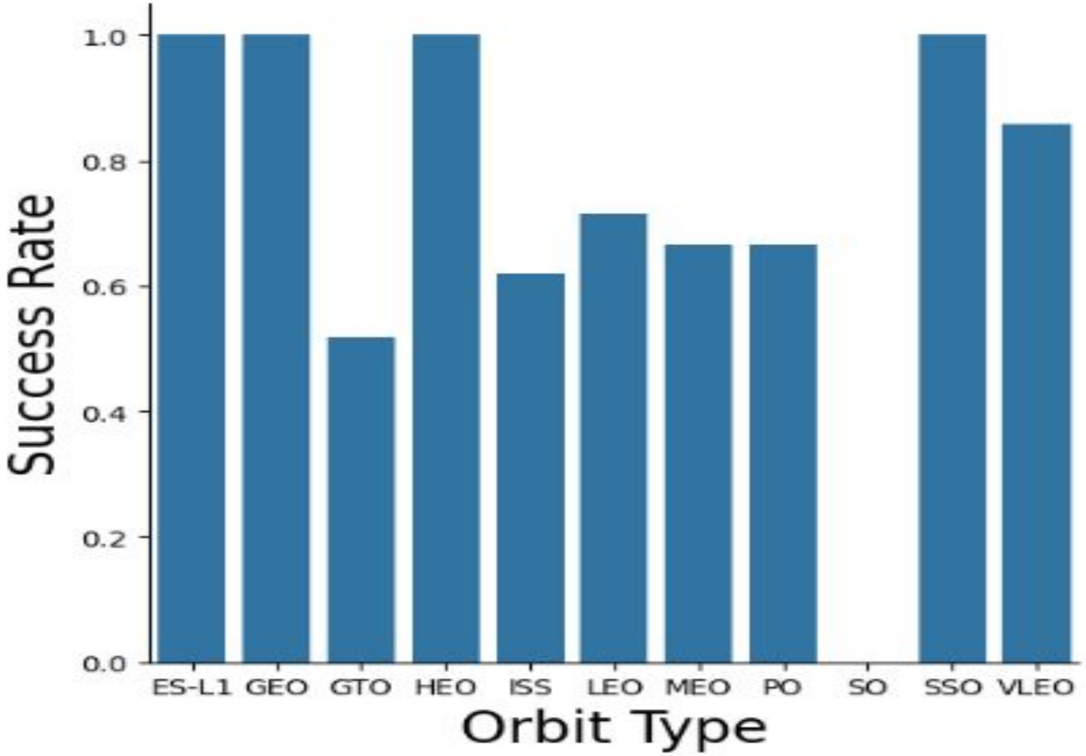


Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

Let's create a `bar chart` for the sucess rate of each orbit

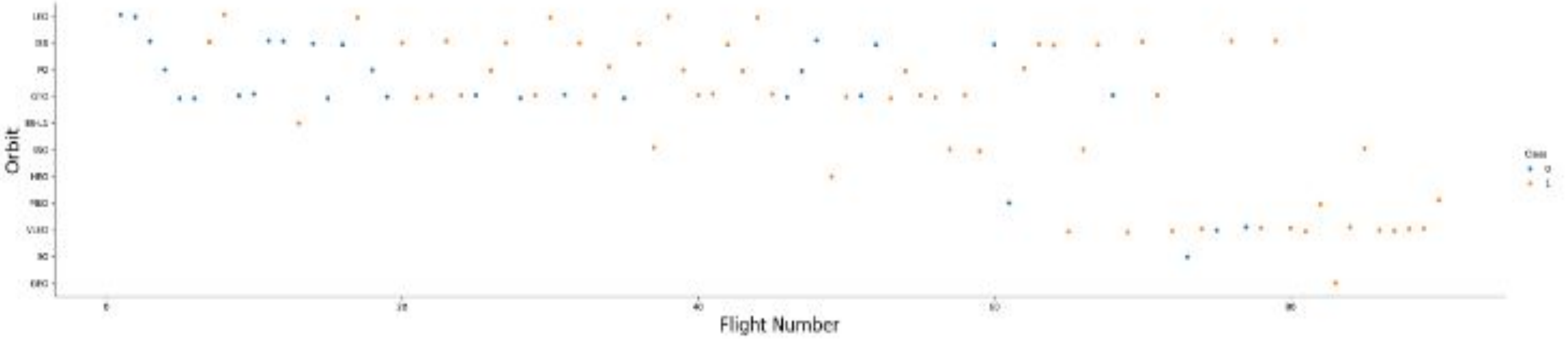
In [7]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
sns.catplot(x= 'Orbit', y = 'Class', data = df.groupby('Orbit')['Class'].mean().reset_index(), kind = 'bar')
plt.xlabel('Orbit Type',fontsize=20)
plt.ylabel('Success Rate',fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

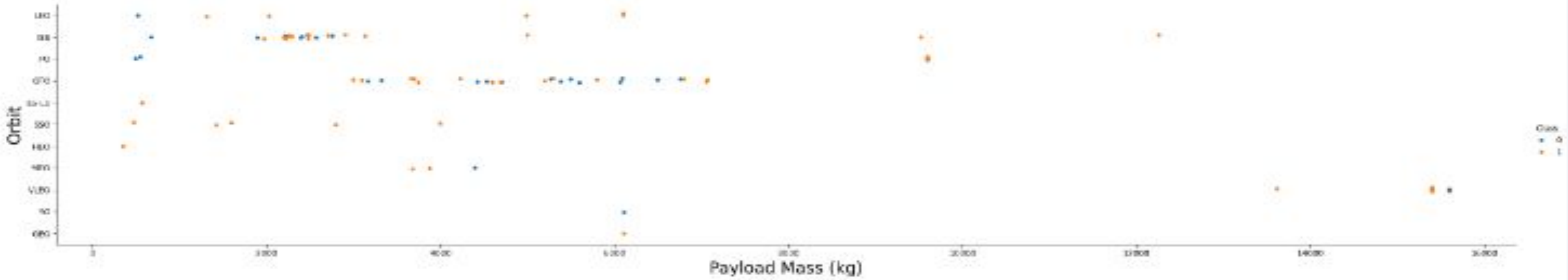
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x = 'FlightNumber', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Flight Number', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```



Payload vs. Orbit Type

Similarly, we can plot the Payload Mass vs. Orbit scatter point charts to reveal the relationship between Payload Mass and Orbit type

```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(x = 'PayloadMass', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Payload Mass (kg)', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```

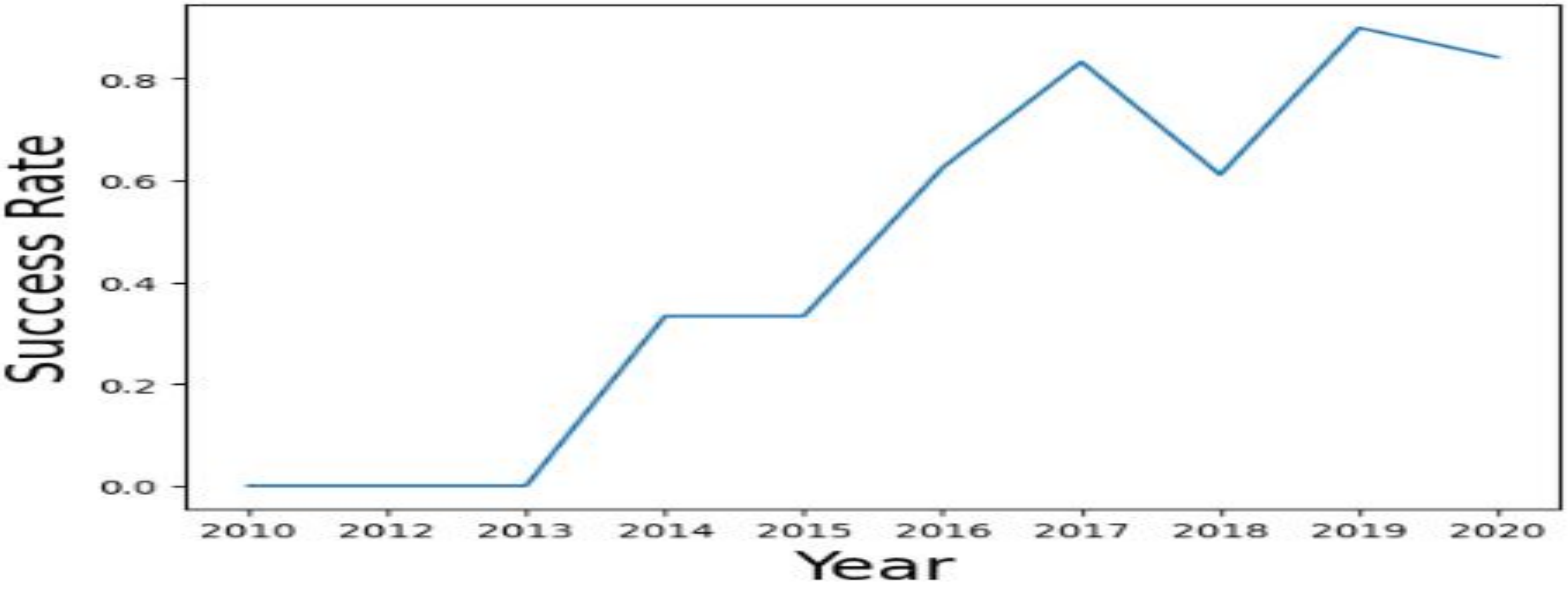


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rat
# Plot the line chart if there is data
sns.lineplot(x=years.index, y=years)
plt.xlabel('Year', fontsize=20)
plt.ylabel('Success Rate', fontsize=20)
plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

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

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE LIMIT 5
```

```
* sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

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

```
%sql SELECT * FROM SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
* sqlite:///my_data1.db
Done.
```

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass  FROM SPACEXTABLE WHERE customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTABLE WHERE booster_version LIKE '%F9 v1.1%';
```

```
* sqlite:///my_data1.db
Done.
```

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT MIN(date) AS first_successful_landing FROM SPACEXTABLE WHERE landing_outcome = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT booster_version FROM SPACEXTABLE WHERE landing_outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
* sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, COUNT(*) AS total_number FROM SPACEXTABLE GROUP BY mission_outcome;
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

```
%sql SELECT booster_version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

```
* sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
%sql SELECT substr(Date, 6, 2) AS month, Date, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome = 'Failur
```

```
* sqlite:///my_data1.db
Done.
```

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	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)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
%sql SELECT Landing_Outcome, COUNT(*) AS count_outcomes FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'GROUP BY Landing_Out
```

```
* sqlite:///my_data1.db
Done.
```

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

Section 3

Launch Sites Proximities Analysis

All launch sites' location markers on a global map

The generated map with marked launch sites should look similar to the following:

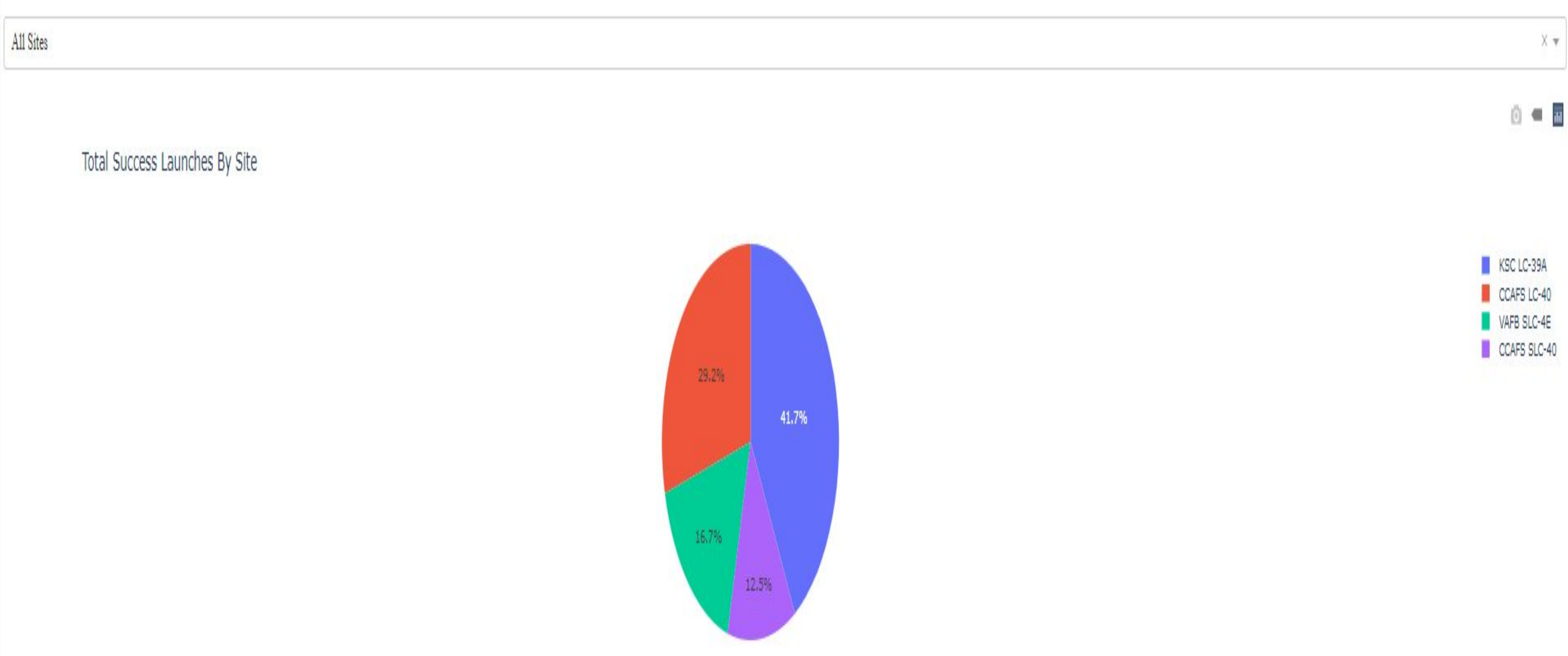
Color-labeled launch outcomes on the map

Proximities such as railway, highway, coastline, with distance calculated and displayed

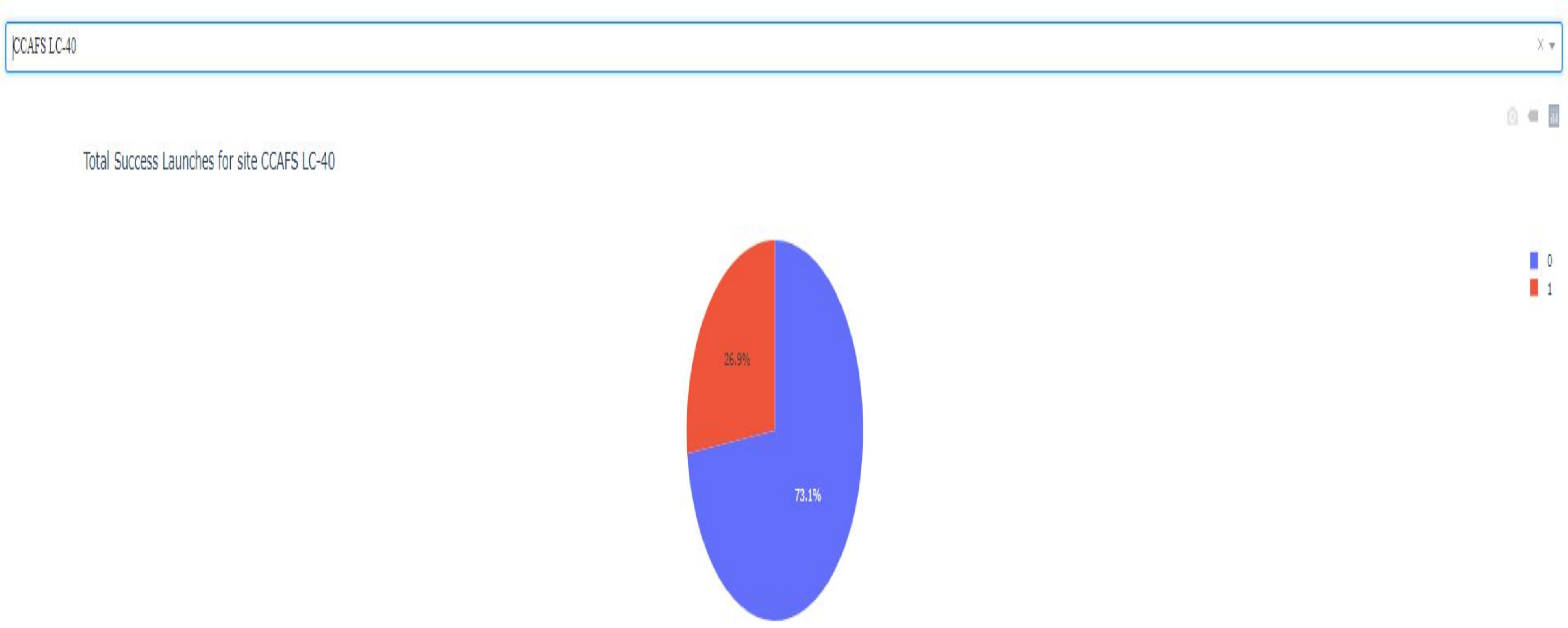
Section 4

Build a Dashboard with Plotly Dash

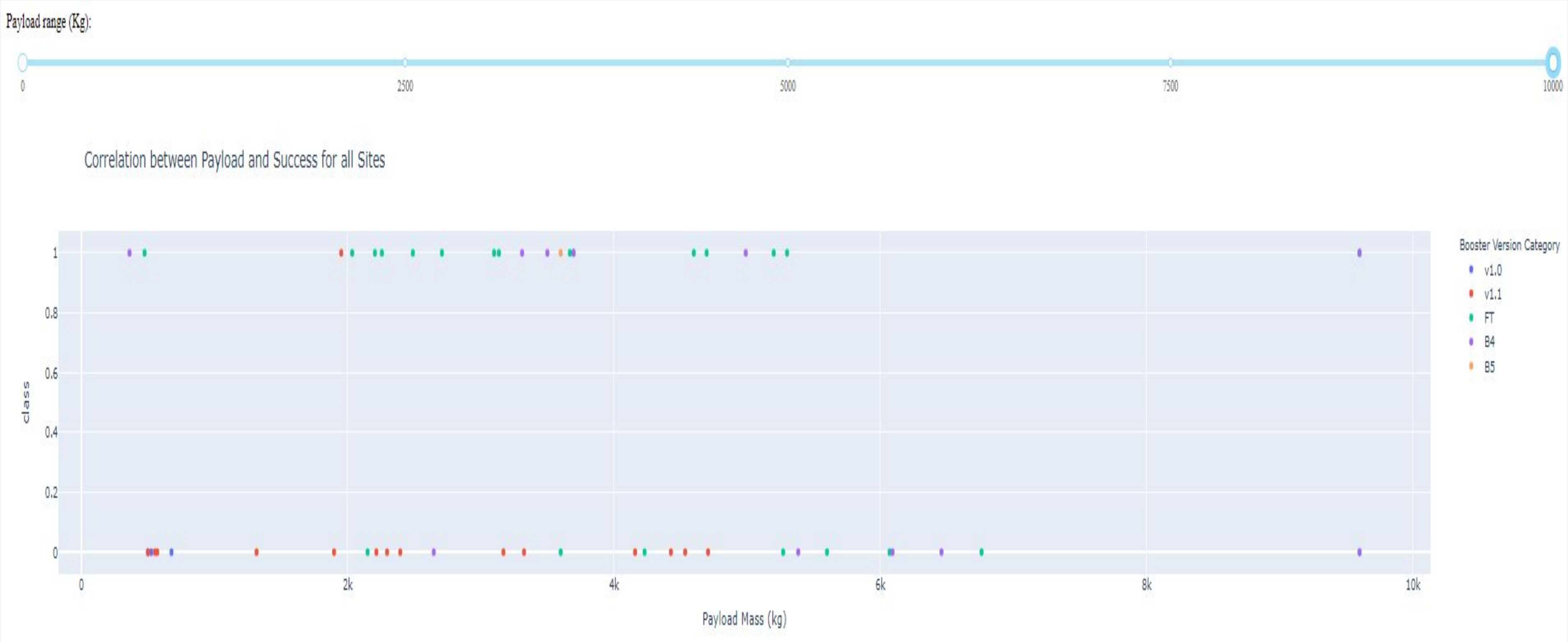
Piechart launch success count for all sites



Piechart for the launch site with highest launch success ratio



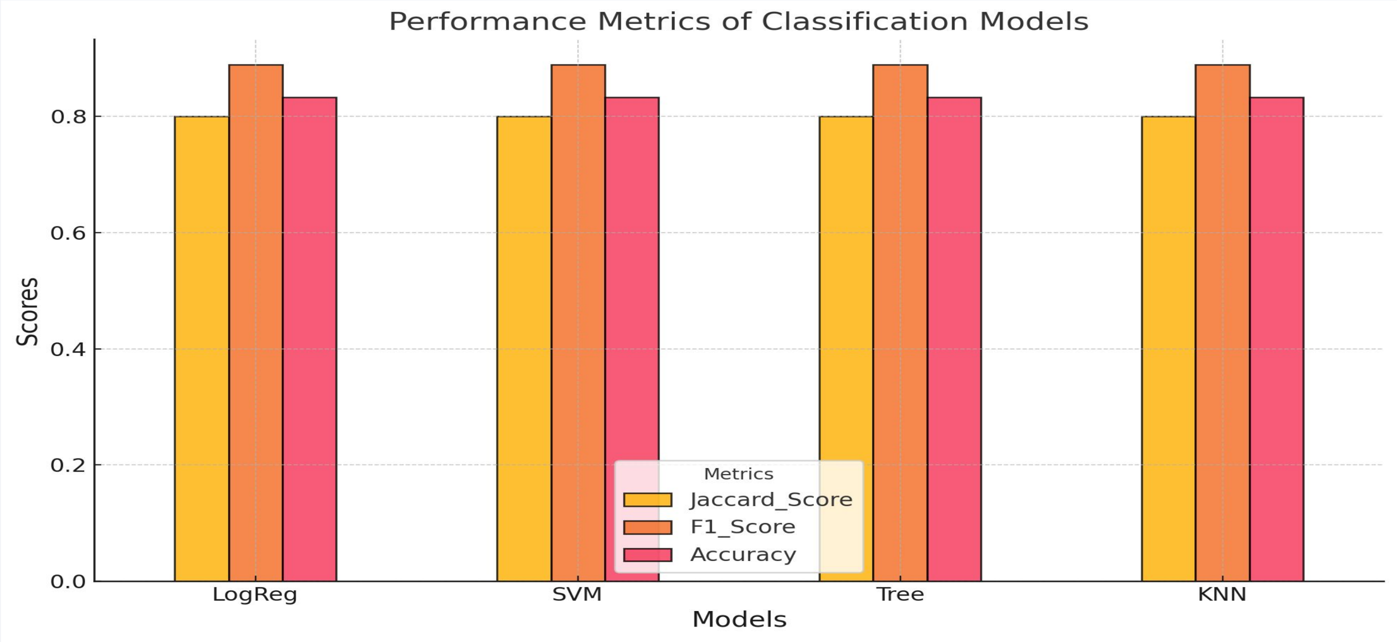
Payload vs. Launch Outcome



Section 5

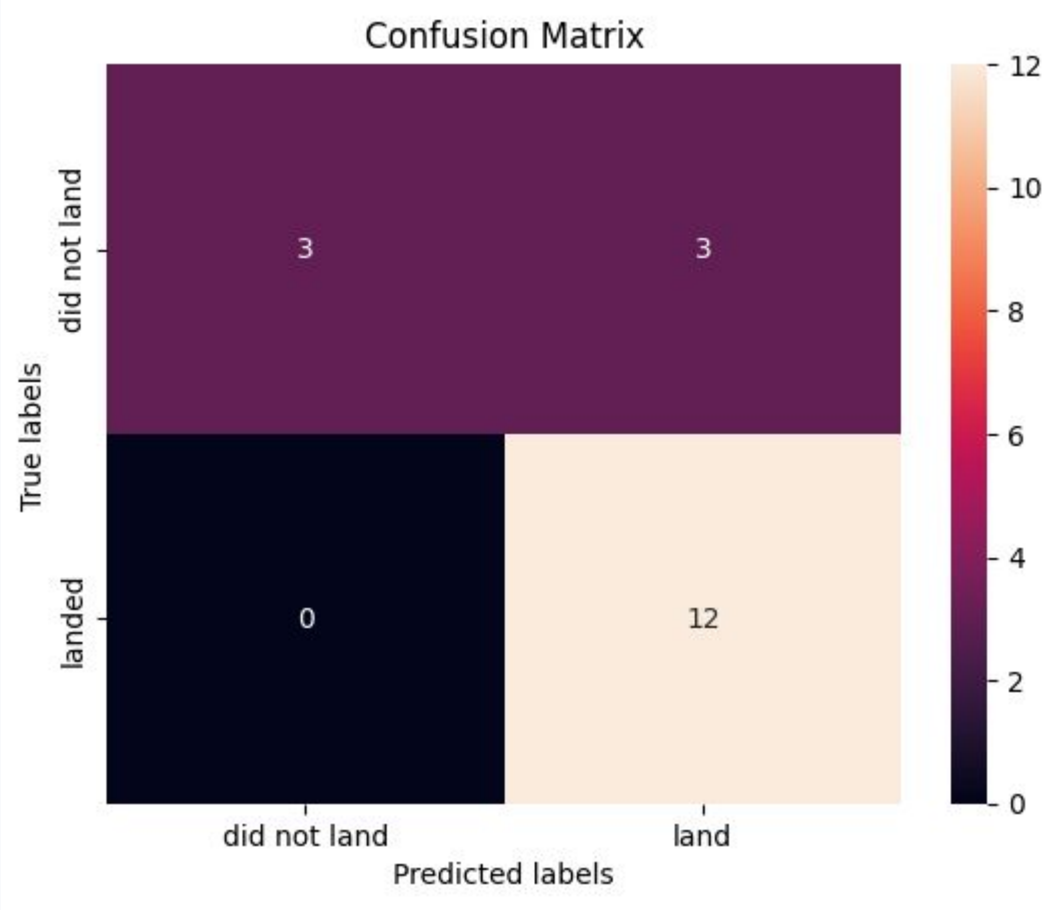
Predictive Analysis (Classification)

Classification Accuracy



Here is the bar chart visualizing the performance metrics (Jaccard Score, F1 Score, and Accuracy) for all classification models (LogReg, SVM, Tree, and KNN). Each model demonstrates identical performance across all metrics, making it easier to compare.

Confusion Matrix



Conclusions

- KSC LC-39A and VAFB SLC 4E launch sites have demonstrated higher success rates, making them key locations for future mission optimizations.

- Payload mass and orbit type are critical factors influencing landing success. Specifically, heavier payloads tend to have higher success rates for certain orbits.

- The steady improvement in success rates since 2013 indicates that SpaceX's iterative approach and technological advancements are yielding significant results.

Conclusions

- Predictive models, particularly the Decision Tree Classifier, achieved high accuracy (85%) and can be instrumental in forecasting landing outcomes, aiding operational decisions and cost savings.

- Interactive tools, such as dashboards with dynamic filters and visualizations, provide valuable insights for analyzing launch performance and identifying patterns across key variables.

- By leveraging these insights, SpaceX can maintain its competitive advantage in the space exploration market, reducing costs and improving mission reliability.

Appendix

1. **Data Collection API Lab** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb
2. **Data Collection with Web Scraping** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb
3. **Data Wrangling** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb
4. **EDA with Visualization** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/edadataviz.ipynb
5. **EDA with SQL** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb
6. **Interactive Visual Analytics with Folium** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/DV0101EN-Exercise-Generating-Maps-in-Python.ipynb
7. **Build an Interactive Dashboard with Ploty Dash** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb
8. **Machine Learning Prediction** - https://github.com/Frey87/IBM_Courses/blob/8228645a7e0cbb9a48117c5d6b4a7cf7462ac11d/Data%20Science/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Thank you!