SpaceX Launch Prediction and Analysis

Predicting First-Stage Landing Success using Machine Learning and Data Analytics

Executive Summary

- This project aims to predict the success of Falcon 9 first-stage landings using machine learning, focusing on optimizing launch costs and operational efficiency for SpaceX. By analyzing historical launch data, we identify patterns that determine whether the first-stage landing will be successful.

- Key Steps:
- 1. Data Collection: SpaceX's API, web scraping from Wikipedia.
- 2. Data Wrangling: Data cleaning and preprocessing.
- 3. Exploratory Data Analysis (EDA): Visualizing key factors.
- 4. Predictive Modeling: Logistic Regression, SVM, Decision Trees, KNN.
- 5. Results: Decision Tree emerged as the best model.

Introduction

- SpaceX has revolutionized the space industry with the Falcon 9, a reusable rocket designed to reduce launch costs by landing the first stage.

- This project focuses on predicting whether the first stage of Falcon 9 will land successfully. Accurate predictions can help optimize SpaceX's launch cost-efficiency strategy.

Problem Statement

- SpaceX's Falcon 9 launches cost $62 million, but reusing the first stage saves a significant amount of money.
- To make SpaceX's operations more efficient, predicting whether the first stage will land successfully is crucial.
- Accurate predictions could help SpaceX achieve cost savings and optimize operations by preventing unsuccessful landings.
- This project aims to develop a model to predict landing success based on historical data and key features like payload, launch site, and booster version.

Data Collection Overview

- The data used in this project was collected using two methods:
- 1. SpaceX API : Data was fetched using the SpaceX API, including launch records, payload details, and launch outcomes.
- 2. Web Scraping: Web scraping from Wikipedia to gather Falcon 9 launch records.
- Data from these sources was cleaned, preprocessed, and integrated for analysis and modeling.

API Data Collection Process

- 1. API Requests : Data was fetched using the SpaceX API with GET requests.
- 2. Data Processing: The API response was decoded using `.json()` and converted to a pandas DataFrame.
- 3. Data Integration: The data was cleaned and merged with additional data scraped from Wikipedia.

- The resulting dataset was used for exploratory analysis and machine learning modeling.

Web Scraping Overview

- 1. Scraping from Wikipedia : Launch data was scraped from the Falcon 9 Wikipedia page using BeautifulSoup.
- 2. Data Parsing: HTML tables were parsed, and the relevant data (e.g., launch dates, payloads) was extracted.
- 3. Data Conversion: The data was converted into a pandas DataFrame and integrated with the SpaceX API data.

- The combined dataset provided the foundation for the next steps in the project: EDA and machine learning modeling.

Data Wrangling - Cleaning and Preprocessing

- 1. Missing Values: Missing values were handled by filling or dropping based on the feature.
- 2. Feature Encoding: Categorical features such as launch site and booster version were encoded using one-hot encoding.
- 3. Feature Scaling: Features were scaled to ensure that models could work efficiently.

- After these steps, the dataset was ready for exploratory analysis and model training.

Exploratory Data Analysis (EDA)

- Exploratory Data Analysis (EDA) was performed to uncover relationships in the data. Key findings included:

- - Success rate differences between launch sites.
- - The impact of payload mass on launch outcomes.
- - Yearly success trends and how they correlate with other variables.

EDA – Success Rate vs Launch Site

- By analyzing launch site data, it was observed that the success rate of landings varied significantly by site. For example, the KSC LC-39A site showed higher success rates compared to other sites.

- Visualizations, such as pie charts, were used to show the success distribution for each site.

EDA – Payload Mass vs Launch Outcome

- The analysis revealed a correlation between payload mass and the success of the landing. Lighter payloads were often associated with successful landings, while heavier payloads had mixed results.

- This relationship was visualized through scatter plots and bar graphs to make it easier to identify trends.

SQL-Based Analysis Overview

- SQL queries were applied to extract deeper insights from the data, focusing on:

- - Launch site success rates.
- - Payload mass carried by boosters.
- - The correlation between orbit type and landing success.

SQL Query Example 1 - Successful vs Failed Launches

- A SQL query was used to find the total number of successful and failed launches for each launch site.

- Example query:
- SELECT `Launch Site`, COUNT(*) AS `Total Launches`, SUM(`Class`) AS `Successful Launches` FROM `spacex_data` GROUP BY `Launch Site`;

- The results provided insights into which sites had the most successful landings.

SQL Query Example 2 - Payload Mass by Booster Version

- Another SQL query explored the average payload mass for different booster versions.

- Example query:
- SELECT `Booster Version`, AVG(`Payload Mass (kg)`) AS `Average Payload Mass` FROM `spacex_data` GROUP BY `Booster Version`;

- This analysis revealed how payload capacity impacted the likelihood of a successful landing.

SQL Query Results - Launch Site Success Rate

- The query results highlighted which launch sites had the highest success rates for landing. For example, KSC LC-39A had a high success rate of over 90%, while VAFB SLC-4E had a lower rate.

- These results helped identify the most reliable launch sites.

Interactive Analytics with Folium - Map Overview

- A map was created using Folium to visualize the success rate of launches for each launch site. Each site was marked on the map with a color-coded circle representing success (green) or failure (red).

- The map provided insights into spatial relationships between launch sites and landing success.

Interactive Analytics with Plotly Dash - Overview

- **Plotly Dash** was used to create an interactive dashboard that visualized the relationship between payload mass and launch success.

- Key features of the dashboard:
- 1. Dropdown menu to select launch site.
- 2. Pie chart to show success vs. failure for each site.
- 3. Scatter plot to analyze payload mass vs launch success.

Plotly Dash - Launch Site Success Distribution

- A pie chart was created in the dashboard to visualize the success distribution for each launch site.

- This interactive chart allowed users to explore the percentage of successful vs failed launches at each site.

Plotly Dash - Payload vs Success

- A scatter plot was added to the dashboard to visualize the relationship between Payload Mass (kg) and Launch Success.

- The scatter plot helped identify patterns and correlations between the two features, aiding in predicting success based on payload.

Predictive Analysis Overview

- A machine learning pipeline was implemented to predict whether the Falcon 9 first-stage will land successfully.

- Key classification models used:
- - Logistic Regression
- - Support Vector Machines (SVM)
- - Decision Trees
- - K-Nearest Neighbors (KNN)

- The models were trained and evaluated on the dataset to determine the best classifier for the task.

Logistic Regression Model

- Logistic Regression was used as a baseline classifier to predict landing success.
- Model parameters included regularization strength and the solver type, which were tuned using GridSearchCV.
- Logistic Regression helped establish a baseline for comparison with more complex models.

SVM Model

- Support Vector Machines (SVM) were used to classify landing success with a non-linear kernel.

- SVM was chosen for its robustness in handling high-dimensional data and its ability to work well in complex datasets.

- The SVM model showed competitive performance but required careful tuning of kernel parameters.

Decision Tree Model

- Decision Trees were used for classification, where the dataset is split based on feature values to predict the target variable.

- Key hyperparameters were tuned, including tree depth and max features.

- The Decision Tree model provided excellent interpretability and achieved high accuracy.

K-Nearest Neighbors (KNN) Model

- K-Nearest Neighbors (KNN) was used to classify landing success based on proximity to nearest data points.
- KNN is a simple algorithm that requires careful tuning of the number of neighbors parameter.
- While KNN performed decently, its accuracy was lower than Decision Trees and SVM.

Limitations and Challenges

- Despite the comprehensive analysis, some limitations and challenges remain:
- - Data Quality: Some missing data points were imputed, but this could impact accuracy.
- - Model Complexity: While decision trees performed well, other models like KNN showed lower performance.
- - Feature Selection: Some features were not fully explored, and more detailed feature engineering could improve results.

Future Work – Model Improvement

- The model could be improved in several ways:

- - Feature Engineering: Including additional features like weather conditions or booster type.
- - Hyperparameter Tuning: Further tuning of models to reduce errors and improve prediction accuracy.
- - Ensemble Methods: Combining multiple models (e.g., Random Forests, XGBoost) for better performance.

Future Work – Additional Features

- Future work could involve integrating additional features such as:

- - Weather Data: Including weather conditions during launch could provide more insights.
- - Booster Details: Specific details on the boosters used in launches might affect success rates.
- - Launch Site Proximity: Distance from certain landmarks might influence launch outcomes.

Final Thoughts and Acknowledgments

- This project demonstrated the power of data analytics and machine learning to optimize SpaceX's operations.

- Acknowledgments:
- - SpaceX for providing the data.
- - Tools used: Python, Pandas, Plotly, Scikit-Learn, Folium, SQL.

- The insights gained will contribute to SpaceX's ability to predict landing success and reduce operational costs.