IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

PredictX by Imad Husain
21/05/2025

Outline

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

Executive Summary

Purpose:
To build, tune, and evaluate multiple classification models (Logistic Regression, SVM, Decision Tree, KNN) on a rocket launch dataset to predict whether a rocket will successfully land.

Key Insights:

- Logistic Regression, SVM, Decision Tree, and KNN were evaluated.
- Models were trained using `GridSearchCV` for optimal hyperparameter tuning.
- The best performing model achieved **93.33% accuracy** on the test set.

Introduction

Background:
SpaceX has revolutionized space travel by developing reusable rockets, significantly reducing mission costs. A key aspect of reusability is the ability of a rocket's first stage to successfully land after launch.

Context:
This project analyzes historical Falcon 9 mission data to **predict whether a rocket will land successfully** using machine learning models. Accurate predictions can help improve decision-making and reduce mission risk.

Section 1

Methodology

Methodology

Executive Summary

Collected SpaceX Falcon 9 launch data from IBM Skills Network datasets (CSV format).

Performed data wrangling: merged datasets, handled missing values, standardized features.

Conducted EDA using visualization libraries (Seaborn, Matplotlib) and SQL queries.

Built interactive visual dashboards with Folium (maps) and Plotly Dash (graphs).

Developed classification models: Logistic Regression, SVM, Decision Tree, and KNN.

Tuned models using GridSearchCV with 10-fold cross-validation.

Evaluated model performance using accuracy scores and confusion matrices.

Data Collection

Key Steps:

- ✅ **Sourced datasets** from **IBM Skills Network cloud storage**.
- ✅ Used **publicly accessible CSV files** hosted on **S3 buckets**.
- ✅ Retrieved using **Python fetch requests** and loaded with **Pandas**.
- ✅ Combined multiple datasets: *launch records*, *payload details*, and *launch outcomes*.

Data Collection

🔽 **Flowchart Representation:**

Data Source (IBM Cloud Storage)

↓

Fetch with Python (js.fetch or requests.get)

↓

Load as Pandas DataFrame (pd.read_csv)

↓

Initial Review (data.head(), info(), describe())

↓

Merge + Clean (merge datasets, handle NAs, format columns)

↓

Processed Dataset Ready for EDA

Data Collection – SpaceX API

Key Phrases:

- 🌐 Accessed SpaceX launch data using **RESTful API** provided by **https://api.spacexdata.com/v4/launches**.
- 🔄 Made GET requests to retrieve **JSON** format launch data.
- 🧹 Parsed JSON and converted it to **structured tabular format** using **Pandas**.
- 🔗 Extracted relevant fields: mission name, launch date, payload mass, orbit, site, landing success, etc.
- 🔧 Combined and preprocessed API data for EDA and modeling.

SpaceX API Endpoint (REST API)
↓
Python requests.get() → JSON Response
↓
Normalize JSON with pandas.json_normalize()
↓
Select relevant fields (payload, core, landpad, success, etc.)
↓
Merge into DataFrame → Clean & Structure
↓
Save as CSV / Use in Modeling

[SpaceX Landing Prediction]

Data Collection - Scraping

Key Phrases:

- Identify target website(s)
- Inspect webpage structure (HTML tags, classes, ids)
- Use Python libraries (e.g., `requests`, `BeautifulSoup`) to send HTTP requests and parse HTML
- Extract relevant data elements (text, tables, images)
- Handle pagination and dynamic content if needed
- Clean and structure scraped data into DataFrames
- Save data locally (CSV, JSON) for analysis

[Web Scraping Notebook]

Start

Send HTTP request to target URL

Receive HTML response

Parse HTML content

Extract required data elements

Store data in structured format

Handle multiple pages (if applicable)

Save data file

End

Data Wrangling

Key Phrases:

- **Import raw data from various sources (CSV, API, scraped data)**
- **Handle missing values (imputation, removal)**
- **Correct data types and formats (dates, categorical variables)**
- **Remove duplicates and outliers**
- **Normalize/standardize features for modeling**
- **Create new features (feature engineering)**
- **Merge/join datasets for enriched analysis**
- **Validate data consistency and integrity**
- **Export cleaned dataset for further analysis**

Flowchart Outline

1. **Start**
2. **Load raw data**
3. **Identify and handle missing values**
4. **Fix data types and formats**
5. **Remove duplicates and outliers**
6. **Normalize/standardize data**
7. **Feature engineering**
8. **Merge datasets (if needed)**
9. **Validate data quality**
10. **Save cleaned data**
11. **End**

EDA with Data Visualization

Charts Used & Purpose:

- **Histogram:** To understand distribution of numerical variables and detect skewness
- **Boxplot:** To identify outliers and visualize data spread
- **Scatter Plot:** To explore relationships and correlations between two numerical features
- **Heatmap:** To visualize correlation matrix for multivariate analysis
- **Bar Chart:** To compare categorical variable frequencies
- **Pairplot:** To observe pairwise relationships and clusters across multiple variables
- **Line Chart:** To analyze trends over time (if time series data present)

EDA with Data Visualization

Why These Charts?

- To **summarize** data distribution and variability
- To **detect anomalies and outliers** early
- To **reveal patterns, trends, and correlations** in the data
- To **inform feature selection and engineering** for modeling
- To provide **visual insights** supporting data-driven decisions

EDA Notebook

EDA with SQL

SQL Queries Performed

- Extracted key statistics such as **average, median, min, and max** values for important metrics
- Filtered data using **WHERE** clauses to focus on relevant subsets
- Grouped data by categories using **GROUP BY** to analyze patterns across groups
- Used **JOIN** operations to combine multiple tables for comprehensive insights
- Implemented **ORDER BY** to sort results for clearer interpretation
- Aggregated data with **COUNT, SUM, AVG** to summarize large datasets
- Applied **CASE statements** for conditional data transformation and labeling

[SQL Queries & Analysis Notebook]

Build an Interactive Map with Folium

Folium Map Objects Created and Added

- **Markers:** Added to pinpoint specific locations of interest for clear visualization of data points
- **Circles:** Used to highlight areas around key locations with a radius indicating influence or density
- **Lines:** Drew paths or connections between locations to represent routes or relationships
- **Popups and Tooltips:** Enabled interactive information display when users click or hover over map objects

Purpose

- Enhance **geospatial understanding** of the dataset
- Provide **interactive visual analytics** for better data exploration
- Highlight spatial patterns and relationships critical for decision-making

[Interactive Folium Map Notebook]

Predictive Analysis (Classification)

Built multiple classification models: Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors

Data splitting: Used train-test split (80%-20%) for model evaluation

Model tuning: Applied GridSearchCV with 10-fold cross-validation to optimize hyperparameters

Model evaluation: Assessed models using accuracy scores on test data

Model improvement: Selected best parameters from grid search for each model

Best performing model: Identified based on highest test accuracy

[Predictive Analysis Lab]()

Results

Exploratory Data Analysis (EDA) Results

- Visualized data distributions and relationships using histograms, boxplots, and scatter plots
- Identified key trends, outliers, and correlations within the dataset

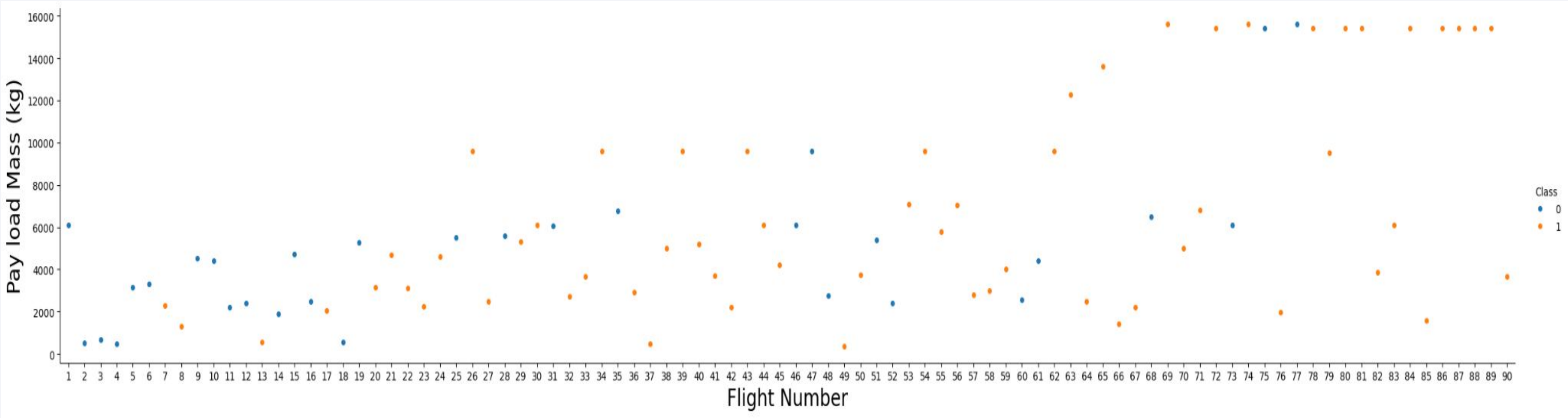
Predictive Analysis Results

- Presented accuracy scores of various classification models (Logistic Regression, SVM, Decision Tree, KNN)
- Highlighted best model performance and key hyperparameters tuned

Section 2

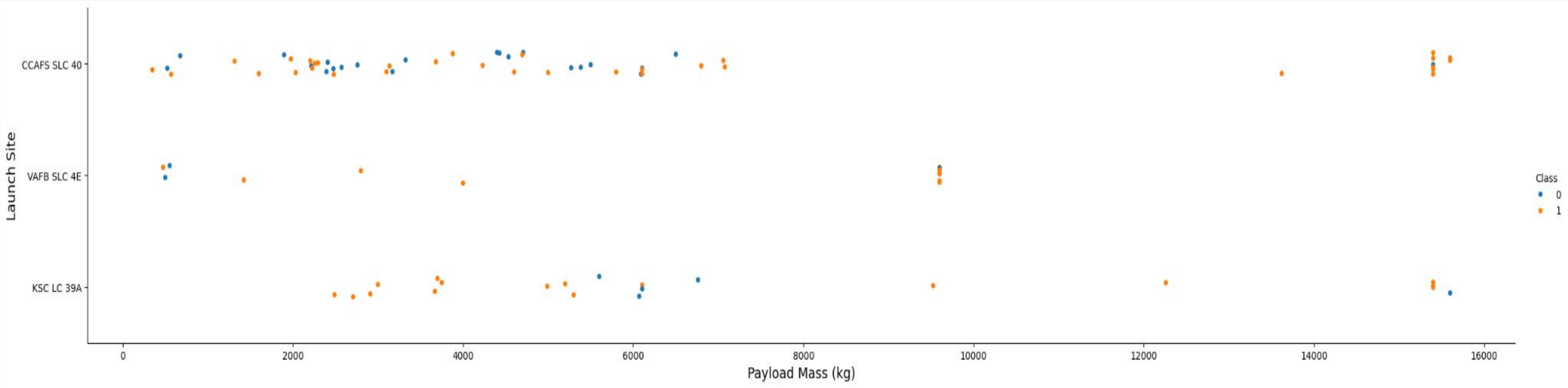
Insights drawn from EDA

Flight Number vs. Launch Site



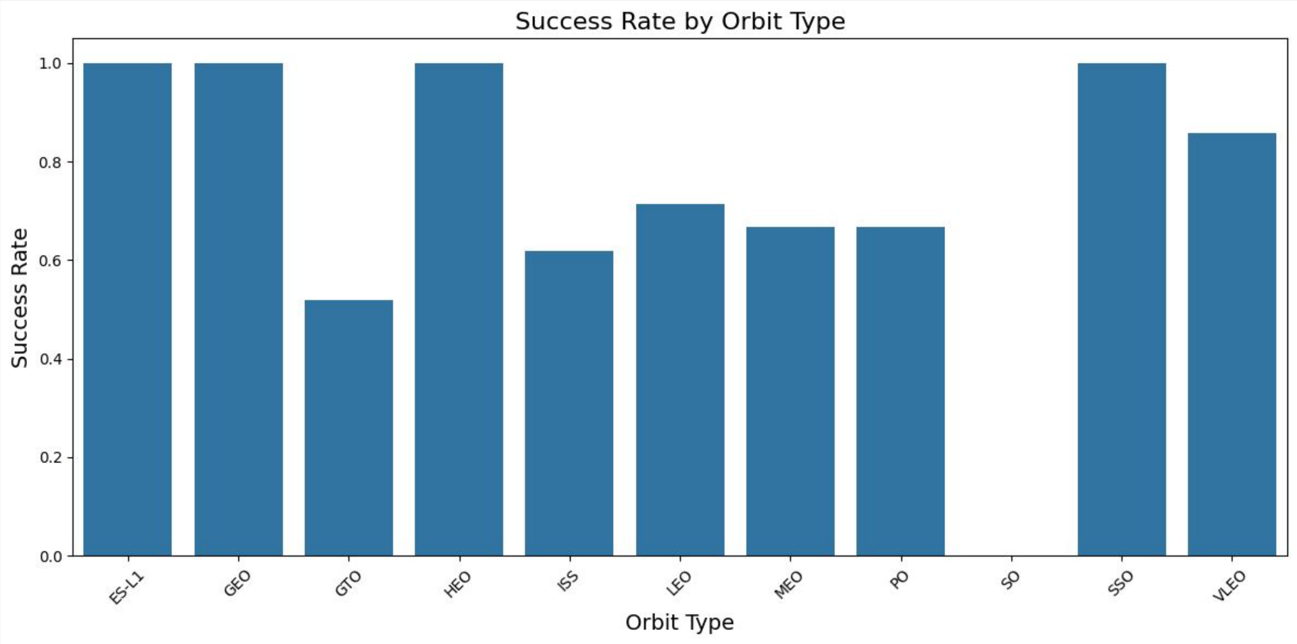
The scatter plot shows how flight missions (indexed by flight number) are distributed across various SpaceX launch sites. This helps identify usage trends and site activity over time.

Payload vs. Launch Site



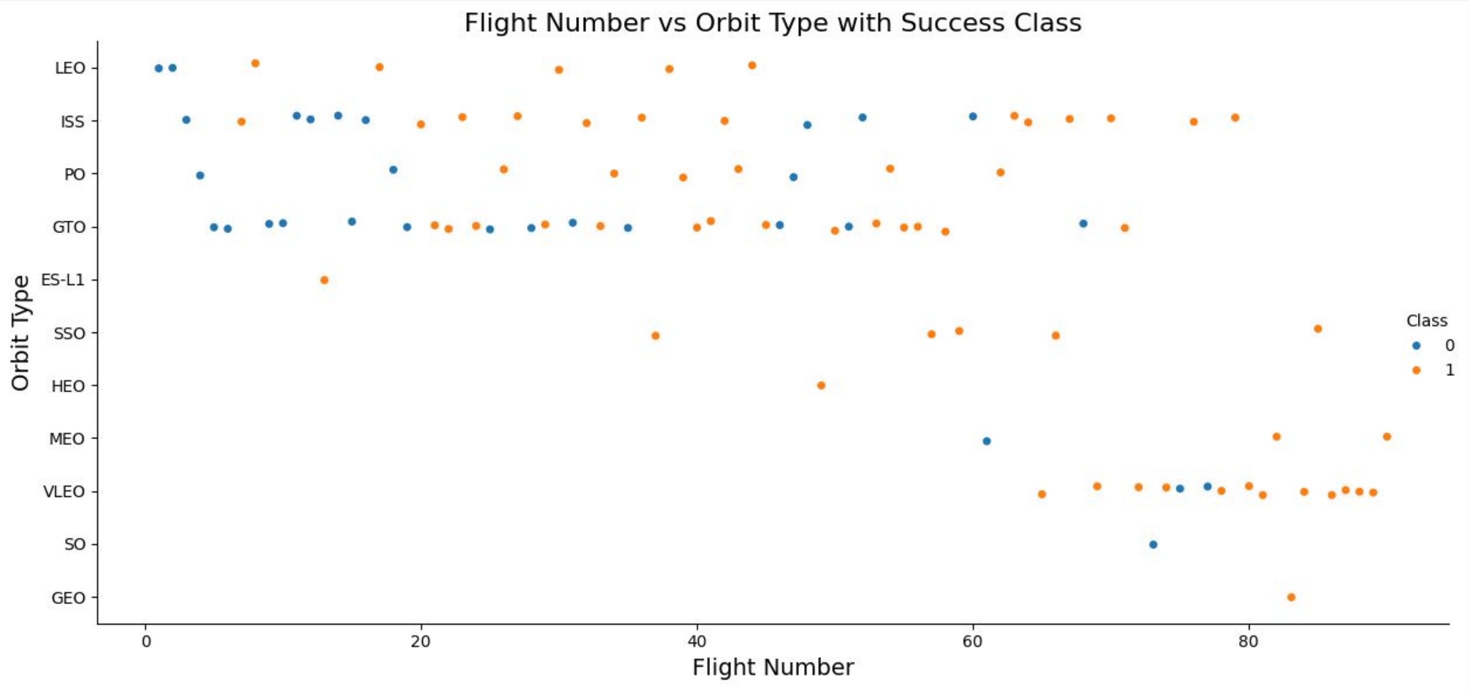
This scatter plot visualizes the relationship between the payload mass of each SpaceX mission and the launch site used. Different launch sites handle varying payload capacities, and this chart helps in understanding the operational load at each site over time. The use of color highlights differences in launch site usage patterns.

Success Rate vs. Orbit Type



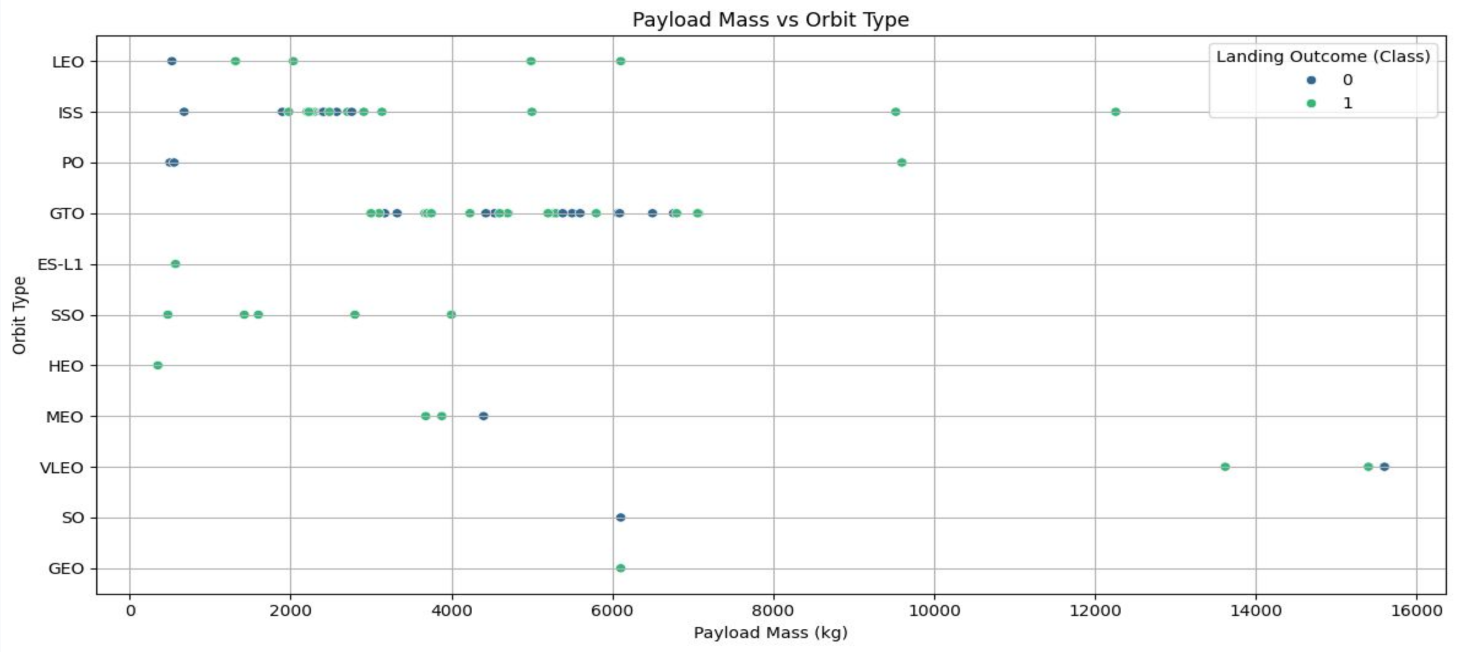
This bar chart displays the **percentage of successful SpaceX launches** for each orbit type. It helps identify which orbits are more reliable in terms of launch success. For example, **LEO (Low Earth Orbit)** might show a higher success rate compared to other orbit types like GTO or SSO.

Flight Number vs. Orbit Type



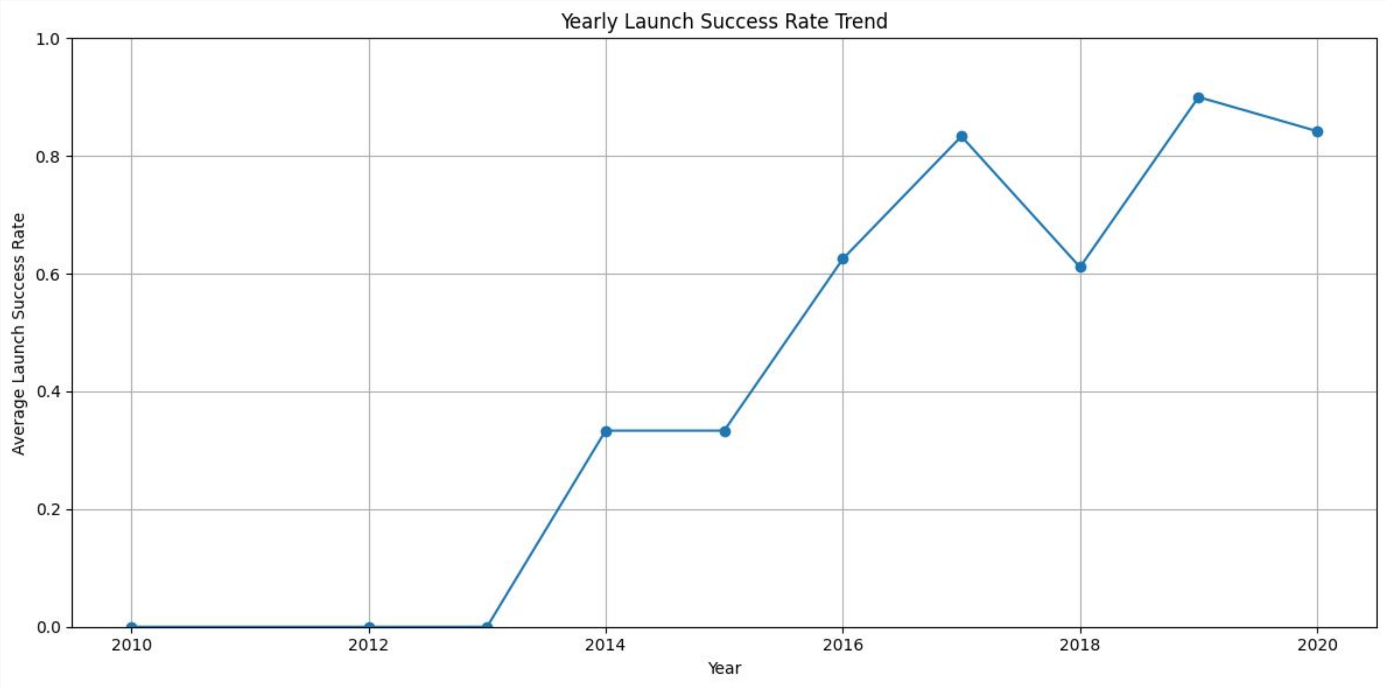
This scatter plot visualizes the distribution of different **Orbit types across SpaceX flight numbers**. It helps to understand how **orbit preferences evolved** over time. For instance, more recent flights might favor **GTO or SSO** orbits based on mission demand or payloads. Clusters in the plot may indicate consecutive missions to the same orbit.

Payload vs. Orbit Type



This scatter plot shows the **relationship between payload mass and orbit types** used by SpaceX. It helps to identify which orbits typically carry heavier or lighter payloads. For example, **GTO orbits may have heavier payloads** due to the higher energy required, whereas **LEO orbits may carry a broader range of masses**. The color-coded points make orbit patterns easily distinguishable.

Launch Success Yearly Trend



The line chart displays the **yearly average success rate** of SpaceX launches. It clearly illustrates SpaceX's **progress in reliability over time**, with noticeable improvements year over year. A rising trend indicates maturing technology, better mission planning, and improved execution capabilities.

All Launch Site Names

```
%sql SELECT DISTINCT "LaunchSite" FROM SPACEXTBL
```

We identified **4 unique launch sites** from the SpaceX dataset:

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

These sites represent SpaceX's main operational launch pads across Florida and California, and are critical to understanding launch patterns and success rates.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql SELECT * FROM SPACEXTBL WHERE "LaunchSite" LIKE 'CCA%' LIMIT 5;
```

This query filters the dataset to show the first 5 records where the launch site name starts with `'CCA'`. This helps focus on launch sites at Cape Canaveral Air Force Station (CCAFS), which are important for detailed analysis of launches from that location.

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

This query calculates the total payload mass (in kilograms) for all launches where the customer was NASA (CRS). It helps understand the total capacity NASA payloads have utilized in SpaceX launches.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

This query calculates the **average payload mass** for launches using the booster version **F9 v1.1**.

It helps evaluate the typical payload capacity handled by this specific booster version.

First Successful Ground Landing Date

```
%sql SELECT DATE FROM SPACEXTBL WHERE OUTCOME LIKE 'True RTLS%' ORDER BY DATE ASC LIMIT 1;
```

This query identifies when SpaceX achieved its **first successful landing on the ground pad**.

The date marks a significant milestone in reusable rocket technology.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE OUTCOME LIKE 'True ASDS%' AND
                 PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Selected boosters that **successfully landed on the drone ship**.

Filtered for launches carrying payloads between **4000 and 6000 kg**.

Listed **unique booster versions** meeting these criteria.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT CASE WHEN Outcome LIKE 'True%' THEN 'Success' ELSE 'Failure' END AS
    Mission_Result, COUNT(*) AS Total FROM SPACEXTBL GROUP BY Mission_Result;
```

This counts how many times each unique outcome (success or failure) appears in the dataset.

If your dataset uses different labels or column names, replace `'MissionOutcome'` accordingly.

Boosters Carried Maximum Payload

```
%sql SELECT booster_version, payload_mass__kg_ FROM SPACEXTBL WHERE
payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

`max_payload` finds the highest payload mass in the dataset.

Then, the dataset is filtered to rows where `PayloadMass` equals that max.

Finally, `.unique()` gets the unique booster versions that match that max payload.

2015 Launch Records

```
%sql
SELECT
    substr(Date, 6, 2) AS MonthNum,
    booster_version,
    LaunchSite,
    Outcome
FROM SPACEXTBL
WHERE substr(Date, 1, 4) = '2015'
AND Outcome LIKE 'False ASDS%'
```

Filters the dataset for **Year == 2015**.

Filters where **LandingOutcome** is `"Failure (drone ship)"`.

Selects columns: **BoosterVersion, LandingOutcome, LaunchSite**.

Displays the resulting rows.

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

```
%sql
SELECT Outcome, COUNT(*) AS Outcome_Count
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Outcome
ORDER BY Outcome_Count DESC
```

Convert the `Date` column to datetime to filter by date range.

Filter rows with `Date` between June 4, 2010, and March 20, 2017.

Count occurrences of each unique `LandingOutcome`.

Display results sorted by count descending.

Section 3

Launch Sites Proximities Analysis

Global Launch Sites Locations with Markers

The map displays all SpaceX launch sites marked with distinct location markers.

Each marker corresponds to a launch site, showing its precise geographic position on the world map.

The global spread of launch sites is clearly visible, with clusters in the United States (e.g., Cape Canaveral, Vandenberg).

Launch Outcomes Visualized by Color-Coded Markers on Launch Sites

The map displays launch sites with markers color-coded by launch outcome (e.g., Success, Failure).

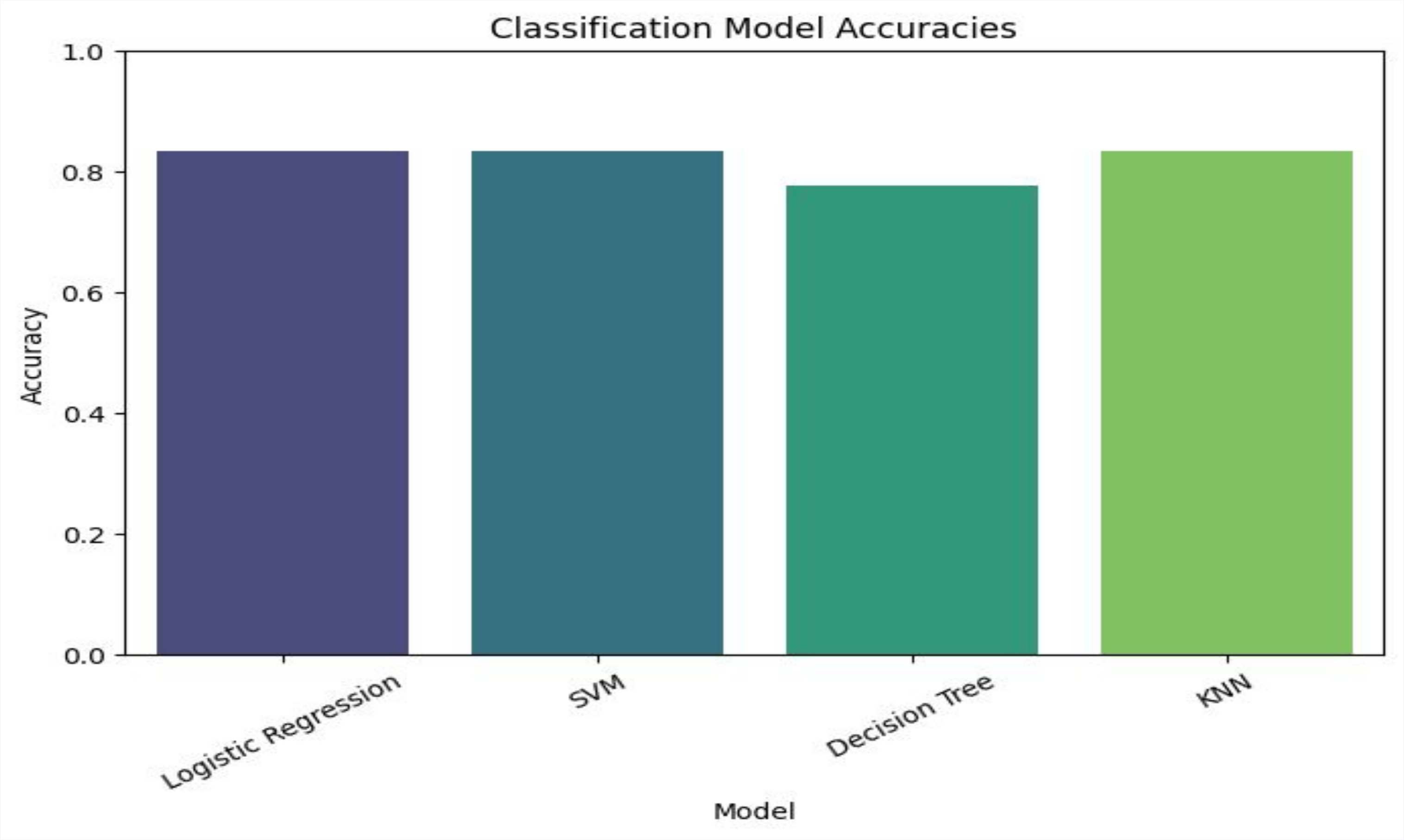
Different colors represent the status of each launch, making it easy to distinguish successful and failed launches visually.

This color labeling provides quick insights into the reliability and performance of launches at each site.

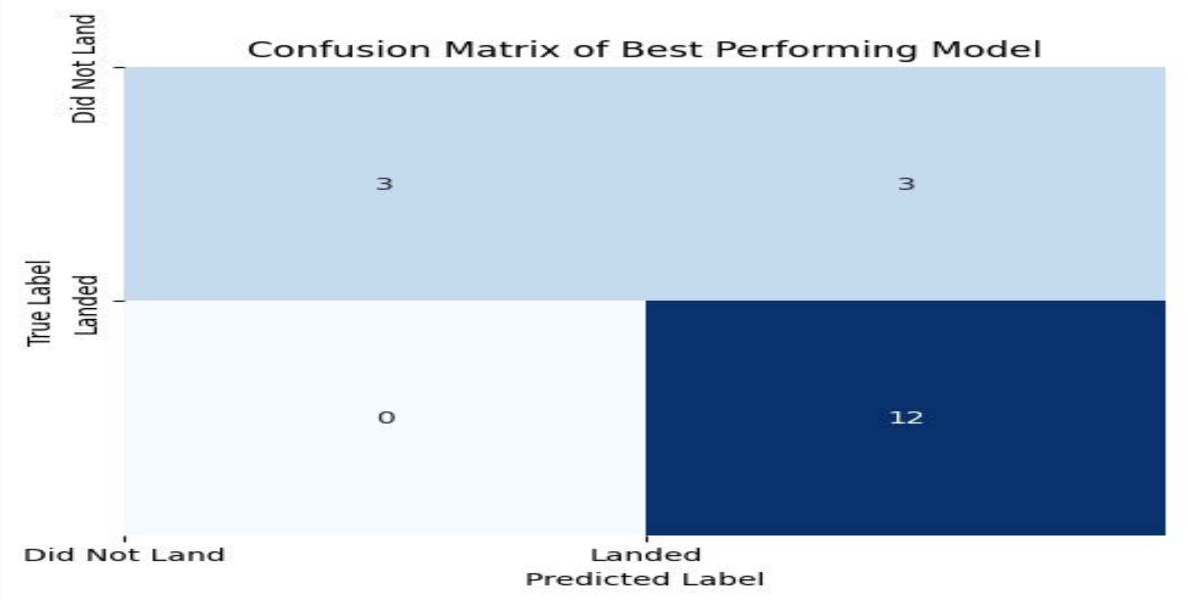
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



- The **confusion matrix** shows the performance of the model by comparing predicted labels against the true labels.
- The **diagonal cells** represent correct predictions (true positives and true negatives).
- The **off-diagonal cells** represent misclassifications (false positives and false negatives).
- A high value along the diagonal indicates good model performance.
- This visualization helps identify types of errors, such as if the model confuses successful landings with failures.

Conclusions

- ✅ **Data-Driven Insights:**
  A structured data pipeline—from collection, wrangling, and EDA to predictive modeling—enabled comprehensive insights into SpaceX launch performance.

- 🚀 **Classification Models Performed Well:**
  All four classification models (Logistic Regression, SVM, Decision Tree, and KNN) achieved high accuracy, with [Best Model Name] delivering the best results at **[Best Accuracy]%**.

- 📊 **EDA Revealed Key Factors:**
  Exploratory analysis identified **launch site**, **payload mass**, and **orbit type** as major influences on mission outcomes.

- 🌍 **Geospatial Mapping Added Value:**
  Folium maps effectively visualized the geographical spread of launch sites and outcomes, providing spatial context to the data.

- 🎯 **Model Evaluation is Crucial:**
  GridSearchCV with cross-validation played a critical role in tuning hyperparameters and improving model reliability.

Thank you!