IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Peace Owoeye
April, 2025

Outline

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

Executive Summary

The goal of this project was to develop a model for predicting the success of Falcon 9 first-stage landings using real-world SpaceX launch data, with the aim of supporting cost estimation and strategic decision-making in aerospace operations. A complete data pipeline was employed—starting with data collection through APIs and web scraping from SpaceX and Wikipedia, followed by data wrangling to merge and clean datasets while engineering a binary target variable for landing success.

Subsequent analysis involved exploratory data techniques to identify trends related to launch sites, payload mass, and orbital types. Geo-analytics in providing spatial context to the data. A dashboard for dynamic visualization of key metrics. Multiple machine learning models were trained and optimized to predict landing outcomes.

The analysis revealed that Low Earth Orbit (LEO) missions and mid-range payloads exhibited higher landing success rates. Launch site and payload mass stood out as the most influential features in predicting Falcon 9 first-stage landings, with Decision Tree and Logistic Regression models achieving approximately 85% prediction accuracy. This offers actionable insights for optimizing mission planning, reducing risk, and improving the cost-effectiveness of reusable rocket technology.

Introduction

SpaceX has revolutionized space travel through the development of reusable launch vehicles, with the Falcon 9 first-stage booster playing a pivotal role in reducing mission costs. A critical aspect of this reusability is the successful landing of the booster after launch, making it essential to understand the factors that influence landing outcomes. This project explores key questions such as what determines the success or failure of a Falcon 9 first-stage landing, and how variables like launch site, payload mass, and orbit type impact these outcomes. It also investigates whether a reliable machine learning model can be built to predict landing success and how interactive data visualizations can enhance insights into launch performance and mission optimization.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

The dataset for this project was collected through a combination of structured API calls and web scraping techniques. Using the SpaceX REST API, detailed information on each Falcon 9 launch was retrieved in JSON format, including attributes such as flight number, launch site, payload mass, and landing outcome. To enrich this dataset, HTML tables from Wikipedia containing mission descriptions, payload names, and launch dates were scraped using BeautifulSoup. After data retrieval, the two sources were merged using unique identifiers like flight numbers. The merged dataset was cleaned by handling missing values, resolving inconsistencies, and standardizing formats. Finally, feature engineering was performed to create meaningful variables like a binary 'Landing Success' outcome, producing a comprehensive dataset ready for exploratory analysis and predictive modeling.

Data Collection – SpaceX API

API Endpoint Access: Connected to SpaceX's public REST API using Python's requests library.

JSON Response Handling: Retrieved and parsed JSON data into Python dictionaries.

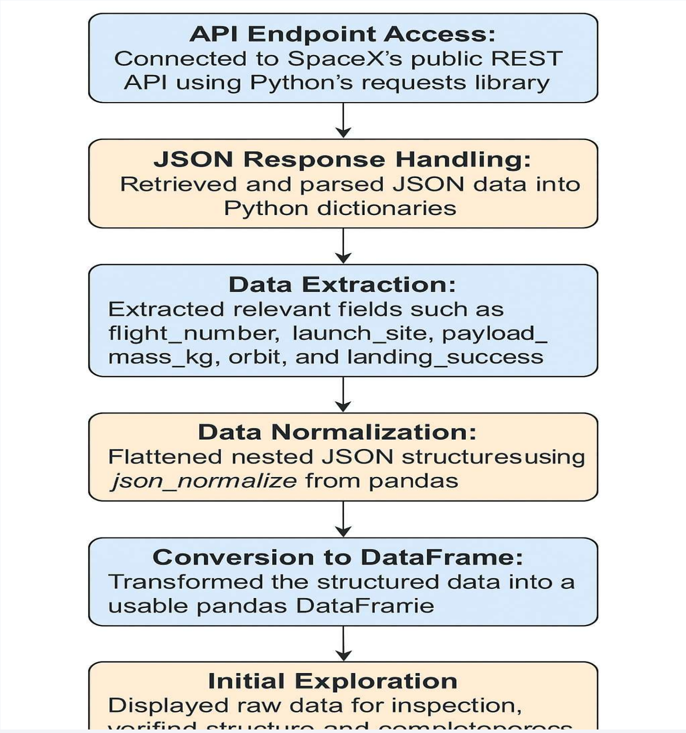
Data Extraction: Extracted relevant fields such as flight_number, launch_site, payload_mass_kg, orbit, and landing_success.

Data Normalization: Flattened nested JSON structures using json_normalize from pandas.

Conversion to DataFrame: Transformed the structured data into a usable pandas DataFrame.

Initial Exploration: Displayed raw data for inspection, verified structure and completeness.

Completed [SpaceX API calls notebook]



Data Collection - Scraping

Target Source: Wikipedia Falcon 9 launch records page

Library Used: BeautifulSoup for HTML parsing

Table Identification: Located HTML <table> elements containing mission and payload data

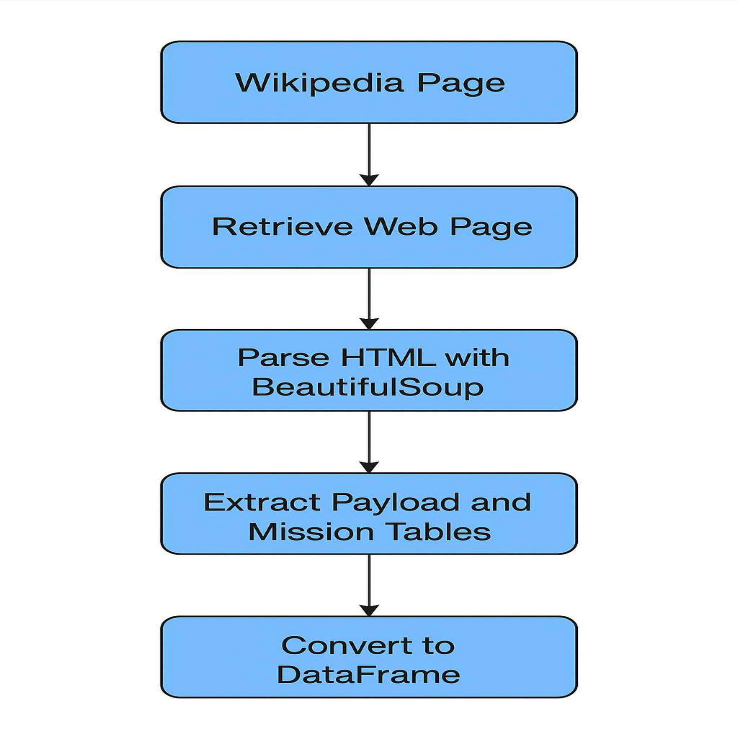
Data Extraction: Extracted launch dates, payload names, launch vehicles, and mission outcomes

Conversion to Structured Format: Parsed data stored in a pandas DataFrame

Cleaning & Formatting: Removed HTML tags, standardized column names, and handled missing entries

Export: Saved the structured dataset as a .csv for further integration

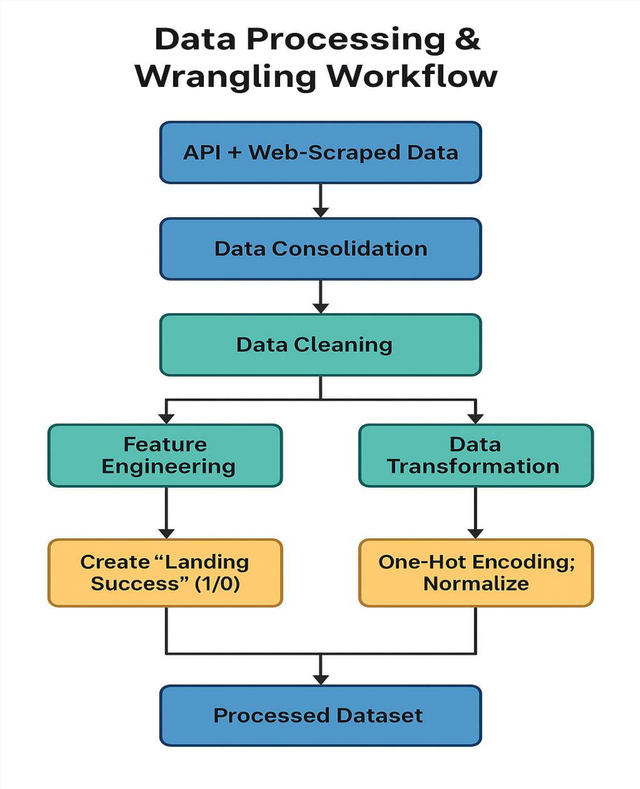
Completed [web scraping notebook]



Data Wrangling

- **Data Consolidation**: Merged API and web-scraped data into a unified DataFrame using pandas
- **Data Cleaning**: Handled missing values and duplicate records Standardized units (e.g., payload mass in kilograms)Normalized inconsistent formats (e.g., date-time strings)
- **Feature Engineering**:
  - *Created a binary classification target:*- Landing Success (1/0)
  - *Extracted fields like Booster Version, Launch Site, and OrbitData Transformation:*- Applied One-Hot Encoding for categorical variables
  - Normalized numerical features for machine learning stability
- **Exploratory Preview**: Displayed processed dataset with relevant features and labels

Completed [data wrangling related notebooks]



EDA with Data Visualization

1. Scatter Plots:

- Flight Number vs. Payload Mass: To examine the relationship between the sequence of flights and the payload mass, identifying any trends or anomalies.
- Flight Number vs. Orbit: To assess how the orbit type varies with the flight sequence, potentially revealing changes in mission objectives over time.
- Payload Mass vs. Orbit: To explore how different orbit types are associated with varying payload masses, which could influence landing success.

2. Line Plot:

- Flight Number vs. Launch Site: To visualize the distribution of launches across different sites over time, highlighting any shifts in launch site preferences.

Completed [EDA with data visualization notebook]

EDA with Data Visualization

3. Bar Charts:

- **Landing Success by Orbit Type:** To compare the success rates of landings across various orbit types, identifying which orbits are more conducive to successful landings.
- **Landing Success by Launch Site:** To determine which launch sites have higher success rates, potentially due to location-specific factors.

4. Pie Chart:

- **Overall Landing Success Rate:** To provide a clear overview of the proportion of successful versus unsuccessful landings, offering a quick assessment of overall performance.

Completed [EDA with data visualization notebook]

EDA with Data Visualization

5. Correlation Heatmap:

- To identify the strength and direction of relationships between numerical variables, aiding in feature selection for predictive modeling.
- These visualizations were instrumental in uncovering patterns, trends, and relationships within the data, guiding further analysis and model development.

Completed [EDA with data visualization notebook]

EDA with SQL

The analysis of SpaceX mission data provided several key insights

- The process of retrieving unique launch sites identified all distinct locations used by SpaceX, giving insight into the geographical distribution of their missions.
- Calculating the total payload for NASA (CRS) missions measured SpaceX's overall cargo delivery under NASA’s Commercial Resupply Services, showcasing its contribution to space logistics.
- The average payload calculation for the 'F9 v1.1' booster version provided insight into the efficiency and capacity of that specific rocket model.

Completed [EDA with SQL notebook]

EDA with SQL

- Counting successful and failed missions offered a clear metric of SpaceX's overall mission reliability across its launch history.
- Identifying failed drone ship landings helped pinpoint which boosters and launch sites were involved in unsuccessful sea recoveries, supporting improvements in landing technology.
- Analyzing failed drone ship landings in 2015 gave temporal context to recurring issues in that year, aiding targeted investigations.
- Ranking successful landings within a defined date range showed how SpaceX's landing success rate improved over time, reflecting progress in recovery systems.

Completed [EDA with SQL notebook]

Build an Interactive Map with Folium

Map Objects Added

Markers: were placed at each SpaceX launch site to indicate their exact locations. These markers included pop-up labels displaying the name of each launch site for easy identification. The purpose of the markers was to help users quickly identify launch locations and access site-specific information with a single click.

Circles: were drawn around each launch site to represent a specific radius, effectively highlighting the area of interest surrounding each site. These circles visually emphasized spatial coverage and the proximity of launch sites to surrounding features, aiding in geographic context analysis.

Lines (Polylines): were used to connect launch sites to nearby landmarks such as coastlines, rail lines, and roads. They illustrated the distances between launch sites and these infrastructural elements, supporting spatial analysis of logistical and operational connections critical to SpaceX's launch activities.

Completed [interactive map with Folium map]

Build a Dashboard with Plotly Dash

The interactive dashboard integrates multiple visual and control components, each added with a clear purpose to enhance user insight and engagement. The **launch site dropdown** allows users to filter the visualized data based on specific launch locations or view data across all sites, providing targeted analysis capability. The **success pie chart** visually conveys the proportion of successful launches, either overall or for a selected site, enabling quick comparison of success rates. The **payload range slider** lets users narrow the focus to a specific payload mass range, which directly influences the **success-payload scatter chart**, a graph designed to explore the relationship between payload size and mission outcome. This scatter plot is color-coded by booster version and filtered based on the selected launch site and payload range, offering layered insights into performance trends. These components were added primarily to allow **interactive filtering**, promoting dynamic exploration of the data; to improve **visual clarity**, by using intuitive visualizations for complex data relationships; and to boost **user engagement**, by enabling a hands-on, personalized analytical experience.

Completed Plotly Dash lab

Predictive Analysis (Classification)

Performed **data preparation** through feature selection, categorical encoding, and normalization of numerical features.

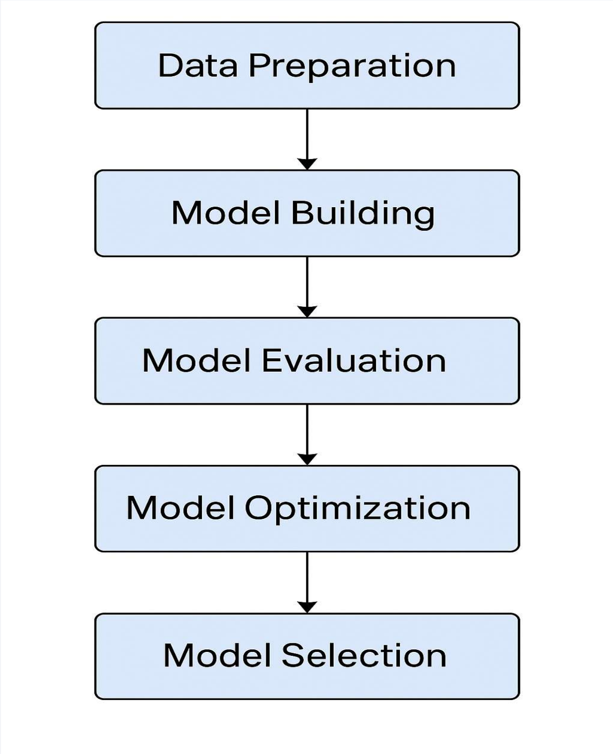
Build multiple machine learning algorithms: Logistic Regression, SVM, Decision Tree, KNN, and Random Forest.

Evaluated models using accuracy score, confusion matrix, classification report (precision, recall, F1-score), and k-fold cross-validation.

Applied GridSearchCV for hyperparameter tuning to **optimize** each model.

Selected the best-performing model based on evaluation metrics for final deployment.

Completed [predictive analysis lab]



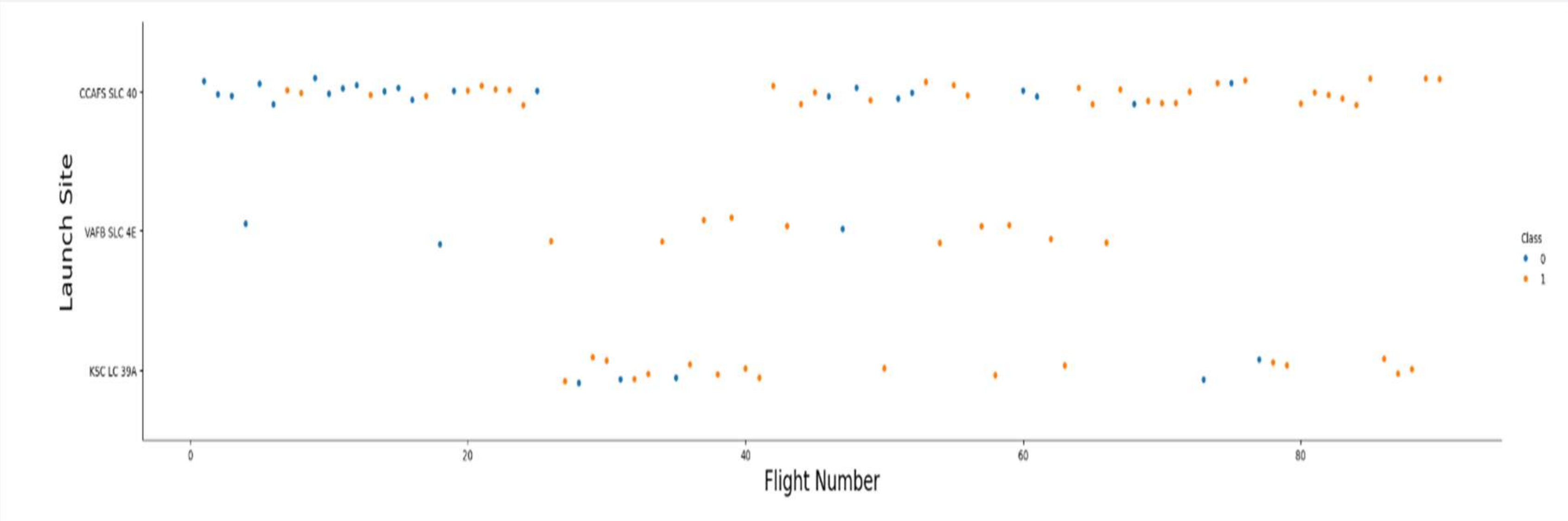
Results

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

Section 2

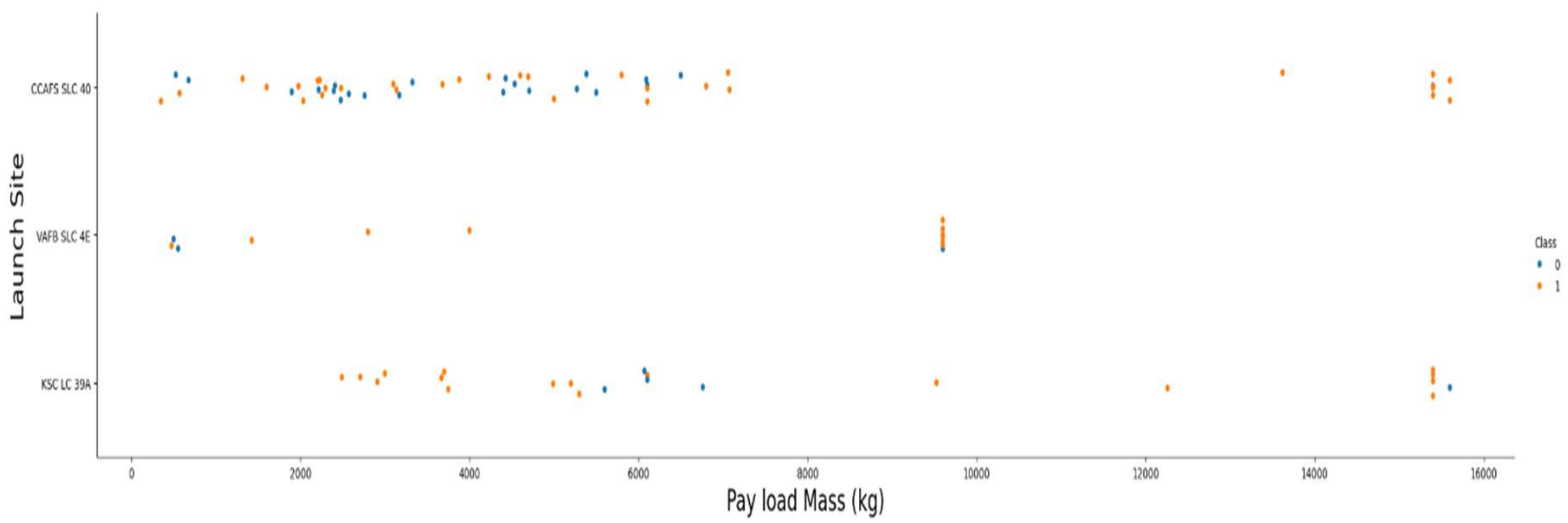
Insights drawn from EDA

Flight Number vs. Launch Site



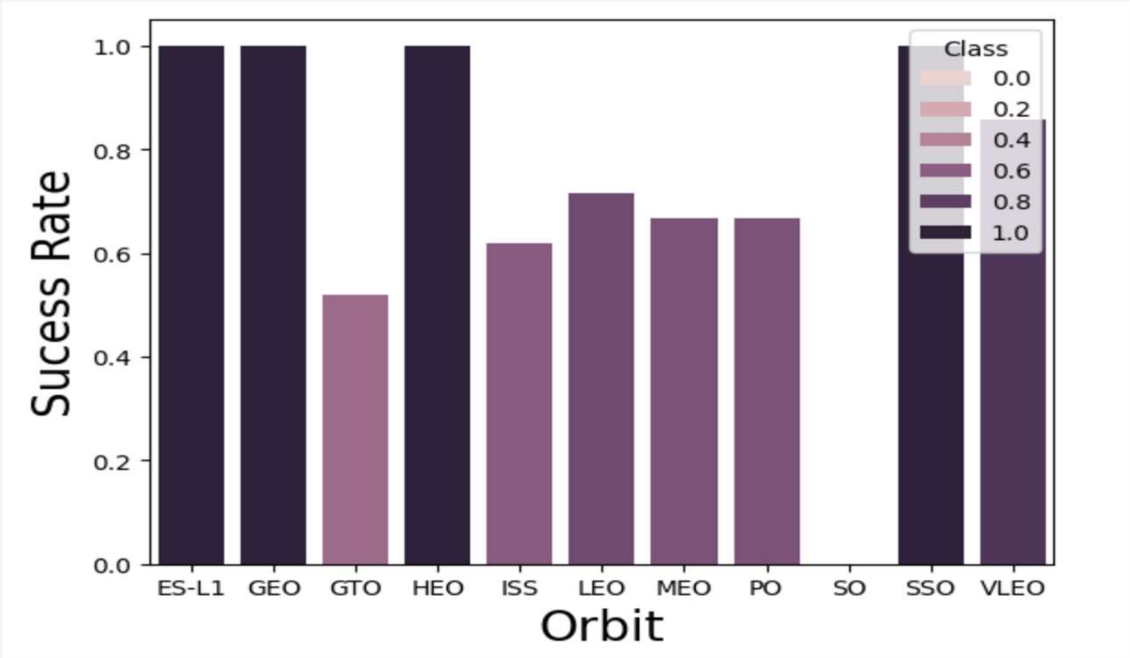
- CCAFS SLC 40 is the dominant site but had more failures early on.
- KSC LC 39A and VAFB SLC 4E appear to have fewer launches but relatively higher success rates in later flights.

Payload vs. Launch Site



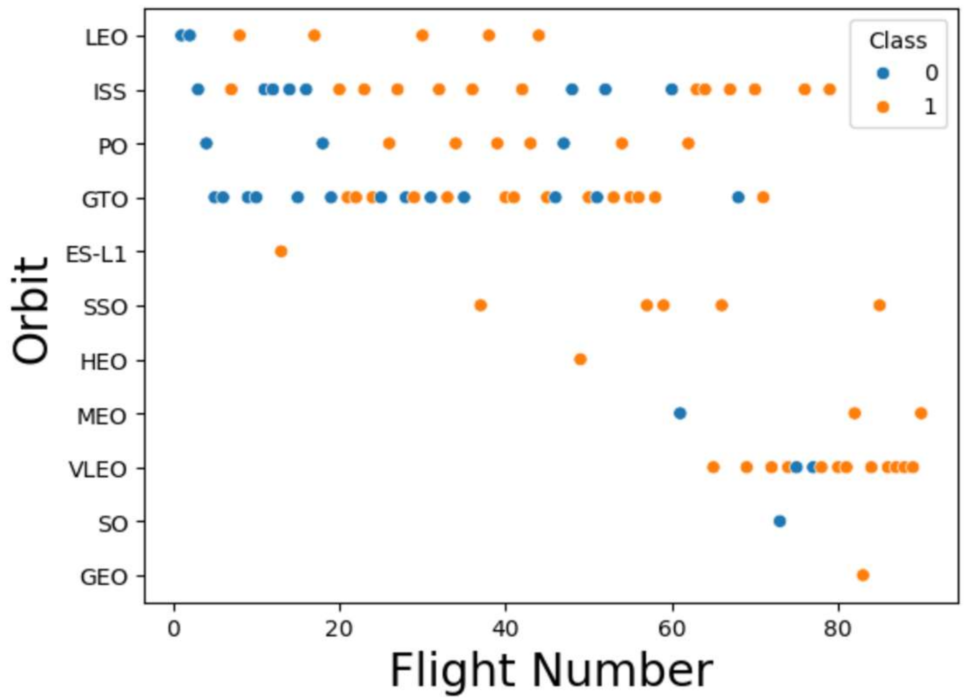
Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



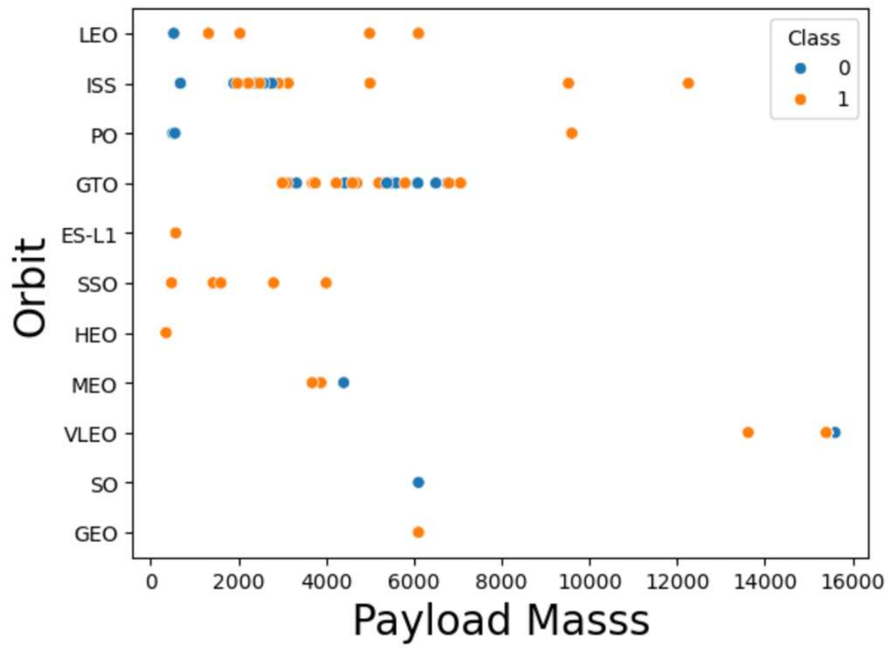
- Safest orbits (historically): ES-L1, GEO, HEO, SO/SSO.
- Moderate-risk orbits: LEO, MEO, PO, ISS.
- Highest-risk orbit: GTO.

Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

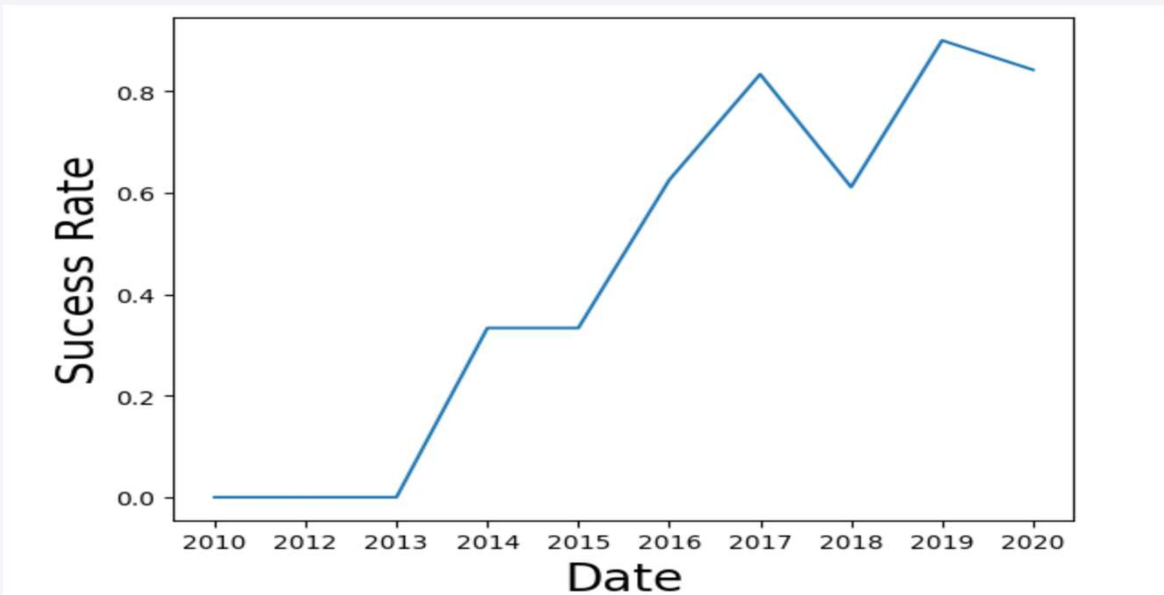
Payload vs. Orbit Type



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



SpaceX's journey from unreliable early attempts to near-perfect launch success reflects strong innovation and learning.

- The major boost in 2016–2019 likely aligns with:
- Reusable rocket tech (Falcon 9 Block 5).
- More frequent launches.
- Stronger mission control and engineering refinements. Despite minor dips (like in 2018), the long-term trajectory is upward.

All Launch Site Names

- These sites are key hubs for SpaceX missions, including commercial, military, and crewed launches, giving insight into the geographical distribution of their missions.

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

Launch Site Names Begin with 'CCA'

- All missions were successful — showing strong early performance in orbital missions.
- Landing attempts were mostly not made except for early parachute recovery tests (both failed).
- Payload mass increased over time, indicating maturing launch capability.
- All launches were from CCAFS LC-40, showing reliance on this primary launch site in the early days.

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

The total payload mass carried by boosters launched by NASA (CRS). The **total payload mass carried by boosters launched by NASA (CRS)** is an important metric that reflects the cumulative weight of cargo delivered under the **Commercial Resupply Services (CRS)** missions to the International Space Station (ISS).

total_payload_mass_kg_
45596

Average Payload Mass by F9 v1.1

This average indicates that Falcon 9 v1.1 was typically used for **medium-weight missions**, likely including ISS resupply (e.g., NASA CRS), satellite deployments, or demo flights.

avg_payload_mass_kg_
2534.6666666666665

First Successful Ground Landing Date

This historic achievement marked the first time an orbital-class rocket's first stage returned to Earth and landed vertically on land after delivering a payload to orbit.

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

These missions demonstrate SpaceX's capability to recover boosters via drone ship landings while delivering medium-weight payloads to orbit.

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

Total Number of Successful and Failure Mission Outcomes

- Falcon 9's high success rate, of 99.01%, underscores SpaceX's leadership in commercial spaceflight, with 100 successful missions and a proven track record of booster reusability.

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

Boosters Carried Maximum Payload

- Most boosters are Falcon 9 Block 5 (F9 B5) versions.
- Booster B1051 appears 3 times (versions .3, .4, .6) — indicating strong reuse or consistent high payload performance.
- Boosters B1049 and B1060 appear 3 and 2 times respectively — also notable.

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

Reusability is central to SpaceX's goal of **cutting launch costs**. Each failed landing meant the **loss of a multi-million-dollar booster**.

These failures led to:

- Design improvements in grid fins and landing leg stabilization.
- Better control algorithms and more onboard sensors.
- Refinements in drone ship engineering (like deck dampening systems).

These “controlled failures” were much cheaper than full mission failures and helped pave the way for eventual success.

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

The ranking of landing outcomes reflects SpaceX's progressive improvement in rocket recovery, with a shift towards more successful drone ship landings to accommodate larger payloads, while failures highlight the continuous refinement of their landing technologies.

Landing_Outcome	no_times	RANK() OVER(ORDER BY no_times DESC)
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

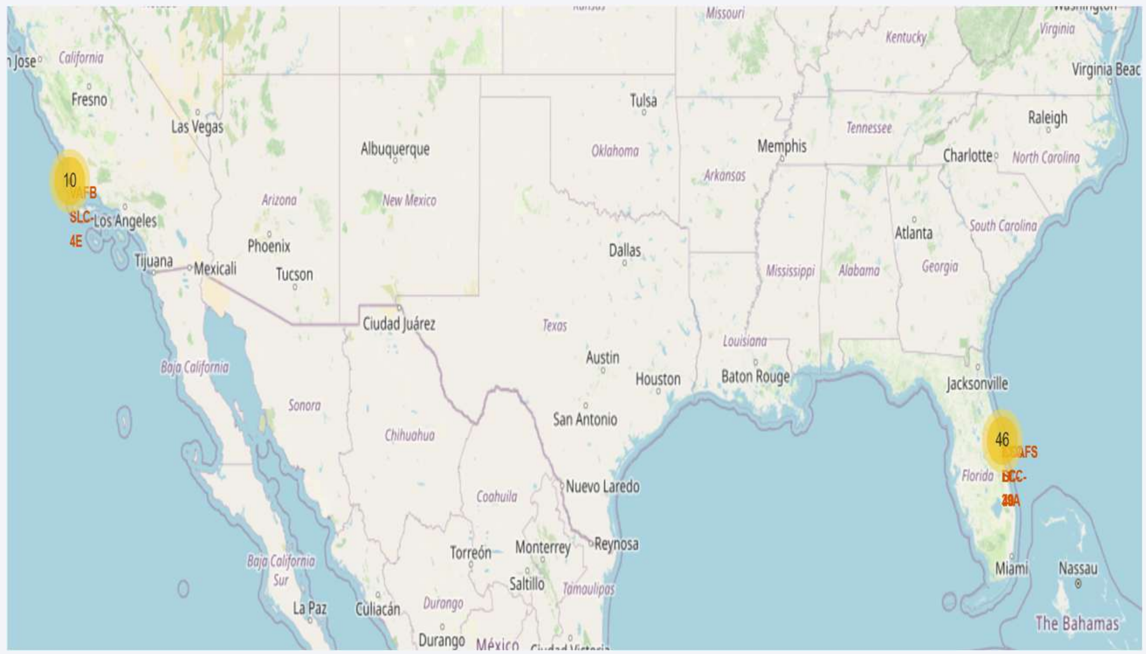
Section 3

Launch Sites Proximities Analysis

GeoPostion of Lauch Sites

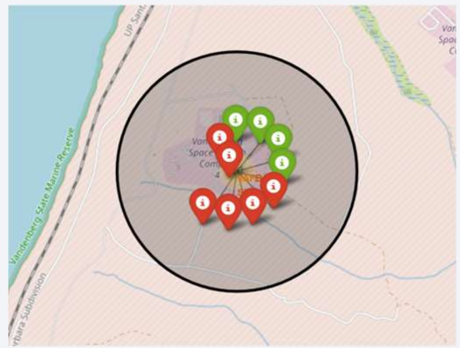
- The position of a launch site near the **equator** provides a **velocity boost** from Earth's rotation, making launches more energy-efficient for certain orbits like geostationary orbits.

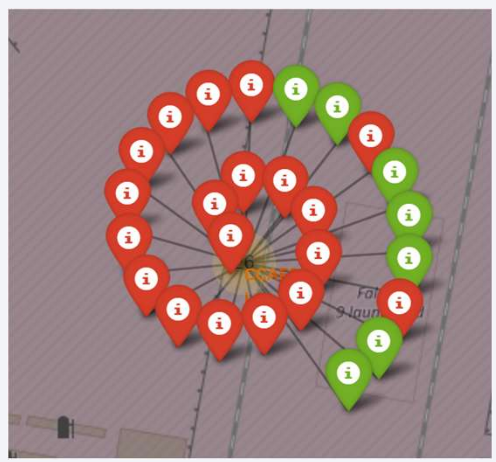
- Launch sites in **remote or coastal areas** ensure **safety** by minimizing the risk to populated zones and providing clear flight paths for rockets.

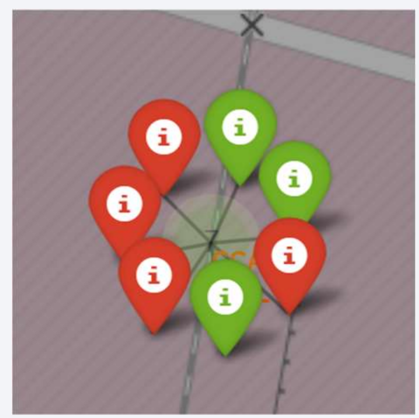


Lauch Outcome GeoRepresntation

- Launch outcomes at various sites help assess the reliability and safety of each location, influencing mission planning and site selection for future launches.
- Sites with higher success rates gain strategic and commercial value, attracting more clients, partnerships, and investments in space infrastructure.

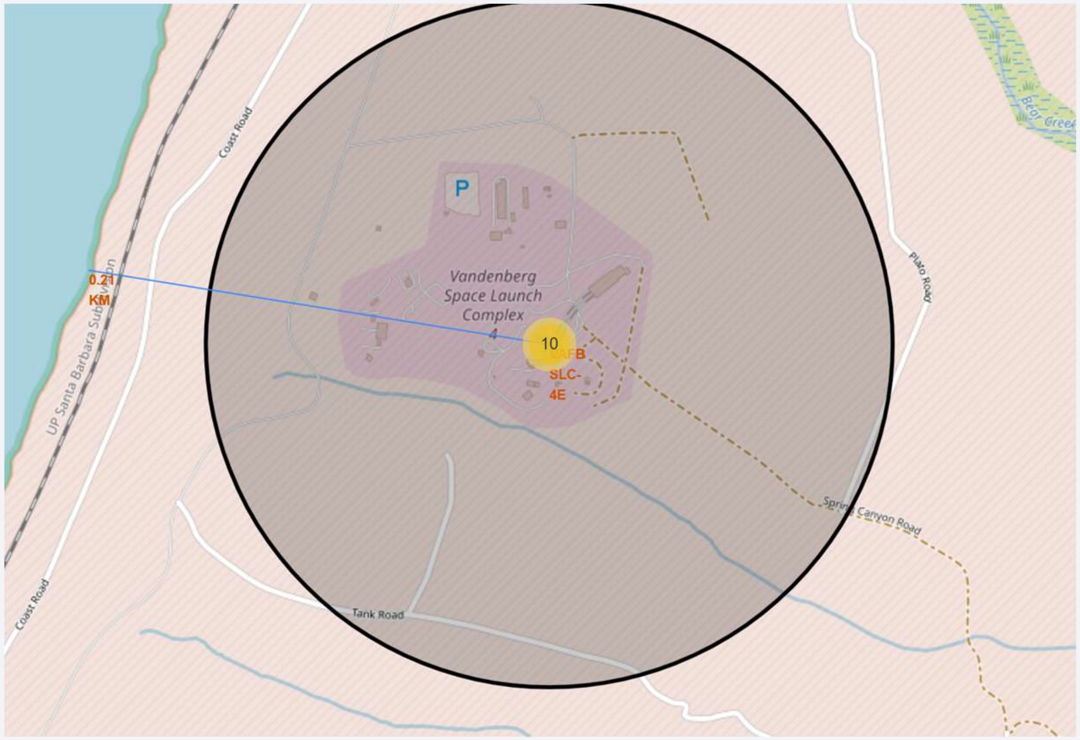






Optimal Launch Site Position

A launch site's close proximity to railways, highways, and coastlines significantly enhances its operational efficiency, mission safety, and launch flexibility, making it a critical factor in both national and commercial space program planning.



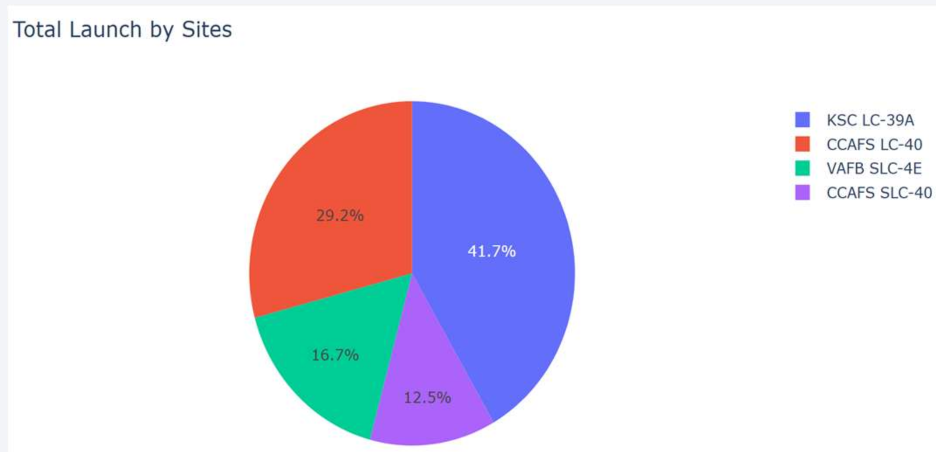
Section 4

Build a Dashboard with Plotly Dash

Total Launch at each Site

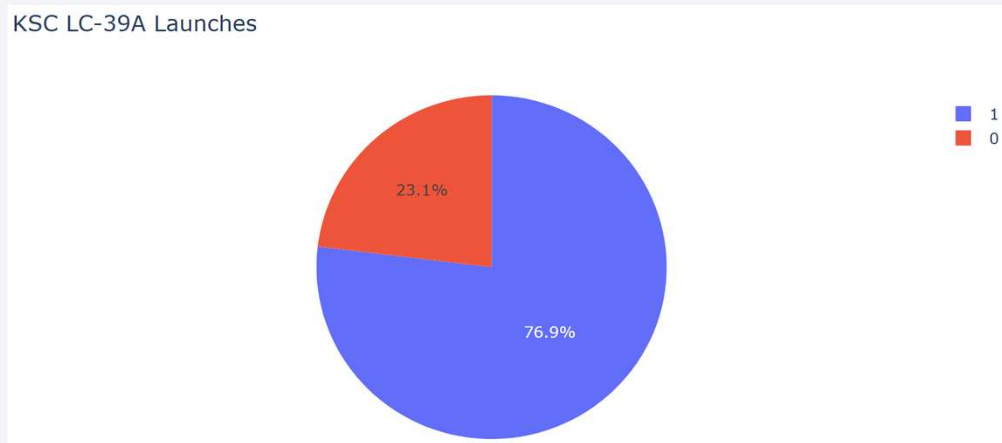
KSC LC-39A had the highest number of launches, followed by CCAFS LC-40, VAFB SLC-4E, and finally the second CCAFS SLC-40 entry.

KSC LC-39A had the highest number of launches likely due to the high success rate in recovery of the first stage



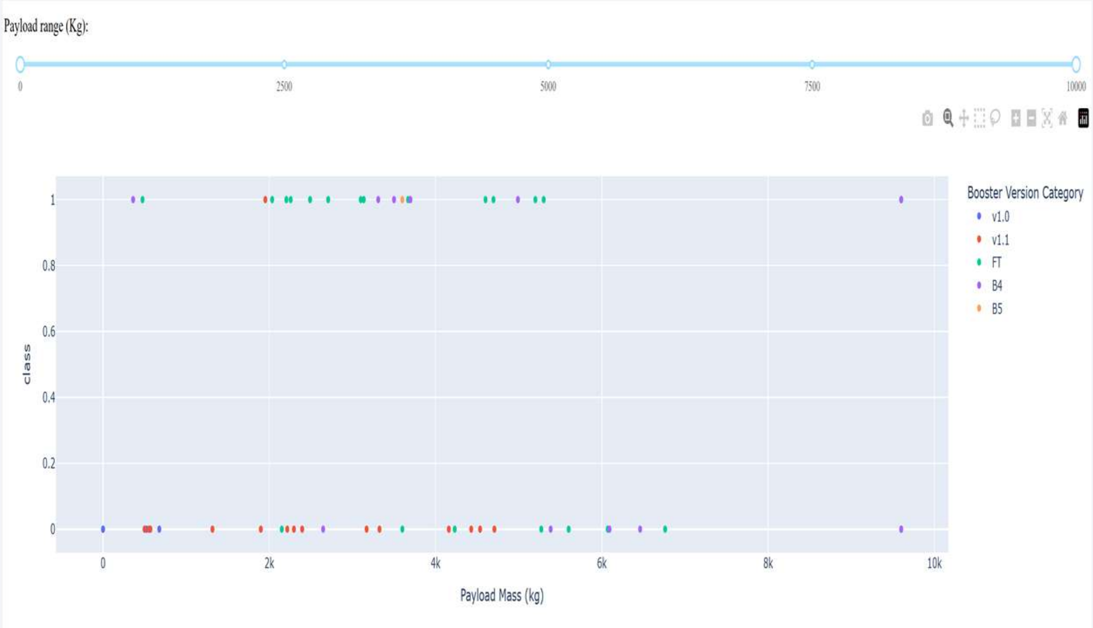
Best Performing Site

KSC LC-39A (Kennedy Space Center Launch Complex 39A) has a Strong Success Rate, With nearly 77% of launches successful. Most launches from this site are succeeding.
But there is room for improvement, a 23.1% failure rate is quite significant in aerospace. This could highlight technical, operational, or logistical challenges at this site during the timeframe studied.



Booster Version Success Rate

- Launch success is not solely dependent on payload mass.
- Newer booster versions (like FT, B4, and B5) show higher success rates compared to older versions (v1.0 and v1.1).
- Heavier payloads tend to have a slightly higher chance of failure, especially when using older booster versions.
- The data suggests that booster design improvements over time have enhanced the reliability of launches.

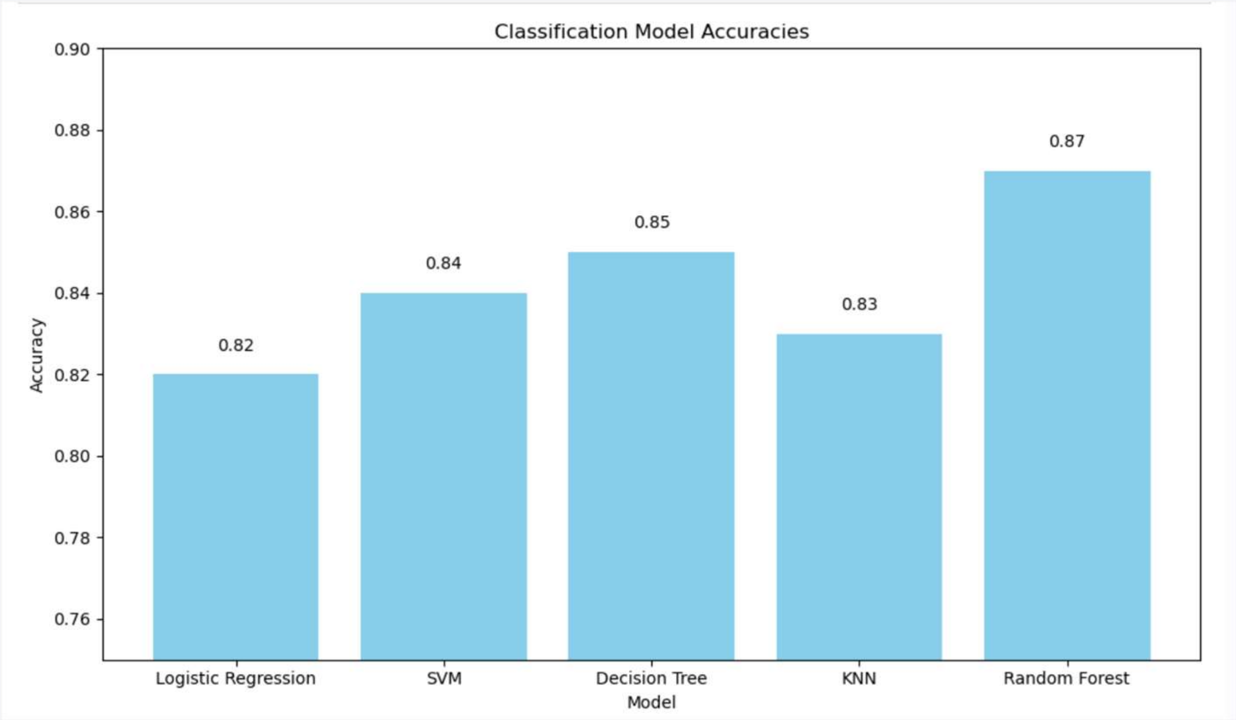


Section 5

Predictive Analysis (Classification)

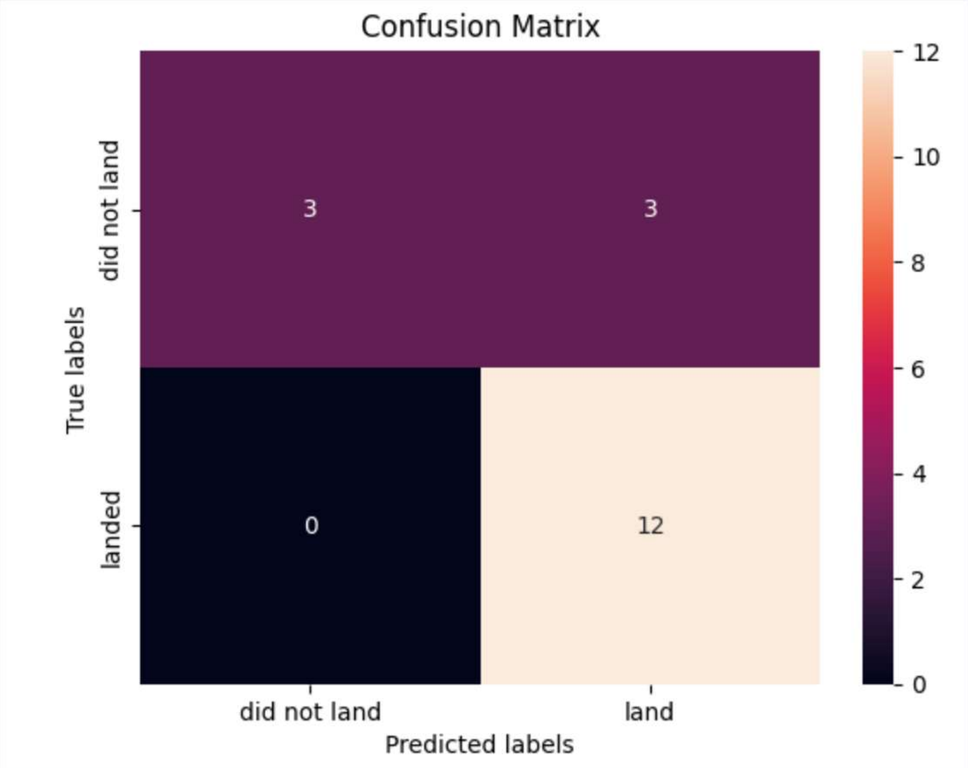
Classification Accuracy

- Random Forest Has the highest Accuracy upon feature optimization and hyper-parameter selection. Showing Random forest as the best classification model



Confusion Matrix

Confusion matrix of the test data show how well the model performed in determining if the rocket landed. With only 3 cases being mis classified and instead of successfully landed



Conclusions

- SpaceX should prioritize the use of Block 5 boosters, as they have demonstrated the highest launch and landing success rates in the dataset.
- Efforts should be made to optimize landing systems and thrust management for missions carrying heavier payloads, as landing success slightly decreases with increased mass.
- Launches should be scheduled from KSC LC-39A whenever possible, since this site has shown the highest rate of successful landings due to favorable infrastructure and conditions.
- Machine learning models like Random Forest should be integrated into mission planning workflows to improve the prediction and reliability of landing outcomes.

Appendix

- The full python code, SQL queries, charts, Notebook outputs, or data sets created during this project are available on github: https://github.com/owoeye/SpaceX-Analysis/tree/main

Appendix

TASK 4

Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters` .

In [13]:

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

In [42]:

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

# Grid search with cross-validation
logreg_cv = GridSearchCV(estimator=lr,
                            param_grid=parameters,
                            cv=10,            # 5-fold cross-validation
                            scoring='accuracy',
                            n_jobs=-1)        # Use all CPU cores

logreg_cv.fit(X_train, Y_train)
```

Out[42]:

```
GridSearchCV(cv=10, estimator=LogisticRegression(), n_jobs=-1,
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']},
             scoring='accuracy')
```

Appendix

- The full python code, SQL queries, charts, Notebook outputs, or data sets created during this project are available on github: https://github.com/owoeye/SpaceX-Analysis/tree/main

Appendix

```python
# Import required libraries
import pandas as pd
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output
import plotly.express as px

# Read the airline data into pandas dataframe
spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()
# print(spacex_df)

# Create a dash application
app = dash.Dash(__name__)

# TASK 2:
# Add a callback function for `site-dropdown` as input, `success-pie-chart` as output
# Function decorator to specify function input and output
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'))
def get_pie_chart(entered_site):
    filtered_df = spacex_df
    if entered_site == 'ALL':
        fig = px.pie(filtered_df, values='class',
        names='Launch Site',
        title='Total Launch by Sites')
        return fig
    else:
        # return the outcomes pie-chart for a selected site
        site_df = spacex_df[spacex_df["Launch Site"] == entered_site]
        fig = px.pie(site_df,
                     names='class',
```

Appendix

```
In [3]: def date_time(table_cells):
            """
            This function returns the data and time from the HTML  table cell
            Input: the  element of a table data cell extracts extra row
            """
            return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

        def booster_version(table_cells):
            """
            This function returns the booster version from the HTML  table cell
            Input: the  element of a table data cell extracts extra row
            """
            out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if
            return out

        def landing_status(table_cells):
            """
            This function returns the landing status from the HTML table cell
            Input: the  element of a table data cell extracts extra row
            """
            out=[i for i in table_cells.strings][0]
            return out


        def get_mass(table_cells):
            mass=unicodedata.normalize("NFKD", table_cells.text).strip()
            if mass:
                mass.find("kg")
                new_mass=mass[0:mass.find("kg")+2]
            else:
                new_mass=0
            return new_mass


        def extract_column_from_header(row):
            """
            This function returns the landing status from the HTML table cell
            Input: the  element of a table data cell extracts extra row
            """
```

Thank you!