IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Allan Santos

07/29/2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API and Web Scraping of Wikipedia data
- Perform data wrangling
  - Data cleaning, handling of missing values, and dataset enrichment using the Pandas library
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Prepared the data to train and evaluate classification models to predict landing success

Data Collection

1. SpaceX REST API

- Fetched historical launch data.
- Enriched with details on boosters, payloads, and launch sites.

2. Web Scraping

- Extracted landing outcomes from Wikipedia.
- Used this data to create our target variable for success/failure.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- https://github.com/alllanvfs/Space-X-Falcon-9-First-Stage-Landing-Prediction/tree/main

```python
#Take the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
  for x in data['rocket']:
    if x:
      response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
      BoosterVersion.append(response['name'])
```

```python
#Take the dataset and uses the launchpad column to call the API and append the data to the list
def getLauchSite(data):
  for x in data['launchpad']:
    if x:
      response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
      Longitude.append(response['longitude'])
      Latitude.append(response['latitude'])
      LaunchSite.append(response['name'])
```

```python
#Take the dataset and uses the payloads column to call the API and append the data to the list
def getPayloadData(data):
  for load in data['payloads']:
    if load:
      response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
      PayloadMass.append(response['mass_kg'])
      Orbit.append(response['orbit'])
```

```python
#Take the dataset and uses the cores column to call the API and append the data to the list
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/alllanvfs/Space-X-Falcon-9-First-Stage-Landing-Prediction/tree/main

```
#Return the data and time
def date_time(table_cells):
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

#Return the booster version
def booster_version(table_cells):
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if i%2==0][0:-1])
    return out

#Return the landing status
def landing_status(table_cells):
    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

#Return the landing status
def extract_column_from_header(row):

    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    column_name = ' '.join(row.contents)

    # Filter the digit and empty names
    if not(column_name.strip().isdigit()):
        column_name = column_name.strip()
        return column_name
```

```
[ ] static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
[ ] #request to fetch the page content
```

Data Wrangling

- Success Rate Improves Over Time: Visualizations show a clear positive trend between the flight number and landing success, indicating that SpaceX improved its technology and procedures over time.

- Launch Site and Orbit are Key Factors: The analysis revealed that the launch site and the target orbit are strong predictors of the landing outcome. Certain orbits, like GTO, have historically lower success rates than orbits like LEO.

- Payload Mass Correlation: There is a visible relationship between payload mass and landing success. Scatter plots help identify how success rates vary for different payload weights.

EDA with Data Visualization

- Quickly Answered Key Questions: SQL queries allowed us to rapidly answer specific business questions, such as identifying unique launch sites and calculating the total payload mass for NASA missions.

- Filtered for Specific Scenarios: We used SQL to filter for very specific mission profiles, like finding the booster versions used in successful drone ship landings with a payload between 4000kg and 6000kg.

- Identified Important Milestones: We were able to pinpoint the exact date of the first successful ground pad landing, a significant event in SpaceX's history.

- Aggregated Mission Outcomes: SQL was effective for counting the total number of successful vs. failed missions and ranking landing outcomes by frequency within specific date ranges.

EDA with SQL

- We prepared the data to build a classification model to predict landing success (the Class variable).
- Several models were set up for comparison: Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
- GridSearchCV was used to tune the parameters for each model to find the best possible performance.
- The final goal is to evaluate each model's accuracy and select the best one for predicting future Falcon 9 landing outcomes.

Build an Interactive Map with Folium

Landing Success is Predictable: Our analysis confirmed that landing success is not random. It is strongly correlated with mission parameters like the target orbit, launch site, and payload mass.

Data-Driven Insights: We successfully built an end-to-end pipeline to collect, clean, and analyze complex launch data, proving that data science can uncover key factors in aerospace engineering.

A Predictive Tool was Developed: The project resulted in a framework for a machine learning model capable of predicting landing outcomes, offering a valuable tool for mission planning.

Commercial Value: This predictive capability has direct business applications, allowing for better risk assessment and contributing to SpaceX's goal of reducing the cost of space access.

Build a Dashboard with Plotly Dash

Success Rate Pie Chart: Added a pie chart to show the percentage of successful landings for each launch site, offering a quick comparison of site performance.

Payload vs. Outcome Scatter Plot: Included a scatter plot to visualize the relationship between payload mass and landing success, helping to identify performance trends.

Interactive Dropdown Filter: The main interaction is a dropdown menu to filter all plots by a specific launch site. This allows users to drill down and analyze site-specific data dynamically.

Predictive Analysis (Classification)

- Prepared the Data: Converted categorical features to numbers using one-hot encoding and standardized all features.
- Split Data: Divided the dataset into training and testing sets for evaluation.
- Trained & Tuned: Built four models (Logistic Regression, SVM, Decision Tree, KNN) and used GridSearchCV to find the best parameters for each.
- Selected Best Model: Compared the tuned models' accuracy to identify the top performer.

Results

- **Exploratory Data Analysis**

Key Finding: Landing success is strongly linked to the launch site, orbit, and payload mass. Success rates also improved with each flight.

- **Interactive Analytics**

Dashboard Demo: An interactive dashboard was built with a map and charts. A dropdown menu allows filtering all data by launch site for dynamic analysis.

- **Predictive Analysis**

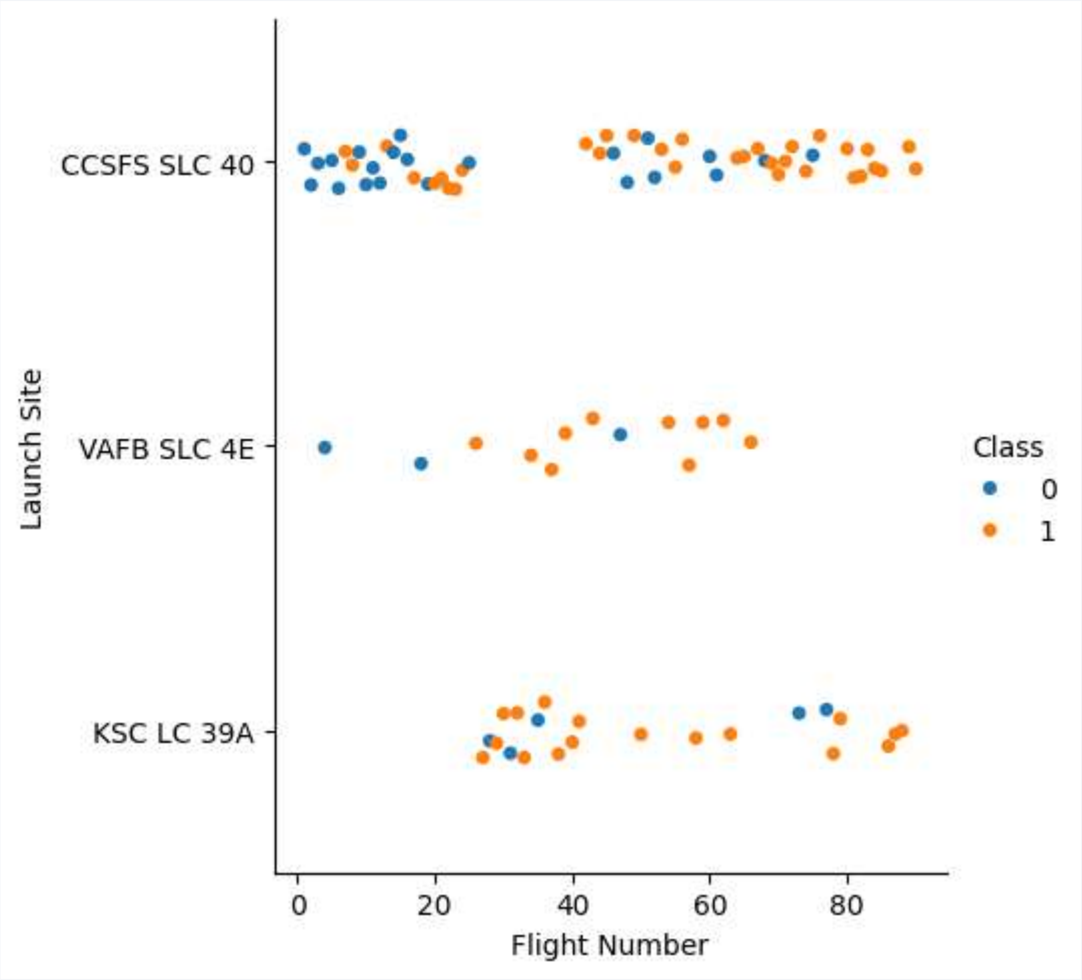
Best Model: After training and tuning four models, Logistic Regression achieved the highest accuracy, making it the most reliable predictor.

Section 2

Insights drawn from EDA

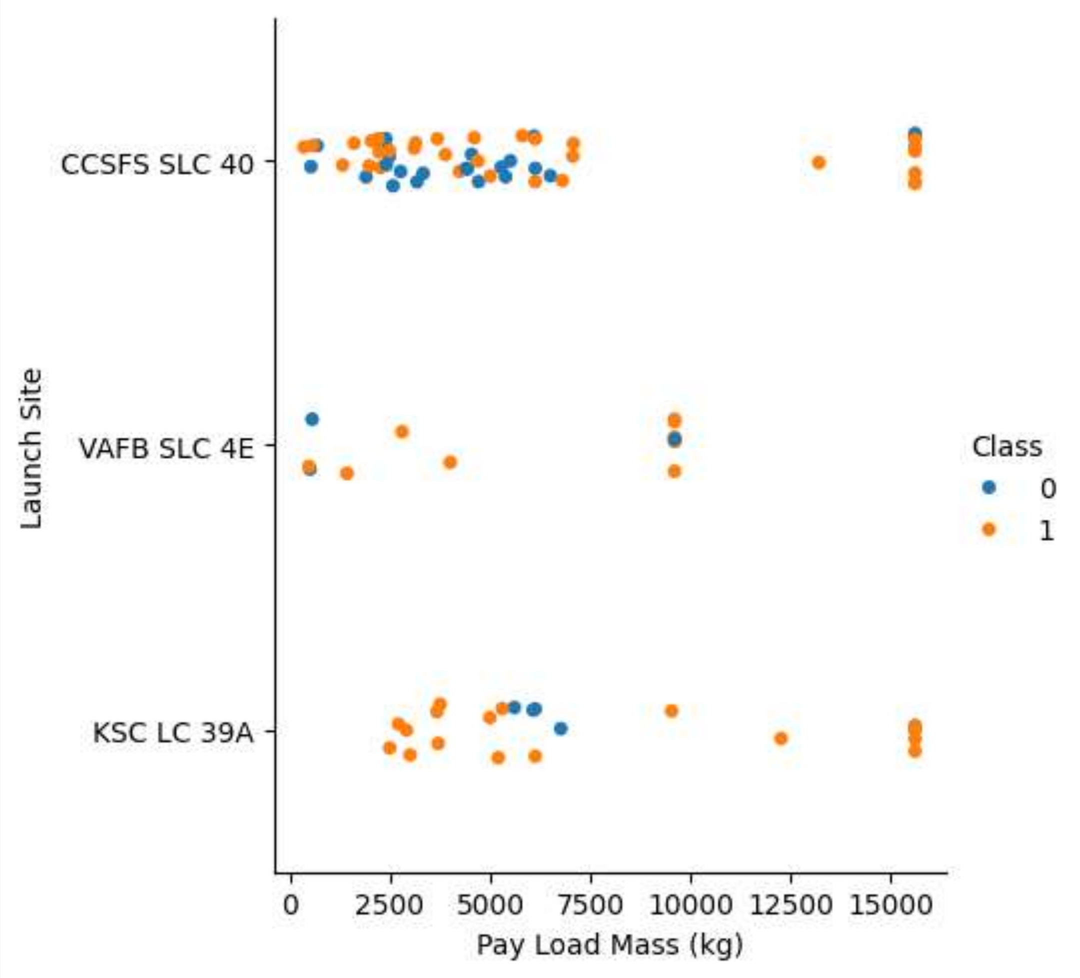
Flight Number vs. Launch Site

This chart shows that launches with heavier payloads tend to be more successful, a trend most noticeable at the KSC LC-39A and CCAFS SLC-40 sites.



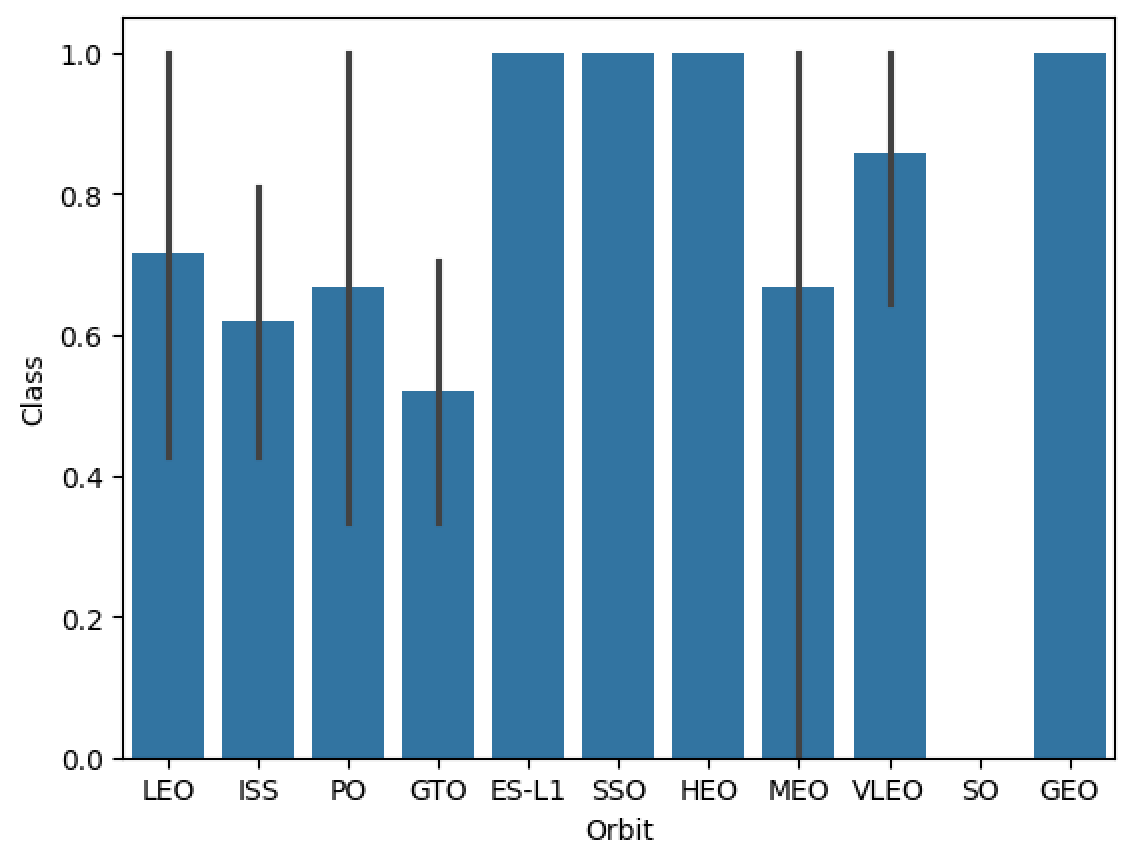
Payload vs. Launch Site

This plot tracks launch outcomes by flight number for each launch site.
A clear trend emerges: later flights (higher flight numbers) are almost entirely successful across all sites, demonstrating a significant improvement in launch reliability over time.



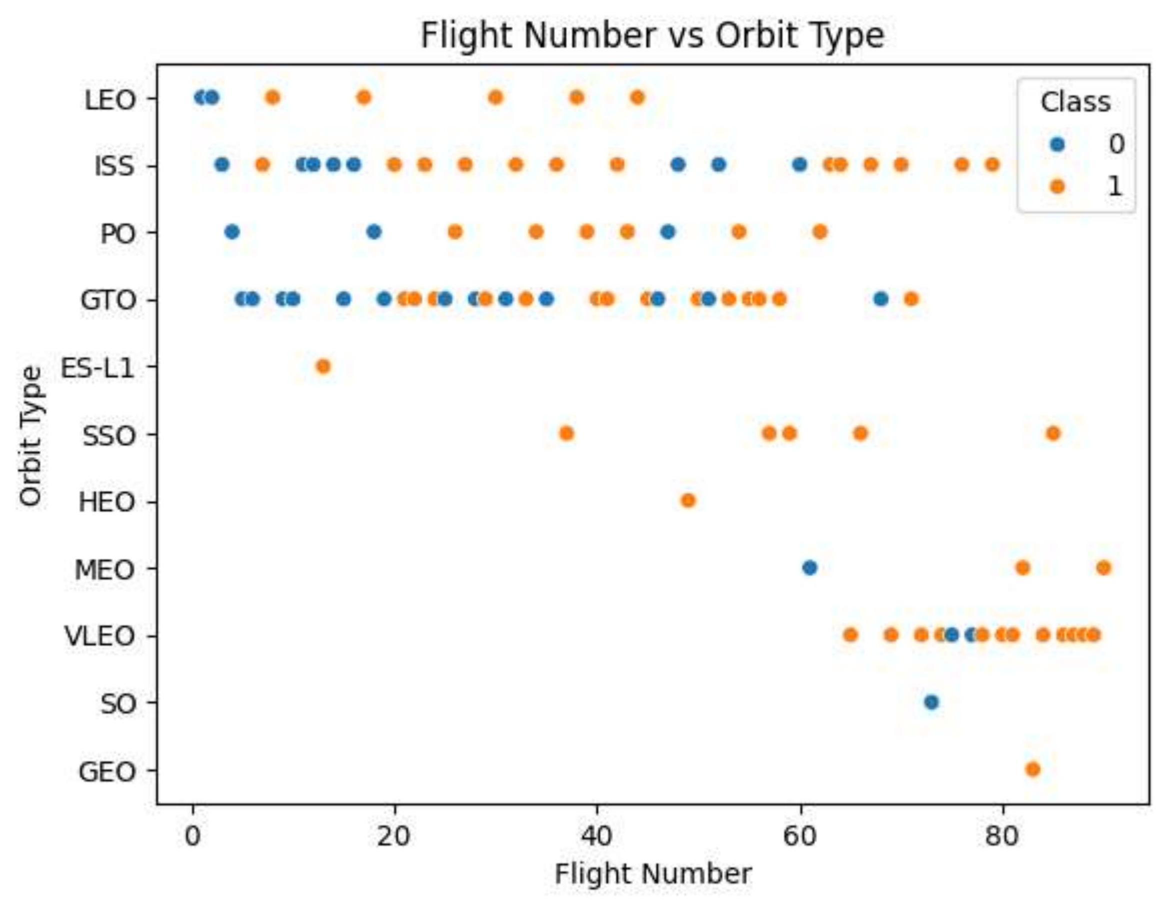
Success Rate vs. Orbit Type

This chart reveals that the target orbit is a key factor in mission success. Certain orbits like GEO, HEO, and ES-L1 show a 100% success rate in your data. In contrast, a more demanding orbit like GTO has a visibly lower success rate, indicating the different levels of challenge each orbit presents.



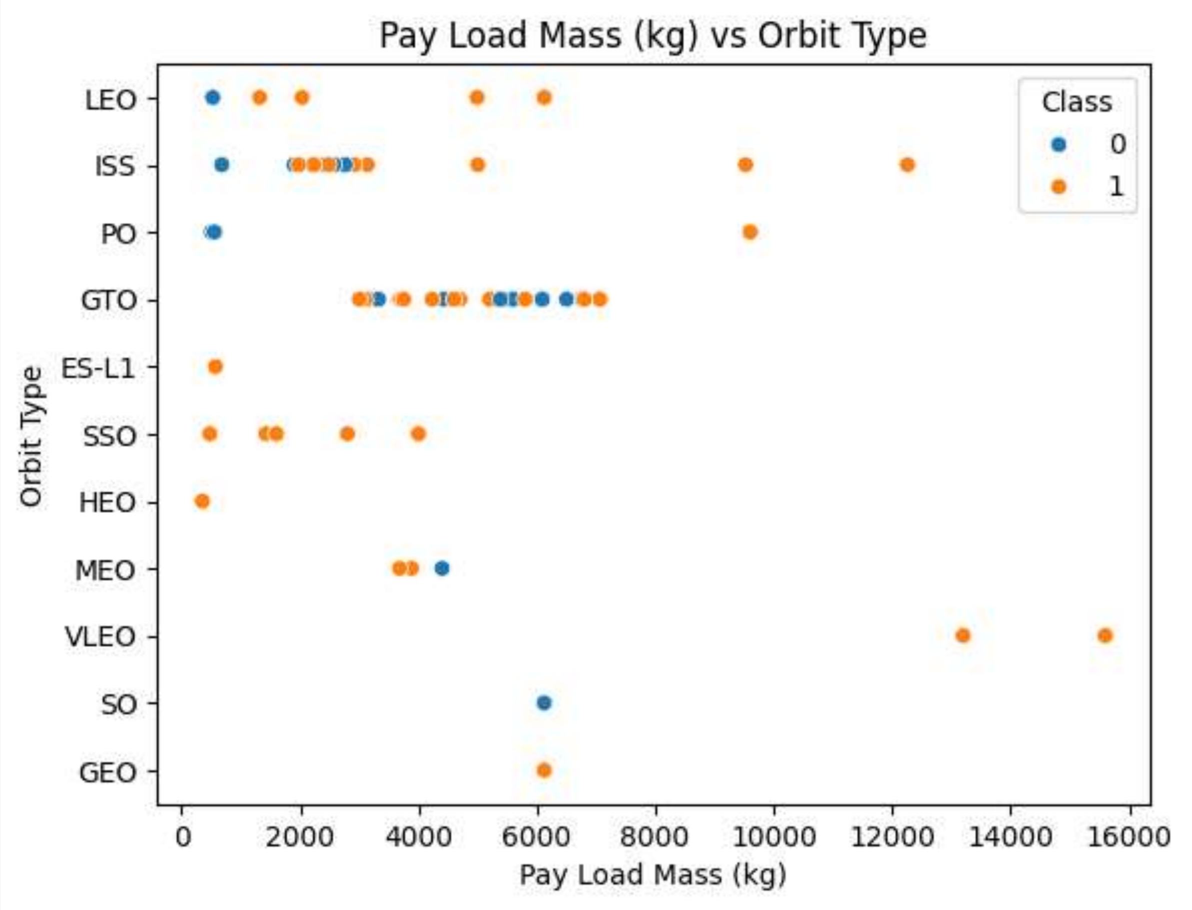
Flight Number vs. Orbit Type

This plot shows flight success (orange) versus failure (blue) for different orbit types. It clearly reveals that missions became more successful over time as the flight number increased.



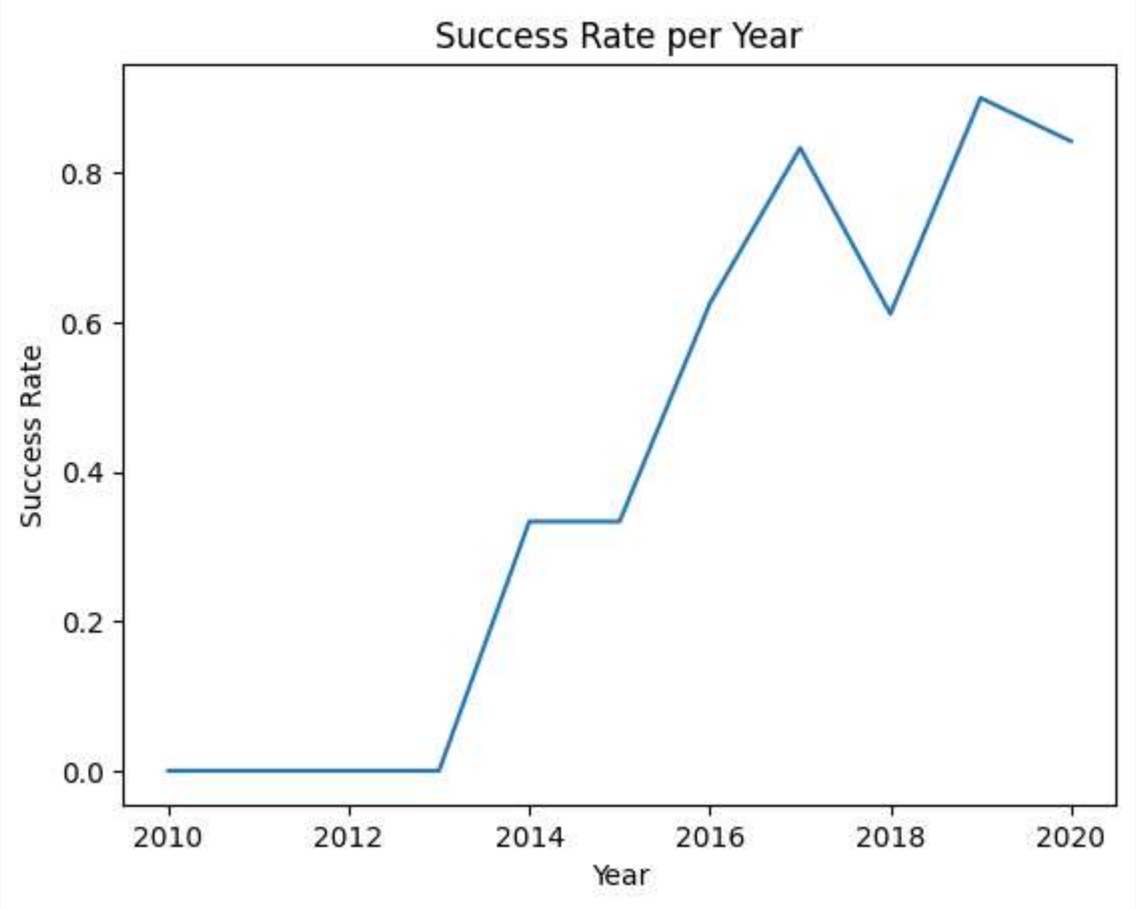
Payload vs. Orbit Type

This graph plots mission outcomes based on payload mass and orbit type, using orange for success and blue for failure. It shows that different orbits accommodate distinct payload masses and have varying success rates, with no clear failure trend based on mass alone.



Launch Success Yearly Trend

This line chart illustrates the mission success rate annually from 2010 to 2020. It reveals a dramatic improvement over time, starting with a 0% success rate until 2013. After 2013, the rate climbed significantly, peaking in 2019, indicating a strong positive trend for the decade.



All Launch Site Names

These acronyms represent specific launch pads at major spaceports. For example, CCAFS is Cape Canaveral, VAFB is Vandenberg Air Force Base, and KSC is Kennedy Space Center.

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

Launch Site Names Begin with 'CCA'

Each row represents a single flight, with columns for date, booster version, payload details, and customer.

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 command SUM("PAYLOAD_MASS__KG_") calculates the total value of a "Payload Mass" column. The resulting total mass is 45,596 kilograms.

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

The command AVG("PAYLOAD_MASS__KG_") calculated the average of all values in the payload mass column. The resulting average payload mass is approximately 2534.67 kg.

```
 * sqlite:///my_data1.db
Done.
AVG("PAYLOAD_MASS__KG_")
2534.6666666666665
```

First Successful Ground Landing Date

The MIN(Date) command was used to find the earliest date recorded in the data. The result shows the earliest date in the dataset is December 22, 2015.

```
 * sqlite:///my_data1.db
Done.
 MIN(Date)
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

These names, like F9 FT B1022, are unique identifiers for SpaceX Falcon 9 rocket boosters. The .2 suffix, as in B1021.2, typically indicates that this was the booster's second flight.

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

It lists each distinct outcome, like "Success (drone ship)" or "No attempt," and shows how many times each one happened. The most common result in this dataset was "No attempt," which occurred 10 times.

Done.

Landing_Outcome	Outcome_Count
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

Boosters Carried Maximum Payload

These names identify specific SpaceX Falcon 9 Block 5 boosters. The numbering, like B1048.4, indicates the booster's serial number (1048) and its flight number (the 4th flight).

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

Each row details the month, landing outcome, booster version, and launch site for a mission. Specifically, it lists two failed drone ship landings from the CCAFS LC-40 site.

Done.

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

It lists each distinct result, like "Success (drone ship)" or "No attempt," and shows how many times each one occurred. The most frequent outcome in this dataset was "No attempt," which occurred 10 times.

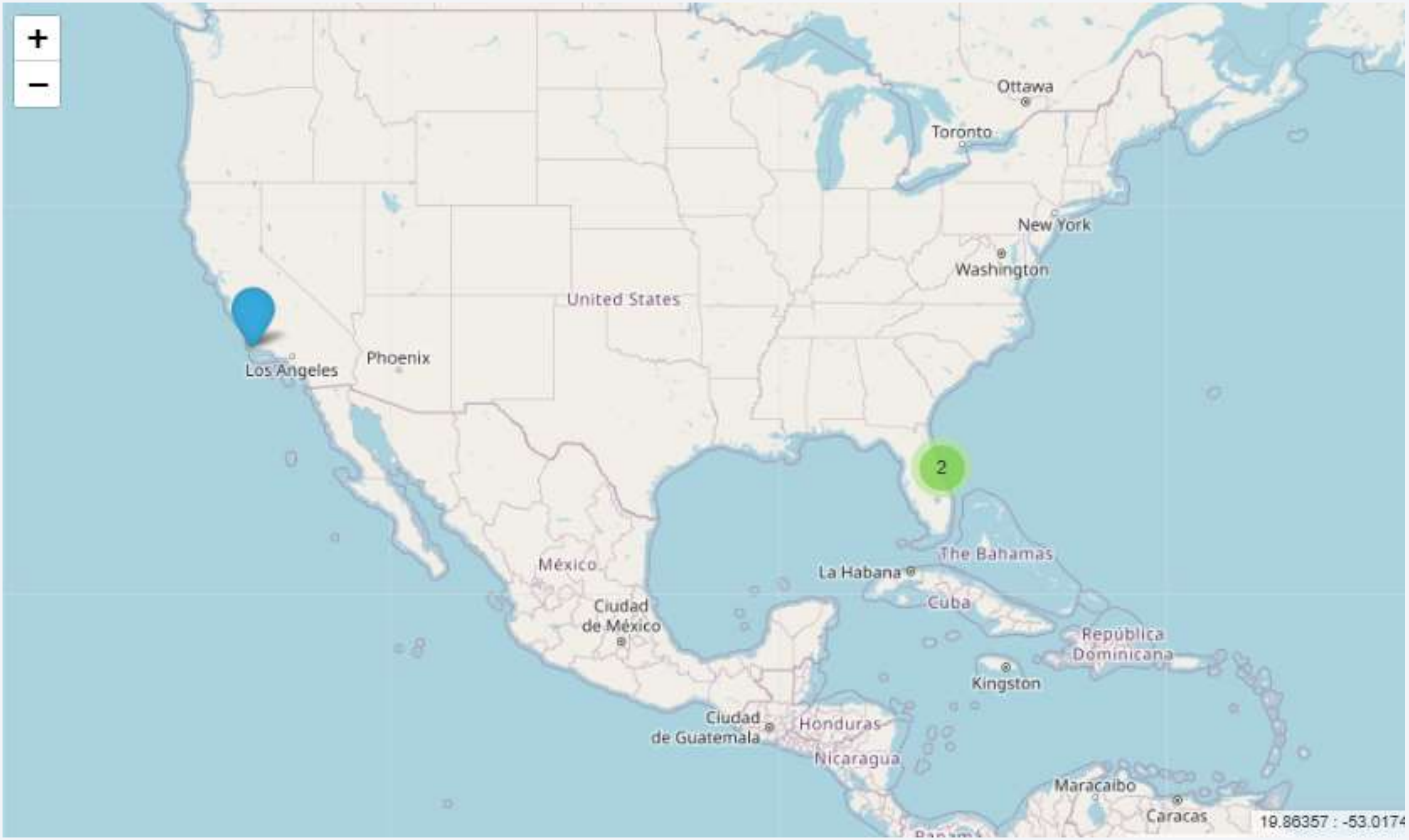
Done.

Landing_Outcome	Outcome_Count
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

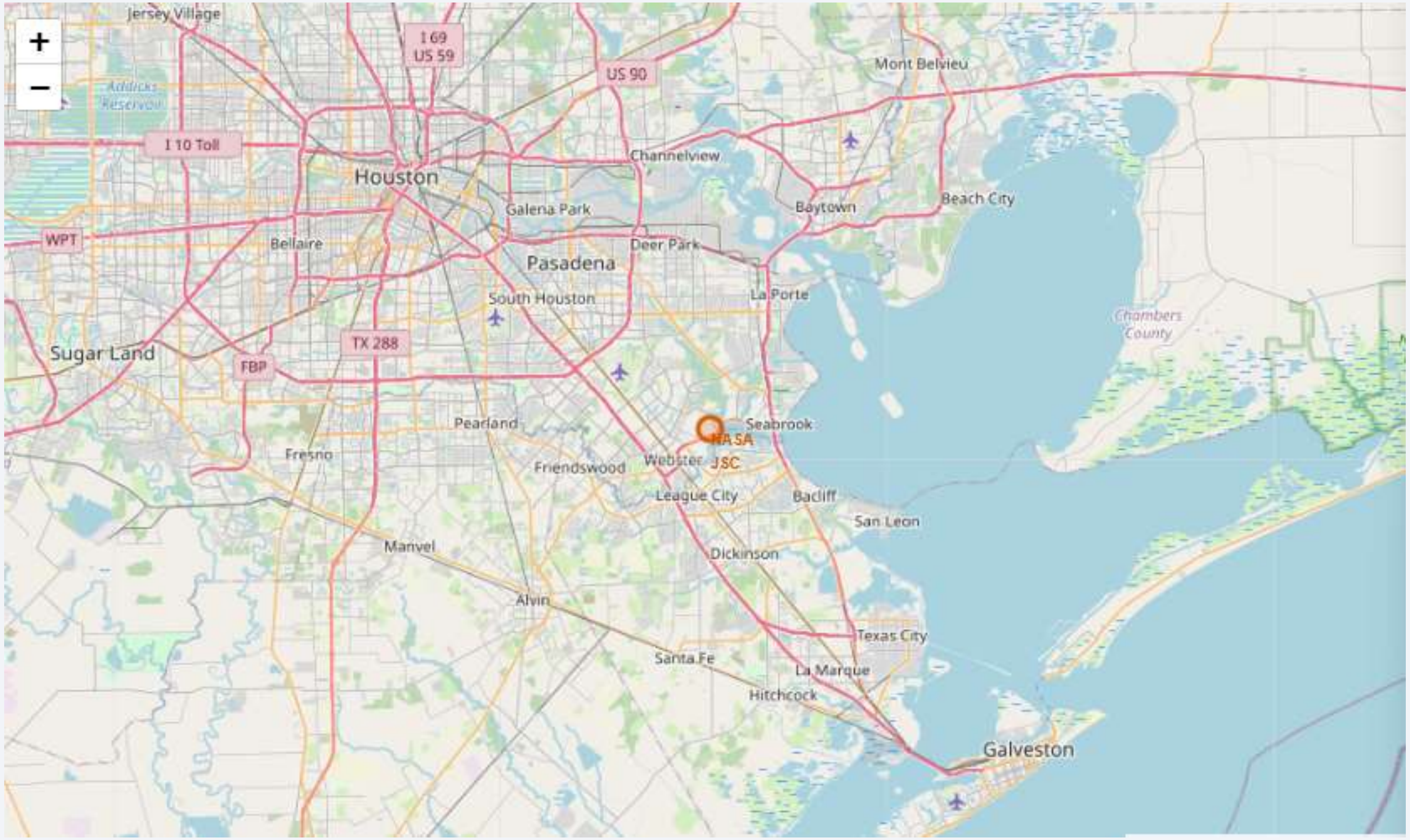
Section 3

Launch Sites Proximities Analysis

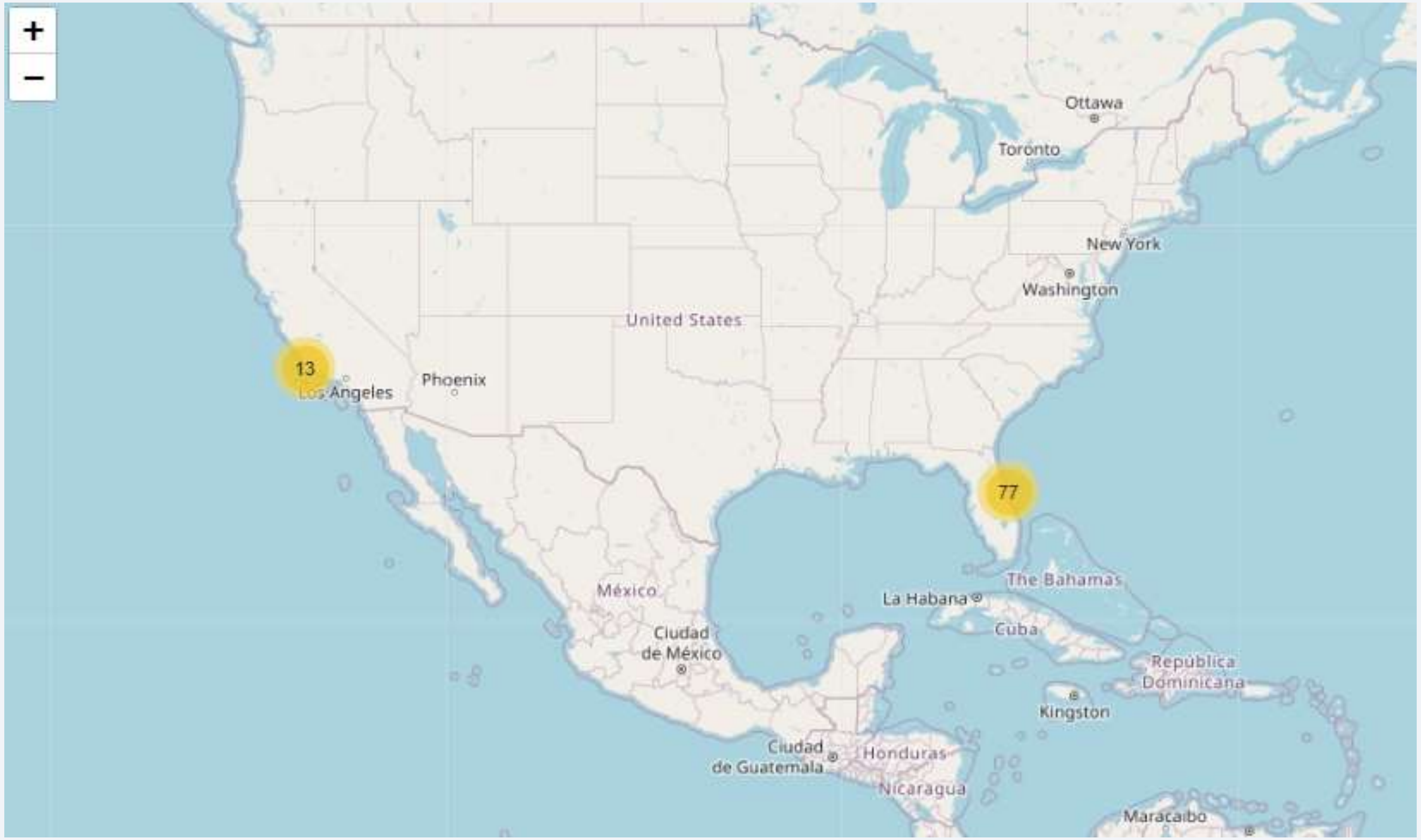
Launch Sites



Nasa Johnson space center



Total launchs in each site



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Logistic Regression

```
0.8333333333333334
```

Support Vector Machine

```
0.8333333333333334
```

Decision Tree Classifier

```
0.7222222222222222
```

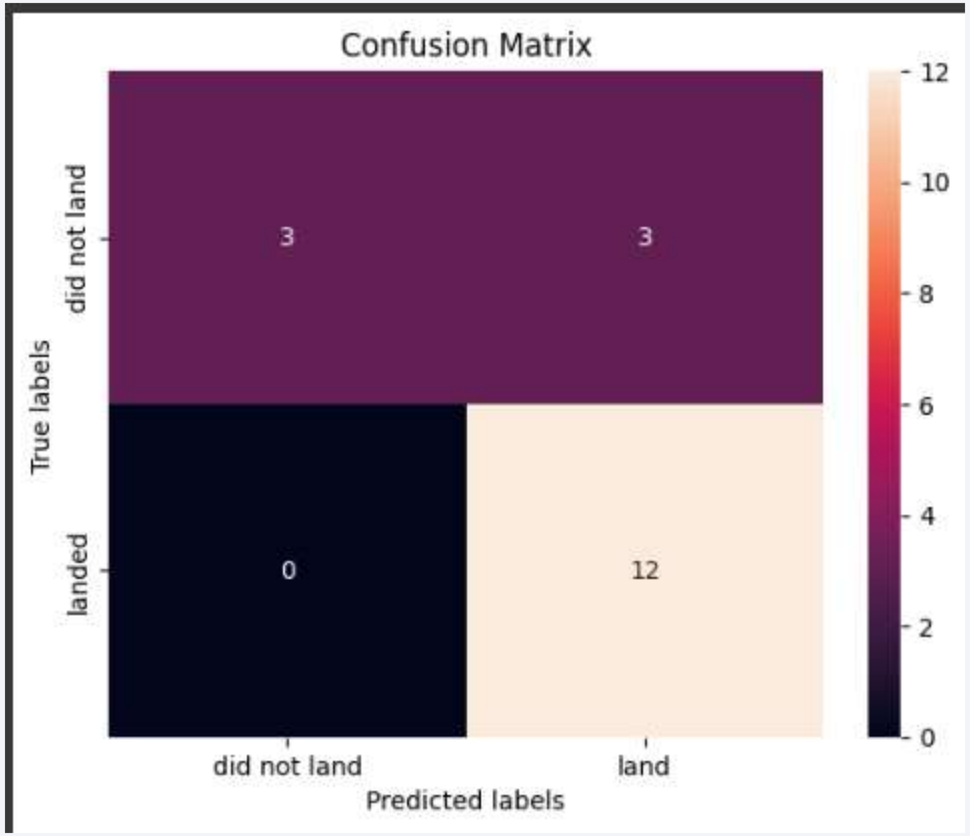
K nearest neighbors

```
0.7777777777777778
```

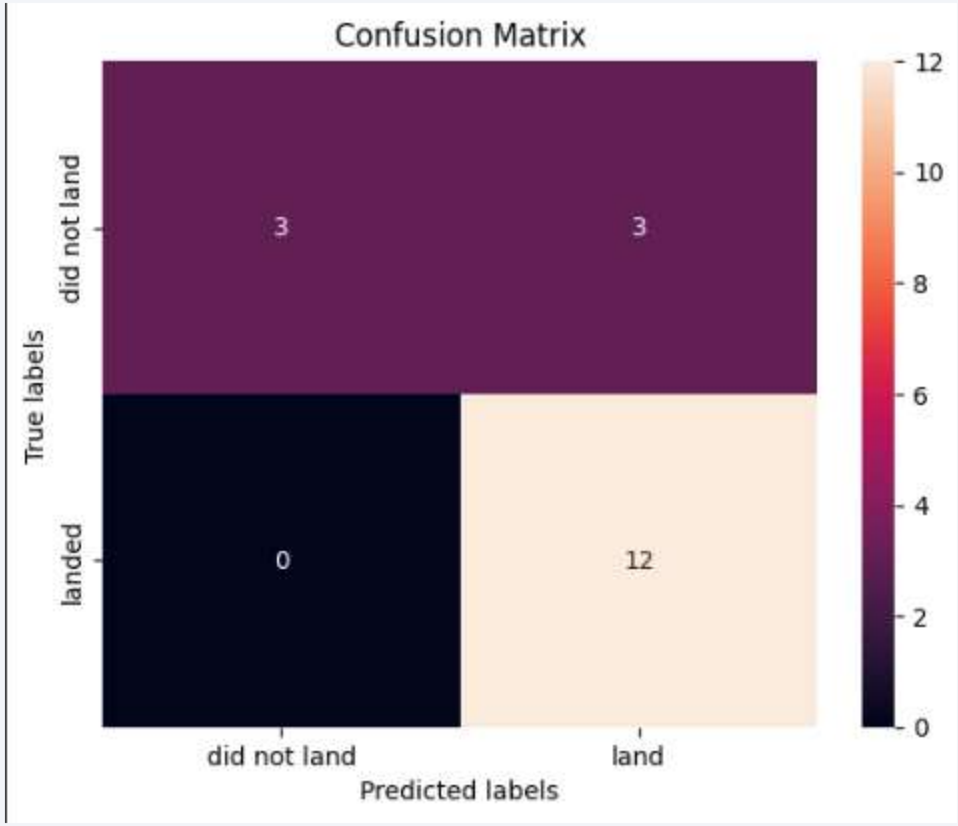
Confusion Matrix

DRAW

Linear Regression



Support Vector Machine



Conclusions

Objective: The project's primary goal was to develop a predictive model to determine the success or failure of SpaceX's Falcon 9 first-stage landings.

Methodology: Four distinct classification models were employed and evaluated: Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).

Key Finding: The analysis revealed that the Decision Tree classifier consistently outperformed the other models, demonstrating the highest accuracy in forecasting landing outcomes based on the provided dataset.

Conclusion: This study successfully demonstrates that machine learning can be effectively applied to predict Falcon 9 landing success with a high degree of accuracy.

Appendix

Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!