IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Javier Dalmau
Feb-2025

Outline

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

Executive Summary

The goal of this project is to analyze historical SpaceX Falcon 9 launch data and develop machine learning classification models to predict the success of the first-stage landing. Since the successful recovery of the first stage results in a substantial cost reduction, accurately modeling landing outcomes is critical for assessing the economic viability of future missions.

- Summary of methodologies
  - Data Collection: Leveraged two primary sources—the official SpaceX API and automated Web Scraping from Wikipedia launch records.
  - Data Wrangling & EDA: Performed extensive data cleaning and filtering using Pandas and SQL to isolate key predictive variables.
  - Exploratory Data Analysis (EDA) and visualization (Matplotlib/Seaborn) were used to identify trends and correlations.
  - Geospatial Analysis & Interactive Visualization: Utilized Folium to map launch site proximities and Plotly to build an interactive dashboard for real-time analysis of launch records and success rates.
  - Machine Learning Modeling: Developed and trained four classification algorithms. GridSearchCV was utilized for hyperparameter tuning to optimize model performance and ensure robust predictions.
- Summary of all results
  - Data Insights: Launch site location, payload mass, and orbit type were identified as significant predictors of landing outcomes.
  - Model Performance: All models demonstrated high accuracy levels; however, through rigorous hyperparameter tuning and cross-validation, the SVM was identified as the optimal model, achieving an accuracy of 84.8%.

Introduction

- Project background and context
  - SpaceX has revolutionized the aerospace industry by making rocket launches more affordable. The cornerstone of this achievement is the reusability of the Falcon 9 first stage. Traditionally, rockets were "expendable," meaning they were destroyed or lost after a single use. SpaceX’s ability to land and reuse the first stage has slashed the cost of access to space. While SpaceX advertises Falcon 9 launches at approximately $62 million, competitors still face costs exceeding $165 million. Much of this price difference is attributed to the successful recovery and refurbishment of the booster.
- Problems you want to find answers
  - Despite high success rates, landing a rocket is a complex feat of engineering influenced by numerous variables. To compete in the market or provide insurance for these missions, it is vital to understand the factors that lead to a successful landing.

Section 1

Methodology

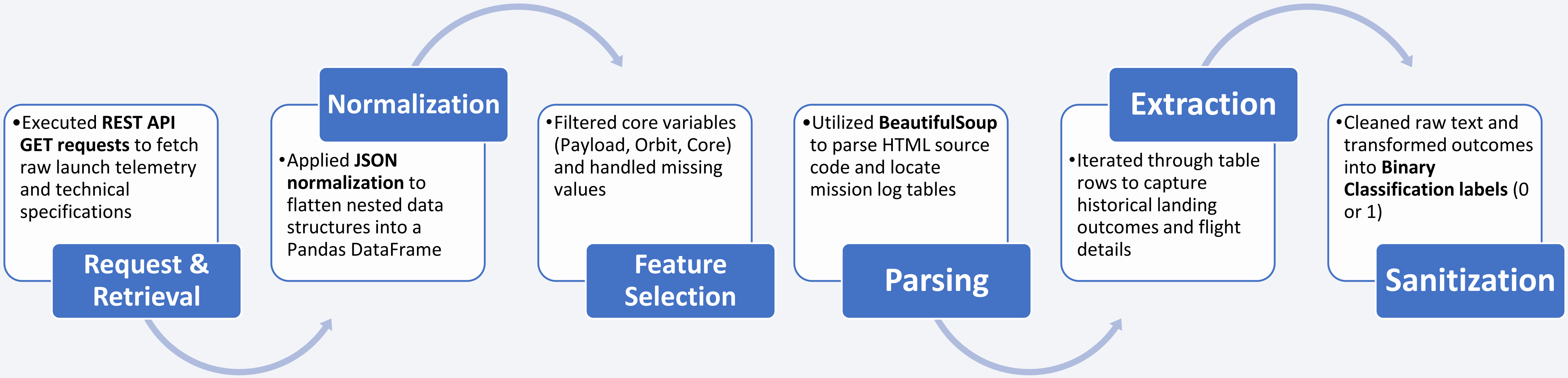
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API: Extracted technical data including payload mass, orbit type, and booster versions.
  - Web Scraping: Captured historical launch outcomes and mission details from Wikipedia using BeautifulSoup.
- Perform data wrangling
  - Data was cleaned by handling missing values and converting outcomes into a binary variable: 1 for a successful landing and 0 for a failure.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We optimized four classification models using GridSearchCV with 10-fold cross-validation to find the best hyperparameters, then evaluated their performance on unseen data using Accuracy scores and Confusion Matrices to identify the most reliable predictor.

Data Collection

- SpaceX API: Utilized the requests library to fetch JSON data from the official SpaceX API. This provided granular technical details such as payload mass, booster versions, and core reuse history.
- Web Scraping: Used BeautifulSoup to parse HTML tables from Wikipedia's "List of Falcon 9 and Falcon Heavy launches.".



Data Collection – SpaceX API

GitHub URL

Request & Retrieval

- Executed **REST API GET requests** to fetch raw launch telemetry and technical specifications

Normalization

- Applied **JSON normalization** to flatten nested data structures into a Pandas DataFrame

Feature Selection

- Filtered core variables (Payload, Orbit, Core) and handled missing values

```
spacex_url = "https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
print(response.content)
```

'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"s

```
# We will use the following static response object for this project

static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.a
response = requests.get(static_json_url)
print(response.content)
```

```
data = response.json()
data = pd.json_normalize(data)
data.head()
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data = pd.DataFrame(launch_dict)
data.head()
```

Data Collection - Scraping

GitHub URL

Parsing

- Utilized **BeautifulSoup** to parse HTML source code and locate mission log tables

Extraction

- Iterated through table rows to capture historical landing outcomes and flight details

Sanitization

- Cleaned raw text and transformed outcomes into **Binary Classification labels** (0 or 1)

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falc

headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/91.0.4472.124 Safari/537.36"
}
```

```python
response = requests.get(static_url, headers=headers)
html = response.text
soup = BeautifulSoup(html, 'html.parser')
print(soup.title.string)
```

```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

```python
html_tables = soup.find_all('table')
```

```python
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```python
landing_class = []

for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```python
df['Class']=landing_class
df[['Class']].head(8)
```

Data Wrangling

GitHub URL

- Data Cleaning & Imputation
  - Handled missing values in the PayloadMass column (Mean Imputation Strategy).
  - Replaced null values with the mean of the available payload data.
- Feature Engineering (Categorical to Numerical)
  - Prepared categorical variables for the algorithms (One-Hot Encoding).
  - Used pd.get_dummies to convert categories into binary columns (0 or 1), creating a feature set suitable for classification models.
- Binary Labeling (Target Variable)
  - Defined the success criteria for the first-stage landing (Binary Classification Mapping).
  - Created the Class column where 1 represents a successful landing and 0 represents a failure or unsuccessful outcome.

EDA with Data Visualization

GitHub URL

We utilized Pandas, Matplotlib, and Seaborn to identify trends and correlations within the SpaceX dataset.

- **Scatter Plots**: To visualize the "learning curve" of SpaceX. These charts revealed that as the Flight Number increased (time progressed), the frequency of successful landings (Class 1) improved across all launch sites.
- **Bar Charts**: To determine which orbits are "safer" for booster recovery. We discovered that ES-L1, GEO, HEO, and SSO orbits showed the highest success rates.
- **Line Charts**: To illustrate the company's evolution. This chart clearly showed a positive slope in landing success from 2013 to 2020, proving the maturity of the Falcon 9 recovery technology.

EDA with SQL

GitHub URL

By loading the processed dataset into a DB2 or SQLite database, we executed complex queries to derive specific statistical insights:

- **Launch Site Identification**: Queried unique launch sites to understand the geographical distribution of SpaceX operations.
- **Mission Success Filters**: Isolated the first 5 successful landings where the ground pad was used and the payload mass was between 4,000 and 6,000 kg.
- **Payload Statistics**: Calculated the total, average, and maximum payload masses to understand the heavy-lifting capabilities of the Falcon 9.
- **Failure Analysis**: Identified the names of boosters that failed to land successfully.
- **Booster Performance**: Grouped data by booster version to identify which specific "Blocks" showed the most significant improvements in reliability.
- **Date-Based Trends**: Founded the most successful year and the specific month with the highest mission count.

Build an Interactive Map with Folium

GitHub URL

- Geospatial Objects & Implementation
  - Color-coded Markers & Clusters: Used to visualize landing outcomes and manage high-density launch data.
  - Circles: Applied as safety perimeters around launch pads to analyze site isolation.
  - PolyLines: Plotted to measure the direct distance between launch sites and critical infrastructure.
- Strategic Rationale
  - Proximity Analysis: To prove that all sites are strategically located near coastlines for safety and railways for booster logistics.
  - Distance Optimization: Used to quantify how site location influences recovery operations.

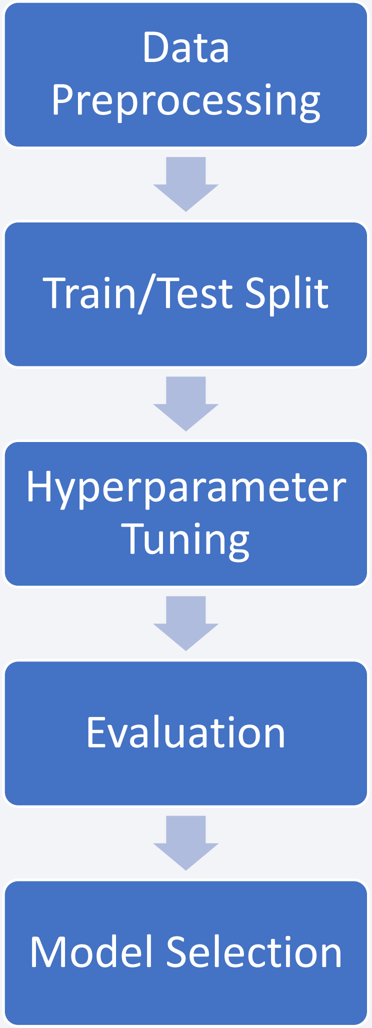
Build a Dashboard with Plotly Dash

GitHub URL

- Key Visualizations
  - **Pie Charts**: Integrated to show the total success launches for all sites or a specific selected site.
  - **Scatter Plots**: Implemented to visualize the relationship between Payload Mass and Success, color-coded by booster version.
  - **Interactive Dropdowns**: Allowed users to filter data by specific Launch Sites.
  - **Range Sliders**: Enabled real-time filtering of payload weight to observe how mass impacts landing probability.
- Strategic Rationale
  - **Dynamic Data Discovery**: To allow stakeholders to instantly compare success rates across different geographical locations.
  - **Correlation Analysis**: To visually confirm if heavier payloads correlate with higher failure rates and identify the performance of different booster versions.

Predictive Analysis (Classification)

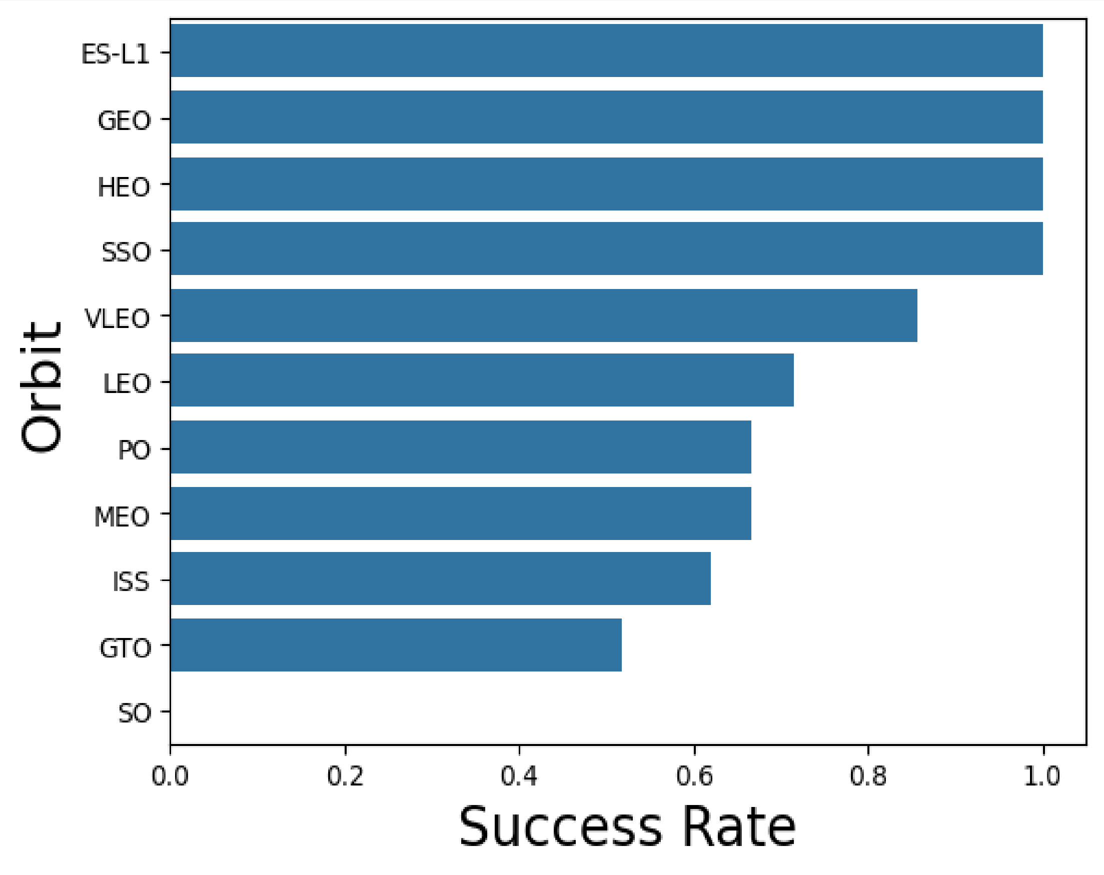
GitHub URL

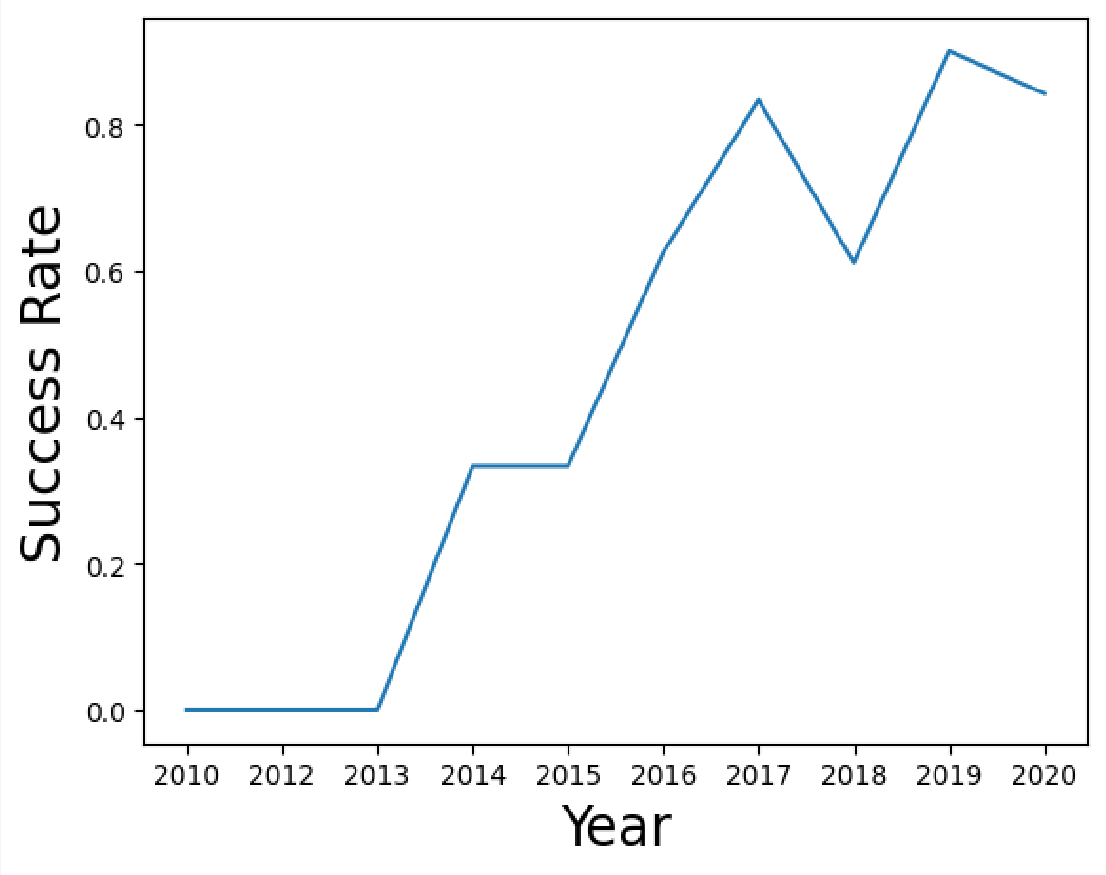


- **Preprocessing**: Standardized the feature matrix using StandardScaler to ensure all variables contributed equally, particularly for distance-based models like SVM and KNN.
- **Hyperparameter Tuning**: Implemented GridSearchCV with 10-fold cross-validation to systematically test combinations of parameters for four algorithms:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - k-Nearest Neighbors (KNN)
- **Evaluation**: Validated model performance using Accuracy Scores and Confusion Matrices to measure precision and recall.

Results

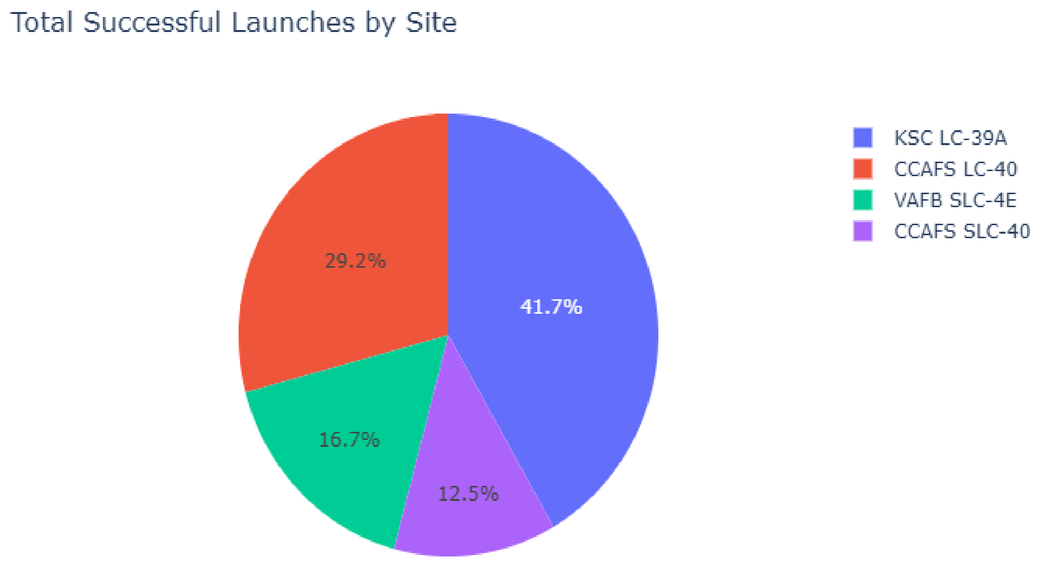
- Exploratory data analysis results

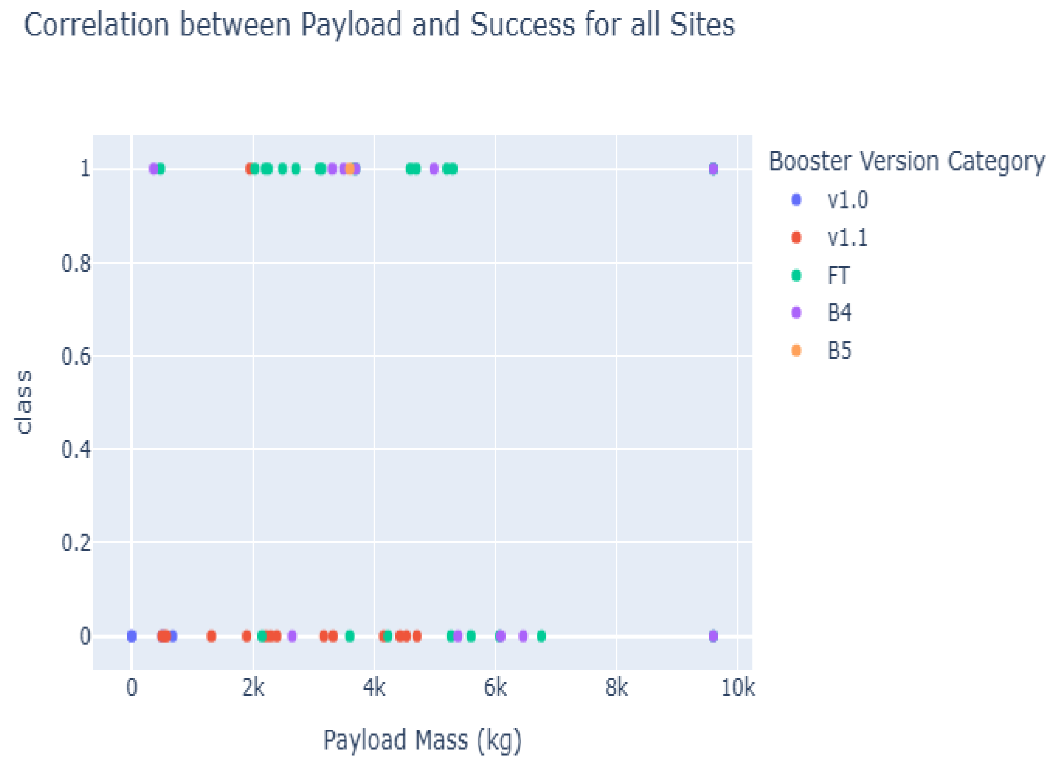




Results

- Interactive analytics demo in screenshots





Results

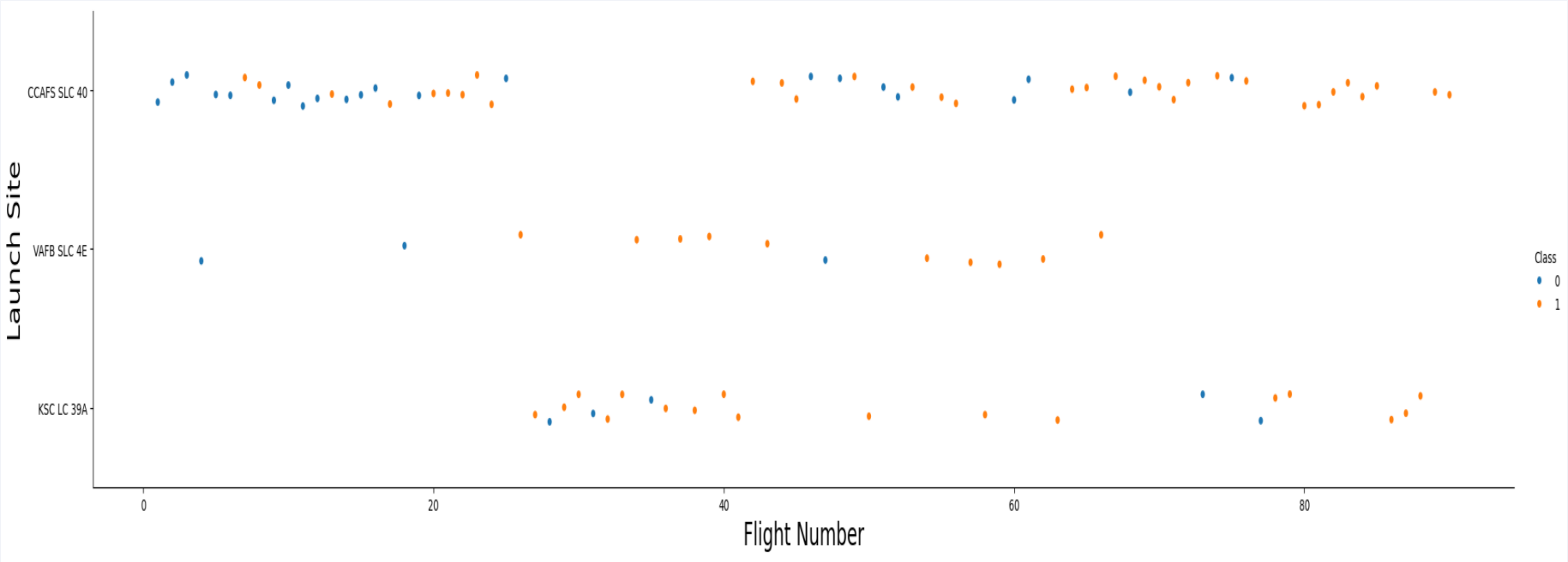
Model	Best Score
Logistic Regression	0.821429
SVM	0.848214
KNN	0.833929
Decision Tree	0.875000

- Predictive analysis results

Section 2

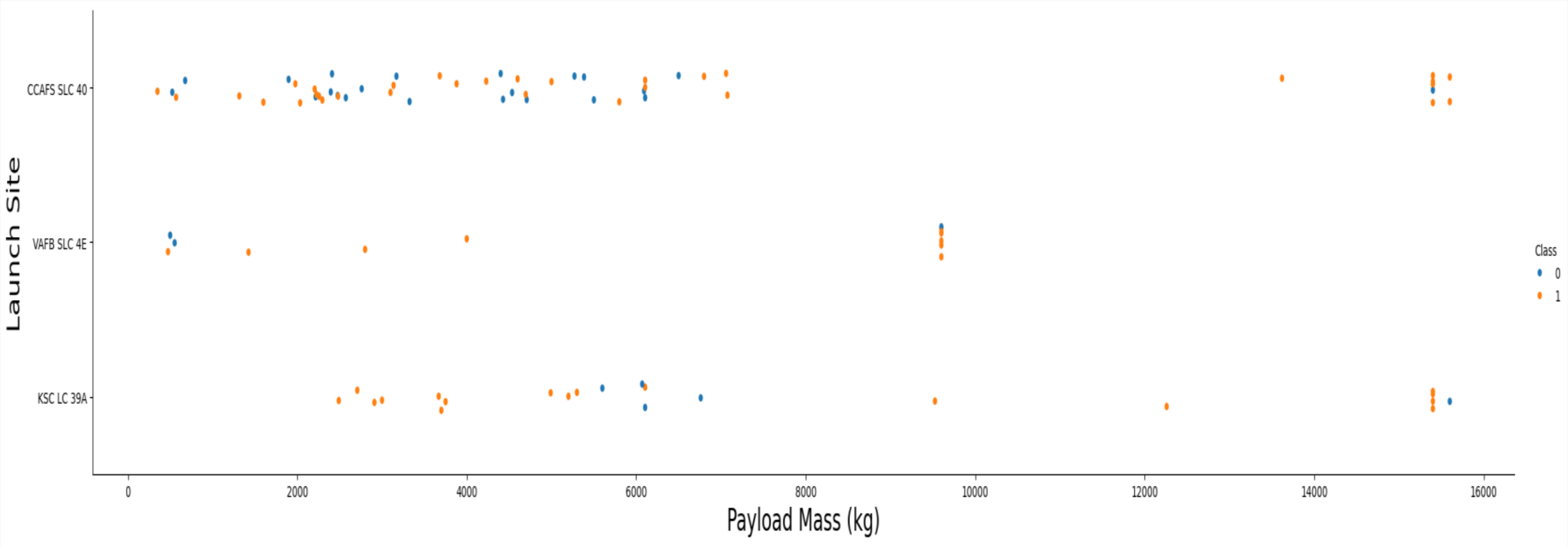
Insights drawn from EDA

Flight Number vs. Launch Site



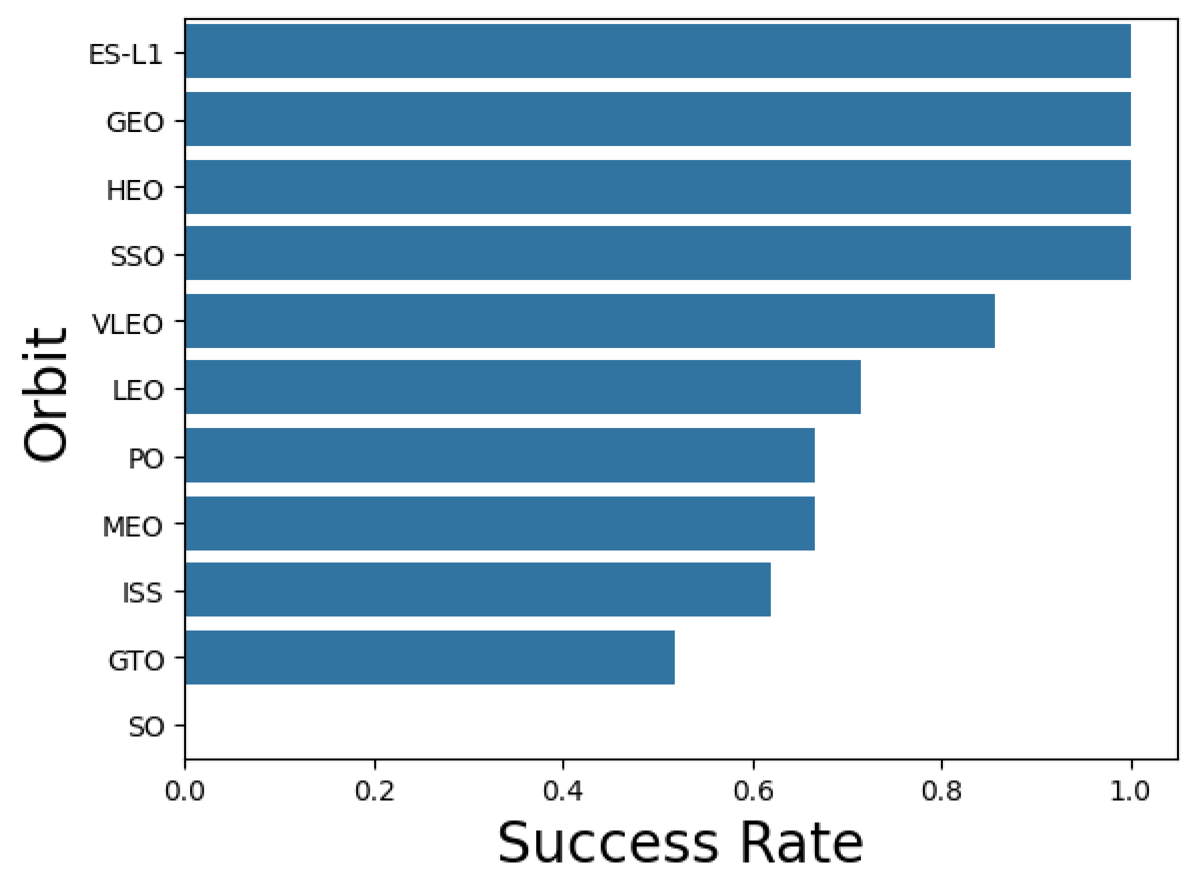
- Success rates improved significantly as Flight Number increased.
- Newer sites (like KSC LC 39A) achieved high reliability faster by leveraging lessons from earlier missions.

Payload vs. Launch Site



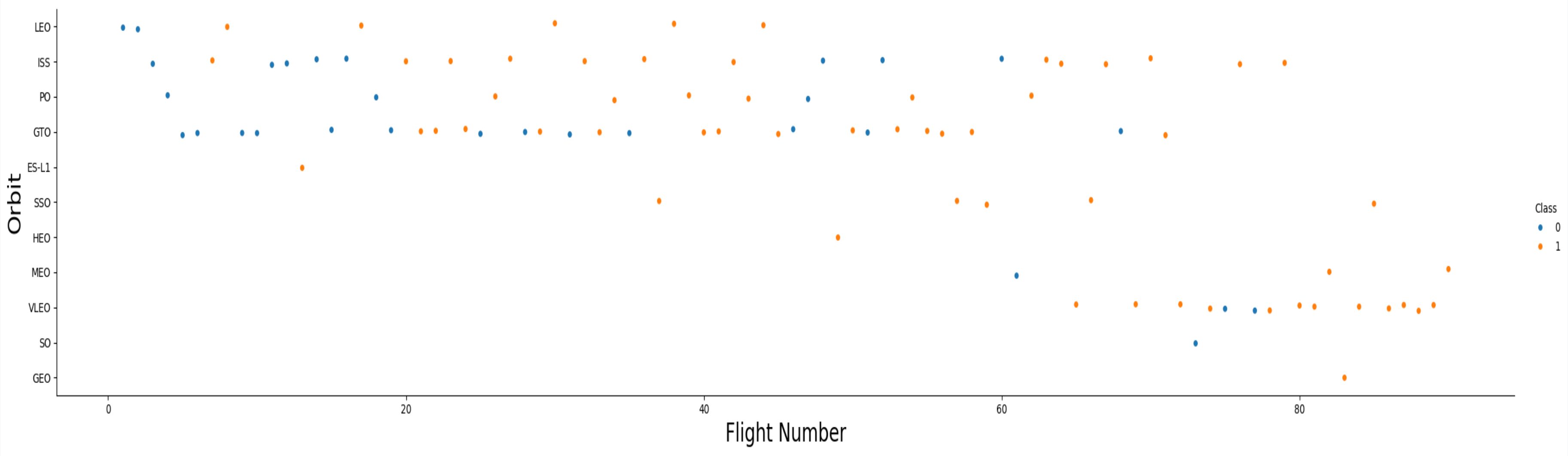
- Successes are distributed across the entire mass spectrum, from light payloads to heavy loads exceeding 15,000 kg.
- KSC LC 39A and VAFB SLC 4E show high reliability for heavy payloads.

Success Rate vs. Orbit Type



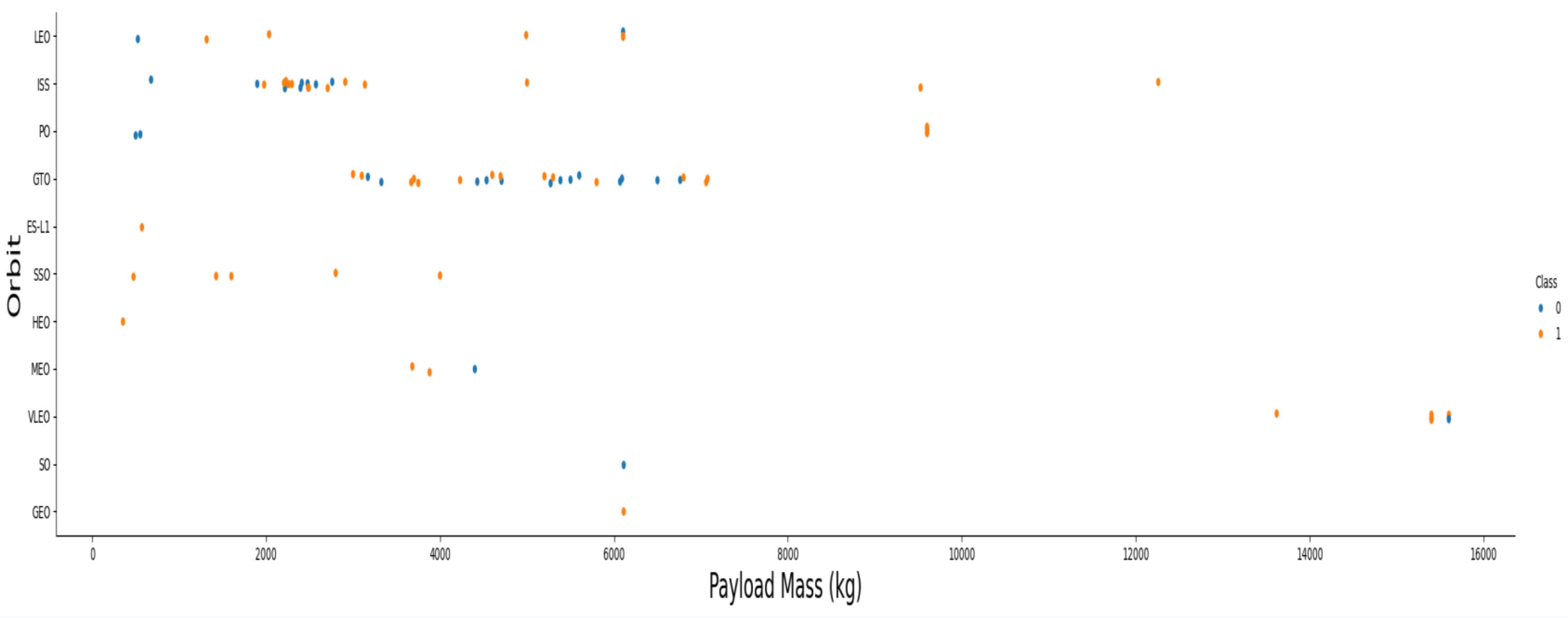
- Orbits ES-L1, GEO, HEO, and SSO achieved a perfect 100% success rate.
- Success rates fluctuate across other orbits, with GTO showing the lowest success rate at approximately 50%

Flight Number vs. Orbit Type



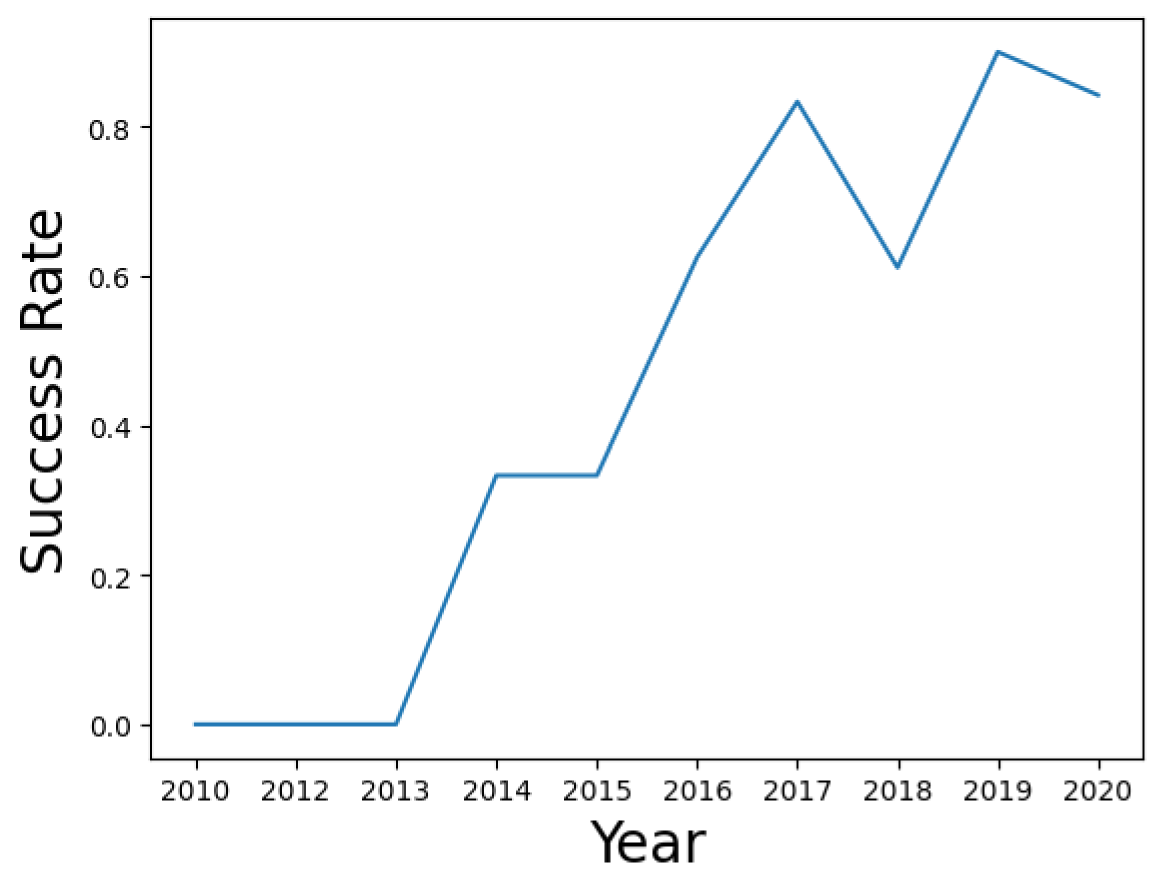
- Early missions were concentrated in LEO, ISS, and GTO orbits, while later missions expanded heavily into VLEO and other specialized orbits.
- The frequency of successful landings increases significantly as flight numbers progress.

Payload vs. Orbit Type



- Successful landings are achieved across diverse orbits even with heavy payloads exceeding 15,000 kg.
- Certain orbits, such as SSO, HEO, and ES-L1, show consistent success with lighter payloads.

Launch Success Yearly Trend



- The landing success rate shows a strong upward trajectory from 0% in 2013 to reaching its peak in 2019.
- After initial stagnation between 2010 and 2013, the significant climb after 2015 demonstrates the successful stabilization of recovery technology.

All Launch Site Names

- %sql SELECT Launch_Site, COUNT(*) as Count FROM SPACEXTABLE GROUP BY Launch_Site;

Launch_Site	Count
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

Launch Site Names Begin with 'CCA'

- %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	M
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	
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	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	

Total Payload Mass

- %sql SELECT BOOSTER_VERSION, SUM(PAYLOAD_MASS__KG_) AS Total_Payload FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)' GROUP BY BOOSTER_VERSION;

Booster_Version	Total_Payload
F9 B4 B1039.2	2647
F9 B4 B1039.1	3310
F9 B4 B1045.2	2697
F9 B5 B1056.2	2268
F9 B5 B1058.4	2972
F9 B5 B1059.2	1977
F9 B5B1050	2500
F9 B5B1056.1	2495
F9 FT B1035.2	2205
F9 FT B1021.1	3136
F9 FT B1025.1	2257
F9 FT B1031.1	2490
F9 FT B1035.1	2708
F9 v1.0 B0006	500
F9 v1.0 B0007	677
F9 v1.1	2296
F9 v1.1 B1010	2216
F9 v1.1 B1012	2395
F9 v1.1 B1015	1898
F9 v1.1 B1018	1952

Average Payload Mass by F9 v1.1

- %sql SELECT BOOSTER_VERSION, AVG(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1';

Booster_Version	Avg_Payload
F9 v1.1	2928.4

First Successful Ground Landing Date

- %sql SELECT Date, Landing_Outcome from SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' ORDER BY Date ASC LIMIT 1;

Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

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

Total Number of Successful and Failure Mission Outcomes

- %sql SELECT CASE WHEN Mission_Outcome LIKE 'Success%' THEN 'Success' WHEN Mission_Outcome LIKE 'Failure%' THEN 'Failure' END AS Outcome_Category, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY 1;

Outcome_Category	Total
Failure	1
Success	100

Boosters Carried Maximum Payload

- %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ >= (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) GROUP BY 1;

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- %sql SELECT substr(Date, 6,2) as Month, substr(Date,0,5) as Year, Landing_Outcome, Booster_Version, Launch_Site Total FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date,0,5)='2015';

Month	Year	Landing_Outcome	Booster_Version	Total
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

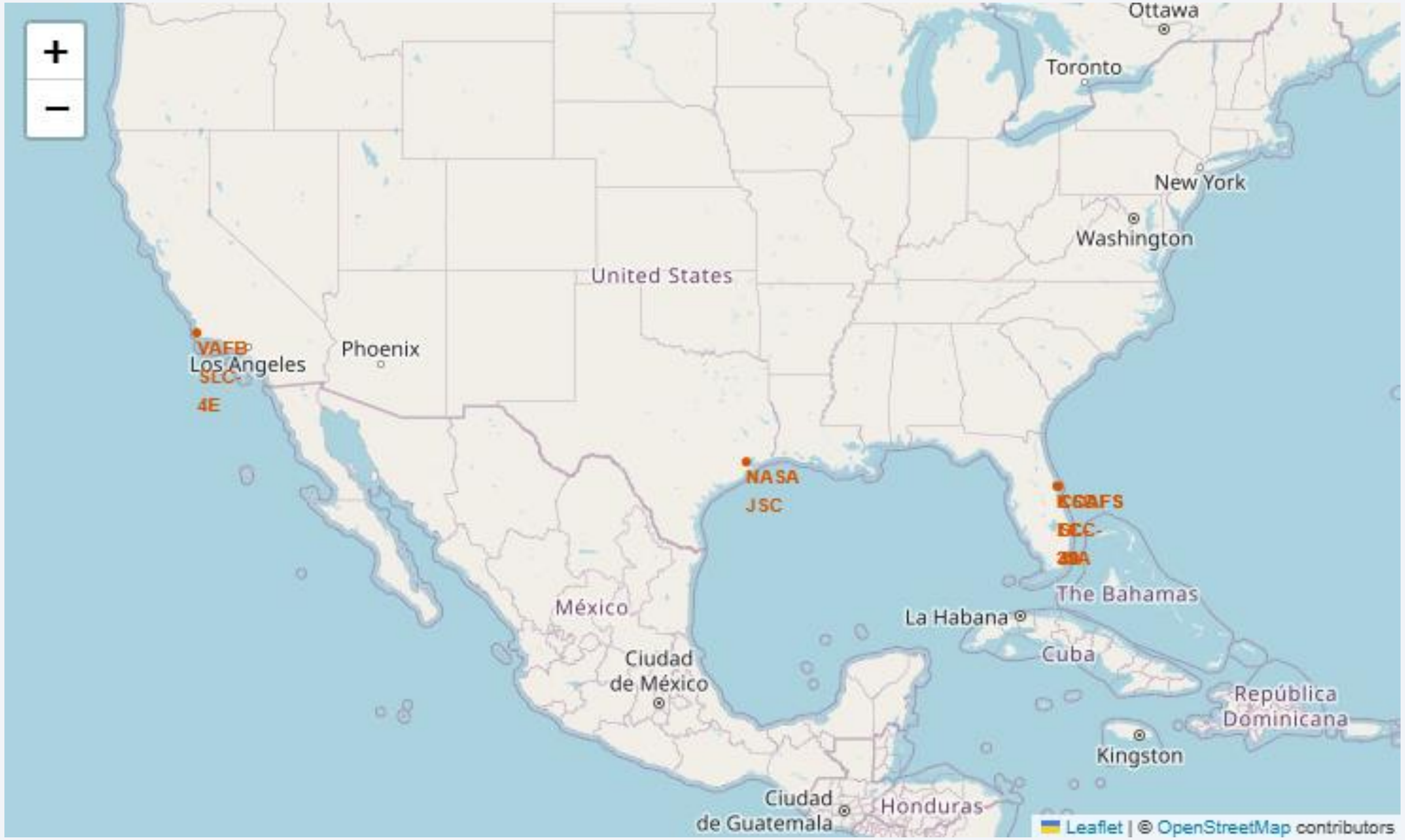
- %sql SELECT Landing_Outcome, COUNT(*) AS Total FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Total DESC;

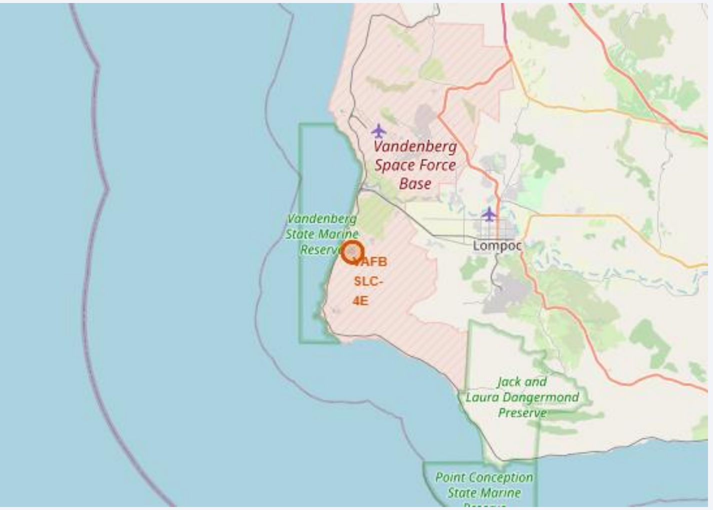
Landing_Outcome	Total
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

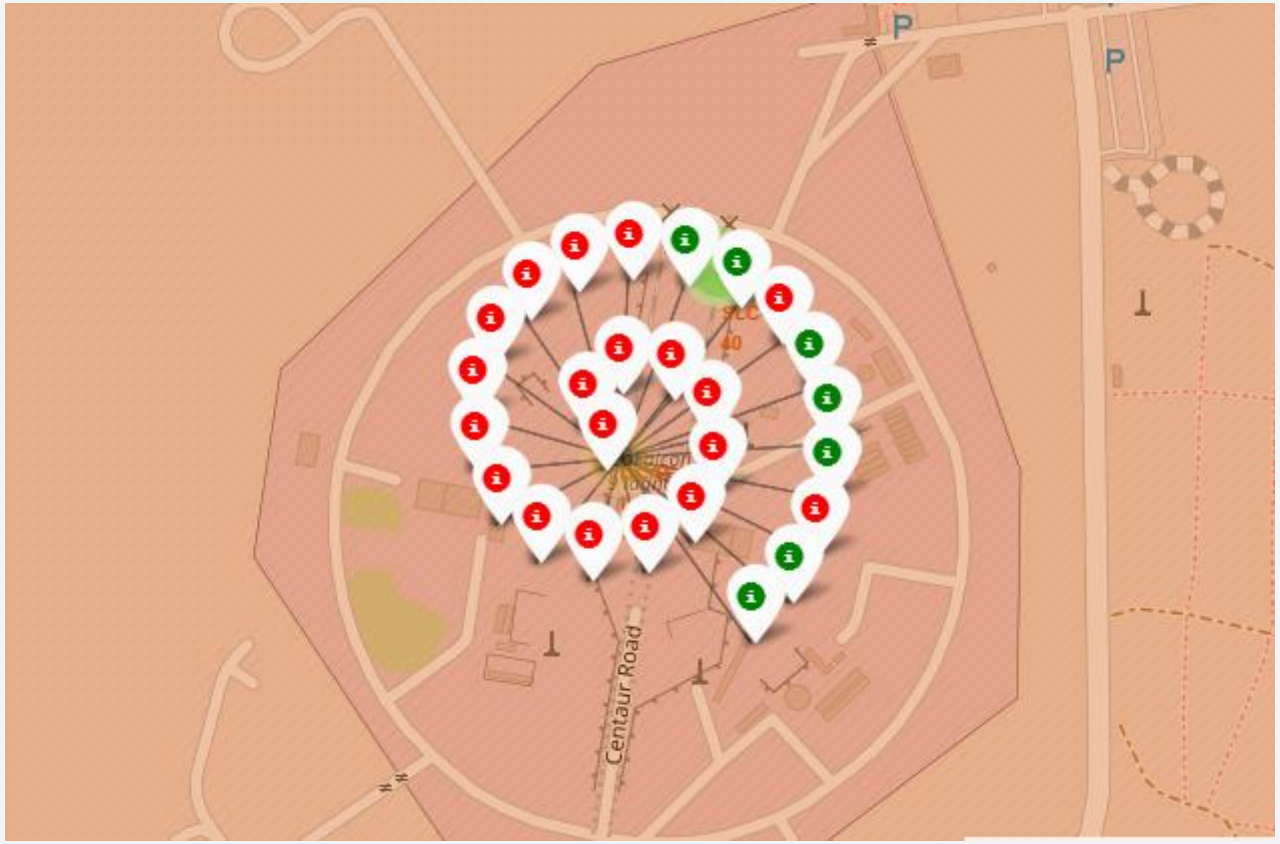
SpaceX Falcon9 – Launch Sites Map





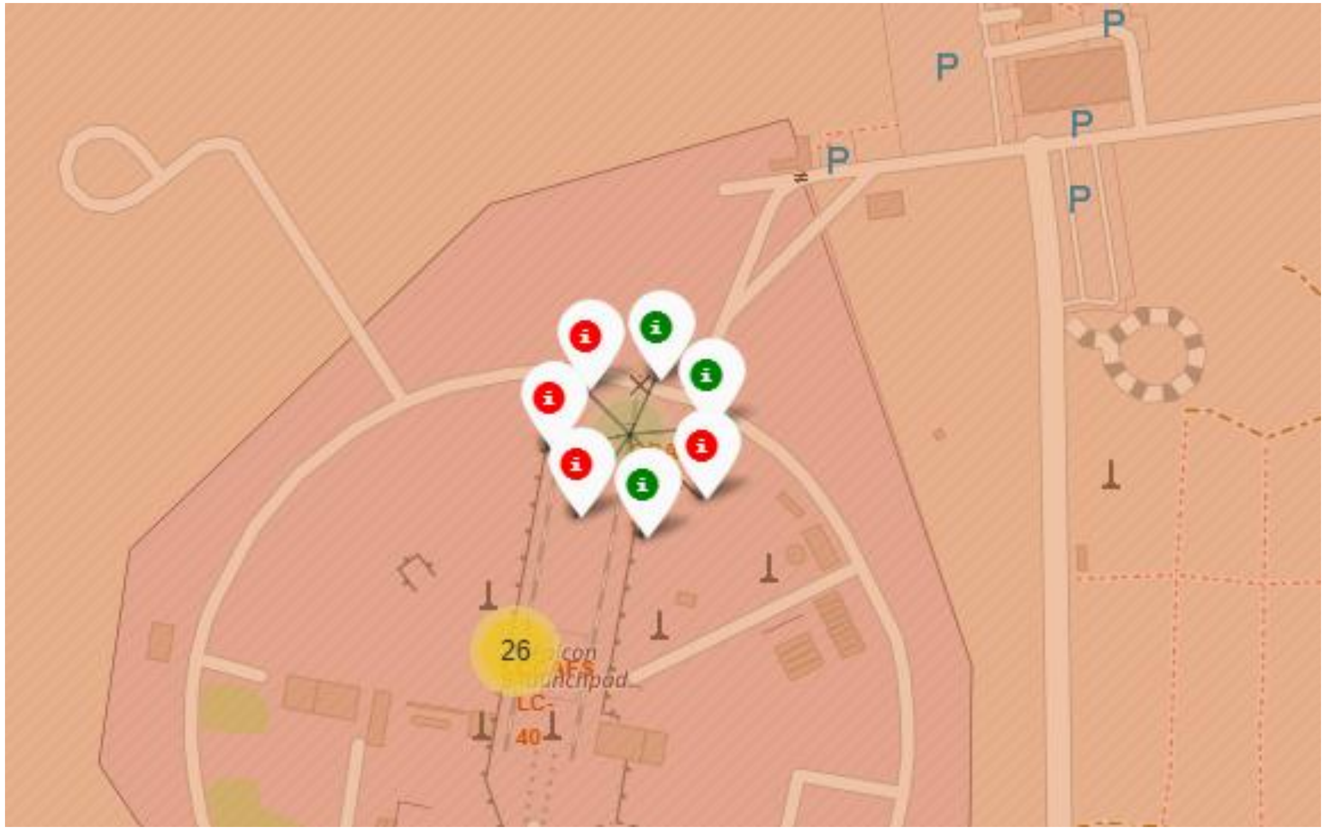
SpaceX Falcon9 – Succes/Failed Launch Map

- CCAFS LC-40



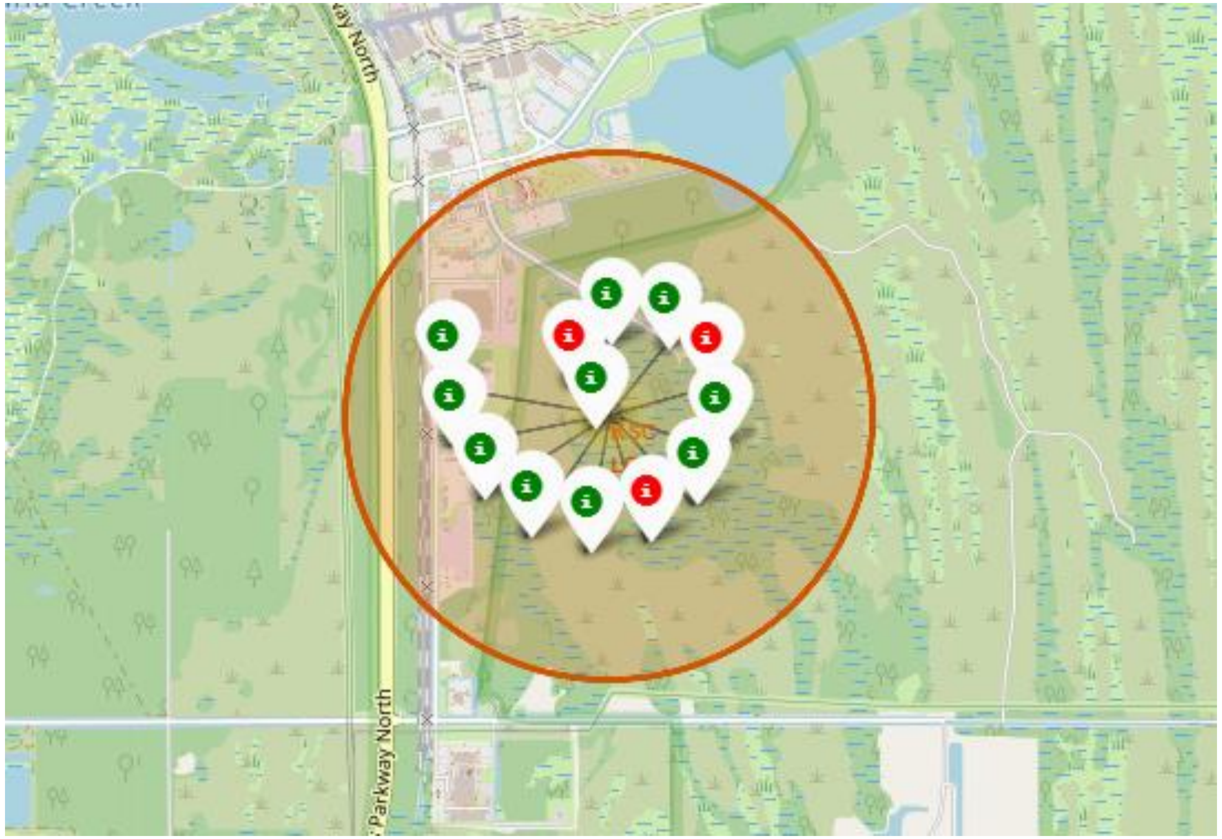
SpaceX Falcon9 – Succes/Failed Launch Map

- CCAFS SLC-40



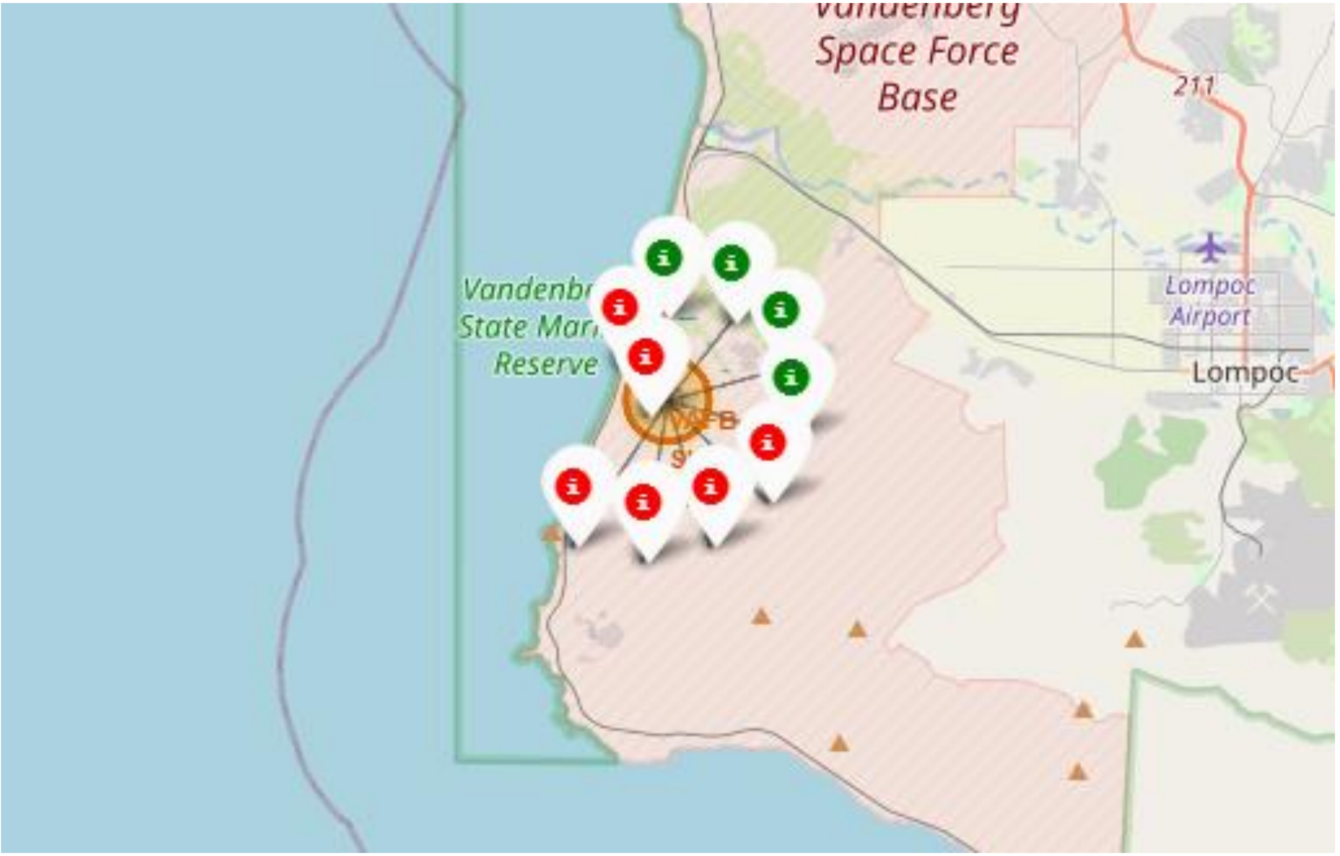
SpaceX Falcon9 – Succes/Failed Launch Map

- KSC LC-39A

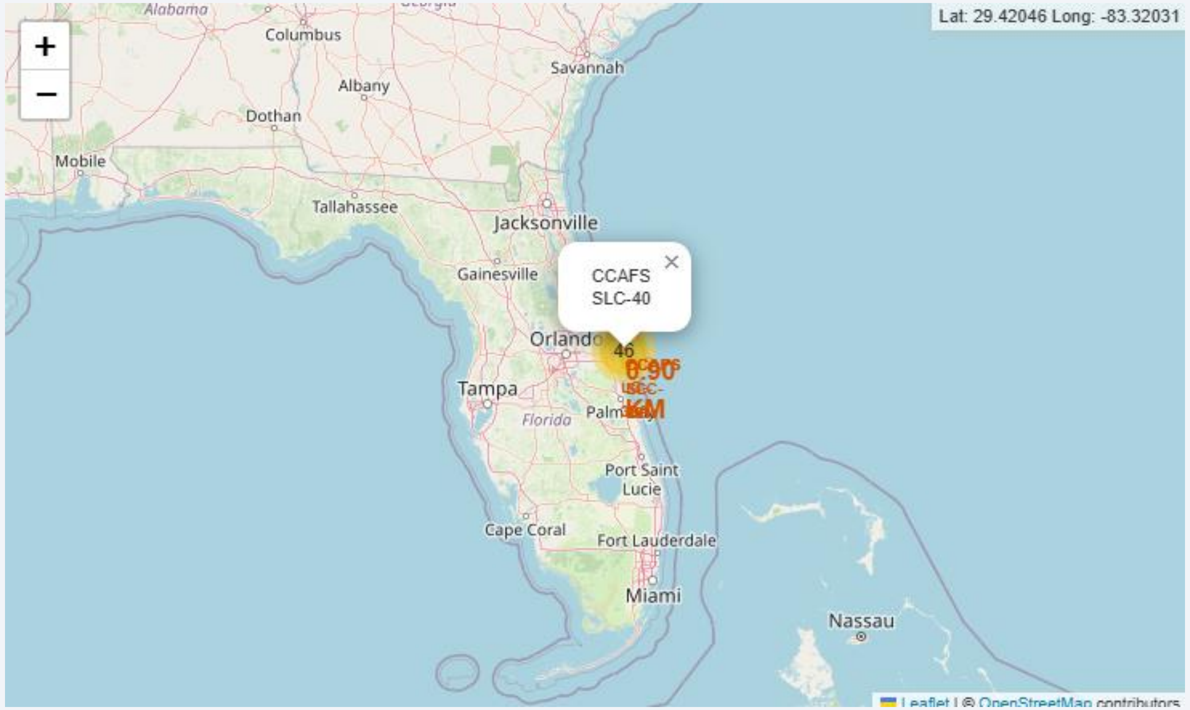


SpaceX Falcon9 – Succes/Failed Launch Map

- VAFB SLC-4E



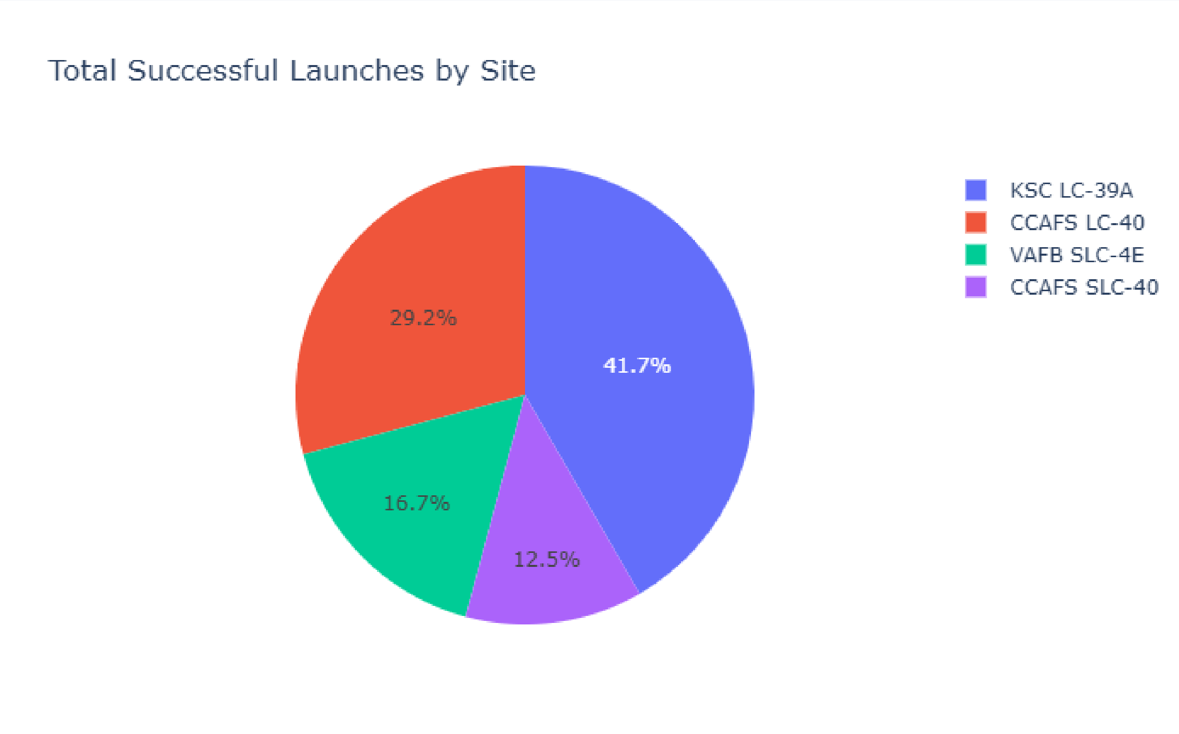
SpaceX Falcon9 – Launch Site to proximity map



Section 4

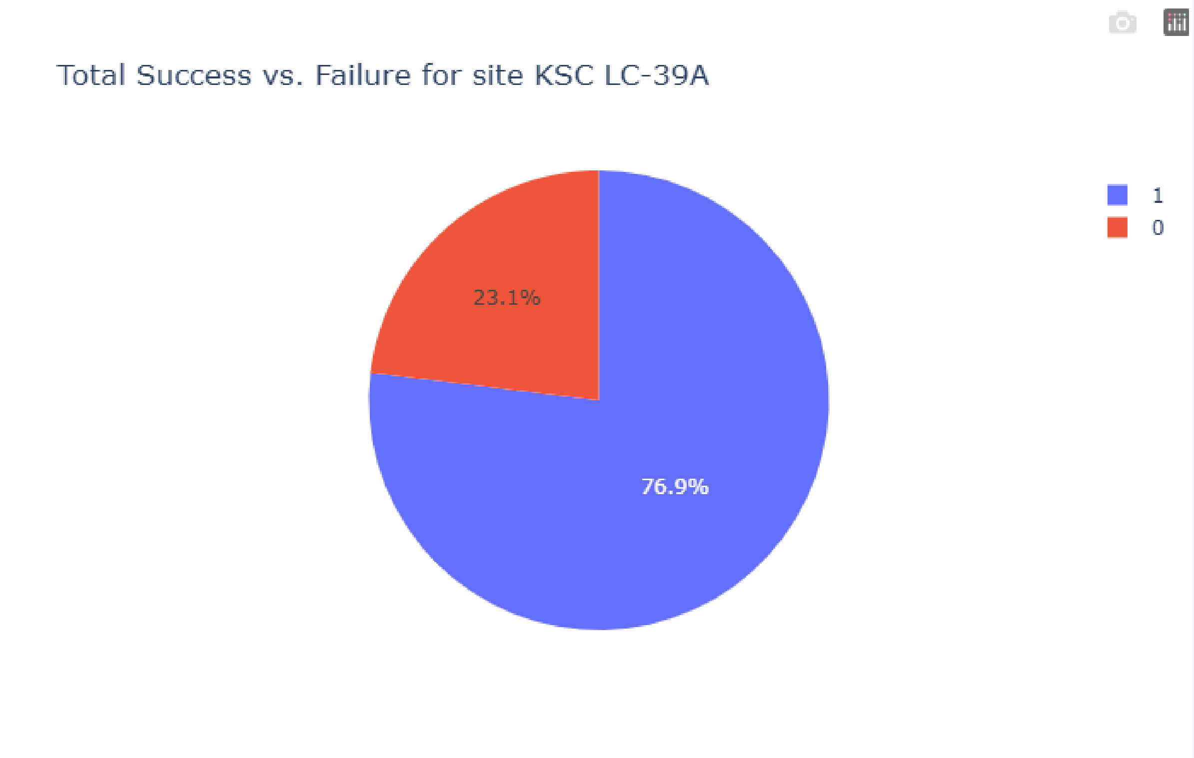
Build a Dashboard with Plotly Dash

Launch Success For All Sites



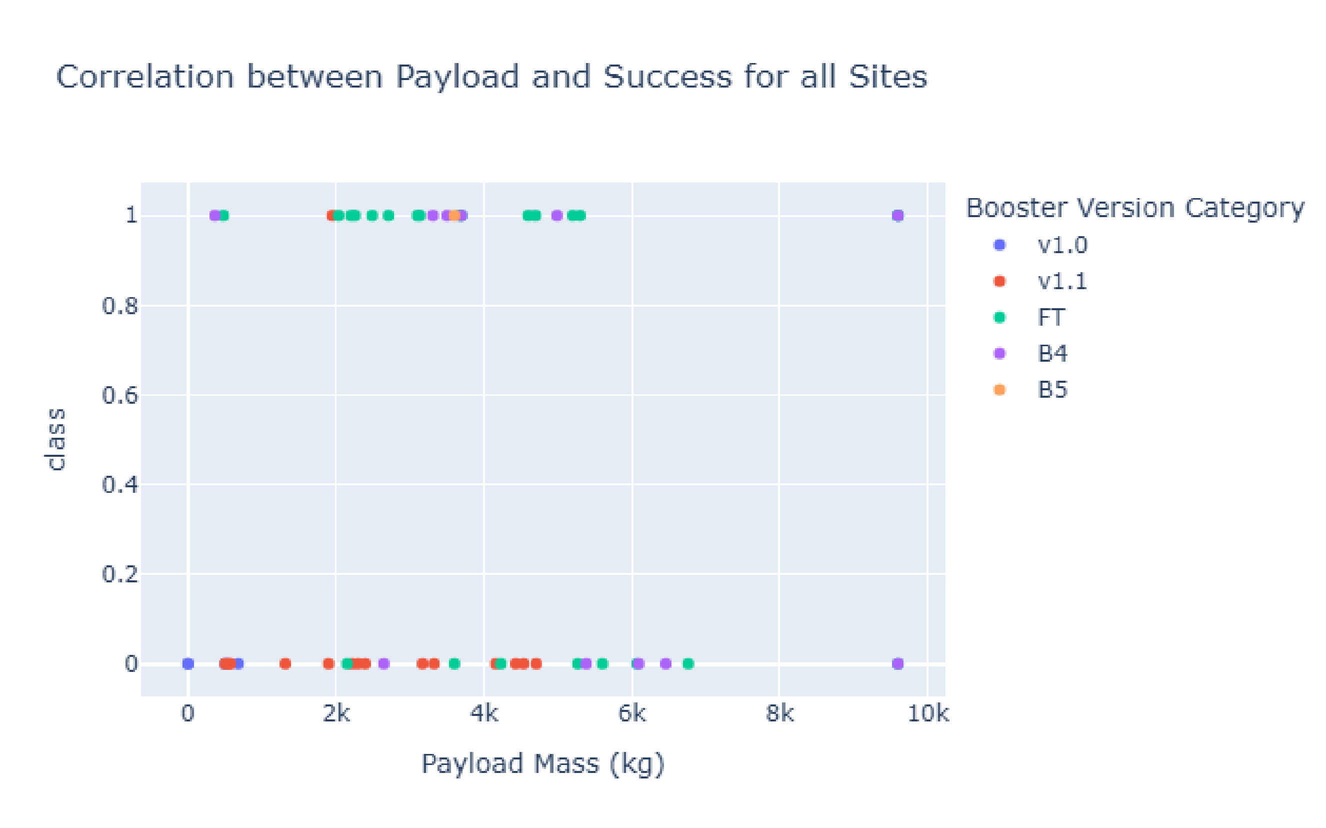
- KSC LC-39A leads all sites, accounting for 41.7% of the total successful landings.
- Successful recoveries are shared across the remaining sites, with CCAFS LC-40 contributing 29.2%, followed by VAFB SLC-4E (16.7%) and CCAFS SLC-40 (12.5%)

Launch Success with Highest Launch Sucess



- The KSC LC-39A launch site exhibits high operational performance, with a success rate of 76.9% versus a failure rate of 23.1%.
- This distribution confirms that KSC LC-39A is one of the most strategic and reliable points for SpaceX missions

Payload v. Launch Outcome For All Sites



- The B5 booster version shows a perfect success record across various payload masses.
- While failures are scattered at lower masses, successes are consistently achieved even as payloads approach 10,000 kg.

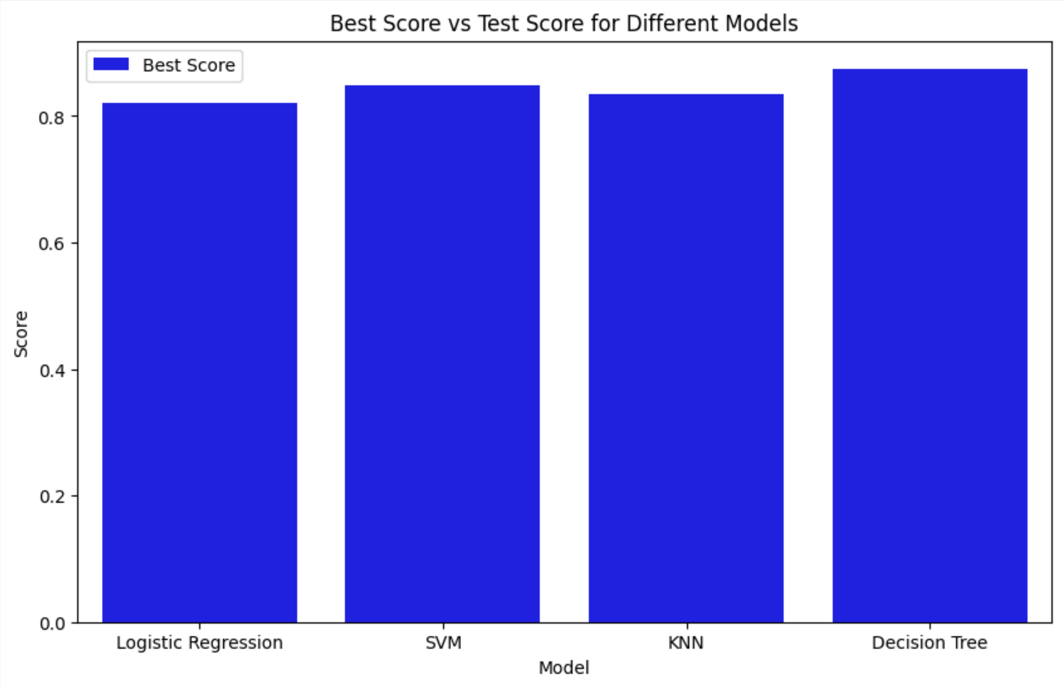
Section 5

Predictive Analysis (Classification)

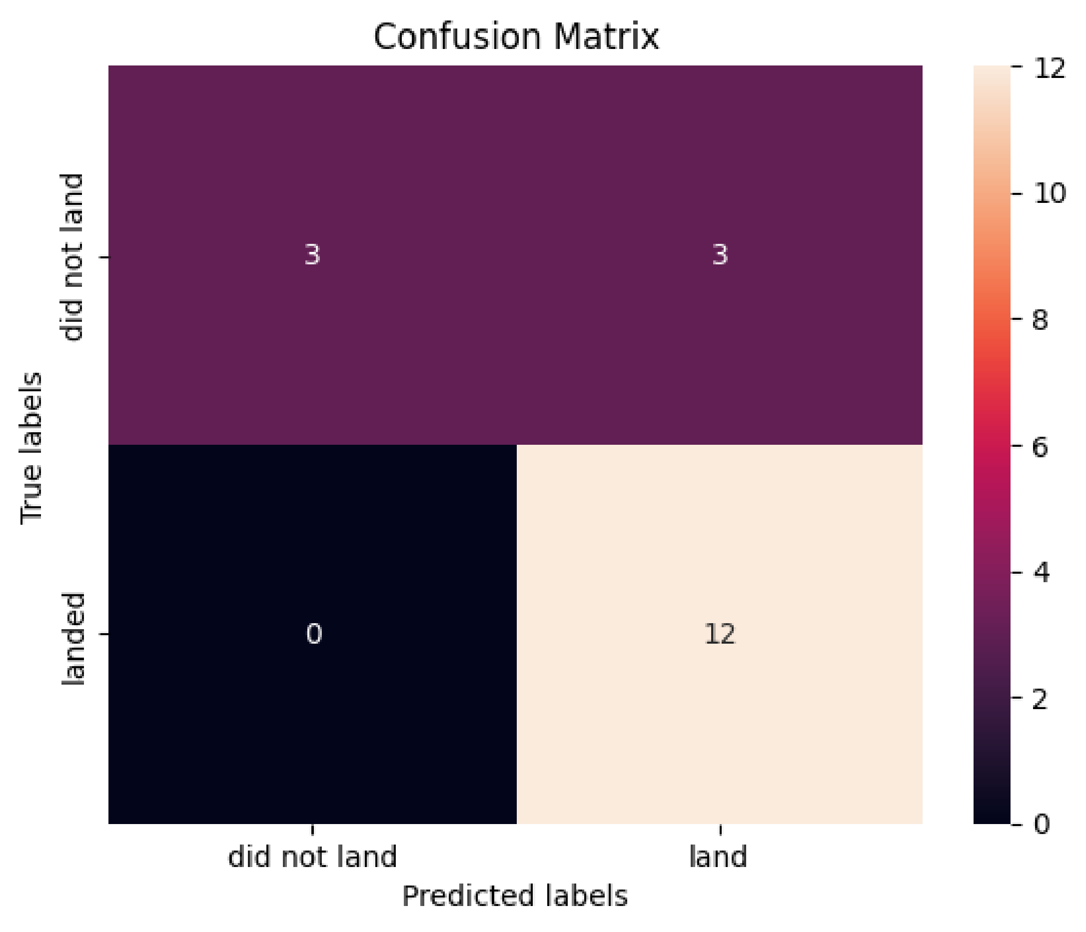
Classification Accuracy

- **Consistency in Testing**: The Logistic Regression, SVM, and KNN models achieved identical accuracy of 83.33% on the test dataset.
- **Best Overall Model**: The Decision Tree model scored highest during training (87.5%), but its accuracy in testing dropped to 72.22%, suggesting slight overfitting.
- **The Winner**: Based on generalizability, SVM emerges as the most robust model.

Model	Best Score	Test Score
Logistic Regression	0.821429	0.833333
SVM	0.848214	0.833333
KNN	0.833929	0.833333
Decision Tree	0.875000	0.722222



Confusion Matrix



- The model is highly effective at identifying successful landings. It correctly predicted 12 successful landings (True Positives) with zero instances of predicting a failure when it succeeded.
- The model correctly identified 3 actual failures.
- Model Error: The only area of error occurred when the model predicted a "landed" status for 3 missions that failed.

Conclusions

- SpaceX has demonstrated a significant learning curve, with landing success rates increasing from 0% in 2013 to over 80% by 2020.
- KSC LC-39A is the most successful launch site with a success rate of 76.9%.
- High-value orbits such as ES-L1, GEO, HEO, and SSO achieved a perfect 100% success rate, proving the system's reliability for specialized missions.
- Machine learning models (specifically SVM, KNN, and Logistic Regression) successfully predict landing outcomes with an accuracy of 83.33% on unseen data.
- The SVM model is exceptionally reliable at confirming successes, which is crucial for operational planning.
- By correctly predicting 15 out of 18 test cases, the model achieved an overall accuracy of 83.33%.

Appendix

EDA using SQL

```python
%load_ext sql

import csv, sqlite3
import prettytable
prettytable.DEFAULT = 'DEFAULT'

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

[2] Python

```python
%sql sqlite:///my_data1.db
```

[3] Python

```python
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

[4] Python

```
101
```

```python
#DROP THE TABLE IF EXISTS

%sql DROP TABLE IF EXISTS SPACEXTABLE;
```

[5] Python

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

[]
```

```python
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null;
```

[6] Python

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

[]
```

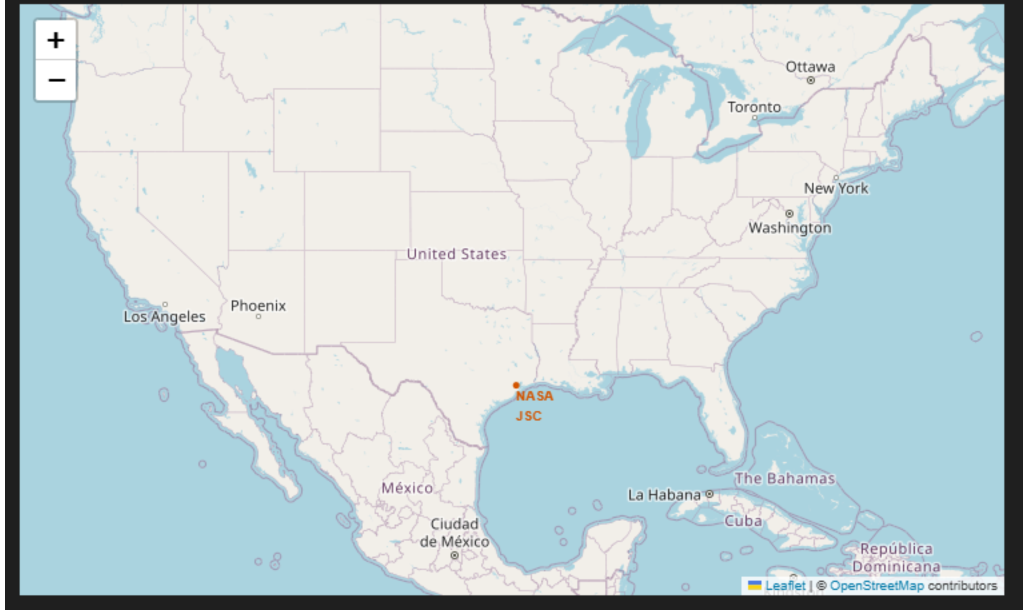
Appendix

```python
# Start location is NASA Johnson Space Center
nasa_coordinate = [29.559684888503615, -95.0830971930759]
site_map = folium.Map(location=nasa_coordinate, zoom_start=10)
```

Python

```python
# Create a blue circle at NASA Johnson Space Center's coordinate with a popup label showing
circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400', fill=True).add_chi
# Create a blue circle at NASA Johnson Space Center's coordinate with a icon showing its na
marker = folium.map.Marker(
    nasa_coordinate,
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'NASA JSC',
        )
    )
site_map.add_child(circle)
site_map.add_child(marker)
```

Python



Appendix

Support Vector Machine (SVM)

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

✓ 0.0s

```
svm_cv = GridSearchCV(estimator=svm, param_grid=parameters, cv=10)
svm_cv.fit(X_train, Y_train)
```

✓ 1.5s

Outputs are collapsed …

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

✓ 0.0s

```
tuned hpyerparameters :(best parameters)  {'C': np.float64(1.0), 'gamma': np.float
accuracy : 0.8482142857142858
```

```
svm_cv.score(X_test, Y_test)
```

✓ 0.0s

```
0.8333333333333334
```

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

✓ 0.0s

Thank you!