Wining the Space Race With Data Science

Gregory L. Gunther

IBM Data Science Professional Certification Capstone

2025

Outline

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

NEXT

Executive Summary

1.

Results suggestion launches have a correlation with the outcome of the launches

2.

Decision Tree is the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

3.

Srd Executive Summary Element

NEXT

Introduction

- **Project Overview**
  Analyze SpaceX Falcon 9 rocket landing success patterns using machine learning techniques.
- **Business Problem**
  Predict first stage landing success to determine launch costs and competitive bidding strategies against SpaceX.
- **The Challenge**
  SpaceX Falcon 9: $62M per launch
  Other providers: $165M+ per launch
  Cost advantage from first stage reusability
  Need to predict landing success for competitive analysis
- **Project Goal**
  Develop a machine learning model to predict whether the Falcon 9 first stage will land successfully, enabling accurate cost estimation for competitive bidding.
- **Success Criteria**
  Create accurate predictive models using data science methodology including data collection, wrangling, exploratory analysis, visualization, and machine learning model development.

Section 1

Methodology

Methodology

Data Collection
- API Access and Web Scraping

Data Wrangling
- Cleaning and Processing

Exploratory Data Analysis
- Exploration

Visualization
- Infographics Interactive Dashboards

Modeling
- Algorithm Development

Evaluation
- Results and Insights

NEXT

Data Collection

SpaceX REST API

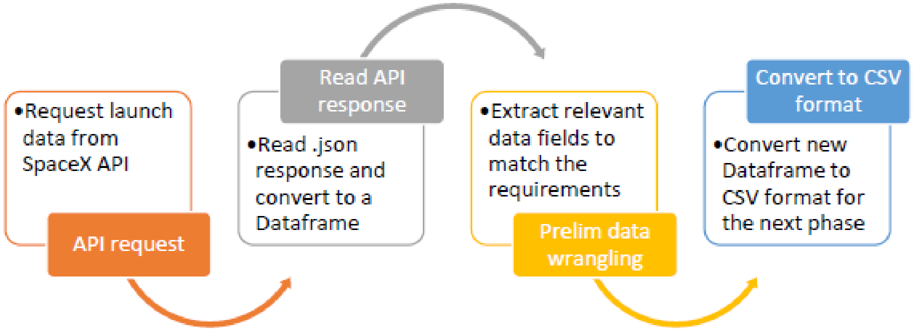
- Historical launch data (2010-2020)
- Rocket specifications & configurations
- Mission details & outcomes
- Landing success/failure records
- Launch site information

Web Scraping (Wikipedia)

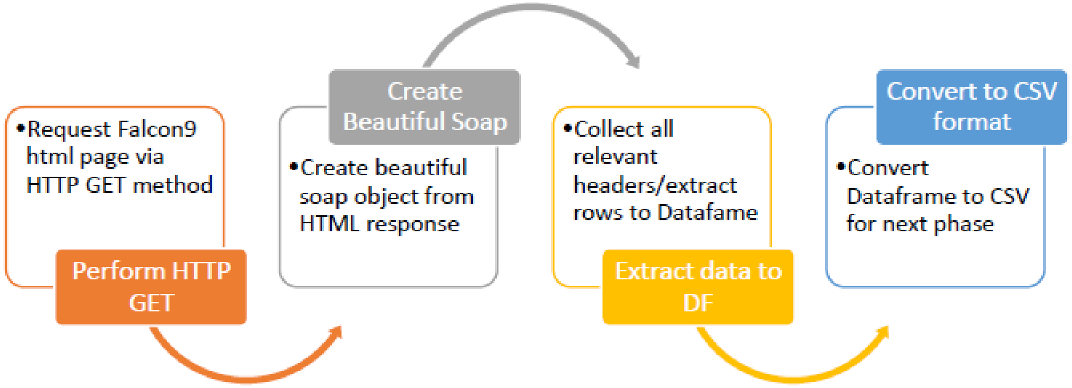
- Falcon 9 launch history tables
- Mission payload information
- Orbit type classifications
- Launch site details
- Additional mission context

NEXT

SpaceX API



Web scraping data from Wiki



Graphic Source: [IBM-AppliedDataScience-Capstone-FINAL/FINAL_FINDINGS_IBM-DataScience-Professional-Cert_Applied_Capstone_Project.pdf at master · vikthak/IBM-AppliedDataScience-Capstone-FINAL · GitHub]

Data Collection Using SpaceX API

```python
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
data.head()
```

The Good Stuff!

[Notebook](#)

[Output Dataset](#)

Output

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9

Cleaning Things Up!

```python
# Calculate the mean value of PayloadMass column

payload_mean = data_falcon9['PayloadMass'].mean()

# Replace NaN values with the mean
data_falcon9.loc[:, 'PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mean)
```

NEXT

Web Scraping

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
th_elements = first_launch_table.find_all('th')
for th in th_elements:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

The Good Stuff!

```
    # Orbit
    if row[5].a:
        orbit = row[5].a.string
    else:
        orbit = row[5].get_text(strip=True)
    launch_dict['Orbit'].append(orbit)

    # Customer
    if row[6].a:
        customer = row[6].a.string
    else:
        customer = row[6].get_text(strip=True)
    launch_dict['Customer'].append(customer)

    # Launch outcome
    launch_outcome = list(row[7].strings)[0]
    launch_dict['Launch outcome'].append(launch_outcome)

    # Booster landing
    booster_landing = landing_status(row[8])
    launch_dict['Booster landing'].append(booster_landing)
```

Parsing

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            launch_dict['Flight No.'].append(flight_number)

            datatimelist=date_time(row[0])

            # Date value
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
```

[Notebook]

[Output Dataset]

NEXT

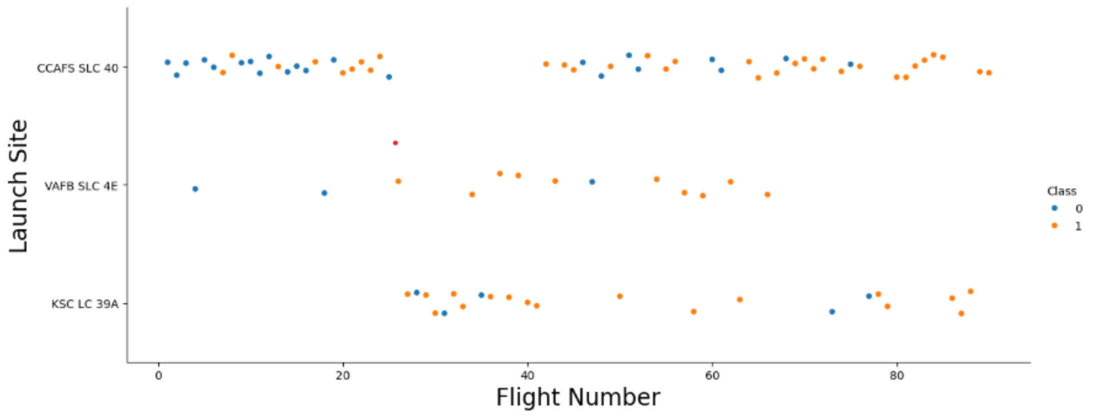
Data Wrangling

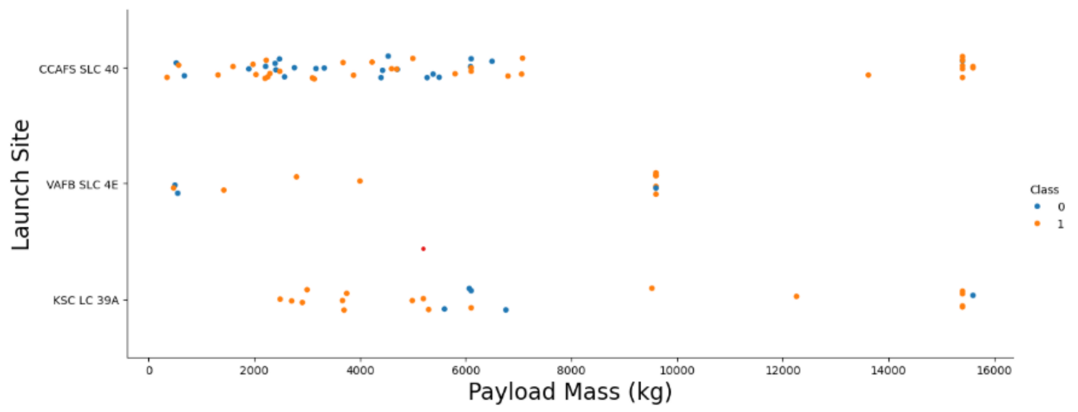
[Notebook](#)

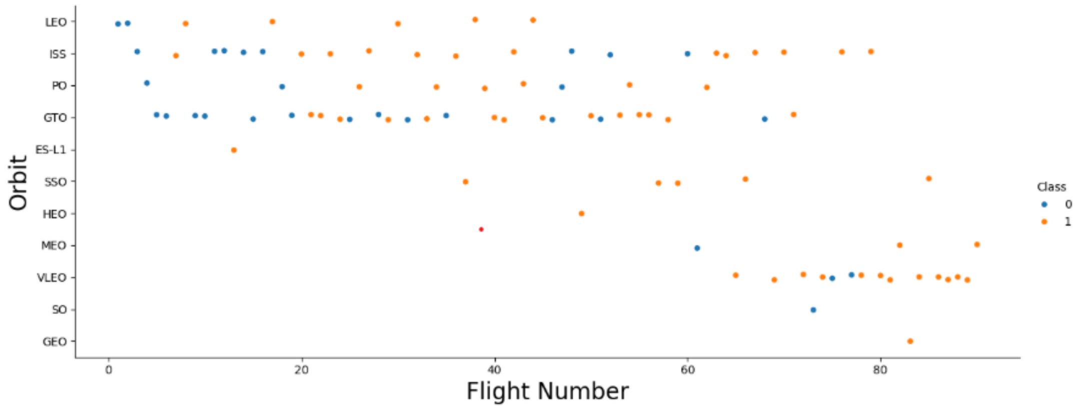
[Output Dataset](#)

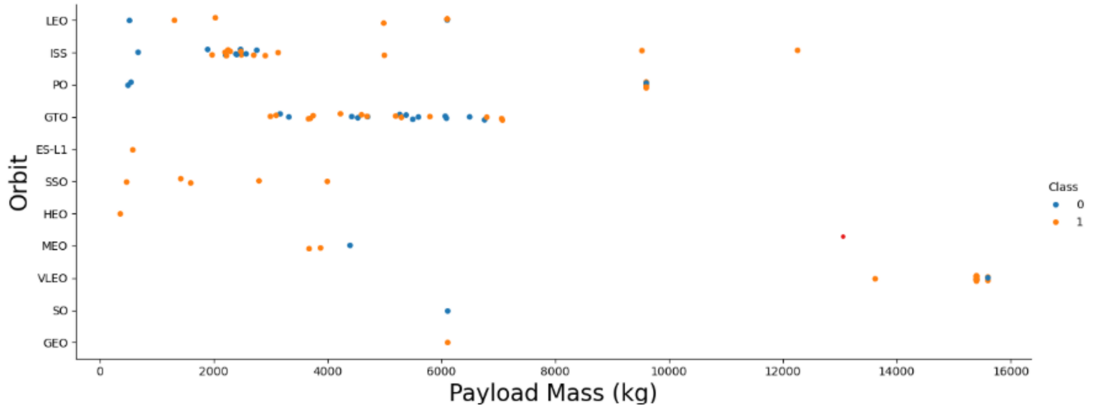
- **Data Quality Assessment**
  Analyzed SpaceX Falcon 9 dataset with 90+ launch records, identifying that only LandingPad had missing values (28.9% missing data)
- **Launch Site Analysis**
  Discovered CCAFS SLC 40 had the most launches (55), followed by KSC LC 39A (22) and VAFB SLC 4E (13), with GTO being the most common orbit type (27 missions)
- **Binary Classification Labels**
  Created training labels by converting mission outcomes into binary format - successful landings (True Ocean, True RTLS, True ASDS) = 1, unsuccessful landings (False outcomes and None outcomes) = 0
- **Success Rate Calculation**
  Determined overall landing success rate of 66.7% across all missions, providing baseline performance metric for predictive modeling
- **Dataset Preparation**
  Exported cleaned and labeled dataset for machine learning model development, with all features properly categorized as numerical or categorical variables

Visualization







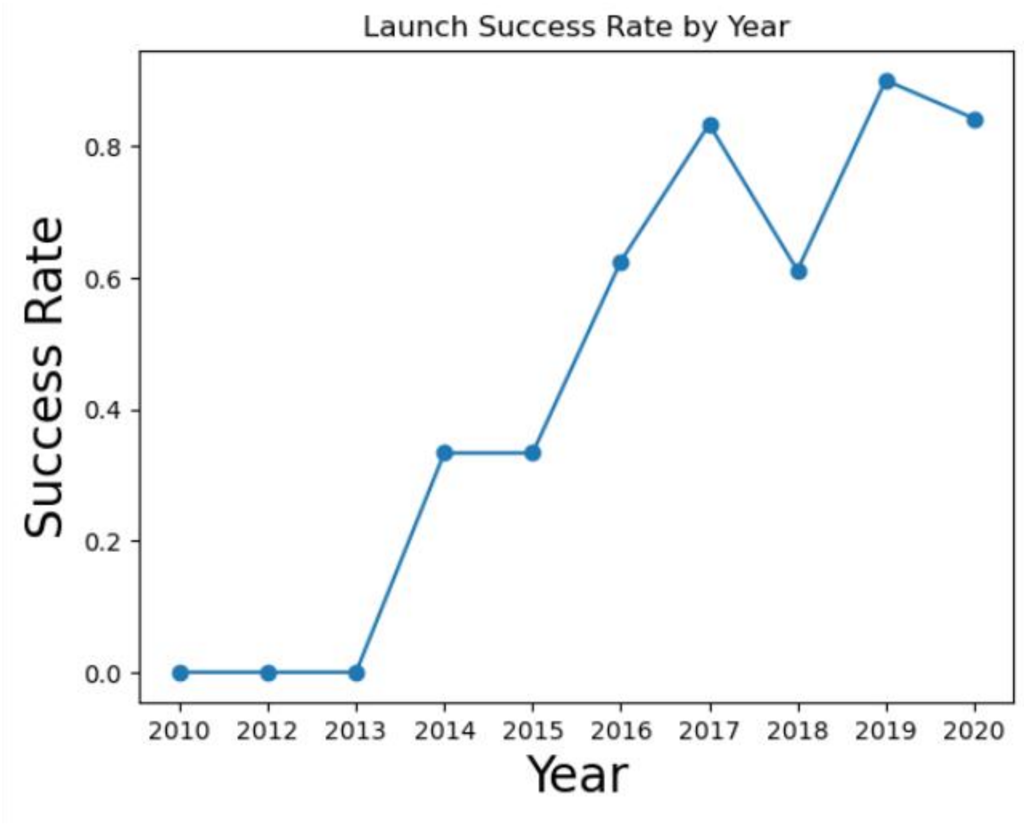


NEXT

Visualization

Notebook

Output Dataset



N E X T

Analysis Using SQL

[Notebook](#)

[Database](#)

```sql
%%sql
SELECT DISTINCT "Launch_Site"
FROM SPACEXTABLE;
```

Unique launch site

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

Launch sites with ‘CCA”

```sql
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS "Total_Payload_Mass"
FROM SPACEXTABLE
WHERE "Customer" LIKE '%CRS%';
```

Total payload mass carried by boosters launched by NASA (CRS)

```sql
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS "Average_Payload_Mass"
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

Average payload mass carried by booster version F9 v1.1

```sql
%%sql
SELECT MIN("Date") AS "First_Successful_Ground_Landing"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)'
```

Date when the first successful landing outcome

```sql
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
```

Names of boosters between 4 and 6K

```sql
%%sql
SELECT "Mission_Outcome", COUNT(*) AS "Count"
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

Mission success

```sql
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
```

Booster version

```sql
%%sql
SELECT
    SUBSTR("Date", 6, 2) AS "Month",
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE SUBSTR("Date", 0, 5) = '2015'
  AND "Landing_Outcome" = 'Failure (drone ship)';
```

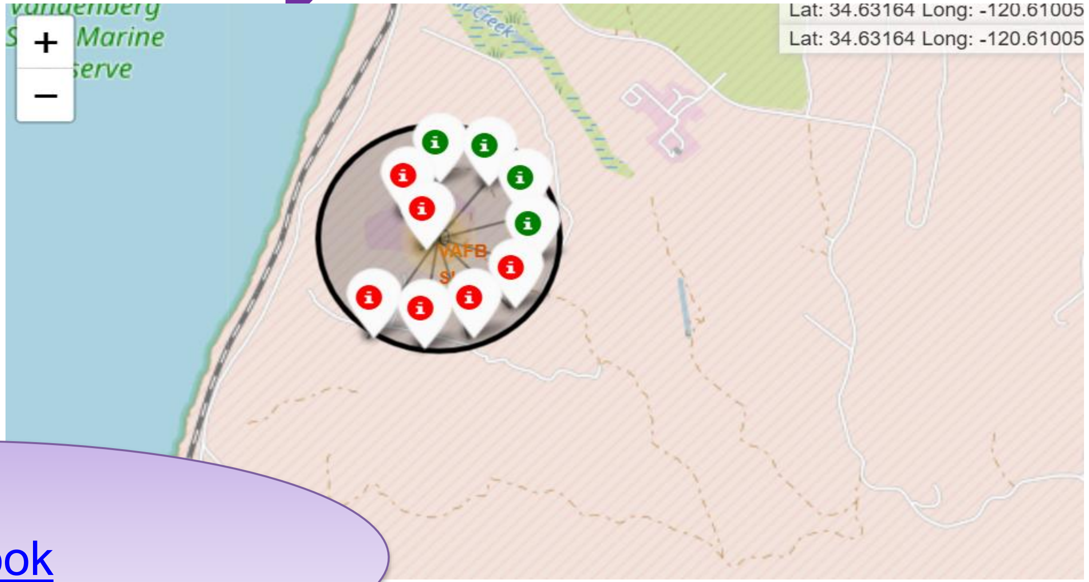
Proximaty Analysis

Site Location Mapping

Launch Success/Failure

Measuring Distances

[Notebook]



Cluster Markers

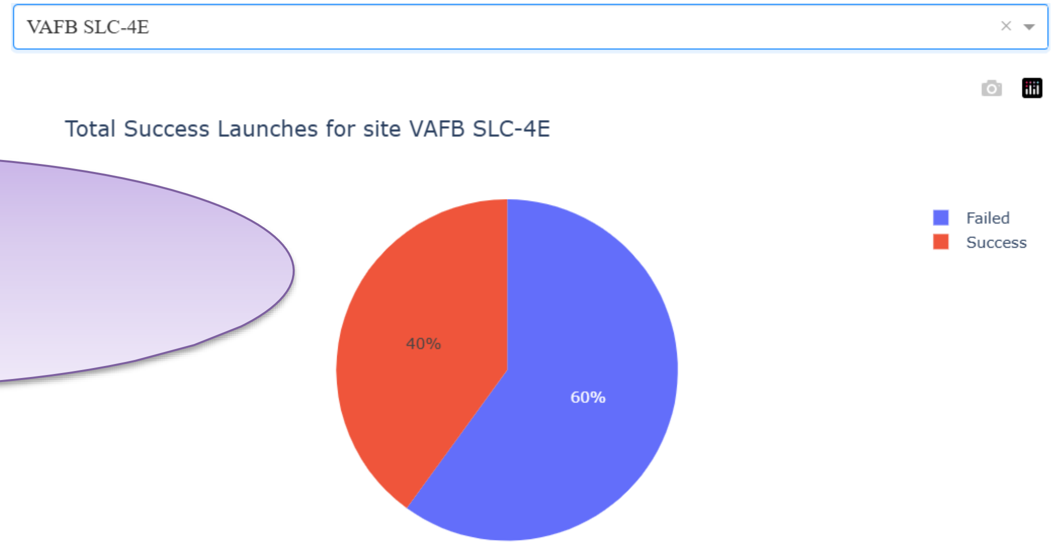
NEXT

- Real-time filtering capabilities
- Dynamic success analysis
- Launch site comparison

[Python]()

Launch Sites Visualization:

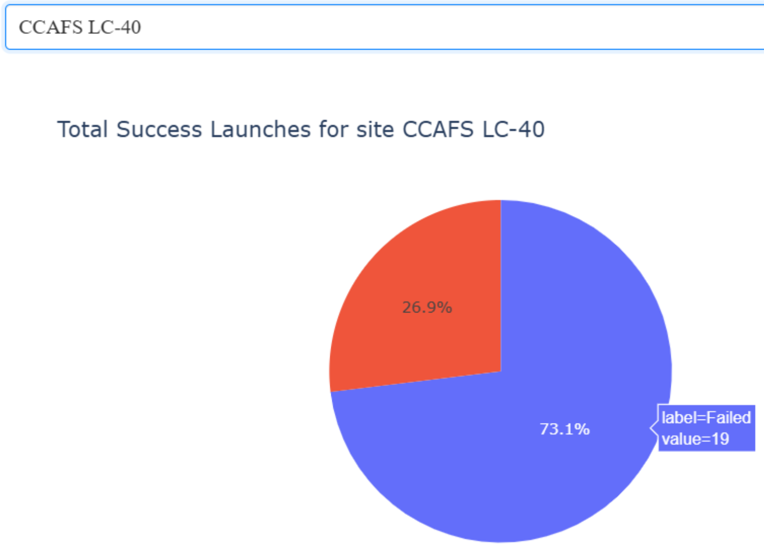
- *Kennedy Space Center (KSC) - Florida*
- *Cape Canaveral Air Force Station (CCAFS) - Florida*
- *Vandenberg Air Force Base (VAFB) - California*



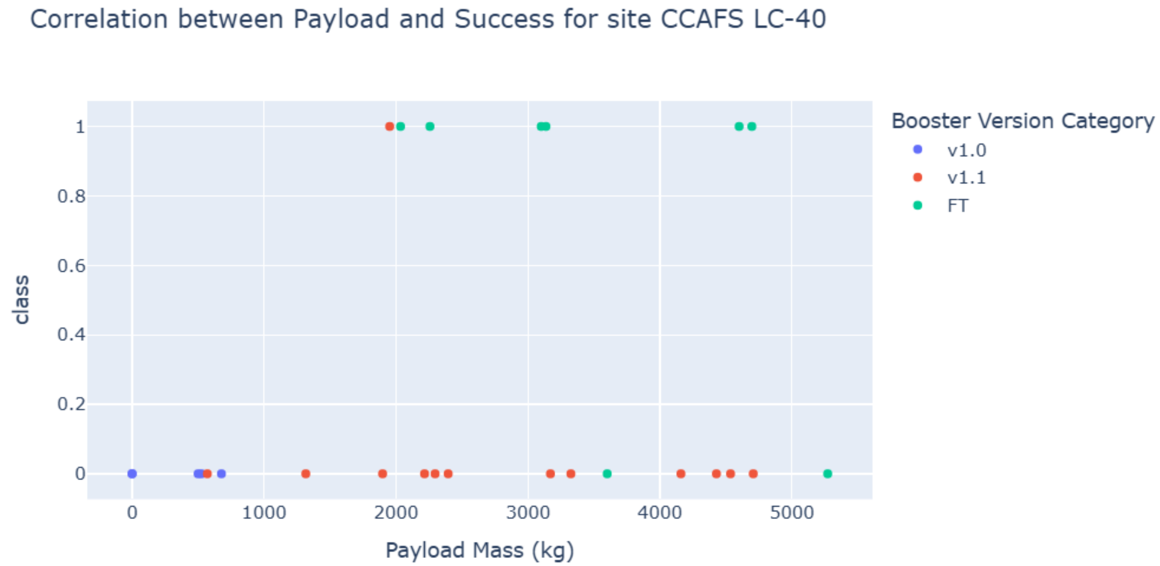
VAFB SLC-4E

Results Dashboard

NEXT



CCAFS LC-40



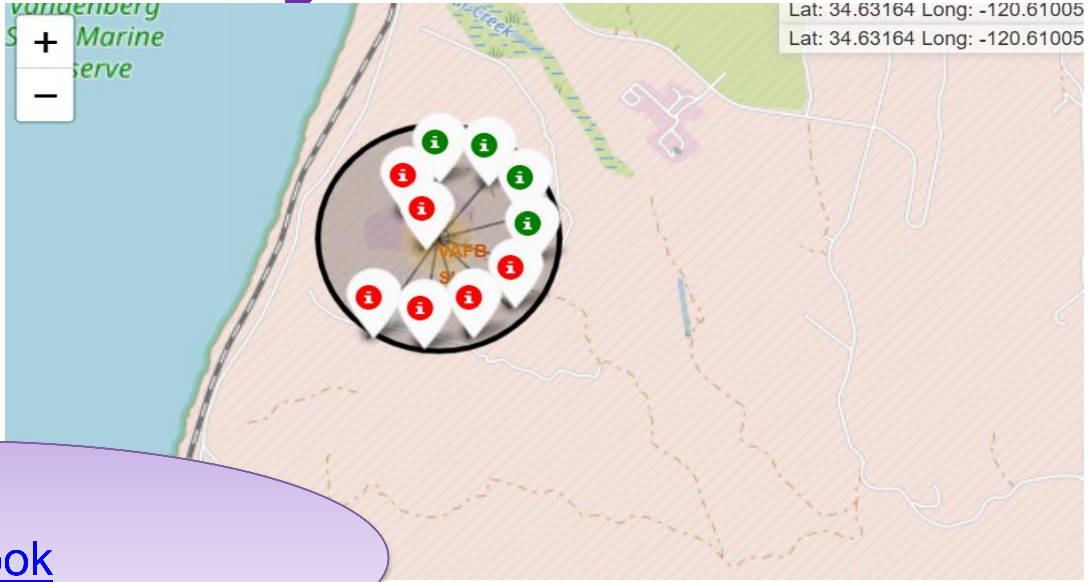
Correlation: Payload and Success—CCAFS LC-40

Proximaty Analysis

Site Location Mapping

Launch Success/Failure

Measuring Distances

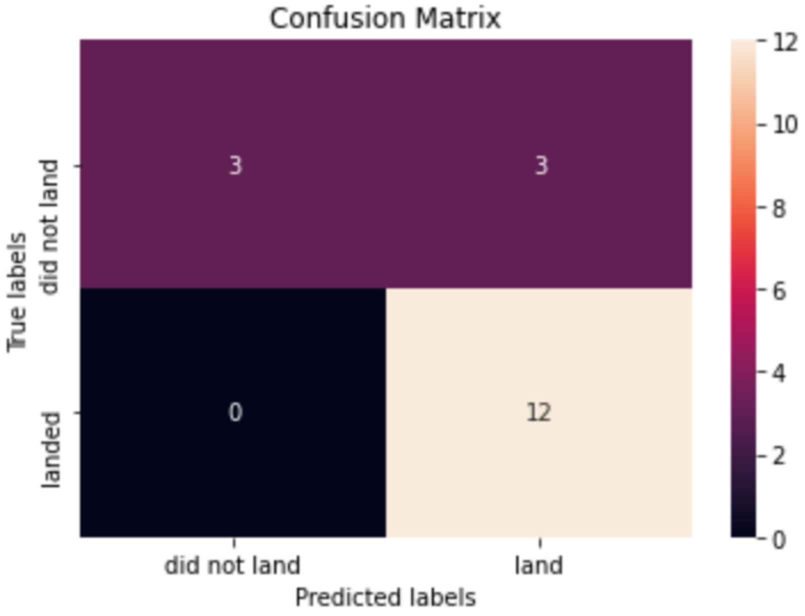


Cluster Markers

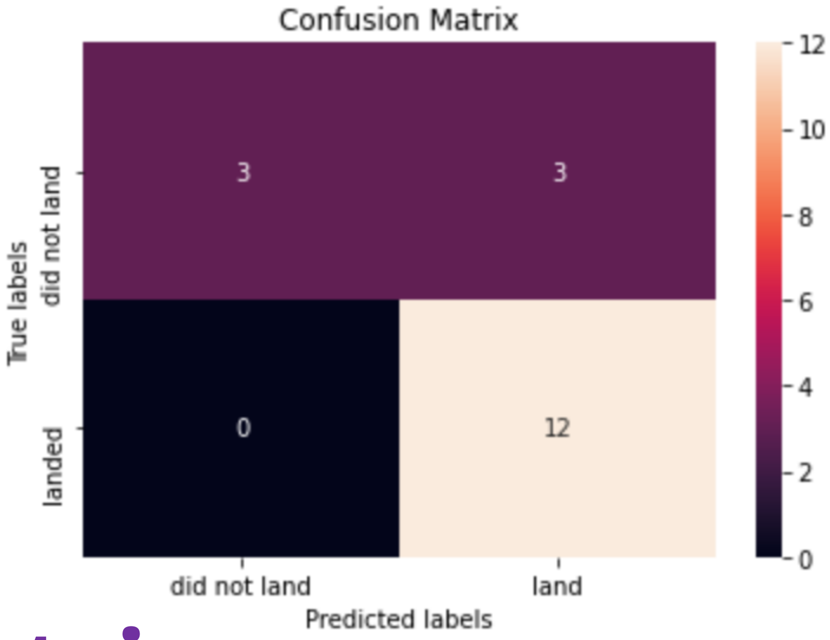
[Notebook]

N E X T

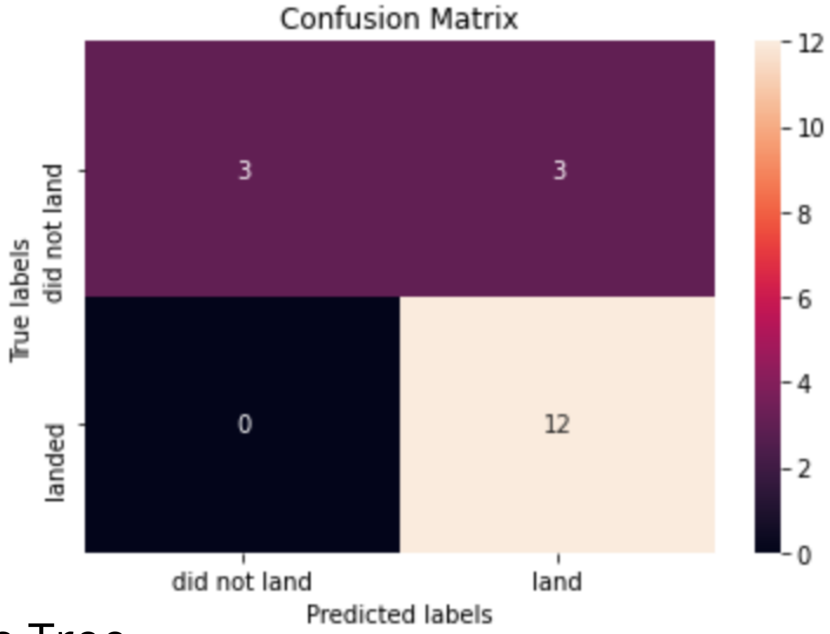
Confusion Matrices



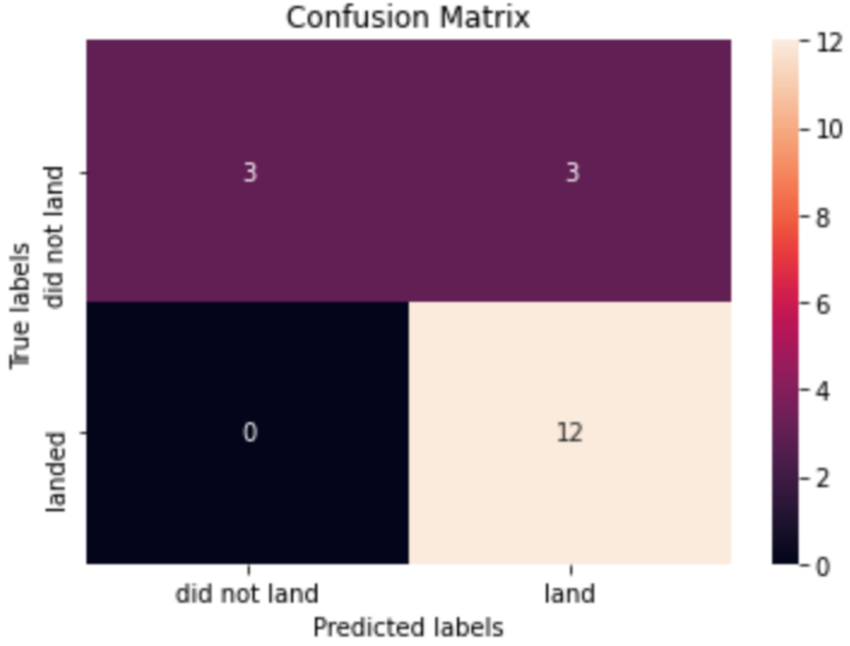
Logistic regression



SVM



Decision Tree



KNN

Predictive Analysis

Models Used

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree Classifier
- K-Nearest Neighbors (KNN)

Input Features

- Flight Number (experience proxy)
- Payload Mass (kg)
- Orbit Type (LEO, GTO, etc.)
- Launch Site (encoded)
- Grid Fins, Legs, Landing Pad

Training Process

- **Data Split:** 80% training, 20% testing
- **Preprocessing:** StandardScaler for feature normalization
- **Optimization:** GridSearchCV with 5-fold cross-validation
- **Evaluation:** Accuracy, Precision, Recall, F1-Score, AUC-ROC

NEXT

Model Performance Results

Notebook

83.3%	83.3%	80.0%	66.7%
Decision Tree Accuracy	SVM Accuracy	Logistic Regression	KNN Accuracy

🏆 Best Performing Models

- Decision Tree & SVM tied at 83.3%
- Excellent precision and recall balance
- Consistent cross-validation performance
- Low overfitting risk

📊 Detailed Metrics

- Precision: ~85%
- Recall: ~80%
- F1-Score: ~82%
- AUC-ROC: ~0.85

🎯 Model Selection Rationale

Decision Tree was selected as the final model due to its interpretability and robust performance. The 83.3% accuracy provides reliable predictions for business decision-making while maintaining model transparency.

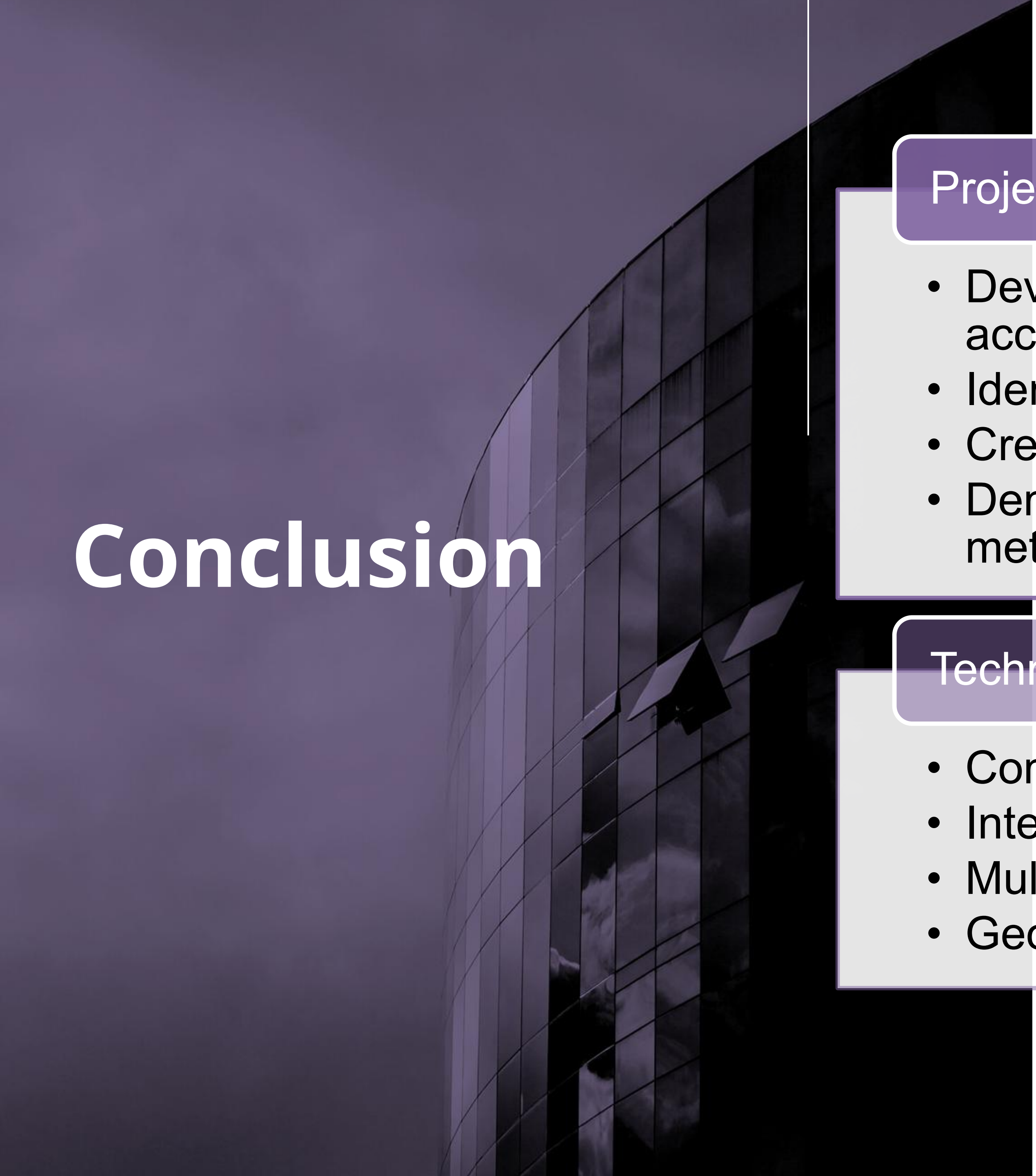
Key Insights

Success Factors Identified

- Experience matters: Success improves with flight number
- Heavy payloads correlate with higher success rates
- KSC LC-39A shows superior performance
- LEO missions have different patterns than GTO

Temporal

- Dramatic improvement from 2015-2020
- Clear organizational learning curve
- Recent missions achieve 80%+ success
- Technology advancement visible in data



Conclusion

Project Achievements

- Developed predictive models with 83% accuracy
- Identified critical success factors
- Created actionable business intelligence
- Demonstrated complete data science methodology

Technical Accomplishments

- Comprehensive data collection & processing
- Interactive visualizations & dashboards
- Multiple ML model evaluation & optimization
- Geospatial analysis with mapping

NEXT

Appendix

- Data Collection (SpaceX API)
  - Notebook
  - Output Dataset
- Web Scraping (Wikipedia)
  - Notebook
  - Output Dataset
- Data Wrangling
  - Notebook
  - Output Dataset
- Visualization
  - Notebook
  - Output Dataset
- SQL
  - Notebook
  - Database

- Proximity Analysis
  - Notebook
- Results Dashboard (Plotly Dash)
  - Python
- Predictive Modeling
  - Notebook

NEXT