
# Winning Space Race with Data Science

Iryna Kushnirova

12.2024

# Content

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

# Executive Summary

- **Summary of Methodologies**
  - To identify patterns in landing outcomes, the following approaches were utilized:
    - Data Collection:
    - Using API GET requests and Web Scraping
    - Data Wrangling
    - Exploratory Data Analisis: using SQL and Data Visualization
    - Interactive Visualization:  using Folium and Potly Dash
    - Machine Learning Prediction
- **Summary of Results**
  - Exploratory Data Analysis result
  - Visualization Insights in screenshots
  - Machine Learning Prediction Model that demonstrated the best performance across evaluation metrics

# Introduction

**Background**

- SpaceX promotes Falcon 9 rocket launches at a cost of $62 million, significantly lower than the $165 million or more charged by other providers. This cost reduction is largely due to the reusability of the rocket's first stage.
- By predicting whether the first stage will successfully land, we can estimate the potential cost of a launch.
- To achieve this, I leverage publicly available data and apply machine learning models.

**Explore**

- Analyze how payload mass, launch sites, orbit types, and flight numbers influence outcomes.
- Examine the rate of successful landings over time to identify trends.
- Identify the best predictive model for successful landings using binary classification.

# Methodology

Section 1

# Methodology

## Data Collection

API Utilization

WEB Scraping

## EDA using SQL

Execute SQL queries to understand the Spacex DataSet

## Interactive visual analytics

Folium

Plotly Dash

Dealing with Missing Values

Perform some EDA

Creates a landing outcome label from Outcome column

Determine the success rate

## Data Wrangling

Find relationships between data and outcame

Feature Engineering

## EDA using Visualization

Build

Tune

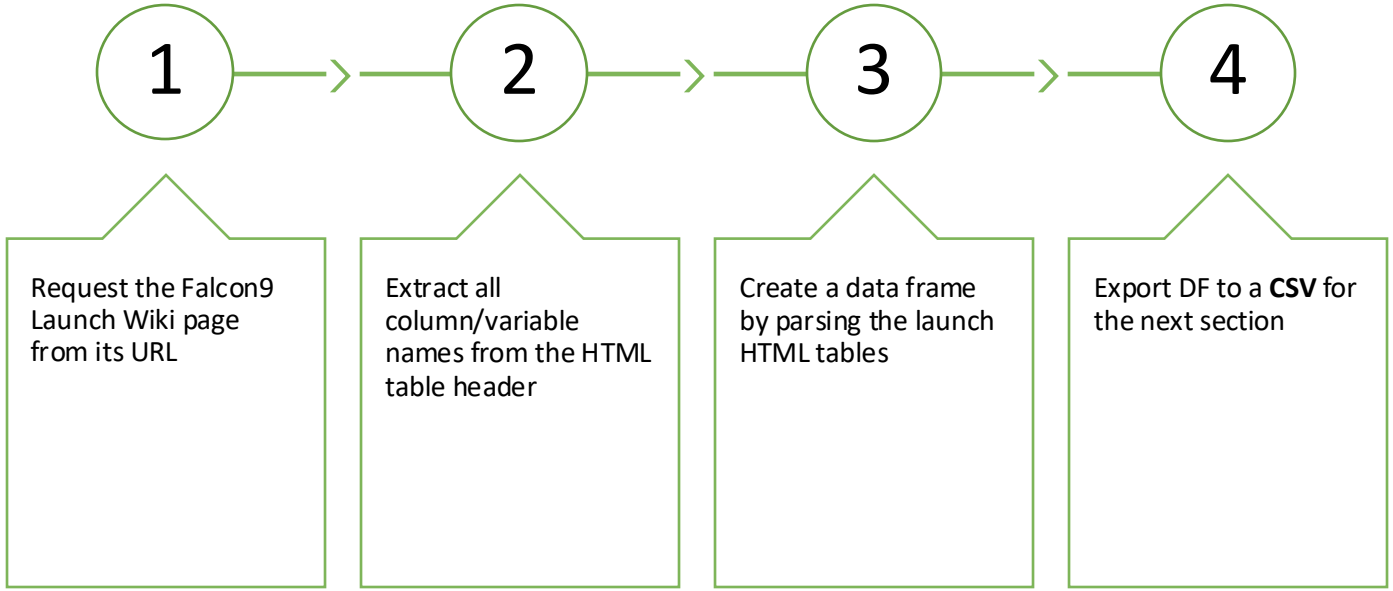
Evaluate

## Predictive analysis using classification models

# Data Collection – SpaceX API (GitHub URL)

- Request and parse the SpaceX launch data using the GET request
- Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
- Request information about the launches from SpaceX API using custom functions
- Create dictionary from the data
- Create dataframe from the dictionary
- Filter the dataframe to only include Falcon 9 launches

# Data Collection – **Web Scraping** (GitHub URL)



# Data Wrangling (GitHub URL)

Dealing with Missing *Values*

- *Replace the np.nan values with its mean value using the .mean() and .replace() functions for PayloadMass*

Perform some Exploratory Data Analysis :

- # of launches on each site;
- # of each orbit;
- # of mission outcome of the orbits

Creates a landing outcome label from Outcome column

- *A list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. Then assign it to the variable landing_class:*

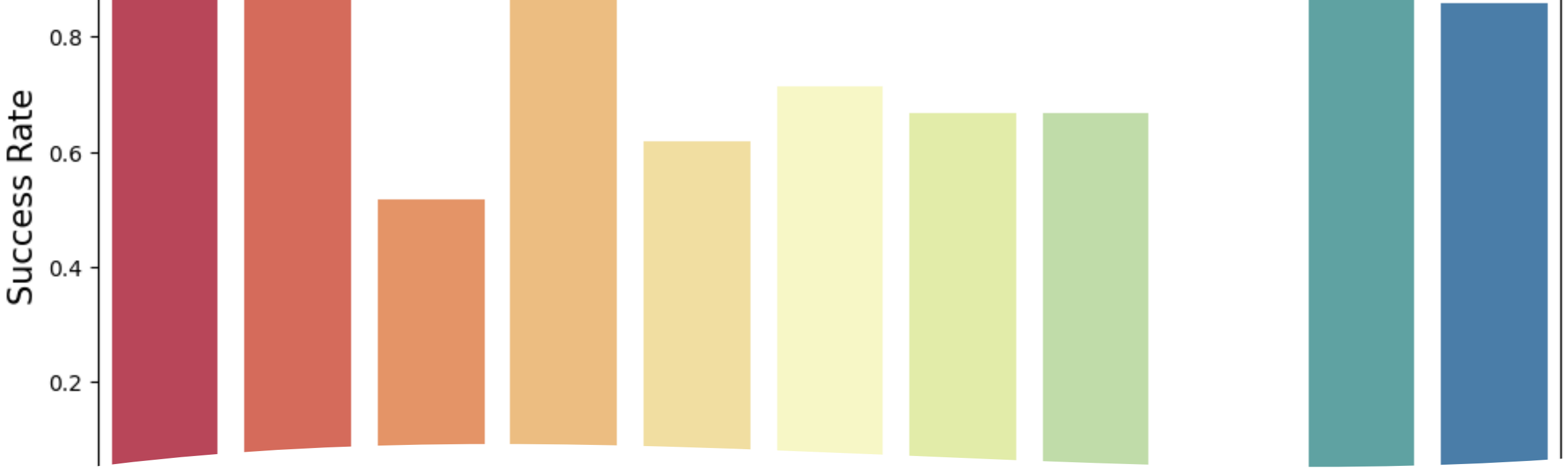
Determine the success rate

- code to determine the success rate: *df["Class"].mean()*
  *df["Class"].value_counts()*

# EDA with SQL (GitHub URL)

SQL queries to solve the assignment tasks

- Display the names of the unique launch sites in the space mission
- Find the total payload mass
- Find the average payload mass
- List the total # of successful and failure  outcomes
- List the names of the booster_versions which have carried the maximum payload mass using a subquery
- Rank the count of landing outcomes




# EDA using Data Visualization (GitHub URL)

EDA and Feature Engineering using Pandas and Matplotlib

- Explore the relationships between:
  - Flight Number and Launch Site
  - Payload Mass and Launch Site
  - Success rate of each orbit type
  - Flight Number and Orbit type
  - Payload Mass and Orbit type
  - The launch success yearly trend
- Create dummy variables to categorical columns
- Cast all numeric columns to float64

# Build an Interactive Map with Folium (URL)

### Mark all launch sites on a map

All launch sites are in proximity to the Equator line

All launch sites in very close proximity to the coast

### Mark the success/failed launches for each site on the map

From the color-labeled markers in marker clusters, it is easily identify which launch sites have relatively high success rates

### Calculate the distances between a launch site to its proximities

Launch sites are close to railways

Launch sites are close to highways

Launch sites Are near coastline

Launch sites keep a 16.27km distance away from cities

| | City | Coast | Highway | Railway |
|---|---|---|---|---|
| **Mean** | 25.57245228 | 1.883154559 | 4.259669479 | 1.131817113 |
| **MIN** | 16.26841382 | 0.959590734 | 0.5850337 | 0.70512059 |
| **MAX** | 39.69652866 | 3.987148927 | 14.96106243 | 1.292136884 |

# Build a Dashboard with Plotly Dash (GitHub URL)

- **Interactive Dashboard Features**
- The dashboard application enables user interaction through:
  - **Input Components:**
    - A **dropdown list** for selecting categories or filters.
    - A **range slider** for adjusting numerical values, such as payload range.
  - **Visualization Tools:**
    - A **pie chart** to display proportional data (e.g., success rates per launch site).
    - A **scatter point chart** for exploring correlations (e.g., payload vs. success rate).

- **Insights Derived from the Dashboard**
- Through visual analysis, the following questions can be answered:
  - **Which site has the largest number of successful launches?**
  - **Which site has the highest launch success rate?**
  - **Which payload range(s) have the highest launch success rate?**
  - **Which payload range(s) have the lowest launch success rate?**
  - **Which Falcon 9 booster version (e.g., v1.0, v1.1, FT, B4, B5) has the highest launch success rate?**

# Predictive Analysis (Classification - GitHub URL)

- To build the predictive classification model, the following steps were taken:
- Data Preparation:
  - Created a new column to define the classification target (e.g., success/failure).
  - Standardized the data to normalize feature values.
  - Split the dataset into training and testing subsets.
- Model Optimization using GridSeach:
  - Tuned hyperparameters for the following models:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Classification Tree
    - K-Nearest Neighbors
- Model Evaluation:
  - Assessed performance using the test data to identify the best-performing model.
  - Evaluation Metrics:
  - The following metrics were used to comprehensively evaluate model performance:
  - ROC AUC: Measures classification performance across different thresholds.
  - Precision, Recall, F1 Score: Provides insights into the trade-offs between precision and completeness.
  - Log Loss: Evaluates model predictions based on probabilities.
  - Accuracy: Assesses overall classification correctness.

# Results

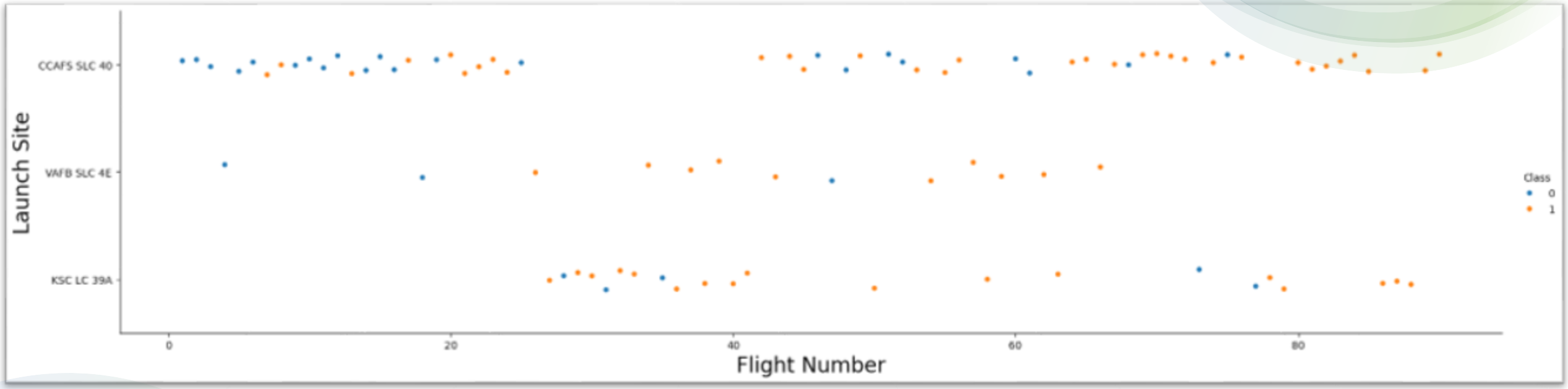
- **Exploratory Data Analysis Results**
  - Orbit Success Rates:
    - SSO Orbit: 100% success rate
    - For VLEO, LEO, ISS, MEO Orbits: 60-90% success rate.
    - LEO, ISS, PO Orbits: Success correlates with higher flight numbers and greater payloads.
    - GTO Orbit: Success remains unpredictable.
  - Trends and Sites:
    - Success rates increased steadily from 2013 to 2020.
    - KSC LC-39A has the highest success rate.
  - Booster Version:
    - FT Booster: Highest success rate overall.
    - B4 Booster: Performs best at KSC LC-39A.
- **Visual Insights**
  - All launch sites are located near the Equator and along coastal areas.
  - They are strategically placed far from urban areas, highways, and railways to minimize risks from potential launch failures, while still being close enough for logistical support.
- **Predictive Analysis Results**
  - The Decision Tree Classifier is the best-performing model for predicting launch outcomes.

# Insights drawn ADE

Section 3

1 class – success
0 class - failure

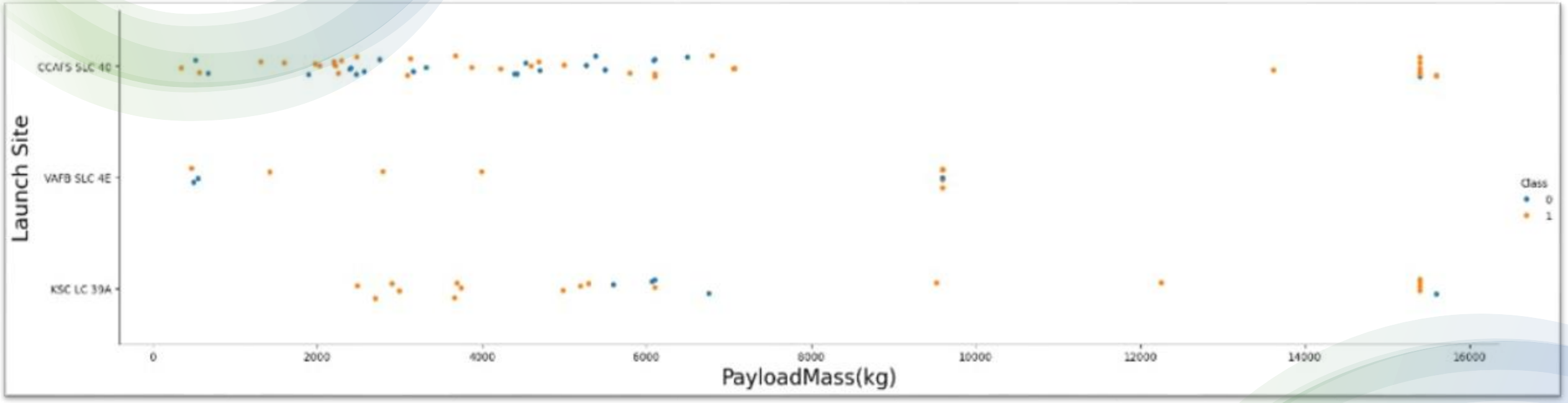
# Flight Number vs. Launch Site




- Success increase with flight number
- Launch Site KSC LC 39A has highest success rate

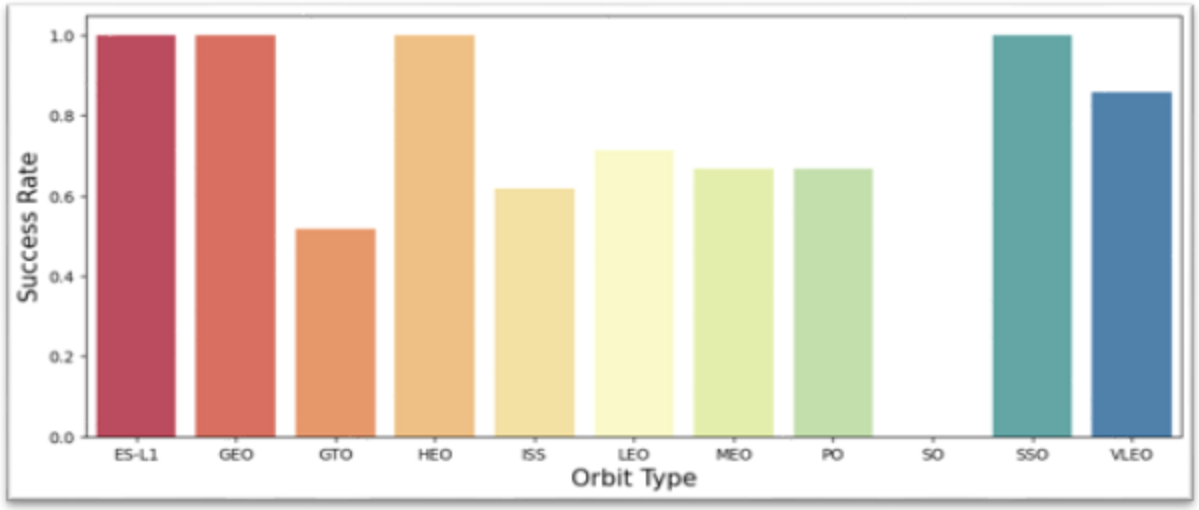
1 class – success
0 class - failure

# Payload vs. Launch Site

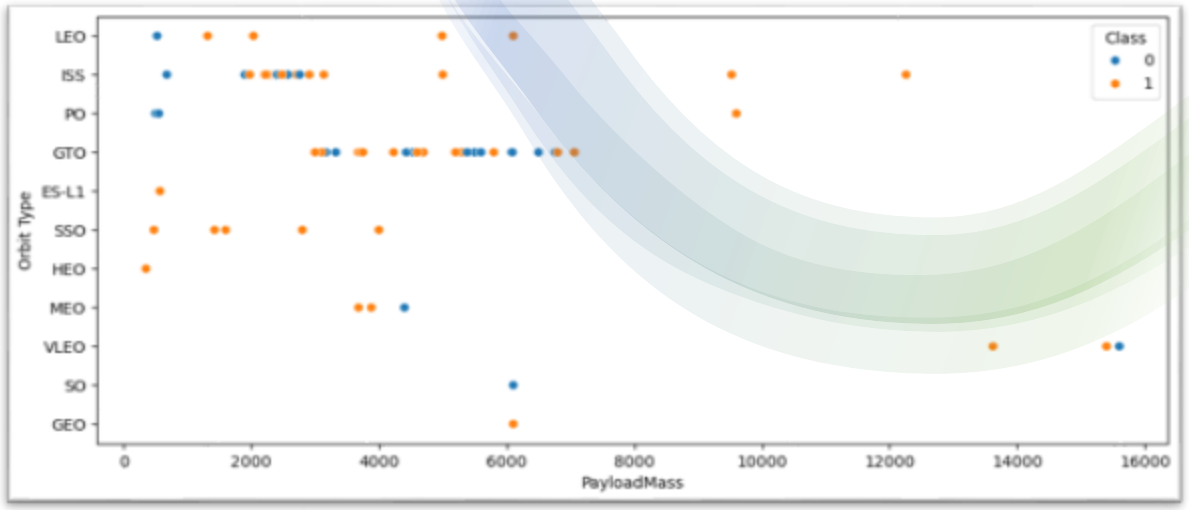



- Success rate shows a **positive correlation** with payload mass across launch sites.
- The **highest success rates** are observed for payloads >6000 kg.
- **VAFB SLC-4E** has not been used for payloads >9000 kg.

# Success Rate vs. Orbit Type

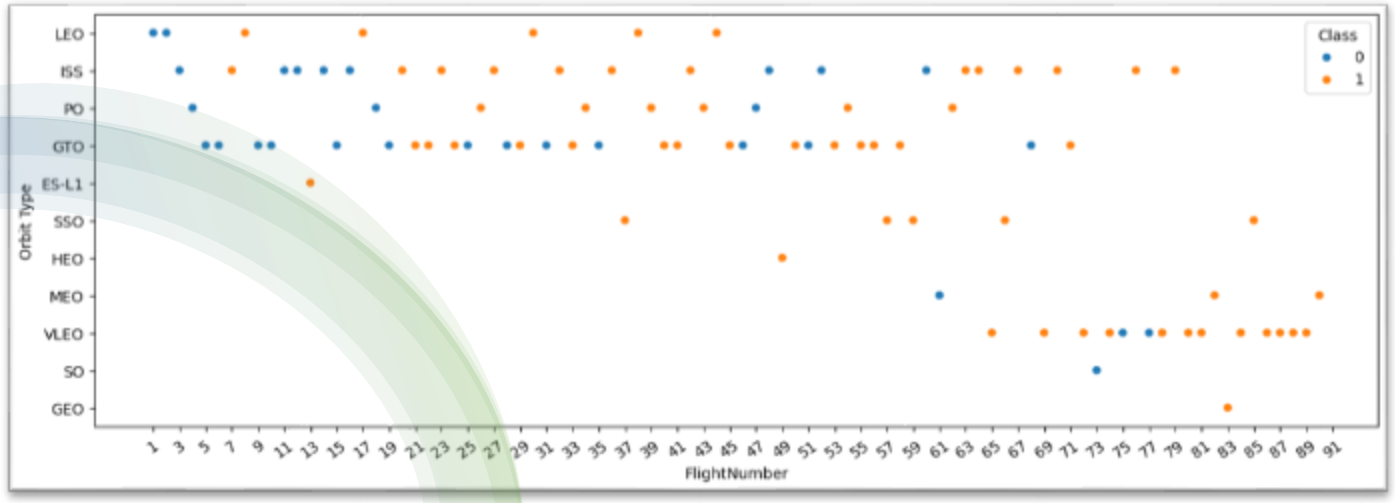



# Payload vs. Orbit Type




1 class – success
0 class - failure

# Flight Number vs. Orbit Type



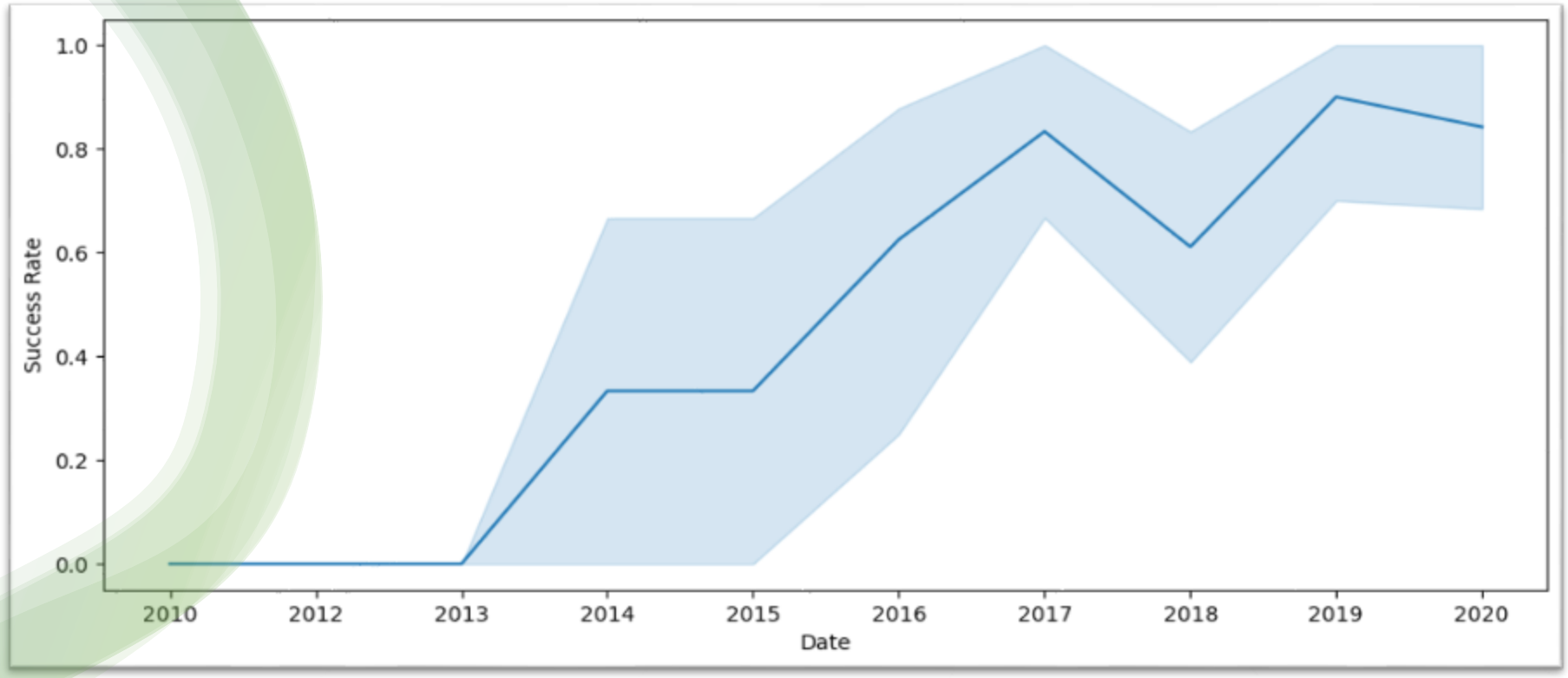

- ✓ For SSO Orbit: **100% success rate**
- ✓ For VLEO, LEO, ISS, MEO Orbits: **60-90% success rate**

- The SSO Orbitt excellent with payload mass <= 4000kg
- For LEO, ISS, PO Orbits: Success correlates with **higher flight numbers and greater payloads** (>4000kg).

- For ES-L1, GEO, HEO Orbits: 100% success rate but with **limited flight data.**
- For **SO** Orbit unsuccess for the first flight
- For GTO Orbit success remains **unpredictable**.

# Launch Success Yearly Trend

- **Improved**: 2013–2017 and 2018–2019.
- **Decreased**: 2017–2018 and 2019–2020.
- **Overall**: Success rate has increased since 2013.



# Launch Site Information

```
%sql SELECT DISTINCT (Launch_Site)\
    FROM SPACEXTABLE
```

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

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

*SQL query to display the names of the unique launch sites in the space mission*

- *The SELECT DISTINCT statement is used to return only distinct (different) values*

Launch Site Names

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

## Launch Site Names Begin with 'CCA'

*SQL queriy to display 5 records where launch sites begin with the string 'CCA'*

- *The WHERE clause is used to filter records.*
- *The operator LIKE to search for a pattern*
- *The percent sign % represents zero, one, or multiple characters*

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

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

| 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 |

# Payload Mass

- The total payload carried by boosters from NASA - **45596 kg**
- The average payload mass carried by booster version F9 v1.1 - **2928.4 kg**

*SQL query to display Total Payload Mass*

- *The SUM() function returns the total sum of a numeric column.*
- *The WHERE clause is used to filter records.*
- *The operator = to search for a pattern that equal "NASA (CRS)"*

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_)
AS TOTAL_PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE Customer  = "NASA (CRS)"
```

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

| TOTAL_PAYLOAD_MASS__KG_ |
|---|
| 45596 |

*SQL query to display the Average Payload Mass by F9 v1.1*

- *The AVG() function returns the average value of a numeric column.*
- *The WHERE clause is used to filter records.*
- *The operator = to search for a pattern that equal "NASA (CRS)"*

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

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

| AVARAGE_PAYLOAD_MASS__KG_ |
|---|
| 2928.4 |

# First Successful Ground Landing Date

*SQL query to display the First Successful Ground Landing Date*

- *The MIN() function returns the smallest value of the selected column.*
- *The WHERE clause is used to filter records.*
- *The operator = to search for a pattern that equal "Success (ground pad)"*

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

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

| First_succesful_landing |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT Booster_Version, Payload, Landing_Outcome, PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
ORDER BY  PAYLOAD_MASS__KG_ DESC
```

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

| Booster_Version | Payload | Landing_Outcome | PAYLOAD_MASS__KG_ |
|---|---|---|---|
| F9 FT B1021.2 | SES-10 | Success (drone ship) | 5300 |
| F9 FT B1031.2 | SES-11 / EchoStar 105 | Success (drone ship) | 5200 |
| F9 FT B1022 | JCSAT-14 | Success (drone ship) | 4696 |
| F9 FT B1026 | JCSAT-16 | Success (drone ship) | 4600 |

*SQL query to display the The boosters names which have success in drone ship and have payload mass greater than 4000 but less than 6000*

- *The WHERE clause is used to filter records.*
- *The operator BETWEEN to search for a pattern that between a certain range*
- *The ORDER BY keyword is used to sort the result-set in descending order.*

# Total Number of Successful and Failure Mission Outcomes

- 100 Success (one of them have payload status unclear
- 1 Failure in flight

```
%%sql SELECT
COUNT(Mission_Outcome) AS "# OF Success"
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE "Success%"
```

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

| # OF Success |
|---|
| 100 |

```
%%sql SELECT
COUNT(Mission_Outcome) AS "# OF Failure"
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE "Failure%"
```

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

| # OF Failure |
|---|
| 1 |

*SQL query to display Total Number of Successful and Failure Mission Outcomes*

- *The COUNT() function returns the number of rows that matches a specified criterion.*
- *The WHERE clause is used to filter records.*
- *The operator LIKE to search for a pattern*
- *The percent sign % represents zero, one, or multiple characters*
- *The GROUP BY statement groups rows that have the same values into summary rows*

```
%%sql SELECT Mission_Outcome,
COUNT(Mission_Outcome) AS "#"
FROM SPACEXTABLE
group by Mission_Outcome
```

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

| Mission_Outcome | # |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# The names of the booster_versions which have carried the maximum payload mass

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

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

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

*SQL query to display the names of the booster_versions which have carried the maximum payload mass using subquery*

- *The WHERE clause is used to filter records*
- *The MAX() function returns the largest value of the selected column.*

List the names of the booster_versions

- 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

# The month , failure landing_outcomes in drone ship ,booster versions, launch_site in year 2015

*SQL query to display the names of the month , failure landing_outcomes in drone ship ,booster versions, launch_site in year 2015*

- *The WHERE clause is used to filter records*
- *The operator = to search for a pattern*

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

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

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

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

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

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

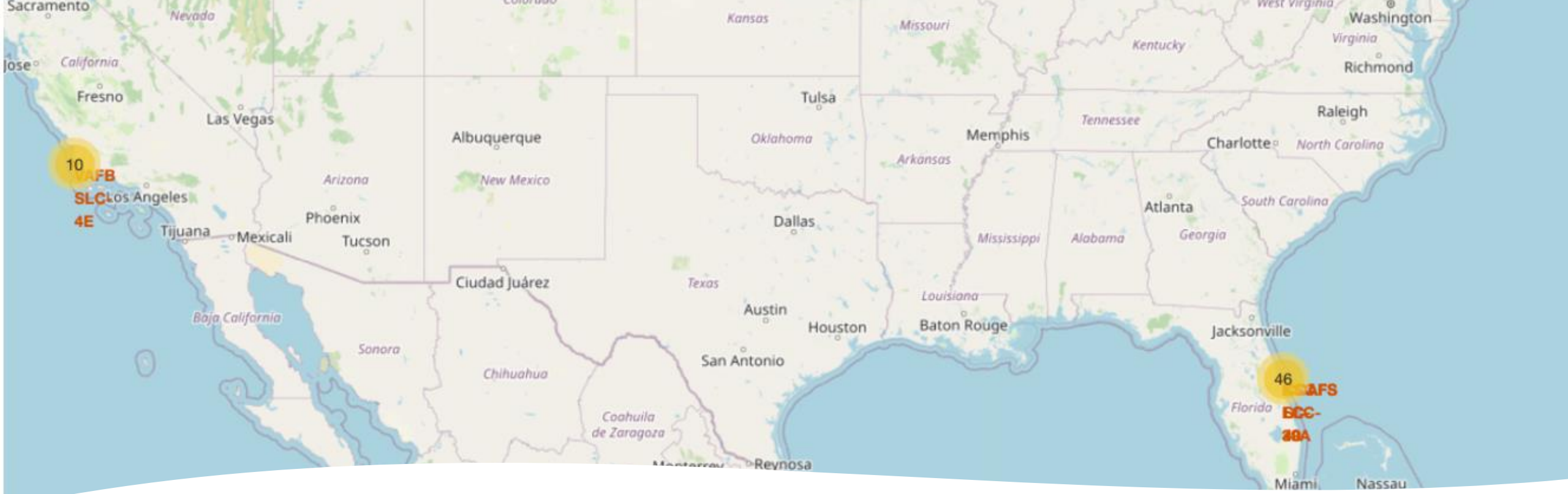
| Landing_Outcome | # |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

*SQL query to displayRank Landing Outcomes Between 2010-06-04 and 2017-03-20*

- *The COUNT() function returns the number of rows that matches a specified criterion.*
- *The WHERE clause is used to filter records.*
- *The operator BETWEEN to search for a pattern that between a certain range*
- *The GROUP BY statement groups rows that have the same values into summary rows*
- *The ORDER BY keyword is used to sort the result-set in descending order*

# Launche Sites Proximities Analisis

Section 3



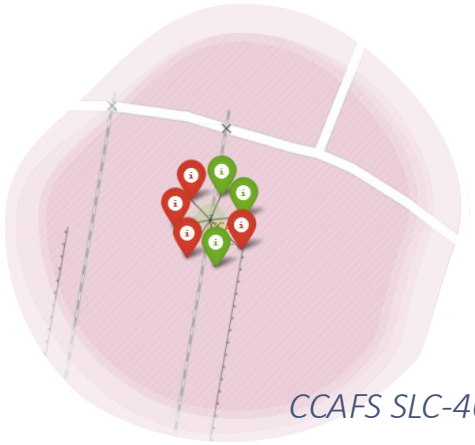

# Map with marked launch sites

- All launch sites are situated near the Equator line, which helps optimize orbital launches.
- Sites are located near the coast to ensure safe launch and landing trajectories, minimizing risks to populated areas.

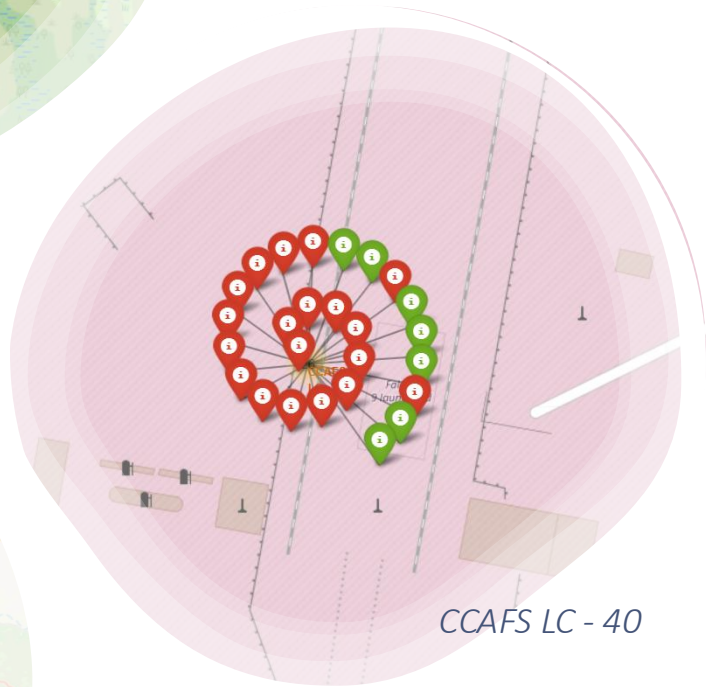
Proximity Benefits:

- Coastal positioning ensures safety and accessibility for transporting materials and personnel.

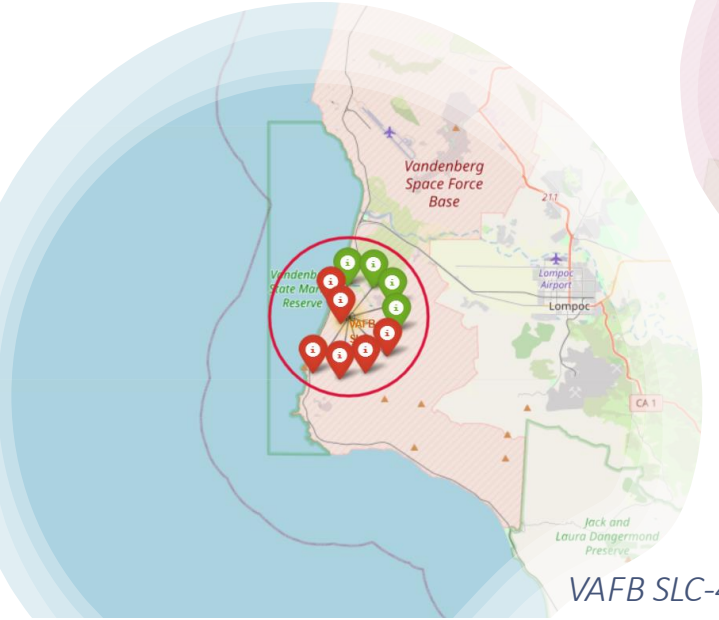

KSC LC-39A



CCAFS SLC-40



CCAFS LC - 40




VAFB SLC-4E

# Map with the color-labeled markers in marker clusters

Green markers indicate successful launches, while red markers indicate failures.

**Insights**:

- **KSC LC-39A** shows the highest success rate among all launch sites.
- Other sites, such as **CCAFS SLC-40** and **VAFB SLC-4E**, also contribute, but with fewer successful outcomes compared to KSC LC-39A.
- This visualization effectively highlights the geographical distribution and success patterns of SpaceX launches

# Folium Map with distances between a launch site to its proximities

**City Proximity**:

- The closest city to a launch site is KSC LC-39A at 16.26 km.
- VAFB SLC-4E is furthest from its nearest city at 39.70 km.

**Coast Proximity**:

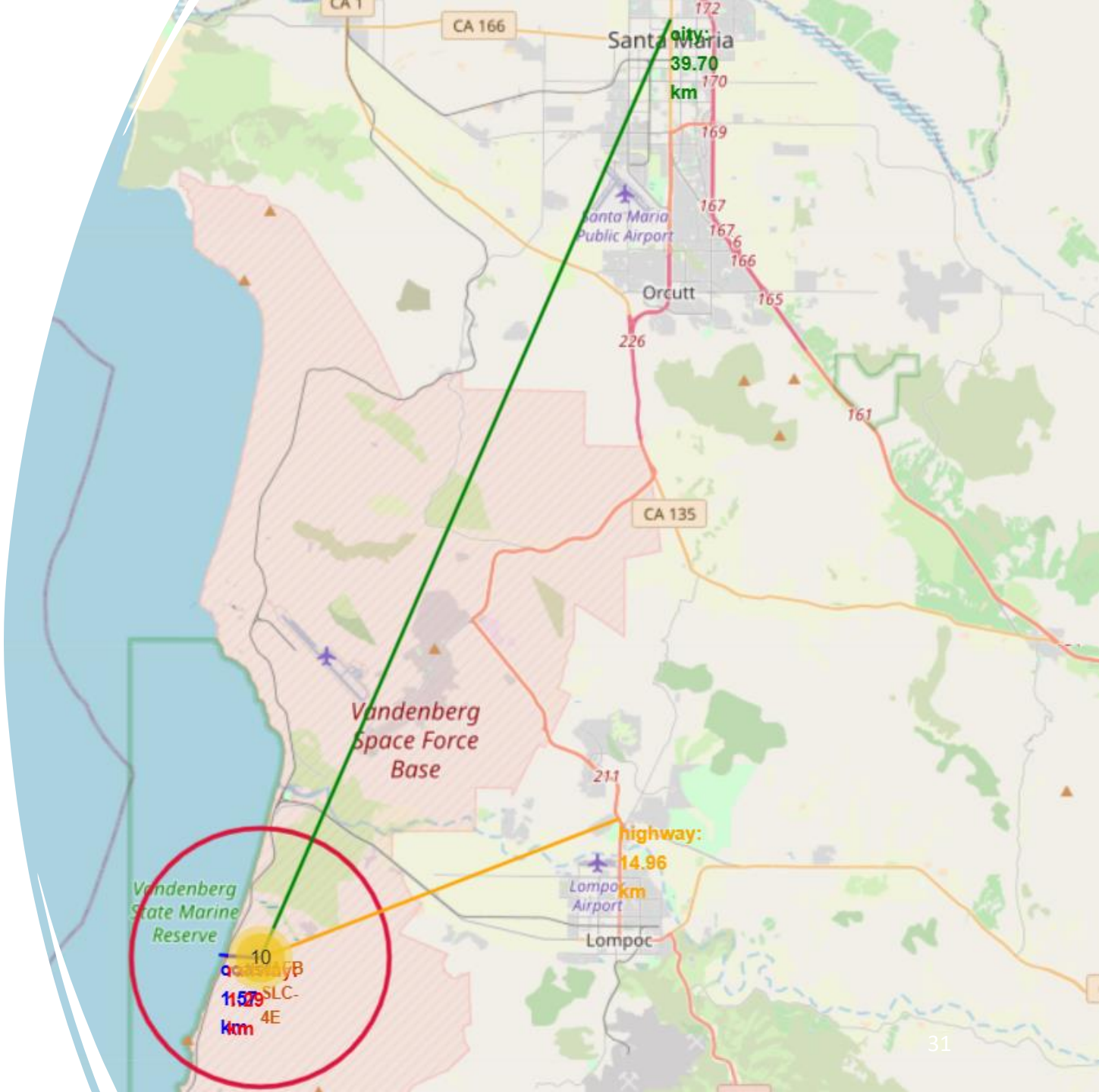
- All launch sites are in close proximity to the coast, ensuring safe launch paths over water.

**Highway Proximity**:

- VAFB SLC-4E has a significantly higher distance to highways compared to other sites (14.96 km).

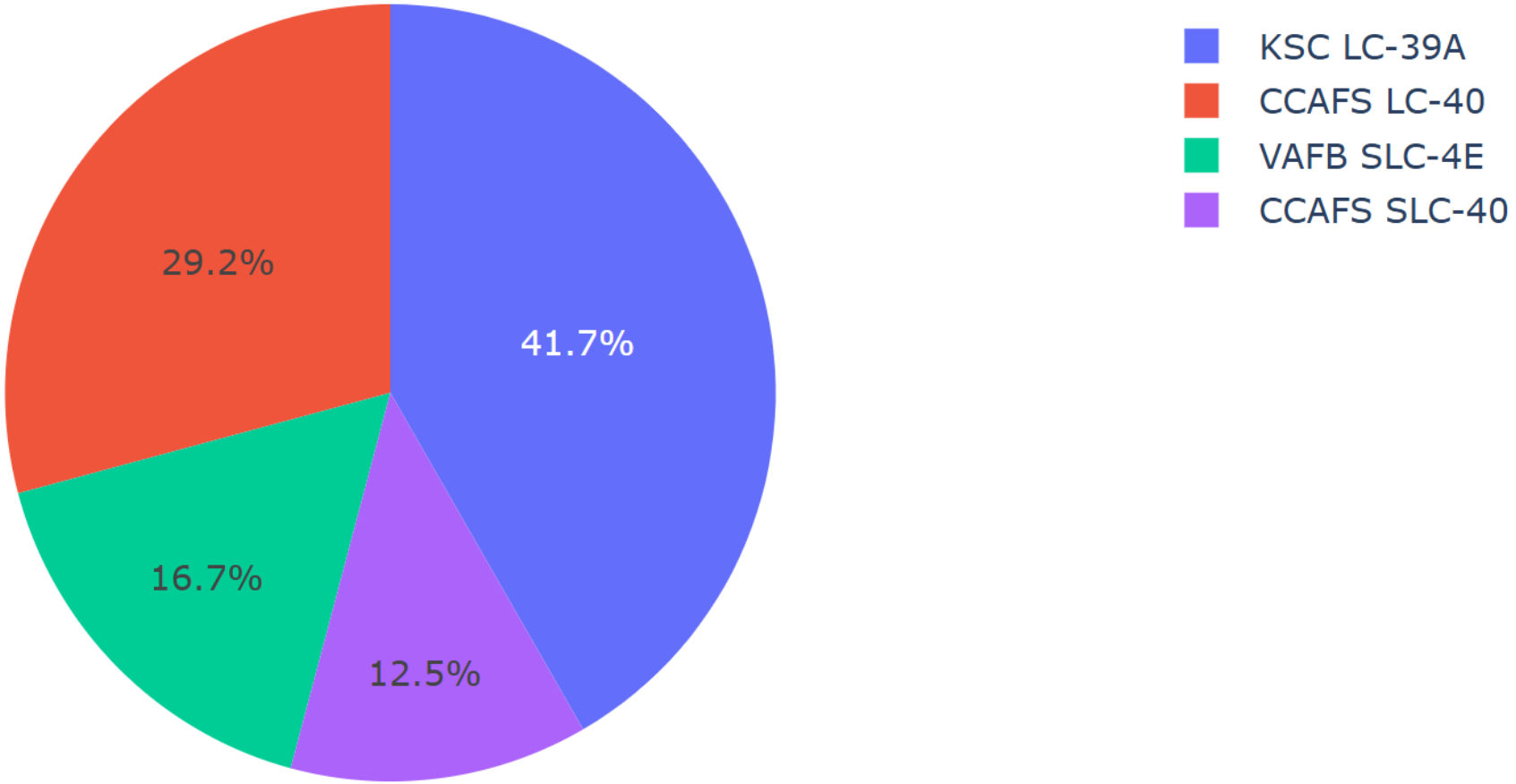
**Railway Proximity**:

- All sites are located near railways, ensuring efficient transportation of materials and equipment.



# Build a Dashboard with Plotly Dash

Section 4




# Total Succes Launches by site

- KSC LC-39A has the most successful launches amongst launch sites (41.7%)

# Total Succes Launches for KSC LC-39A

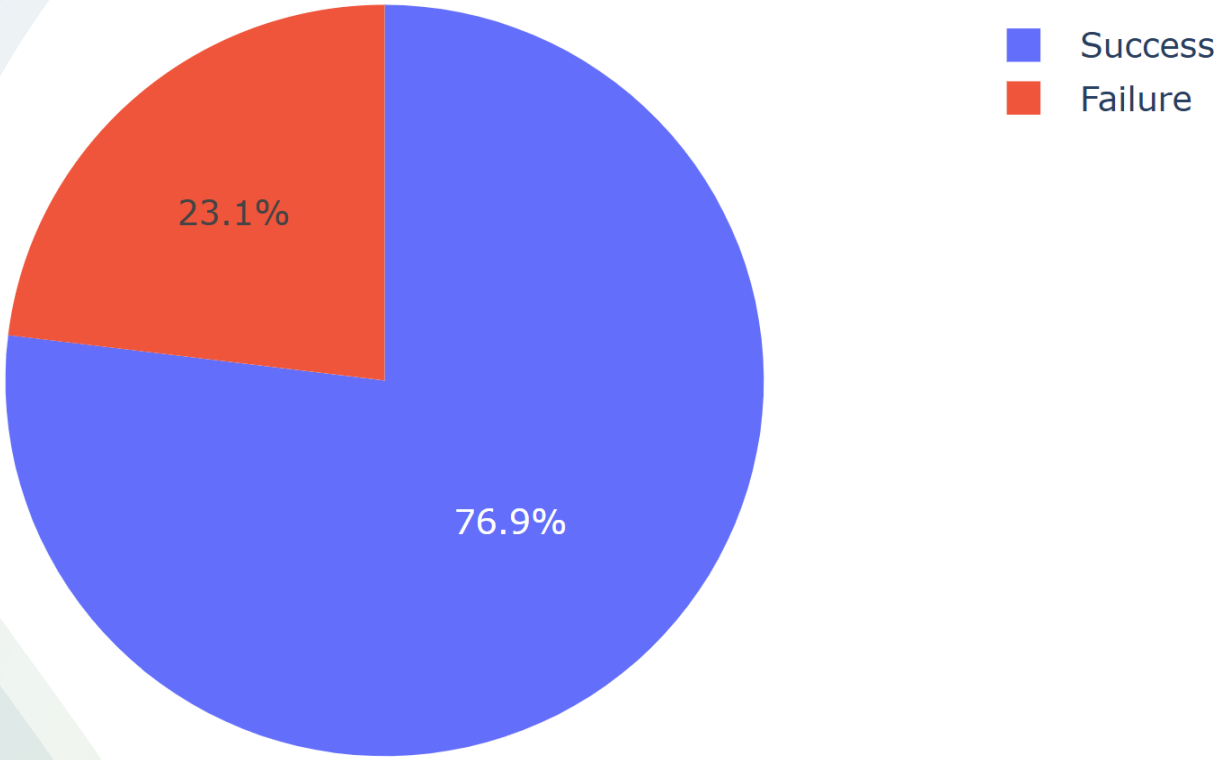
**Success Rate**: KSC LC-39A holds the highest success rate among launch sites at 76.9%.

**Launch Outcome**:

Successful Launches: 10

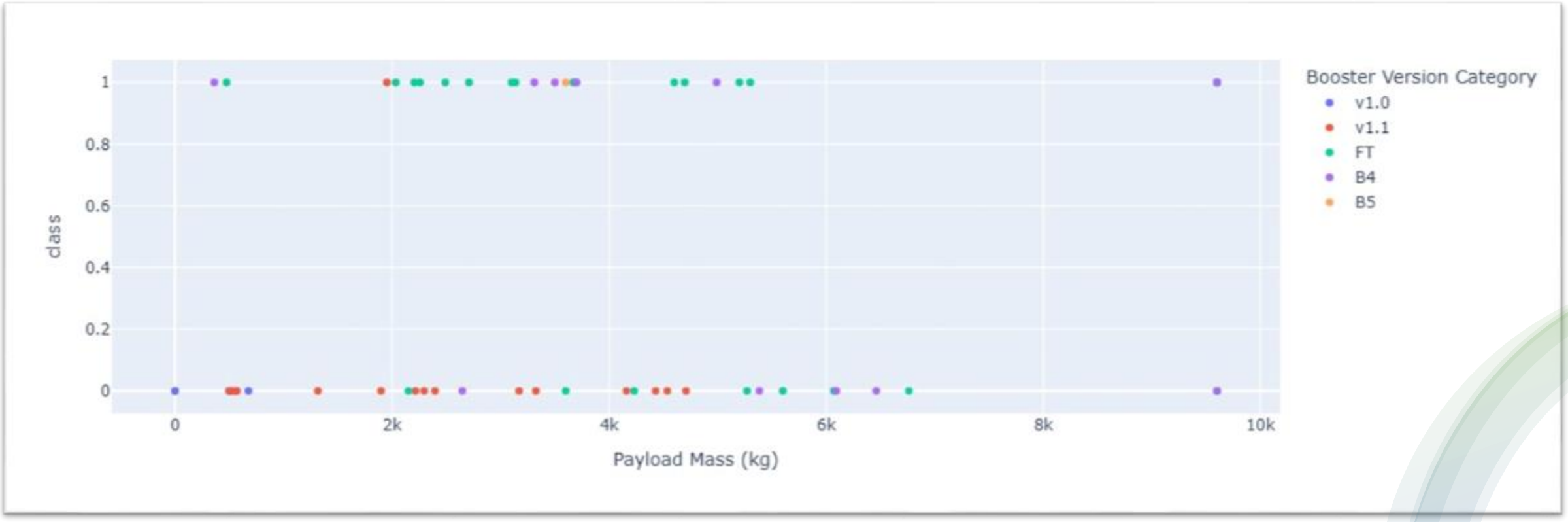
Failed Launches: 3

This indicates the reliability and efficiency of KSC LC-39A for launches.



# Correlation Between Payload and Success for all Sites

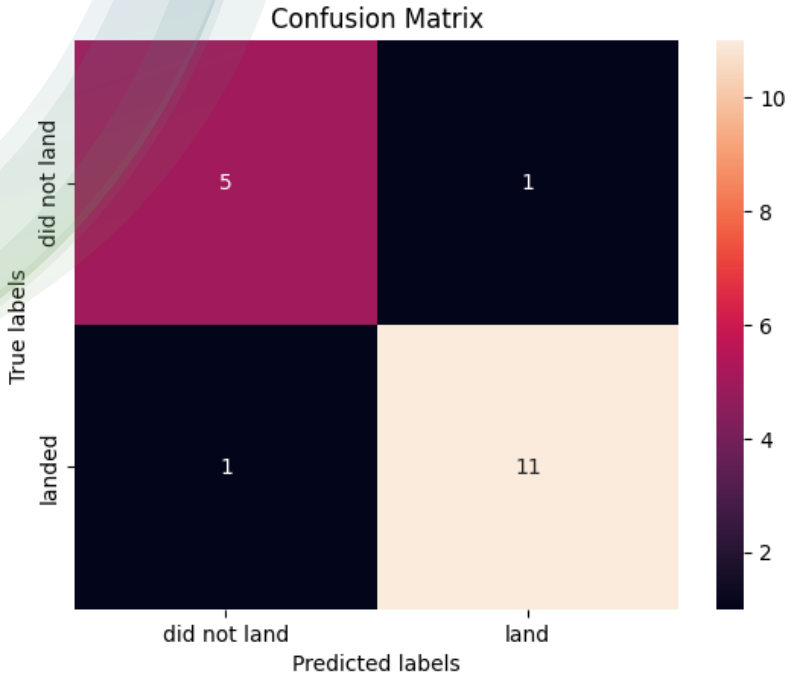
- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- FT Booster: Highest success rate overall.



# Predictive Analysis (Classification)

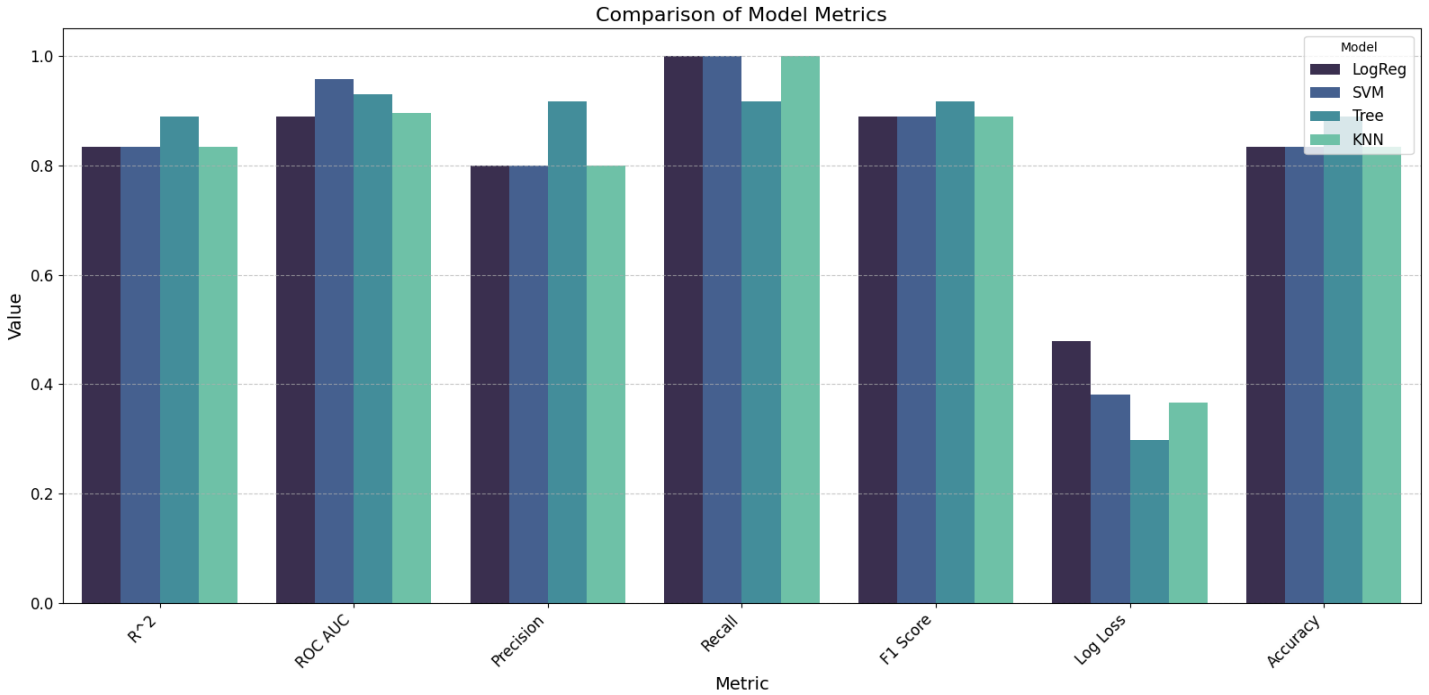
Section 5

# Confusion Matrix




- **Precision**: 91.67%
- **Recall**: 91.67%
- **F1 Score**: 91.67%
- The decision tree classifier demonstrates strong performance in both precision and recall, achieving a balanced and reliable F1 score.

# Classification Accuracy




- **Model Comparison**
- **Decision Tree Classifier** has the highest overall evaluation metrics:
- **Accuracy**: 88.89%
- **F1 Score**: 91.67%
- **Log Loss**: 0.298668 (lowest among all models).
- Other models also performed well but were slightly less effective compared to the decision tree:
- **SVM** achieved the highest ROC AUC of 95.83%.
- **LogReg and KNN** showed consistent accuracy at 83.33% but slightly lower precision and F1 scores.

# Also to find the method performs best i used the following code

```
models = {
    'KNeighbors': knn_cv.best_score_,
    'DecisionTree': tree_cv.best_score_,
    'LogisticRegression': logreg_cv.best_score_,
    'SupportVector': svm_cv.best_score_
}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm, 'with a score of', models[bestalgorithm])


best_params = {
    'DecisionTree': tree_cv.best_params_,
    'KNeighbors': knn_cv.best_params_,
    'LogisticRegression': logreg_cv.best_params_,
    'SupportVector': svm_cv.best_params_
}
print('Best params is:', best_params[bestalgorithm])
```

```
Best model is DecisionTree with a score of 0.8875000000000002
Best params is: {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_spli
```

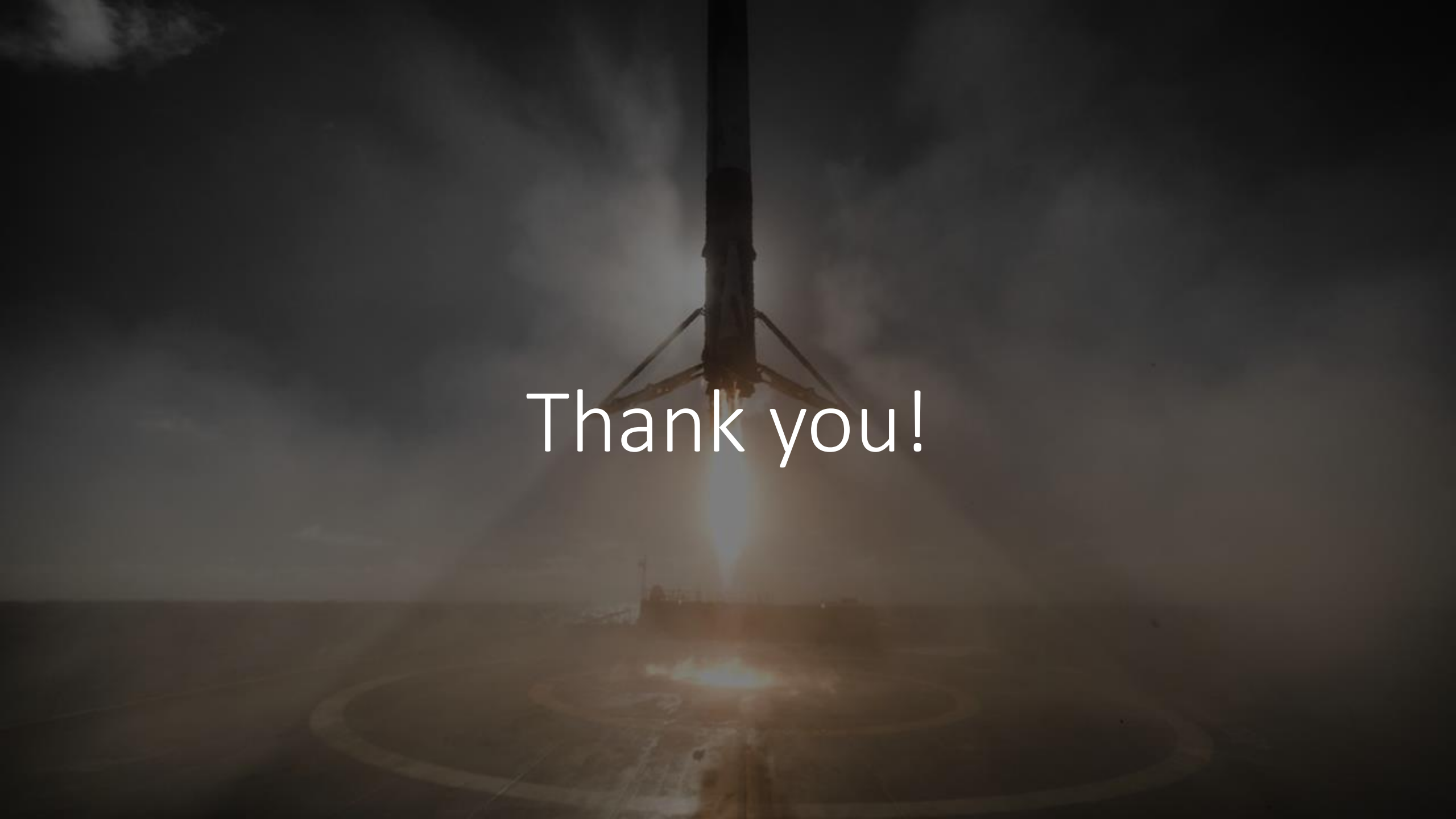
Best model is **DecisionTree** with a score of 0.8875000000000002

**Best params is**:

- Criterion = entropy
- max_depth = 14
- max_features = sqrt
- min_samples_leaf= 2
- min_samples_split = 10
- Splitter = random

# Conclusions

- **Orbit:**
  - **SSO Orbit** demonstrates a perfect **100% success rate**, highlighting its reliability.
  - For **VLEO, LEO, ISS, and MEO orbits**, success rates vary between **60-90%**, indicating a need for further refinement in these categories.
  - **LEO, ISS, and PO orbits** show a positive correlation between success rates and **higher flight numbers** as well as **greater payloads**.
- **Launch Sites**
  - Among all sites, **KSC LC-39A has the highest success rate**, making it the most reliable.
- **Booster Performance**
  - The **FT Booster** version has the **highest overall success rate** across all sites.
  - The **B4 Booster** performs exceptionally well when launched from **KSC LC-39A site**.
- **Geography**
  - All launch sites are positioned **close to the Equator** and **coastal areas**, leveraging physics for efficient launches.
  - Sites are located **far from urban centers, highways, and railways**, reducing potential risks from failed launches while remaining logistically viable.
- **Predictive Modeling**
  - The **Decision Tree Classifier** emerged as the most effective model for predicting launch outcomes, supported by superior evaluation metrics.
- **Overall Key Takeaways**
- The steady improvement in success rates indicates strong growth in space launch capabilities.
- Strategic placement of sites and advancements in booster technology significantly contribute to mission success.
- Data-driven predictive models offer promising accuracy for future mission planning.


Thank you!