IBM Developer
SKILLS NETWORK

**abdelouahed akharaze**
**March 2024**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone

# Outline

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

# Executive Summary

- I gathered data from both the public SpaceX API and the SpaceX Wikipedia page. I established a 'class' column to categorize successful landings. Utilizing SQL, visualization methods, folium maps, and dashboards, I explored the data. I selected pertinent columns to serve as features, converting categorical variables into binary using one-hot encoding. After standardizing the data, GridSearchCV was employed to identify the best parameters for the machine learning models. The accuracy scores of all models were visualized.

- Four machine learning models were created: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. Each produced similar outcomes, achieving an accuracy rate of roughly 83.33%. Notably, all models tended to overpredict successful landings. It's evident that additional data would improve model accuracy and determination.

# Introduction

- **Background:**

- The era of commercial space exploration has dawned, marked by the emergence of companies like SpaceX. SpaceX has gained prominence for its competitive pricing, offering launches at $62 million compared to the industry average of $165 million USD. This affordability is largely attributed to SpaceX's innovative capability to recover and reuse parts of its rockets, notably Stage 1. Now, a contender in the space industry, Space Y, seeks to challenge SpaceX's dominance.

- **Problem:**

- Space Y has enlisted our expertise to develop a machine learning model capable of predicting the successful recovery of Stage 1 rockets.

# Section 1

# Methodology

- **Data Collection Methodology:**

- - Integrated data from the SpaceX public API and the SpaceX Wikipedia page.
- - Executed data wrangling procedures.
- - Classified true landings as successful and labeled others as unsuccessful.
- - Conducted exploratory data analysis (EDA) employing visualization techniques and SQL.
- - Utilized interactive visual analytics tools like Folium and Plotly Dash.
- - Conducted predictive analysis using classification models.
- - Fine-tuned models using GridSearchCV.

# Data Collection

- Data collection involved a dual approach, combining API requests from SpaceX's public API with web scraping of data from a table within SpaceX's Wikipedia entry.
- **SpaceX API Data Columns:**
- - FlightNumber, Date, BoosterVersion
- - PayloadMass, Orbit, LaunchSite
- - Outcome, Flights, GridFins
- - Reused, Legs, LandingPad
- - Block, ReusedCount, Serial
- - Longitude, Latitude
- **Wikipedia Webscrape Data Columns:**
- - Flight No., Launch site, Payload
- - PayloadMass, Orbit, Customer
- - Launch outcome, Version, Booster
- - Booster landing, Date, Time

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection – SpaceX API

**Filter data to only include Falcon 9 launches**

**Impute missing PayloadMass values with mean**

**Request SpaceX APIs for: - .JSON file Lists of Launch Site, Booster Version, Payload Data**

**Cast dictionary to a DataFrame**

**Use Json_normalize to convert JSON data to a DataFrame**

**Extract relevant data from the dictionary**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection - Scraping

Request Wikipedia HTML

Use BeautifulSoup with html5lib Parser

Find launch info HTML table

Create dictionary

Iterate through table cells to extract data to dictionary

Cast dictionary to DataFrame

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

# Data Wrangling

**Extract 'Mission Outcome' and 'Landing Location' components from the 'Outcome' column.**

- **- Create a new training label column 'class'.**
- **- Assign a value of 1 to 'class' if 'Mission Outcome' is True and 'Landing Location' is either ASDS, RTLS, or Ocean. Assign 0 otherwise.**
-

**Value Mapping:**

- **- True ASDS, True RTLS, True Ocean → Set to 1**
- **- None None, False ASDS, None ASDS, False Ocean, False RTLS → Set to 0**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

**Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year.**

**Plots Used:**

**- Flight Number vs. Payload Mass**

**- Flight Number vs. Launch Site**

**- Payload Mass vs. Launch Site**

**- Orbit vs. Success Rate**

**- Flight Number vs. Orbit**

**- Payload vs. Orbit**

**- Success Yearly Trend**

**Visualizations:**

**- Scatter plots, line charts, and bar plots were utilized to examine relationships between variables.**

**- The objective was to determine if significant relationships exist among the variables, aiding in the decision-making process for their inclusion in the machine learning model training.**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

# EDA with SQL

- **Loaded data set into IBM DB2 Database.**
- **Queried using SQL Python integration.**
- **Queries were made to get a better understanding of the dataset.**
- **Queried information about launch site names, mission outcomes, various payload sizes of customers and booster versions, and landing outcomes.**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

**Folium maps:**

**- Mark Launch Sites, successful and unsuccessful landings**

**- Include proximity to key locations: Railway, Highway, Coast, and City**

**- Enable understanding of launch site locations**

**- Visualize successful landings relative to location**

https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

# Build a Dashboard with Plotly Dash

- **Dashboard Features:**
- **- Includes a pie chart and a scatter plot.**
- **- Pie chart can be selected to show:**
- **- Distribution of successful landings across all launch sites.**
- **- Individual launch site success rates.**
- **- Scatter plot takes two inputs:**
- **- All sites or individual site.**
- **- Payload mass on a slider between 0 and 10000 kg.**
- **- Purpose:**
- **- Pie chart: Visualize launch site success rate.**
- **- Scatter plot: Explore variation in success across launch sites, payload mass, and booster version category.**

# Predictive Analysis (Classification)

Split label column

Split test set for all models

Fit and Transform data

Use GridSearchCV to find optimal parameters

GridSearchCV with Train_test_split (cv=10) to find optimal parameters

Score models on Confusion Matrix

Barplot to compare scores of models on LogReg, SVM, Decision Tree, and KNN models

Use Standard Scaler for features

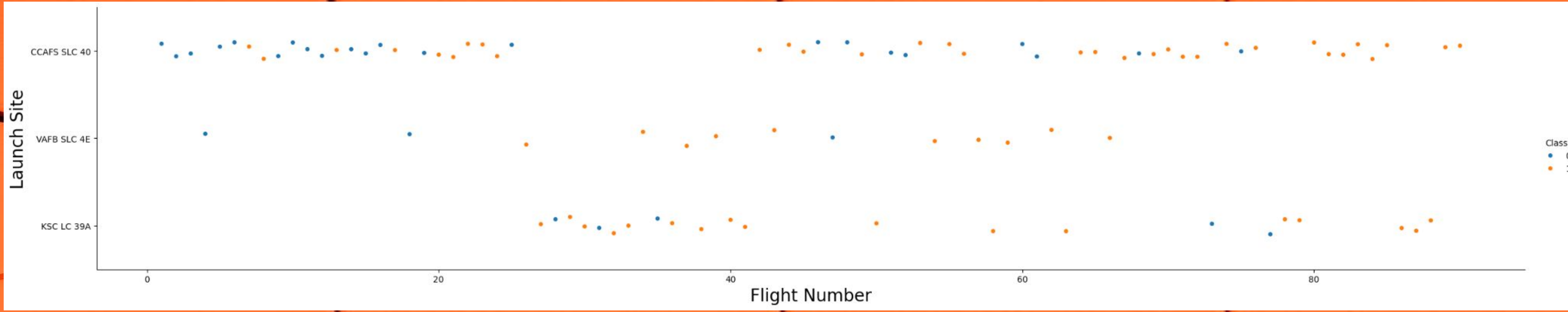
https://github.com/abdelouahedakharaze/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

# Results

- **In the following slides, I will present the results of each step outlined previously**

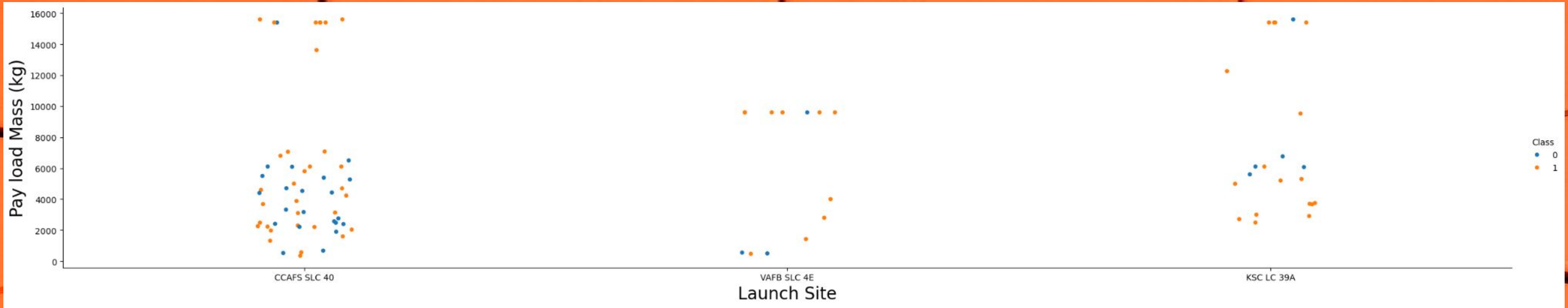
# Section 2

# Flight Number vs. Launch Site




**Explanation:**
**- The earliest flights all failed while the latest flights all succeeded.**
**- The CCAFS SLC 40 launch site has about a half of all launches.**
**- VAFB SLC 4E and KSC LC 39A have higher success rates.**
**- It can be assumed that each new launch has a higher rate of success.**

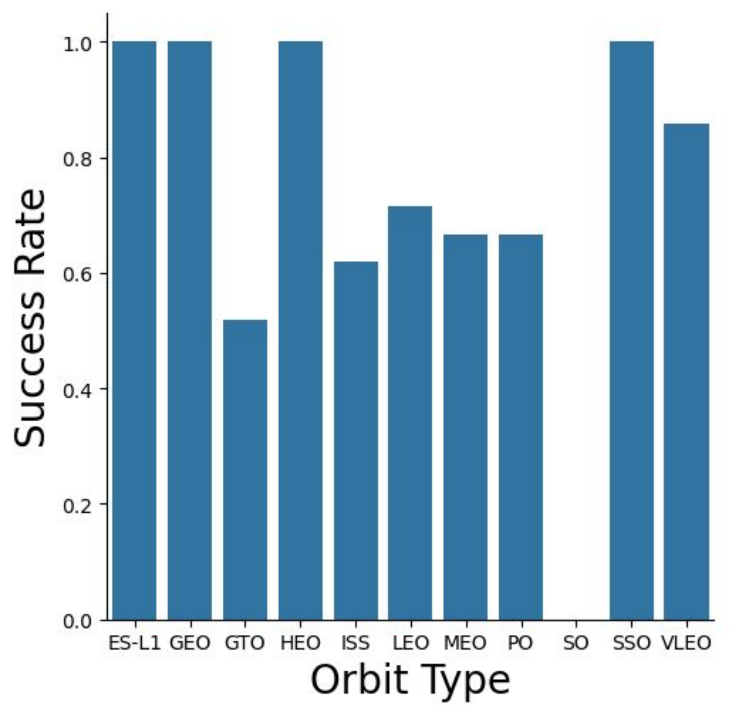
# Payload vs. Launch Site



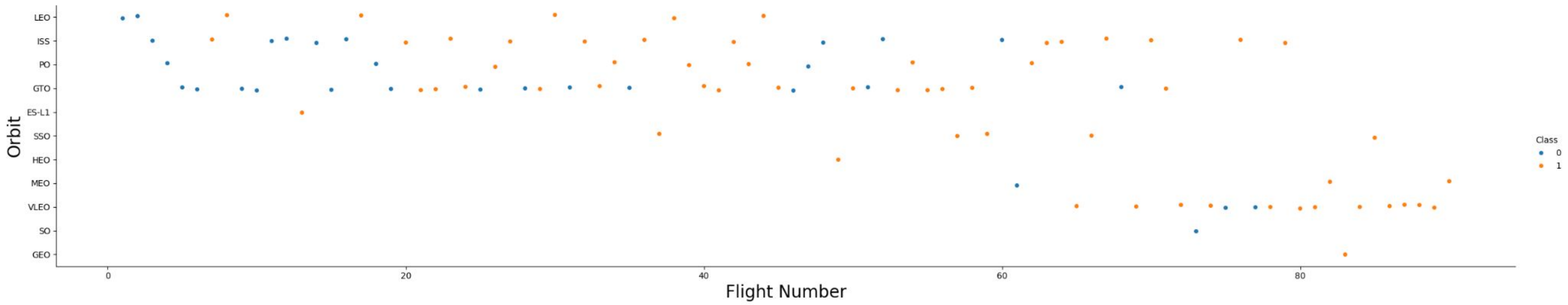

Insight:
- Across all launch sites, there's a positive correlation between payload mass and success rate.
- Notably, the majority of launches with a payload mass exceeding 7000 kg achieved success.
- Particularly at KSC LC 39A, there's a noteworthy trend where launches with a payload mass under 5500 kg consistently attained a 100% success rate.

# Success Rate vs. Orbit Type

- Insight:
- - Orbits with a 100% success rate include ES-L1, GEO, HEO, and SSO.
- - Conversely, the SO orbit recorded a 0% success rate.
- - Orbits with success rates ranging from 50% to 85% encompass GTO, ISS, LEO, MEO, and PO.

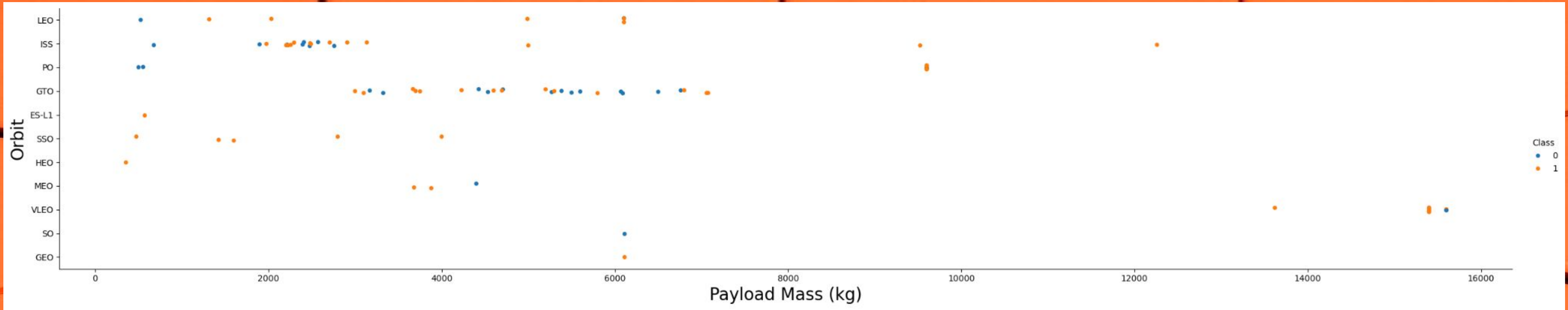


# Flight Number vs. Orbit Type




Insight:
- Success in the LEO orbit appears to correlate with the number of flights.
- Conversely, in the GTO orbit, there seems to be no discernible relationship between flight number and success.

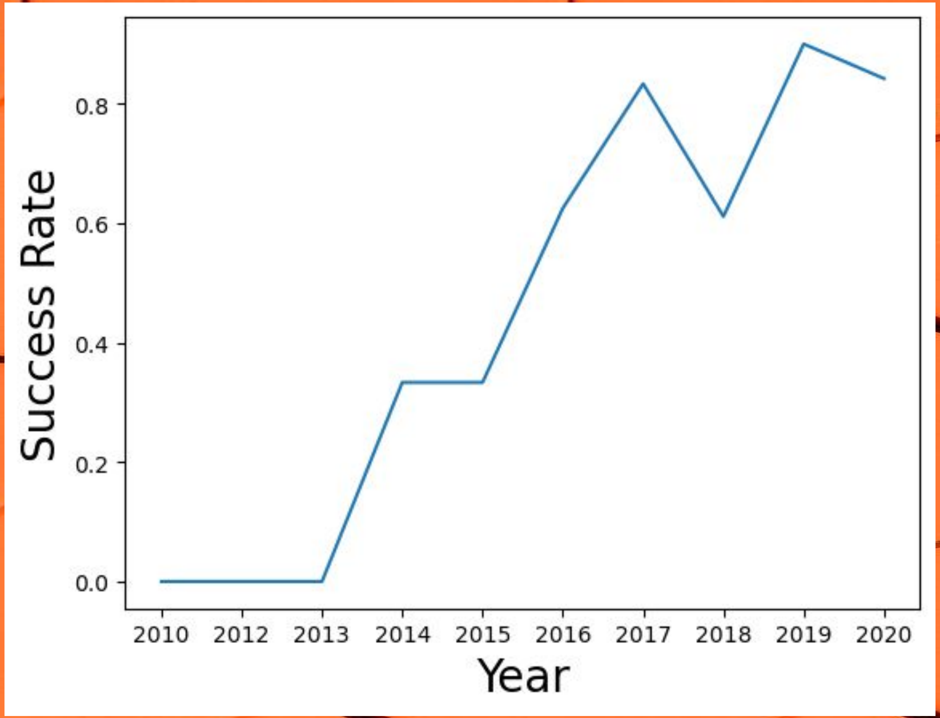
# Payload vs. Orbit Type




Insight:
- Heavy payloads negatively impact success rates in GTO orbits, while they have a positive effect on GTO and Polar LEO (ISS) orbits.

# Launch Success Yearly Trend

- Insight:
- \- The success rate has shown a consistent increase from 2013 to 2020.



# All Launch Site Names

Insight:
- This section displays the names of the unique launch sites involved in the space missions.

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

[10]

* sqlite:///my_data1.db
Done.

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

# Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

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

[26]

Python

```
* 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 |

Task 3

Display 5 records where launch sites begin with the string 'CCA'

# Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS "TotalPayloadMassNASA_CRS" FROM SPACEXTBL WHERE "Customer" = 'NASA (CRS)';
```

[15]

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

| TotalPayloadMassNASA_CRS |
|---|
| 45596 |

Displaying the total payload mass carried by boosters launched by NASA (CRS)

# Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

[16]

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS "AveragePayloadMass_F9" FROM SPACEXTBL WHERE "Booster_Version" LIKE 'F9%';
```

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

| AveragePayloadMass_F9 |
|---|
| 6138.287128712871 |

Displaying average payload mass carried by booster version F9 v1.1

# First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

*Hint:Use min function*

```
%sql SELECT MIN("Date") AS "FirstSuccessfulLandingOnGroundPadDate" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)';
```

Python

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

| FirstSuccessfulLandingOnGroundPadDate |
| --- |
| 2015-12-22 |

Listing the date when the first successful landing outcome in ground pad was achieved.

# Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_'
```

Python

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

| Date | Time (UTC | Booster_Version | L unch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2016-05-06 | 5:21:00 | F9 FT B1022 | CCAFS LC-40 | JCSAT-14 | 4696 | GTO | SKY Perfect JSAT Group | Success | Success (drone ship) |
| 2016-08-14 | 5:26:00 | F9 FT B1026 | CCAFS LC-40 | JCSAT-16 | 4600 | GTO | SKY Perfect JSAT Group | Success | Success (drone ship) |
| 2017-03-30 | 22:27:00 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO | SES | Success | Success (drone ship) |
| 2017-10-11 | 22:53:00 | F9 FT B1031.2 | KSC LC-39A | SES-11 / EchoStar 105 | 5200 | GTO | SES EchoStar | Success | Success (drone ship) |

Explanation:

• Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

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

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

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

Listing the total number of successful and failure mission outcomes

# Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

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

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

Listing the names of the booster versions which have carried the maximum payload mass.

# 2015 Launch Records

```
            WHEN '10' THEN 'October'
            WHEN '11' THEN 'November'
            WHEN '12' THEN 'December'
        END AS Month,
        "Landing_Outcome",
        "BoosterVersion",
        "LaunchSite"
    FROM SPACEXTBL
    WHERE substr("Date", 0, 5) = '2015'
    AND "Landing_Outcome" = 'Failure (drone ship)';
```

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

| Month | Landing_Outcome | "BoosterVersion" | "LaunchSite" |
| --- | --- | --- | --- |
| January | Failure (drone ship) | BoosterVersion | LaunchSite |
| April | Failure (drone ship) | BoosterVersion | LaunchSite |

Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015.

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

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

[32]

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

| Landing_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 |

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

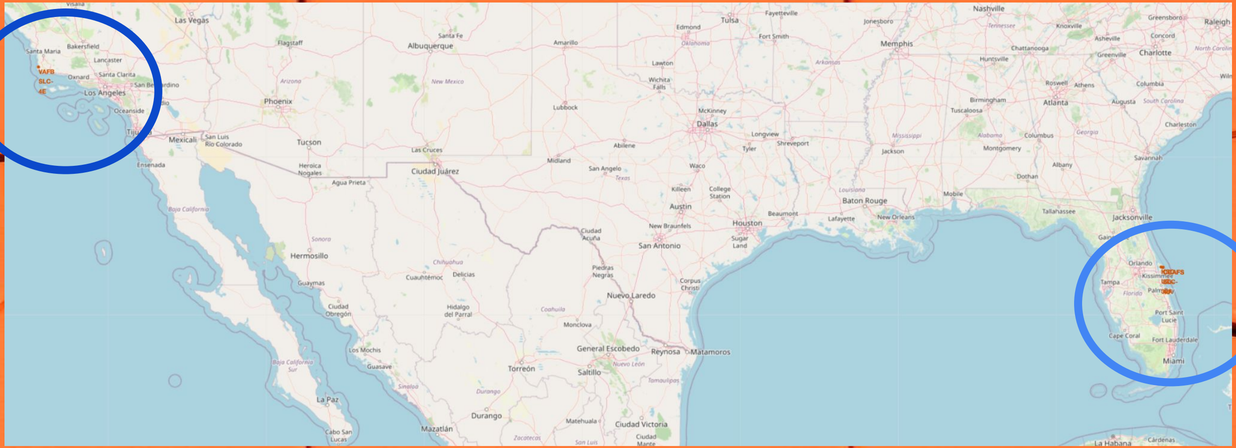
# Section 3

# All launch sites' locations

Insight:

- Most launch sites are situated near the Equator line, leveraging the higher speed of the Earth's rotation at the equator. Launching from the equator allows spacecraft to inherit the Earth's rotational velocity, aiding in achieving and maintaining orbit due to inertia.

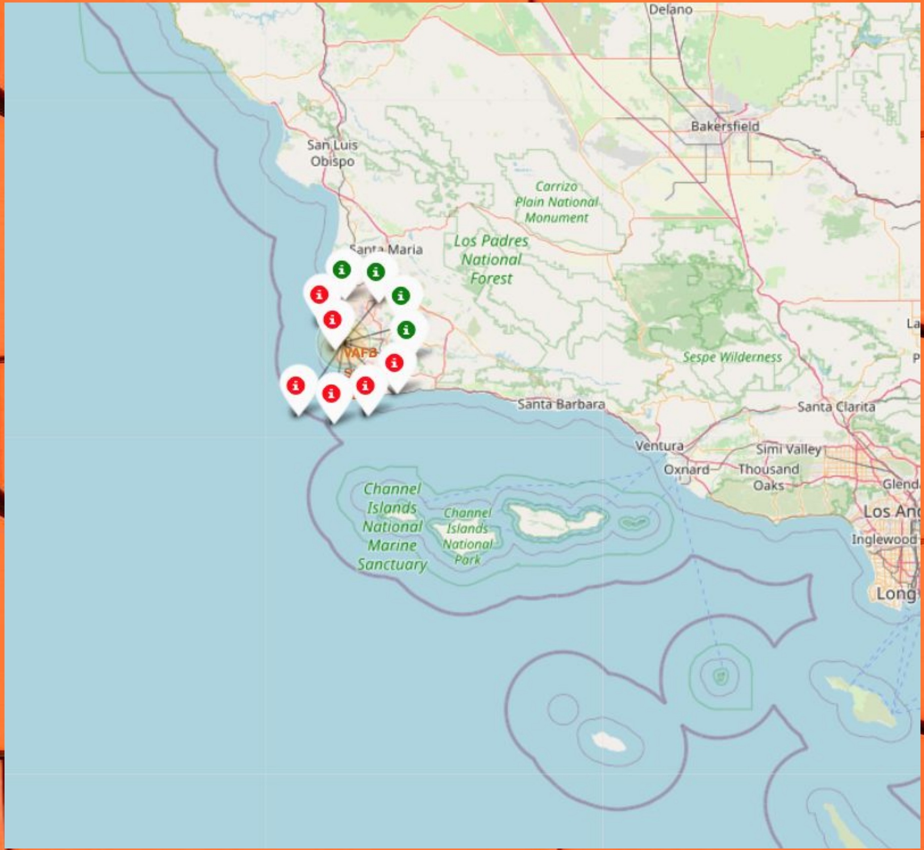
- Additionally, all launch sites are strategically located close to the coast, minimizing the risk of debris from rockets falling or exploding near populated areas when launched towards the ocean.



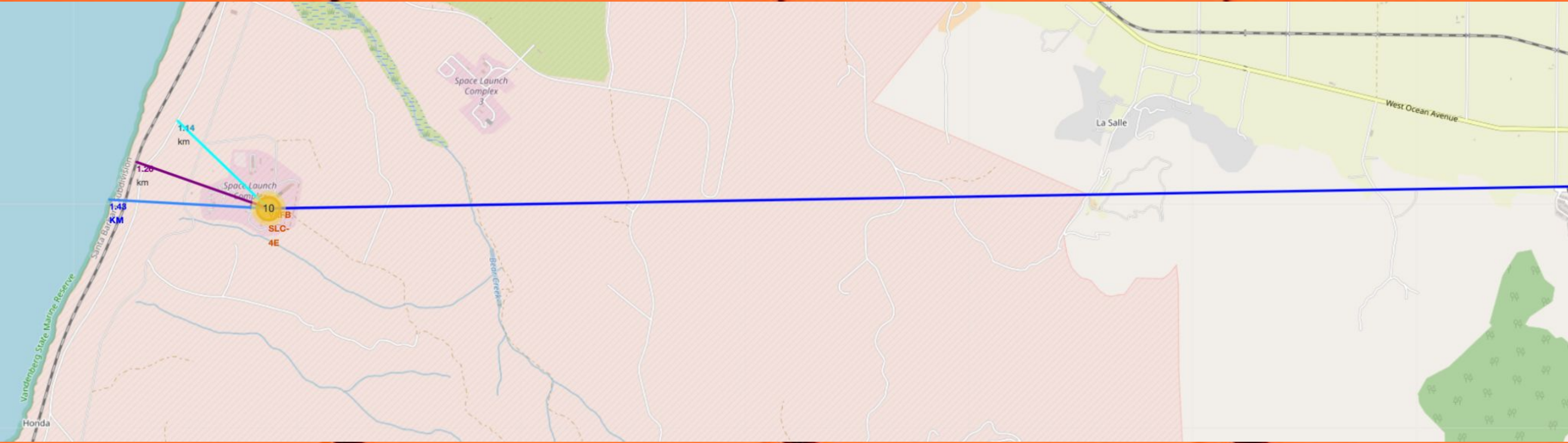
# Colour-labeled launch records on the map

Insight:

- The markers labeled with colors enable straightforward identification of launch sites with comparatively high success rates.
- Green markers denote successful launches, while red markers signify failed launches.
- It's worth noting that KSC LC-39A stands out with a particularly high success rate.



# Distance from the launch site SLC 4E to its proximities

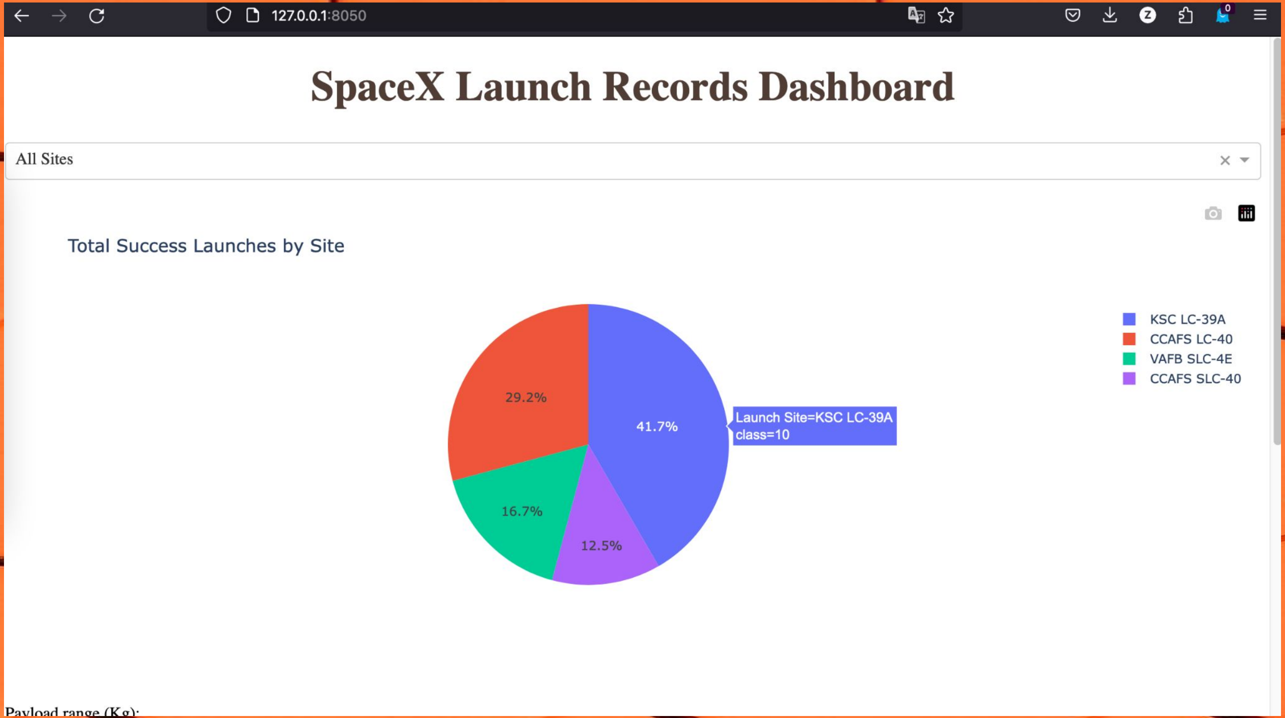



This launch site is too far from the next city , so it's secure
Close to railway , but very far from highway

# Section 4

# Dashboard Launch success count for all sites

The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.



# Section 5

# Classification Accuracy

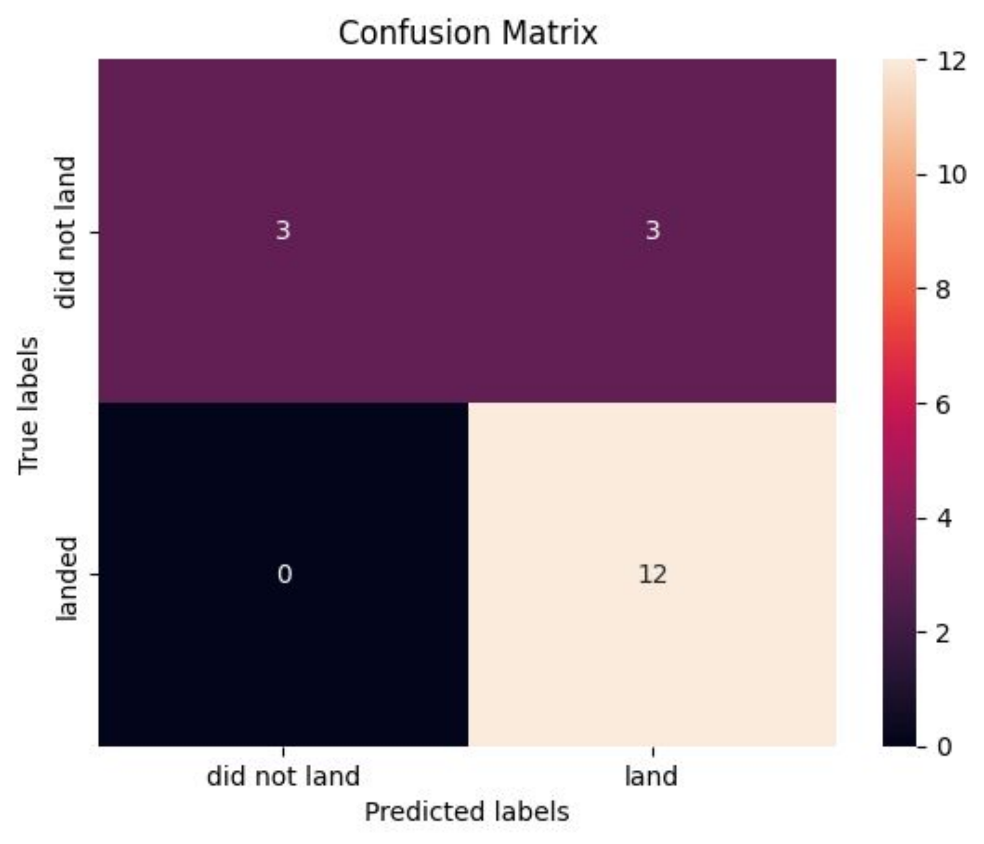
```
algo_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tree': [tree_cv.best_score_],
df = pd.DataFrame.from_dict(algo_score, orient='index', columns=['Best scores'])
df
```

Python

| | Best scores |
|---|---|
| Logistic regresssion | 0.846429 |
| SVM | 0.848214 |
| Decision tree | 0.875000 |
| KNN | 0.848214 |

# Confusion Matrix

Explanation:
- Upon examining the confusion matrix, it's evident that logistic regression can effectively differentiate between different classes.
- However, the primary issue highlighted is the occurrence of false positives.



# Conclusions

- In summary:
- - The Decision Tree Model emerges as the most effective algorithm for this dataset.
- - Launches with lower payload masses tend to yield better results than those with larger payload masses.
- - The concentration of launch sites near the Equator line and their close proximity to coastlines indicate strategic positioning for launches.
- - Over the years, there's a consistent upward trend in the success rate of launches.
- - Notably, KSC LC-39A stands out with the highest success rate among all launch sites.
- - Orbits such as ES-L1, GEO, HEO, and SSO demonstrate a remarkable 100% success rate.
- These findings highlight the importance of considering factors such as payload mass, launch site location, and historical success rates when planning space missions. Additionally, the identified trends can inform future decision-making processes in the space industry, ultimately leading to more successful and efficient space exploration endeavors.