Winning Space Race with Data Science

Luís Felipe

January 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

The primary objective of this undertaking is to assess the likelihood of a successful landing for the first stage of the Falcon 9 rocket, given its substantial impact on forecasting the relaunch cost. The dataset utilized in this endeavor was gathered from the SpaceX REST API and Wikipedia. Following a series of data wrangling procedures to identify optimal predictors for our target variable, Exploratory Data Analysis (EDA) and feature scaling techniques were applied. Visualization, employing scatter and line plots, played a pivotal role in these analytical processes. Subsequently, a set of Machine Learning (ML) models were developed to make predictions about future outcomes.

The findings indicated that the result was influenced by the orbit, payload mass, launch site, and various other technical parameters.

Introduction

SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62 million, while other providers charge over $165 million each. A significant portion of these savings is attributed to SpaceX's ability to reuse the first stage.

Therefore, the goal is to ascertain whether the first stage will successfully land. By determining this, we can establish the launch cost. The information becomes valuable if an alternative company intends to bid against SpaceX for a rocket launch.

Methodology

- Data collection methodology:
  - The first data was collected by SpaceX REST API
  - The data was scraped from Wikipedia
- Perform data wrangling
  - Numeric representations were generated for categorical features through the application of one- hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different classifiers were tested through Sci-Kit Learn framework

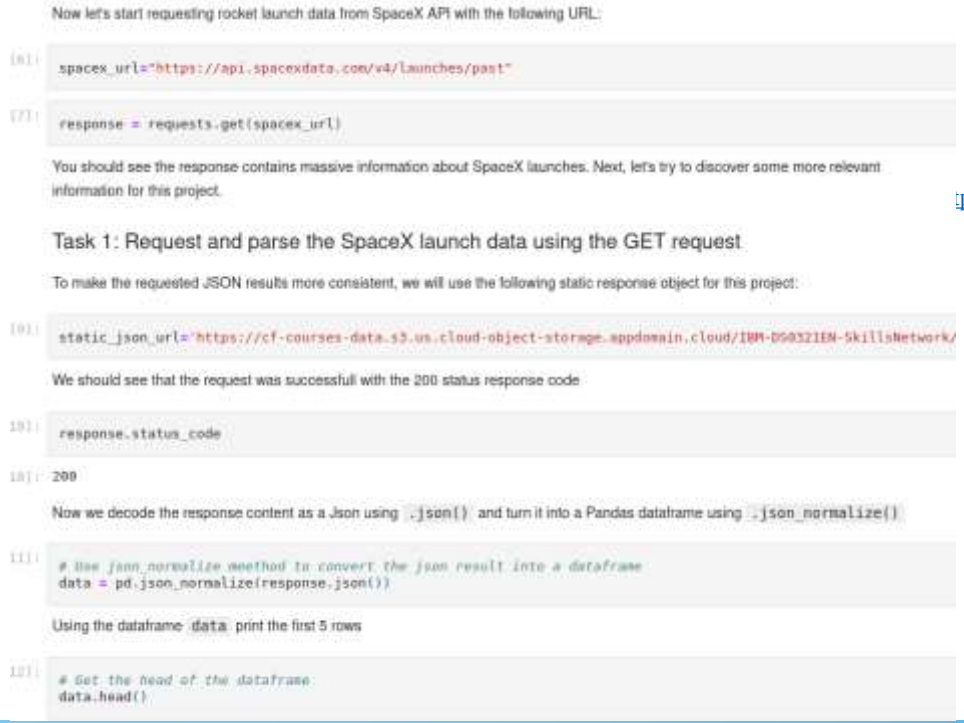
Data Collection

The data acquisition process employed a diverse set of methodologies. Initially, data retrieval involved sending a GET request to the SpaceX API ("https://api.spacexdata.com/v4/launches/past"). Following this, the response content was decoded into a JSON format using the json() function and further transformed into a pandas dataframe using the json_normalize function.

In parallel, comprehensive data cleansing procedures were implemented, addressing the identification and treatment of missing values as needed. Additionally, web scraping techniques were utilized to extract Falcon 9 launch records from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches), making use of the BeautifulSoup library. The primary aim was to retrieve the launch records presented as an HTML table, parsing the table content, and converting it into a Pandas DataFrame for subsequent analytical purposes.

Data Collection - API

The GET request was utilized on the SpaceX API to gather data, followed by cleaning the obtained data and conducting basic data wrangling and formatting procedures.



Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

You should see the response contains massive information about SpaceX launches. Next, let's try to discover some more relevant information for this project.

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

The complete process:

ps://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week1_task1.ipynb

Data Collection - Wikipedia

Web scraping techniques were employed to extract Falcon 9 launch records from the web using BeautifulSoup. The parsing of the table was executed, and the data was transformed into a pandas dataframe.

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
response.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

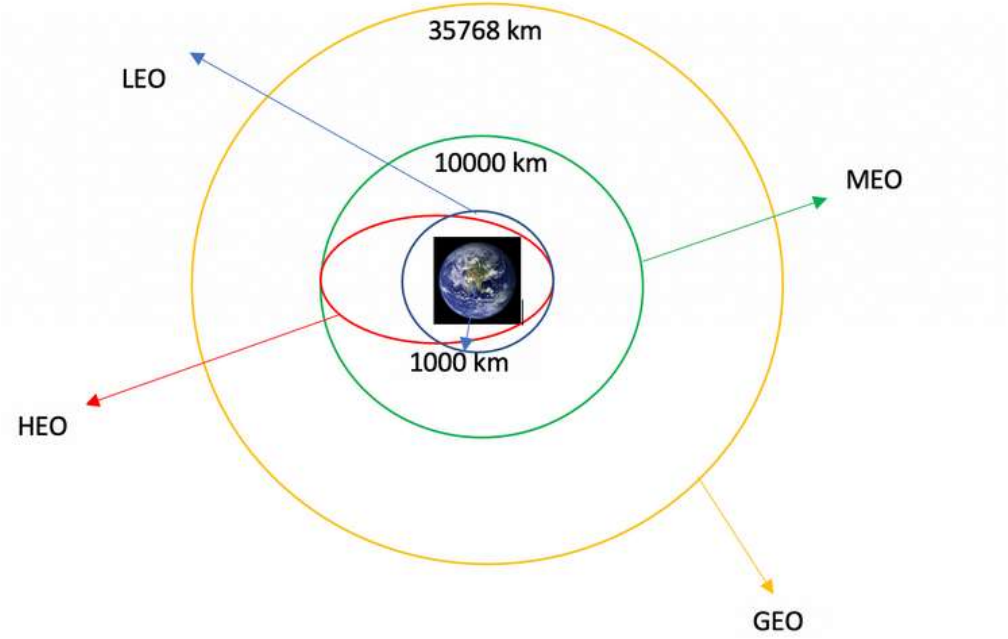
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text)
```

The complete process:

ub.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week1_task2.ipynb

Data Wrangling

Exploratory data analysis was conducted to ascertain the training labels. The count of launches at each site and the frequency of each orbit were computed.



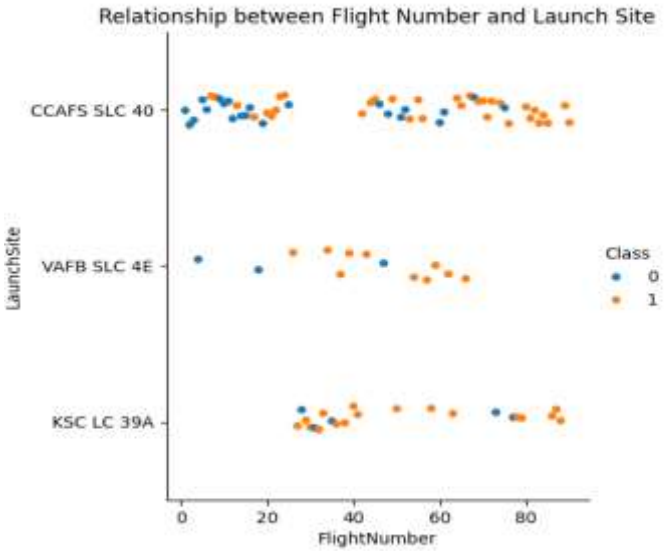
The complete process:

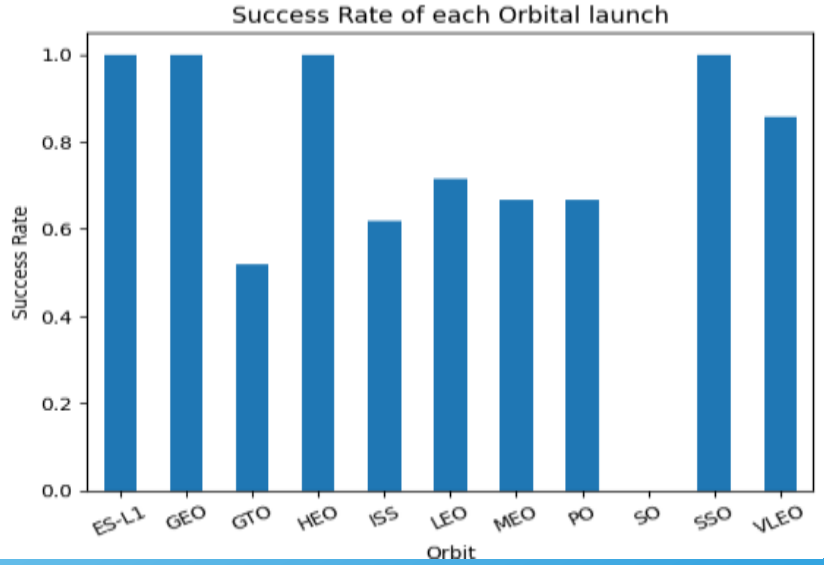
https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week1_Task3.ipynb

EDA with Data Visualization

Data exploration involved visualizing relationships, including those between flight numbers and launch sites, payloads and launch sites, success rates across various orbit types, flight numbers and orbit types, and the yearly trend in launch success.

https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week2_Task2.ipynb





EDA with SQL

The SpaceX dataset was loaded into a PostgreSQL database seamlessly within the Jupyter notebook. Exploratory Data Analysis (EDA) was conducted using SQL to derive insights from the data. Queries were formulated to discover, for example:

Names of unique launch sites in the space mission.

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

Average payload mass carried by booster version F9 v1.1.

Total number of successful and failure mission outcomes.

Failed landing outcomes in drone ship, along with their booster versions and launch site names.

The complete EDA:

https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week2_task2.ipynb

Interactive Visual Analytics with Folium

- All launch sites were marked, and map objects such as markers, circles, and lines were incorporated to signify the success or failure of launches for each site on the Folium map.
- The feature launch outcomes (failure or success) were designated to class 0 and 1, i.e., 0 for failure and 1 for success.
- Utilizing color-labeled marker clusters, identification was made of launch sites with relatively high success rates.
- Distances between a launch site and its proximities were calculated, addressing questions such as:
  - Proximity to railways, highways, and coastlines.
  - Maintenance of a certain distance from cities.

- The complete process:

https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week3_task1.ipynb

Build a Dashboard with Plotly Dash

An interactive dashboard was constructed using Plotly Dash.

Pie charts were generated to display the overall launches from specific sites.

Scatter graphs were employed to illustrate the correlation between outcomes and payload mass (Kg) for various booster versions.

The complete process:

https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/task2_week3.py

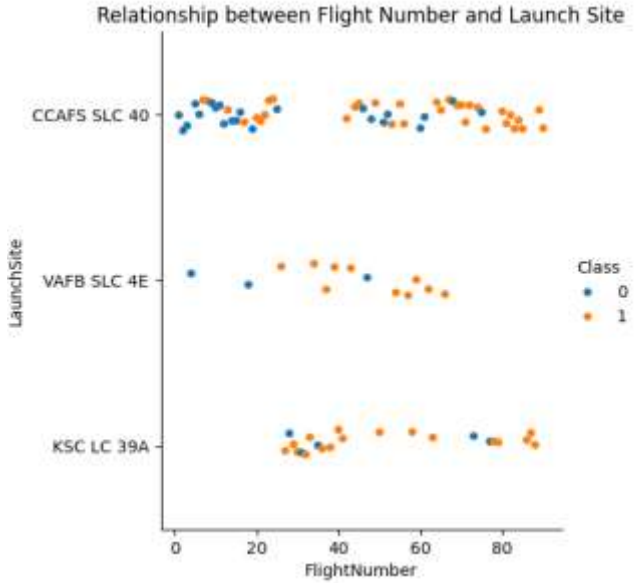
Machine Learning Prediction

Data was loaded utilizing numpy and pandas, followed by data transformation and segmentation into training and testing sets. Various machine learning models were constructed, and distinct hyperparameters were fine-tuned through GridSearchCV. Accuracy served as the metric for the model, and enhancements were made through feature engineering and algorithm tuning. The most effective classification model was identified.

The notebook link is provided below:
https://github.com/felipealmsz/IBM-Data-Science-Space-x/blob/main/Week4_task1.ipynb

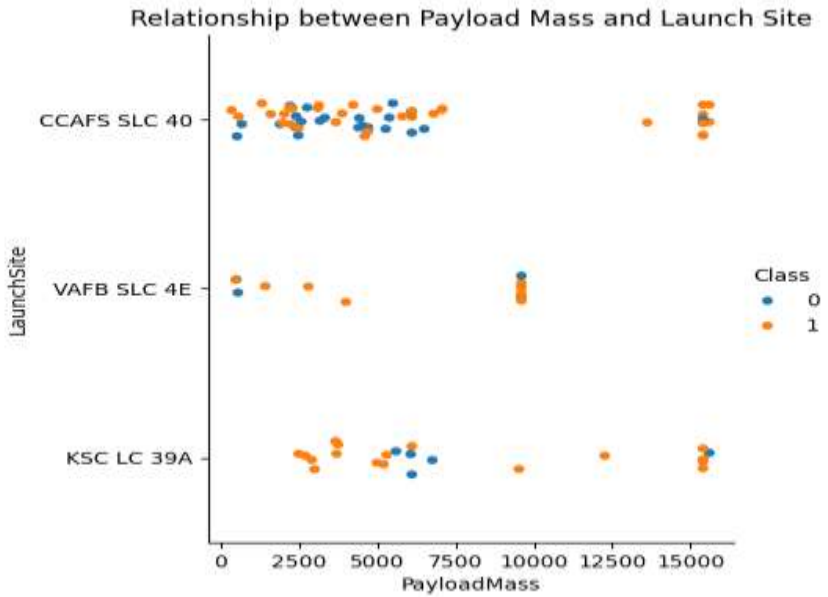
Flight Number vs Launch Site

The majority of launches are from the CCAFS SLC-40 site.



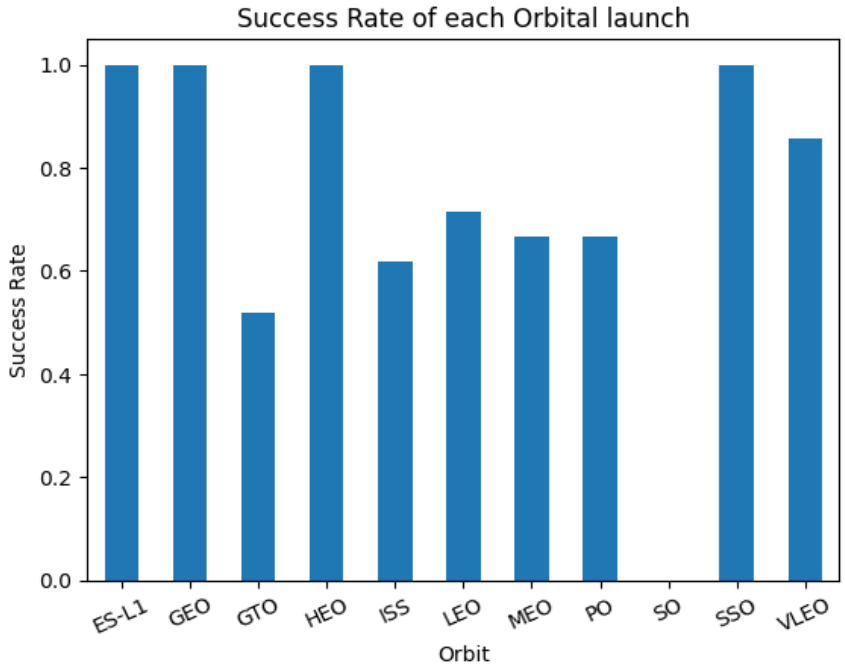
Payload vs Launch Site

If the payload mass exceeds 7000 kilograms, the probability of a successful launch significantly increases. However, there is no clear pattern indicating that the success rate is dependent on the payload mass for a specific launch site



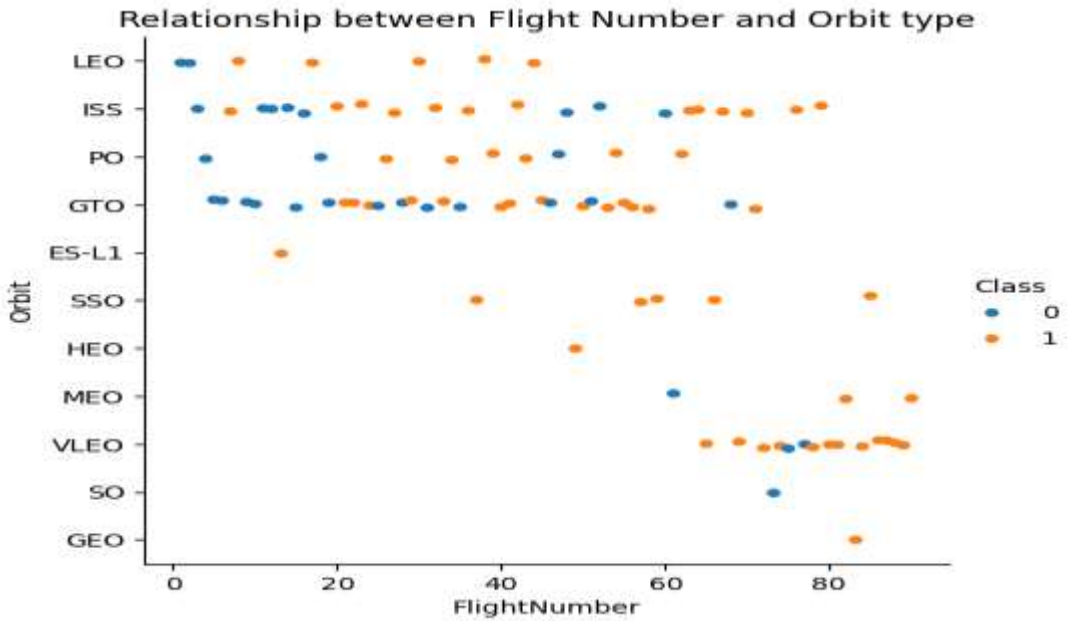
Sucess rate vs Orbit Type

Orbits ES-L1, SSO, HEO and GEO have the highest success rate



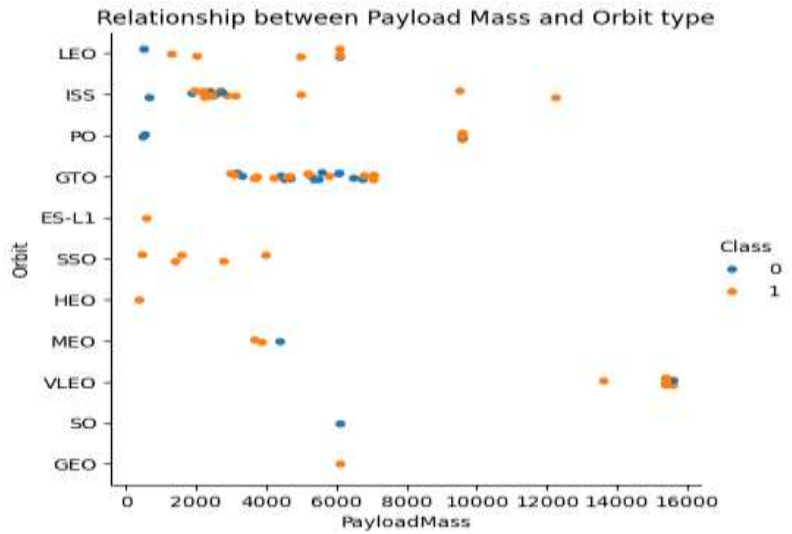
Flight Number vs Orbit Type

The graph shows that when there are more flights for each orbit, there's usually a higher chance of success, especially for the Low Earth Orbit (LEO). But, the Geostationary Transfer Orbit (GTO) is different because there's no clear connection between the number of flights and the success rate.



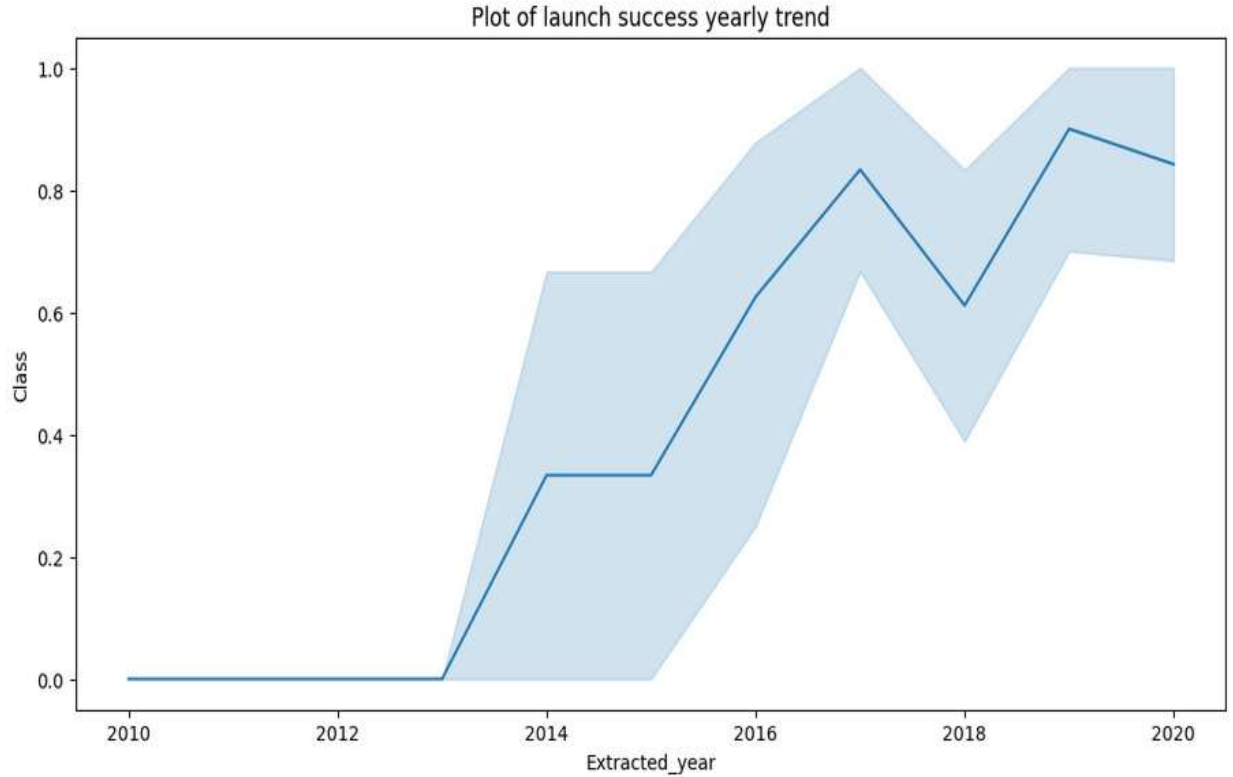
Payload vs Orbit Type

More weight in the payload is good for Low Earth Orbit (LEO), International Space Station (ISS), and Polar (P0) orbits, but it's not good for Medium Earth Orbit (MEO) and Very Low Earth Orbit (VLEO) orbits. For Geostationary Transfer Orbit (GTO), there doesn't seem to be any clear connection between the two. On the other hand, for Sun-Synchronous Orbit (SO), Geostationary Orbit (GEO), and High Earth Orbit (HEO), we need more data to figure out any patterns or trends.



Launch Success Yearly Trend

Success rate has increased since 2013



All Launch Site Names

- The term "DISTINCT" was used to showcase solely the unique launch sites found in the SpaceX data.

```
%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

Launch Site Names Begin with “CCA”

The query illustrated in the figure was employed to highlight five instances where the launch sites begin with "CCA."

```
%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
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

The combined payload transported by NASA's boosters was determined to be 45596 units .

```
%sql select Customer, sum("PAYLOAD_MASS__KG_") as 'Total Payload Mass (kg)' from spacextable group by Customer having Customer='NASA (CRS)';
```

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

Customer	Total Payload Mass (kg)
NASA (CRS)	45596

Average Payload Mass by Falcon 9 v1.1

The mean payload weight transported by booster version F9 v1.1 to be 2928.4.

```
%sql select Booster_Version, avg(PAYLOAD_MASS__KG_) as 'Average Payload Mass (kg)' from spacextable group by Booster_Version having Booster_Version = 'F9 v1.1';
```

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

Booster_Version	Average Payload Mass (kg)
F9 v1.1	2928.4

First Successful Ground Landing Date

The observation indicates that the first successful landing on a ground pad took place on December 22, 2015.

```
%sql select min(Date) as 'Date of 1st successful ground pad landing' from spacextable where Landing_Outcome = 'Success (ground pad)';
```

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

Date of 1st successful ground pad landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The WHERE clause was used to selectively isolate boosters that had achieved successful landings on a drone ship, and the AND condition was subsequently applied to determine successful landings with payload masses falling within the range of greater than 4000 but less than 6000.

```
%sql select Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_ from spacextable where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Missing Outcomes

The COUNT(*) function is applied to count the occurrences of each unique mission outcome. The result is then labeled as "Count" using the AS keyword. The FROM clause specifies that the data is retrieved from the "spacextable" table. Lastly, the GROUP BY Mission_Outcome clause organizes the results based on distinct values in the "Mission_Outcome" column, enabling a count of occurrences for each unique outcome.

```
%sql select Mission_Outcome, count(*) as "Count" from spacextable group by Mission_Outcome;
```

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

Mission_Outcome	Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

%sql, selects the columns "Booster_Version" and "PAYLOAD_MASS__KG_" from the "spacextable" table. It focuses on rows where the payload mass matches the overall maximum payload mass in the dataset, identified through a subquery using MAX(PAYLOAD_MASS__KG_)

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ as 'Max Payload Mass (kg)' from spacextable where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable);
```

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

Booster_Version	Max 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

2015 Launch Records

%sql, extracts and presents concise information from the "spacextable." It focuses on capturing details such as the 'Month,' 'Landing_Outcome,' 'Booster_Version,' and 'Launch_Site' for instances where drone ship landings were unsuccessful in the year 2015.

```
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)';
```

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

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

Ranking Landing outcome between 2010-2017

- %sql retrieves information from the "spacextable." It selects the 'Landing_Outcome' and counts occurrences, labeling the count as 'Count.' The GROUP BY clause organizes the results based on 'Landing_Outcome.' The HAVING clause filters results for entries with a 'Date' between June 4, 2010, and March 20, 2017. The results are then ordered by the count in descending order ('Count desc').

```
%sql select Landing_Outcome, count(*) as 'Count' from spacextable group by Landing_Outcome having Date between '2010-06-04' and '2017-03-20' order by Count des
```

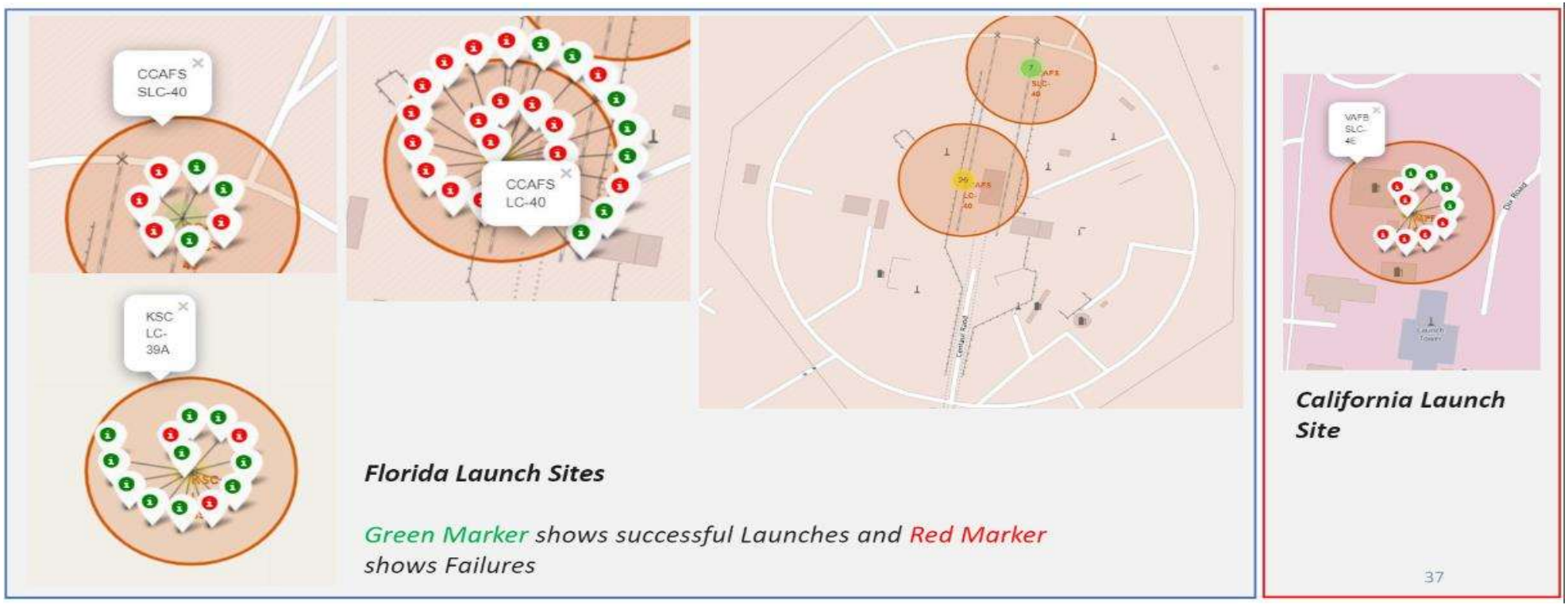
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Count
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Location Launch Sites

All SpaceX launch sites are situated within the boundaries of the United States.

Markers Showing Launch Sites with Color-Coded Labels



Launch Site Distance to Landmarks

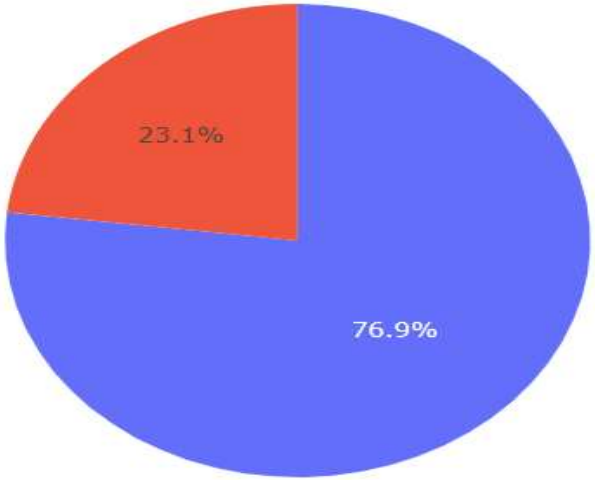
Launch sites are strategically placed near the equator, taking advantage of Earth's eastward rotation at approximately 30km/s. This rotation helps in more fuel-efficient orbit insertion Additionally, these sites are located close to coastlines for safety reasons. Launching over the ocean allows for a water landing option in case of aborts and reduces risks to people and structures from falling debris. Proximity to major highways is essential for easy movement of personnel and equipment, and being near railways helps transport bulky cargo. To minimize risks to populated areas, launch sites are intentionally situated away from major cities.

Pie chart showing the success percentage achieved by each launch site

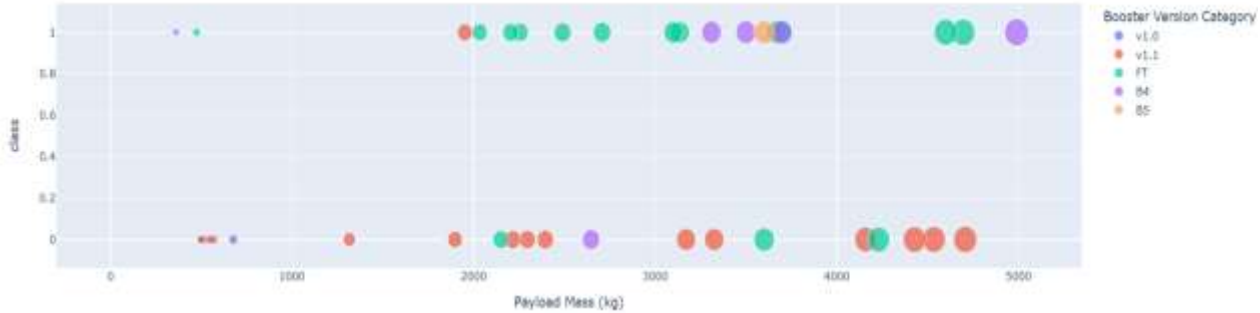
KSC LC -39A has the most success percentage

Launch Analysis of KSC LC-39A Launch Site

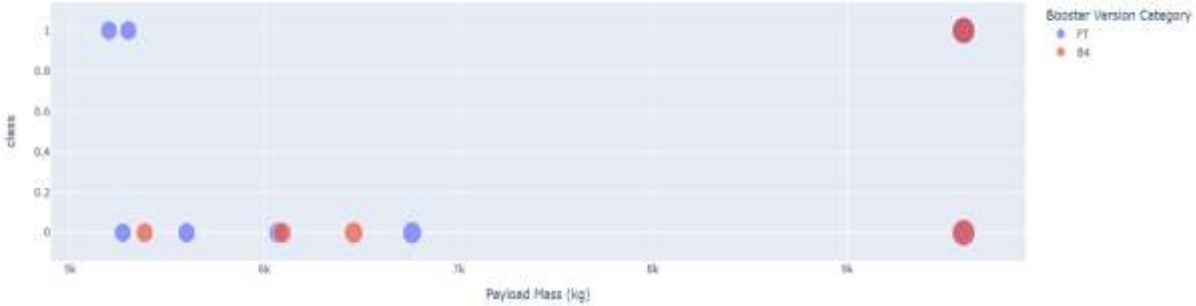
The graph shows that approximately 3 out of every 4 launches are successful.



Scatter Plot Comparing Payload to Launch Outcome



The observation reveals a higher success rate for payloads in the low weight category (0-5000 KG) compared to the heavy weight category (5000-10000 KG).



Classification Accuracy

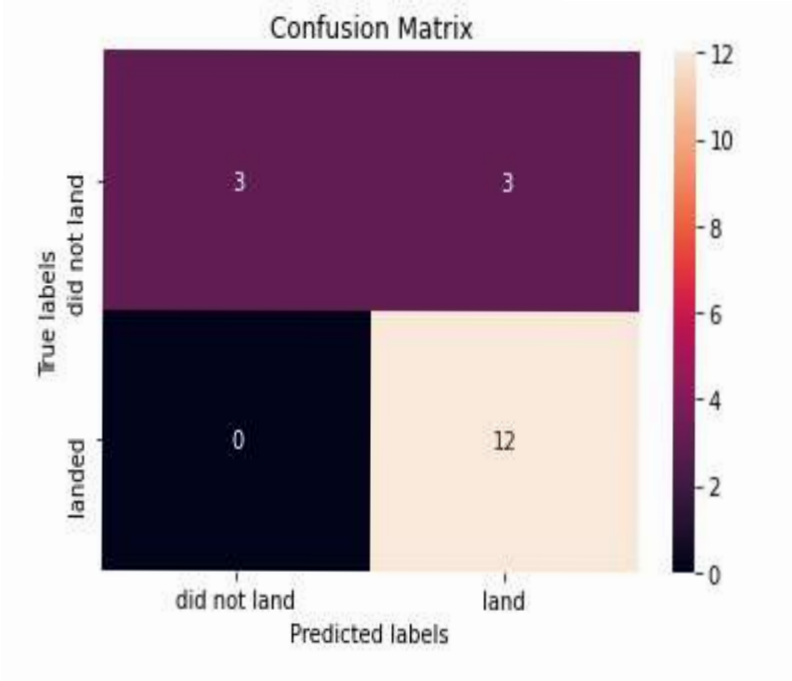
```
rankings = list(
    (name, model.score(X_test, Y_test))
    for name, model in [
        ("Logistic Regression", logreg_cv),
        ("SVM", svm_cv),
        ("Decision Tree", tree_cv),
        ("KNN", knn_cv),
    ]
)
rankings.sort(key=lambda elem: elem[1], reverse=True)

print(
    f"Model that performs the best is {rankings[0][0]} with an accuracy of {rankings[0][1]}"
)
```

```
Model that performs the best is Decision Tree with an accuracy of 0.8888888888888888
```

Confusion Matrix

- The decision tree classifier's confusion matrix indicates its ability to distinguish between classes. However, a notable issue is the occurrence of false positives, where the classifier wrongly identifies unsuccessful landings as successful ones.



Conclusion

The larger the flight amount at a launch site, the greater the success rate at a launch site.

Launch success rate started to increase in 2013 till 2020.

Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

KSC LC-39A had the most successful launches of any sites.

The Decision tree classifier is the best machine learning algorithm for this task.