Winning Space Race with Data Science

Elizabeth Birchenall
02/06/2024

Outline

EXECUTIVE SUMMARY

- The Falcon 9 is a rocket built by the company Space X. Space X is a private aerospace manufacture and space transporter from America. What makes this company different from the other aerospace companies? They have discovered that they can save millions of dollars by using the first stage of their landing procedure. In this analysis we will determine how successful the Falcon 9 is at landing in the first stage to see if this a plausible statement.
- We will be using the following stages of data analysis to answer this question: Can Space X reuse the first stage of its landing procedures with the Falcon 9 and how much is the price for each launch? The first stage of our analysis is data collection, this is where we collect all the relevant data and clean the data making sure there are no missing values. The next stage we used was data wrangling, we looked through the data we collected and determined which data had an impact to train the models with. Along with data wrangling was the web scraping where we pulled data from tables on reliable websites for any additional data we needed. We used SQL queries to collect the payload data from what the Falcon 9 sent to outer space and the landing outcomes. The next step was to visualize all the data that we collected and analyze any correlations between the different data points such as payload mass and orbit and orbit to successful landing ratio using several graphs such as bar charts and scatter plots. Finally, we used all the data we collected and analyzed to train models while observing them.
- During these stages we found a lot of important information to help us answer our questions. While visualizing the data we found that certain launch sites performed better regarding successful landing. The launch site CCAFS LC-40 had a success rate of 60% where the sites KSC LC-40 and VAFB SLC 4E had a success rate of 77%. It was also found that where the rocket goes in orbit when delivering a payload determines the success rate of landing: GEO, ES-L1, and HEO had the highest success rate. In the model stage we found that the decision tree did the best at determining the successful landing from the data it was given. We determined that it is possible for Space X to use this first stage of their landing procedures successfully under the right circumstances.

Introduction

Background:

Space X, a private American aerospace manufacturer and space transportation, advertises their Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas other providers cost up to 165 million dollars each. This is because SpaceX saves money by reusing the first stage of its landings. The amount of money that this company is saving is very promising, but much studying needs to be done to make sure that they can reuse the first stage effectively.

- **The Big Question:**

The question, we need to answer is what is the probability that these Falcon 9 landings will be successful in the first stage? We can determine this by gathering the data from different launch sites. Also viewing and studying the data regarding other aspects of the rocket and landing such as what type of landing pads the rockets are landing on and what the payload mass of the rockets are and how they impact the success of the landing.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

The project will involve collecting data from primary and secondary sources. We will first need to define the questions that need to be answered and what data we need to answer these questions. Then it is important to find this data in this case we used the SpaceX API to collect the data. Organizing the data in a new data frame is essential in being able to read the specific data you need without getting stuck on irrelevant aspects.

Perform data wrangling:

First, we need to understand and explore the data that we have gathered. Then we will need to structure the data by organizing and formatting the raw data that helps facilitate effective analysis. Cleaning the data is very important because any missing data or inaccuracies can cause unwanted results when trying to train the model. Next, we need to enrich the data by merging datasets, extracting relevant features or incorporating external data sources. Validation ensures the quality and reliability of the processed data, so this is the next step. Once again, the data needs to be checked for inconsistencies, verify the data integrity and make sure the data sticks to the predefined standards. This step helps the data analysists be confident in the accuracy of the data set and that it meets the requirements for a meaningful analysis.

Methodology

Perform exploratory data analysis (EDA) using visualization and SQL:

In this project we used SQL to explore if where the rocket took off (Launch Site) had any impact on the successful landing of the rockets. It was found that some launch sites did perform better than others which we will show in later slides. We also explored if payload mass and where the rocket went in orbit also impacted the success rate of rocket landings. As well as if the rockets landings got more successful as the years went on.

Perform interactive visual analytics using Folium and Plotly Dash:

In the Folium and Plotly Dash stage of the interactive visual analysis, we created interactive maps for stakeholders to look at the data of launch sites and successful or failed landings. We also created an interactive dashboard so that the different launch sites showed their individual statistics on successful or failed landings in regarding payload mass.

Perform predictive analysis using classification models:

The predictive analysis performed to classify the models included using supervised training models to find the most accurate model to used. The ones studied were logistic regression, support vector machine, decision tree classifier and k nearest neighbors. Most of the models had a similar accuracy reading we used a GridSearchCV to visualize how well the different models performed.

Data Collection

Define Research Question/ Objective → Data Requirements → Find the Source of the Data → Data Collection Procedure → Examination of Information

1 → 2 → 3 → 4 → 5

Define Research Question/ Objective: The first stage in data collection is defining the question that the business is trying to answer. Then set and outline the project analysis intentions and goals. This step requires communication and clarity to over come stakeholder's differing objectives, biases and information modalities. This step is necessary to avoid wasted time and resources.

Data Requirements: In the data requirement stage, where we identify the necessary data content, formats, and sources for data collection. **WHO** is the data for? Is it for the stakeholders or is it for the public? **WHERE** will we get the data from? **HOW** will we collect this data? From company data, from public records or online? **WHAT** types of data do we need? **WHY** are we collecting this data? **WHEN?** Will it be data from current statistics or from historical data?

Find the Source of the Data: The data collection sources needed vary depending on the project, goals and assets. These can be put under either primary or secondary sources. The primary data source generates raw, untouched data that is collected directly. It always pertains to newly acquired information that is captured for the intended purpose of the project. This could include customer questionnaires or surveys. The secondary data sources come from third party software like analytic tools or websites.

Data Collection Procedure: Now that the research goals and data sources are figured out, we need to work on the data collection plan. How long will it take to collect this data and what tools do you need to track, compile and report the data?

Examination of Information: Finally, its time to manage the data that has been collected. Data cleansing and presenting the data insights that you have found. Data cleansing includes removing duplicate data, detecting and correcting errors, finding and dealing with incomplete data, organizing outliers and standardizing data.

Data Collection – SpaceX API

Import relevant libraries for data collection: For this study we will need to import pandas, numpy, datetime for the data collection.

Send a request to the SpaceX API: We are looking at the past launching results for the Falcon 9 rocket. But at the beginning of the process, we just request all the data from the spacex_url using requests.get.

Normalize json file: We normalize the .json file that is received so that it is easier to read and then view the head of the data received by inputting data.head(5): This will give us the first 5 rows of the data.

Request needed data: This is when we noticed that a lot of the data were IDs rather than the names that are understandable, so we request data from the API again to get information about the launches using the IDs given for each launch. We requested information from the rocket, payloads, launchpad and cores columns, keeping only the features we want. We also needed to use the pd.to_datetime on the date column so that it was easier to read.

https://github.com/LizzySwoop/SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Request	Request to the SpaceX API
Normalize	Normalize the json file
Request	Request the needed data
Import	Import relevant libraries for data collection

Data Collection – SpaceX API

- Create a new Data Frame for the collected data
- Filter Data Frame to only include Falcon 9 launches
- Reset Data Frame shape

Create a new Data Frame (DF) for the collected data: To differentiate between the old data we have collected and the new data that we have requested we create a new data frame. We do this by first creating empty lists for the data we want to store, using the get method to retrieve the data from the old data frame and then adding the data in by creating a dictionary with the titles of the columns and the values the names as they appear in the old DF. Then create the new Data Frame by using this call: launchdf = pd.DataFrame.from_dict(launch_dict). Then print the head of the launch df to make sure that all the data that is needed is inside of the new DF.

Filter Data Frame for Falcon 9 only: Since this project is only answering questions about the Falcon 9 rockets, we need to remove all the other rockets. In this case the data for the types of rockets are listed under the column 'BoosterVersion'. To do this we drop the other values in the Booster Version column. Then setting the new data without the other versions to a data frame called falcon_9.

Reset Data Frame Shape: To make the data frame easier to read we reset the shape of the frame so that it is easier to read. Now its time to move onto the data scraping aspect of the project.

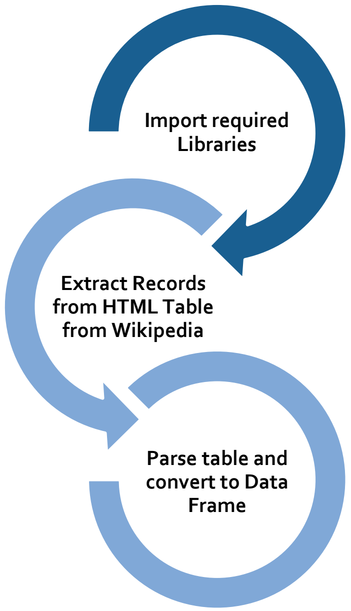
https://github.com/LizzySwoop/SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Import required libraries: To web scrap we need to use BeautifulSoup and other libraries that will help us to gather all the information needed from the website Wikipedia: sys, requests, BeautifulSoup from bs4, re, unicodedata, and of course pandas.

Extract records from HTML table from Wikipedia: Using the requests.get method from the url we provide it will give us an HTML response. We then must create a BeautifulSoup object from this response to make sure that the data was received properly. Next, we extracted all of the column and variable names from the HTML table header, then created an empty list for the column names and used the extract_column_from_header method to get our column names for our data frame.

Parse table and convert to a Data Frame: We then create an empty dictionary with keys from the extracted column names that will be converted into the data frame. Then parse the table by finding specific key letters in the HTML file to find the correct location of the values that we need. We then put these keys and values into a data frame so that this data can be used with other data from the primary source. This data from the web is a secondary source from a third party.



https://github.com/LizzySwoop/SpaceX/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Discovering: In this stage of data wrangling, we need to organize and explore the data that we have collected. Making sure that everything in the data is what we need to answer the big questions and explore the data to see if there is anything we need to fix later. This is where we call different aspects of the data to identify if all the types are correct for the data we need. So, for this project we needed to calculate the number of launches on each site, the number and occurrences of each orbit and then number of mission outcomes for the orbits.

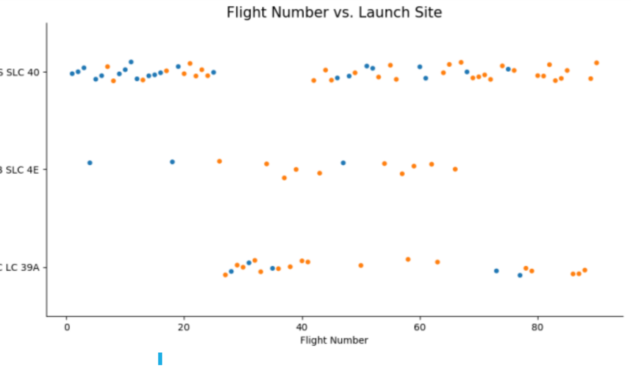
Structuring: Here is where we do the most organization and formatting to make sure that the data that we have collected will facilitate effective analysis. For the SpaceX project we used data structuring to create classification variable that represented whether the outcome of landing was a good outcome labelled as 1 or a bad outcome labeled as 0 to be able to use it with the other numerical data.

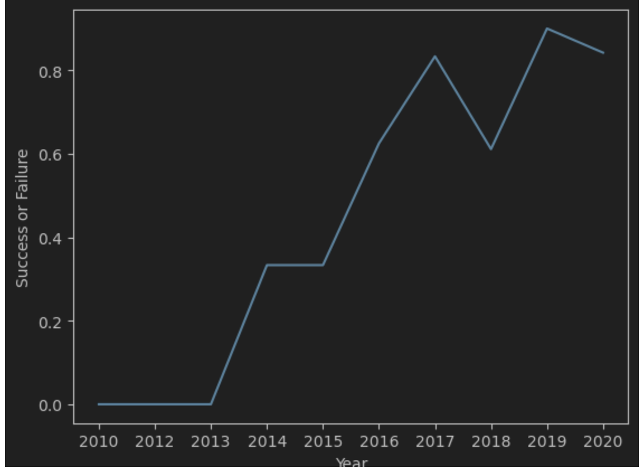
Cleansing: This is one of the most important stages of the data wrangling process. We must check for any inconsistencies or missing data. If we don't identify these issues in the data, it can cause inaccurate results. This may lead to errors when attempting to train models in later stages. In this project we removed any null values and dealt with the missing values from the payload mass column.

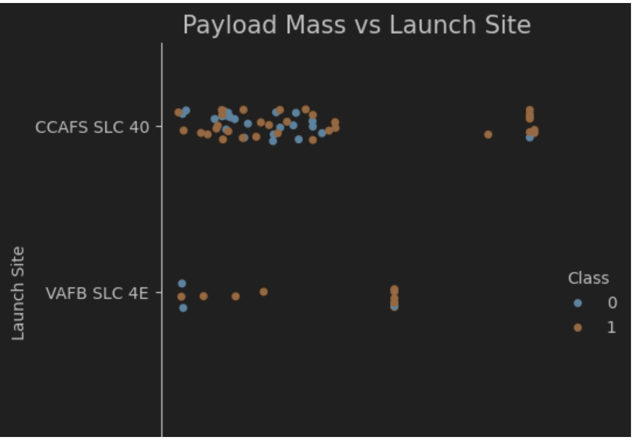
Enriching: In enriching, the people working on this project need to merge datasets and extract relevant features and incorporating the data from external sources such as the table from the Wikipedia page for this SpaceX project.

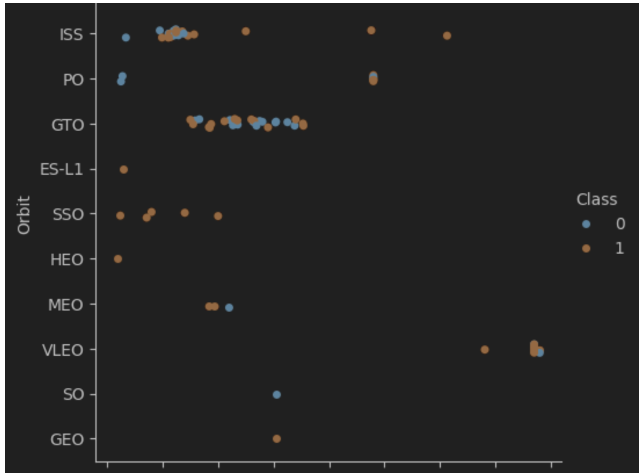
Validating: Making sure that the data that we have collected, cleansed and structured is reliable and of good quality. This ensures that the data analysts are confident that their data is as accurate as possible before moving to the visualization stage.

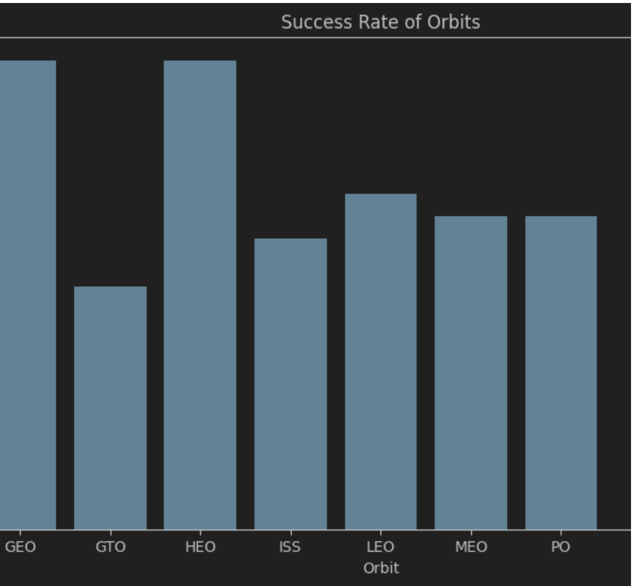
https://github.com/LizzySwoop/SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

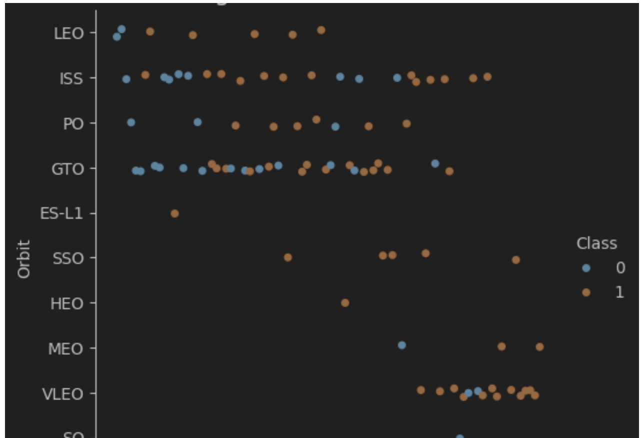












EDA with Data Visualization

The plots used for the data visualization were graphs such as scatter plots, bar charts and line graphs. These were used to better visualize the data that we wanted to explore. Most of the graphs used a hue which was set as the class or the success rate of the landing as a comparison between the two variables.

Scatterplot one (Flight Number vs. Launch Site): We created this graph to see if there was any changes in the success rate of launch sites based on the number of times rockets had been launched.

Scatterplot two (Payload Mass vs Flight number): Find a correlation between the amount of payload per number flights performed for each rocket.

Orbit Bar Chart: We then wanted to see if the success rate was determined by where the flights reached in orbit.

Orbit vs Flight number scatterplot: Does both the number of flights performed by the rocket and where the rocket reached in orbit determine the success rate of the landing?

Payload vs Orbit scatterplot: Does the amount of payload mass and where the rocket reached in orbit change the success rate?

Launch Success Rate Per Year line graph: Has the success rate improved over the years and how significantly?

https://github.com/LizzySwoop/SpaceX/blob/main/edadataviz%20(1).ipynb

EDA with SQL

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;

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

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

```
%sql SELECT MIN(Date) AS FirstSuccess FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success
 (ground pad)';

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

FirstSuccess
2015-12-22

```
%sql SELECT COUNT(MISSION_OUTCO

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

COUNT(MISSION_OUTCOME)
1
98
1
1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTBL WHERE
 BOOSTER_VERSION = 'F9 v1.1';

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

AveragePayloadMass
2928.4

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTBL WHERE (CUSTOMER)
 LIKE'NASA%';

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

TotalPayloadMass
99980

SELECT DISTICT(LAUNCH_SITE):

This was used to find all the different launch sites for evaluation later.

SELECT SUM(PAYLOAD_MASS_KG_) :

We used this query to find the payload mass for the boosters used by NASA. Result = 99980

SELECT MIN(Date):

This query was used to find when the first rocket successfully landed on a ground pad.

AVG(PAYLOAD_MASS_KG_): This was for boosters that were version F9 v1.1. Result= 2928.4

COUNT(MISSION_OUTCOME): How many landing succeeded, failed or otherwise.

LANDING_OUTCOMES: What were the landing outcomes between 2010-06-04 and 2017-03-20?

LANDING_OUTCOMES: What were the dates that the landing outcome resulted in failure in terms of drone landing in 2015?

MAX(PAYLOAD_MASS_KG_): What booster versions carried the heaviest payload mass?

https://github.com/LizzySwoop/SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

We created an interactive map using folium to view the different launch sites and the landing outcomes for each site. This was important to do so that it would easier to physically see for those who may struggle to read graphs or code. The following is the markers, circles and variables we added to the map to make it as effective as possible.

Coordinate: We inputted the information from the previously created table from the columns Lat (Latitude) and Lon (Longitude) for each launch site.

Circle: This was used to make it so that when the map was zoomed around the launch site is visible.

Marker: A marker for each element such as launch site name, we also created markers for the success and failure points for each site.

Distance_coastline: This variable was created to determine the distance from the selected launch site to the nearest coastline with a line added with a marker telling the person viewing the map the distance in km.

Lines: Line to coastline and major railway, highway and city.

https://github.com/LizzySwoop/SpaceX/blob/main/lab_jupyter_launch_site_folium.ipynb

Build a Dashboard with Plotly Dash

An interactive dashboard was created to show the correlation between all launch sites and the launch success rate.

The first chart that was created was a Pie Chart. When all sites are selected in the dropdown menu it will show the ratio of successful landings. When the sites are selected individually it will show the success and failure ratio. This is important to see so that the shareholders know which sites are the most successful.

The other element viewable on this page is the Scatter chart. Once again, when all the sites are selected in the dropdown menu it will show the payload mass for the sites and whether the landing was successful or not. When the sites are selected individually it will only show the payload mass and landing success of that site.

https://github.com/LizzySwoop/SpaceX/blob/main/spacexlaunchdashboard.py

Import Libraries → Load Dataframe → Create NumPy array → Standardize data X and set to Variable X → Train, test, split

Predictive Analysis (Classification)

For this project we used supervised machine learning to predict the success rate of Falcon 9 rockets. First, we needed to train the model by using the function: train_test_split to split the data X (dataframe columns including Flight number, payload mass, orbit, etc..) and data Y (class = success variable of either 1 = success or 0 = failure) into training and test data. Then we used the following ML methods:

Create objects for LR, SVM, Decision Tree and KNN. → Calculate Accuracy for these objects → Predict chance of false positives with confusion matrix for all objects → Determine which method (object) performed the best.

Predictive Analysis (Classification)

Logistic Regression (LR): Created variable parameters, lr for logistical regression and set the CV = 10. Then fit the logistical regression object into a GridSearchCV to find the best parameters and score. Then found the test set accuracy and created a confusion matrix. Used these methods for all other methods.

Support Vector Machine (SVM)

Decision Tree Classifier

K-Nearest Neighbor (KNN)

https://github.com/LizzySwoop/SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

RESULTS

<u>Exploratory data analysis results</u>: Here are some results we found in this part of our research.

1. Number of launches on each site.

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Number of launches that were in each orbit category

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3. The overall success rate of all Falcon 9 Rockets

```
df["Class"].mean()
```

```
0.6666666666666666
```

RESULTS

Predictive Analysis:

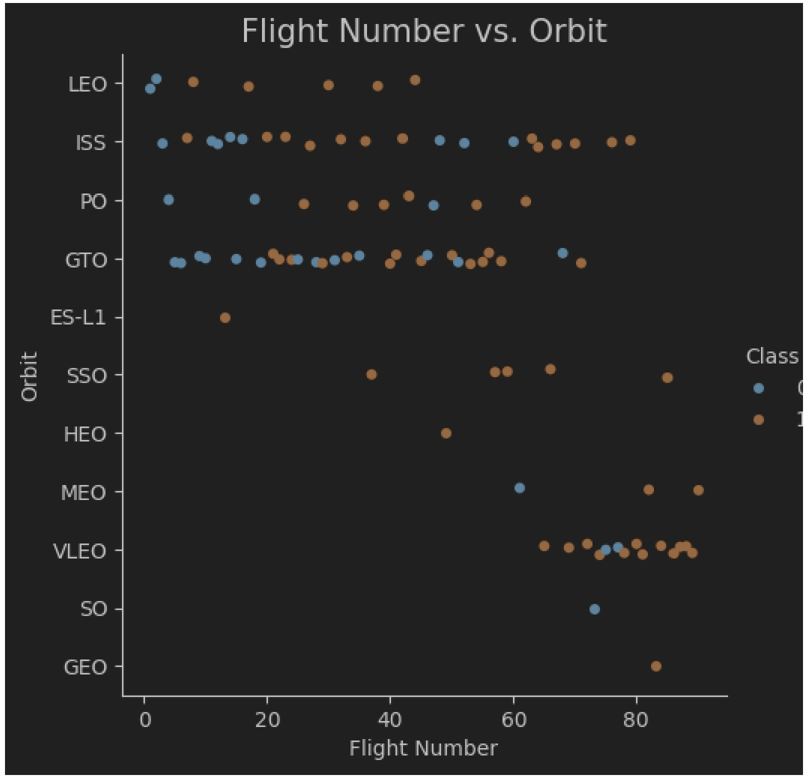
Logistic Regression Accuracy: 84.6%

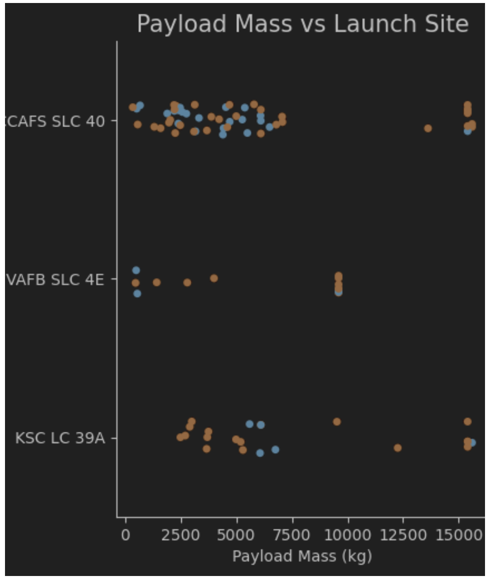
Support Vector Machine Accuracy: 84.8%

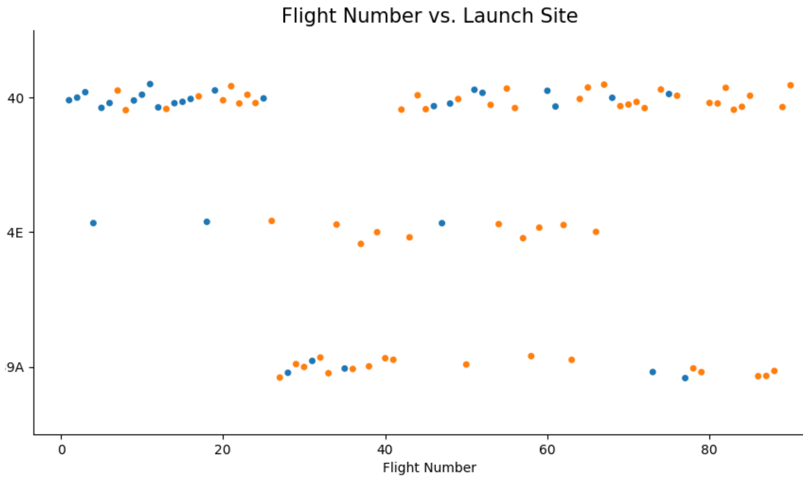
Decision Tree Classifier Accuracy: 88.9%

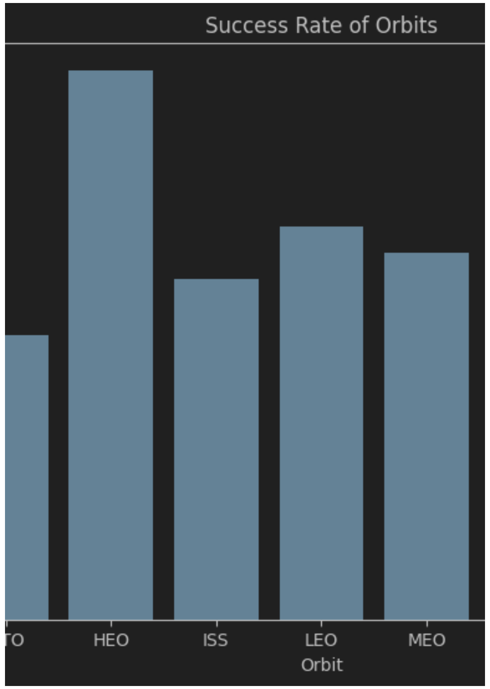
K- Nearest Neighbour Accuracy: 84.8%

The best predicting method for supervised machine learning was the decision tree with 88.9%





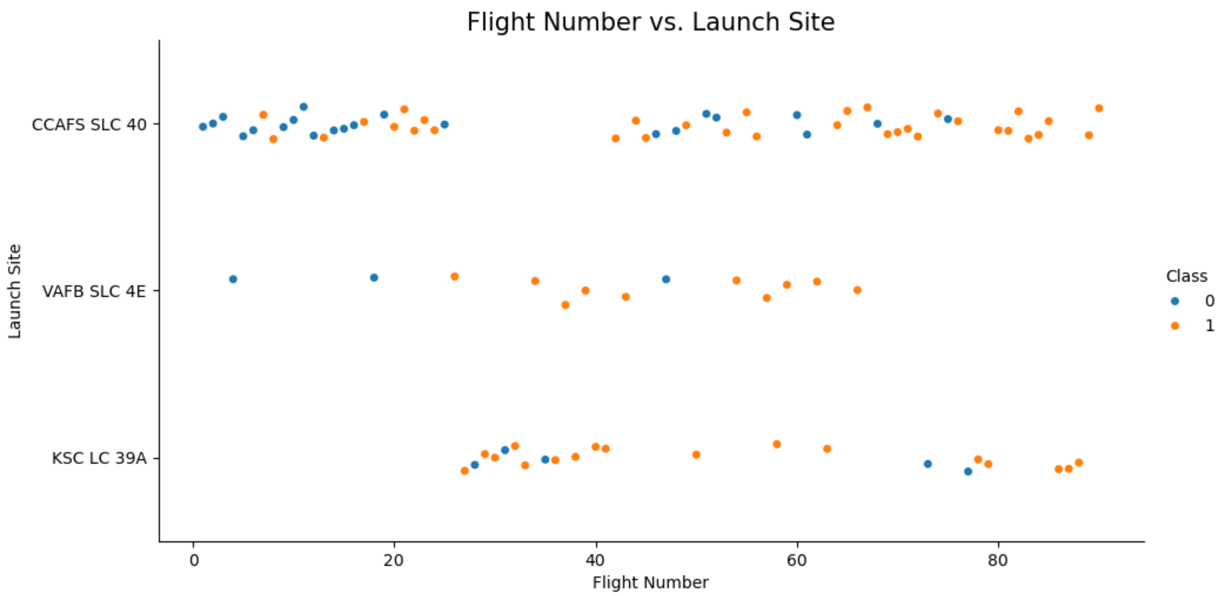




Section 2

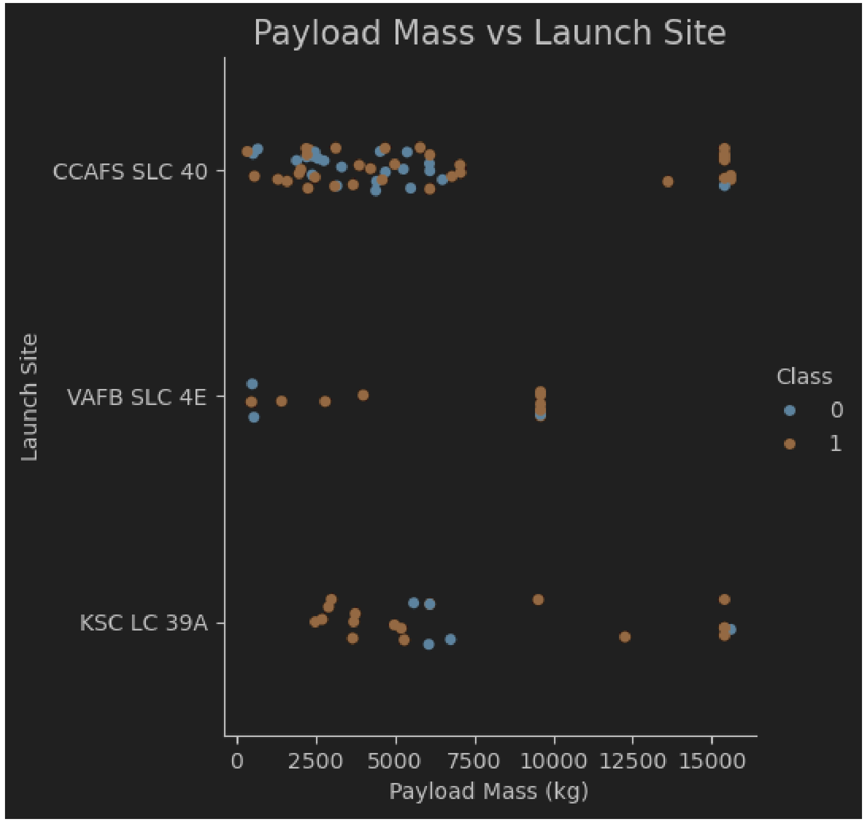
Insights drawn from EDA

FLIGHT NUMBER VS. LAUNCH SITE



In this graph you can see that the number of times a flight has been done on a rocket increases the success rate of all launch sites. There are only a few failed landing as the

PAYLOAD VS. LAUNCH SITE

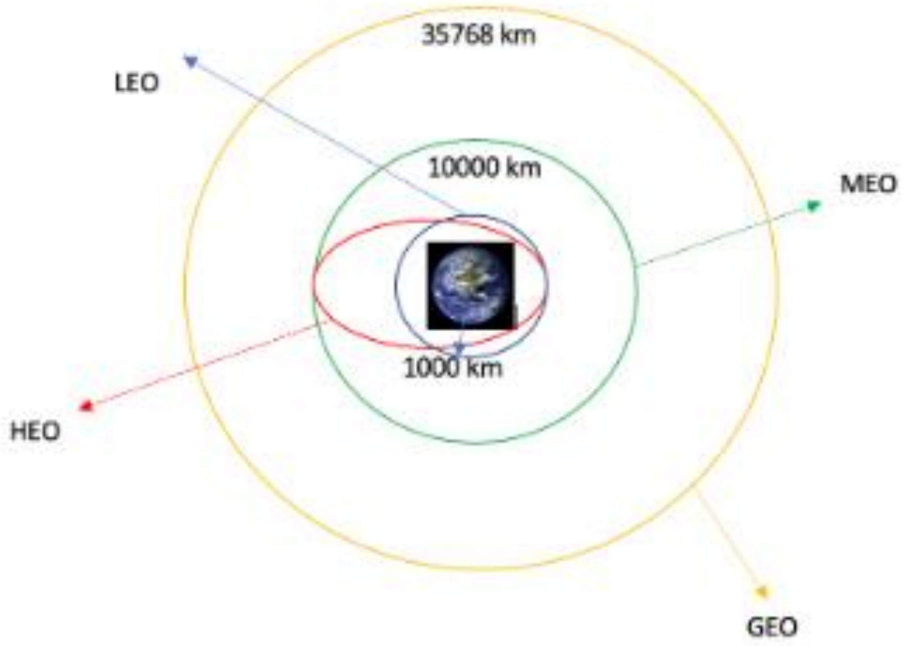


From launch site VAFB SLC 4E there are no payload launches after 10,000 kg of mass.

Both the CCAFS SLC 40 and the KSC LC 39A performed successful landings after 10,000kg but most of the launches were under this amount.

CCAFS SLC 40 carried the highest payload mass.

ORBIT TYPES



Each rocket launch aims for a dedicated orbit area around the Earth. During the analysis of these rockets, we studied the orbits data with the successful landing data as well as Payload Mass and number of flights. Here are some but not all of the most common types:

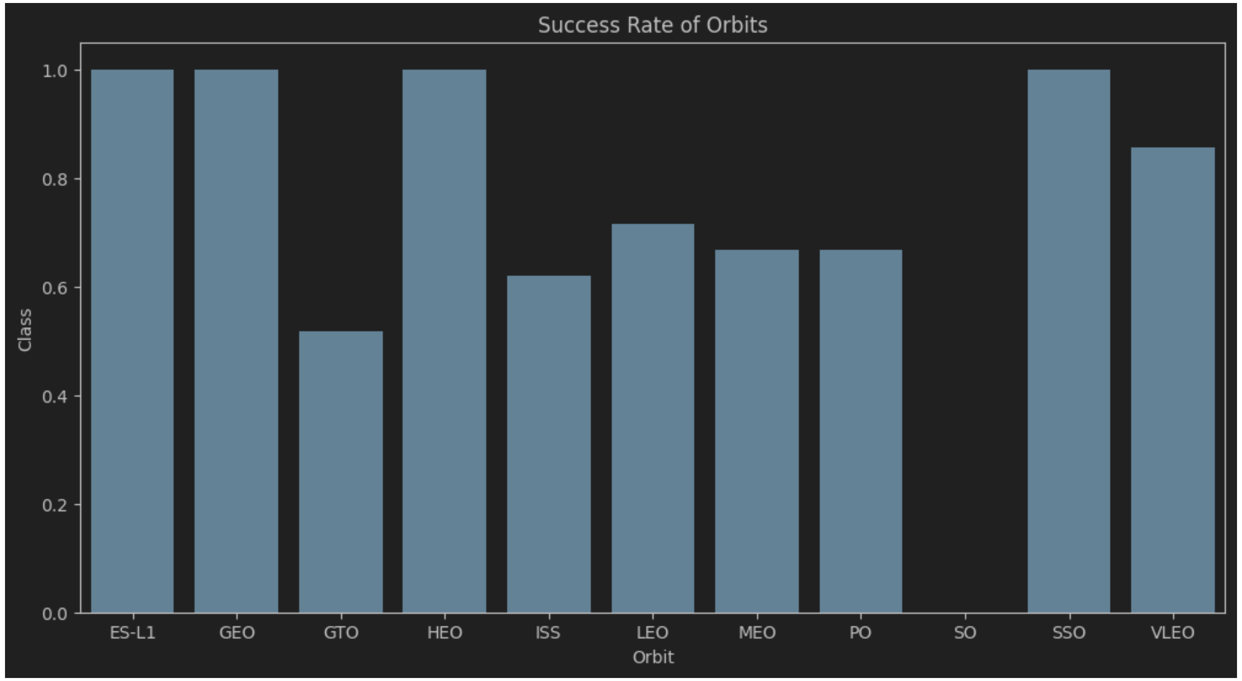
LEO (Low Earth obit): Altitude= 2,000 km or less. Most manmade objects in outer space are in LEO.

HEO (Highly Elliptical orbit): Altitude= 2,000 km to 10,000km in an oval-like shape around the Earth.

MEO: Altitude= 20,200km to 20,650km.

GEO (Geosynchronous orbit): Altitude= 35,786km.

SUCCESS RATE VS. ORBIT TYPE



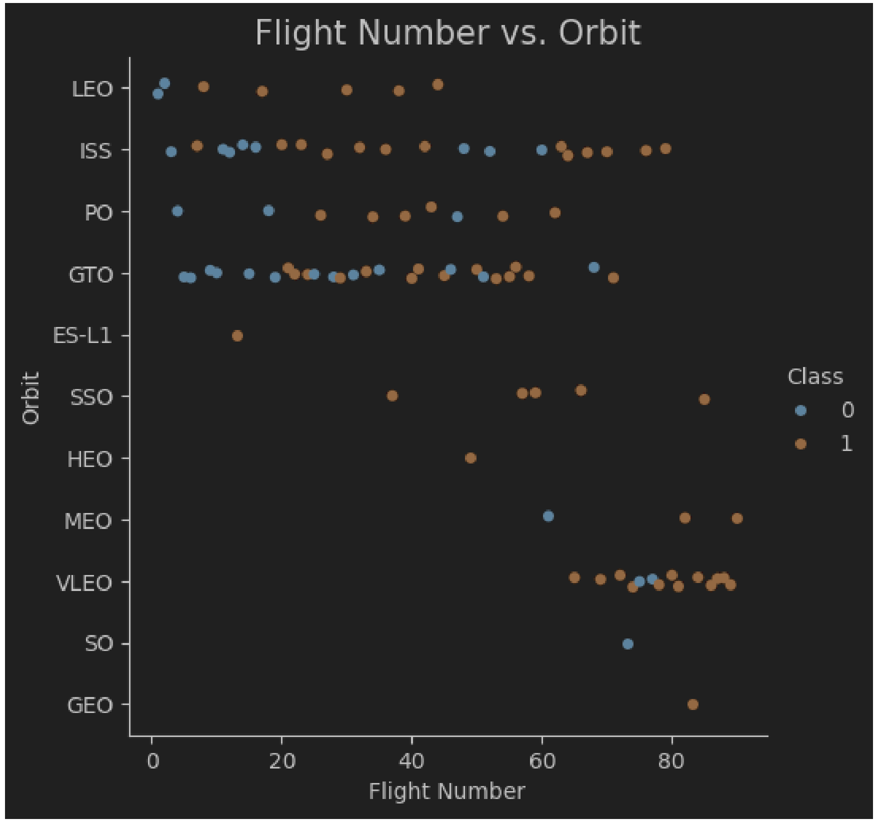
Orbits with a success rate of 100%:

ES-L1, GEO, HEO, SSO.

Lowest success rate: GTO with around 50%

The other orbits had a success rate of anywhere between 0% and 80%.

FLIGHT NUMBER VS. ORBIT TYPE

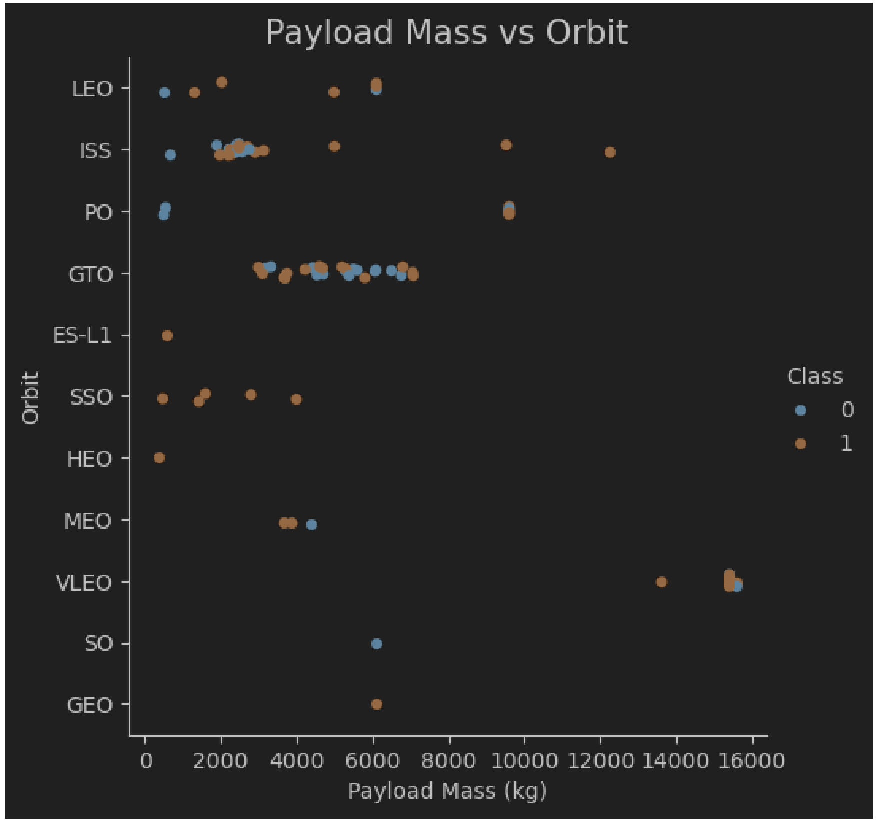


GEO, SO, HEO, MEO, and ES-L1 orbits do not have enough data regarding the number of flights to determine which number of flights would make the landing more successful.

VLEO has a better success rate with a greater number of flights than the rest.

GTO, PO, ISS, and LEO all have a good range of both successful and failed landing attempts making it difficult to determine if the number of flights and orbit impacts whether the landing will be successful but for all mentioned here but the GTO have more successful landings the higher the flight number.

PAYLOAD VS. ORBIT TYPE

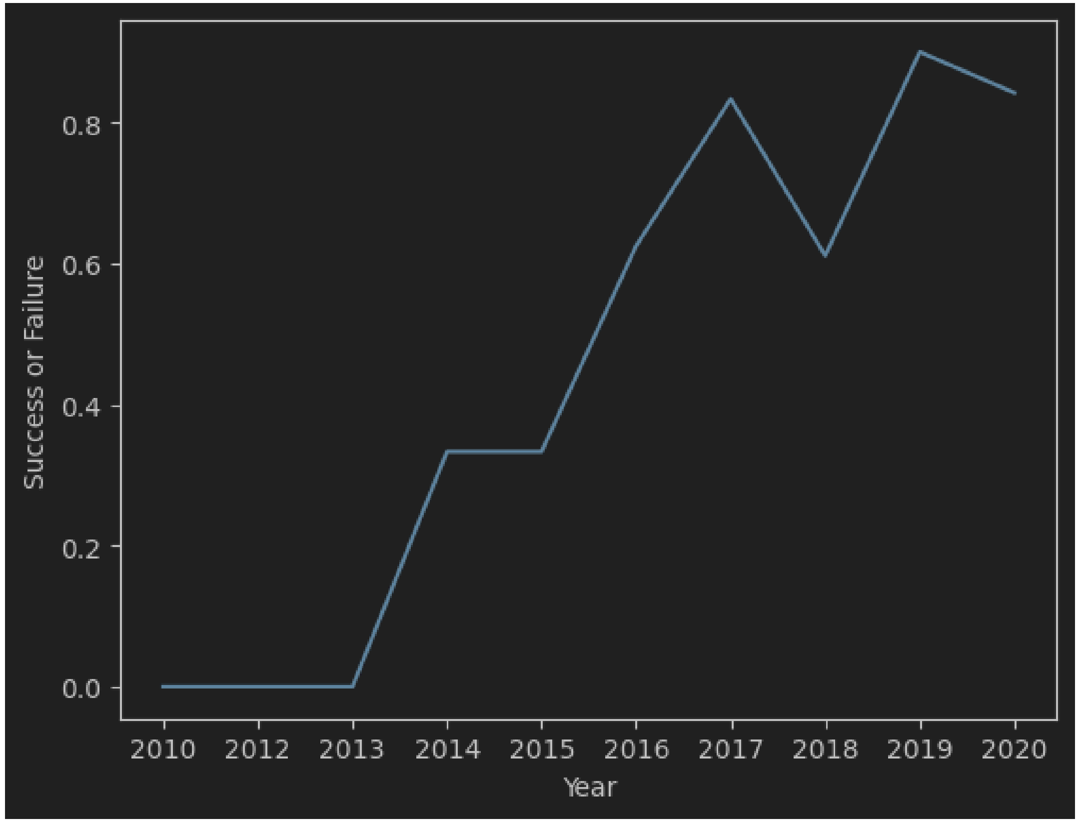


The VLEO orbit landing successfully with a Payload mass of between 14,000 kg and 16,000 kg.

The other orbit types don't really show a correlation between payload type, type of orbit and success rate.

GEO, SO, HEO and ES-L1 don't have enough data to determine if success rate is affected by these conditions.

LAUNCH SUCCESS YEARLY TREND



When the launches began it doesn’t seem like there was much of a change in the success rate of landings for 3 years.

After 2013, the landing has greatly increased to an 80-90% success rate. With a slight dip in 2018 where the success rate went down to around 60%.

ALL LAUNCH SITE NAMES

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

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

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

We then used **SQL Queries** to find the unique names of the launch sites from a table we created. This was essential for determining the success rate for each Launch Site.

LAUNCH SITE NAMES BEGIN WITH 'CCA'

```
%sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

We extracted 5 CCAFS LC-40 launch sites from the table that we created to be able to use it as part of our visualization dashboard later in the project.

TOTAL PAYLOAD MASS

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTBL WHERE (CUSTOMER)
 LIKE'NASA%';

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

TotalPayloadMass
99980

Calculate the total payload carried by boosters from NASA

Using a SQL query, we extracted the total payload mass that the boosters from NASA carried.

AVERAGE PAYLOAD MASS BY F9 V1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTBL WHERE
 BOOSTER_VERSION = 'F9 v1.1';
```

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

AveragePayloadMass
2928.4

Calculate the average payload mass carried by booster version F9 v1.1

We found the Average Payload Mass for all the rockets that used F9 v1.1 boosters.

FIRST SUCCESSFUL GROUND LANDING DATE

Here is the date of the first successful landing of a rocket on a ground pad.

```
%sql SELECT MIN(Date) AS FirstSuccess FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success
 (ground pad)';

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

FirstSuccess
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000KG AND 6000KG

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)'
 AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

TOTAL NUMBER OF SUCCESSFUL AND FAILED MISSION OUTCOMES

```
%sql SELECT COUNT(MISSION_OUTCO
 * sqlite:///my_data1.db
Done.
```

COUNT(MISSION_OUTCOME)
1
98
1
1

BOOSTERS THAT CARRIED MAXIMUM PAYLOAD

```
%sql SELECT BOOSTER_VERSION AS BOOSTERMAXPAYLOAD FROM SPA
 (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

BOOSTERMAXPAYLOAD
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

2015 LAUNCH RECORDS

```
%sql SELECT substr(Date,6,2) as Month, DATE,BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome
 FROM SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015';
```

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

Month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	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(*) as count_outcomes FROM SPACEXTBL WHERE DATE BETWEEN
 '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY count_outcomes DESC;

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

Landing_Outcome	count_outcomes
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

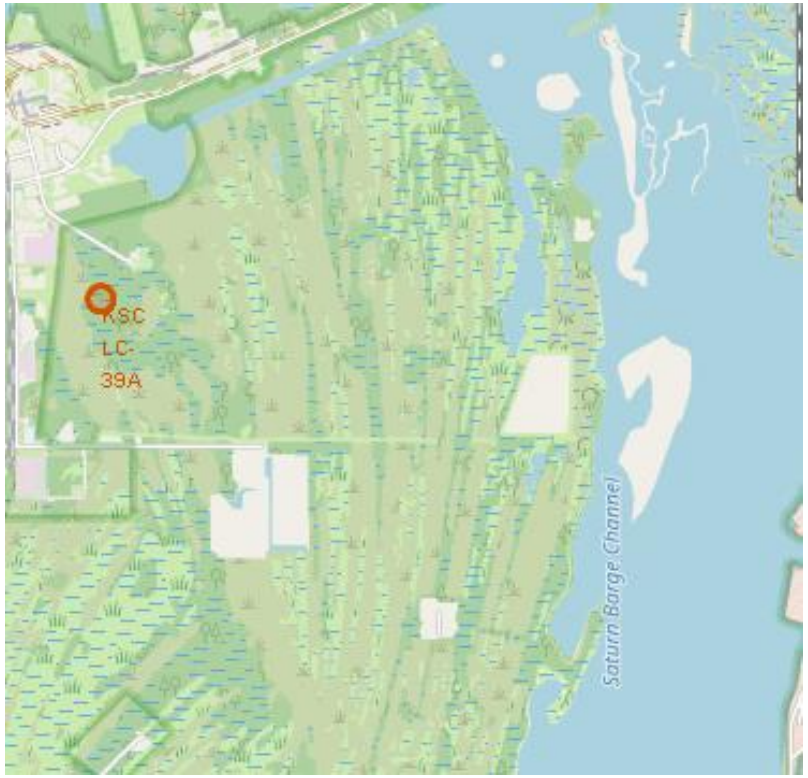
These outcomes are both the successful landing s and failed landings between the dates mentioned above.

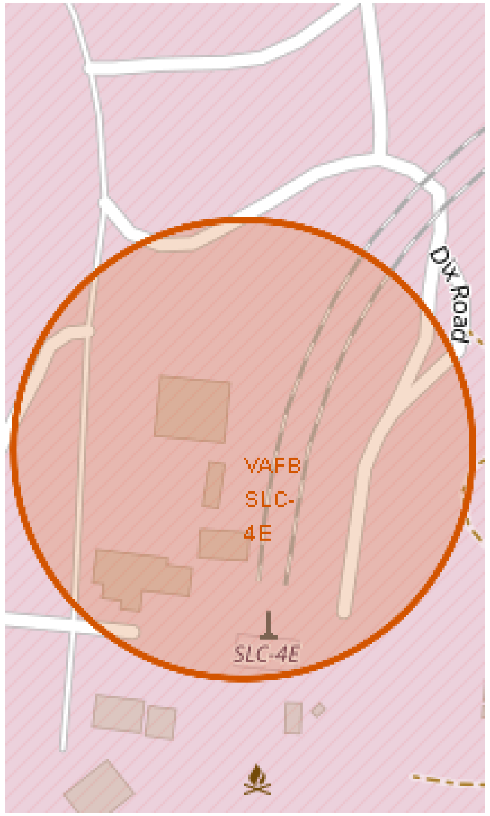
Section 3

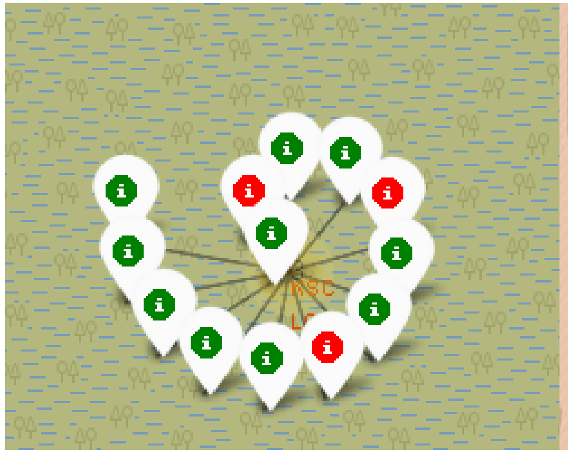
Launch Sites Proximities Analysis

LAUNCH SITE LOCATIONS

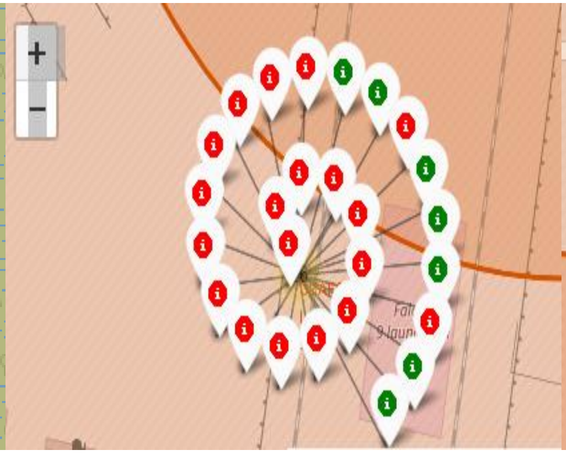
Three launch sites are in Florida while the one is in California.



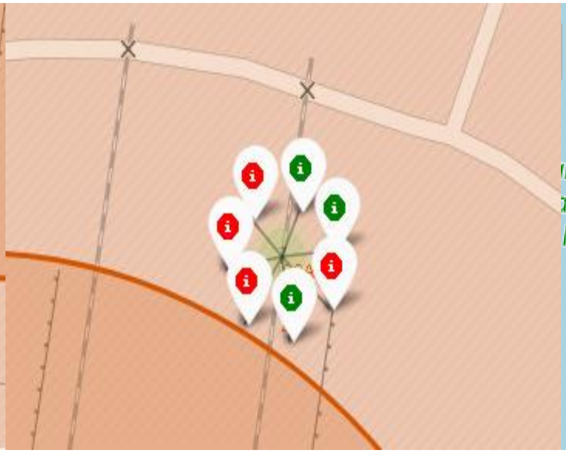




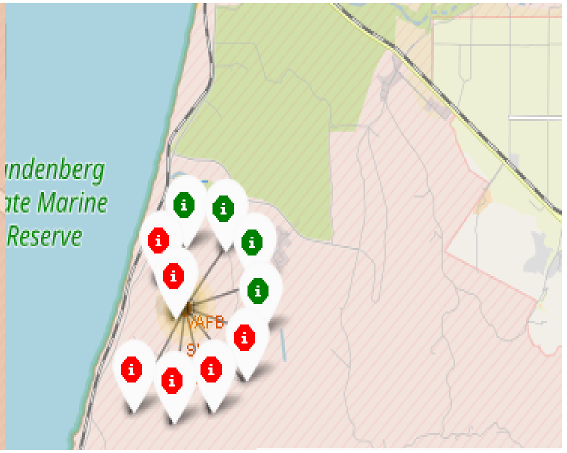
- LC- 39A Florida



CCAFS LC-40 Florida



CCAFS SLC-40 Florida



VAFB SLC 4E California

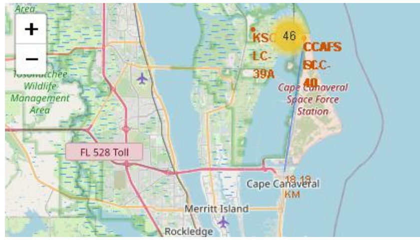
Landing Success/Failure Ratio Per Site

Proximity to Major Locations

CCAFS to coast = 0.91km

- To Major Railway = 1.27 km

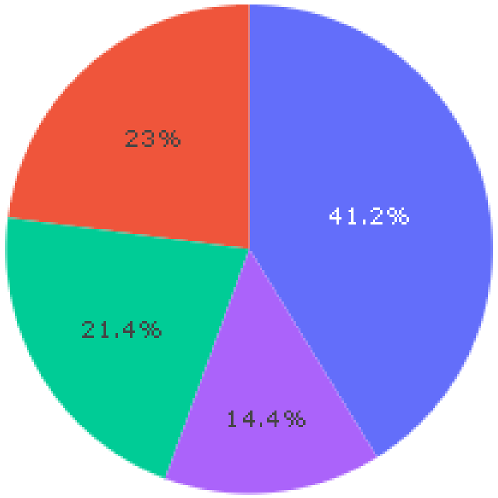
- To Major Highway = 0.59 km



- To Major City = 18.19 km

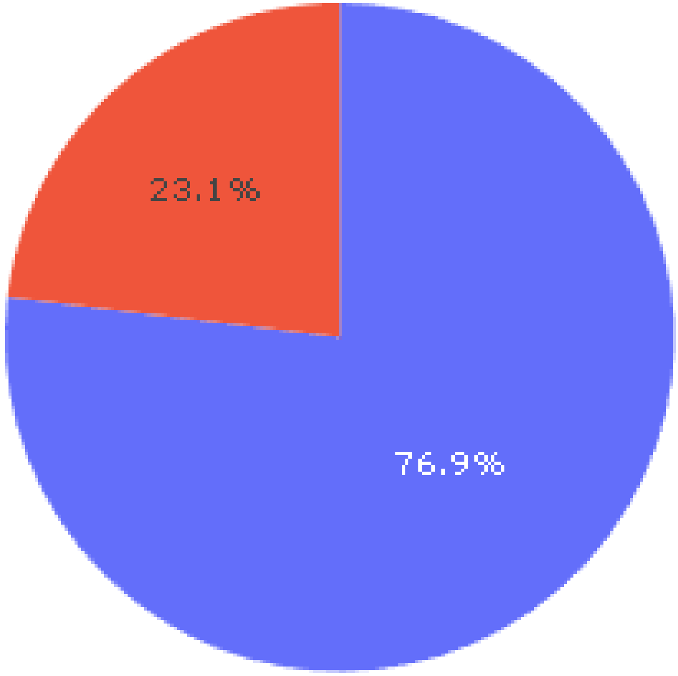
Section 4

Build a Dashboard with Plotly Dash



- KSC LC-39A had the highest success rate with 41.2%.
- CCAFS LC-40 had the lowest success rate with 14.4%.

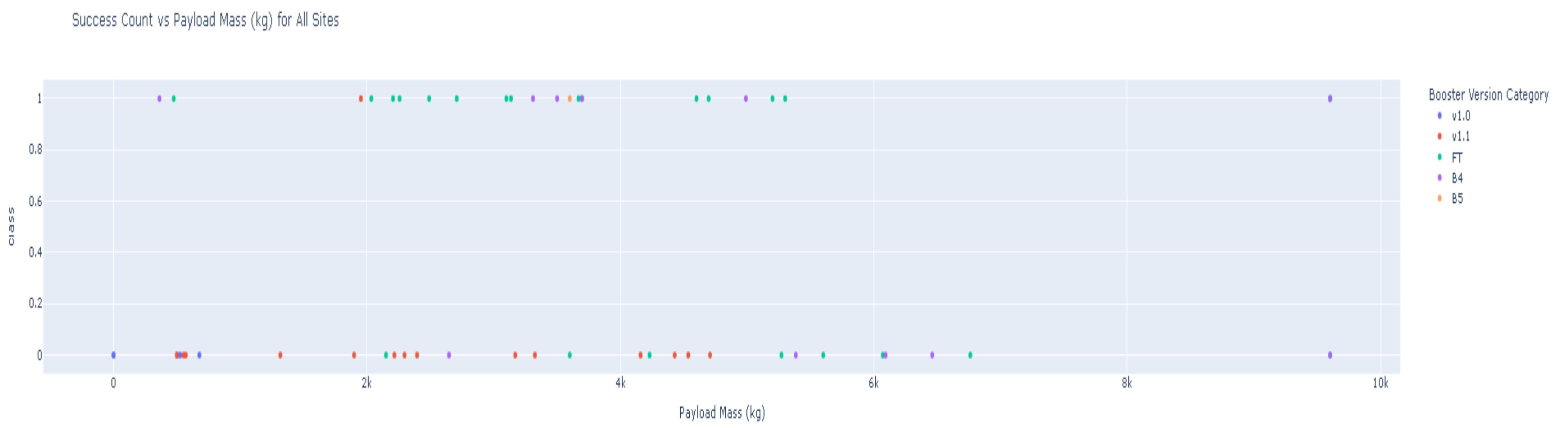
TOTAL SUCCESS FOR ALL SITES



HIGHEST SUCCESS RATE: KS LC-39A

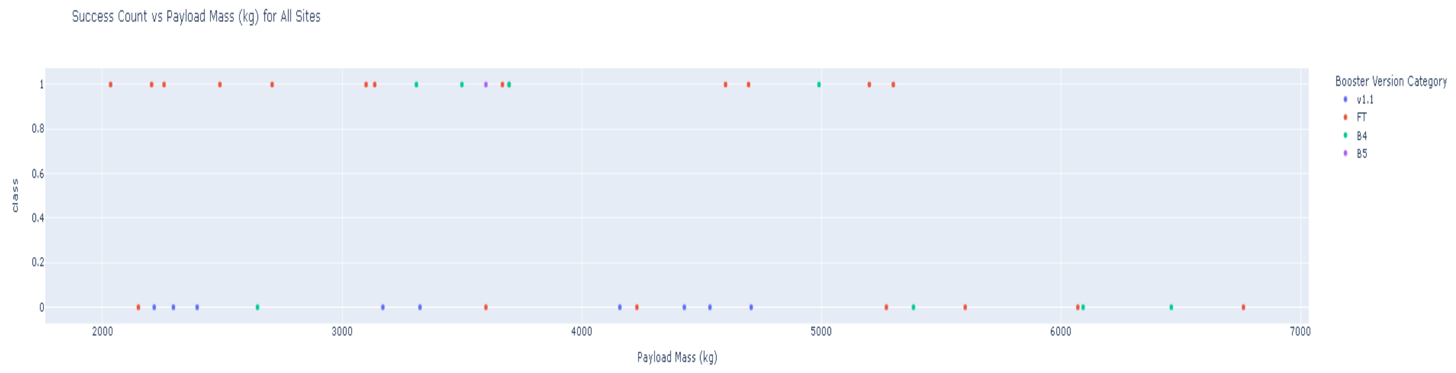
KS LC-39A had the highest success rate of all the launch sites with 76.9% to 23.1% failure.

PAYLOAD VS. SUCCESS RATE



Success rate for all sites based on Payload Mass and Booster Version. As you can see the B4 booster carried the highest payload mass with nearly 9,000 kg but had both a success and failed landing, so the data isn't enough to determine if there is a correlation.

PAYLOAD 2,000 KG – 7,000KG VS. SUCCESS RATE

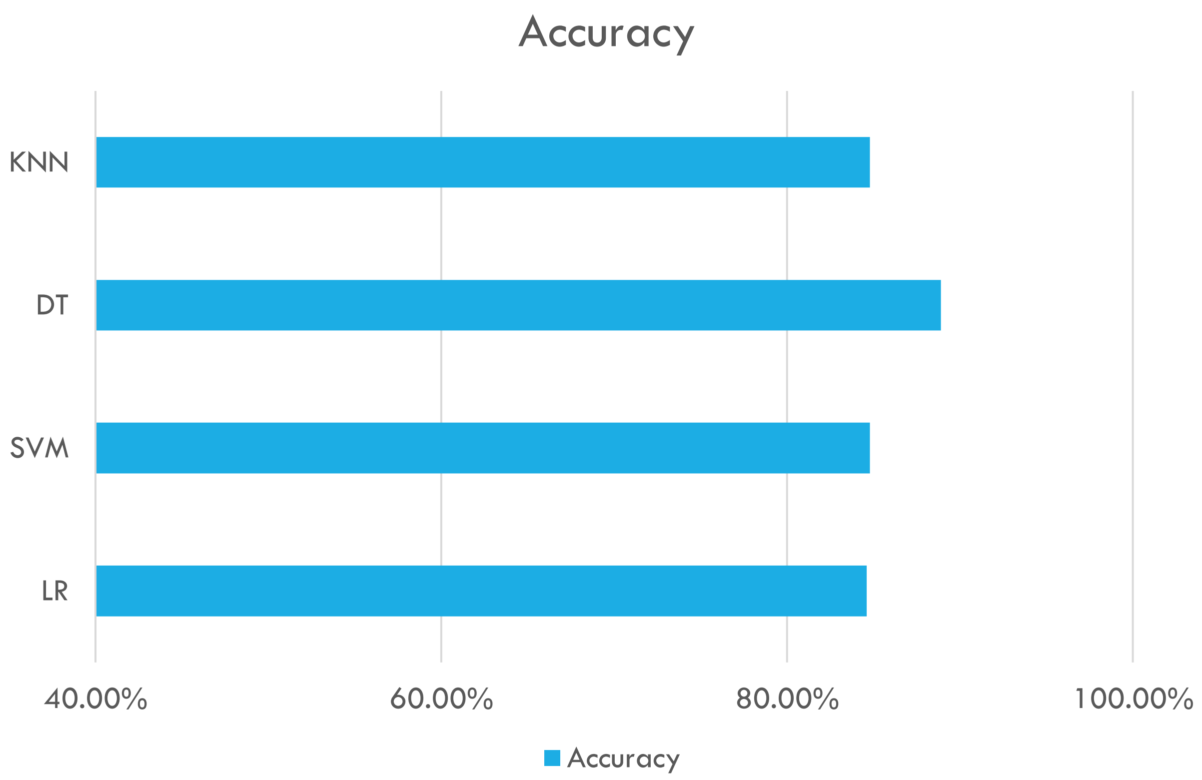


Success rate for all sites based on Payload Mass between 2,000kg and 7,000kg and Booster Version. The B4 booster had a success rate of 100% at 5,000kg.

Section 5

Predictive Analysis (Classification)

CLASSIFICATION ACCURACY

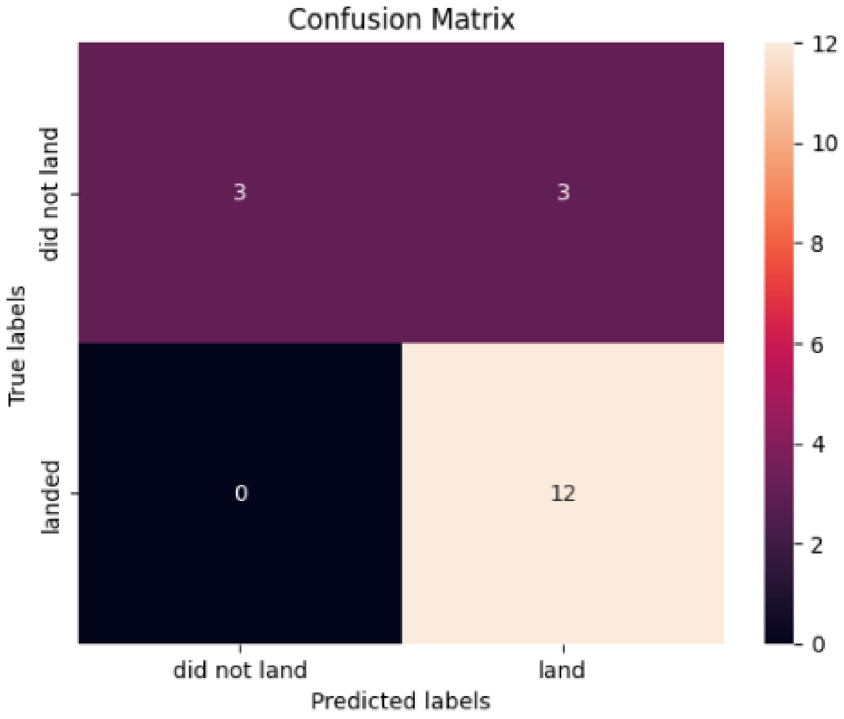


As you can see from this bar chart the Decision Tree Classifier did the best job of determining the success rate of landing.

CONFUSION MATRIX

The Decision Tree Confusion Matrix:

Correctly identified 12 of the 12 successful landings and incorrectly identified 3 that did not land but identified 3 that did not land correctly.



CONCLUSIONS

We determined that it is possible for Space X to use the first stage of their landing stage successfully.

KSC LC-39A was the most successful launch site in terms of landings.

The success rate of the first stage has continued to increase over the years.

The decision tree is the best supervised machine learning method to use with an 88.4% accuracy rating.

SpaceX is using innovative methods to reduce price in spacecraft manufacturing and delivery.

APPENDIX

Identify which columns are numerical and categorical:

```
df.dtypes
```

```
FlightNumber        int64
Date               object
BoosterVersion     object
PayloadMass       float64
Orbit              object
LaunchSite         object
Outcome            object
Flights             int64
GridFins             bool
Reused               bool
Legs                 bool
LandingPad         object
Block             float64
ReusedCount         int64
Serial             object
Longitude         float64
Latitude          float64
dtype: object
```

```
Y = data['Class'].to_numpy()
Y
```

```
array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1], dtype=int64)
```

- API = Interface for accessing relational databases
- Pandas – Data structures (Scientific Computing Library)
- Numpy- Arrays and matrices
- Matplot Lib- Visualization
- SQL- Query data. Used for relational databases.

Thank you!