IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniel Smith
August 24th, 2024

Outline

01 Executive Summary

02 Introduction

03 Methodology

04 Results

05 Conclusion

Executive Summary

Summary of Methodologies:

The following steps were followed in the production of this project:

1. Data Collection
2. Data Wrangling
3. Exploratory Data Analysis
4. Interactive Visual Analytics
5. Predictive Analysis (Classification)

Summary of Results:

This project produced the following outputs and visualizations:

- Exploratory Data Analysis (EDA) results
- Geospatial analytics
- An interactive dashboard
- Predictive analysis of classification models

Introduction

SpaceX launches Falcon9 rockets at the cost of about $62 million USD. This is relatively cheap compared to other space exploration agencies, which typically spend ~$165 million per rocket, due to SpaceX's advanced technologies, allowing for the re-use of the first-stage of the rocket, rather than its destruction.

The aim of this project is to predict whether the first-stage rocket will successfully land, how it affects the cost of the launch, and whether it's feasible for another company, such as SpaceY, to try and compete against SpaceX in the space race.

Simply, this project aims to predict if the SpaceX Falcon 9's first-stage will land successfully.

Section 1

Methodology

Methodology

1. **Data Collection**
   - Making "GET" requests to the SpaceX REST API
   - Web scraping with Python library "Beautiful Soup"
2. **Data Wrangling**
   - Use of Pandas and .value_counts() to quickly measure the:
     - Number of launches per launchpad
     - Number of spacecraft in each type of orbit
     - Number of mission outcomes (landings) per orbit type
   - Creation of a binary function for labelling outcomes:
     - "0" for boosters that failed to land
     - "1" for boosters that landed successfully
3. **Exploratory Data Analysis**
   - Use of SQL Magic queries in Jupyter Notebooks to manipulate and evaluate the extracted SpaceX dataset
   - Use of Python libraries "Pandas" and "Matplotlib" to visualize relationships between independent variables, and to assess and determine quantitative and qualitative patterns.

Methodology (continued)

4. Interactive Visual Analytics

- Geospatial analytics using Folium
- Creation of an interactive dashboard using Plotly Dash

5. Data Modelling and Evaluation

- Use of Scikit-Learn to:
    - Pre-process and standardize collected data
    - Split the data into training and testing sets for cross validation
    - Training of varying classification models
    - Determination of hyperparameters using GridSearchCV
- Plotting of confusion matrices for each classification model
- Assessment of accuracy for each classification model

Data Collection

Data sets were collected from Space X's Wikipedia page (https://en.wikipedia.org/wiki/SpaceX) using common data wrangling practices with Python.

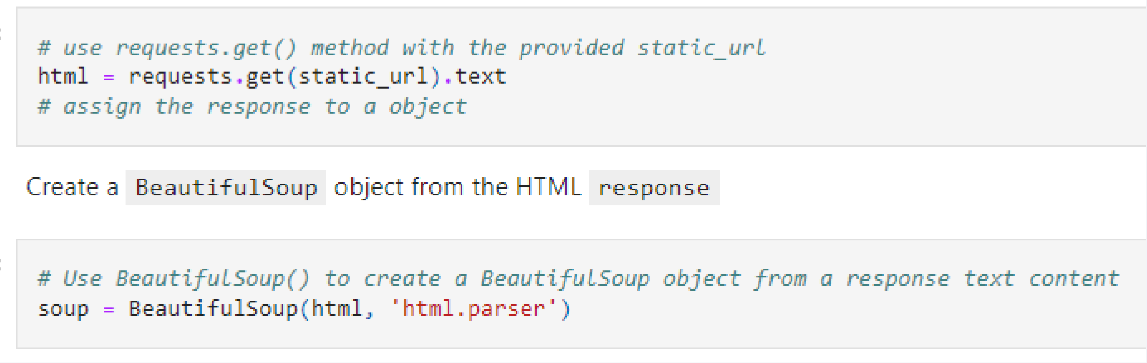
After data was extracted, the tables within the page were analyzed and converted into data frames to be used with Python's data analysis library, "Pandas".

Pandas was used further in the data cleaning process.

Data Collection – SpaceX API

- The SpaceX API was used to retrieve data about launches, including the type of rocket used, if the payload was delivered, and other specifications

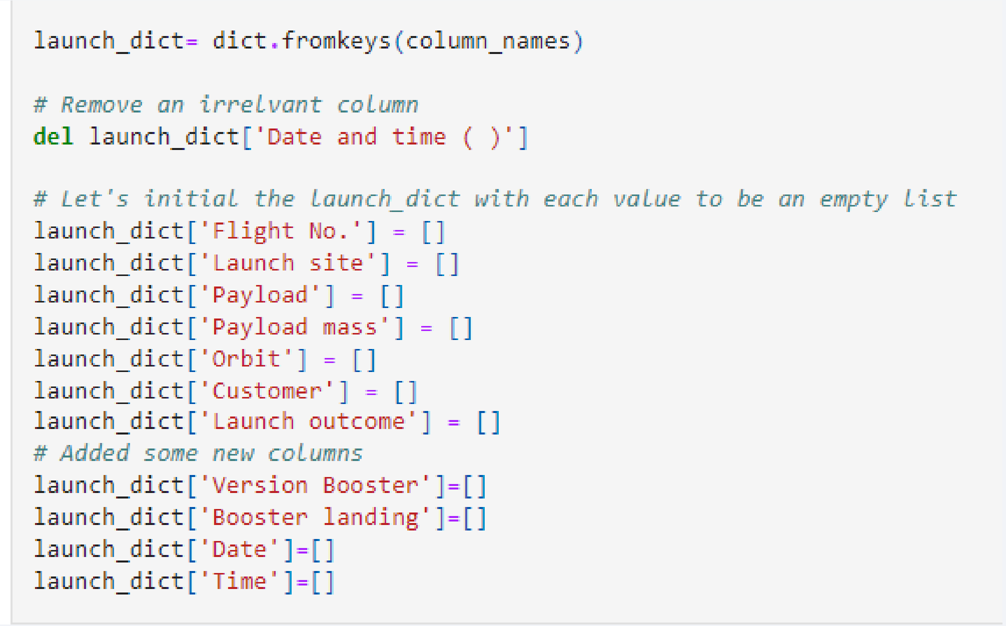
- View the full code on GitHub here



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

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

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



```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

The SpaceX dataset included information on launches originating from multiple varied SpaceX launchpads. Each launchpad was designed to send rockets to varying orbit levels, shown in the 'Orbit' column of the data frame.

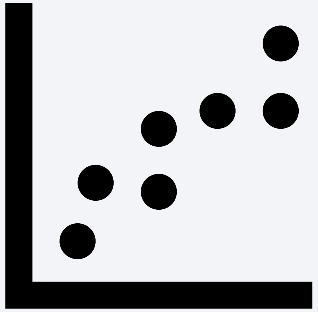
Initially, Pandas was used to explore the data in the data frame, using the value_counts() function to explore the:

- Number of launches per launchpad
- Number and occurrence of each orbit
- Landing outcome for each orbit type

You can view the full process in this GitHub link

EDA with Data Visualization

- **Scatter Charts**
  - Scatter charts are used to show the relationships and correlation values between two quantitative variables. These variables should be continuous and can be independent or dependent variables.
  - These charts were plotted to visualize the relationship between variables, such as:
    - Flight Number and Launch Site
    - Payload and Launch Site
    - Orbit Type and Flight Number
    - Payload and Orbit Type



EDA with Data Visualization

- **Bar Charts**
  - Bar charts are often used to measure and compare a numerical value and a categorical variable. Bar charts can be visualized horizontally or vertically.
  - In this project, a bar chart was used to visualize the relationship between "Success Rate and Orbit Type"
- **Line Charts**
  - Line charts are used when both axis are numerical, continuous values.
  - In this project, line charts were used to visualize the relationship between "Success Rate and the Launch Year"

You can check out the full code here

EDA with SQL

To gather more information about the dataset, SQL queries were formed using SQL magic functions in the Jupyter Notebook.

SQL Queries were used for some of the following:

- Displaying names of unique launch sites throughout the dataset
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying the average payload mass carried by booster version F9 V1.1
- Queried the failed landing outcomes on drone ships in 2015, and displayed their associated booster version and launch site

Build an Interactive Map with Folium

Folium was used in this project to visualize the launch data on an interactive map.

The steps taken include:

1. Marking all launch sites on a map
   - The folium map was initialized, and a circle and marker was placed on the map.
2. Indicating the successful and failed launches for each launchpad
   - As multiple launches were done from the same launchpad/sites, cluster groups were used organize and cleanly and clearly display the data to the user.
3. Calculation of the distances between launch sites
   - Calculation of these distances was made possible by the recording of various latitude and longitude values.

Build a Dashboard with Plotly Dash

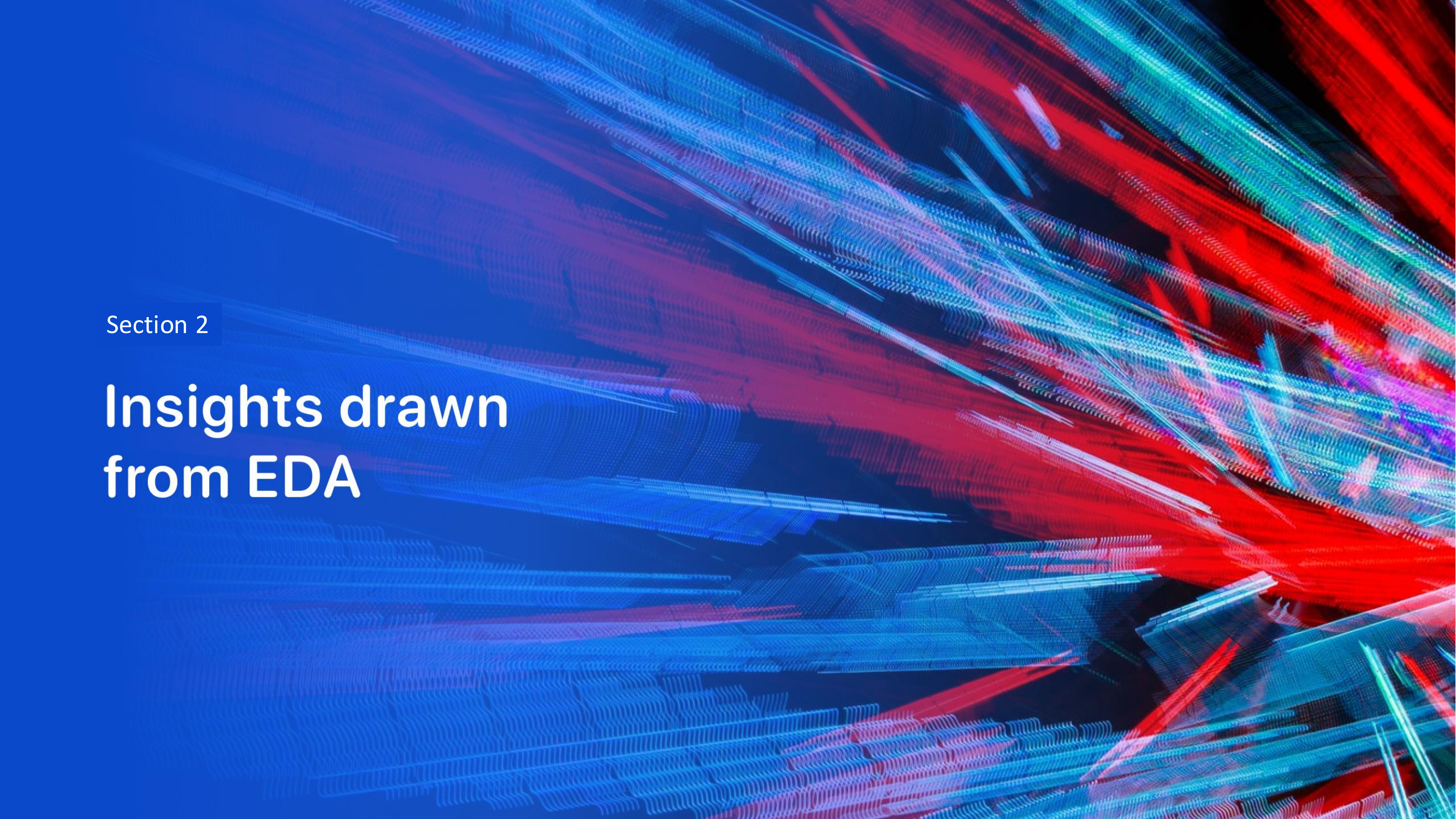
2 plots were built using Plotly Dash, to create an interactive visualization dashboard of the data.

1. **Pie Chart** showing the total successful launches per site
   - This makes it clear to see which sites are the most successful
   - This chart could also be filtered to include/exclude sites depending on user needs
2. **Scatter Graph** to show the correlation between the outcome (successful or not) and the payload mass (measured in kg)
   - This could also be filtered to show ranging payload masses
   - This could also be filtered depending on the boosters used during the launch.

Predictive Analysis (Classification)

The classification model process can be broken into the 3 following steps:

1. **Model Development**
   - The dataset was prepared for evaluation by being standardized, split into training and testing sets, and observation to determine the appropriate machine learning algorithm to apply.
2. **Model Evaluation**
   - For each chosen algorithm, the hyperparameters were tuned to increase the accuracy as much as possible
   - Results were plotted in a confusion matrix to determine their accuracy and precision
3. **Finding the Best Classification Model**
   - The accuracy and precision of each model were compared to find the best machine learning model to fit the dataset



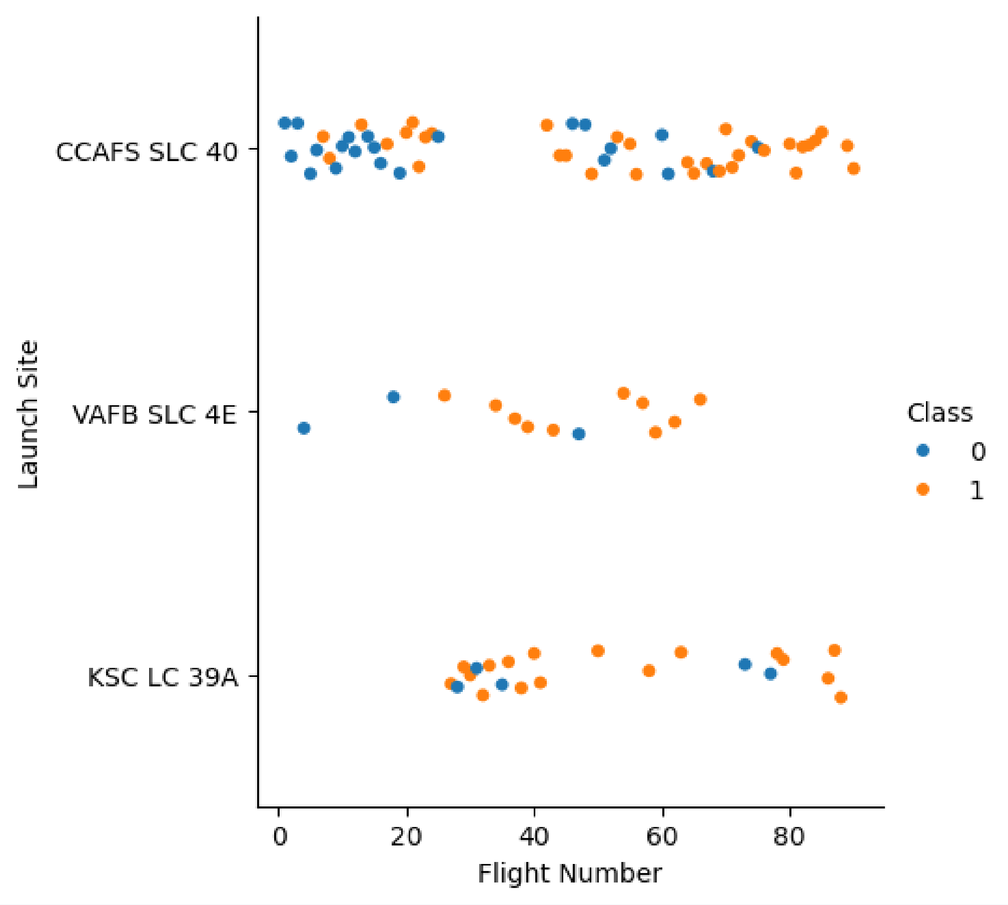
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

The scatter plot of Flight Number and Launch Site can be seen to the right.

From this graph, it can be inferred that more flights from each launch site resulted in a higher level of success, with "0" or a blue dot indicating a failure, and "1" or an orange dot representing a success.

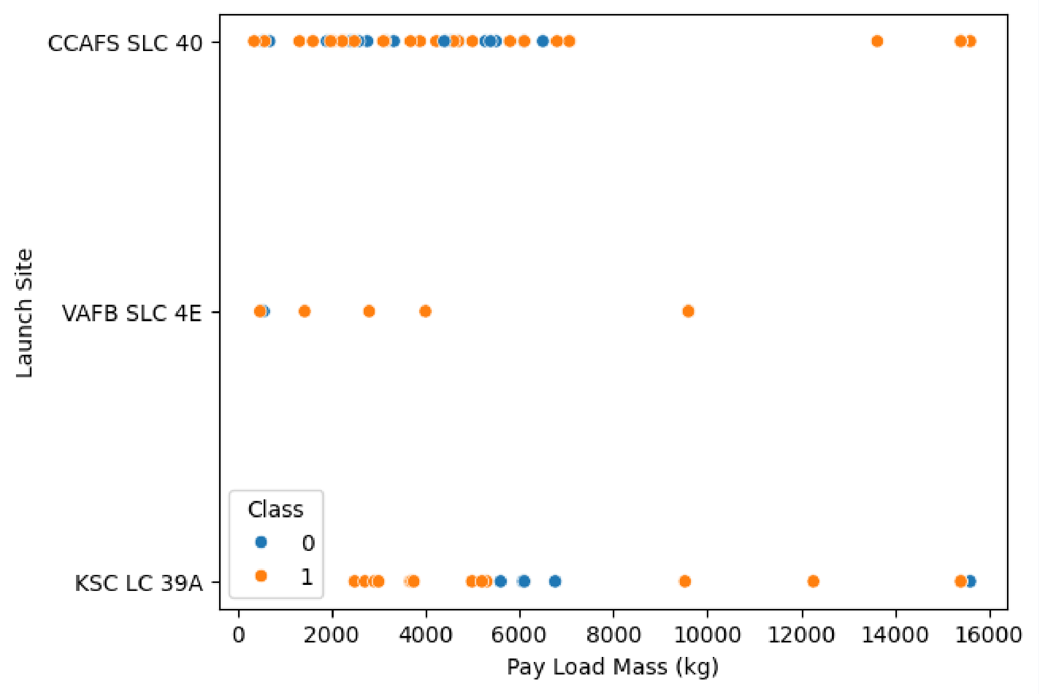


Payload vs. Launch Site

The scatter plot comparing Payload and Launch Site can be seen to the right.

From this graph, payloads with a mass over 7000kg have a very high chance of landing successfully. However, this knowledge is limited due to the low number of launches above this weight.

Overall, it seems that weight does not play a large role in the success of a landing.

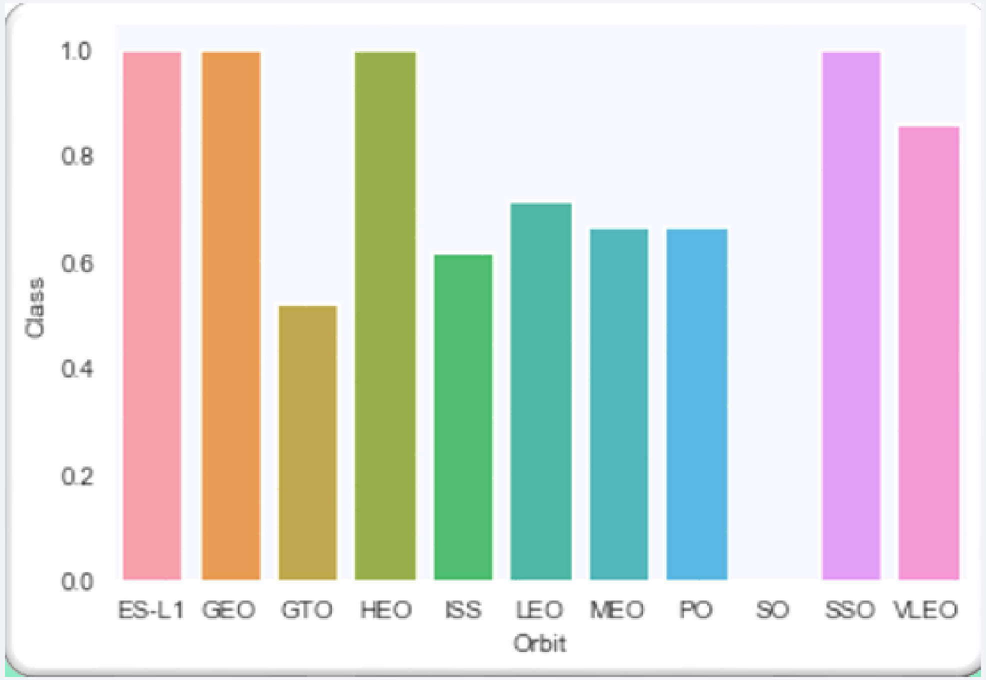


Success Rate vs. Orbit Type

The bar chart to the right shows that the following orbits have a very high (100%) historical success rate.

- ES-L1 (Earth-Sun First Langrangian Point)
- GEO (Geostationary Orbit)
- HEO (High Earth Orbit)
- SSO (Sun-synchronous Orbit)

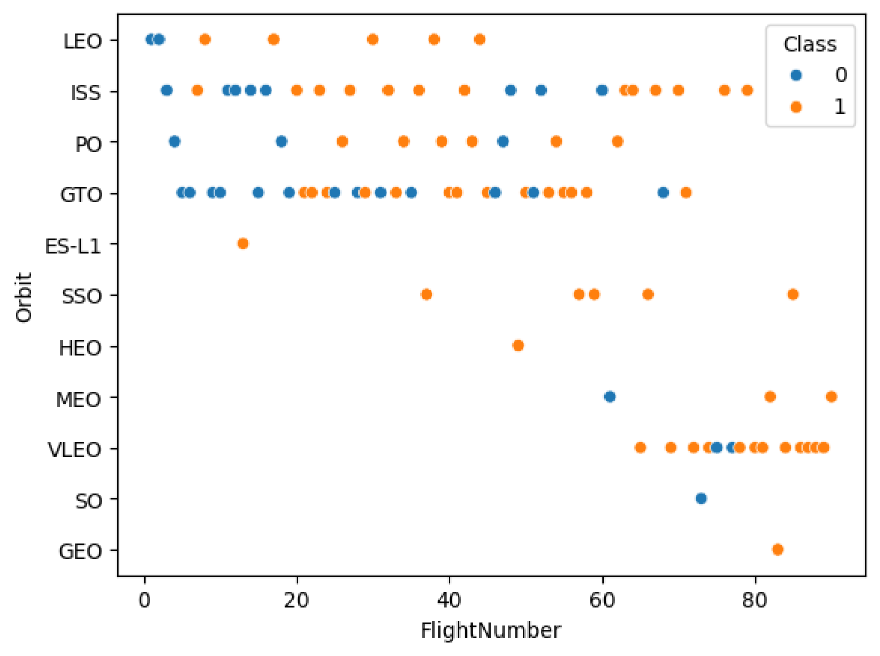
The orbit with the lowest (0%) success rate is SO (Heliocentric Orbit).



Flight Number vs. Orbit Type

This plot reveals data that the previous graphs did not expose, such that

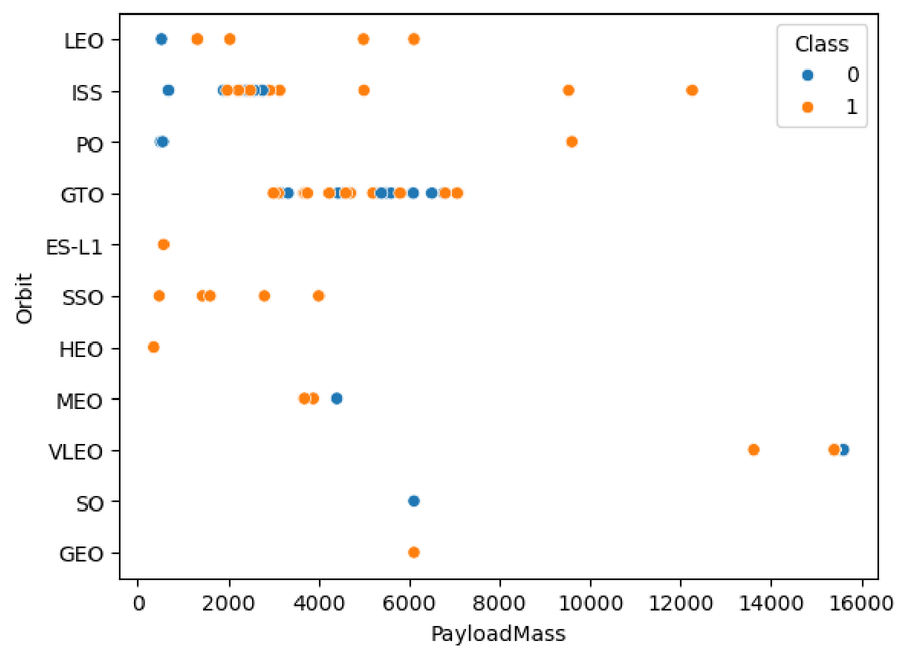
- The 100% success rate of 3 previously orbits (ES-L1, HEO, GEO) only have 1 flight each.
- SSO's 100% success rate is more significant, due to the 5 successful flights.
- There is very little relationship between Flight Number and Success Rate for GTO Orbit.



Payload vs. Orbit Type

The scatter plot of the Orbit Type vs. Flight Number shows that:

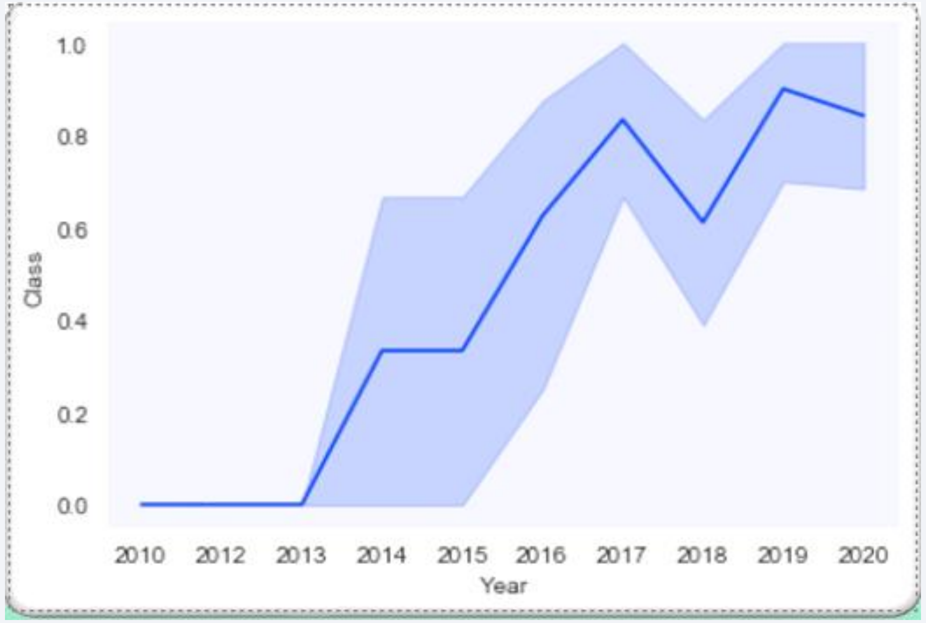
- "PO", "ISS", and "LEO" orbit types have a higher degree of success with heavier payloads.
- For the "GTO" orbit type, the relationship between payload and success rate is unclear.
- "VLEO" launches tend to have very heavy payloads, relative to the other orbit types.



Launch Success Yearly Trend

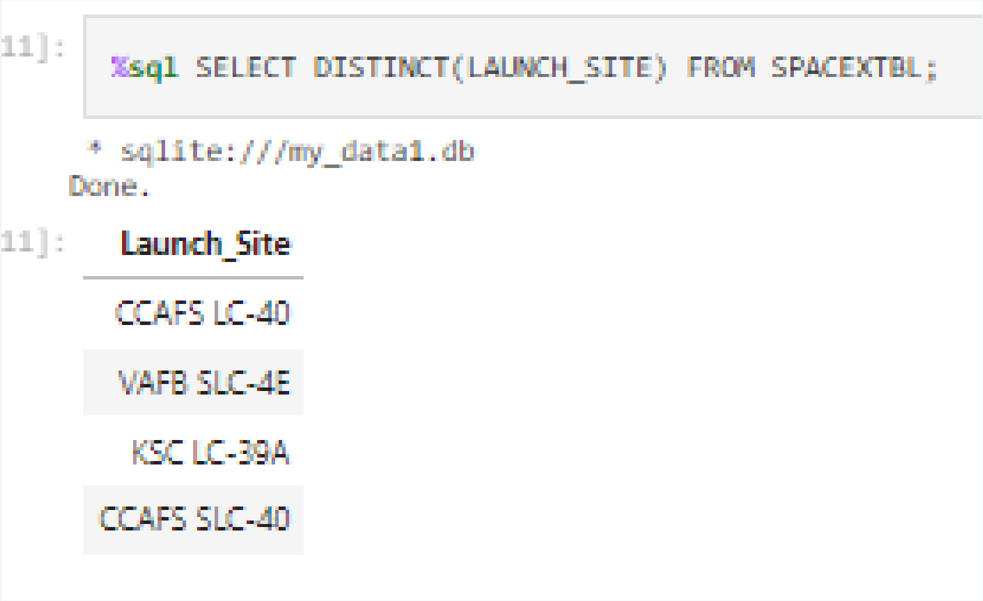
The line chart of yearly average success rate shows:

- Between 2010 and 2013, all landings were unsuccessful.
- After 2013, the success rate had an overall increase.
- In 2016, success rates succeeded 50%



All Launch Site Names

This short magic SQL query uses the 'DISTINCT' keyword to select only unique values.



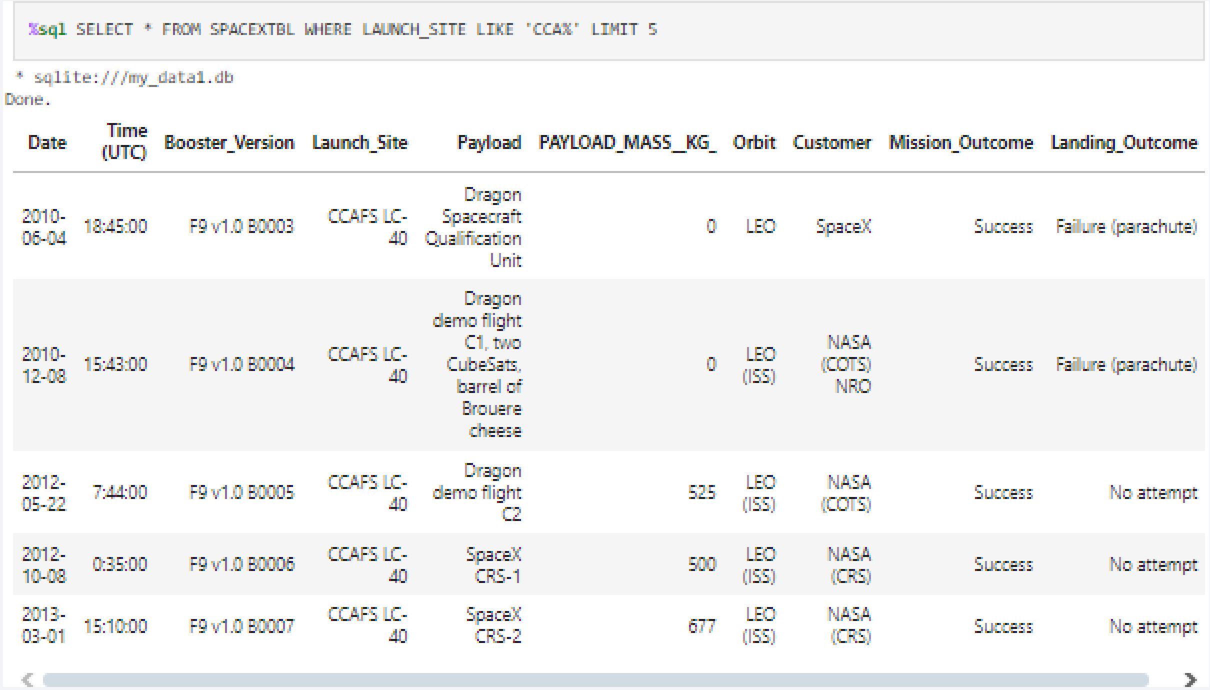
```
11]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;

 * sqlite:///my_data1.db
Done.
11]:
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- 'LIMIT 5' fetches only 5 records, according to the commands passed through the magic SQL query.



```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

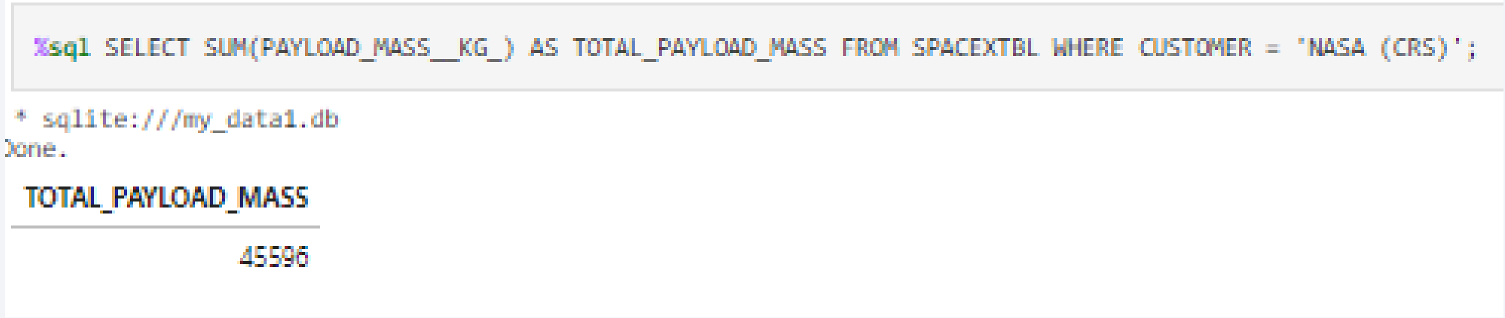
```
* sqlite:///my_data1.db
Done.
```

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

A magic SQL query was passed with the SUM() keyword, and parameters such as "WHERE CUSTOMER = 'NASA (CRS)'" to filter out boosters from non-NASA companies.



```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

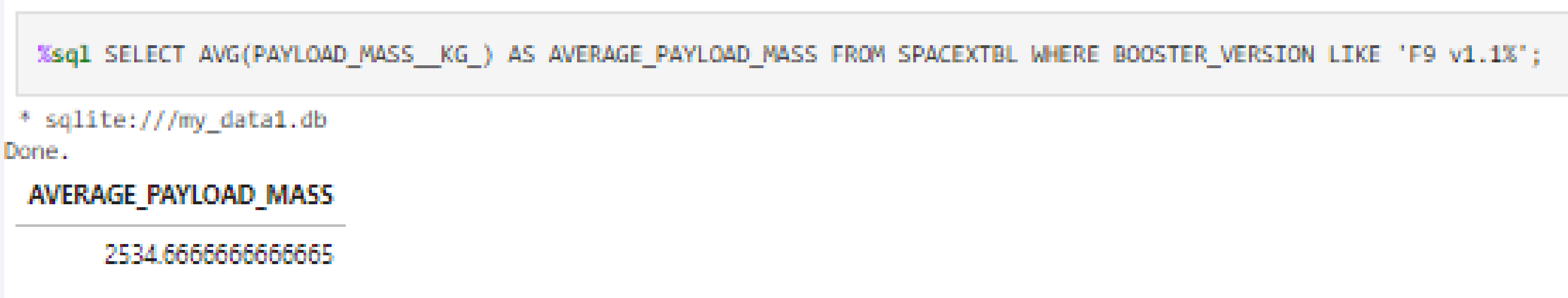
```
 * sqlite:///my_data1.db
Done.
```

TOTAL_PAYLOAD_MASS
45596

Average Payload Mass by F9 v1.1

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

"AVG" was used to calculate the average payload mass (in kilograms) of each payload flown using the specified booster "F9 v1.1"



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

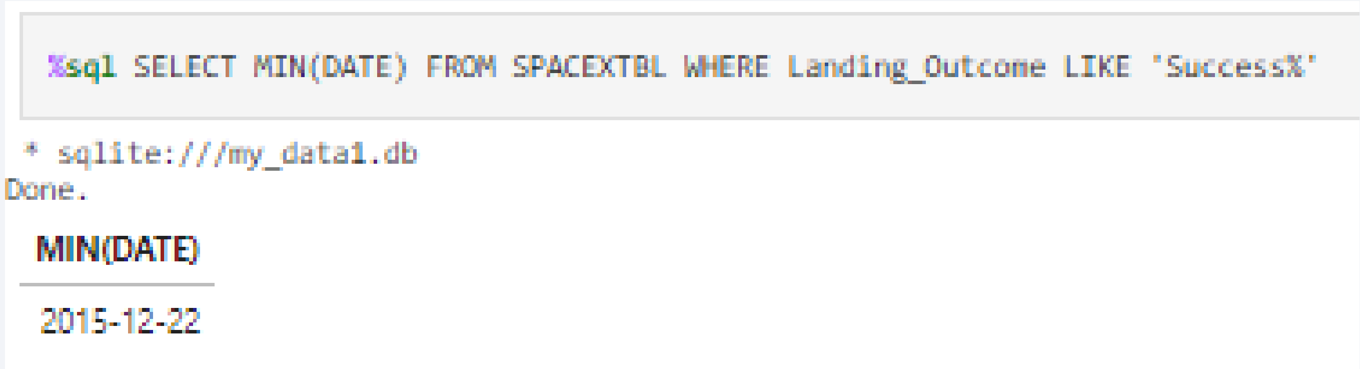
```
 * sqlite:///my_data1.db
Done.
```

AVERAGE_PAYLOAD_MASS
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

The "MIN" parameter was passed alongside filtering parameters to find the first date where the landing outcome was a success.



```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%'
```

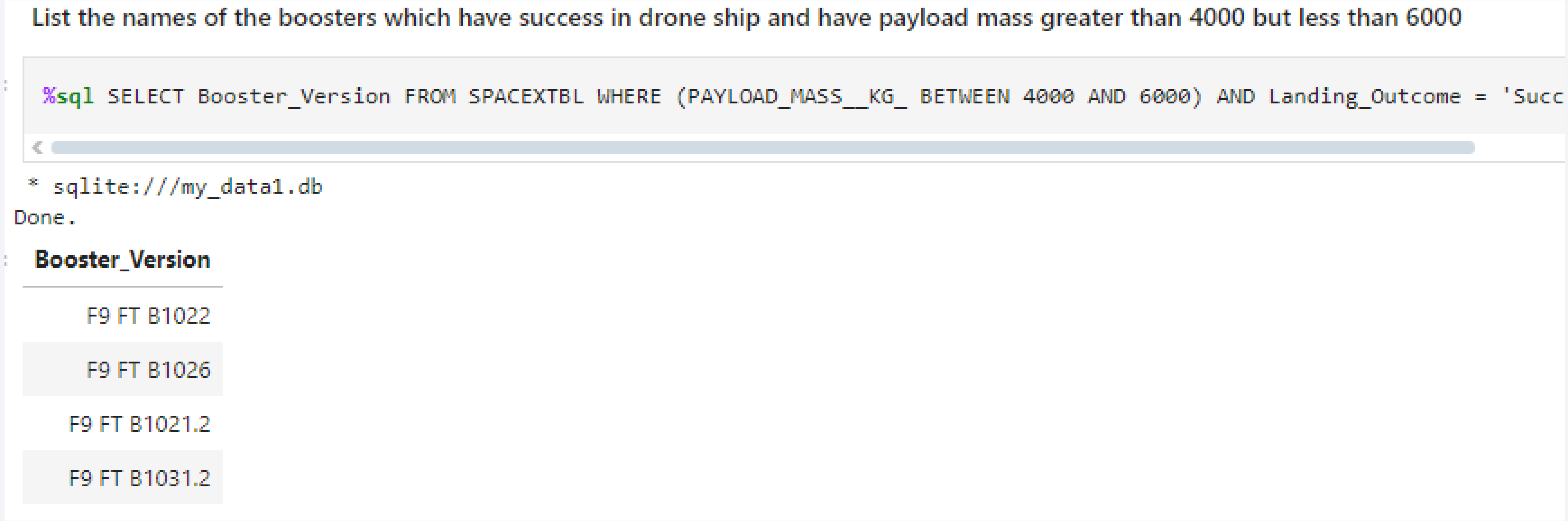
```
 * sqlite:///my_data1.db
Done.
```

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Using the "BETWEEN" keyword, the SQL query would only return 'Booster_Version' with 'Payload_Mass__KG_' BETWEEN 4000kg and 6000kg.



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 Booster_Version FROM SPACEXTBL WHERE (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000) AND Landing_Outcome = 'Succ
```

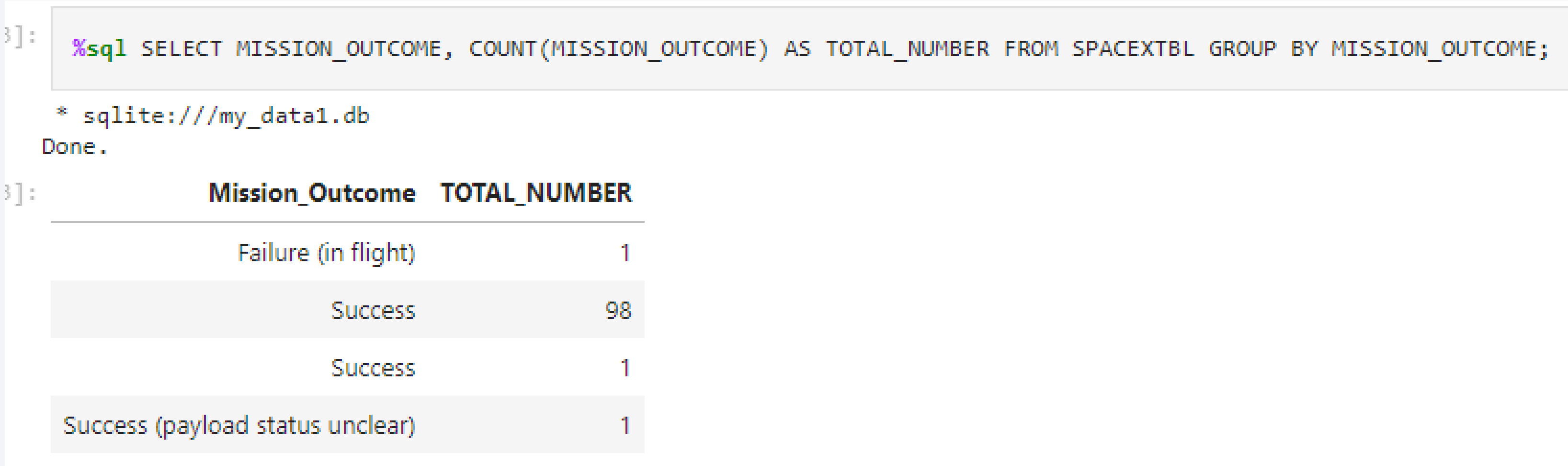
```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

This query was able to successfully show the 'Mission_Outcome' and the associated count number of each outcome.



```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

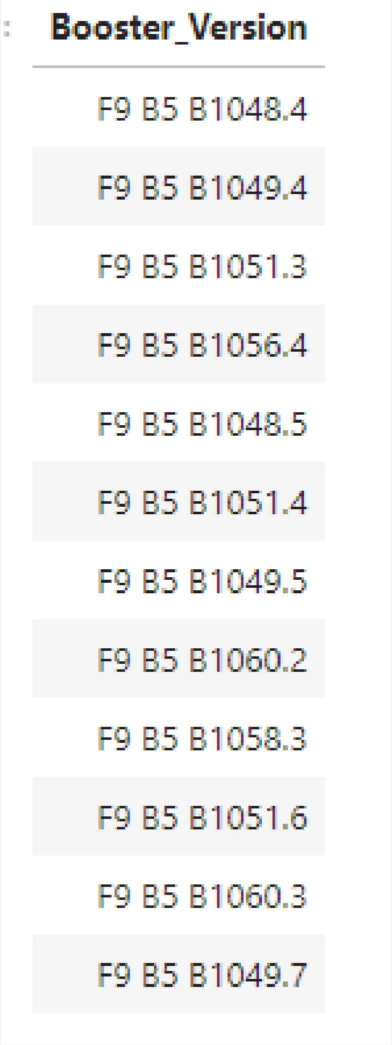
Mission_Outcome	TOTAL_NUMBER
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass

A subquery has been used here. The SELECT statement within the brackets finds the MAX payload mass, so we can see the MAX payload_mass associated with each booster.

Query results:

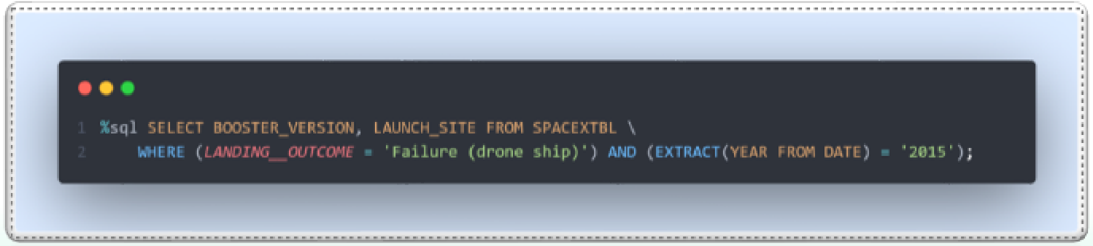


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

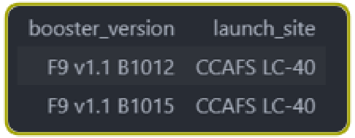
2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

The WHERE keyword and AND keyword allows for query results to be filtered, helping to find the booster version and launchsite for failed landing_outcomes in 2015.



```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL \
    WHERE (LANDING__OUTCOME = 'Failure (drone ship)') AND (EXTRACT(YEAR FROM DATE) = '2015');
```

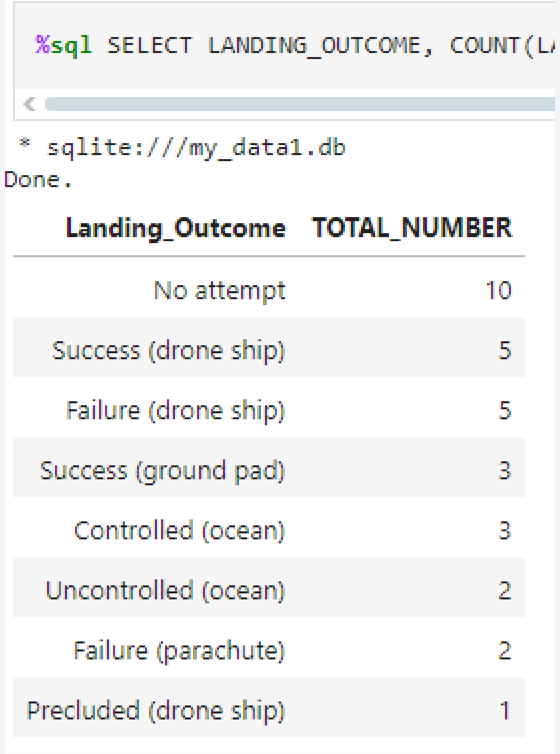


booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- 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

Using a magic SQL query, the landing_outcome and associated number of occurences for each outcome can be found and shown as follows:



```
%sql SELECT LANDING_OUTCOME, COUNT(L
```

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

Landing_Outcome	TOTAL_NUMBER
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

Section 5

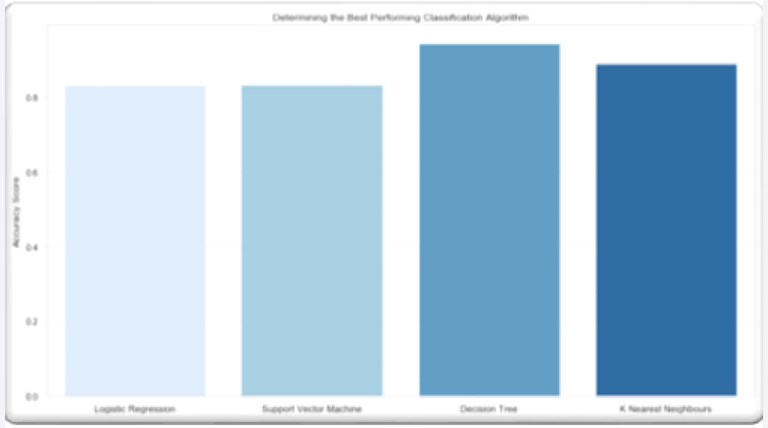
Predictive Analysis (Classification)

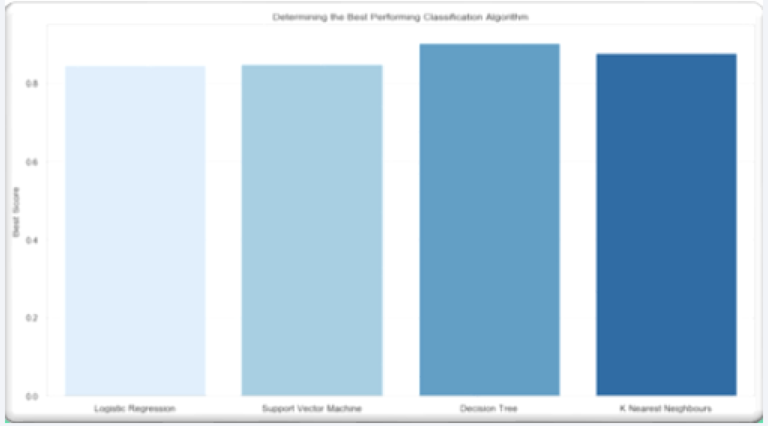
Classification Accuracy

Plotting the Accuracy Score (top right) and the Best Score (bottom right) for each classification algorithm produces the following bar charts:

From these charts, inferences can be made:

- The decision tree model has the highest classification accuracy, with an accuracy score of 94.44% and best score of 90.36%

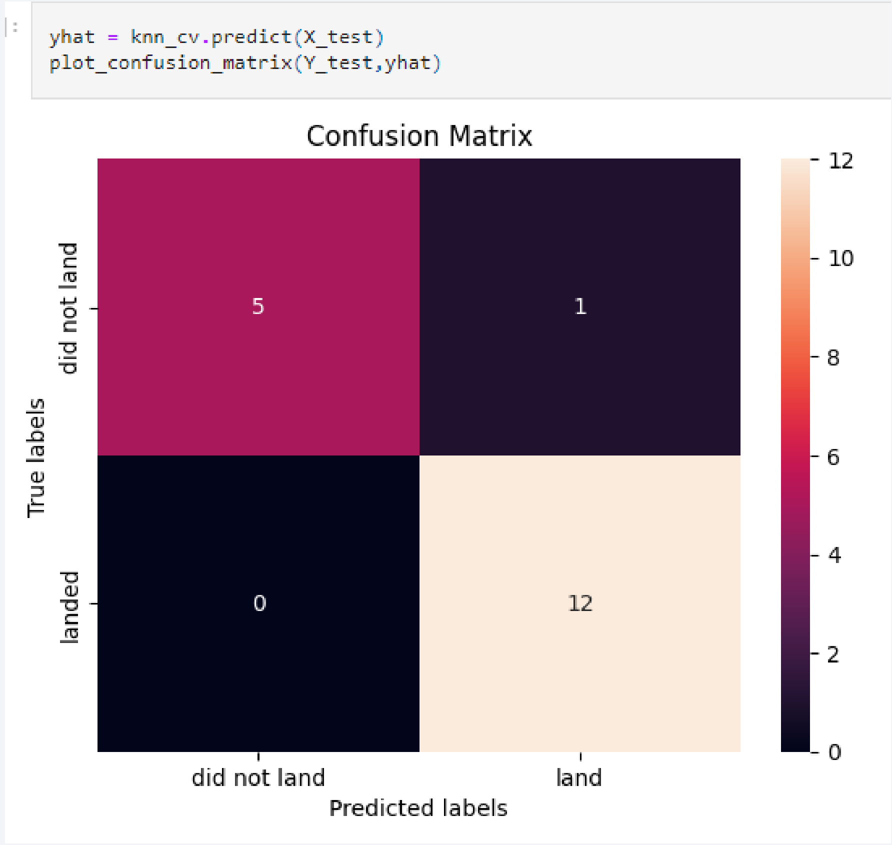




Confusion Matrix

This confusion matrix shows the accuracy of the Decision Tree model. Only 1 of 18 results were classified incorrectly (a false positive) as shown in the top right corner of the matrix.

The other 17 results were correctly classified as true positives or true negatives.



```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Conclusions

- As the number of flights per launch site increases, the rate of success at said launch site increases.
  - Between 2010 and 2013, the success rate was 0%
  - After 2013, success rates began to grow steadily
  - After 2016, success rates stayed above 50%
- The launch site "KSC LC-39 A" had the most successful launches, with 41.7% of the total successful launches from the total dataset, and the highest success rate of all launch sites, with a 76.9% success rate.
- The best performing MLM (Machine Learning Model) is the Decision Tree model, with an accuracy of 94.44%

Thank you!