IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rachata Wichakkhapan
20/04/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- ## Project background and context

The commercial space age has arrived, and companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX are leading the way in the industry. SpaceX's main advantage over its competitors is the cost of its rocket launches, which is due in part to its ability to reuse the first stage of its rockets. However, determining if the first stage will land successfully is a crucial factor in reducing the cost of a launch.

- ## Common problem that needed solving

As a data scientist working for a new rocket company, it would be essential to gather information about SpaceX and create dashboards for your team to determine the price of each launch and if SpaceX will reuse the first stage. Additionally, training a machine learning model to predict if SpaceX will reuse the first stage and using data to predict whether SpaceX will attempt to land a rocket or not could be valuable in competing with SpaceX.

Section 1

Methodology

Methodology

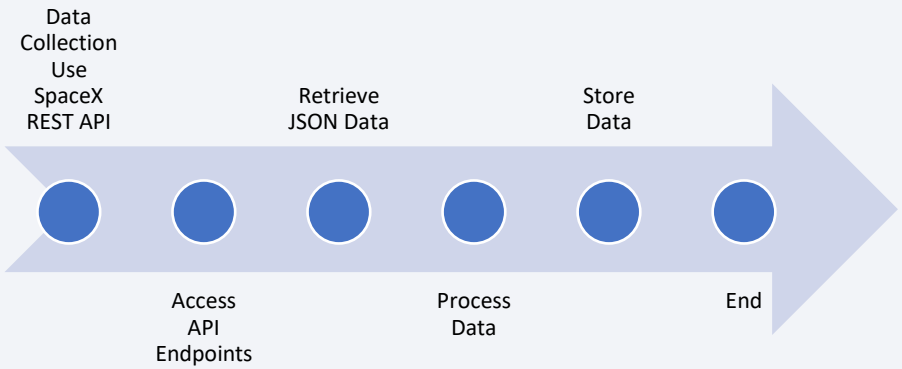
Executive Summary

- Data collection methodology
  - Space X Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One-Hot Encoding to convert categorical data into numerical data that machine learning models can process.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting to show relationships between variables
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

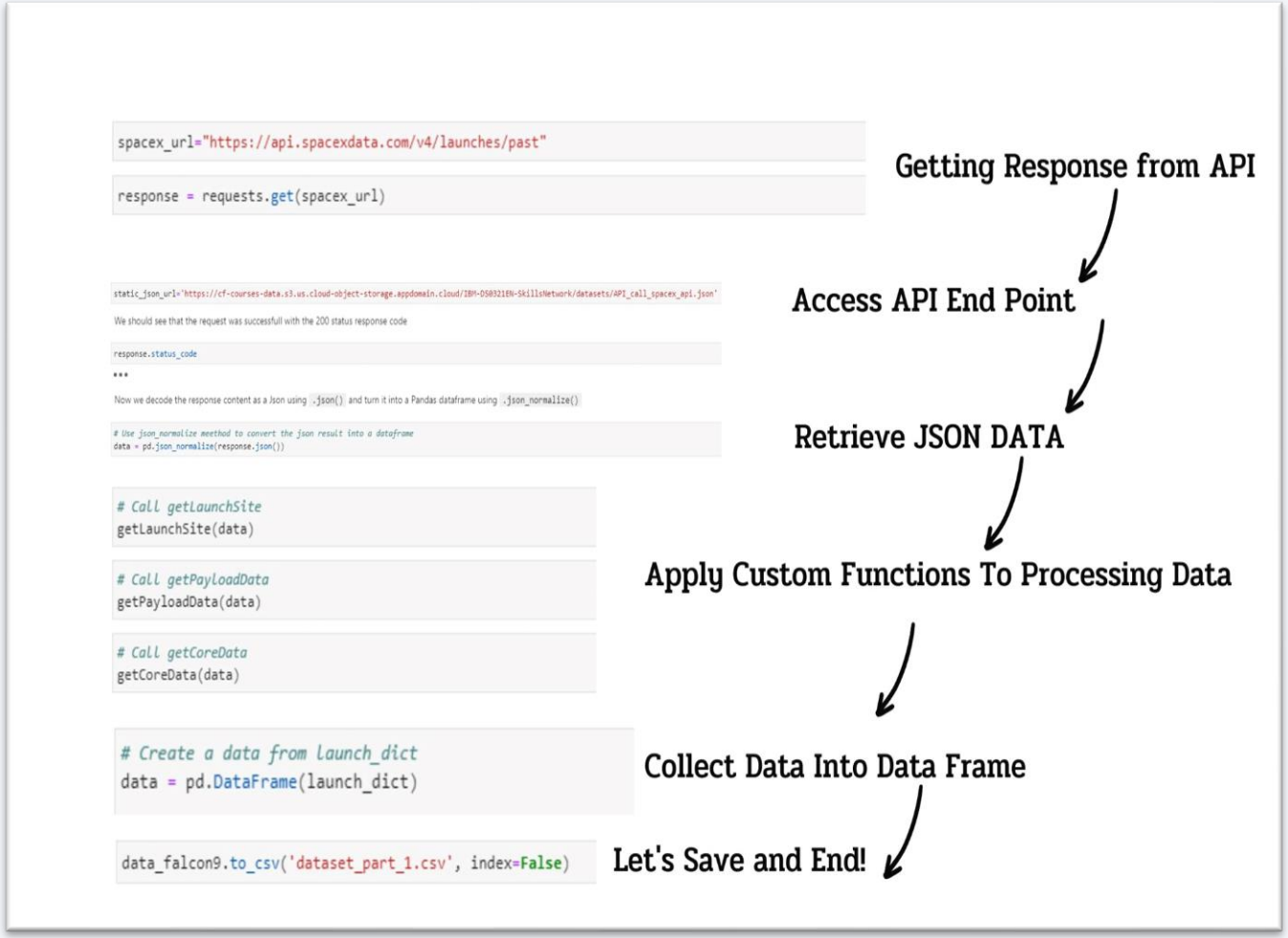
Data Collection

- The data sets was collected by:
  - SpaceX launch data that gathered from the SpaceX REST API.
  - We will get data about information of launch such as the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome
  - Another data collected by using the Python BeautifulSoup to web scrape some HTML table from Wikipedia.

Data Collection – SpaceX API



Data Collection-SpaceX API full process: [Github Click]



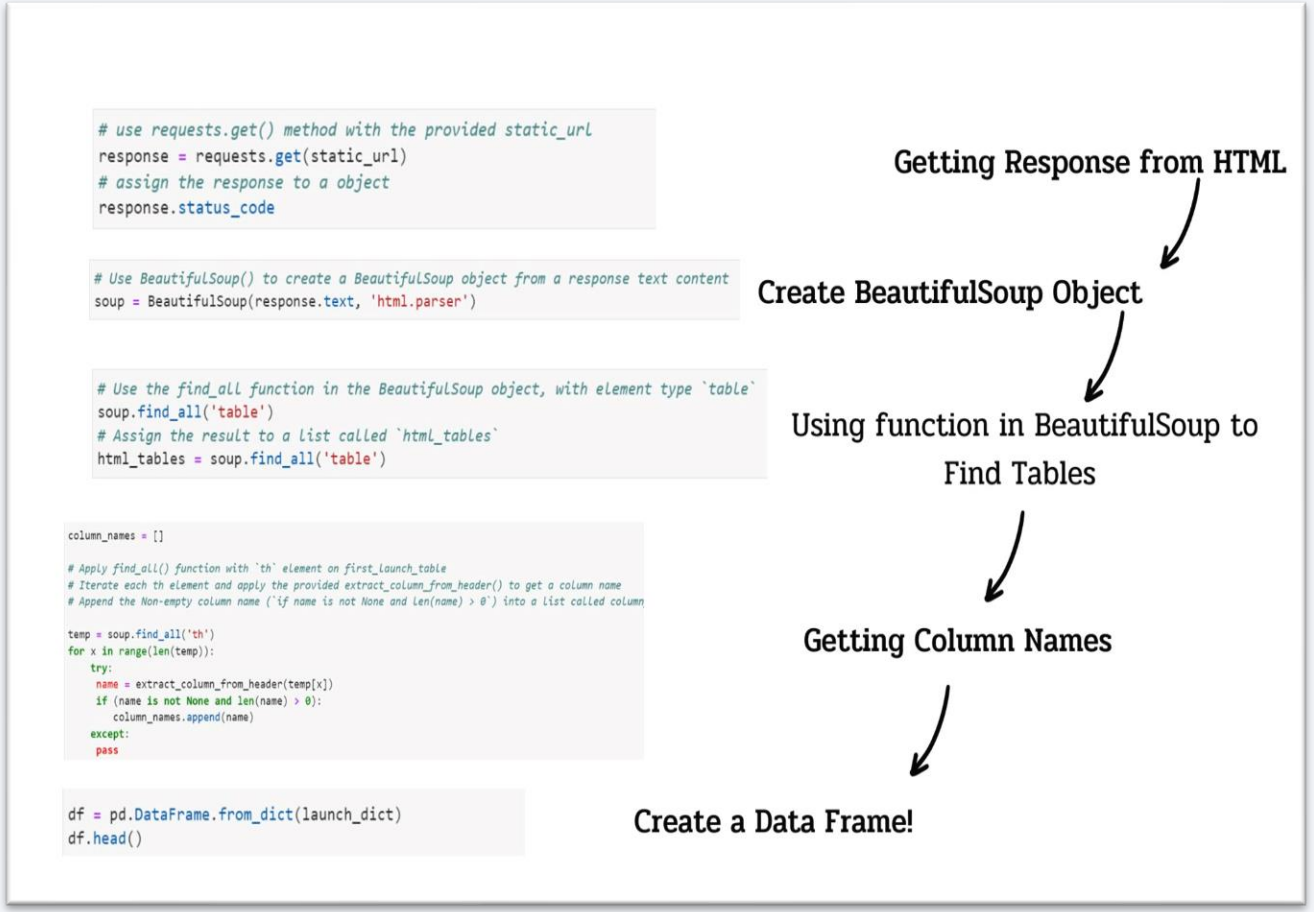
Data Collection - Scraping

Getting Response from Wikipedia

Extract All Data from The HTML Table Header

Create a Data Frame by Parsing The HTML Tables

Data Collection-Web Scrapping full process: [Github Click]

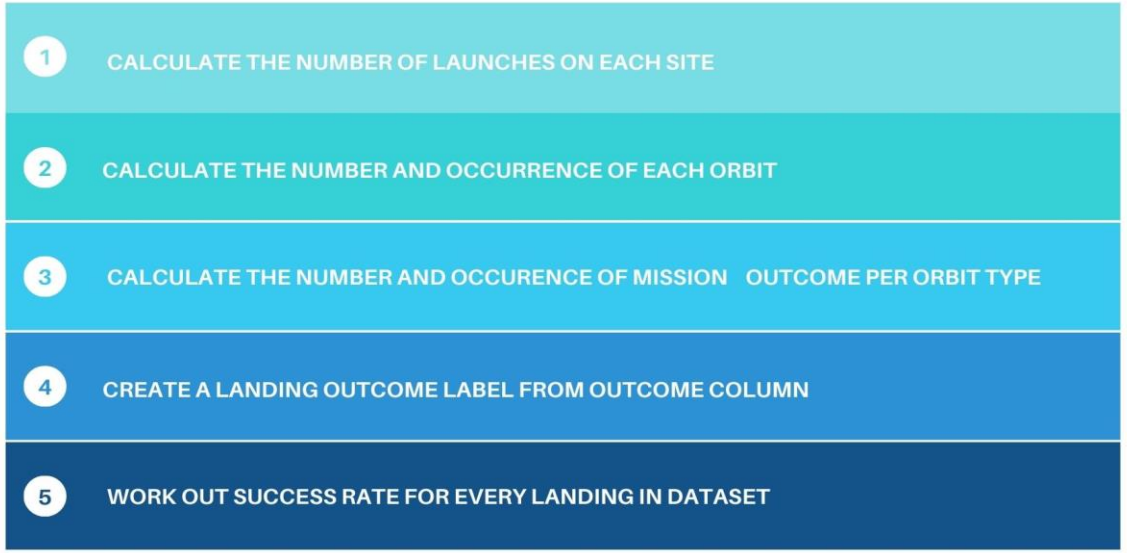


Data Wrangling

Introduction

The dataset includes cases where the booster failed to land successfully. The landing attempts were either successful or unsuccessful due to accidents. True/False Ocean indicates landing outcome in the ocean, and True/False RTLS indicates landing outcome on the ground pad. True/False ASDS refers to landing outcome on a drone ship.

To simplify the dataset, we'll convert these outcomes into training labels. 1 means landed successfully, and 0 means unsuccessful.



Data Wrangling-full process: Github Click

EDA with Data Visualization

- Scatter Graphs being drawn:
  - Flight Number VS. Payload Mass
  - Flight Number VS. Launch Site
  - Payload VS. Launch Site
  - Orbit VS. Flight Number
  - Payload VS. Orbit Type
  - Orbit VS. Payload Mass

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation. Scatter plots usually consist of a large body of data.

- Bar Graph being drawn:
  - Mean Vs. Orbit

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other.

- Line Graph being drawn:
  - Success Rate VS, Year

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

EDA with Data Visualization: GitHub Click

EDA with SQL

Performed SQL queries to gather information about the dataset

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the data when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass.
- List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub Click

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map, we took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.

We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()

Using Haversine's formula we calculated the distance from the Launch Site to measure patterns Lines are drawn on the map to measure distance to landmarks.

Interactive Map with Folium: GitHub Click

After I plot distance lines to the proximities, I can answer the following questions easily:

1. Are launch sites in close proximity to railways?
   - ❖ No, launch sites far from Florida East Coast Railway about 22.01 KM.
2. Are launch sites in close proximity to highways?
   - ❖ Yes, launch sites in close proximity to Samuel C Phillips Parkway about 1.17 KM.
3. Are launch sites in close proximity to coastline?
   - ❖ Yes, launch Sites in close proximity to coastline about 0.86 KM
4. Do Launch sites keep certain distance away from cities?
   - ❖ Yes, Launch Sites keep certain distance away from cities such as keep distance away from Melbourne about 51.75 KM

Build a Dashboard with Plotly Dash

Building a dashboard application with the Python Plotly Dash package. This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.

I can use it to find more insights from the SpaceX dataset more easily than with static graphs.

- Graphs
  - Pie Chart showing the total launches by a certain site/all sites
  - Scatter Graph showing the relationship with Outcome and Payload Mass (KG) for the different Booster Versions.

Dashboard with Plotly Dash: GitHub Click

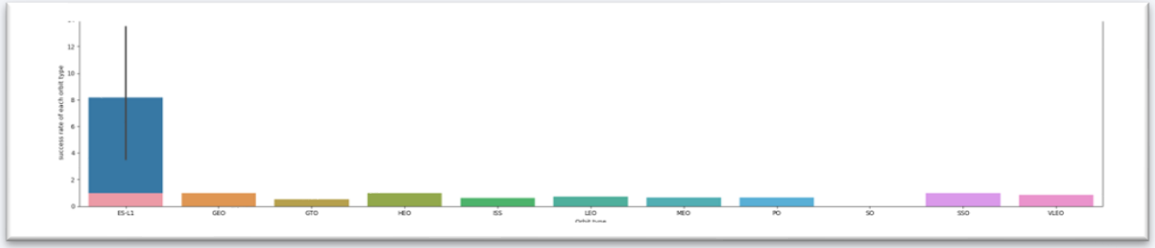
Predictive Analysis (Classification)

Built →	Evaluated →	Improved →	Best Model →
• Create a numpy array • Standardize the data • Split the data X and Y into training and test data • Decide each models we want to use • Set and Fit datasets into the GridSearchCV	• Calculate accuracy for each model • Tuned hyperparameters to find the best parameters • Plot Confusion Matrix	• Feature Engineering • Algorithm Tuning	• Finding the best model • Finding the best accuracy • Finding the best parameters

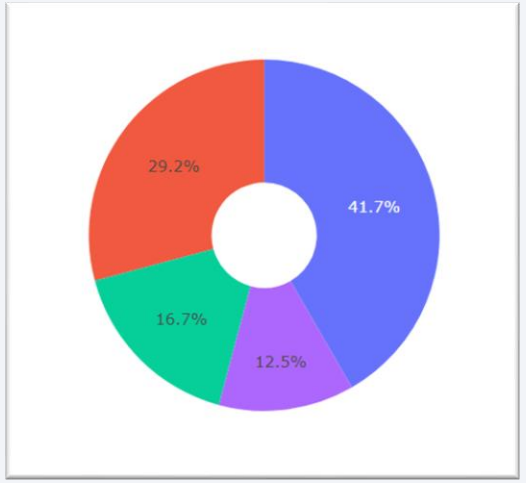
Predictive Analysis(Classification): GitHub Click

Results

Exploratory data analysis results

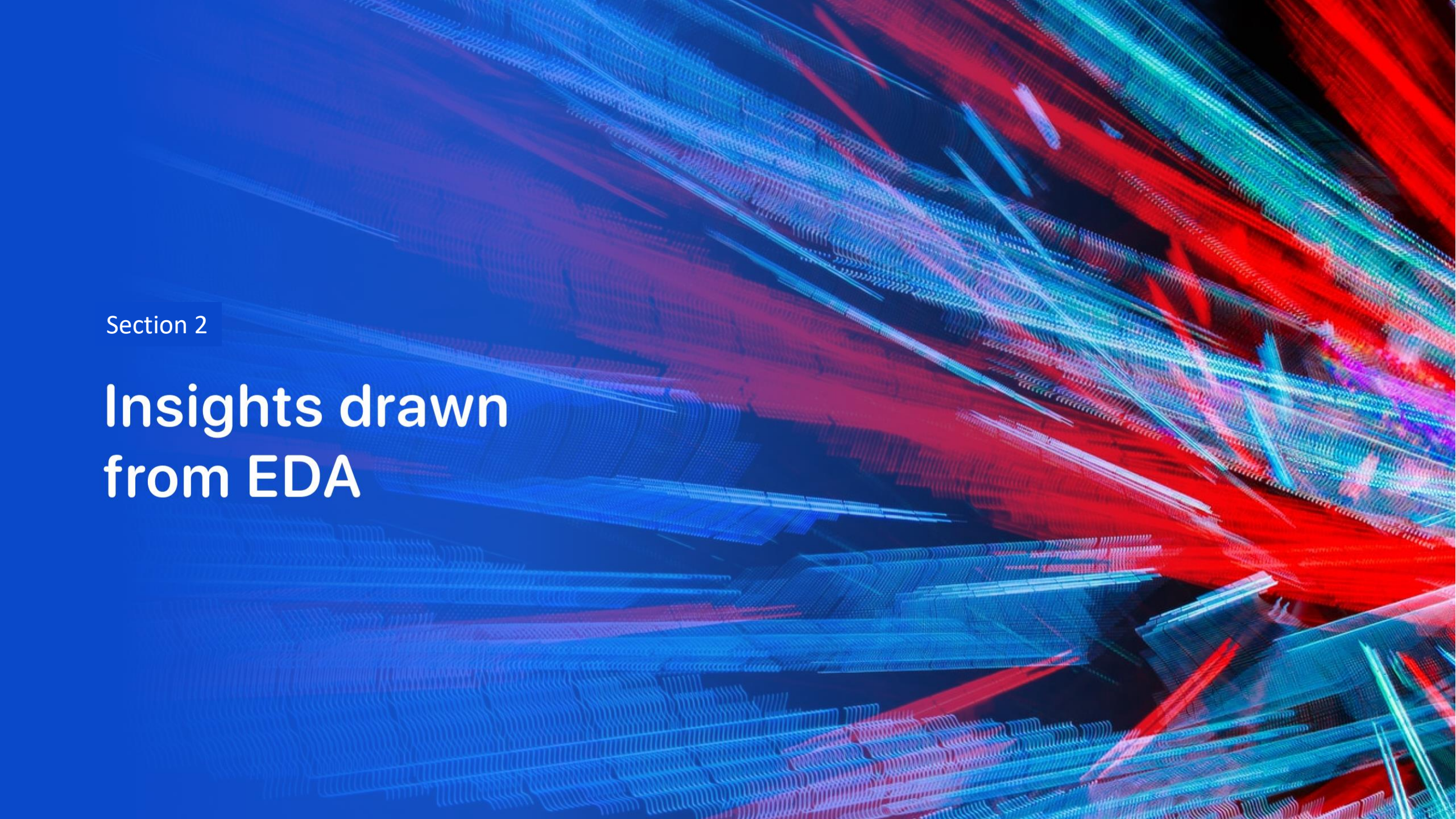


Interactive analytics demo in screenshots

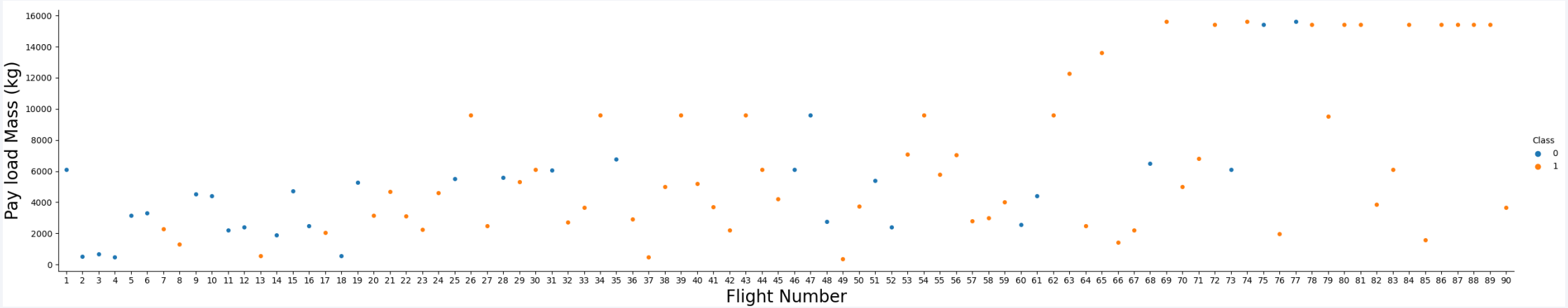


Predictive analysis results

Best Model: Decision Tree Classifier
Best Accuracy: 0.8732142857142857
Best Parameter: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'best'}

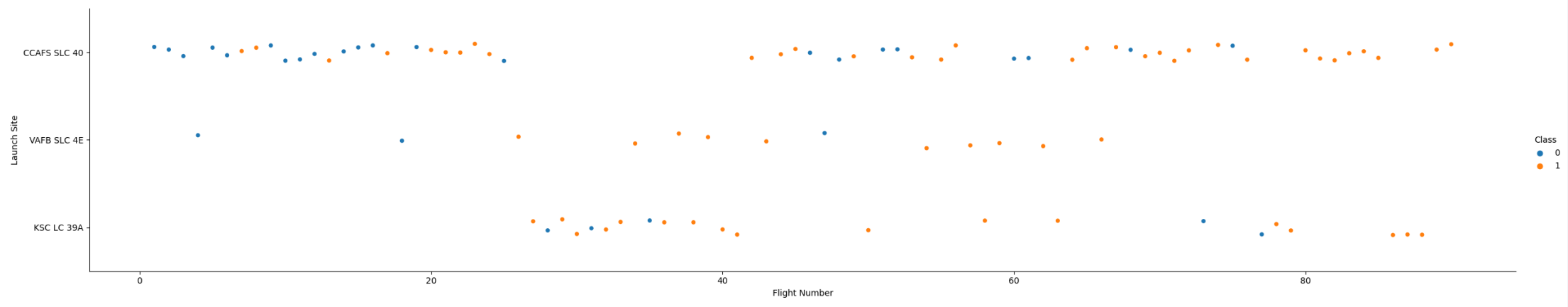

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



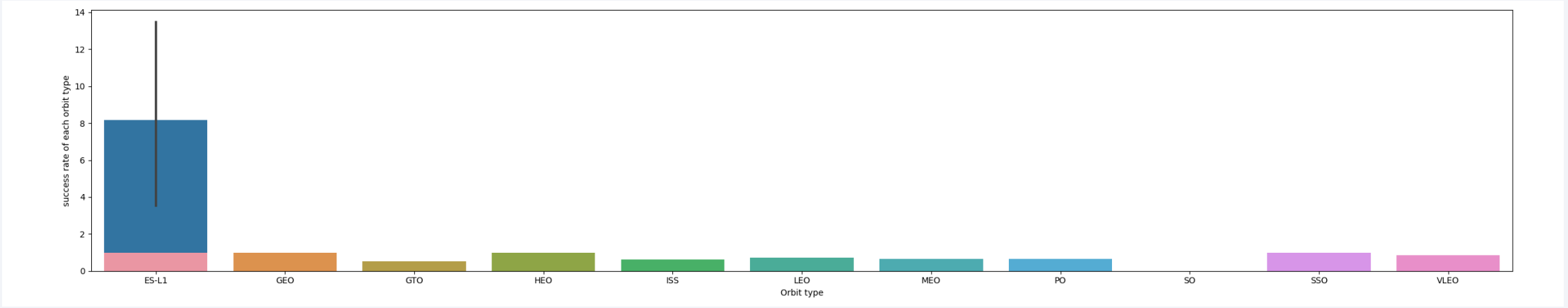
- We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Payload vs. Launch Site



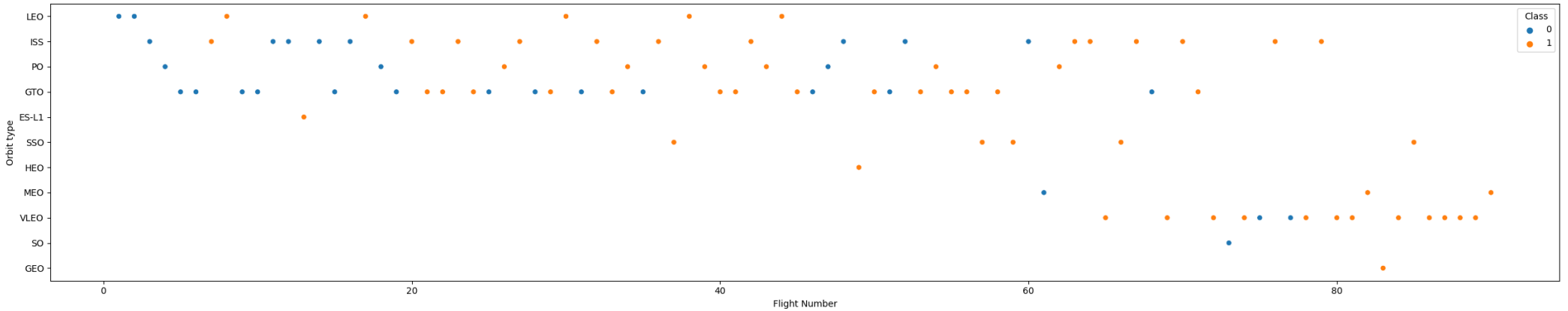
- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

Success Rate vs. Orbit Type



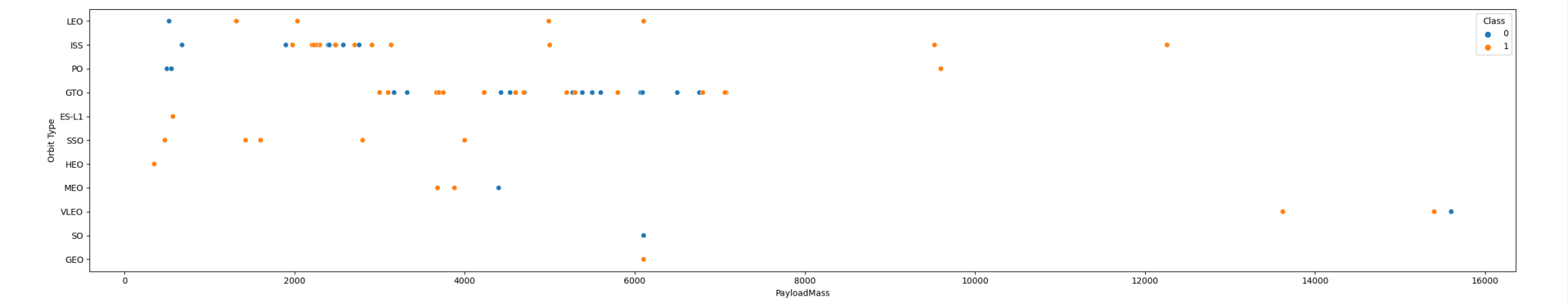
- We can see that the best success rate is ES-L1 orbit.

Flight Number vs. Orbit Type



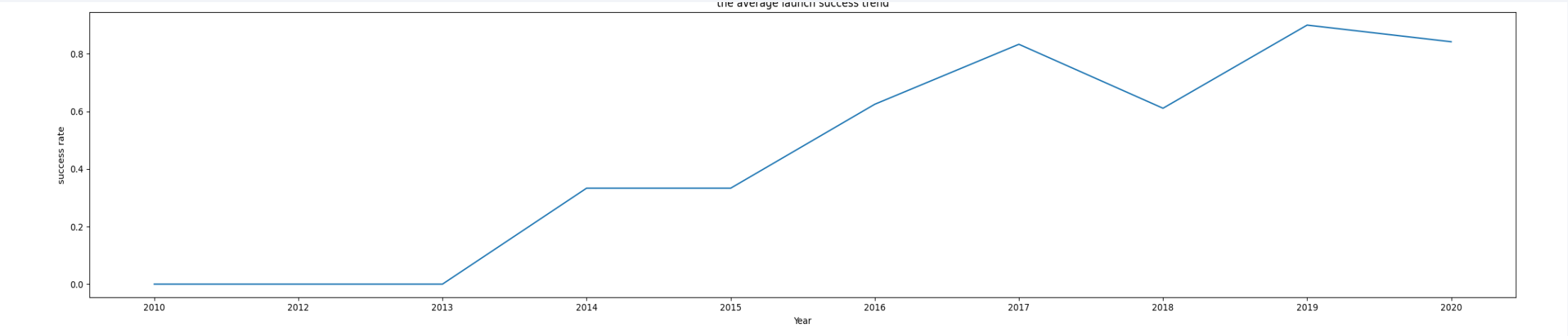
- We can see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

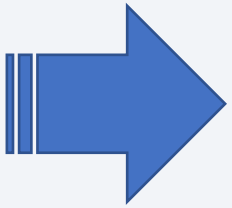


- you can observe that the success rate since 2013 kept increasing till 2020

All Launch Site Names

- **SQL Query**

%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;



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

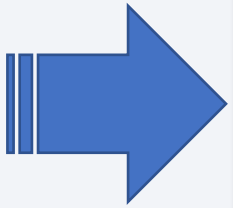
- **Query Explanation**

Using “DISTINCT” in the query to show unique values in the Launch_Site from Spacextbl column.

Launch Site Names Begin with 'CCA'

- **SQL Query**

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



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

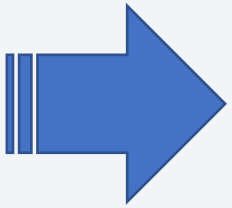
- **Query Explanation**

To display all the columns of the "Spacextbl“ from the "Launch_Site" table, use the "SELECT" statement. To display specific data that have the letters "CCA" in them, use the "LIKE" operator.

Total Payload Mass

- **SQL Query**

```
%sql SELECT sum(PAYLOAD_MASS__KG_) as
Total_Payload_Mass FROM SPACEXTBL WHERE
Customer = 'NASA (CRS)' ;
```



Total_Payload_Mass
45596

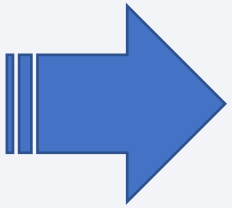
- **Query Explanation**

To retrieve the total Payload Mass from the Spacextbl table, use the SELECT and SUM statements. To display only the data where the Customer column contains the letters 'NASA(CRS)', use the WHERE statement.

Average Payload Mass by F9 v1.1

- **SQL Query**

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as
Average_Payload_Mass from SPACEXTBL WHERE
Booster_Version = 'F9 v1.1';
```



AVG(PAYLOAD_MASS__KG_)
2928.4

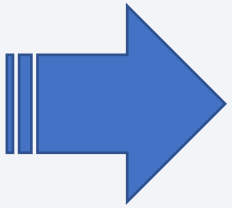
- **Query Explanation**

To retrieve the Average Payload Mass by F9 v1.1 from the Spacextbl table, use the SELECT and AVG statements. To display only the data where the Booster_version column contains the letters ‘F9 v1.1', use the WHERE statement.

First Successful Ground Landing Date

- **SQL Query**

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE
"Landing _Outcome" = 'Success (ground pad)';
```



MIN(Date)
01-05-2017

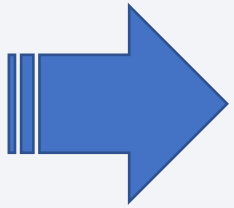
- **Query Explanation**

To retrieve the first successful ground landing date from the Spacextbl table, use the SELECT and MIN statements. To display only the first date where the Landing_Outcome column contains the “Success(ground pad)”, use the WHERE statement.

Successful Drone Ship Landing with Payload between 4000 and 6000

- **SQL Query**

```
%sql SELECT Booster_Version FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (drone ship)'
AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```



Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

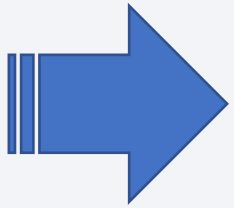
- **Query Explanation**

To select the Booster_Version of all the rows in the SPACEXTBL table where the Landing_Outcome is 'Success (drone ship)' and the PAYLOAD_MASS__KG_ is between 4000 and 6000 kilograms.

Total Number of Successful and Failure Mission Outcomes

- **SQL Query**

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



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

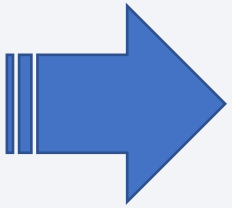
- **Query Explanation**

This SQL query counts the occurrences of each unique value in the "Mission_Outcome" column of the "SPACEXTBL" table, and groups the results by the unique values of the "Mission_Outcome" column. The resulting output shows each unique value and the count of occurrences of that value in the table.

Boosters Carried Maximum Payload

- **SQL Query**

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



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

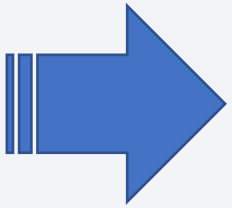
- **Query Explanation**

This SQL query retrieves the value of the Booster_Version column from the SPACEXTBL table for the row(s) that have the highest value of the PAYLOAD_MASS__KG_ column.

2015 Launch Records

- **SQL Query**

```
%%sql SELECT SUBSTR(Date,4, 2) AS Month, "Landing
_Outcome", Booster_Version, Launch_Site FROM SPACEXTBL
    WHERE "Landing _Outcome" = 'Failure (drone ship)' AND
SUBSTR(Date,7,4)='2015' ORDER BY Month;
```



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

- **Query Explanation**
  - The first line of the query uses the SUBSTR function to extract the month from the Date column and rename it as "Month". It also selects the columns "Landing_Outcome", "Booster_Version", and "Launch_Site".
  - The second line uses the WHERE clause to filter the results to only include rows where the "Landing_Outcome" column is equal to 'Failure (drone ship)' and the year in the Date column is 2015.
  - The third line uses the ORDER BY clause to sort the results by month.

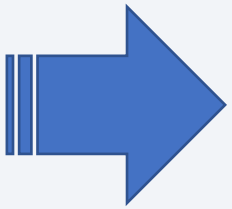
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- **SQL Query**

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

- **Query Explanation**

This SQL query retrieves information from the SPACEXTBL table regarding the landing outcomes of SpaceX missions between the dates of April 6th, 2010 and March 20th, 2017. The query groups the data by the Landing_Outcome column and calculates the count of each group.



Landing _Outcome	Count
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

All locations of SpaceX launch sites

This code is used to create an interactive map that displays the locations of various launch sites and provides additional information about each site when clicked. This type of visualization can be useful for understanding the geographic distribution of launch sites and their relative distances from one another.

- **Result**

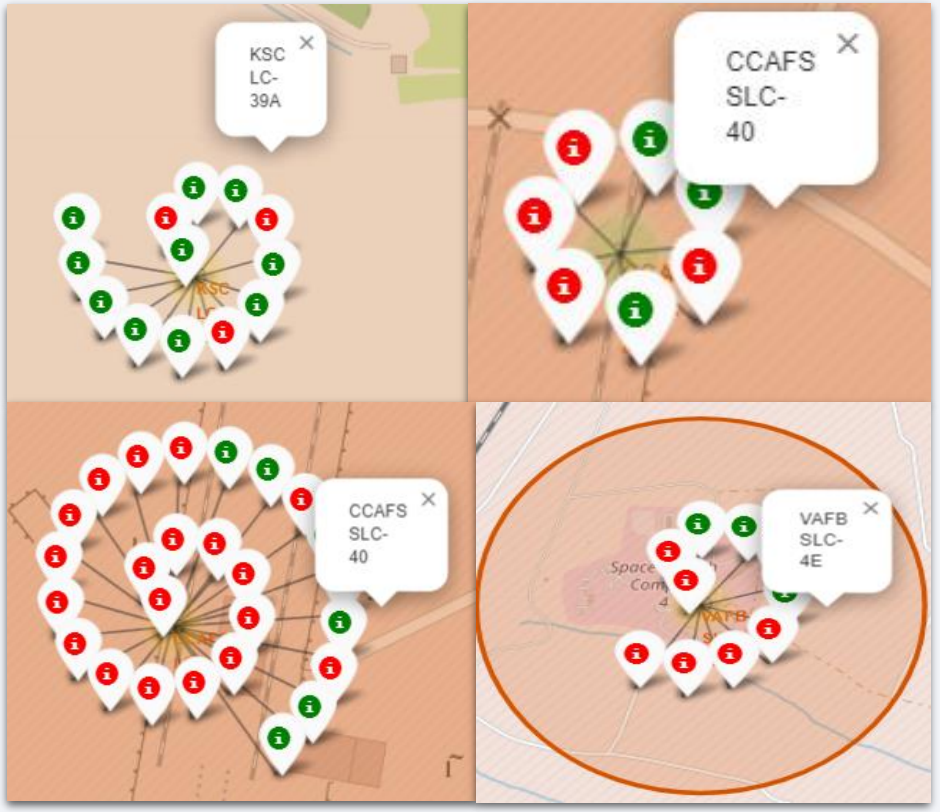
SpaceX launch sites are located in two US states: Florida and California.

Specific launches and their success or failure rates

To creates a map object with a specified location and zoom level, adds a Circle and Marker object for each launch site, and displays the resulting map. To add a Marker Cluster object to the map, and for each row in a spacex_df data frame, creates a Marker object with its coordinate and customizes the Marker's icon property to indicate if this launch was successful or failed.

- **Result**

Green Marker shows successful Launches and Red Marker shows Failures

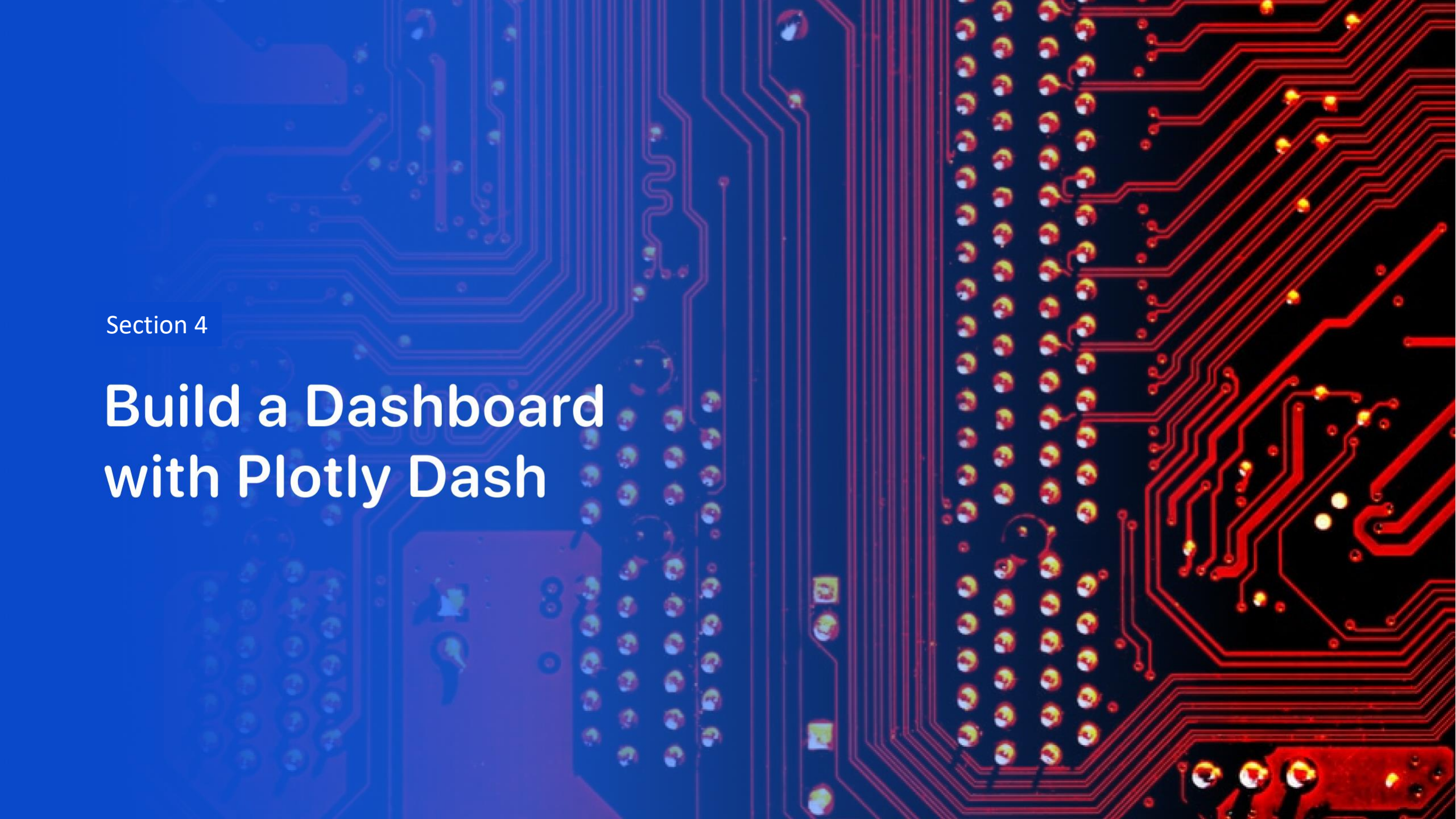


Using MousePosition and calculate the distance between each landmarks and the CCAFS-SLC-40 launch site.

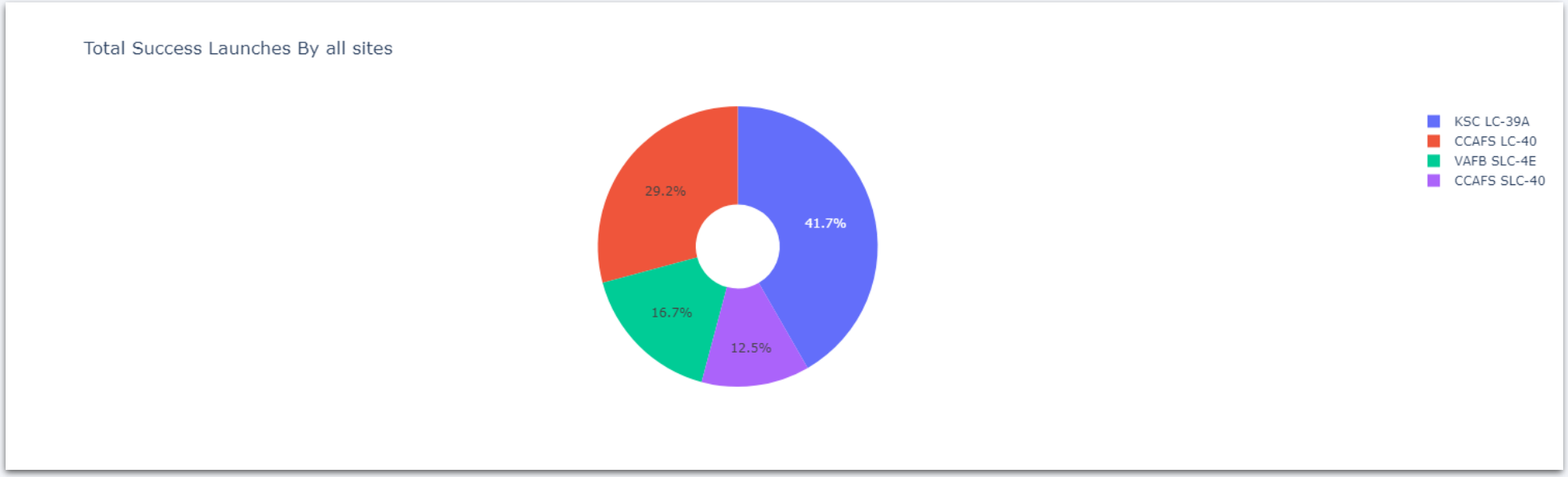
You can see a line between a launch site to its closest city, railway, highway, etc.

- **Result**

1. Are launch sites in close proximity to railways?
   - ❖ No, launch sites far from Florida East Coast Railway about 22.01 KM.
2. Are launch sites in close proximity to highways?
   - ❖ Yes, launch sites in close proximity to Samuel C Phillips Parkway about 1.17 KM.
3. Are launch sites in close proximity to coastline?
   - ❖ Yes, launch Sites in close proximity to coastline about 0.86 KM
4. Do Launch sites keep certain distance away from cities?
   - ❖ Yes, Launch Sites keep certain distance away from cities such as keep distance away from Melbourne about 51.75 KM


Section 4
Build a Dashboard with Plotly Dash

Total Success Launches By all sites

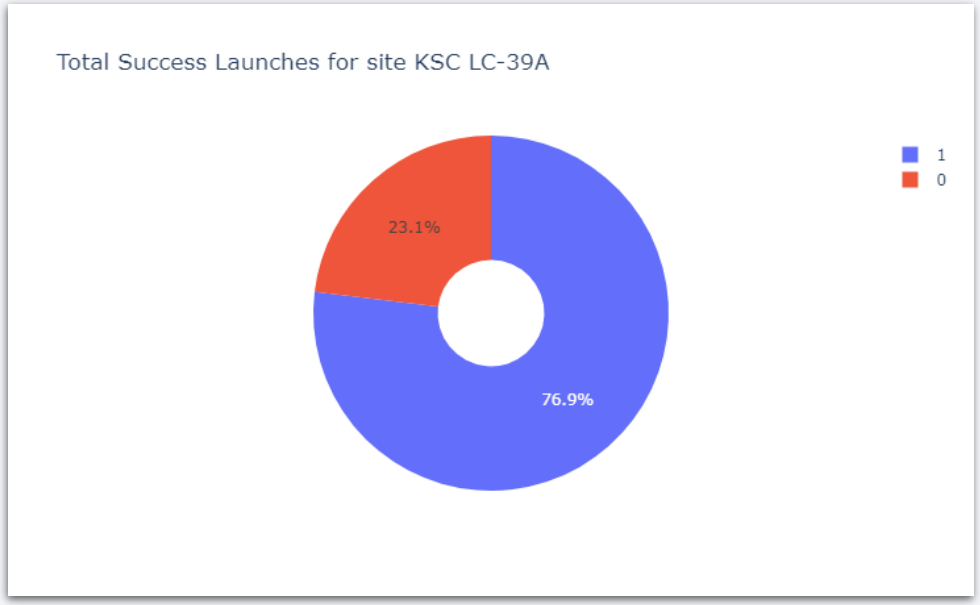


We can see from the pie chart that the KSC LC-39A launch site has had the most successful landings and CCAFS SLC-40 has had the least successful landings.

Compare the ratio of the most successful launches to the most failed launches.

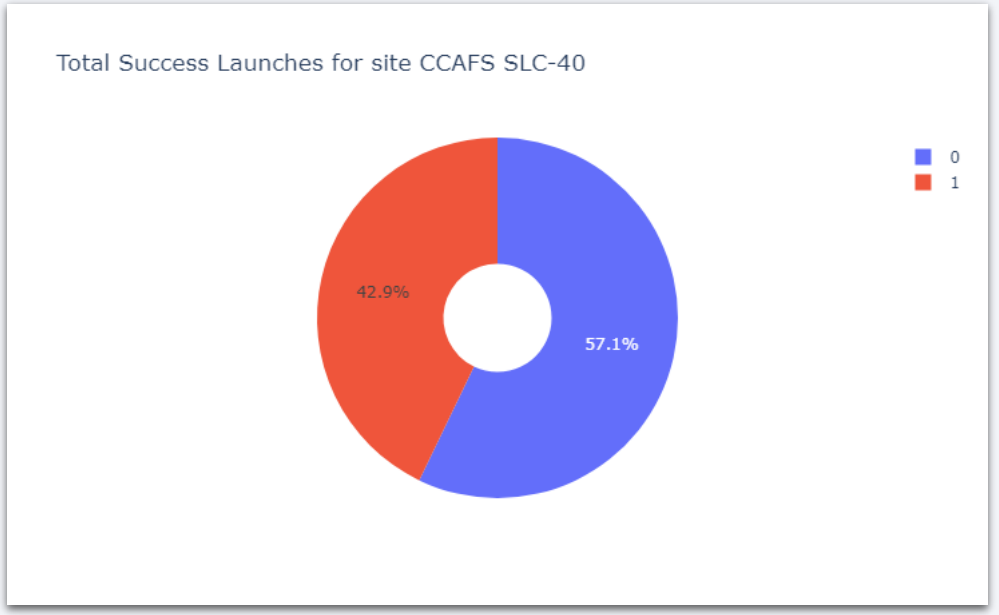
The most successful launches (KSC LC-39A)

- 76.90% for success rates and 23.10% for failure rates



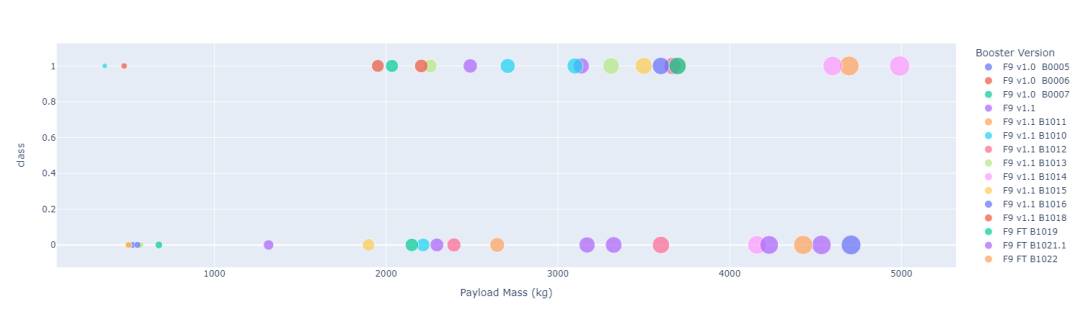
The most failed launches (CCAFS SLC-40)

- 57.10% for success rates and 42.90% for failure rates

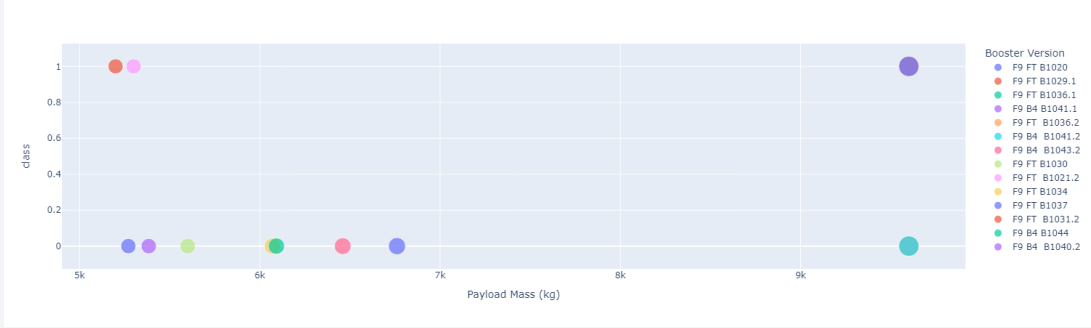


Compare success rates between low-weighted payloads and high-weighted payloads.

Low weighted payloads (0KG-5,000KG)



High weighted payloads (5,000KG-10,000KG)



You can see the success rates for low weighted payloads is higher than the heavy weighted payloads.

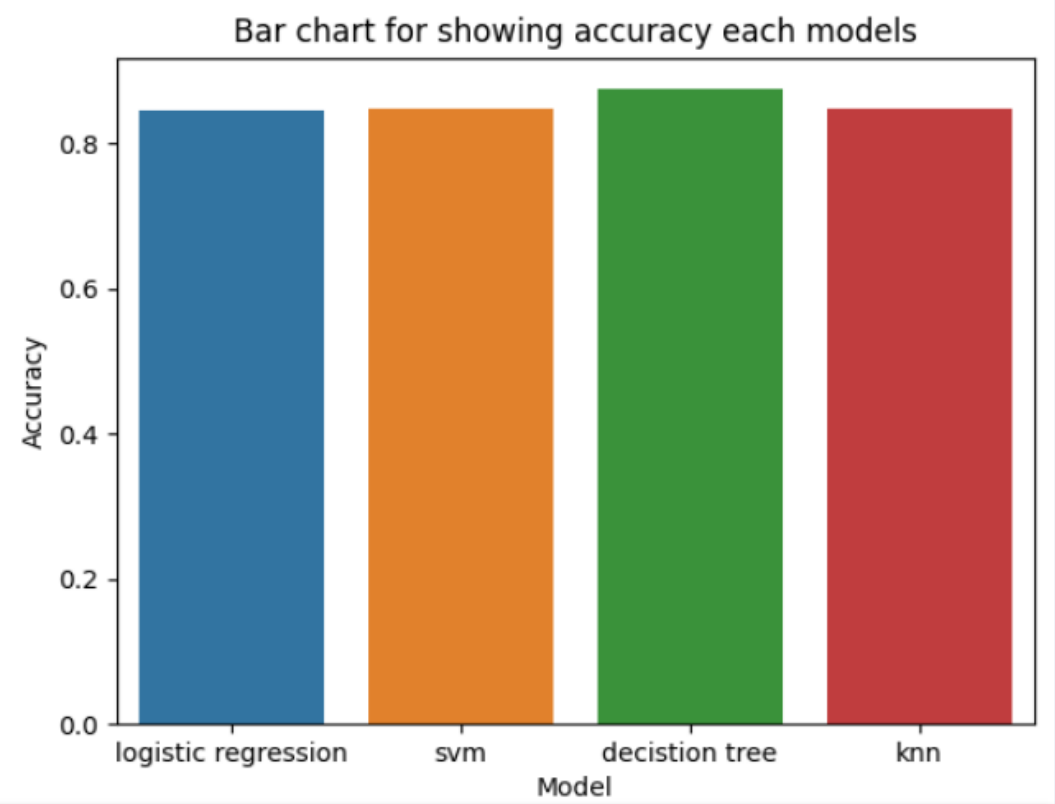
Section 5

Predictive Analysis (Classification)

Classification Accuracy

You can see our accuracy is extremely close but we do have a winner its down to decimal.

So you can see the decision tree model win by accuracy score is 0.875 or 87.50% accuracy.



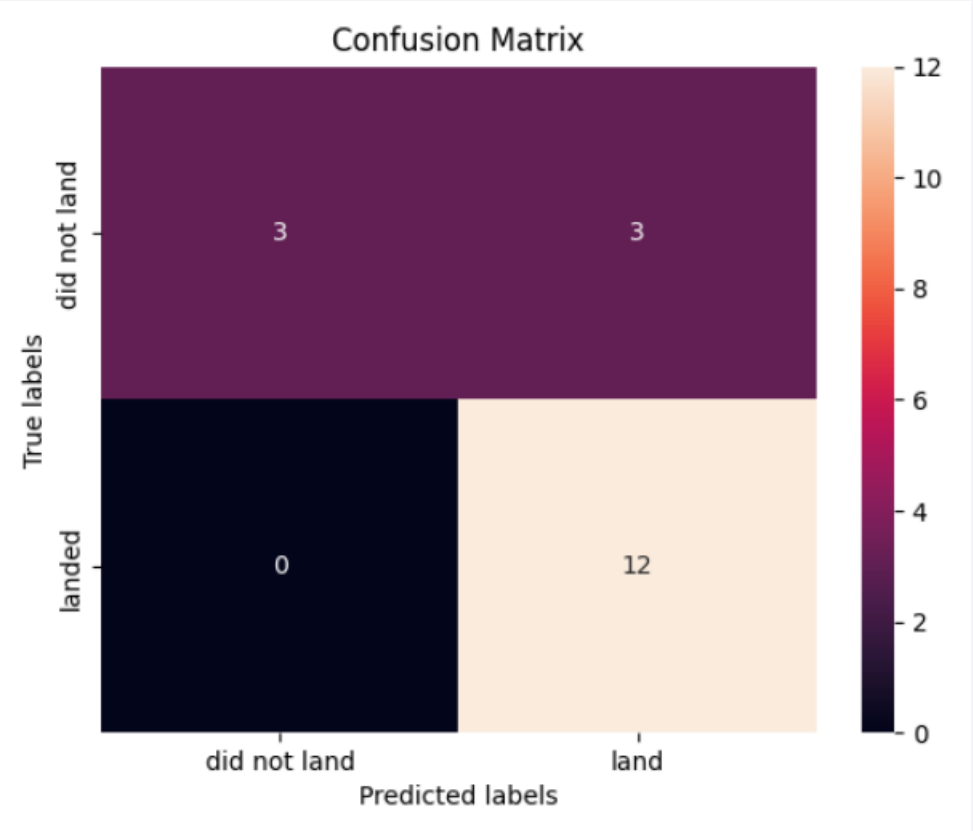

```
Best Model: Decision Tree Classifier
Best Accuracy: 0.875
Best Parameter: {'criterion': 'entropy', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix

You can see that the decision tree algorithm is able to distinguish between the different classes.

However, we observe that the major issue lies with true positives. Specifically, when the true label is 'landed', the decision tree predicted 'landed' correctly only 12 times.

In my opinion that the model may need to be refined to improve its accuracy in predicting this particular class.



Conclusions

- Flight number positively impacts successful landings
- Payload mass negatively impacts the likelihood of first stage return
- CCAFS LC-40 has a lower success rate compared to KSC LC-39A and VAFB SLC 4E
- Success rate has been consistently improving since 2013
- KSC LC-39A has the highest number of successful landings, while CCAFS SLC-40 has the lowest number of successful landings
- Success rates are higher for low-weighted payloads than for heavy-weighted payloads
- Decision tree model has an accuracy score of 0.875 or 87.50%, which is higher than other models

Thank you!