IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chanboribou Prak (Boribou)
22nd August 2024

Outline

Executive Summary

- Summary of methodologies
  - Data Collection with Web Scraping
  - Data Collection with API
  - Data Wrangling
  - Data Analysis with SQL
  - Data Analysis using Pandas and Matplotlib Library
  - Data Visualization with Folium
  - Data Visualization with Plotly Dash
  - Classification on Machine Learning
- Summary of all results
  - Exploratory Data Analysis (EDA)
  - Interactive Visual Analytics and Dashboard
  - Predictive Analysis (Classification)

Introduction

- **Project Background and Context**
  - SpaceX is planning to launch a new rocket called SpaceX Falcon 9 which its price is relatively cheaper compared to other company that has a higher cost (165 million dollars) and can also be reused on the first stage
  - The prediction of a successful rocket launch can be depended on various factors such as the rocket's engineering process, the mission parameters (the factors contributing to the launch), etc.
  - Space Y is competing against SpaceX by gathering information on the SpaceX Falcon 9 Launching Stage in order to determine the price of each launch and using machine learning algorithms to predict if SpaceX will continue to use the first stage
- **Problems you want to find answers**
  - Predict whether or not SpaceX Falcon 9 will be reused on the first stage
  - Determining the cost of the launch

Section
1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected using SpaceX API and extracting information from Wikipedia using Web Scrapping
- Perform data wrangling
  - The Success/Failure of the SpaceX Falcon 9 was converted into Training Labels (1 or 0)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

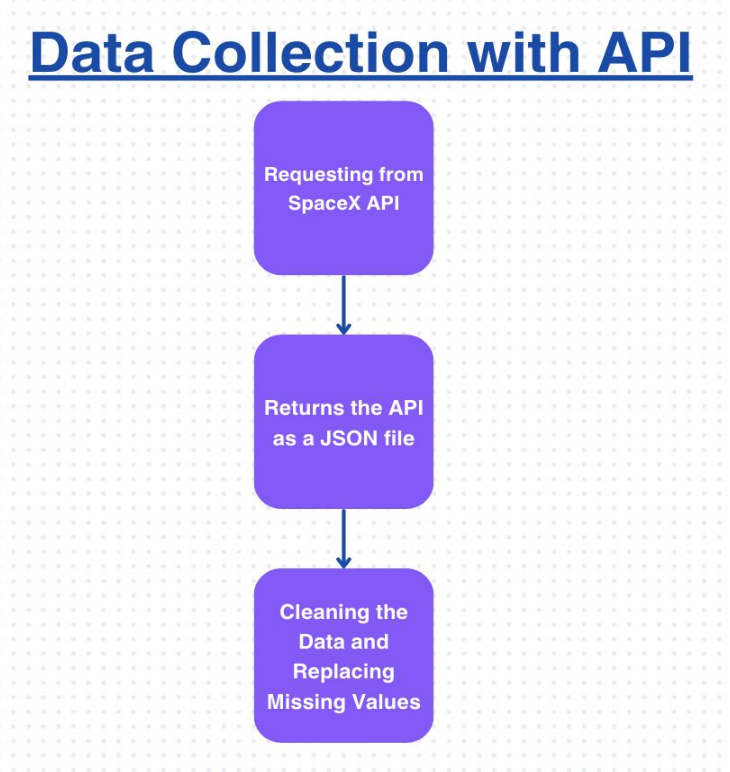
Data Collection – SpaceX API

- We request the information about the SpaceX Falcon 9 through the SpaceX API using the Get Request **(Requesting from SpaceX API)**

- We return the API by using the .json() to return as a JSON file and using the JSON_normalize function to be converted into a pandas DataFrame **(Returns the API as a JSON File)**

- We clean the data by checking if there are duplicates or missing information by converting the information on the given id using the converted dataframe **(Cleaning the Data and Replacing Missing Values)**

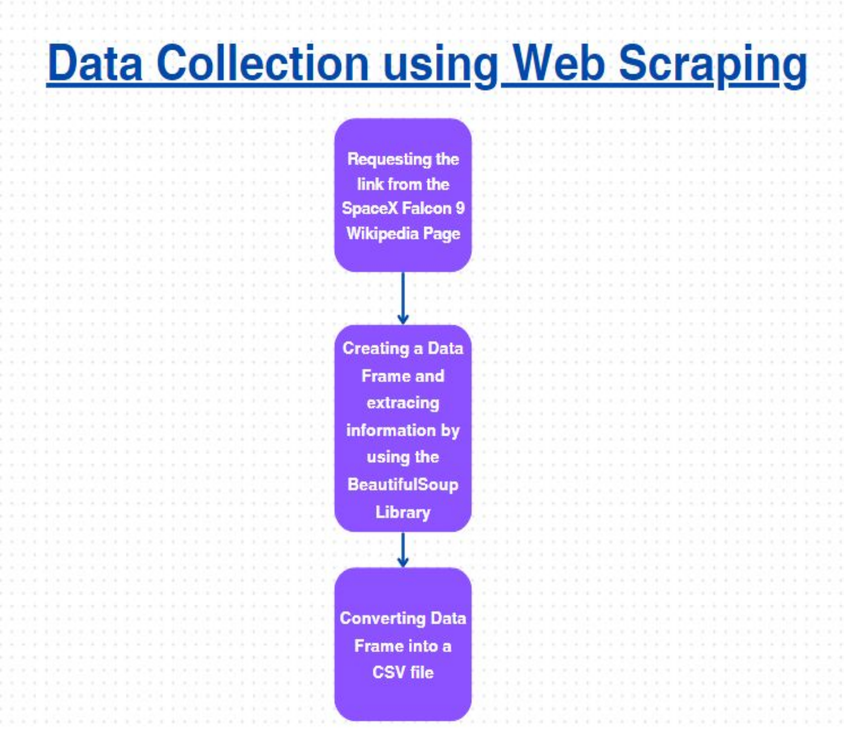
- Finally, we replace the missing values by using the .mean() function to replace it with the np.nan value **(Cleaning the Data and Replacing Missing Values)**



Github URL:
https://github.com/chanboribou/Applied-Data-Science-Capstone

Data Collection - Scraping

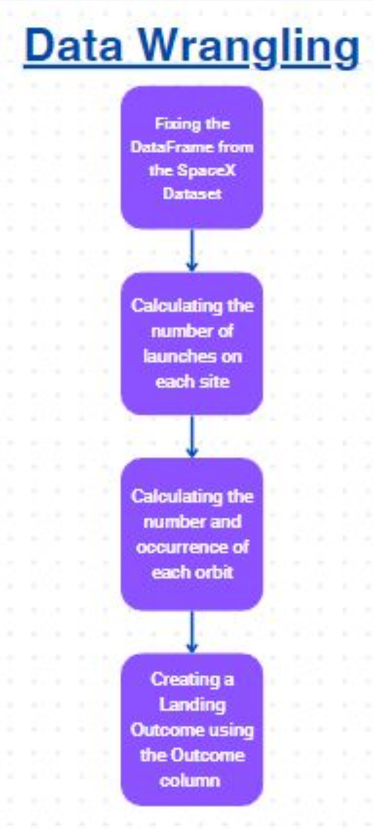
- We use the HTTP Get Method to launch the SpaceX Falcon 9 Wiki Page **(Requesting the link from the SpaceX Falcon 9 Wikipedia Page)**
- We create a BeautifulSoup Object as we can use it to extract information from the Wiki Page and the response variable also acts as the HTML response **(Requesting the link from the SpaceX Falcon 9 Wikipedia Page)**
- Now we use the Beautiful Soup library to extract all of the columns from our Beautifulsoup Object we just created a while ago by using the method extract_column_from_header() **(Creating a DataFrame and Extracting information by using the BeautifulSoup Library)**
- We then parse the HTML pages by first creating a dictionary of the columns then use the method .append() to extract the information to the given key **(Creating a DataFrame and Extracting information by using the BeautifulSoup Library)**



Github URL: https://github.com/chanboribou/Applied-Data-Science-Capstone

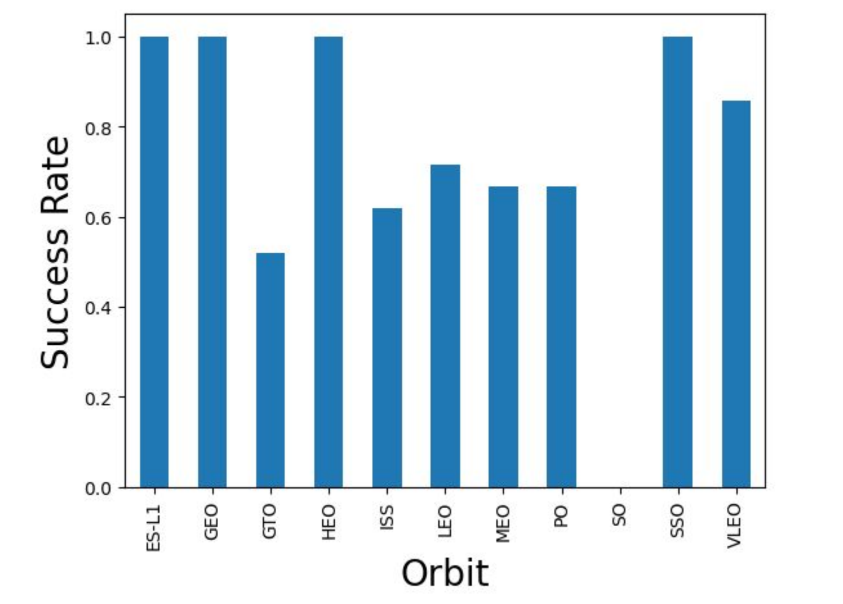
Data Wrangling

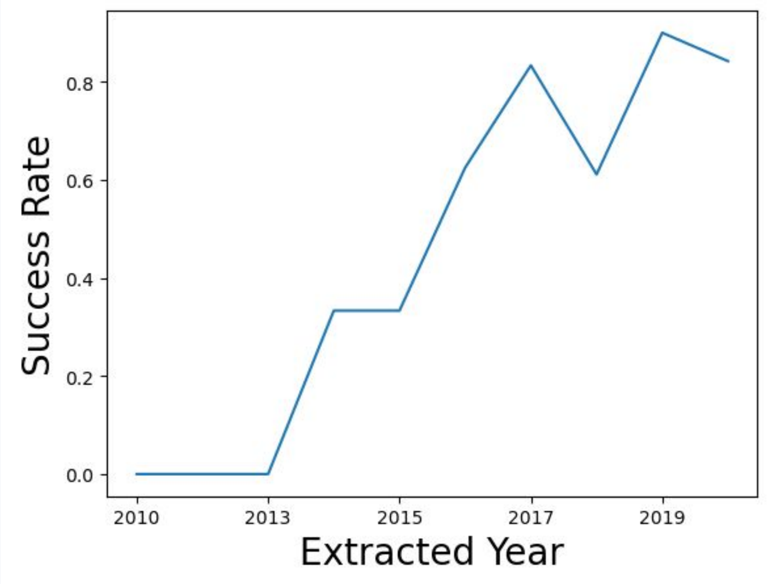
- We first extract the Data Frame from the SpaceX Dataset **(Fixing the Data Frame from the SpaceX Dataset)**
- We fixed the Data Frame by calculating the percentage of the missing values **(Fixing the Data Frame from the SpaceX Dataset)**
- We also calculate the number of launches on each site by using the function .value_counts() to determine the amount **(Calculating the number of launches on each site)**
- We also calculate the number of orbit occurences by using the same function, .value_counts() to determine the amount as well as calculating the number of the mission outcome of the orbit occurence **(Calculating the number of launches on each site)**
- We uses the value_counts() to calculate for the landing outcome label by using the outcome column **(Creating a Landing Outcome using the Outcome column)**



Github URL: https://github.com/chanboribou/Applied-Data-Science-Capstone

EDA with Data Visualization

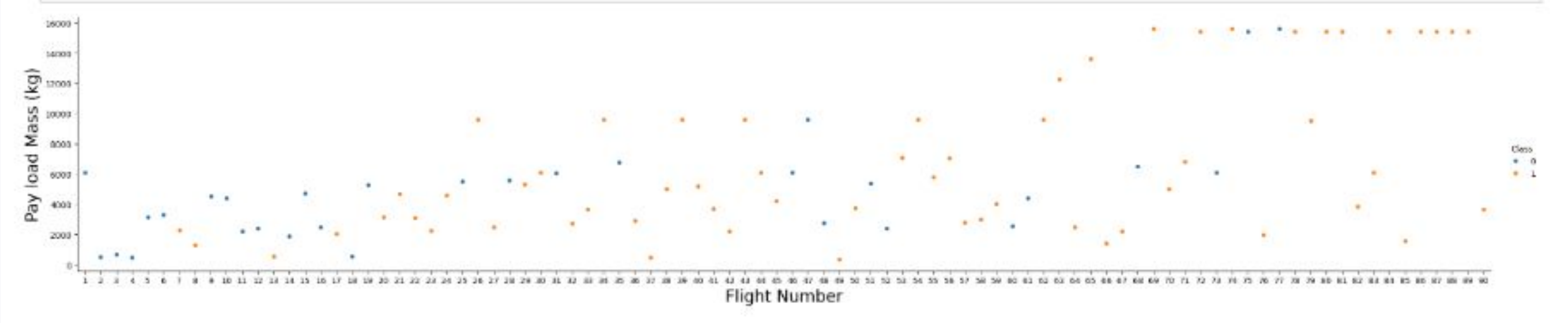


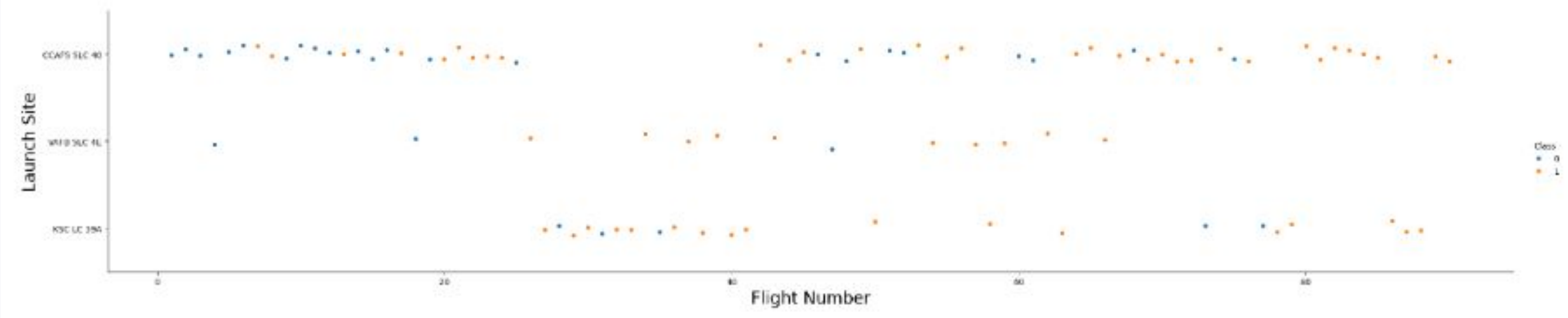


Github Link:
https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

EDA with Data Visualization

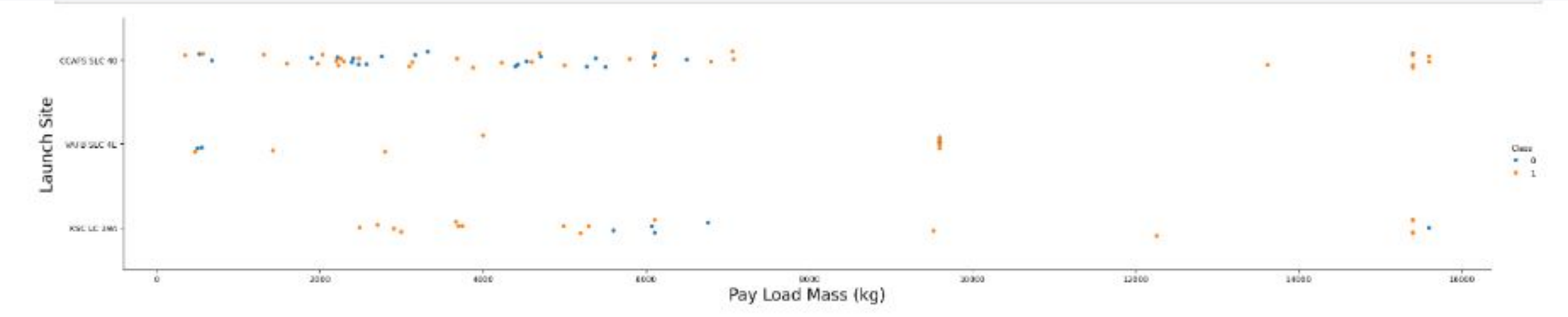
[Github Link:
https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb](https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb)

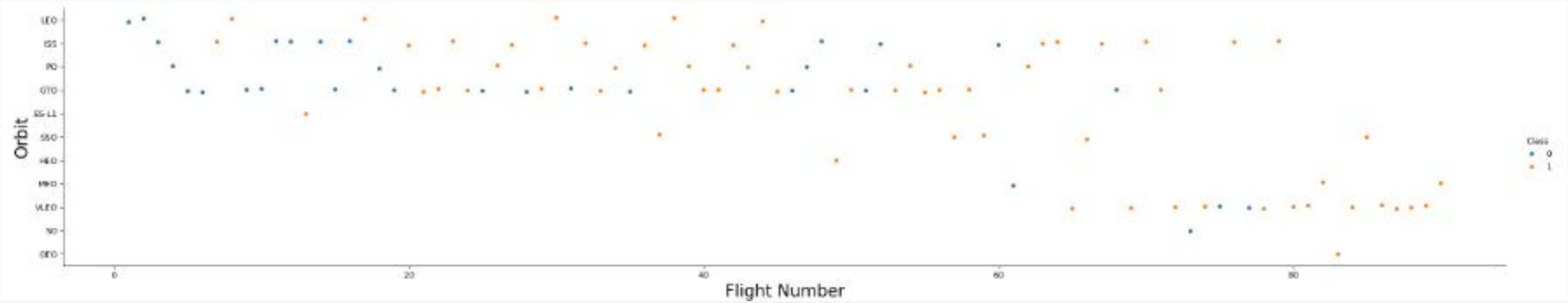




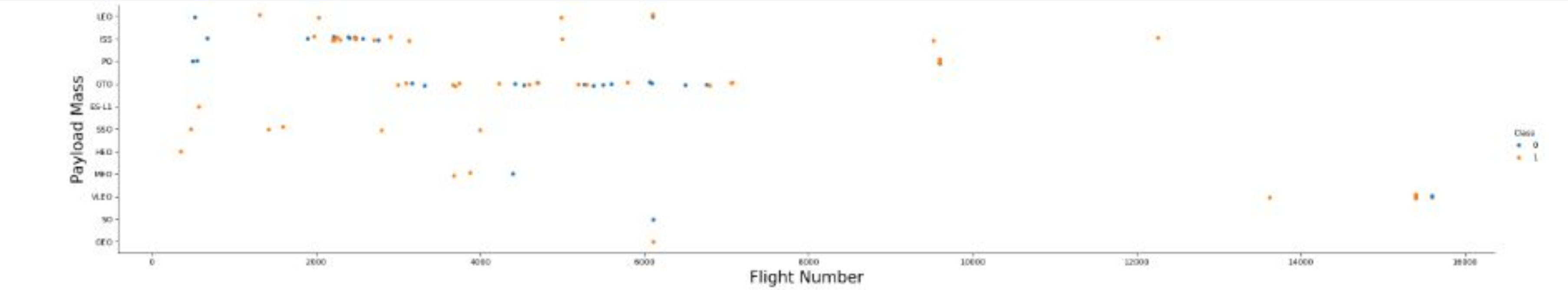
EDA with Data Visualization

Github Link:
https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb





EDA with Data Visualization



Github Link:
https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

EDA with SQL

Github Link:
https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

Done.

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

Out[30]:

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

- This SQL query displays the number of unique launch site using the function DISTINCT
- This SQL query displays the top 5 of booster version where the launch site is CCAFS LC-40 using the LIKE method

EDA with SQL

TOTAL(PAYLOAD_MASS__KG_)
45596.0

- This SQL query displays the total payload mass launched by NASA

min(DATE)
2015-12-22

- This SQL query displays the date of the first successful landing outcome in ground pad

COUNT(*)
101

- This SQL query displays the total number of both successful and unsuccessful mission outcomes

AVG(PAYLOAD_MASS__KG_)
2928.4

- This SQL query displays the average payload mass carried by the Booster Version

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

- This SQL query displays successful drone ship in which the payload mass is greater than 4000 but lesser than 6000

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

- This SQL query displays booster version that has the maximum pay load mass

EDA with SQL

Booster_Version	Launch_Site	Date
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

- This SQL query displays the number of failures from the booster version and launch site from 2015

Date	COUNT
2017-02-19	1
2017-01-14	1
2016-08-14	1
2016-07-18	1
2016-05-27	1
2016-05-06	1
2016-04-08	1
2015-12-22	1

- This SQL query displays the count of landing outcomes between 2010-06-04 and 2017-03-20

Build an Interactive Map with Folium

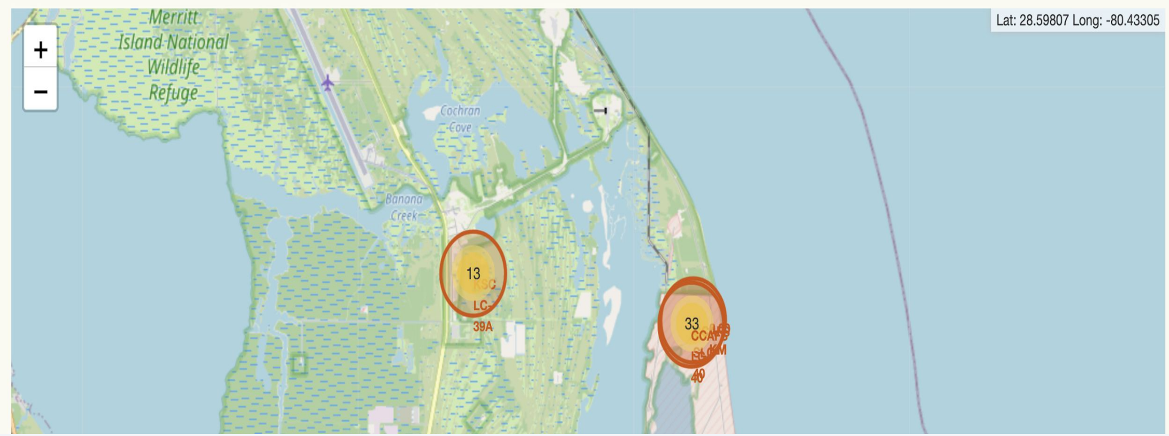
- We use the function folium circle and folium marker to add launch sites on the site map. This helps us to see where the SpaceX Falcon 9 launches during its first stage of launching
- https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb

Build an Interactive Map with Folium

- We add another folium marker to the marker cluster so that we can see the success and the failure of the landing outcome from each launch site launched by Spacex Falcon 9
- https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb

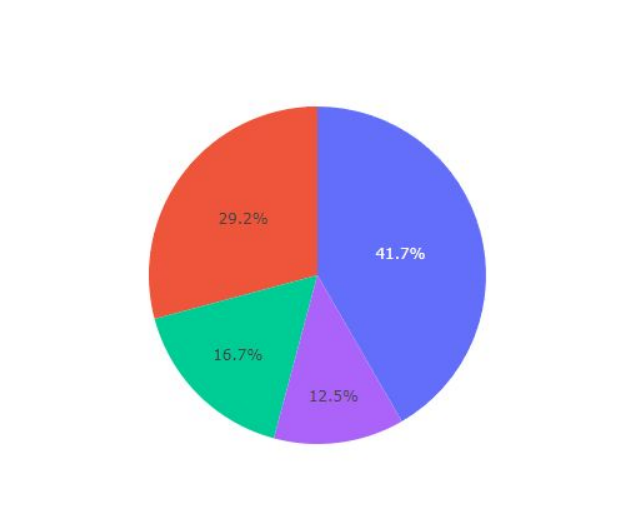
Build an Interactive Map with Folium

- We first calculate the closest coastline using the mouse position afterwards added the closest coastline calculation in the site map using the folium marker
- https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb

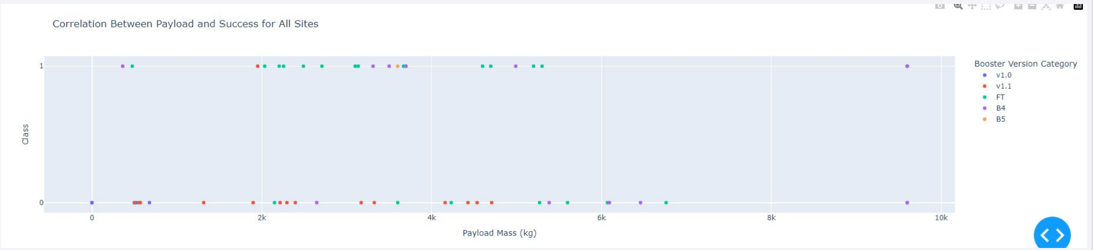


Build a Dashboard with Plotly Dash

- In the first picture, we created a pie chart for all the launch site to see which launch site has the most total success launch in terms of both successful and failed landing outcome



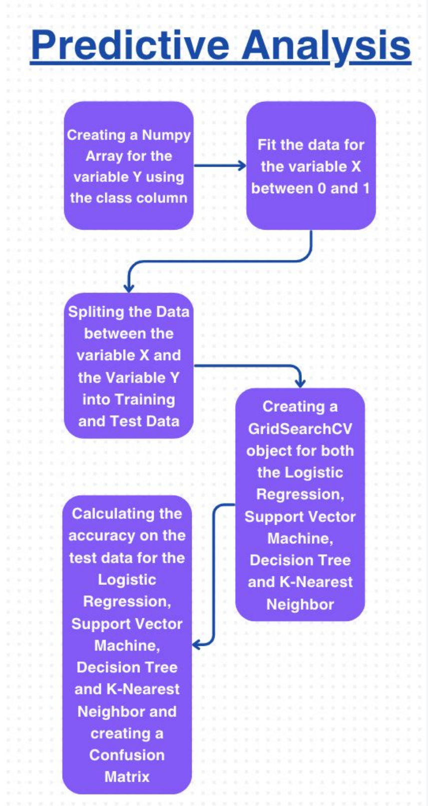
- In the second picture, we created a scatter plot to determine which payload mass has more a successful launch rate based on each launch site



- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Dash%20App.py

Predictive Analysis (Classification)

- We use the method to create objects and make prediction such as GridSearchCV which shows the accuracy from each of the 4 machine learning algorithms and using the confusion matrix to look at the prediction whether landing outcome from the SpaceX Falcon 9 will be reused
- We create a Numpy Array for the variable Y using the class column by using the to_numpy() function **(Creating a Numpy array for the variable Y using the class column)**
- We fit the data for the variable X between 0 and 1 using the StandardScaler() function and .fit_transform() function **(Fit the data for the variable x between 0 and 1)**
- We use the train_test_split function to split the data from both the variable X and the variable Y in order to fit the training and testing data (**Splitting the data between the variable X and variable Y into training and test data)**
- We use the GridSearchCV function in order to use the parameter as a dictionary and fit the data in order to make a prediction and made accuracy for the four machine learning algorithms **(Creating a GridSearchCV object for both the Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbor)**
- Finally we calculate the accuracy by using the score function afterwards we use the confusion matrix function to look at the prediction of the landing outcome of SpaceX Falcon 9 **(Calculating the accuracy on the test data for the Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbor and creating a Confusion Matrix)**
- **Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20ML%20Prediction.ipynb**



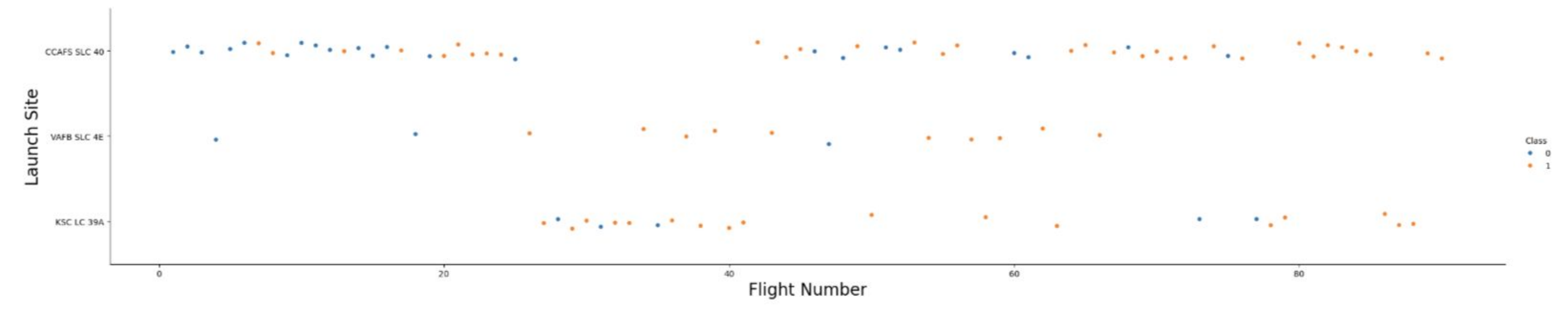
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section
2

Insights drawn from EDA

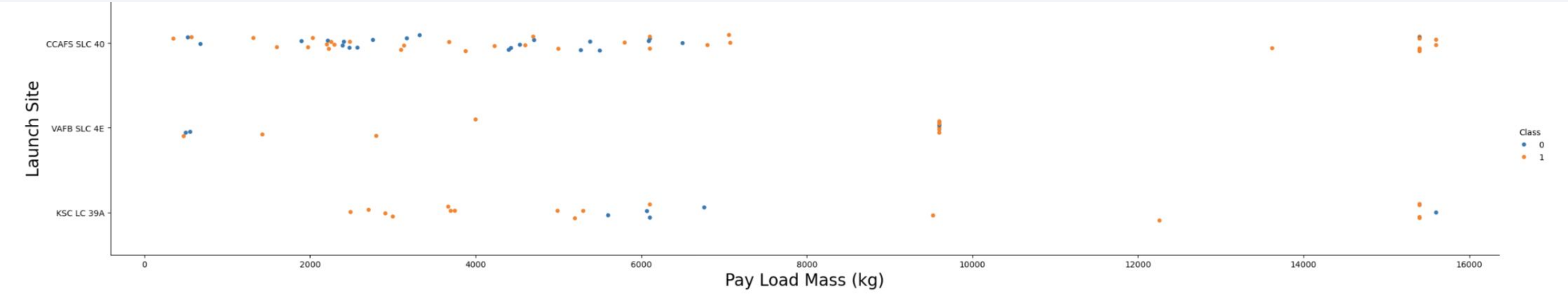
Flight Number vs. Launch Site



This Scatter Plot shows the relationship between Flight number and Launch Site, we can see that they are keep improving their approach on launching the rocket in order to get a successful landing outcome from each of three launch site especially the launch site, CCAFS SLC 40 in which they have a few failed landing outcome at first but overtime they have improved their strategies to every flight number that they have in order to make a successful landing outcome

- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

Payload vs. Launch Site

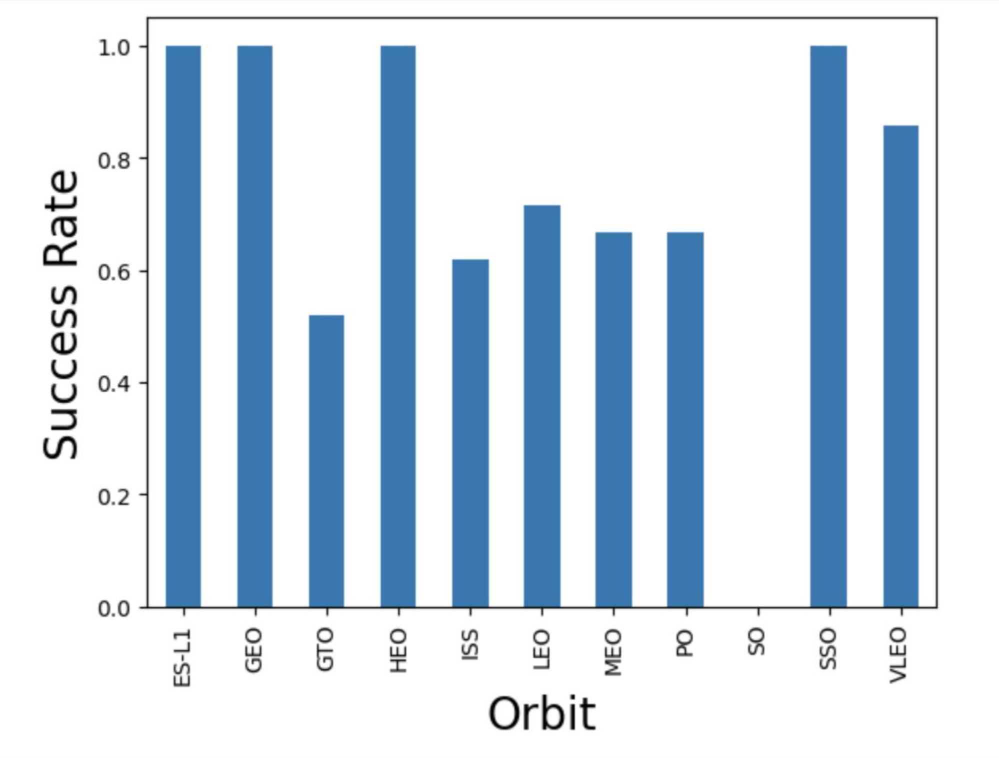


This scatter plot here shows the relationship between the Launch Site and the Pay Load Mass, based on this, we can say that the Pay Load Mass that is less than 8000 kg considers to have a successful landing outcome while those higher than 8000 kg is only applicable in launch site such as CCAFS SLC 40 and KSC LC 39A

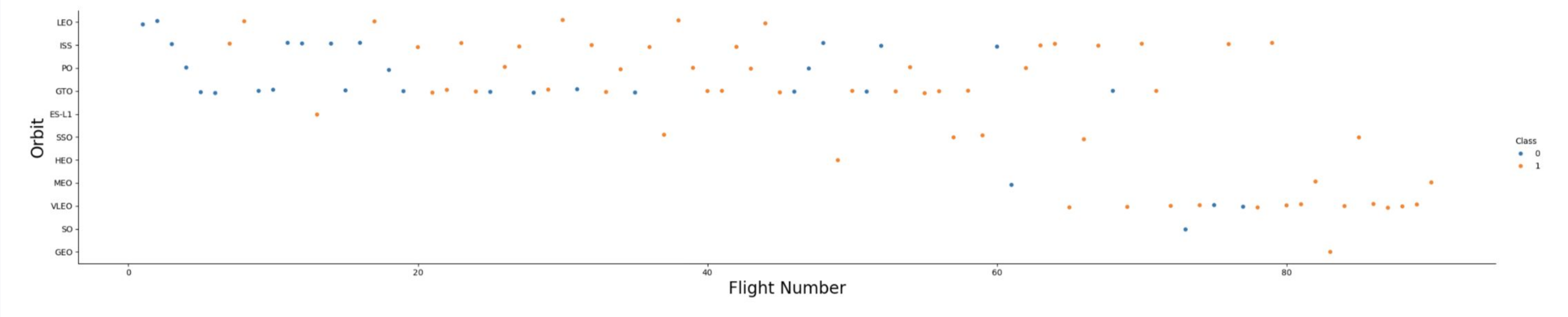
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

Success Rate vs. Orbit Type

- In this bar graph, we can see the relationship between the Success Rate and the Landing of the Orbit, from what we have observed, the orbit with a higher success rate are
  - ES-L1
  - GEO
  - HEO
  - SSO
- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb



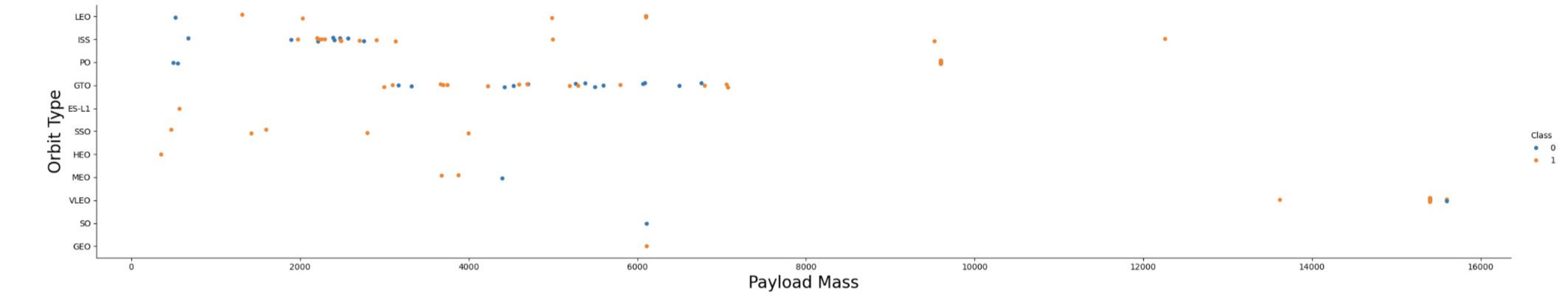
Flight Number vs. Orbit Type



In this scatter plot, we can see the relationship between the Orbit and the Flight Number. In the beginning, the Flight number didn't successfully launched at the assigned orbit at first but over time with their strategies and their improvement on their failed landing outcome, they have landed on their designed orbit successfully.

- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

Payload vs. Orbit Type



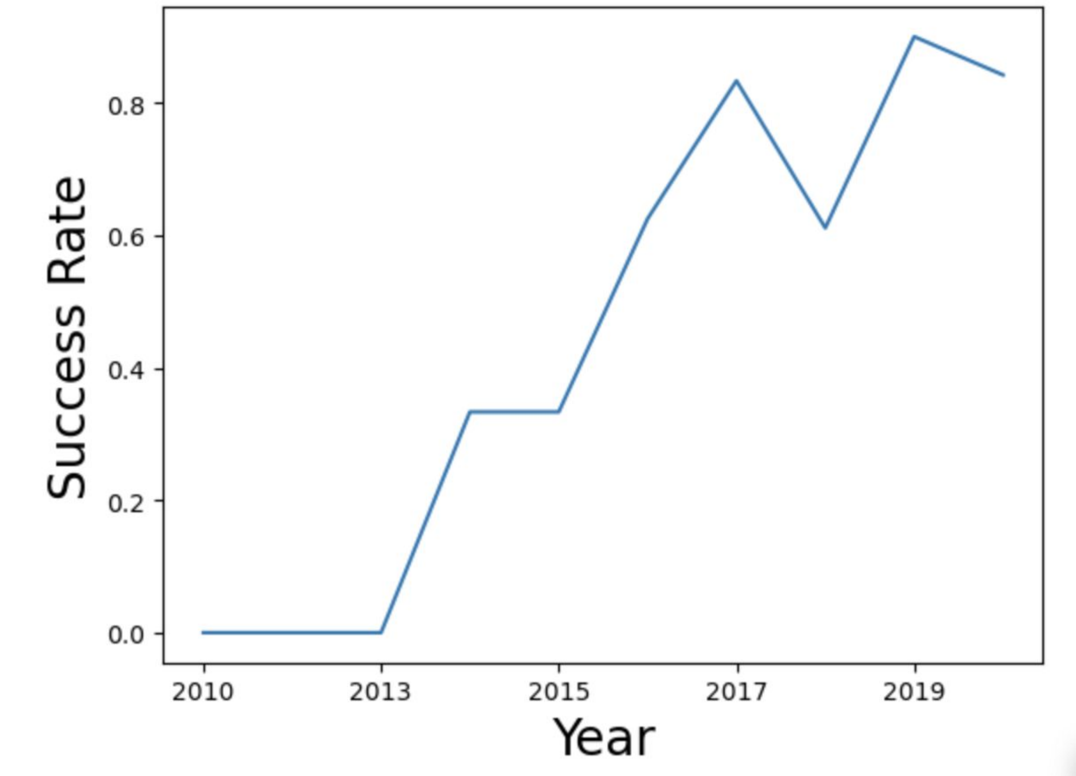
In this scatter plot, we can see the relationship between the Payload mass and Orbit Type.

- We can't determine the relationship with GTO orbit and the Payload Mass (kg)
- The ones with Heavy Payload Mass has a successful rate of landing outcomes in their designated outcome
- ISS has the most successful landing outcome on their designated orbit despite having a smaller Payload Mass
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb

Launch Success Yearly Trend

- Based on the bar graph and the relationship between the Success Rate and Extracted Year, we can say that the launch success trend has been increasing over the a decade until it started to decrease in 2020 due to a global pandemic

- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20EDA%20Visualization.ipynb



All Launch Site Names

- In this SQL Query, we have used DISTINCT function for Launch Site as this function returns a value that is different from others based on the SQL dataframe from the SPACEXTABLE SQL dataframe
- After we queried the SQL, we can see 5 different launch sites as shown in the picture to the right
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

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

Launch Site Names Begin with 'CCA'

- For this SQL query, we used the like function as it is used to search the word or number that is similar to the one that is commended from the SPACEXTABLE SQL dataframe
- We also use the limit function to limit the number of result we wanted to see when the SQL was queried
- After we queried the SQL, we can see 5 different launch site that has name 'CCA'
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_
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

- In this SQL query, we have used the TOTAL function where it sums all the number of a specific integer or float number, in this case, we use the payload mass

- We also use the WHERE function where it is used to find a specific keyword, in this case we use it to find the boosters that was launched by Nasa

- As of this result, we get 45596.0 kg for the total payload mass booster that was carried by Nasa

- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

TOTAL(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- In this SQL query, we use the AVG function where it displayed the average number of that specific integer or float column, in that case, we use the same column from last time to calculate the average
- We also use the WHERE function from last time to find the Booster version that is F9 V1.1
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- In this SQL query, we use the min function for the date in which returns the smallest value in the column
- We also use the where function to find the landing outcome that is successful on ground pad
- As a result to this, We get the date December, 22nd, 2015 in which is when the landing outcome became successful on ground pad
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- In this SQL query, we first SELECT booster version then we use the where function for the payload mass
- Then we use the sub query in order to avoid errors afterwards we get to use the BETWEEN function in which the person specifies the minimum value and the maximum value from a specific variable, in this case, we put 4000 as the minimum value and 6000 as the maximum value from the SPACEXTBL dataframe
- As a result, we get the answer (picture to the right) in which shows the booster version that has a payload mass between 4000 and 6000 kg
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- For this SQL query, We use the function Count which is used to total the number of a specific row, in this case we use SPACEXTBL to count all the rows available

- We also use the LIKE function to find whether the mission outcome is success or failed

- As a result to this, we find out that there are 101 total of flights combined with successful mission outcomes and failed mission outcomes

- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

COUNT(*)
101

Boosters Carried Maximum Payload

- In this SQL query, we are using the WHERE function for the payload mass of the booster version specifically and then we use a subquery before using the max function for payload mass as without using a subquery, it will encounter an error
- We then use the MAX function for the Payload Mass to find the booster version that carried the Maximum Payload Mass
- As a result to this, we queried the booster version with the maximum Payload mass by using the max function
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

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

- From this SQL query, we select both the booster version, the launch site and the date from the SPACEXTBL dataframe
- Then we use the WHERE function for the landing outcome and sets the function to find if the landing outcome failed and if the date happened on 2015 so that the function can do that easily
  - As a result, we find two booster version where the landing outcome failed back in 2015
  - Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

Booster_Version	Launch_Site	Date
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- We first select the landing outcome, use the count function on the * variable (which is for the entire table) to sum all the number from the row and then we put both of the date and the * variable in the variable COUNT from the SPACEXTBL
- We use the BETWEEN function on the date to select between 2010-06-04 and 2017-03-20
- And then we group by landing outcome in which is used to group and count the number of landing outcome and then finally we use the order by function on our count variable to see the types of landing outcome from success, failure, and precluded and controlled and uncontrolled
- As a result, we can see the number of landing outcome that are successful, failed or is being precluded at the moment
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20SQL.ipynb

Landing_Outcome	COUNT
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

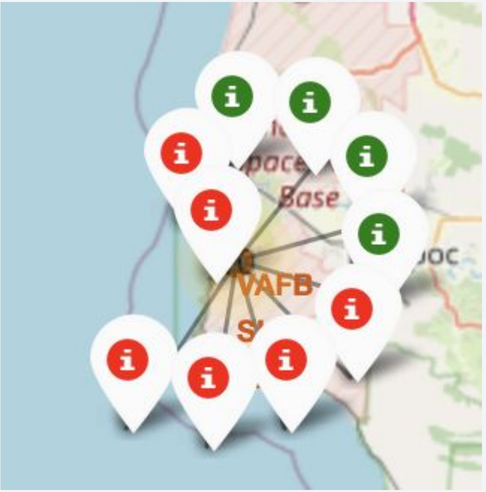
Section
3

Launch Sites Proximities Analysis

Launch Site added on Site Map

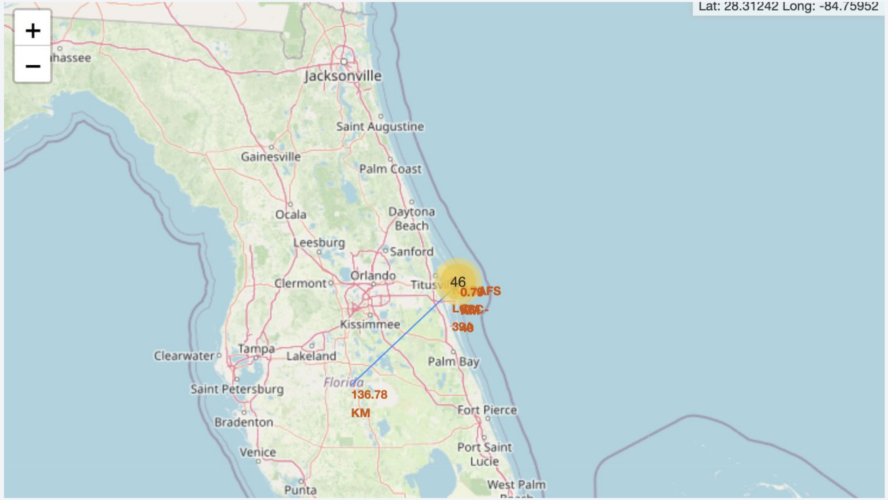
- In this map, we can see that there are four launch sites where the SpaceX Falcon 9 was launched marked as a red circle using the folium marker and folium circle functions such as:
  - **VAFB SLC-4E**
  - **KSC LC-39A**
  - **CCAFS SLC-40**
  - **CCAFS LC-40**
- **Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb**

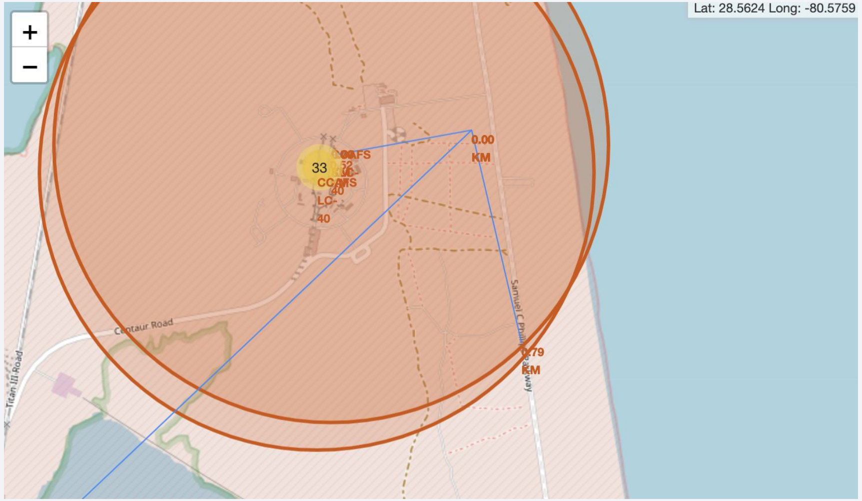
Successful and Failing Landing Outcome on each Launch Site



- In these screenshot, we can see that each flight number are being added to the assigned launch site
- When we click on one of the yellow, red or green circle, we can see the map marker showing how many successful and failed landing outcome are there in each flight number assigned to the launch site
- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb

The distance between the city, highway and railway and the launch site





- In this picture, we can see the distance from the railway, highway, city and the launch site in which we activate using the site map, we can see the number of distance between them

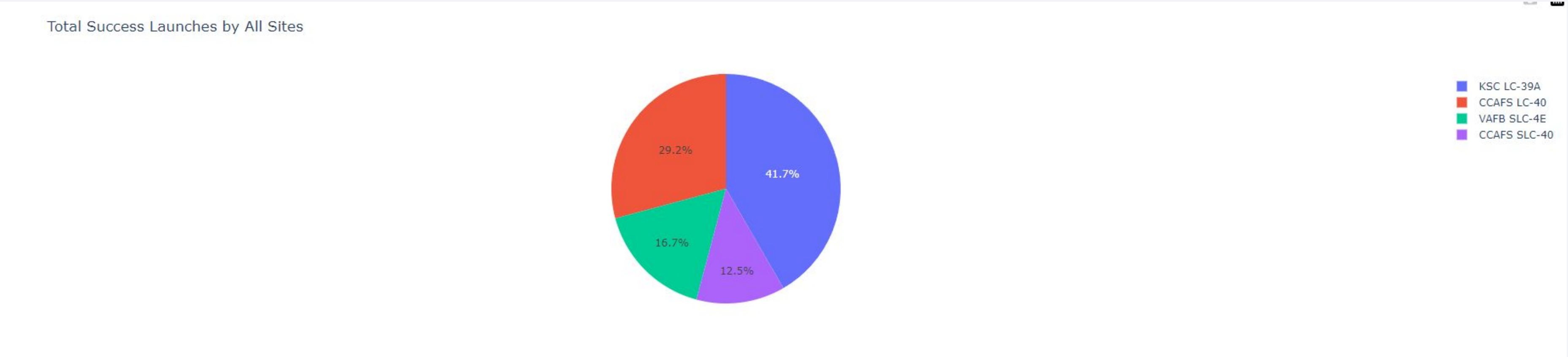
- We also use folium marker to connect the line of the distance between highway and launch site, railway and launch site, and city and launch site

- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Folium.ipynb

Section
4

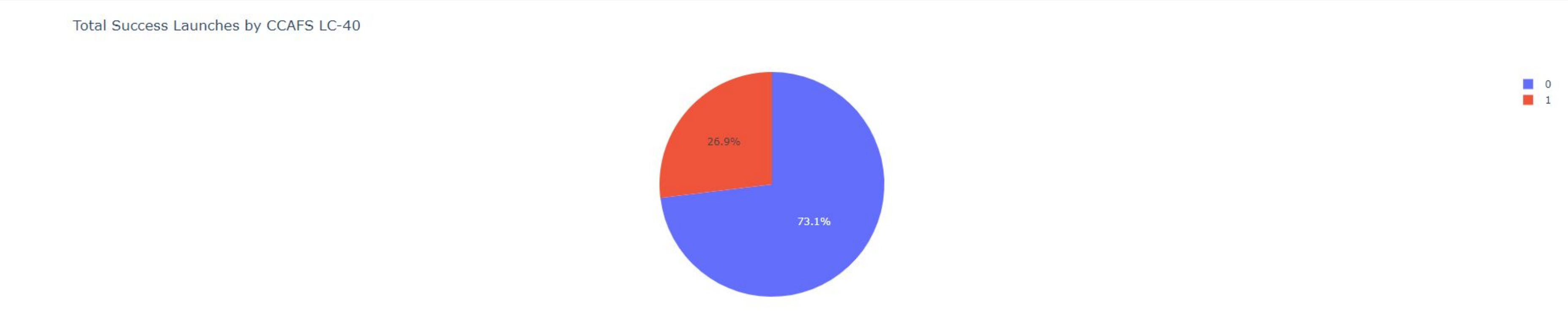
Build a Dashboard with Plotly Dash

Pie Charts of Success Launches from All Launch Site



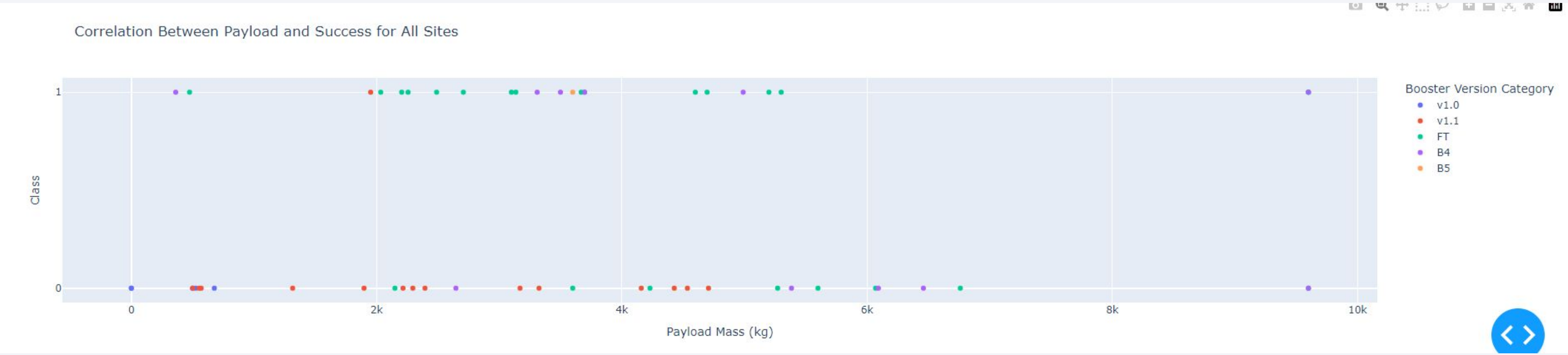
- In this Pie chart, we can see the percentage of total success launches by all sites with
  - KSC LC-39A having highest total success rate by 41.7%
  - CCAFS SLC-40 having lowest total success rate by 12.5%
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Dash%20App.py

Percentage of Success Launches by CCAFS LC-40



- This pie chart shows 73.1% of the successful landing outcome compared to the 26.9% which shows failed landing outcome (the purple is supposed to represent 1, the red is supposed to represent 0, ignore the error column)
- Github link https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Dash%20App.py

Correlation Between Payload and Success for All Sites



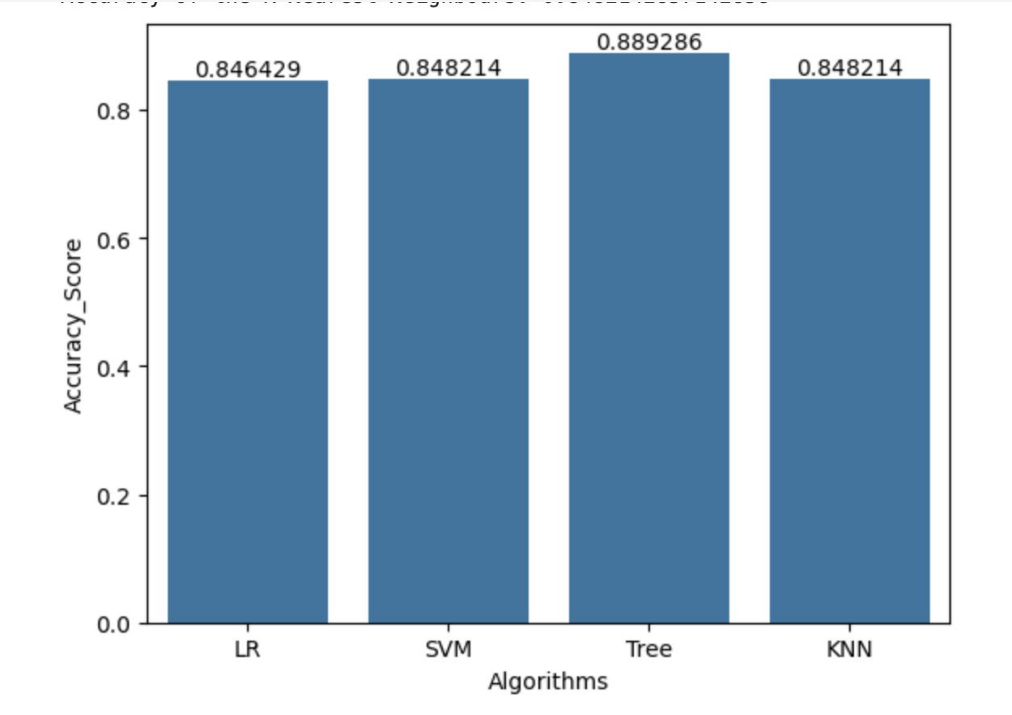
- In this scatter plot, we can see that Payload Mass (kg) that is lower than 6000k mainly have the most successful landing site whereas the B4 from the booster version category can be applicable
- Github link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20Dash%20App.py

Section
5

Predictive Analysis (Classification)

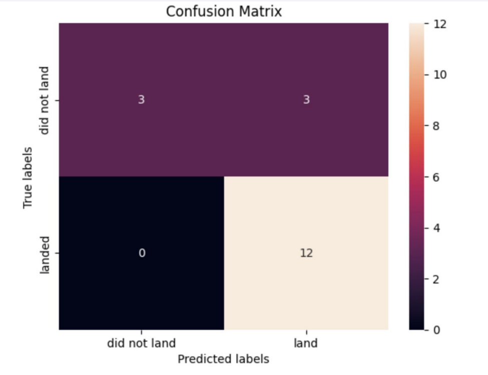
Classification Accuracy

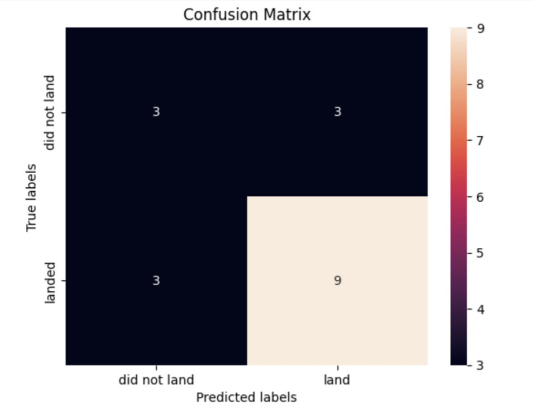
- In this bar graph, we can see the classification accuracy in the four machine learning algorithms that we have used to predict the landing outcome of SpaceX Falcon 9
- Out of all the four algorithms in the bar graph, the decision tree has the highest accuracy score
- KNN and SVM has a similar accuracy score
- Logistic Regression has the lowest accuracy score
- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20ML%20Prediction.ipynb



Confusion Matrix

- The confusion matrix for both Logistic Regression, Support Vector Machine, and K-Nearest Neighbours:
  - True Positive - 3
  - False Positive - 3
  - False Negative - 0
  - True Negative - 12
- The confusion matrix for Decision Trees:
  - True Positive - 3
  - False Positive - 3
  - False Negative - 3
  - True Negative - 9
- Github Link: https://github.com/chanboribou/Applied-Data-Science-Capstone/blob/main/SpaceX%20ML%20Prediction.ipynb





Conclusions

- The Total Launch Site have been successful since 2013 until 2020 due to the global pandemic

- All of the launch sites in the dashboard, KSC LC-39A has the highest total success rate by 41.7%

- ES-L1, GEO, HEO, SSO has the highest success rate by 1.0

- The Decision Trees have the highest accuracy score

Thank you!