Data Science Capstone Project: SpaceX Falcon 9 First Stage Landing Prediction

Obiamaka Ugwumba
April 9, 2022

Background Image: https://www.capstoneaxis.com/success-and-failures-of-space-x/

Outline

Background Image: https://www.capstoneaxis.com/success-and-failures-of-space-x/

Executive Summary

- This Capstone Project is to predict the first stage landing success rate for SpaceX Falcon9 which will also help determine the cost of a launch in comparison with other providers.
- The methodologies implemented for this task are:

  a. Data Collection, Web Scraping, and Wrangling

  b. Exploratory Data Analysis and Visualization

  c. Interactive Data Visualization with Folium

  d. Predictive Analysis (Classification)

- Summary of all results:

  - The data displays the relationship between features such as Flight Number and Pay load mass as well as Flight Number and Launch Site among other factors like orbit type, location, and proximity to the launch site that may impact the success rate of the first stage land.

  - Additionally, our predictive analysis indicated that the best machine learning model for first stage landing for SpaceX Falcon9 is the Decision Tree.

Introduction

- In this project, the aim was to predict the landing success of the first stage SpaceX Falcon9. Currently, SpaceX advertises Falcon9 rocket launches at a cost of 62 million dollars. In contrast, other providers market cost is an upward of 165 million dollars each. SpaceX launch costs are lower due to the reuse of the first stage. Hence, determining if the first stage will land successfully will also determine the cost of a launch.
- Most unsuccessful landings are planned and SpaceX performs a controlled landing in the Ocean.
- This brings us to the main question: Given launch record values such as flight number, launch site, payload, payload mass, orbit, launch outcome, etc. Can we predict the first stage landing success?

Section 1

Methodology

Background Image: https://www.teslarati.com/spacex-falcon-9-booster-launch-debut-photos/

Methodology

The methodology used for out findings are:

1. Data collection and wrangling methodology:
    a. SpaceX API
    b. Web Scraping
    c. Data Wrangling
2. Perform exploratory data analysis (EDA) using visualization and SQL
    a. SQL
    b. Pandas
    c. Matplotlib
3. Perform interactive visual analytics and dashboarding
    a. Folium
    b. Plotly Dash
4. Perform predictive analysis using classification models
    a. Logistic Regression
    b. Decision Tree
    c. Support Vector Machine
    d. K-nearest neighbors (KNN)

Data Collection – SpaceX API

The API used: "https://api.spacexdata.com/v4/launches/past

The API included a lot of information about SpaceX launches, requesting and parsing the SpaceX launch data, filtering the data to only include Falcon9 launches and eliminating Falcon1 yielded results.

The results included 90 Rows, 17 columns to include Flight Number, Date, Booster Version, etc.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	L
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Data Collection – Web Scraping

- For the web scraping process we used the wikipedia link: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- After requesting the HTML page from the above URL and we want to collect all relevant column names from the HTML table header.
- Our column names: Flight No, Date and time, Launch site, Payload, Payload mass, Orbit, Customer, Launch outcome.
- HTML tables in Wiki pages will most likely contain unexpected annotations and missing values N/A, inconsistent formatting, etc. this is a factor to consider while web scraping.

Data Collection – Data Wrangling

- Data wrangling process pertained to the missing values in the dataset.

  - calculating the mean of the Payload Mass and use it to replace the missing values in the data.

  - The number of Payload Mass missing changes to zero

  - The landing pad column will still contain the None values representing the landing pads that were not used.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	I
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Data Wrangling

- During this process we want to find some patterns in the data using EDA.
- There have been several cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident.
- Thus, after loading the dataset, calculate the percentage of the missing values, then we calculate the number of launches at each site.

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

Data Wrangling

- Next, we calculate the number and occurrences of each orbit because each launch aims to a dedicated orbit such as: LEO, GTO, SSO, etc.

```
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
```

Data Wrangling

- Next, we calculate the number and occurrence of mission outcome per orbit type.

  - The results are as follows:

    a. True Ocean means the mission outcome was successfully landed in the ocean.

    b. False Ocean means the mission outcome was unsuccessfully landed in the ocean.

    c. True RTLS means the mission outcome was successfully landed to a ground pad.

    d. False RTLS means the mission outcome was unsuccessfully landed to a ground pad.

    e. True ASDS means the mission outcome was successfully landed to a drone ship.

    f. False ASDS means the mission outcome was unsuccessfully landed to a drone ship.

    g. None ASDS and None None represent a failure to land.

Data Wrangling

- Next, we calculate the number and occurrence of mission outcome per orbit type.
  - The results are as follows:
    * There are 7 False RTLS meaning that the mission was unsuccessful in landing on the ground pad
    * 0 True ASDS implies that there were none that landed on a drone ship.
    * 3 False ASDA meaning they were unsuccessful lands on the drone ships.
    * 4 True Ocean means those were successful in ocean
    * 2 True RTLS means successful landing on the ground pad.

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

Data Wrangling

- Next, we create a landing outcome label from Outcome column
- The code will generate a “zero” for a bad outcome and a “one” for the successful first stage landing outcome.
- We can use this ”class” data to calculate the success rate mean: 0.666666666666666

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Exploratory Data Analysis and Visualization

- SQL was used to query the data and answer questions about the launch:

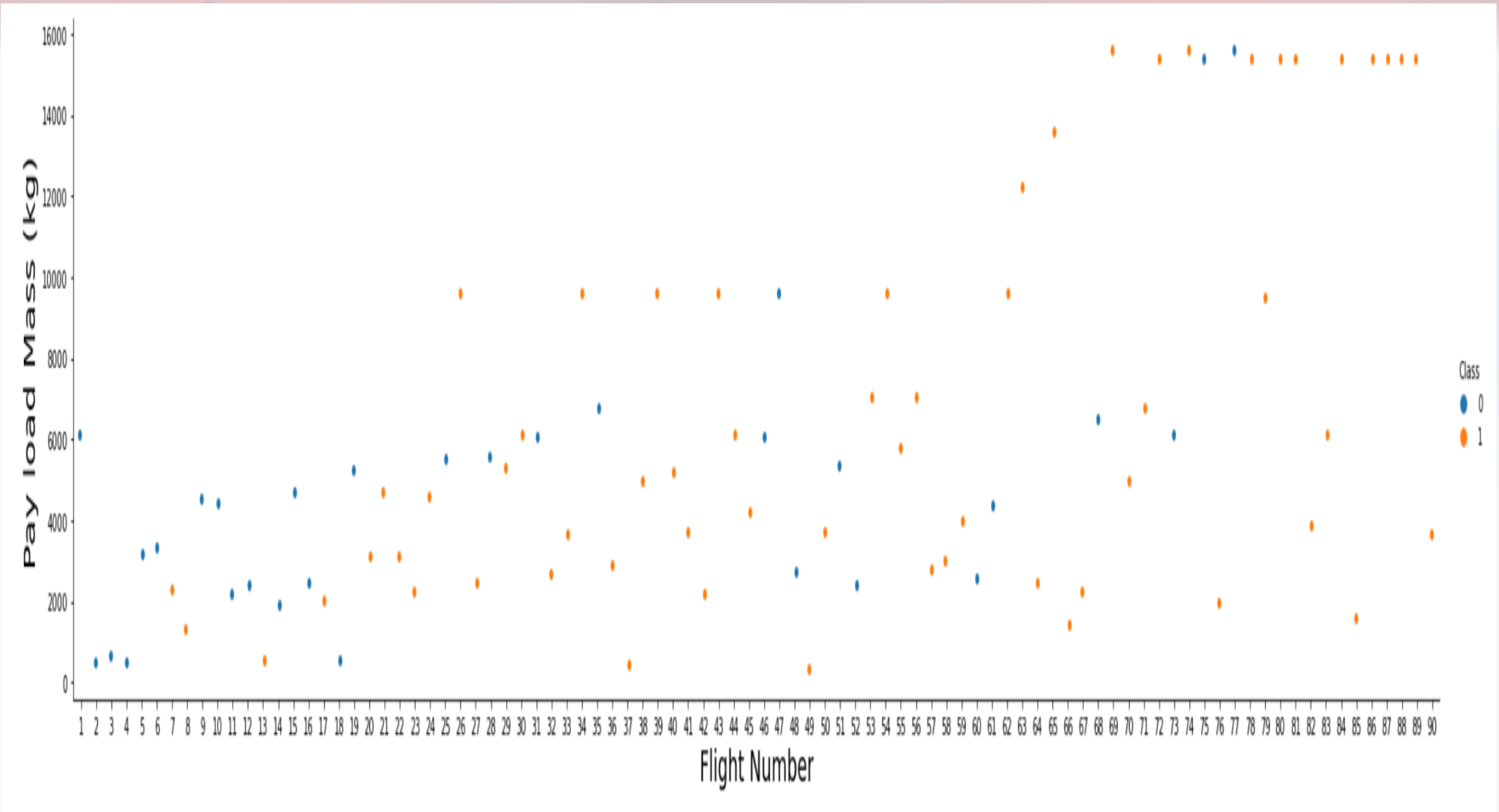
- The unique launch sites.
- Pay load mass carried by the Boosters launched by Nasa.
- Avg Pay load mass
- Total number of successful and failed missions

- Pandas software library was used for data manipulation and analysis

Exploratory Data Analysis and Visualization

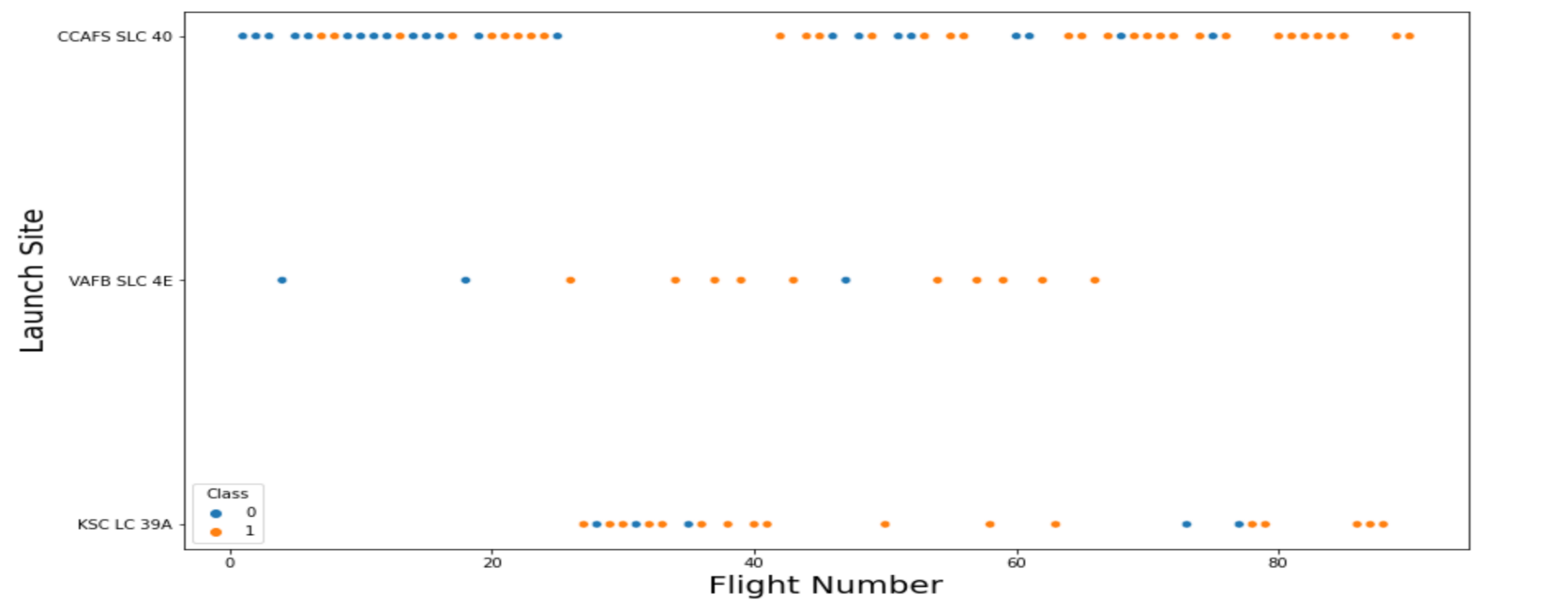
- Matplotlib is a plotting library used to plot out FlightNumber vs. PayloadMass
- Seaborn is a *Python data visualization library, based on matplotlib, providing a high-level interface for drawing attractive and information.*
- *These libraries were used to visualize data in the form of scatterplots, bar charts, and line graphs.*
- *With these were are able to visualize Flight Number vs Pay Load Mass, Flight Number vs Launch Site etc.*

Exploratory Data Analysis and Visualization



Exploratory Data Analysis and Visualization

- We can also visualize the relationship between Flight Number and Launch Site.



Exploratory Data Analysis and Visualization

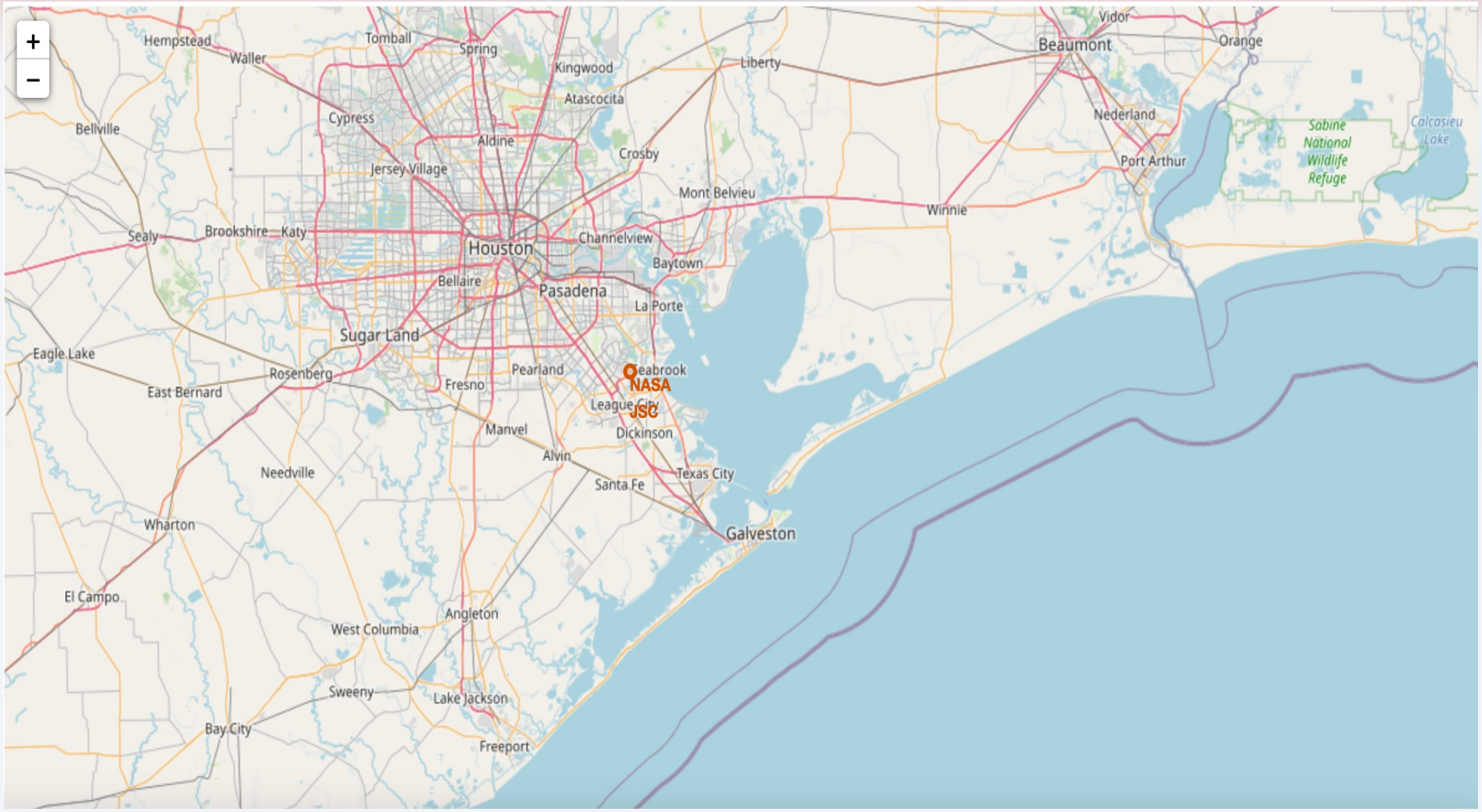
- Folium is an interactive library used to visualize data through the use of maps.
- This library was used to:
  - Mark all launch sites on a map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities

Exploratory Data Analysis and Visualization

- Folium is an interactive library used to visualize data through the use of maps.

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

Interactive Map with Folium



Build a Dashboard with Plotly Dash

- The Dashboard was used to generate an interactive site that permits us to toggle the input with the dropdown menu and range slider.
- It was also useful in creating a pie chart and scatterplot illustrating the outcome of the launches.
- Ploty Dash was able to provide:
  - Total launch success from individual launch sites
  - Displays the correlation between payload mass and mission outcome
  - It visually depicted successes and failliures

Predictive Analysis (Classification)

To perform predictive analysis the steps is as follows:

- Perform exploratory Data Analysis and determine Training Labels
- create a column for the class
- Standardize the data
- Split into training data and test data
-Find best Hyperparameter for
    1. Support Vector Machine (SVM)
    2. Decision Classification Trees
    3. Logistic Regression
    4. K Nearest Neighbor(KNN)
- Find the method performs best using test data
- Illustrate the best results using accuracy tests and confusion matrix.

Results

The sections to follow illustrate the results in the order:

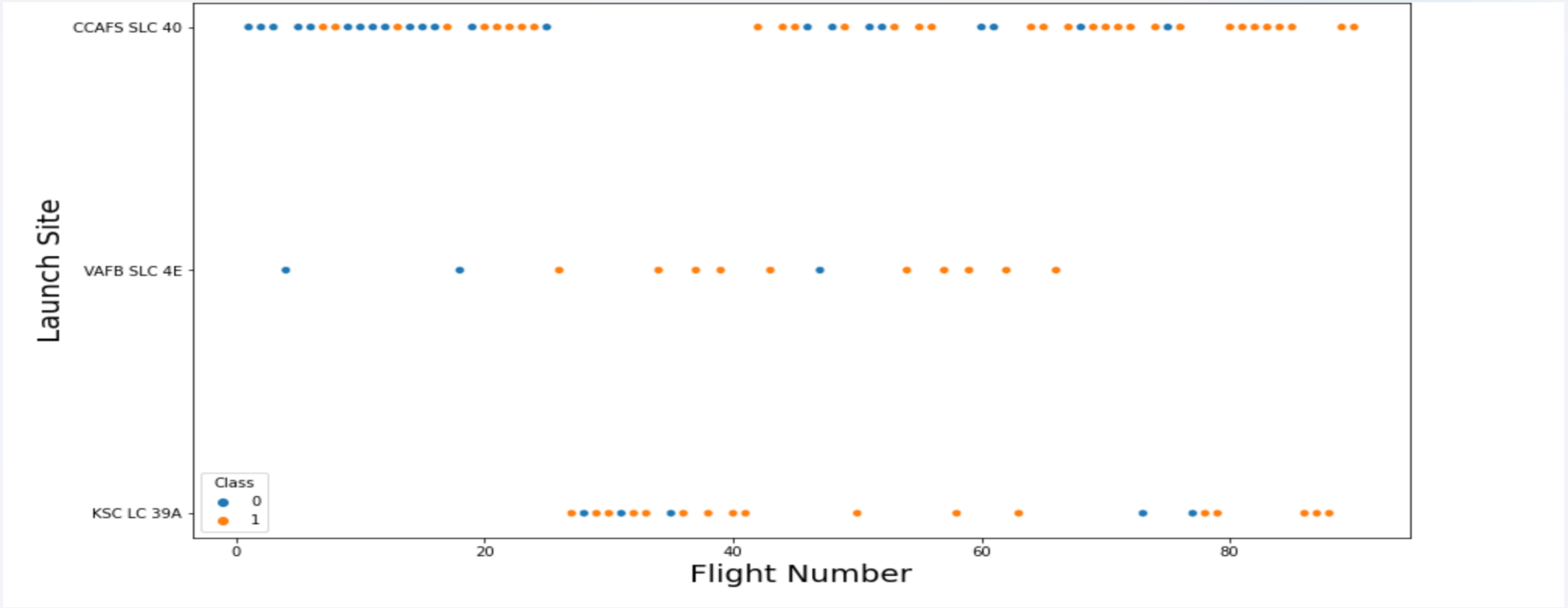
- Exploratory Data Analysis Using SQL
- Launch Sites and Proximity Analysis
- Folium
- Ploty Dash
- Predictive Analysis

Section 2

Insights drawn from EDA

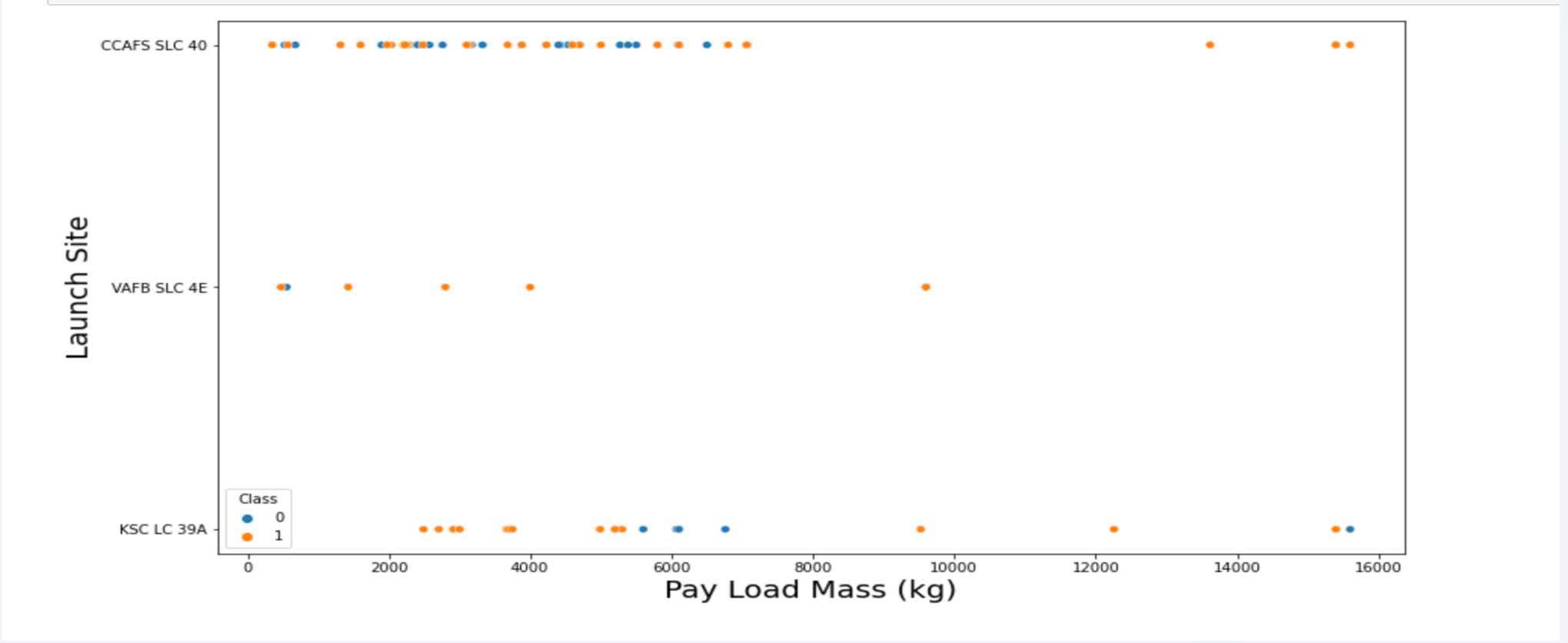
Results: Data Visualization (Matplotlib & Seaborn)

- Flight Number vs. Launch Site



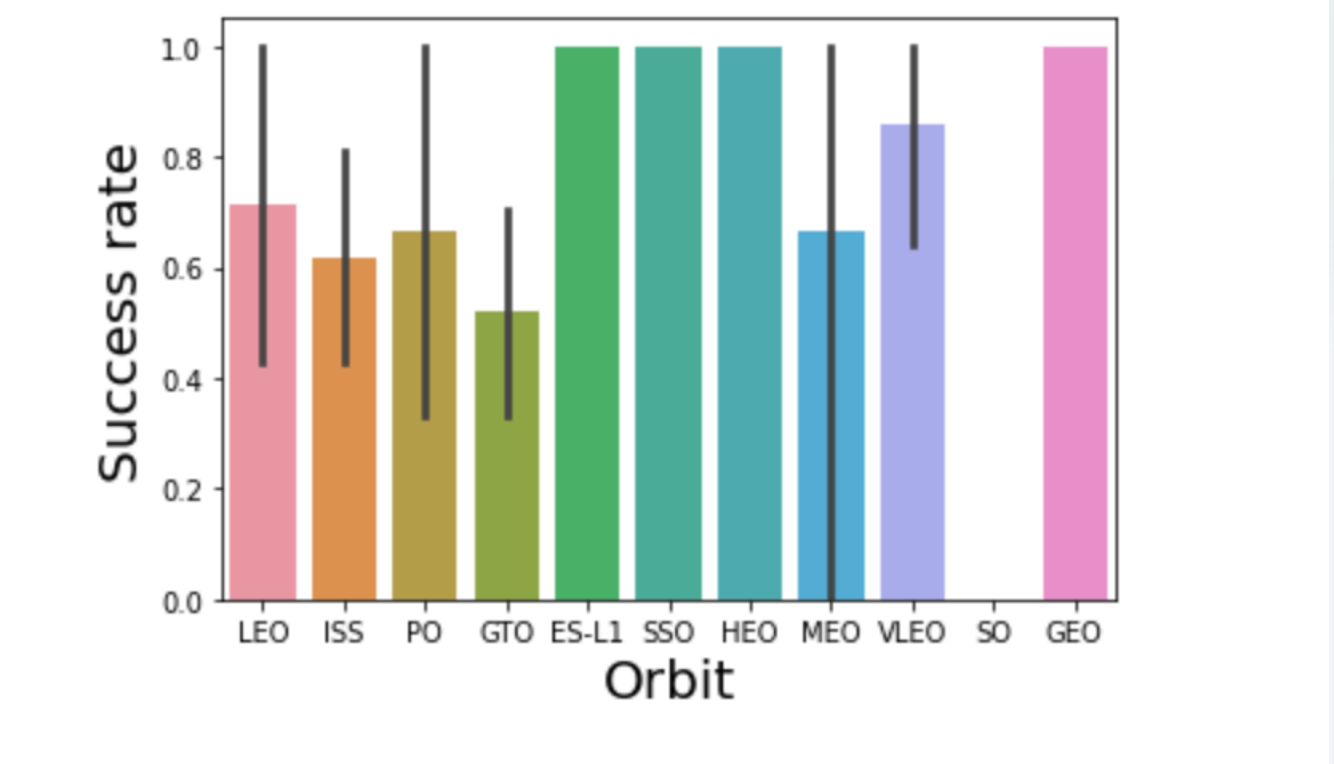
Results: Data Visualization (Matplotlib & Seaborn)

- Payload vs. Launch Site - for the VAFB-SLC launchsite there are no rockets launched for heavy pay load mass(greater than 10000).



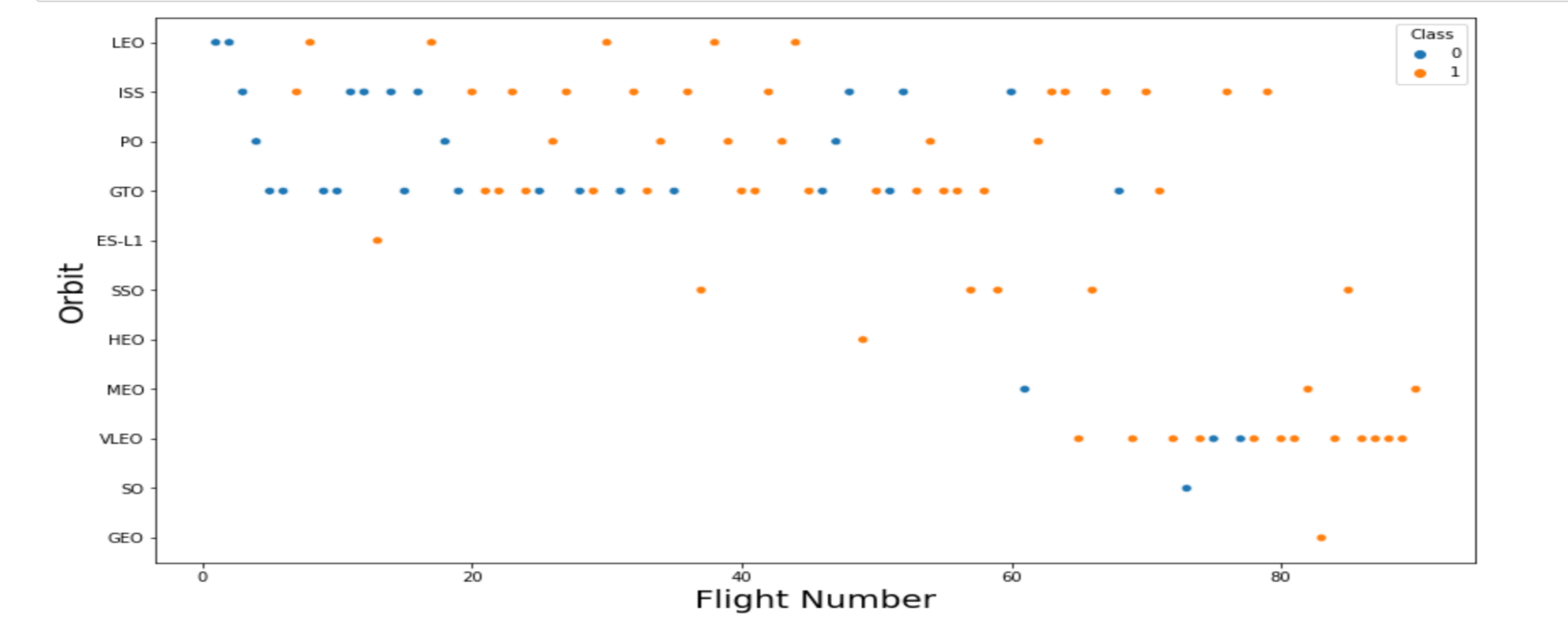
Results: Data Visualization (Matplotlib & Seaborn)

- Success Rate vs. Orbit Type



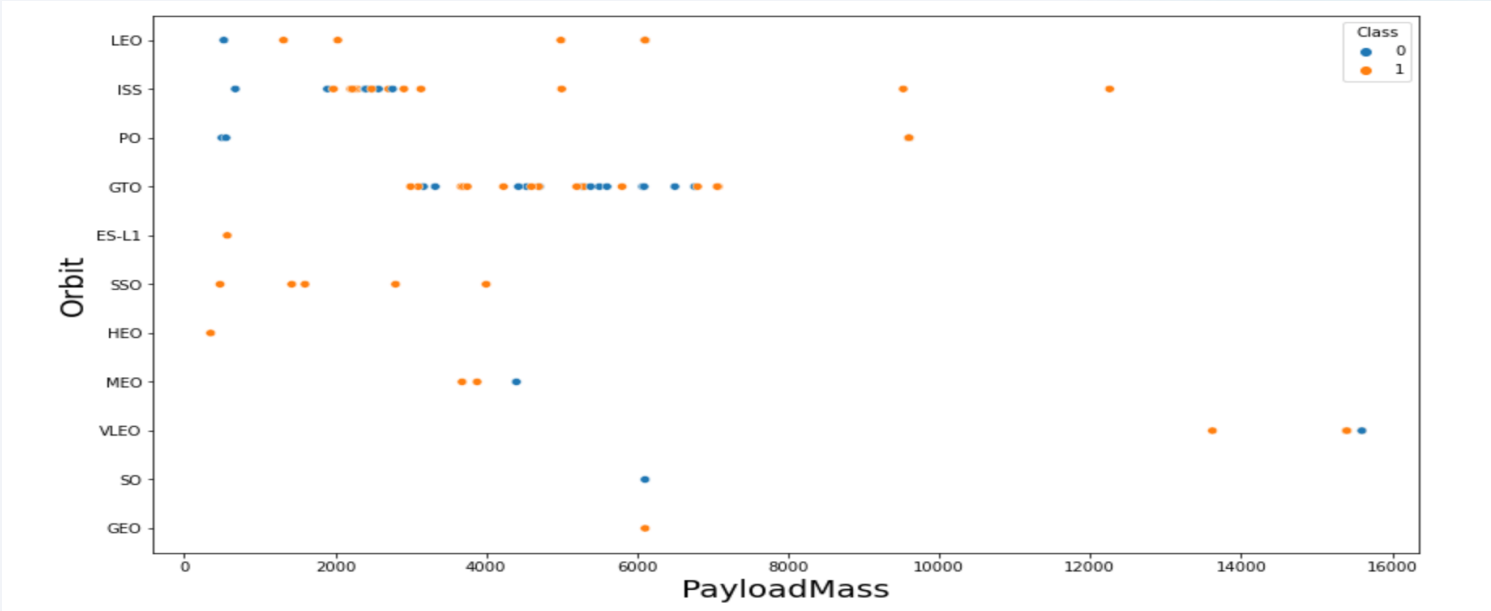
Results: Data Visualization (Matplotlib & Seaborn)

- Flight Number vs. Orbit Type - the success of the LEO orbit appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

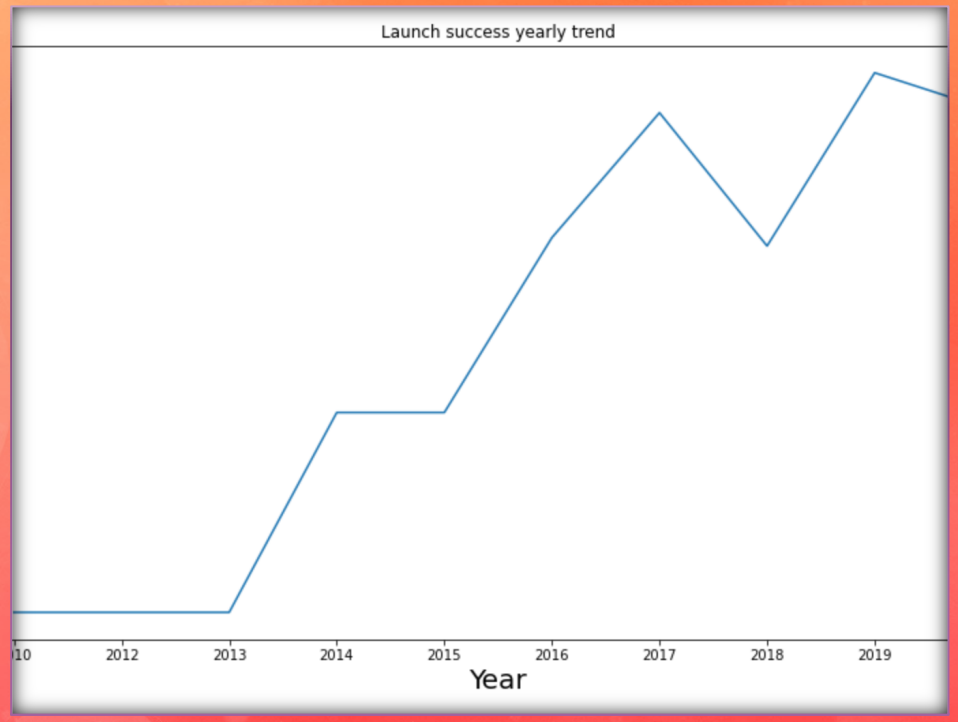


Results: Data Visualization (Matplotlib & Seaborn)

- 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.



Results: Data Visualization (Matplotlib & Seaborn)



- Launch Success Yearly Trend - you can observe that the success rate since 2013 kept increasing till 2020

Results: EDA with SQL

- All Launch Site Names:

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

- Launch Site Names Begin with 'CCA'

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Results: EDA with SQL

- Total Payload Mass:

Total payload mass by NASA (CRS)
45596

- Average Payload Mass by F9 v1.1

Average payload mass by Booster Version F9 v1.1
2928

Results: EDA with SQL

- First Successful Ground Landing Date

Date of first successful landing outcome in ground pad
2015-12-22

- Successful Drone Ship Landing with Payload between 4000 and 6000:

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

Flight Number vs. Launch Site

- Total Number of Successful and Failure Mission Outcomes:

number_of_success_outcomes	number_of_failure_outcomes
100	1

- Boosters Carried Maximum Payload:

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

Flight Number vs. Launch Site

- 2015 Launch Records:

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

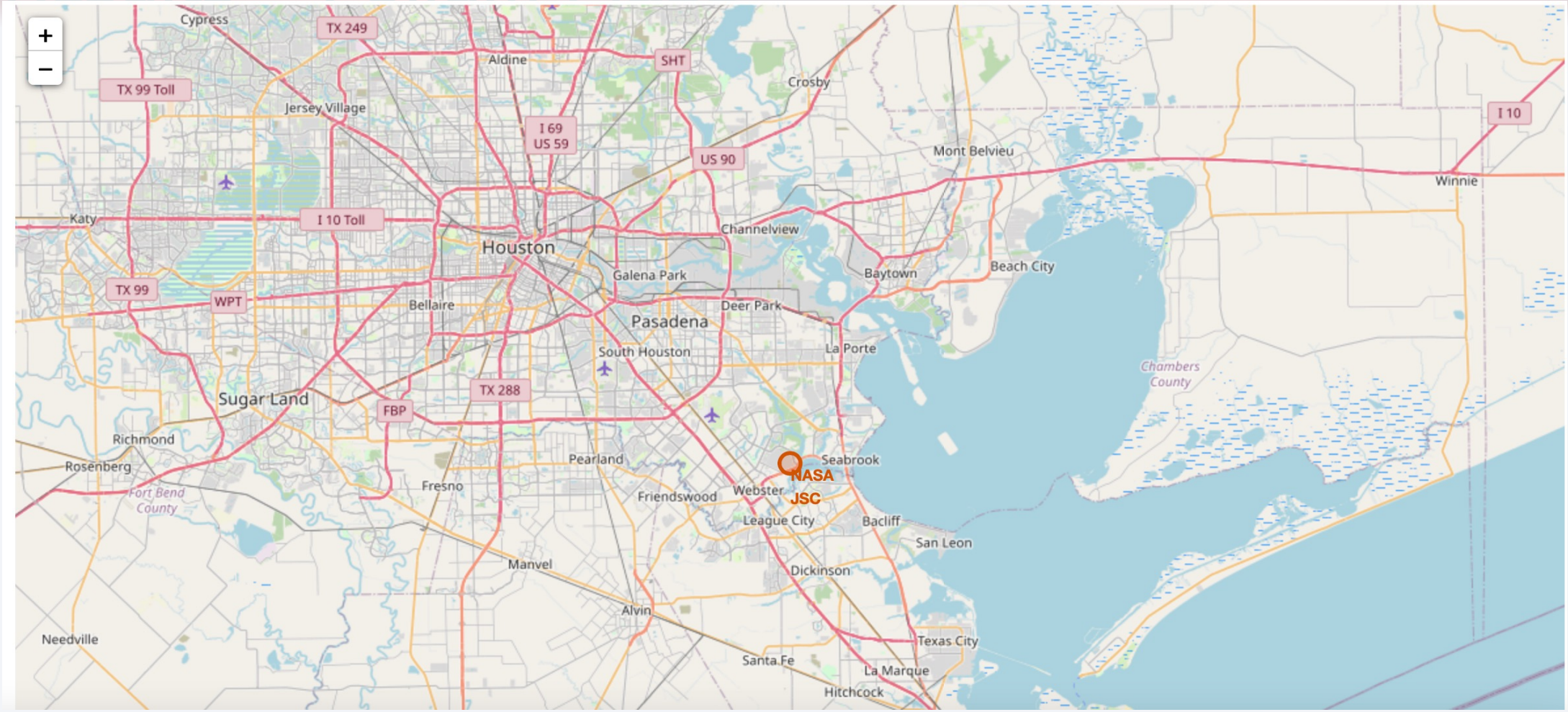
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

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

Launch Sites Proximities Analysis

SECTION 3

Results – Folium Maps



Results – Folium Maps

Results – Folium Maps

Results – Success/Failed Launches For Each Site Map

- Class “1” indicates a successful launch and Class “0” indicates a failed launch.

	Launch Site	Lat	Long	class
46	KSC LC-39A	28.573255	-80.646895	1
47	KSC LC-39A	28.573255	-80.646895	1
48	KSC LC-39A	28.573255	-80.646895	1
49	CCAFS SLC-40	28.563197	-80.576820	1
50	CCAFS SLC-40	28.563197	-80.576820	1
51	CCAFS SLC-40	28.563197	-80.576820	0
52	CCAFS SLC-40	28.563197	-80.576820	0
53	CCAFS SLC-40	28.563197	-80.576820	0
54	CCAFS SLC-40	28.563197	-80.576820	1
55	CCAFS SLC-40	28.563197	-80.576820	0

Section 4

Dashboarding with Ploty Dash

Ploty Dash

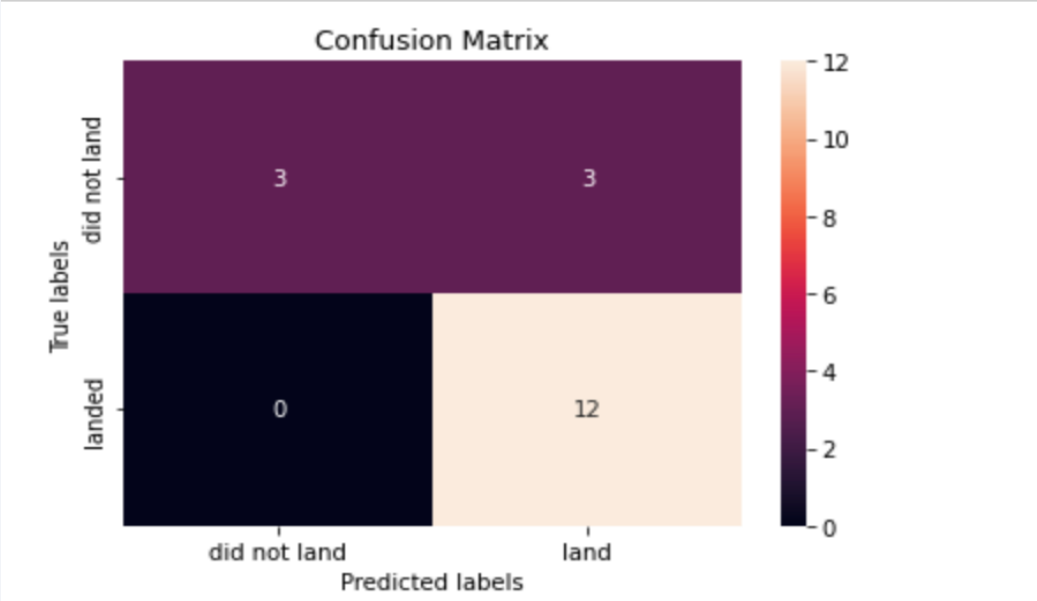
- While I am not able to provide images from Ploty Dash due to a Mac security measure, this visual tool permits the design of a dashboard that:
  - enable Launch Site selection
  - create pie chart to show the total successful launches count for all sites
  - Add slider to select payload range
  - Create a scatter graph depicting the relationship between payload and success for All Sites

Section 5

Predictive Analysis

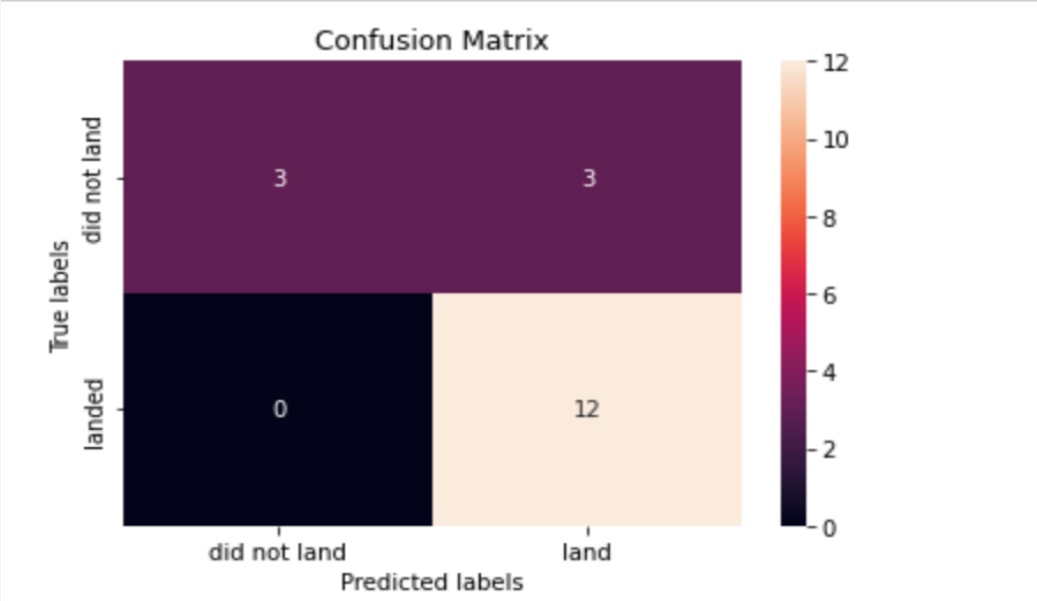
Results – Logistic Regression

- GridSearchCV Score: 0.8464285714285713
- Best Accuracy Score: 0.8333333333333334
- Confusion Matrix: Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



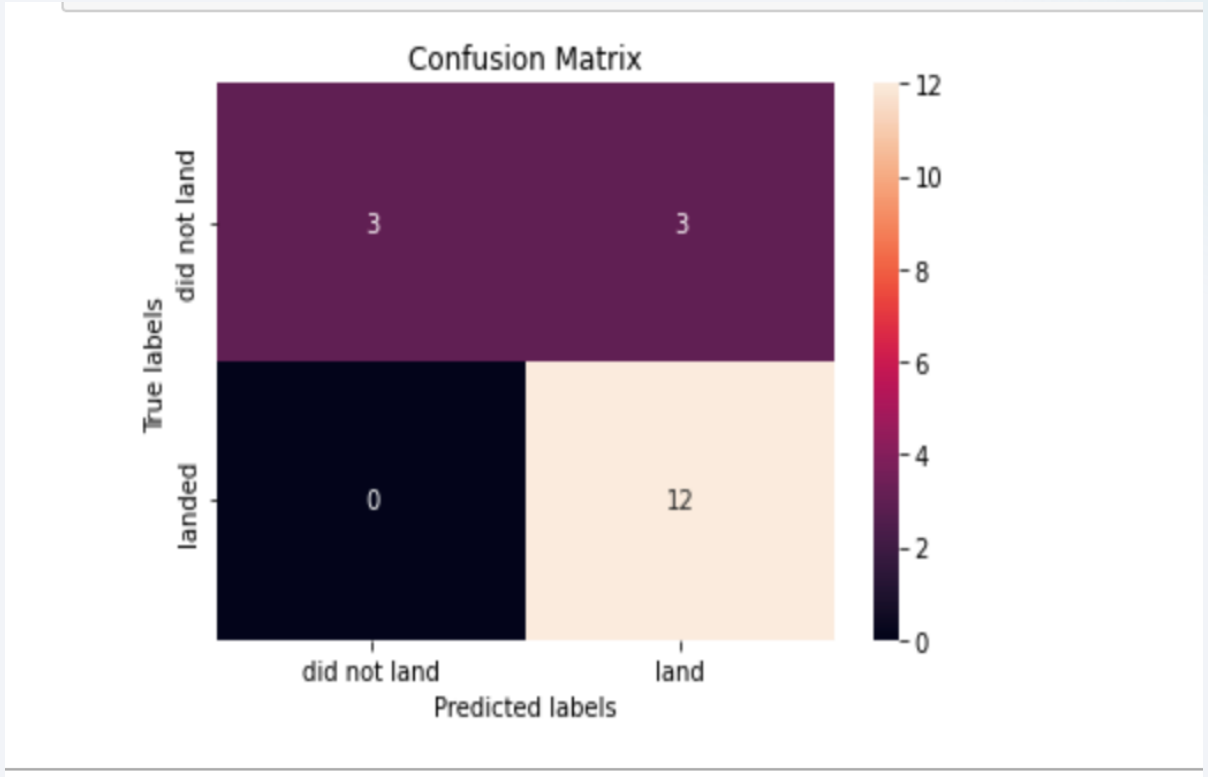
Results – Logistic Regression

- GridSearchCV Score: 0.8464285714285713
- Best Accuracy Score: 0.8333333333333334
- Confusion Matrix: Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



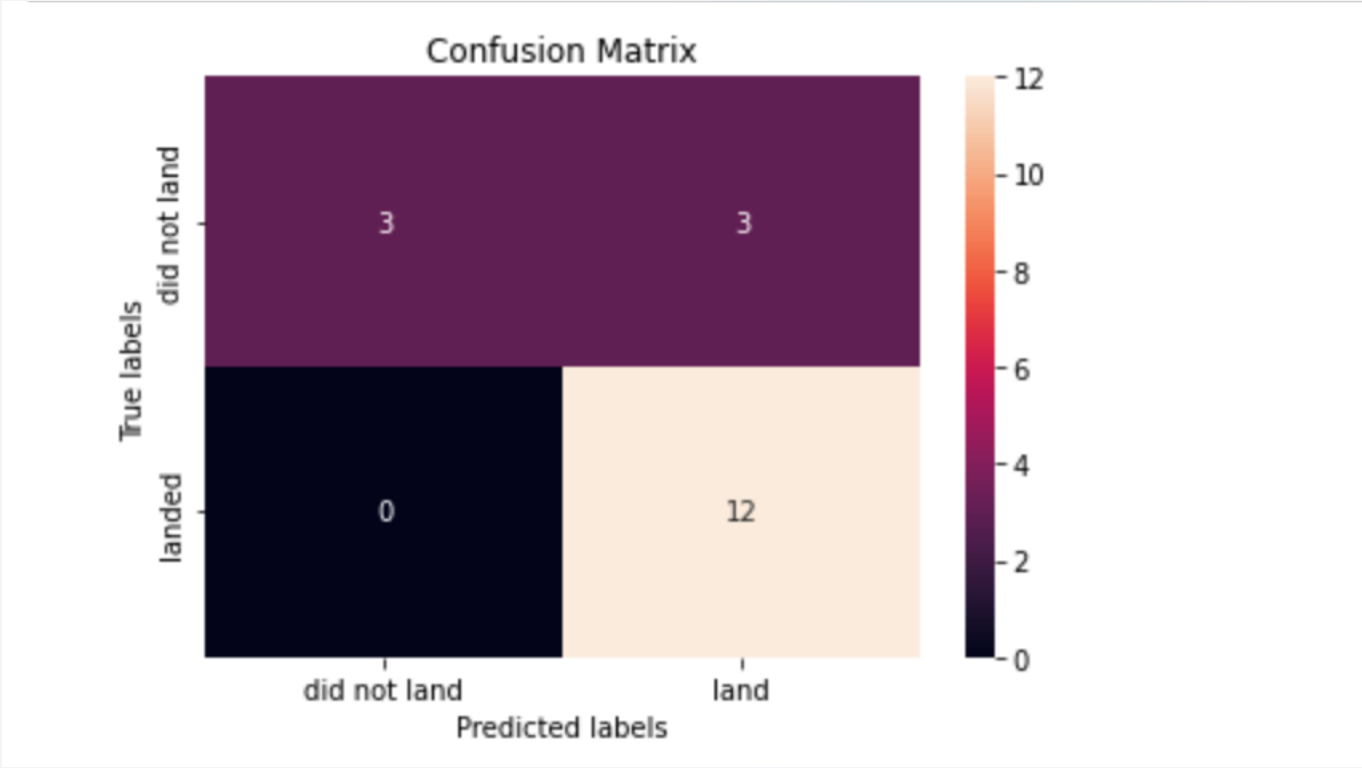
Results – Support Vector Machine(SVM)

- GridSearchCV Score: 0.8482142857142856
- Accuracy Score: 0.8333333333333334
- Confusion Matrix:



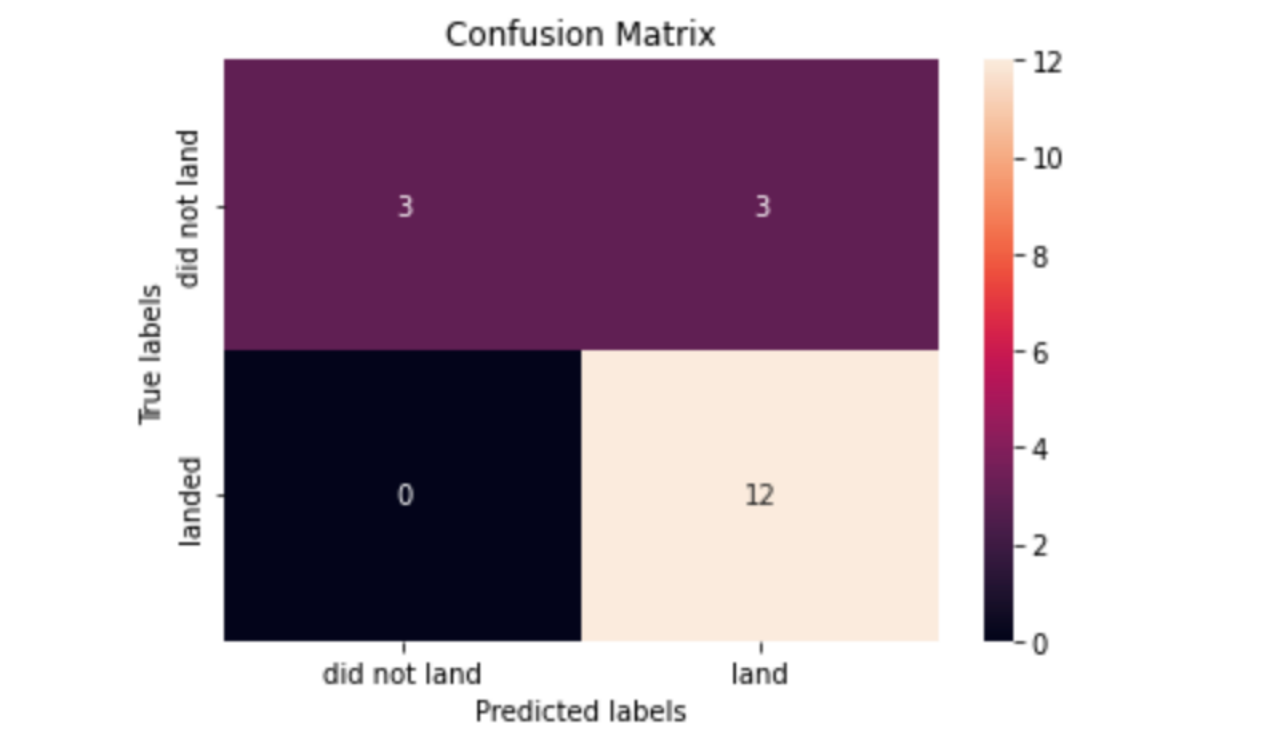
Results – Decision Tree Classifier

- GridSearchCV Score: 0.8767857142857144
- Accuracy Score: 0.8333333333333334
- Confusion Matrix:



Results – K Nearest Neighbor (KNN)

- GridSearchCV Score: 0.8482142857142858
- Accuracy Score: 0.8333333333333334
- Confusion Matrix:



Results – Predictive Analysis

- Considering all four models, it's clear to see that they all returned the same Accuracy Scores as well as Confusion Matrix.
- The alternative method of determining the most efficient model was to analyze the data by the best GridSearchCV scrore.
- Hence, we can conclude that the "Decision Tree" model is the most efficient for prediction

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.876786
KNN	0.848214

Conclusions

- Point 1 – The goal of this project was to determine the landing success rate for SpaceX Falcon9 at the first stage which would also impact the cost of launch.
- Point 2 – After analyzing the date via Python libraries Panda & Numpy, in addition to using visualization tools such as Matplotlib & Folium, it's evident that certain features of the launch may affect the success rate of the landing.
- Point 3 – The features analyzed included the Flight Number, Date of launch, Year, Orbit, Pay Load Mass, and more. All or some of these features are pertinent to achieving a success first stage landing rate.
- Point 4 – Lastly, the application of various Machine Learning algorithms, while performing predictive analysis, resulted in the conclusion that the Decision Tree model is the best algorithm to use because it had a high GridSearchCV score.

Thank you!