Space Race with Data Science

Awuah Godsway

1st January 2024

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Summary of methodologies
  - -SpaceX Data Collection using SpaceX API
  - -SpaceX Data Collection with Web Scraping
  - -SpaceX Data Wrangling
  - -SpaceX Exploratory Data Analysis using SQL
  - -Space-X EDA DataViz Using Python Pandas and Matplotlib
  - -Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and Ploty Dash
  - -SpaceX Machine Learning Landing Prediction
- Summary of all results
  - -EDA results
  - -Interactive Visual Analytics and Dashboards
  - -Predictive Analysis(Classification)

INTRODUCTION

- **Project background and context**

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems you want to find answers**

In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website

METHODOLOGY

Executive Summary

- Data collection methodology:
  - Describes how data sets were collected
- Perform data wrangling
  - Describes how data were processed
- 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, and evaluate classification models

DATA COLLECTION

❑ Description of how SpaceX Falcon9 data was collected.

•Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series of helper functions that would help use the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API URL.

•Finally to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a JSON result which was then converted into a Pandas data frame.

•Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches the launch records are stored in HTML. Using BeautifulSoupand request Libraries, I extracted the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table, and converted it into a Pandas data frame

DATA WRANGLING

- After obtaining and creating a Pandas DF from the collected data, data was filtered using the **BoosterVersion** column to only keep the Falcon 9 launches, then dealt with the missing data values in the **LandingPadand PayloadMass** columns. For the **PayloadMass**, missing data values were replaced using a mean value of the column.
- Also performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

[23]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

[23]:
```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

[26]:
```
landing_class=df['Class']
df[['Class']].head(8)
```

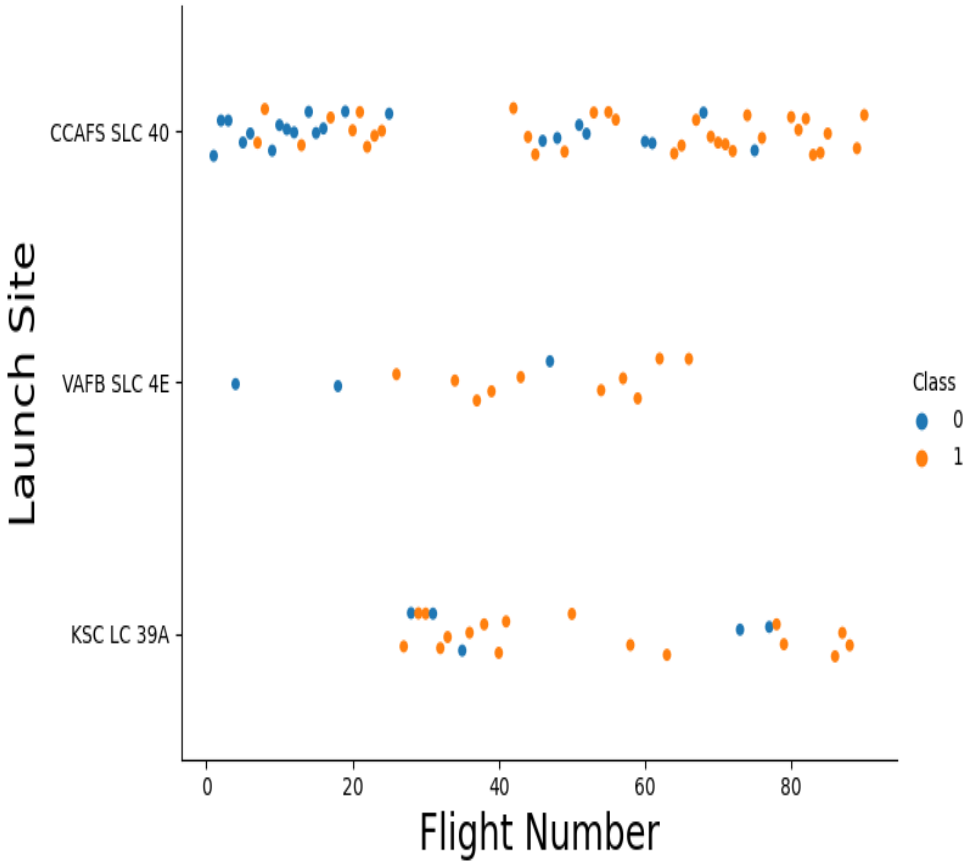
[26]:

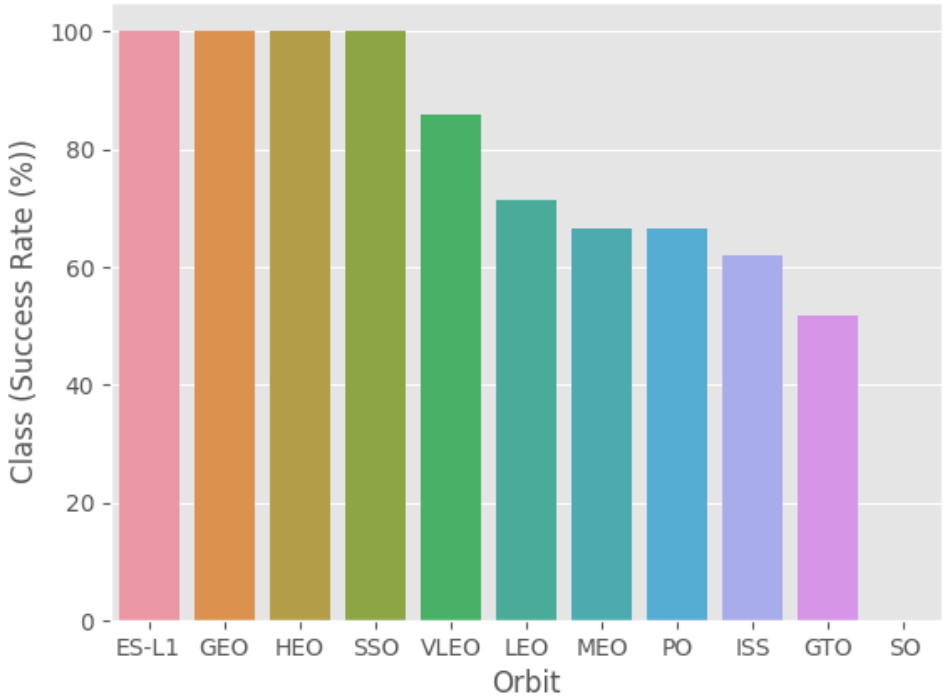
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA WITH DATA VISUALIZATION

- Performed data Analysis and Feature Engineering using Pandas and Matplotlib. i.e.
  - Exploratory Data Analysis
  - Preparing Data Feature Engineering
- Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumberand Orbit type, Payload and Orbit type.
- Used a Bar chart to Visualize the relationship between the success rate of each orbit type.
- Line plot to Visualize the launch success yearly trend.

EDA WITH DATA VISUALIZATION





EDA WITH SQL

- The following SQL queries were performed for EDA

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

Display 5 records where launch sites begin with the string 'CCA'

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

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Versio
```

EDA WITH SQL (CONT…)

- List the date when the first successful landing outcome in ground pad was achieved.

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

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 DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

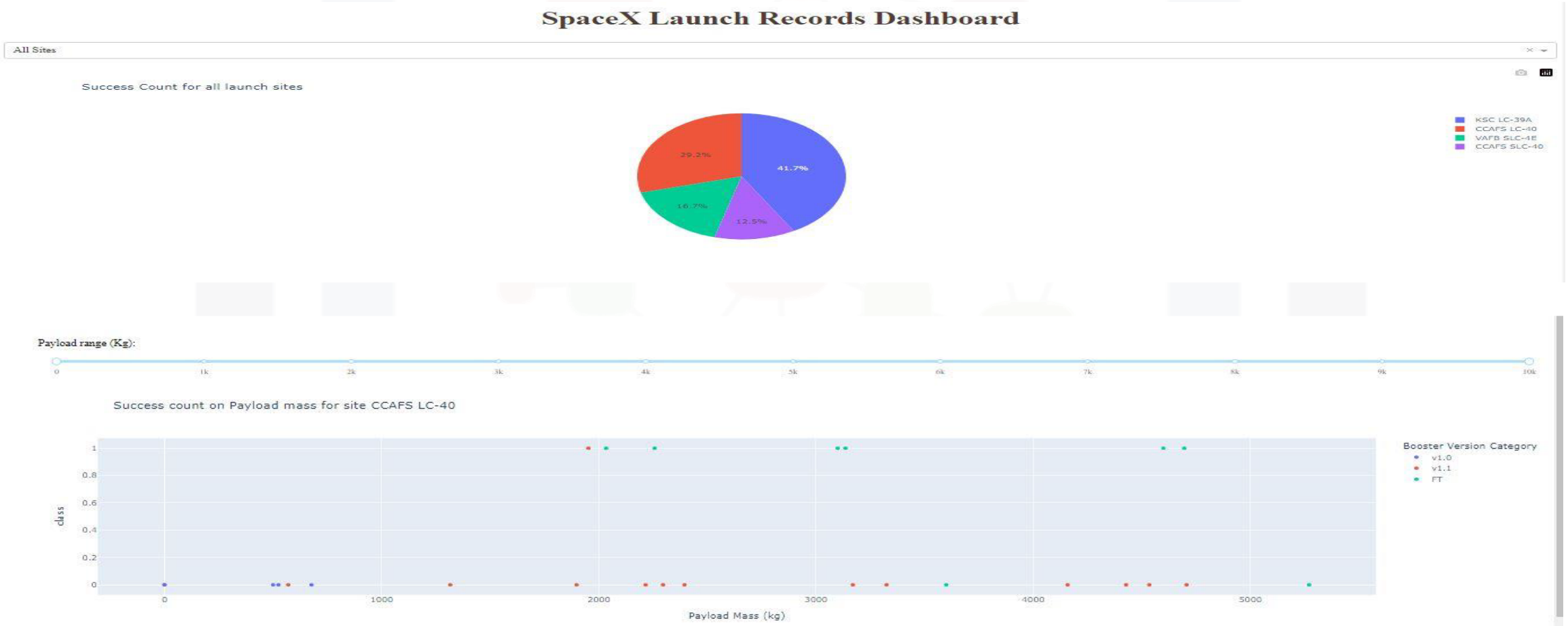
BUILD AN INTERACTIVE MAP WITH FOLIUM

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Created a launch set outcomes (failure=0 or success=1).

BUILD A DASHBOARD WITH PLOTLY DASH

- Built an interactive dashboard application with Plotlydash by:
  - Adding a Launch Site Drop-down Input Component.
  - Adding a callback function to render success-pie-chart based on selected site dropdown.
  - Adding a Range Slider to Select Payload.
  - Add a callback function to render the success-payload-scatter-chart scatter plot.

SpaceX Dash App

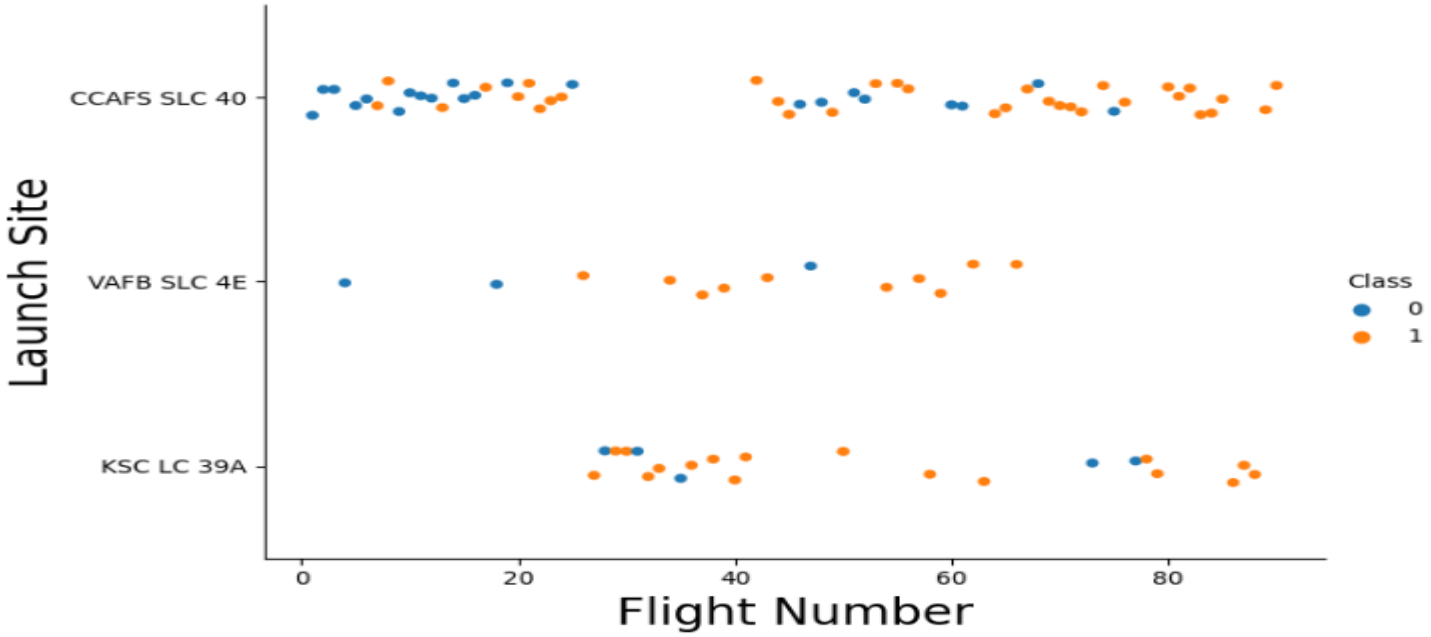


RESULTS

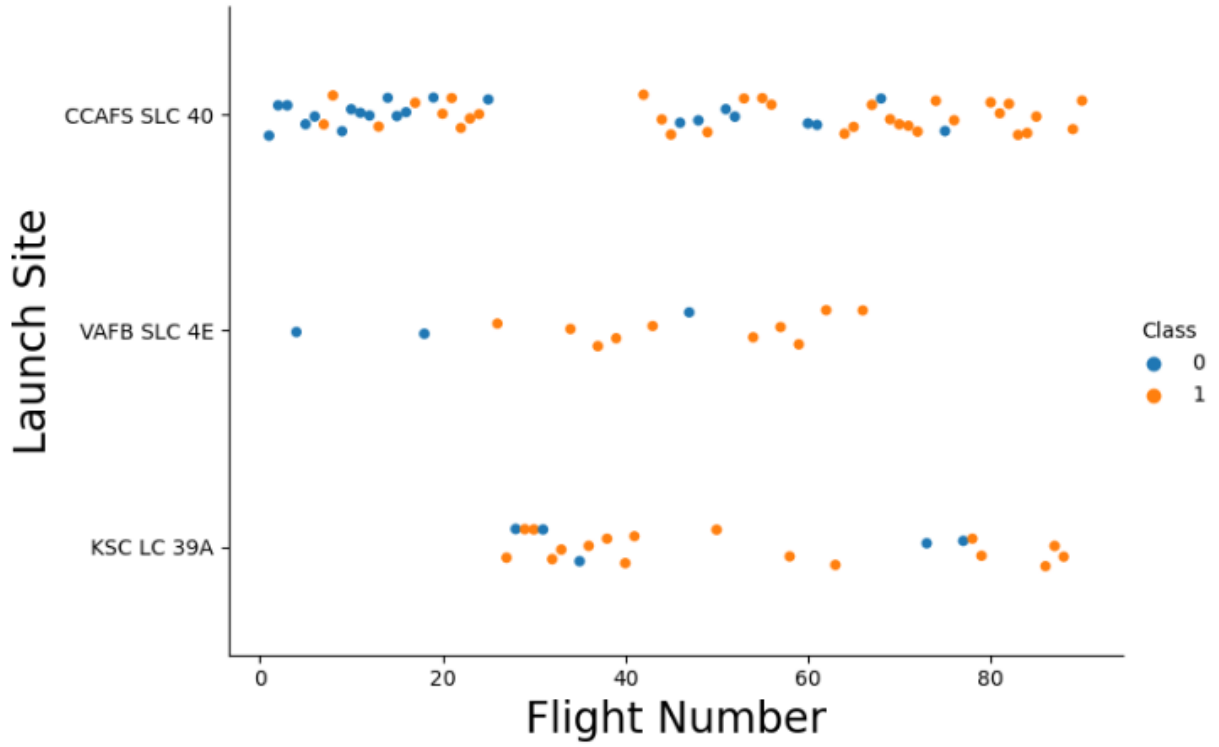
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Flight Number vs Launch Site

- A Scatter plot of Flight Number vs. Launch Site



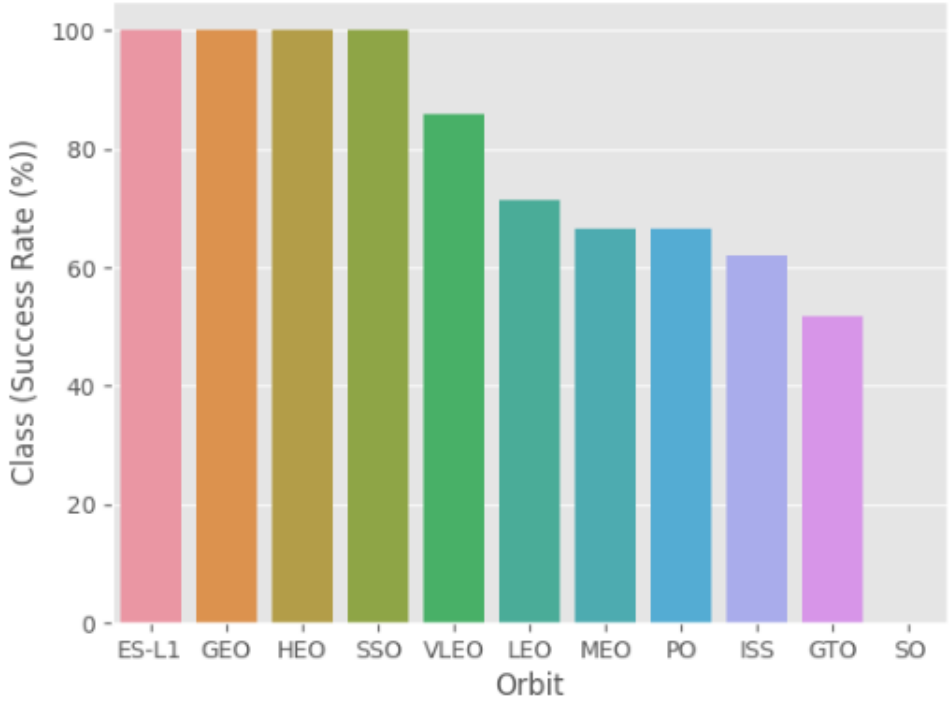
FLIGHT NUMBER VS. LAUNCH SITE WITH EXPLANATIONS



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

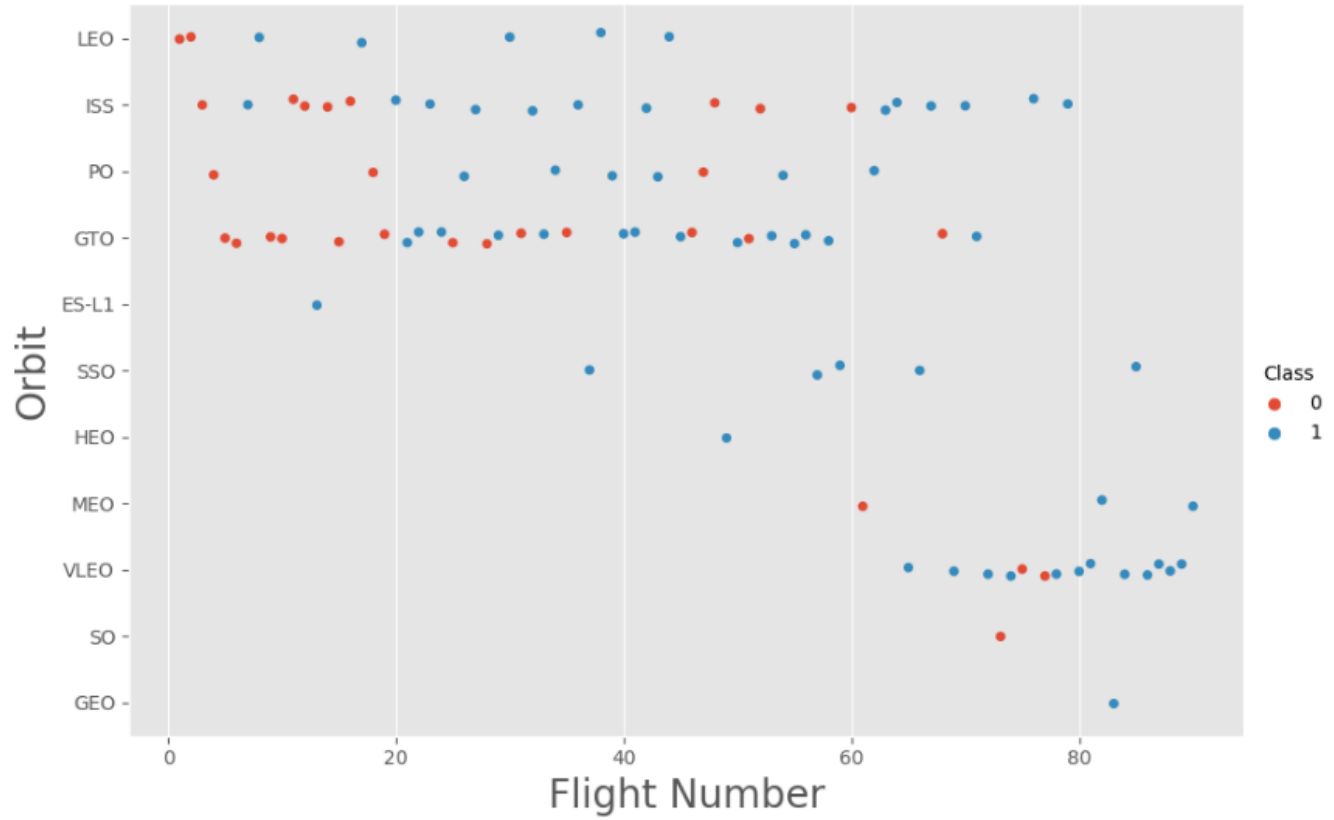
SUCCESS RATE VS. ORBIT TYPE WITH EXPLANATIONS



Analyze the ploted bar chart try to find which orbits have high sucess rate.

Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

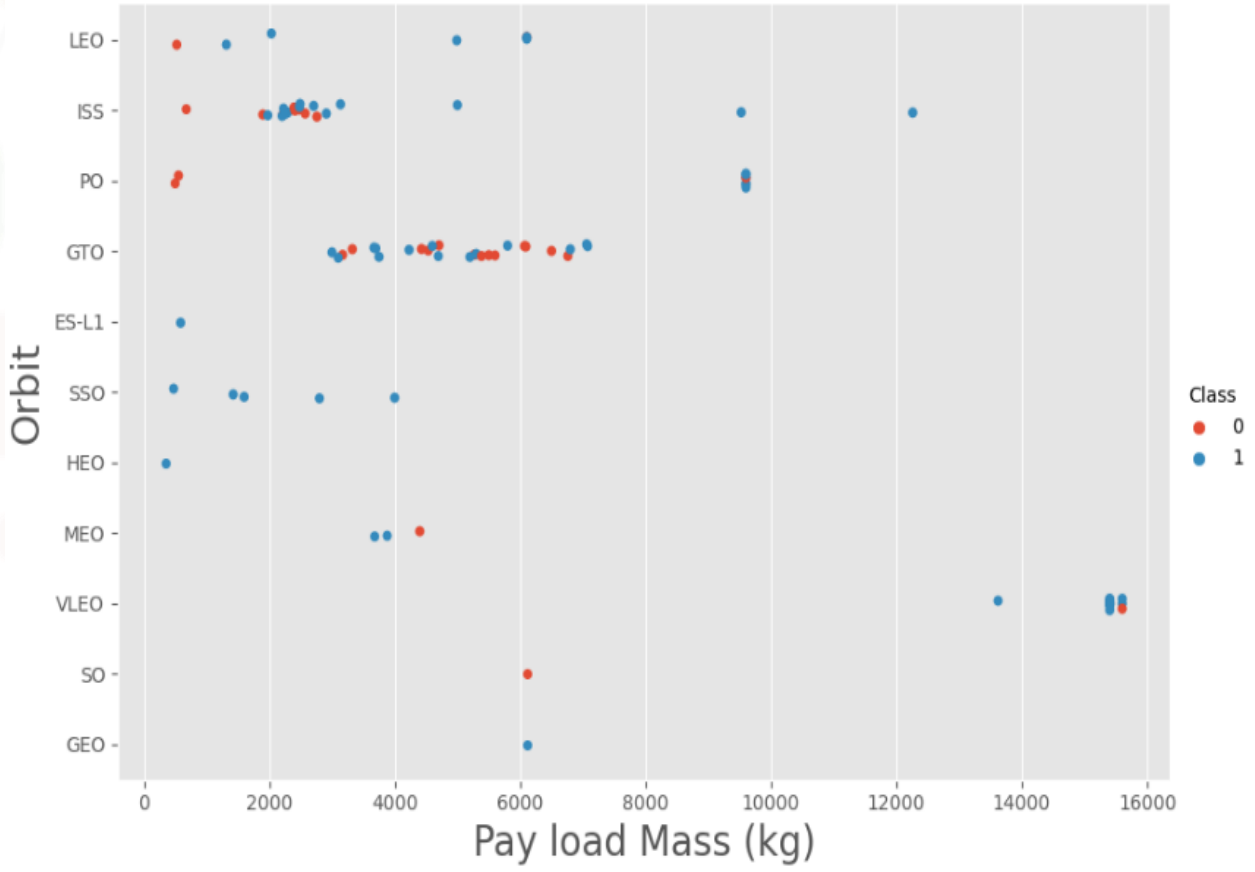
FLIGHT NUMBER VS. ORBIT TYPE WITH EXPLANATIONS



You should 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 is higher for Polar, LEO, and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) have near equal chances.



PREDICTIVE ANALYSIS (CLASSIFICATION)

- ❑ Summary of how I built, evaluated, improved, and found the best-performing classification model
- ❑ After loading the data as a Pandas data frame, I set out to perform exploratory Data Analysis and determine Training Labels by;
- Creating a NumPy array from the column Class in data, by applying the method to_numpy() and then assigning it to the variable Y as the outcome variable.
- Then standardized the feature dataset (x) by transforming it using preprocessing.StandardScaler() function from Sklearn.
- After which the data was split into training and testing sets using the function train_test_split from sklearn.model_selection with the test_size parameter set to 0.2 and random_state to 2.

PREDICTIVE ANALYSIS (CLASSIFICATION)

- To find the best ML model/ method that would perform best using the test data between SVM, Classification Trees, k nearest neighbors, and Logistic Regression;

•First created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.

•For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find the best Hyperparameter.

•After fitting the training set, we output the GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_.

•Finally use the method score to calculate the accuracy of the test data for each model and plot a confusion matrix for each using the test and predicted outcomes.

PREDICTIVE ANALYSIS (CLASSIFICATION)

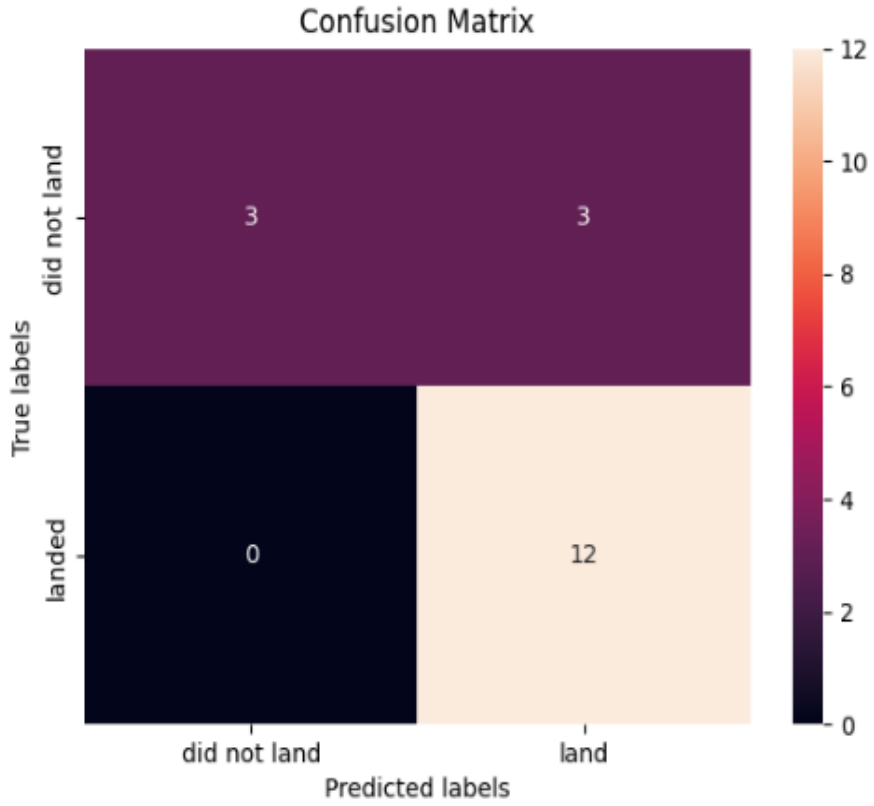
- The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbors, and Logistic Regression;

Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

CONFUSION MATRIX

- All 4 classification models had the same confusion matrixes and were able to equally distinguish between the different classes. The major problem is false positives for all the models.



CONCLUSIONS

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%.
- We can deduce that, as the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have 100% success rates after the 80th flight
- If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has a 0% success rate.
- LEO orbit 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

CONCLUSIONS

- With heavy payloads the successful landing or positive landing rate is higher 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
- And finally the success rate since 2013 kept increasing till 2020.

THANK YOU !!!!