

Winning Space Race with Data Science

Ourad Abdessamad
01/03/2023



Outline

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

Executive Summary

- Summary of methodologies
 - Data Collection via API, Web Scraping
 - Exploratory Data Analysis (EDA) with Data Visualization
 - EDA with SQL
 - Interactive Map with Folium
 - Dashboards with Plotly Dash
 - Predictive Analysis
- Summary of all results
 - Exploratory Data Analysis results
 - Interactive maps and dashboard
 - Predictive results

Introduction

- Project background and context
 - Predicting whether the Falcon 9 first stage will land successfully is the goal of this endeavor. The launch of the Falcon 9 rocket, according to SpaceX, cost 62 million dollars. The price of other services is around 165 million dollars per provider. The fact that SpaceX can reuse the first stage explains the pricing disparity. The price of a launch can be calculated by assessing if the stage will land. If another business plans to compete with SpaceX for a rocket launch, this information may be useful.
- Problems you want to find answers
 - What distinguishes a successful landing from a failed one?
 - What impact does each relationship between the rocket factors have on whether a landing is successful or unsuccessful?
 - Which circumstances will enable SpaceX to reach the highest landing success rate?

Section 1

Methodology

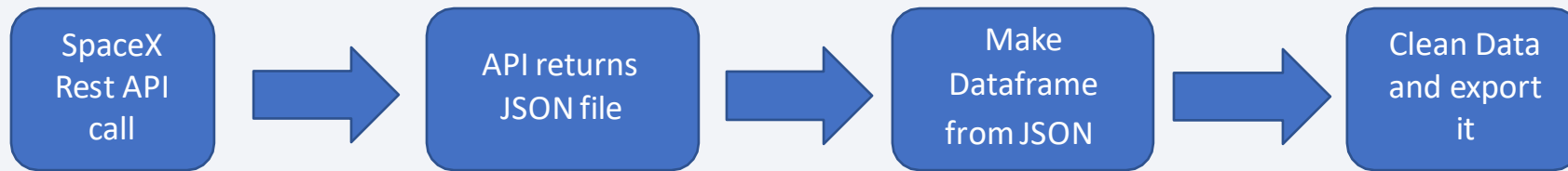
Methodology

Executive Summary

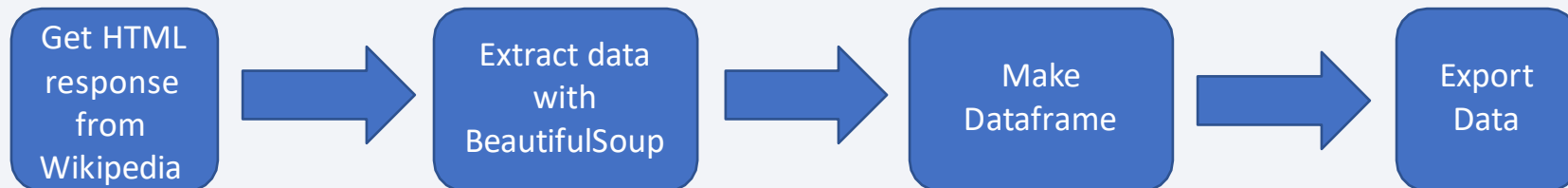
- Data collection methodology:
 - SpaceX REST API
 - Web Scrapping from Wikipedia
- Perform data wrangling
 - Dropping unnecessary columns
 - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
 - The information obtained by the API are rocket, launches, payload information.
 - The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
 - URL is [https://en.wikipedia.org/w/index.php?title=List of Falcon 9 and Falcon Heavy launches&oldid=1027686922](https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)



Data Collection - SpaceX API

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"  
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

```
data = response.json()  
data = pd.json_normalize(data)
```

3. Transform data

```
getLaunchSite(data)  
getPayloadData(data)  
getCoreData(data)  
getBoosterVersion(data)
```

4. Create dictionary with data

```
launch_dict = {'FlightNumber': list(data['flight_number']),  
               'Date': list(data['date']),  
               'BoosterVersion':BoosterVersion,  
               'PayloadMass':PayloadMass,  
               'Orbit':Orbit,  
               'LaunchSite':LaunchSite,  
               'Outcome':Outcome,  
               'Flights':Flights,  
               'GridFins':GridFins,  
               'Reused':Reused,  
               'Legs':Legs,  
               'LandingPad':LandingPad,  
               'Block':Block,  
               'ReusedCount':ReusedCount,  
               'Serial':Serial,  
               'Longitude': Longitude,  
               'Latitude': Latitude}
```

5. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

[Link to code](#)

Data Collection - Scraping

1. Getting Response from HTML

```
response = requests.get(static_url)
```

2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables

```
html_tables = soup.findAll('table')
```

4. Get column names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

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

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

6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all(
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

See notebook for the rest of code

7. Create dataframe from dictionary

```
df=pd.DataFrame(launch_dict)
```

8. Export to file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

[Link to code](#)

Data Wrangling

- In the dataset, there are several cases where the booster did not land successfully.
 - True Ocean, True RTLS, True ASDS means the mission has been successful.
 - False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.

1. Calculate launches number for each site

```
df['LaunchSite'].value_counts()
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from Outcome column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5. Export to file

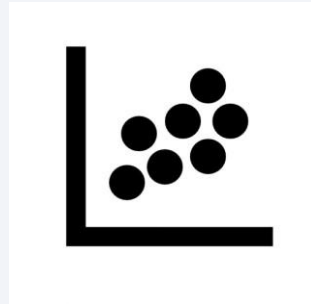
```
df.to_csv("dataset_part_2.csv", index=False)
```

[Link to code](#)

EDA with Data Visualization

- Scatter Graphs

- 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 relationship between variables. This relationship is called the correlation.

- Bar Graph

- Success rate vs. Orbit

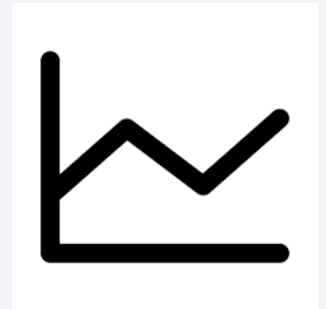
Bar graphs show the relationship between numeric and categoric variables.



- Line Graph

- Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.



EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
 - Displaying 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 date 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 4000 but less than 6000.
 - 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.

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas
 - Red circle at NASA Johnson Space Center's coordinate with label showing its name (*folium.Circle*, *folium.map.Marker*).
 - Red circles at each launch site coordinates with label showing launch site name (*folium.Circle*, *folium.map.Marker*, *folium.features.DivIcon*).
 - The grouping of points in a cluster to display multiple and different information for the same coordinates (*folium.plugins.MarkerCluster*).
 - Markers to show successful and unsuccessful landings. **Green** for successful landing and **Red** for unsuccessful landing. (*folium.map.Marker*, *folium.Icon*).
 - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (*folium.map.Marker*, *folium.PolyLine*, *folium.features.DivIcon*)
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
 - Dropdown allows a user to choose the launch site or all launch sites (*dash_core_components.Dropdown*).
 - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (*plotly.express.pie*).
 - Rangeslider allows a user to select a payload mass in a fixed range (*dash_core_components.RangeSlider*).
 - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (*plotly.express.scatter*).

[Link to code](#)

Predictive Analysis (Classification)

- Data preparation
 - Load dataset
 - Normalize data
 - Split data into training and test sets.
- Model preparation
 - Selection of machine learning algorithms
 - Set parameters for each algorithm to GridSearchCV
 - Training GridSearchModel models with training dataset
- Model evaluation
 - Get best hyperparameters for each type of model
 - Compute accuracy for each model with test dataset
 - Plot Confusion Matrix
- Model comparison
 - Comparison of models according to their accuracy
 - The model with the best accuracy will be chosen (see Notebook for result)

Results

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

The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks are layered over a faint, dark grid pattern, creating a sense of depth and movement.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



We observe that, for each site, the success rate is increasing.

Payload vs. Launch Site



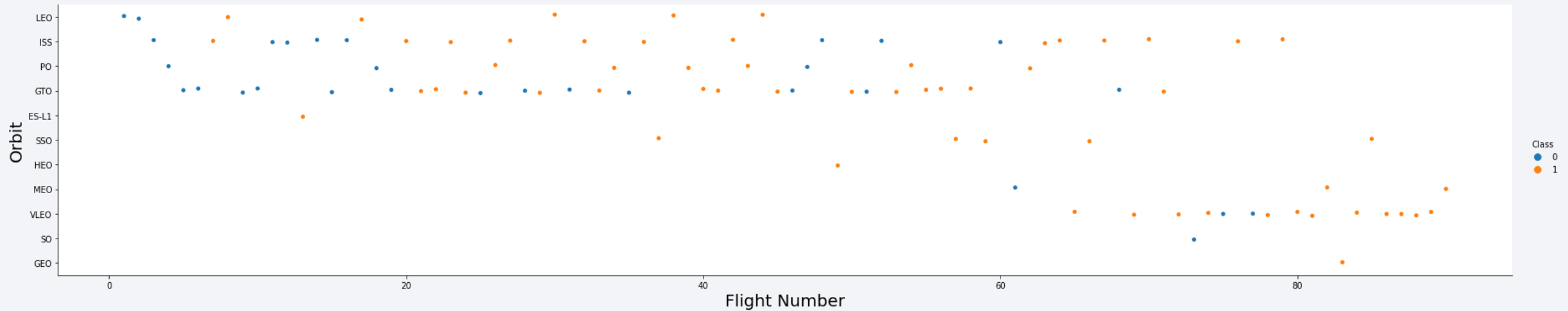
A bigger payload may be taken into account for a successful landing depending on the launch site. On the other hand, a landing may not succeed if the weight is too heavy.

Success Rate vs. Orbit Type



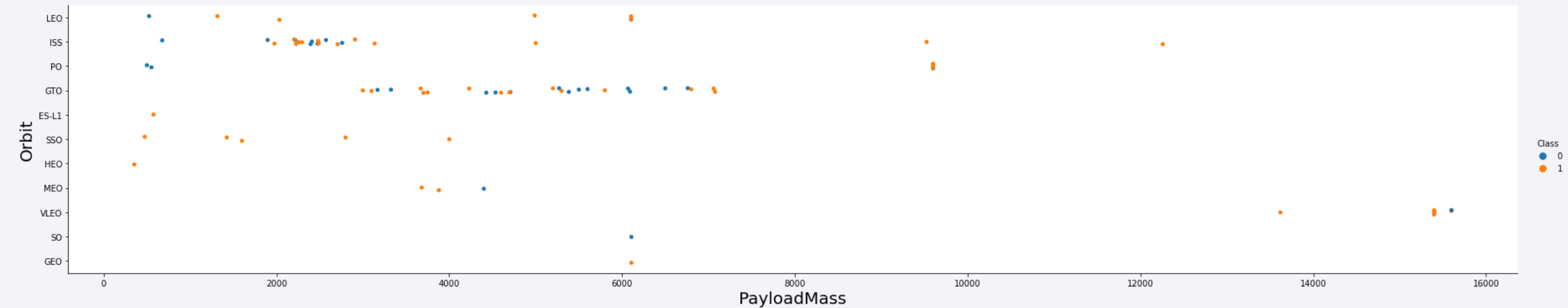
With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



For the LEO orbit, we see that the success rate rises as more trips are made. There is no correlation between the number of flights and the success rate for some orbits, such as GTO. However we might speculate that some orbits, like SSO or HEO, have a high success rate because of the lessons discovered through earlier launches for other orbits.

Payload vs. Orbit Type



The success rate of launches in some orbits can be significantly impacted by the payload weight. For the LEO orbit, bigger payloads, for instance, increase success rates. Another discovery is that reducing the payload weight for a GTO orbit increases launch success.

Launch Success Yearly Trend



Since 2013, we can see an increase in the Space X Rocket success rate.

All Launch Site Names

SQL Query

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Results

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

Explanation

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SQL Query

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

Explanation

The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

Results

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

SQL Query

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

Explanation

This query returns the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL Query

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL Query

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results

MIN("DATE")

01-05-2017

Explanation

With this query, we select the oldest successful landing.

The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

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 "PAYLOAD_MASS_KG_" > 4000 AND "PAYLOAD_MASS_KG_" < 6000;
```

Results

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

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Results

SUCCESS	FAILURE
100	1

Explanation

With the first `SELECT`, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The `WHERE` clause followed by `LIKE` clause filters mission outcome. The `COUNT` function counts records filtered.

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS_KG_" = (SELECT max("PAYLOAD_MASS_KG_") FROM SPACEXTBL)
```

Results

Booster_Version

F9 B5 B1048.4

F9 B5 B1049.4

F9 B5 B1051.3

F9 B5 B1056.4

F9 B5 B1048.5

F9 B5 B1051.4

F9 B5 B1049.5

F9 B5 B1060.2

F9 B5 B1058.3

F9 B5 B1051.6

F9 B5 B1060.3

F9 B5 B1049.7

Explanation

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

2015 Launch Records

SQL Query

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Results

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7,4) shows year.

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

SQL Query

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Results

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Explanation

This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky and a view of the Earth's surface, which is covered in a dense network of city lights and clouds. The lights are concentrated in the lower right portion of the image, while the upper left portion shows a clear blue sky.

Section 4

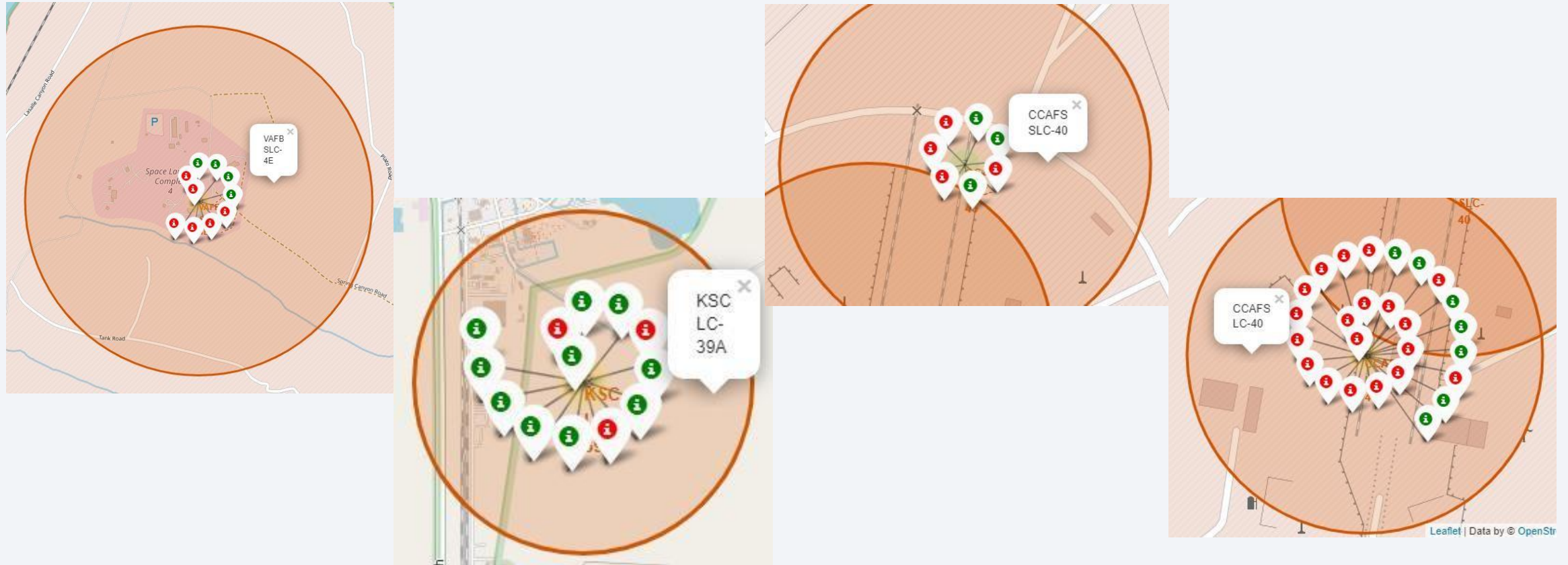
Launch Sites Proximities Analysis

Folium map - Ground stations



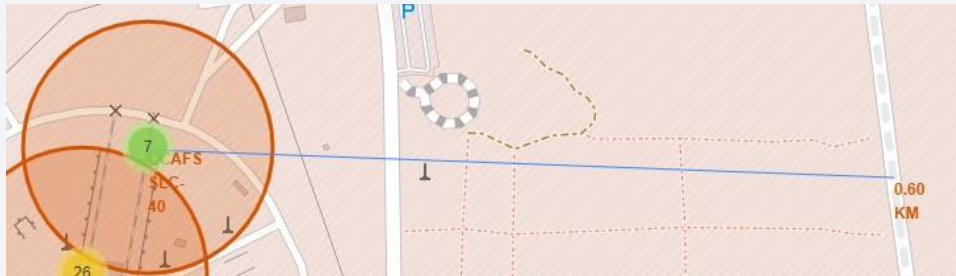
We see that Space X launch sites are located on the coast of the United States

Folium map - Color Labeled Markers



Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

Folium Map - Distances between CCAFS SLC-40 and its proximities



- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40 in close proximity to highways ? Yes
- Is CCAFS SLC-40 in close proximity to coastline ? Yes
- Do CCAFS SLC-40 keeps certain distance away from cities ? No



Section 5

Build a Dashboard with Plotly Dash

Dashboard - Total success by Site

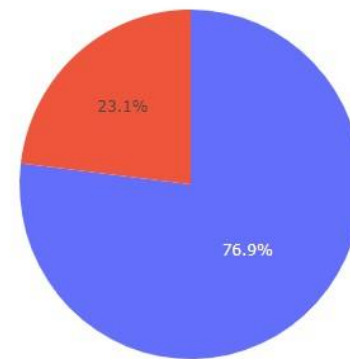
Total Success Launches by Site



We see that KSC LC-39A has the best success rate of launches.

Dashboard - Total success launches for Site KSC LC-39A

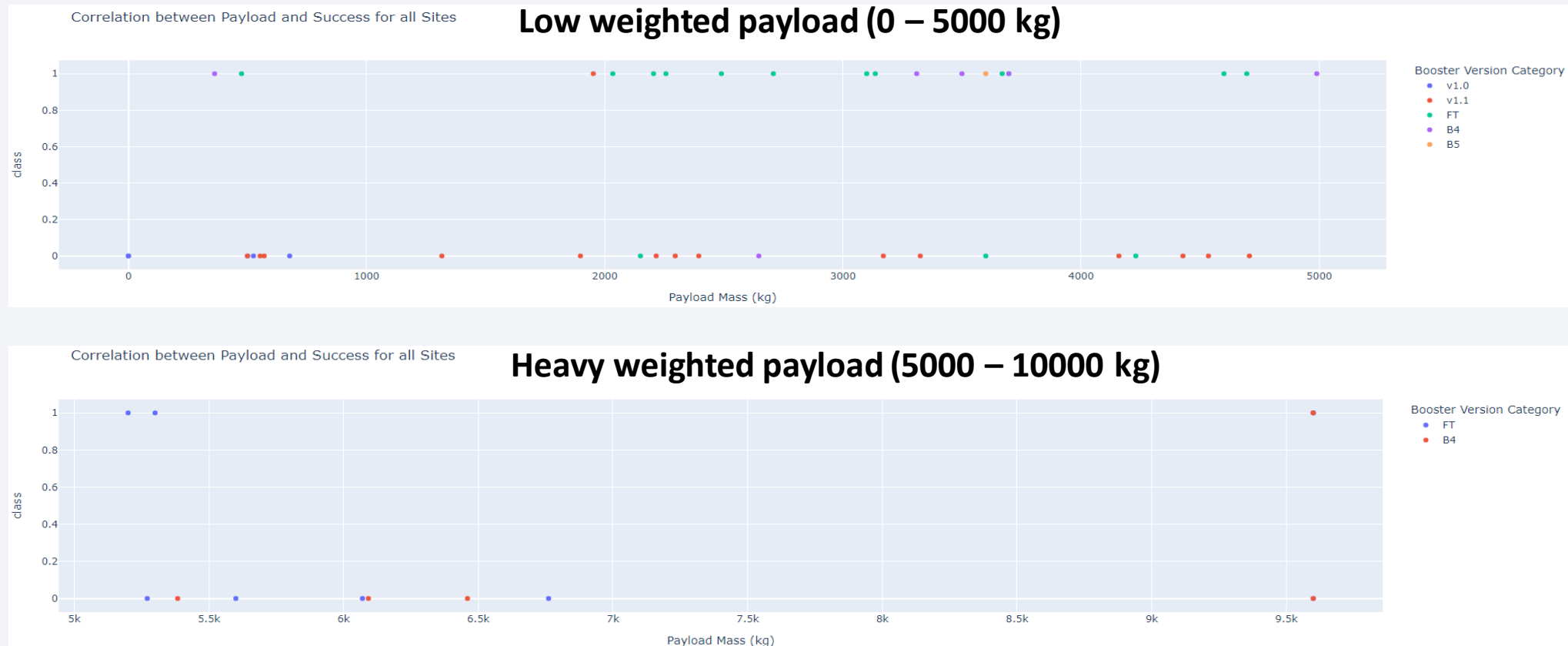
Total Success Launches for Site KSC LC-39A



■ 1
■ 0

We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard - Payload mass vs Outcome for all sites with different payload mass selected



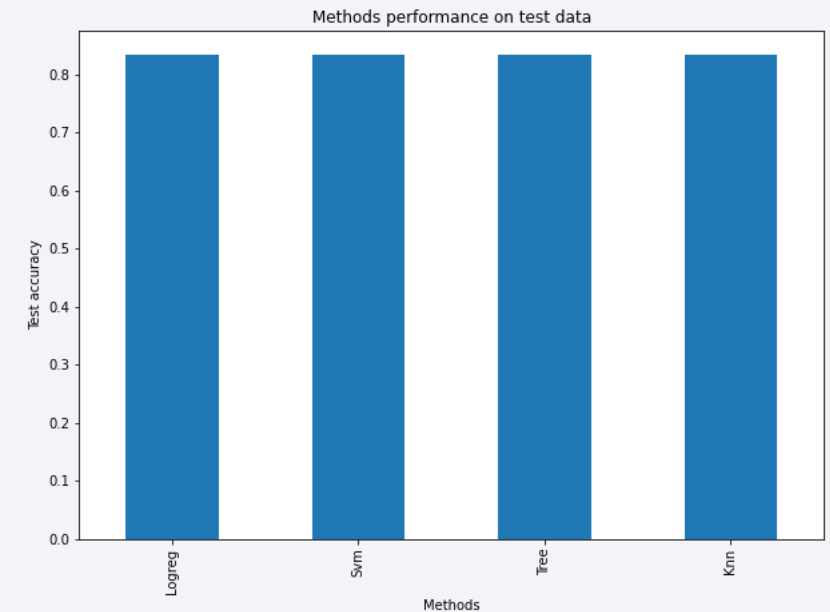
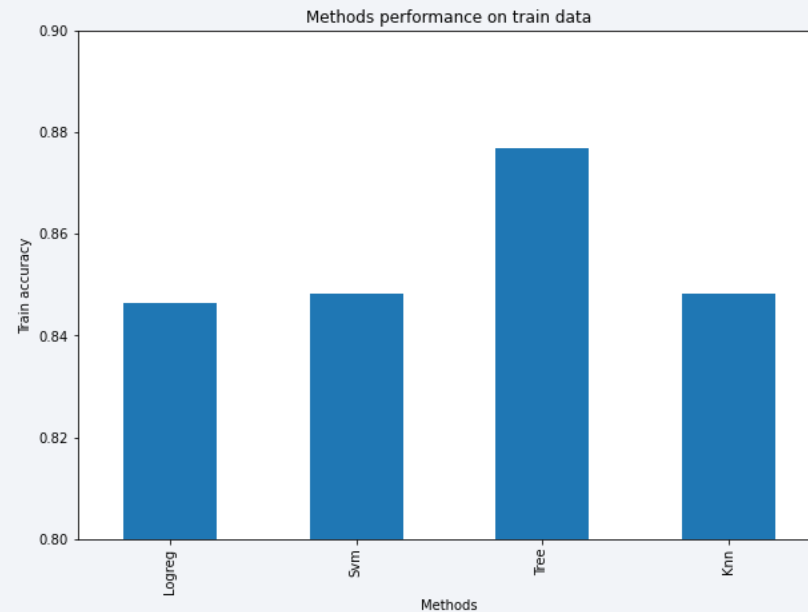
Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



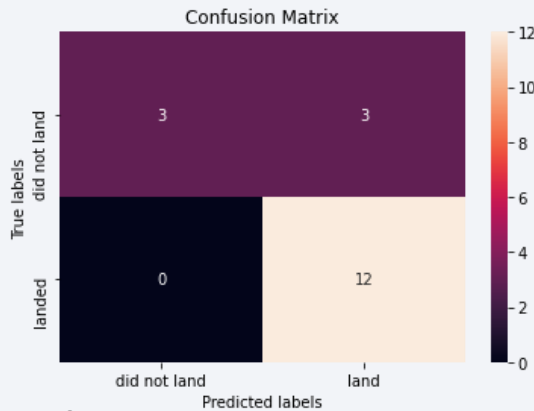
All techniques performed similarly in the accuracy test. To choose between them, we may obtain further test results. Yet if we had to make a decision immediately now, we would prefer the decision tree.

Decision tree best parameters

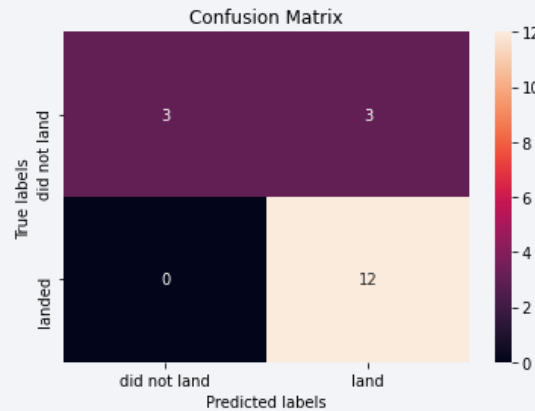
```
tuned hyperparameters :(best parameters) {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix

Logistic regression



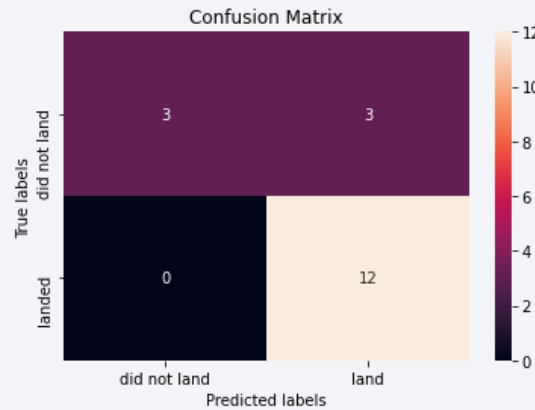
Decision Tree



kNN



SVM



The confusion matrices are the same as the test accuracy matrices. False positives are these models' biggest issue.

		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- A mission's success can be attributed to a number of variables, including the launch site, the orbit, and particularly the quantity of prior launches. In fact, we might infer that between launches, there was knowledge gained that enabled us to move from a launch failure to a launch success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- The payload mass may be a factor to consider for a mission's success depending on the orbits. It depends on the orbit whether the cargo is light or heavy. Nevertheless, smaller weight payloads typically outperform high weight payloads.
- Considering the available information, we are unable to explain why some launch sites are superior to others (KSC LC-39A is the best launch site). We might gather atmospheric data or other pertinent information to find a solution to this issue.
- Even though the test accuracy for all the models utilized is the same, we decide that the Decision Tree Algorithm is the best model for this dataset. Because the Decision Tree Algorithm offers a higher train accuracy, we opt for it.

Thank you!

