IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Denis Wave
2024.12.11.

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through Web Scraping
  - Machine learning models were built
  - Data Visualisation with Python
- Summary of all results
  - Optimal model for Predictive analysis
  - Data Visualisation for decision making

Introduction

- Project background and context
  - The commercial space industry has entered an era where space travel is becoming more accessible and affordable for the broader public. Various companies are now offering suborbital spaceflights, with SpaceX emerging as the most prominent player in this field.
  - SpaceX's notable achievements include:
    - Sending spacecraft to the International Space Station (ISS),
    - Deploying Starlink, a satellite constellation providing global internet coverage,
    - Conducting manned space missions.
  - One key factor behind SpaceX's success is the relatively low cost of its rocket launches.
- Problems you want to find answers
  - The primary goal of this project is to estimate the cost of each SpaceX launch by analyzing the reusability and performance of its Falcon 9 rockets.

Section 1

Methodology

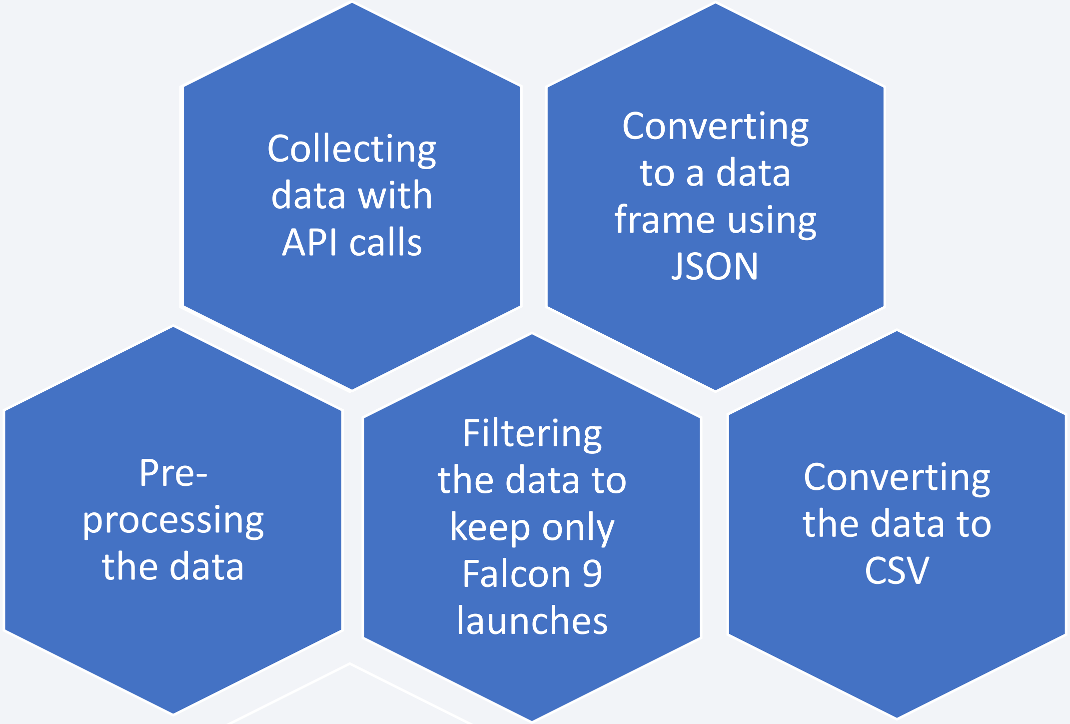
Methodology

Executive Summary

- Data collection methodology:
  - Data collected by Web Scraping
- Perform data wrangling
  - Exploratory Data Analysis to find patterns and determine the training set
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Visualisation with with Pandas, Numpy and Seaborn libraries
- Perform interactive visual analytics using Folium and Plotly Dash
  - Folium and Dash were used for Dynamic Data Visualisation
- Perform predictive analysis using classification models
  - Create machine learnind pipline for prediction

Data Collection

Data were collected using the API call method from the SpaceX API, available at https://api.spacexdata.com/v4.



Data Collection – SpaceX API

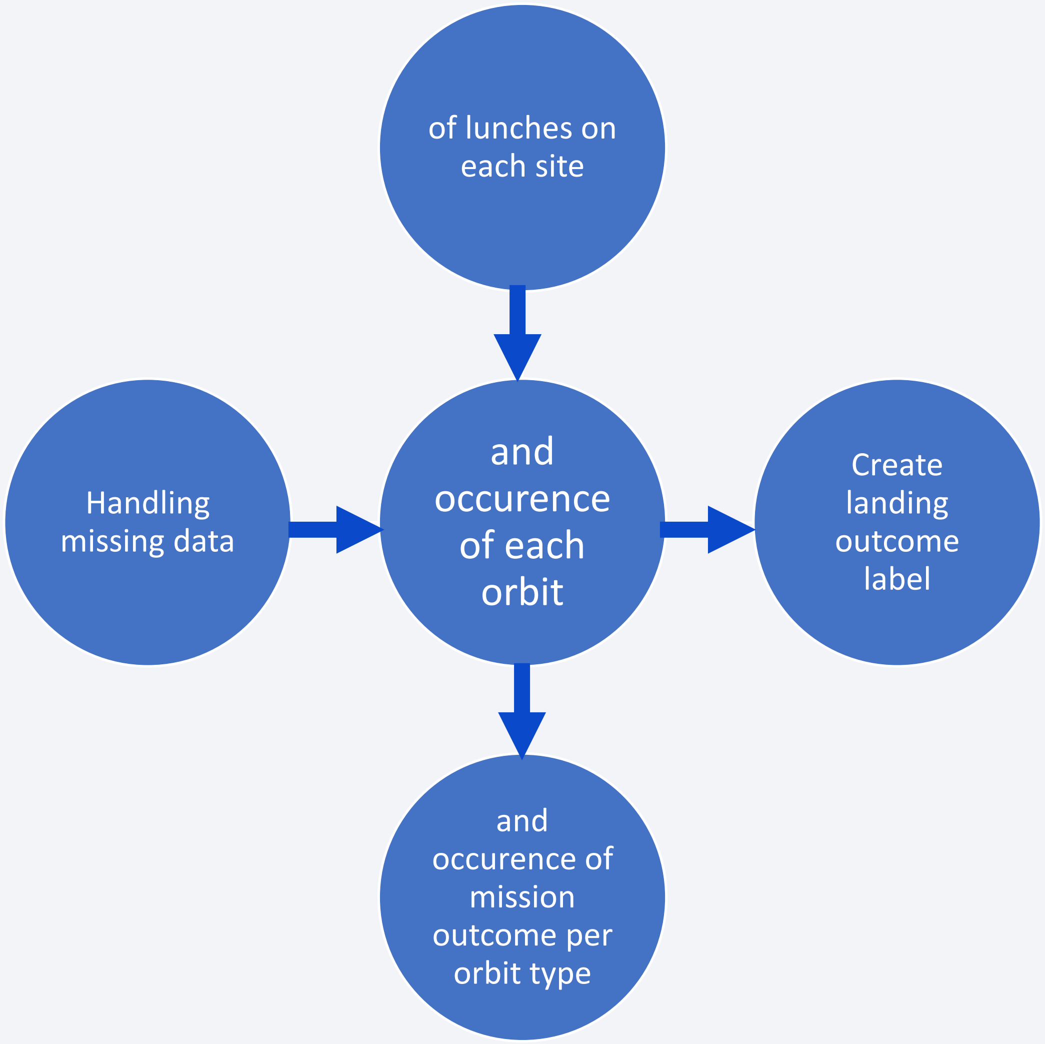
1. Request rocket launch data from SpaceX API
2. Decode the response as JSON and convert it to a DataFrame
3. Pre-process the data and correct column names
4. Construct adataset using the obtained data

Data Collection - Scraping

1. Request the Falcon9 Lunch Wiki page
2. Extract all column names from the HTML header
3. Create a FataFrame by prasing the launch HTML tables

Data Wrangling

Calculate the number:



EDA with Data Visualization

- Used graphs for the better understanding:
  - Scatter Plot
  - Bar Chart
  - Line Chart

EDA with SQL

- Retrieve the names of unique launch sites used in space missions.
- Display five records where the launch site names start with the string "CCA".
- Calculate the total payload mass carried by boosters launched by NASA (CRS).
- Find the average payload mass carried by the booster version "F9 v1.1".
- Identify the date of the first successful ground pad landing.
- List the names of boosters that successfully landed on a drone ship and carried payloads between 4000 and 6000 kilograms.
- Provide the total count of successful and failed mission outcomes.
- Identify the booster versions that carried the maximum payload mass.
- List the failed drone ship landing outcomes along with the corresponding booster versions and launch site names.

Build an Interactive Map with Folium

Map objects summary:

- Markers: Display a geographic location using latitude and longitude data.
- Cluster: Represent a group of markers.
- Circles: Highlight a single location on the map.
- Lines: Indicate the distance or connection between two locations.

Build a Dashboard with Plotly Dash

Plot summary:

- Bar Chart: Illustrates differences between categories.
- Line Chart: Displays changes over time in a time series.
- Pie Chart: Represents the percentage distribution of events.
- Tree Map: Visualizes complex relationships between variables interactively.
- Map: Depicts variables geographically, such as across different states.

Predictive Analysis (Classification)

1. Create a variable Y and standardize the variable assigned to X
2. Split in train and test database
3. Calculate logistic regression, decison tree and knn
4. Calculatethe number and occurence of mission outcome per orbit type
5. Compare score accurancy to chose the best model

Results

- Exploratory data analysis results
    - Web scraping is capable of collecting SpaceX Data
- Interactive analytics demo in screenshots
    - Data analysis with SQL is effective for filtering data.
    - Data analysis with interactive visualization provides insights.
    - Plotly Dash is powerful for displaying data changes.
- Predictive analysis results
    - The Decision Tree Classifier algorithm has good accuracy for prediction.

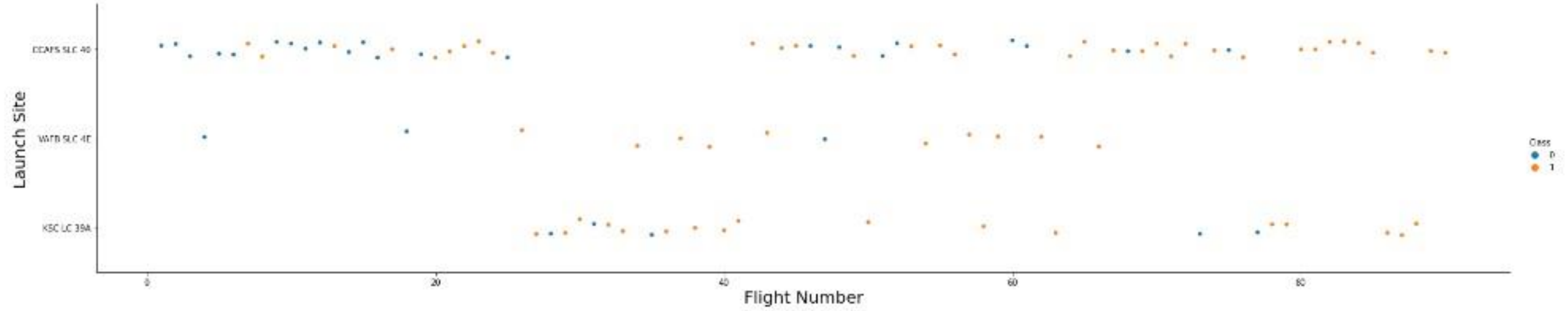
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

In [4]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class va
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```

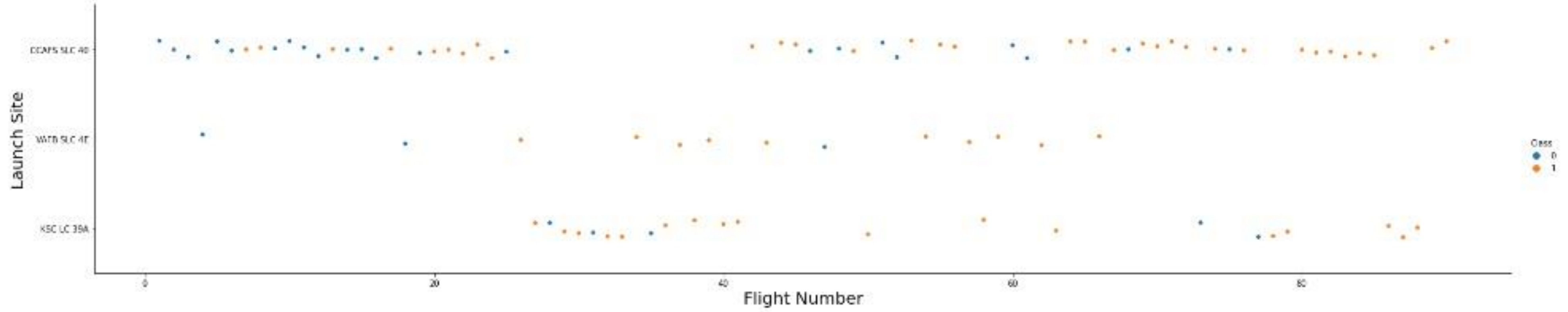


- For the CCFS SLC40 category, there seems to be a higher concentration of flights with a high number compared to class 1

GitHub link: https://github.com/hullamd/DSCapstone

Payload vs. Launch Site

```
In [5]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the cla
        sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
        plt.xlabel("Flight Number",fontsize=20)
        plt.ylabel("Launch Site",fontsize=20)
        plt.show()
```

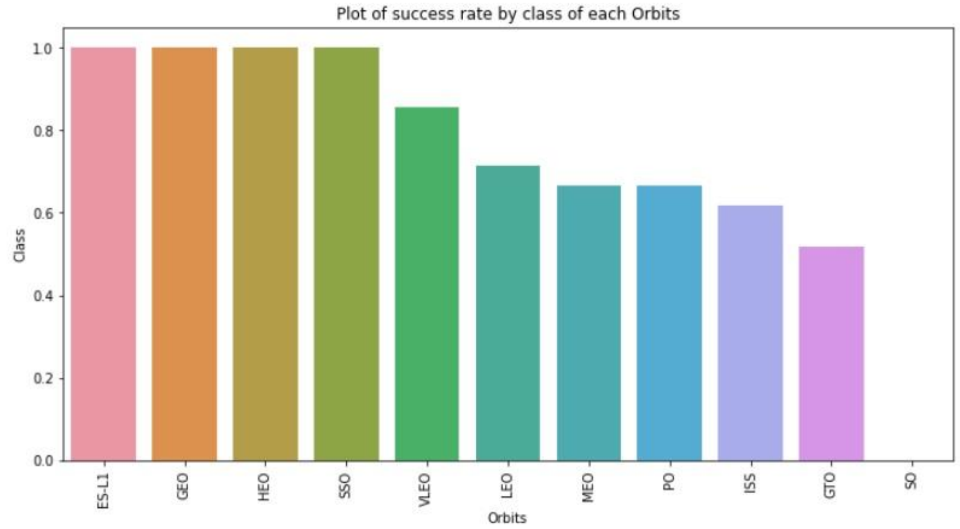


- In this case, there does not appear to be a strong correlation

GitHub link: https://github.com/hullamd/DSCapstone

Success Rate vs. Orbit Type

```
In [8]: # Use groupby method on Orbit column and get the mean of Class column
        grouped_orbits = df.groupby(by=['Orbit'])['Class'].mean().sort_values(ascending=False).reset_index()
        fig, ax=plt.subplots(figsize=(12,6))
        ax = sns.barplot(x = 'Orbit', y = 'Class', data=grouped_orbits)
        ax.set_title('Plot of success rate by class of each Orbits', fontdict={'size':12})
        ax.set_ylabel('Class', fontsize = 10)
        ax.set_xlabel('Orbits', fontsize = 10)
        ax.set_xticklabels(ax.get_xticklabels(), fontsize = 10, rotation=90);
```



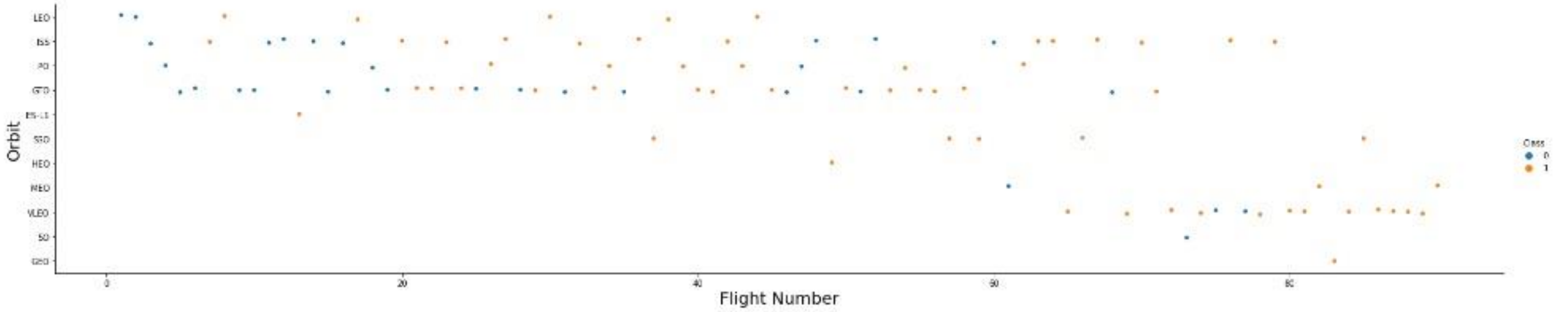
- We can see that the orbits with the highest success rates are SSO, HEO, GEO, and ES-L1, while the GTO orbit has the lowest success rate.

GitHub link: https://github.com/hullamd/DSCapstone

Flight Number vs. Orbit Type

In [9]:

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



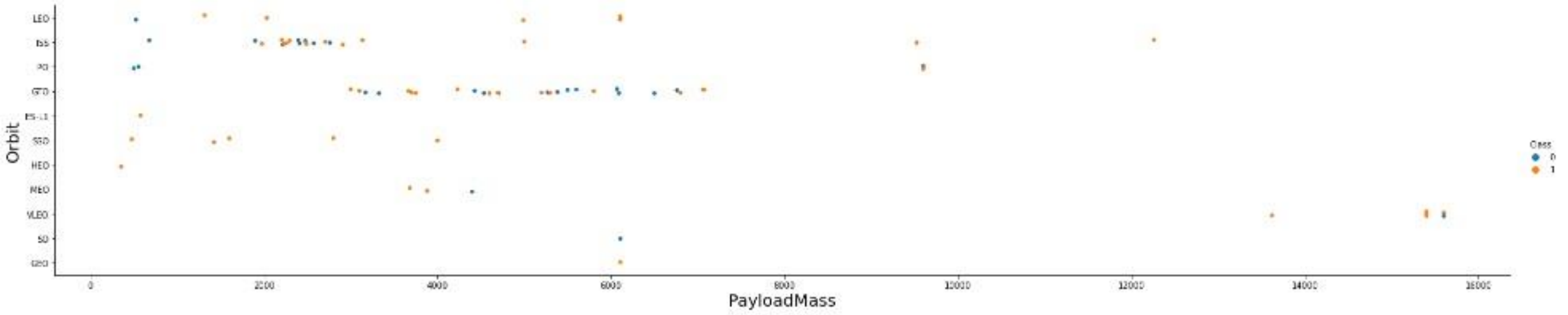
- In the LEO orbit, success appears to be related to the number of flights. On the other hand, there seems to be no relationship between the number of flights and success in the GTO orbit

GitHub link: https://github.com/hullamd/DSCapstone

Payload vs. Orbit Type

In [10]:

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



- Heavy payloads have a negative influence on GTO orbits and a positive influence on GTO and Polar LEO orbits

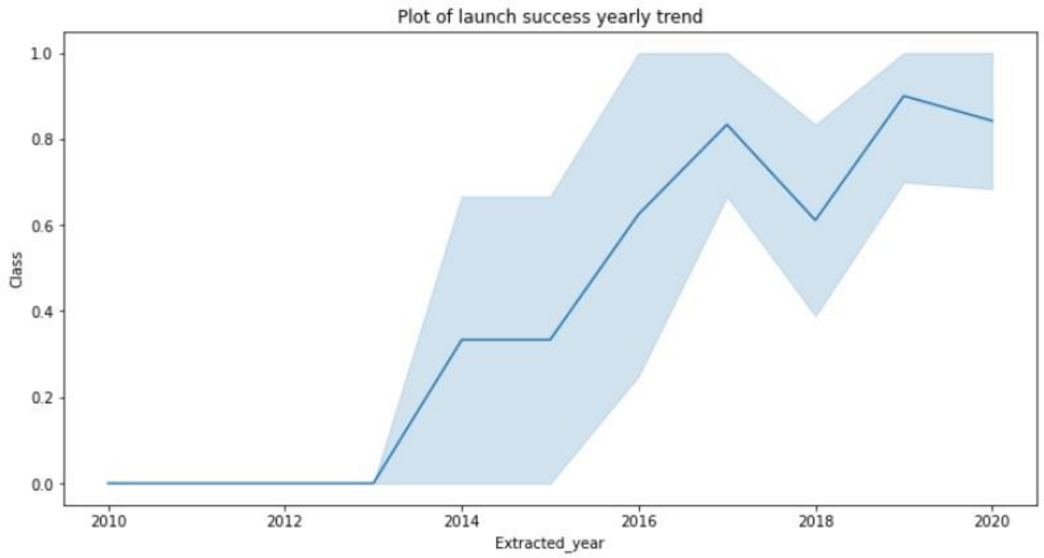
GitHub link: https://github.com/hullamd/DSCapstone

Launch Success Yearly Trend

- The success rate is increasing since 2013

In [12]:
```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df_copy = df.copy()
df_copy['Extracted_year'] = pd.DatetimeIndex(df['Date']).year

# plot line chart
fig, ax=plt.subplots(figsize=(12,6))
sns.lineplot(data=df_copy, x='Extracted_year', y='Class')
plt.title('Plot of launch success yearly trend');
plt.show()
```



GitHub link: https://github.com/hullamd/DSCapstone

All Launch Site Names

```
[10]: %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL

 * sqlite:///my_data1.db
Done.
```

[10]:

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

- 4 sites are presented in the database.

GitHub link: https://github.com/hullamd/DSCapstone

Launch Site Names Begin with 'CCA'

[23]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False

GitHub link: https://github.com/hullamd/DSCapstone

Total Payload Mass

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

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

- We can get the sum of all values by using the ‚SUM’ function,

GitHub link: https://github.com/hullamd/DSCapstone

Average Payload Mass by F9 v1.1

- We van get the average of all values by using ‚AVG’ function.

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

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

GitHub link: https://github.com/hullamd/DSCapstone

First Successful Ground Landing Date

- We can get the first succesful data by using the ‚MIN’ function.

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'

 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)']
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

GitHub link: https://github.com/hullamd/DSCapstone

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ =
```

```
 * sqlite:///my_data1.db
Done.
```

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

GitHub link: https://github.com/hullamd/DSCapstone

Total Number of Successful and Failure Mission Outcomes

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

t[16]:

	failureoutcome
0	1

- We can get the number of all successful and failure missions by using COUNT and LIKE functions

GitHub link: https://github.com/hullamd/DSCapstone

Boosters Carried Maximum Payload

- We can get the max playload masses by using MAX function.

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

GitHub link: https://github.com/hullamd/DSCapstone

2015 Launch Records

- We can get the months using month(DATEW) and WHERE functions.

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

GitHub link: https://github.com/hullamd/DSCapstone

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

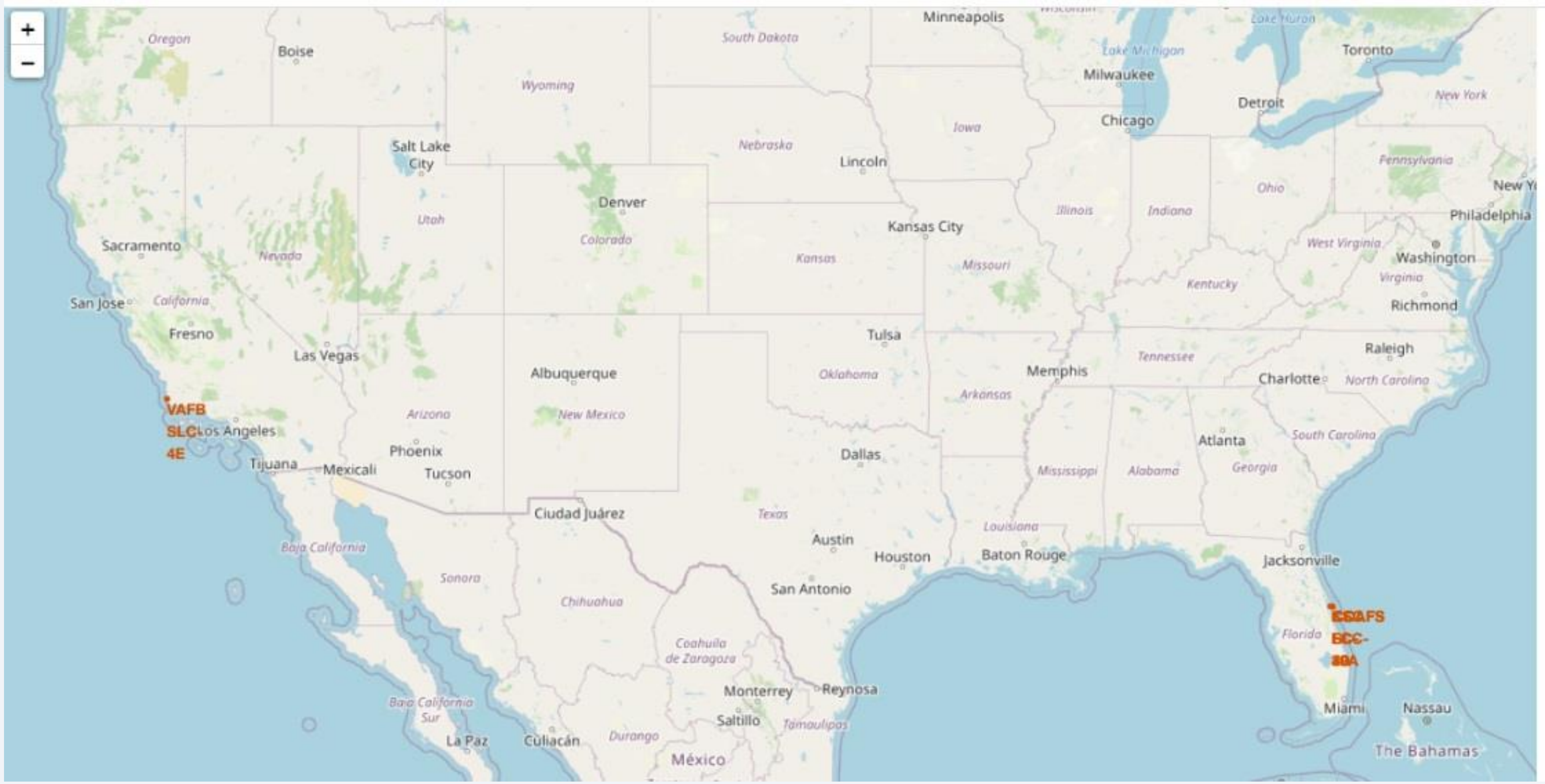
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

GitHub link: https://github.com/hullamd/DSCapstone

Section 3

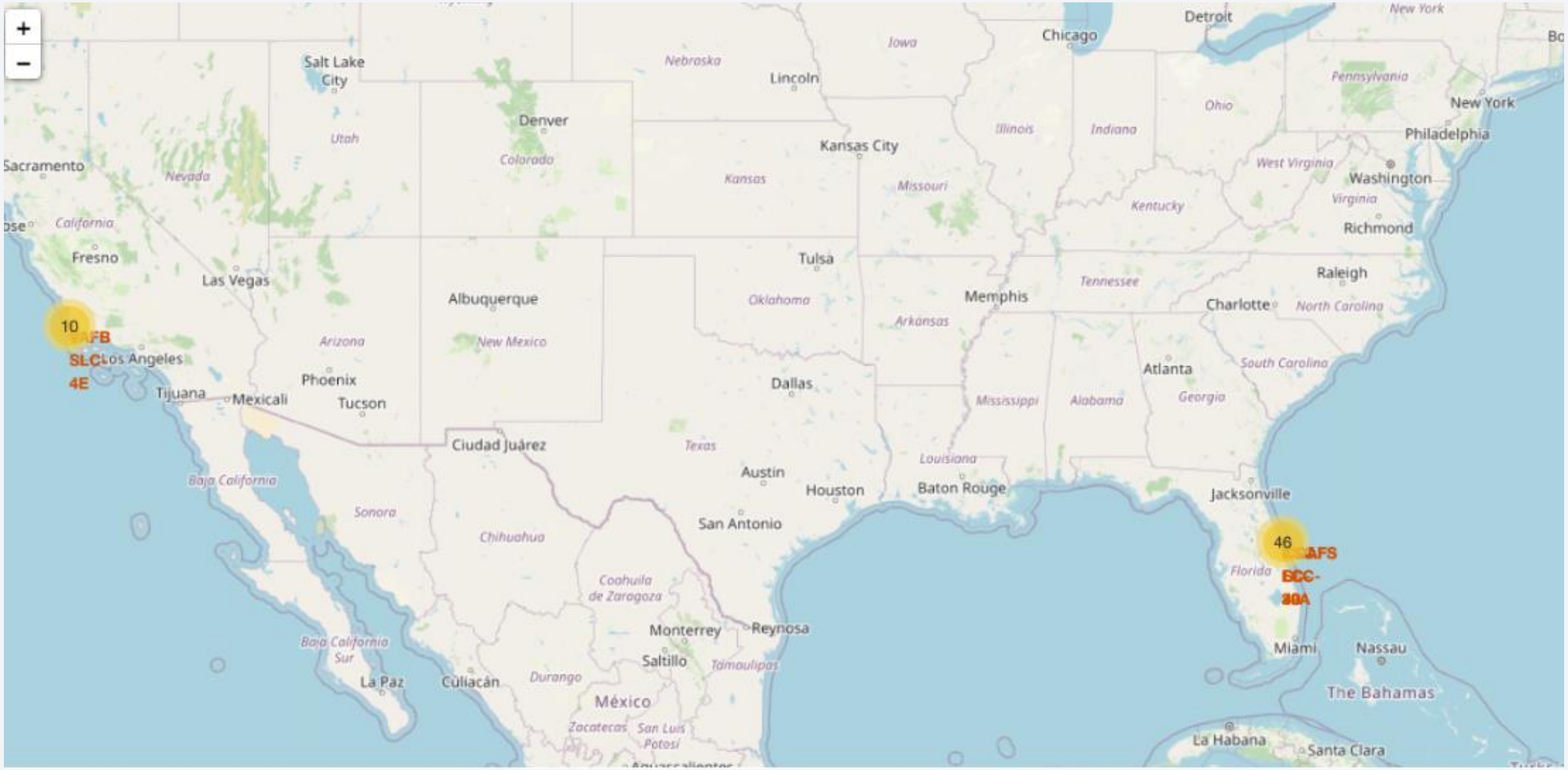
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



GitHub link: https://github.com/hullamd/DSCapstone

<Folium Map Screenshot 2>



GitHub link: https://github.com/hullamd/DSCapstone

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

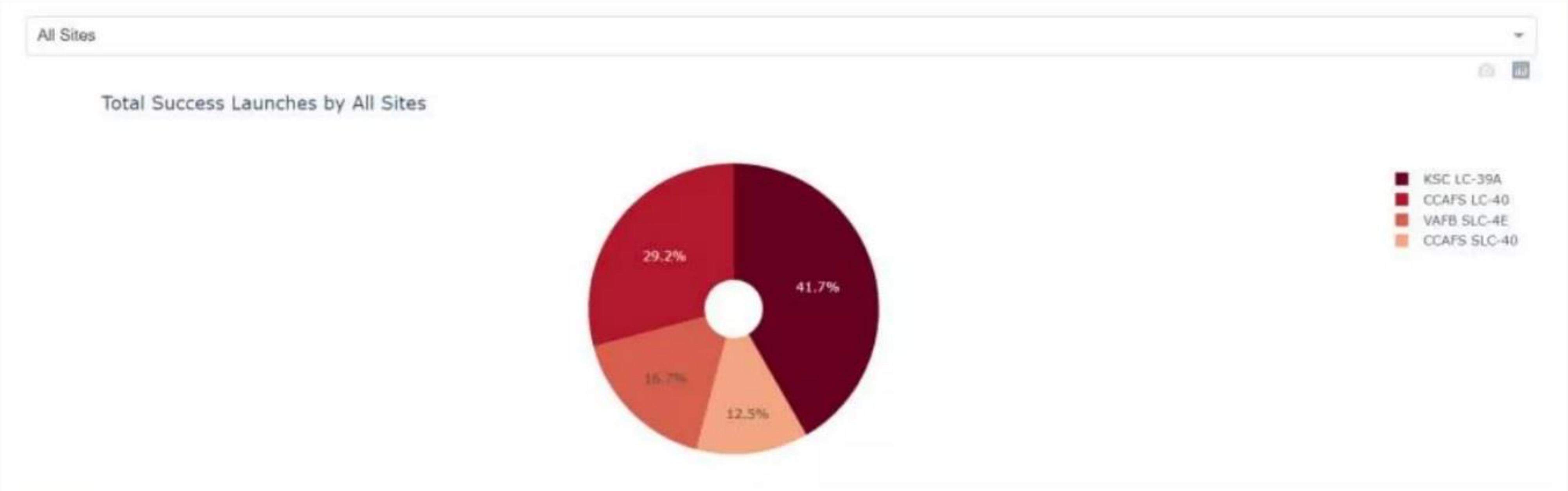
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

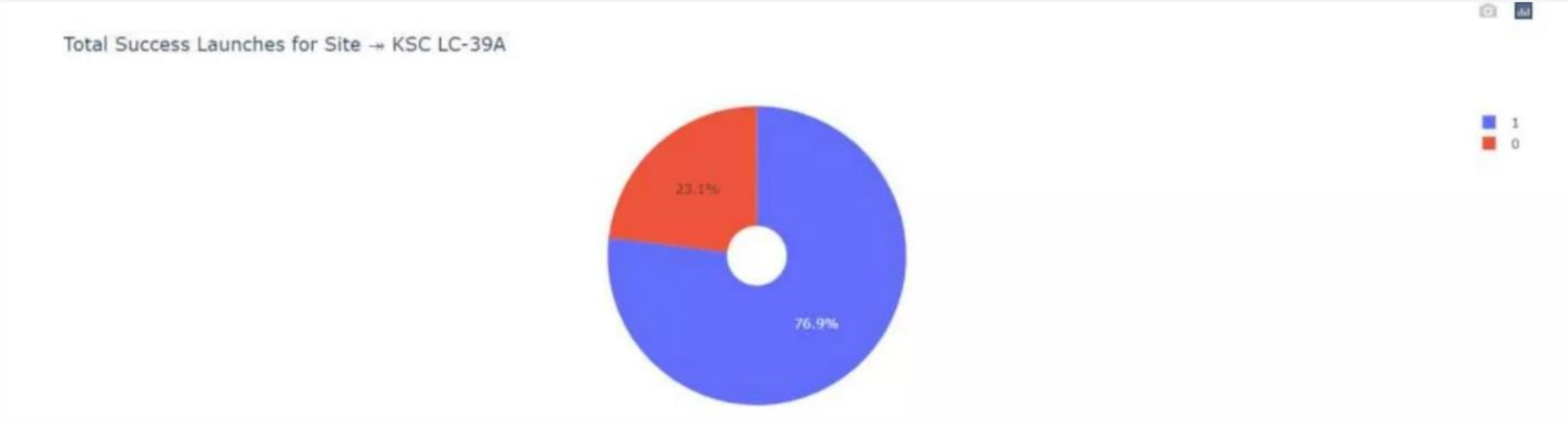
<Dashboard Screenshot 1>



- KSC LC-39A has the highest success score

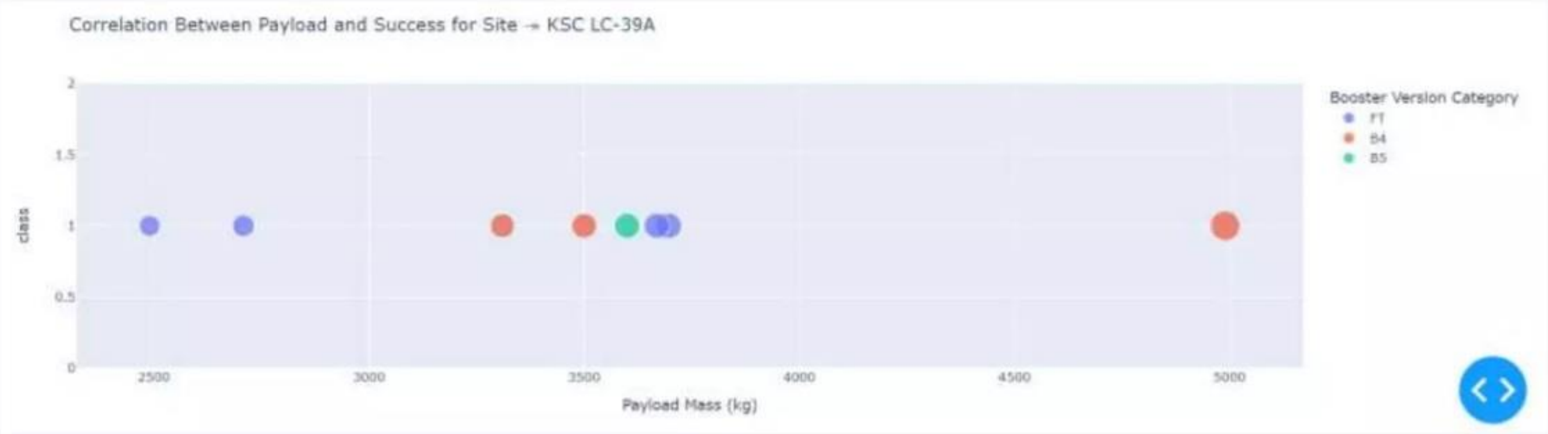
GitHub link: https://github.com/hullamd/DSCapstone

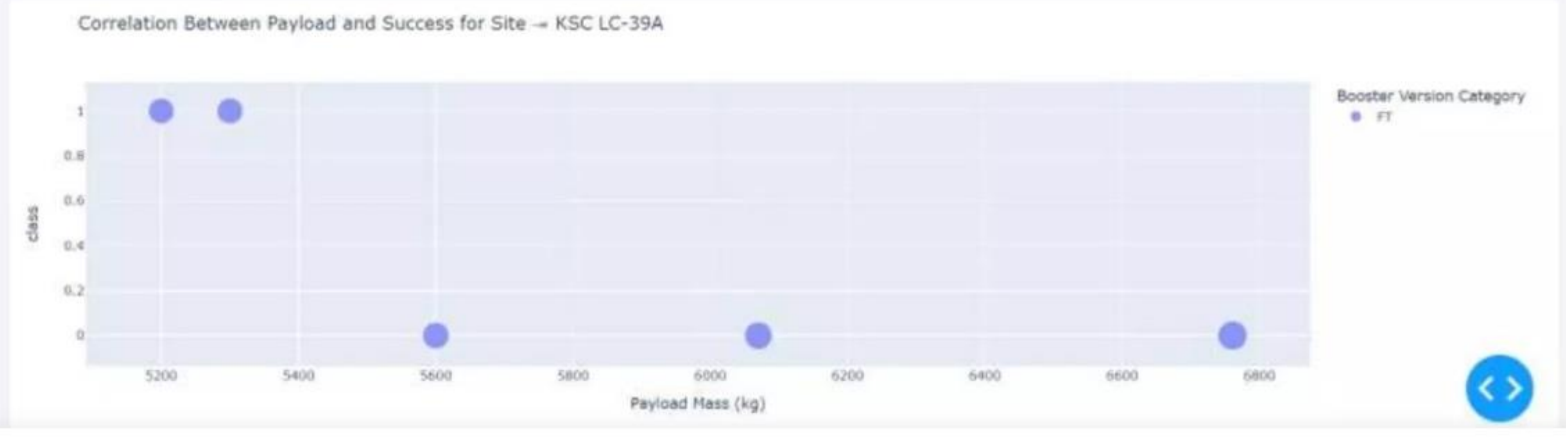
<Dashboard Screenshot 2>



- KSC LC-39A has the highest score

<Dashboard Screenshot 3>





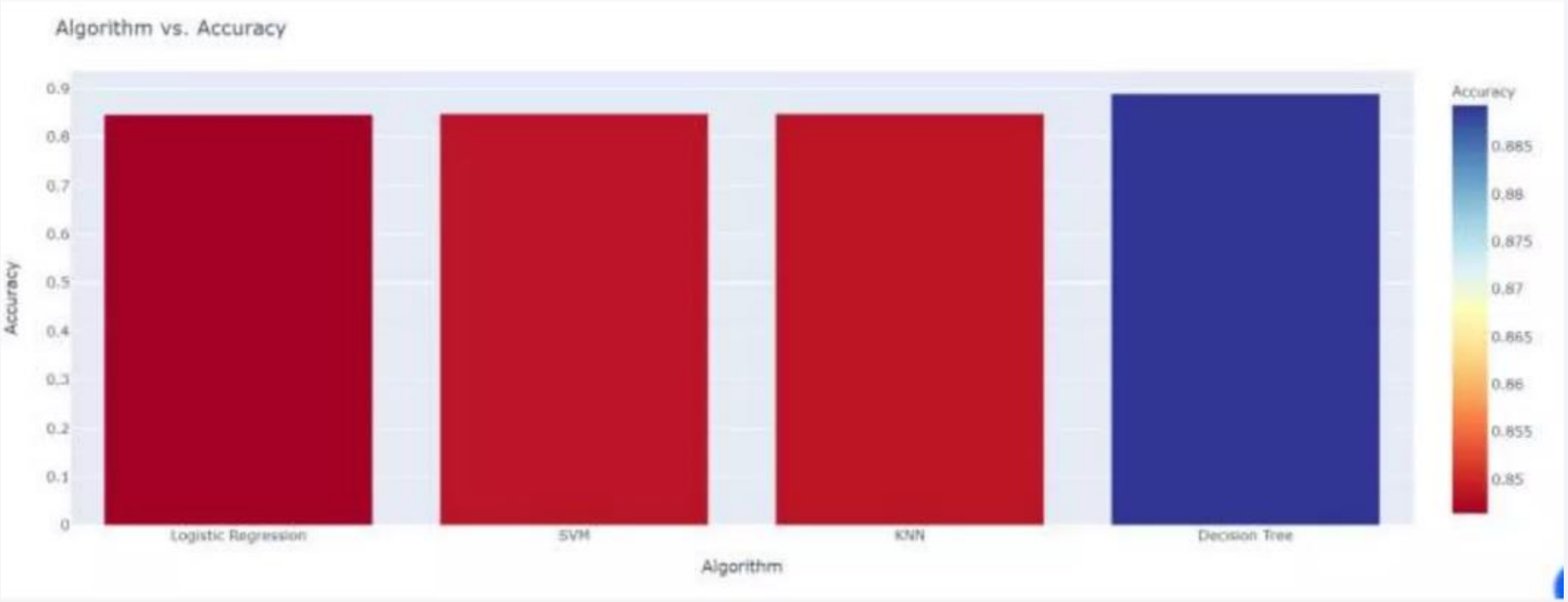
GitHub link: https://github.com/hullamd/DSCapstone

Section 5

Predictive Analysis (Classification)

Classification Accuracy

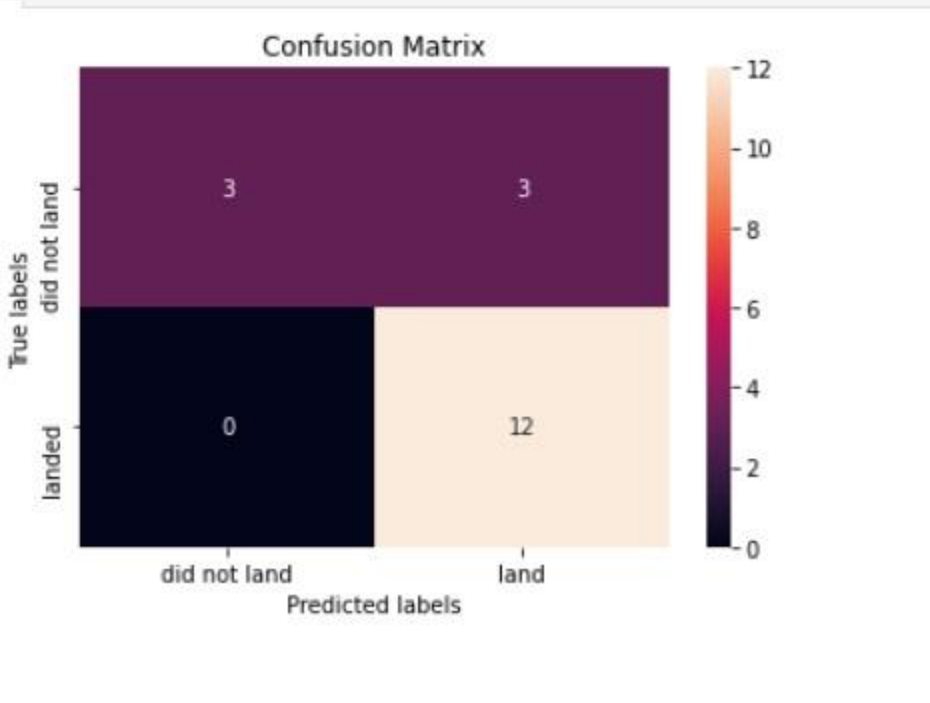
Decision Tree has the highest accuracy



GitHub link: https://github.com/hullamd/DSCapstone

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



GitHub link: https://github.com/hullamd/DSCapstone

Conclusions

- Orbits ES-L1, GEO, HEO and SSO has the hisghest success rates
- Success rates for SpaceX lanches has been increasing with time
- Decision Tree was the optimal model with accuracy of almost 0,89

Thank you!