IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<David>
<August 1, 2023>

Outline

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

Executive Summary

Summary of methodologies

Data Collection with API

Data Collection with Web Scaping

Data Wrangling

Exploratory Data Analysis w/ SQL

Exploratory Data Analysis w/ Data Visualization

Interactive Visual Analytics w/ Folium

Machine Learning Prediction

Summary of all results

Exploratory Data Analysis Results

Screenshots of Interactive Analytics

Predictive Analytics Result

Introduction

Project background and context

- Space X promotes Falcon 9 rocket launches on its website for $62 million, while other suppliers charge upwards of $165 million for each launch. The savings are mostly due to Space X's ability to reuse the first stage. If it can be determined whether the first stage will land safely, then it can be determined how much a launch will cost. The project's objective is to build a pipeline for machine learning that can forecast if the first stage can land successfully.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The features that determine the success rate of a successful landing.
- What conditions need to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data was gathered by scraping Wikipedia’s website and the Space X API.

Perform data wrangling

- One-hot encoding was applied to categorical features.

Perform exploratory data analysis (EDA) using visualization and SQL

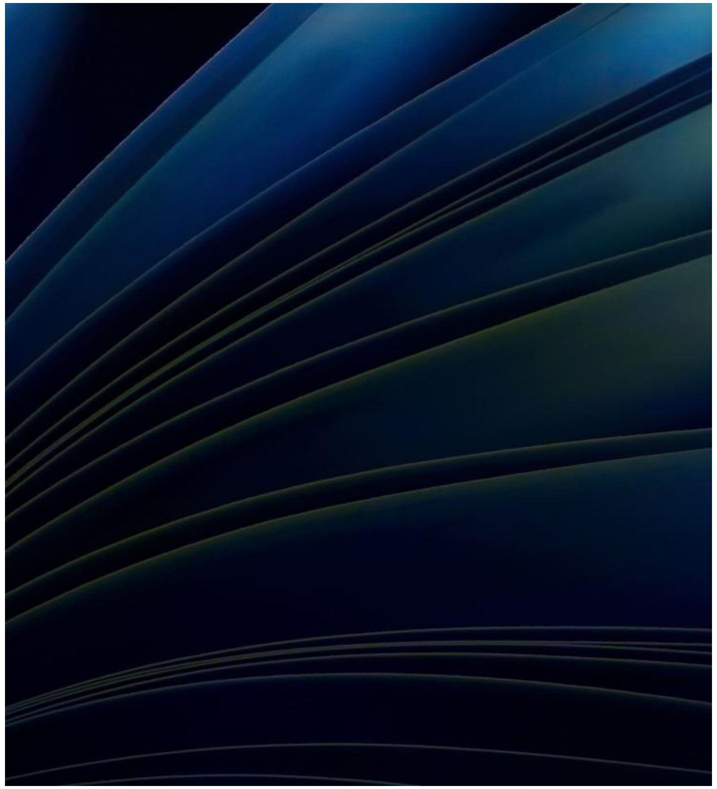
Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- How to build, tune, evaluate classification models.

Data Collection

- The data was collected using various methods -
- Utilizing a get call to the Space X API, data was gathered.
- Used the .json() function to decode the response's content as JSON and the .json_normalize() method to convert it to a pandas data frame.
- The dated was cleansed, missing values were checked for, and filled in as appropriate.
- Using Beautiful Soup, Wikipedia was scraped for information on Falcon 9 launch statistics.
- The goal was to extract the launch records as an html table, parse the table, and then transform the table into a pandas' data frame for analysis.



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

```
%load_ext sql

import csv, sqlite3

con = sqlite3.connect("my_data1.db")
cur = con.cursor()

!pip install -q pandas==1.1.5

%sql sqlite:///my_data1.db

]: 'Connected: @my_data1.db'

import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")

/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/pandas/core/generic.py:2882: UserWarning: The spaces in these
column names will not be changed. In pandas versions < 0.14, spaces were converted to underscores.
  both result in 0.1234 being formatted as 0.12.
```

lote:This below code is added to remove blank rows from table

```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null

 * sqlite:///my_data1.db
Done.

]: []

%sql select landing_outcome from 'SPACEXTABLE'
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

```
from js import fetch
import io

URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.
resp = await fetch(URL)
dataset_part_2_csv = io.BytesIO((await resp.arrayBuffer()).to_py())
df=pd.read_csv(dataset_part_2_csv)
df.head(5)
```

]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	S
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B

e

Data Wrangling

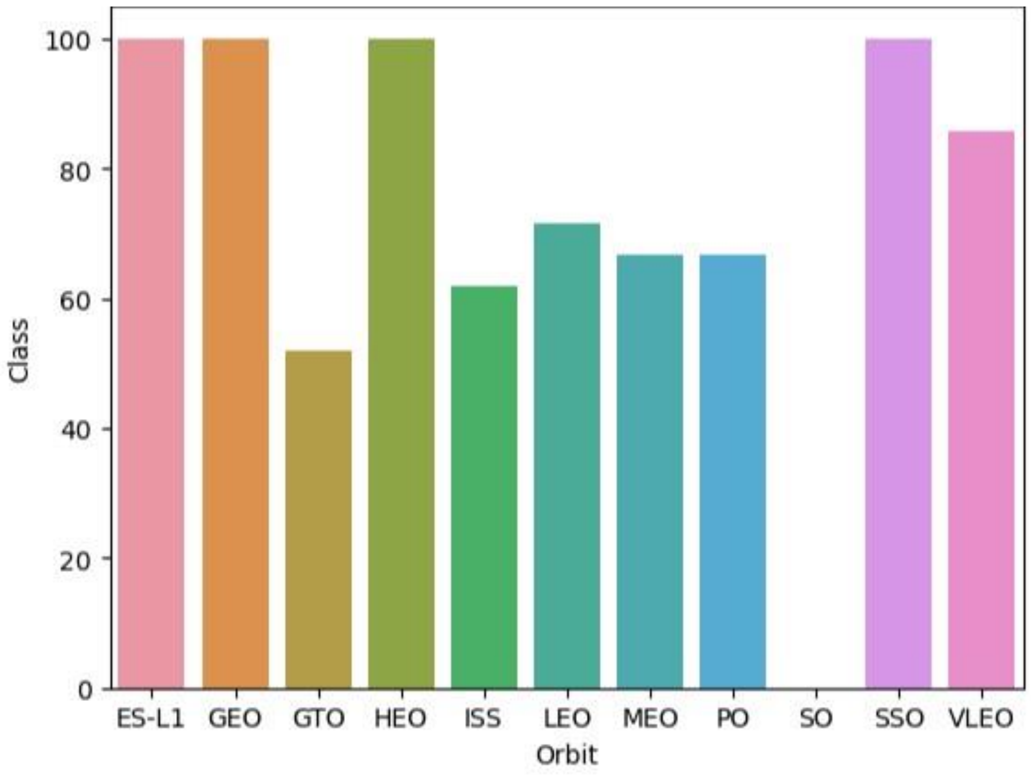
Perfored exploratory data analysis and determined training labels.

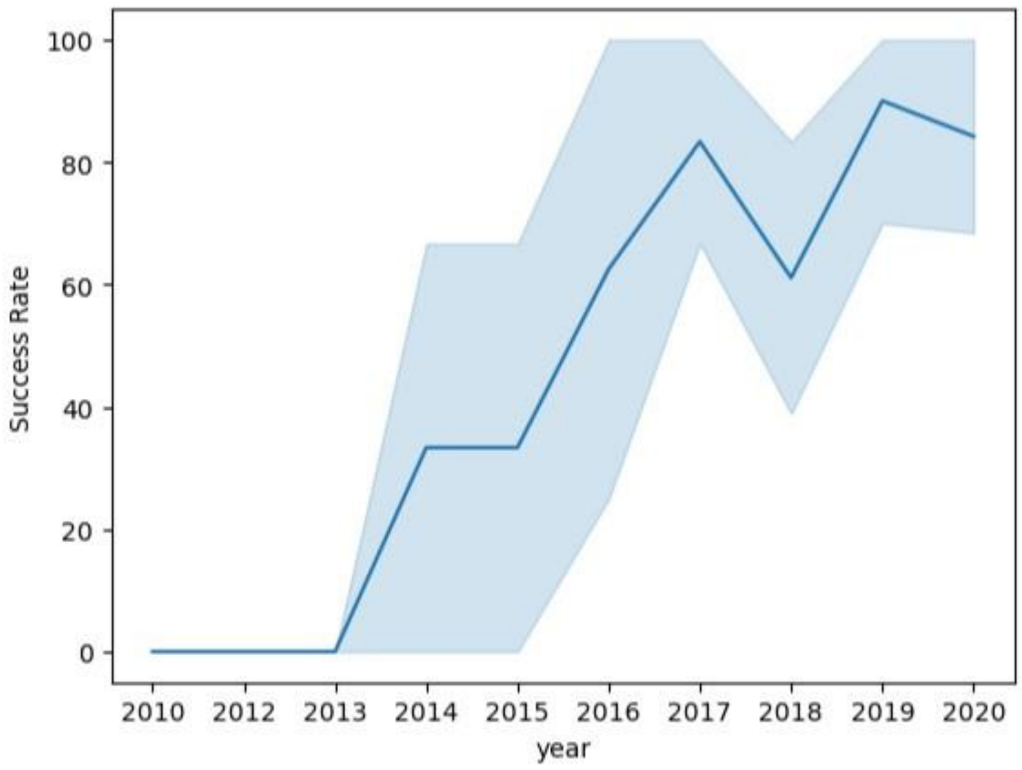
Calculated the number of launches at each site, and number and occurrence of each orbit.

Created landing outcome label from outcome column and exported the results to csv.

EDA with Data Visualization

- Determined the relationships between the variables associated with the launches which include payload, launch site, flight number, yearly trend, and orbit type.





EDA with SQL

- The Space X dataset was loaded into a PostgreSQl database using the jupyter notebook.
    - EDA with SQL was applied to get insight from the data. For example,
    - The names of unique launch sites in the space mission.
    - Total payload mass carried by boosters launched by NASA.
    - Average payload mass carried by booster version Falcon 9.
    - Total number of successful and failed missions.
    - Failed landing outcomes in drone ship, their booster version and launch site names.

Build an Interactive Map with Folium

Marked all launch sites, and added map objects such as markers, circles, lines to mark success or failure of launches for each site on the folium map.

The purpose was to answer questions such as:

- Are launch points close to highways, railroads, and the ocean?
- Do launch locations maintain a specific distance from cities?

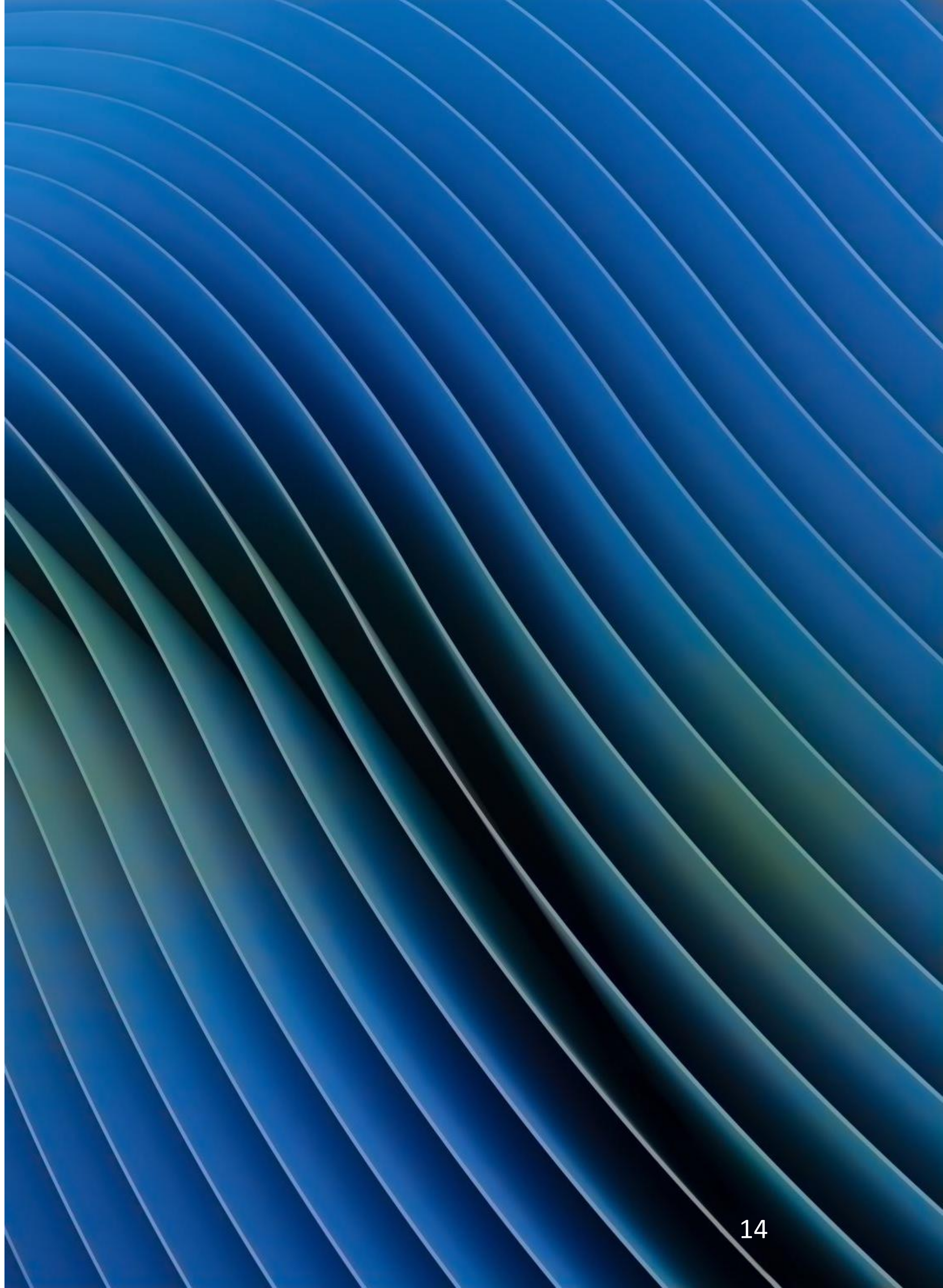
Build a Dashboard with Plotly Dash

An interactive dashboard with Plotly Dash was built.

Plotted pie charts showing total launches by certain sites.

Plotted scatter graph showing the relationship between Outcome and Payload Mass (kg) for different booster versions.

Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose



Predictive Analysis (Classification)

Summarize how you built, evaluated, improved, and found the best performing classification model

You need present your model development process using key phrases and flowchart

Results

Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

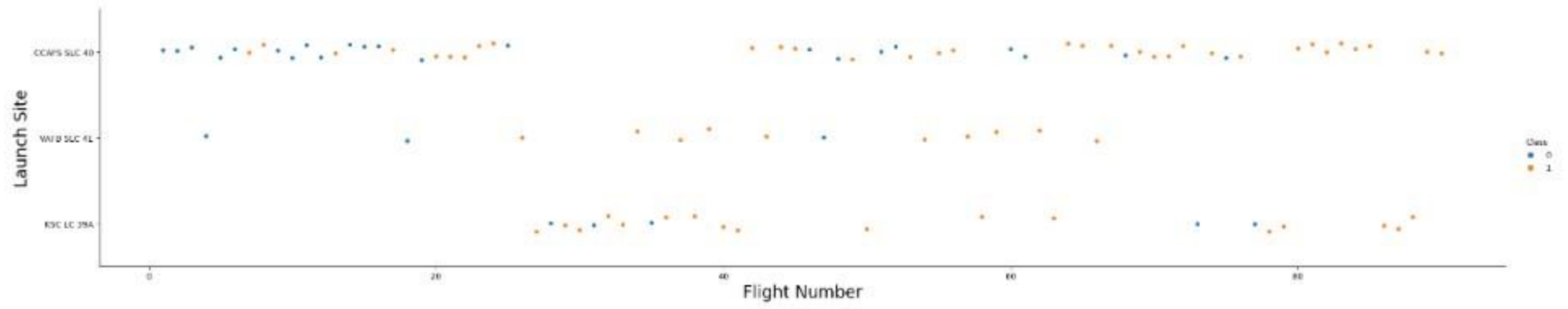
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- According to the plot below, the larger the flight amount at a launch site, the greater success rate at a launch site.

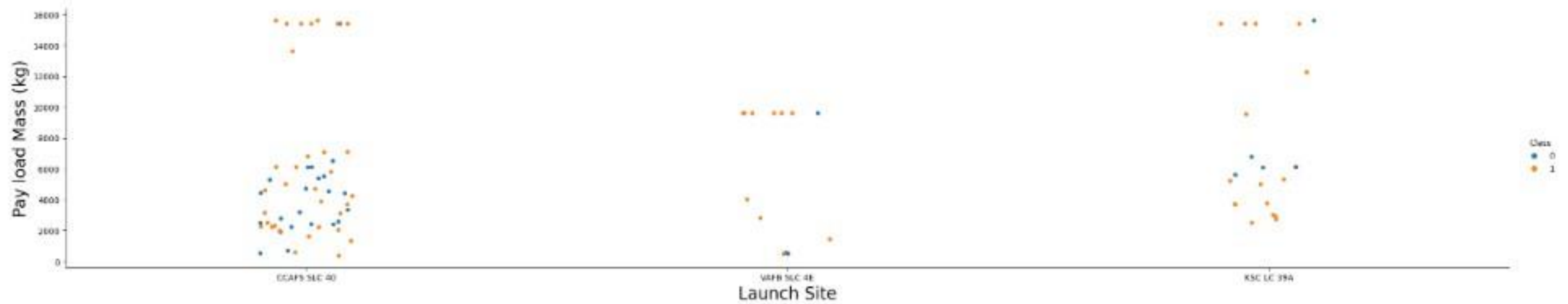
```python
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(data = df, x = "FlightNumber", y = "LaunchSite", hue = "Class", aspect = 5)
plt.xlabel("Flight Number", fontsize = 20)
plt.ylabel("Launch Site", fontsize = 20)
plt.show()
```



Payload vs. Launch Site

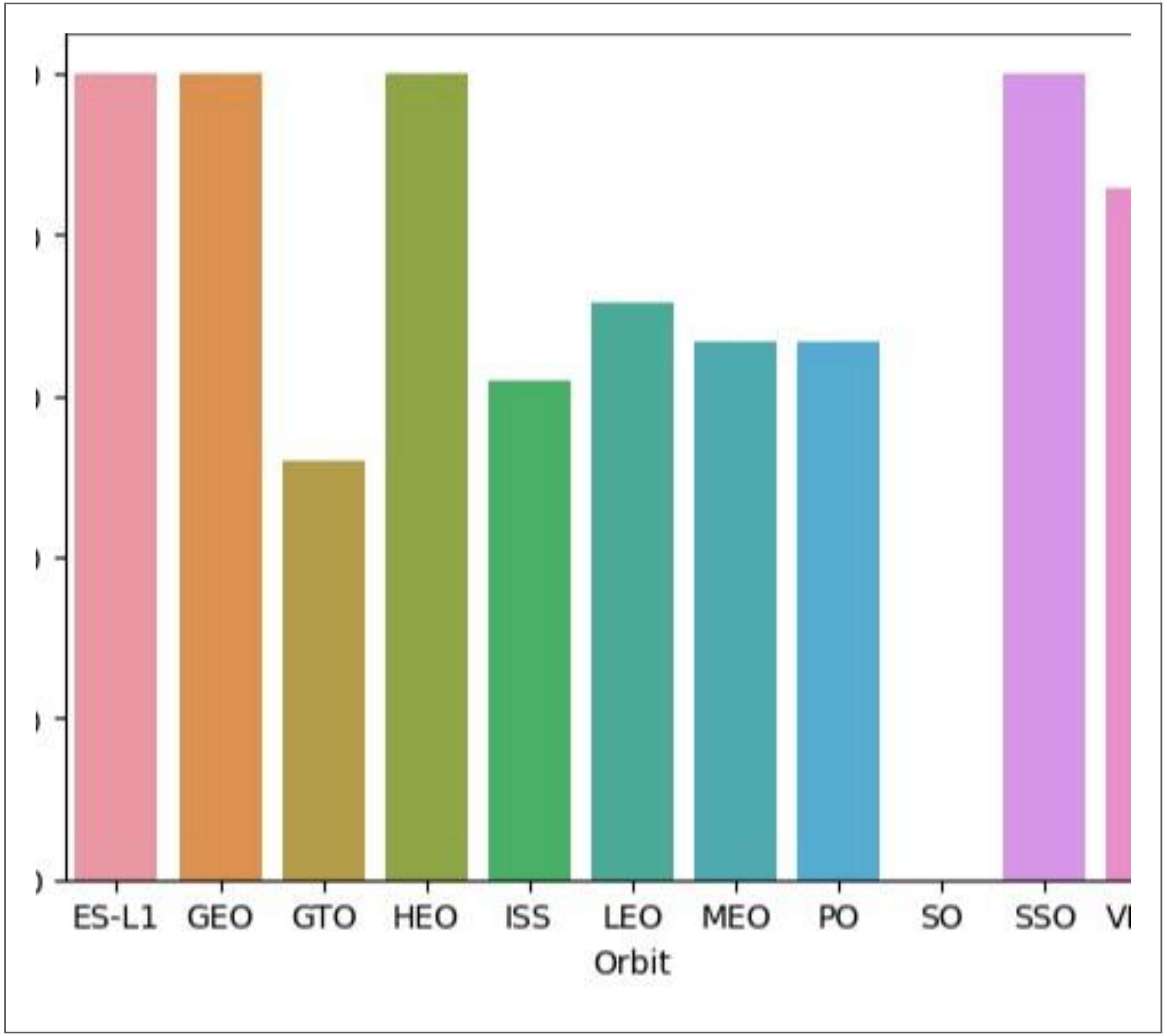
Payload vs. Launch Site

```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(data = df, x = "LaunchSite", y = "PayloadMass", hue = "Class", aspect = 5)
plt.xlabel("Launch Site", fontsize = 20)
plt.ylabel("Pay load Mass (kg)", fontsize = 20)
plt.show()
```



Success Rate vs. Orbit Type

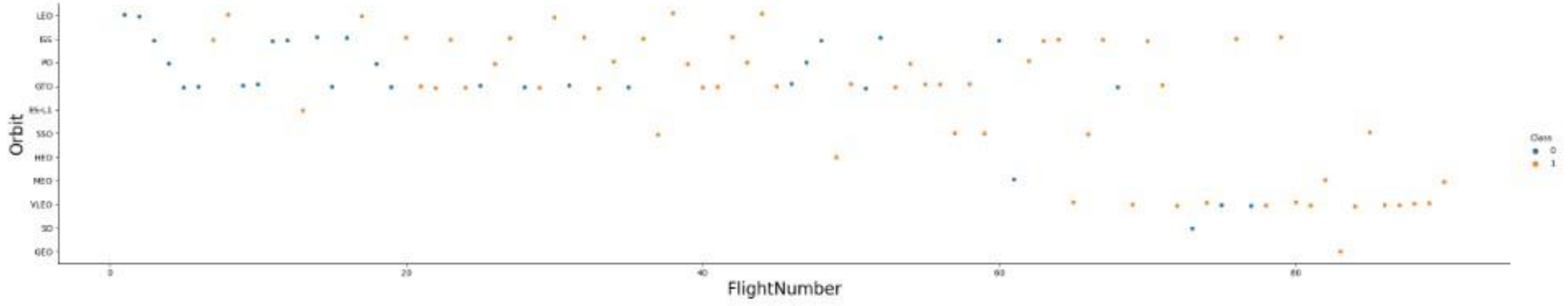
- According to the plot, ES-L1, GEO, HEO, SSO, VLEO had the highest success rate.



Flight Number vs. Orbit Type

- The LEO orbit has a success rate related to the number of flights, whereas the GTO, there is no relationship between flight number and the orbit.

```python
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(data = df, x = "FlightNumber", y = "Orbit", hue = "Class", aspect = 5)
plt.xlabel("FlightNumber", fontsize = 20)
plt.ylabel("Orbit", fontsize = 20)
plt.show()
```



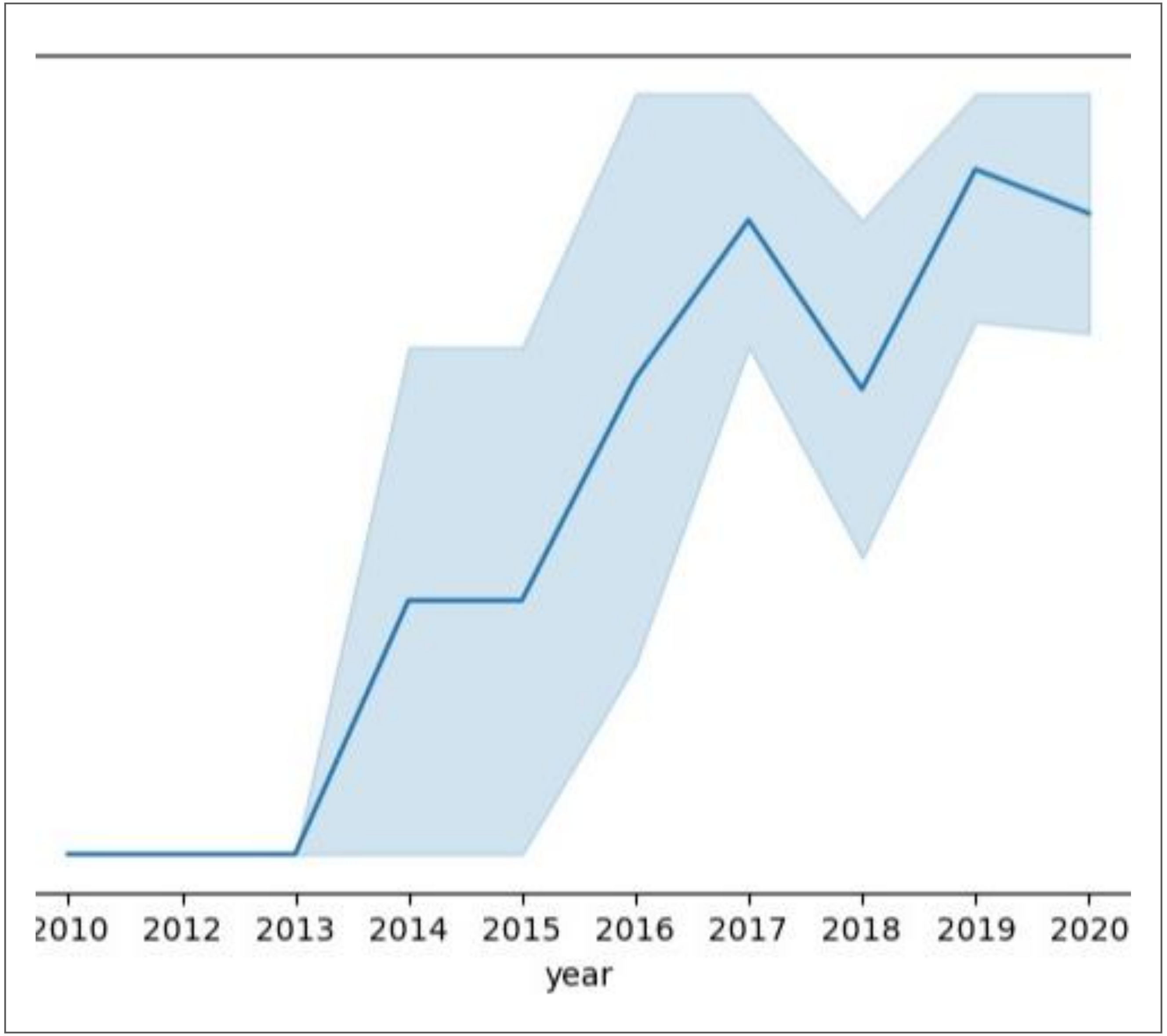
Payload vs. Orbit Type

- With heavy payloads, the successful landings are more for PO, LEO, ISS orbits.

```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(data = df, x = "PayloadMass", y = "Orbit", hue = "Class", aspect = 5)
plt.xlabel("PayloadMass", fontsize = 20)
plt.ylabel("Orbit", fontsize = 20)
plt.show()
```

Launch Success Yearly Trend

- The displays a continuing success rate from 2013 to 2020.



All Launch Site Names

- The key word DISTINCT was used to display the unique launch sites from Space X data.

```
%sql select distinct launch_site from 'SPACEXTABLE'
```

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

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

Launch Site Names Begin with 'CCA'

- The query below displays 5 records where launch sites begin with ‘CCA’.

```
%sql select launch_site from SPACEXTABLE \
where launch_site like 'CCA%' limit 5;
```

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

3]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The calculated total payload carried by boosters from NASA was 45596.

```
In [12]: task_3 = '''
                SELECT SUM(PayloadMassKG) AS Total_PayloadMass
                FROM SpaceX
                WHERE Customer LIKE 'NASA (CRS)'
                '''
         create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 1.1 = 2928.4

```
]: task_4 = '''
           SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
           FROM SpaceX
           WHERE BoosterVersion = 'F9 v1.1'
           '''
   create_pandas_df(task_4, database=conn)
```

]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- The first successful ground landing was on December 22, 2015.

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
ısk_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
reate_pandas_df(task_6, database=conn)
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
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

2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

	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)

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

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

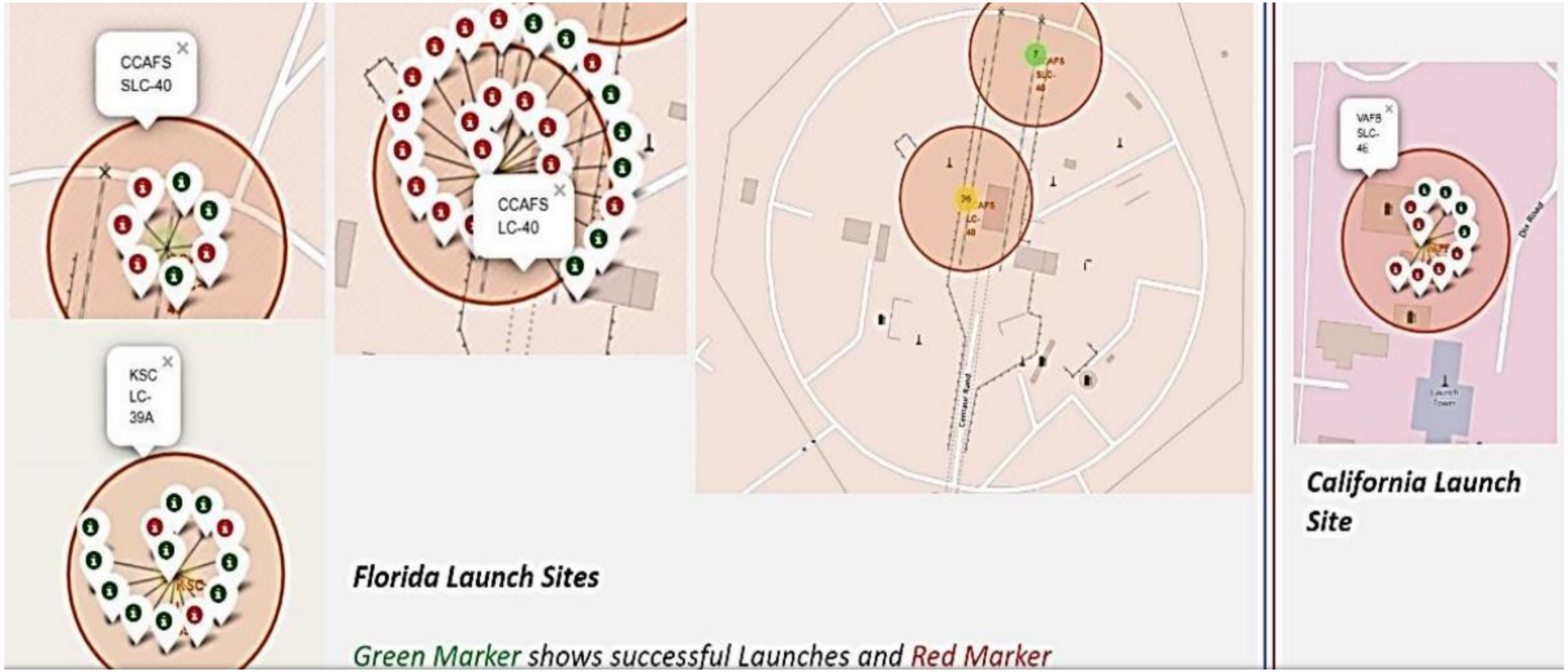
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

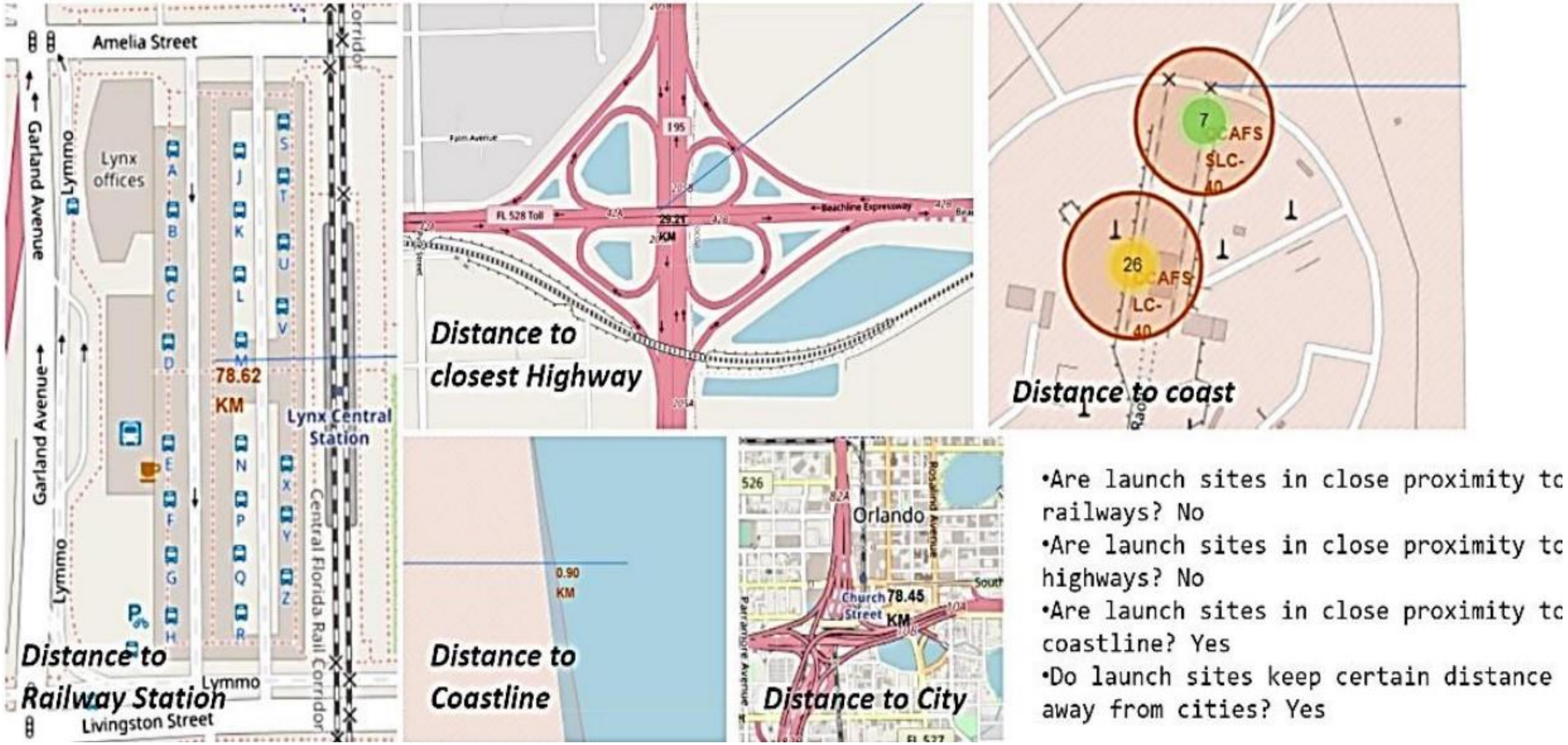
Section 3

Launch Sites Proximities Analysis

Folium Map Screenshot 1



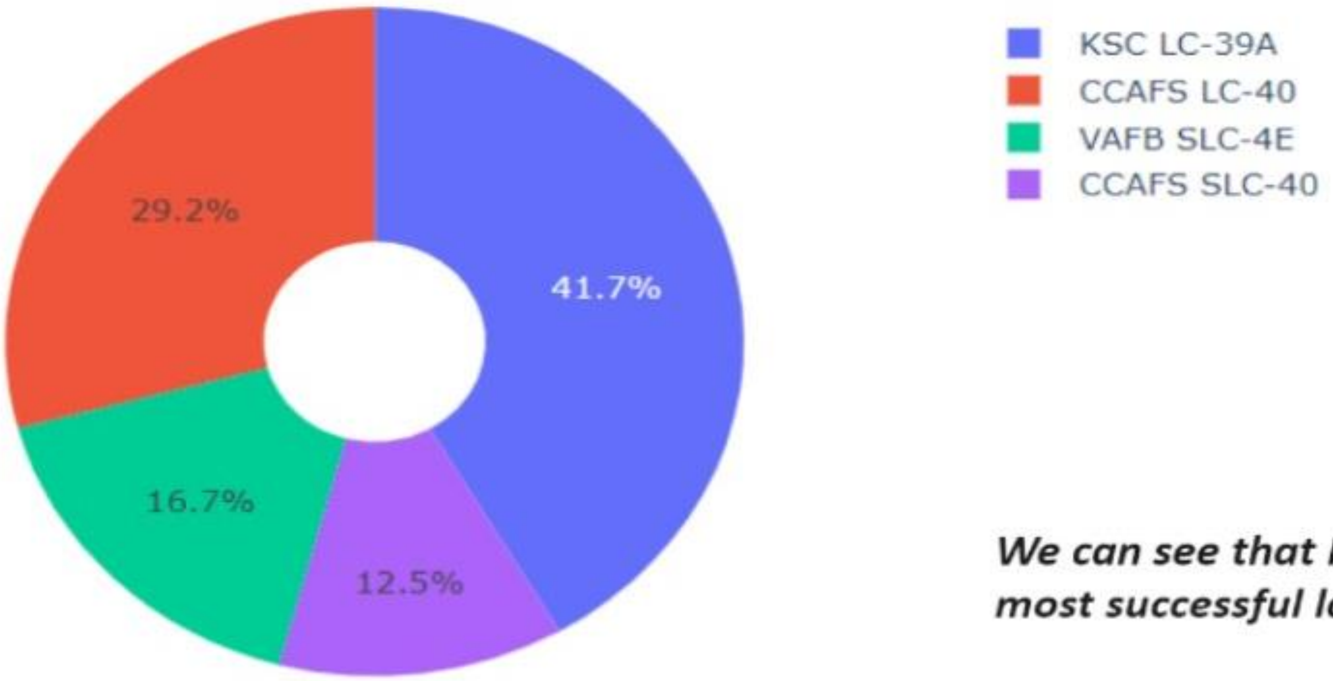
Folium Map Screenshot 2



Folium Map Screenshot 3

Section 4

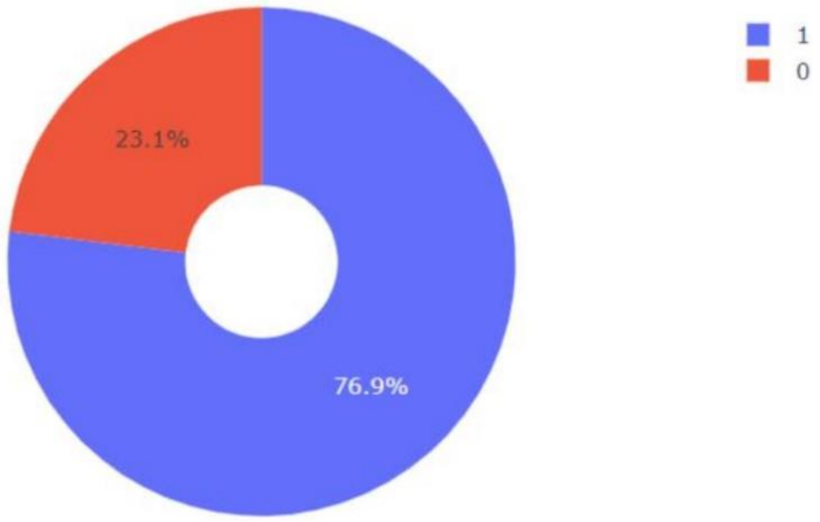
Build a Dashboard with Plotly Dash



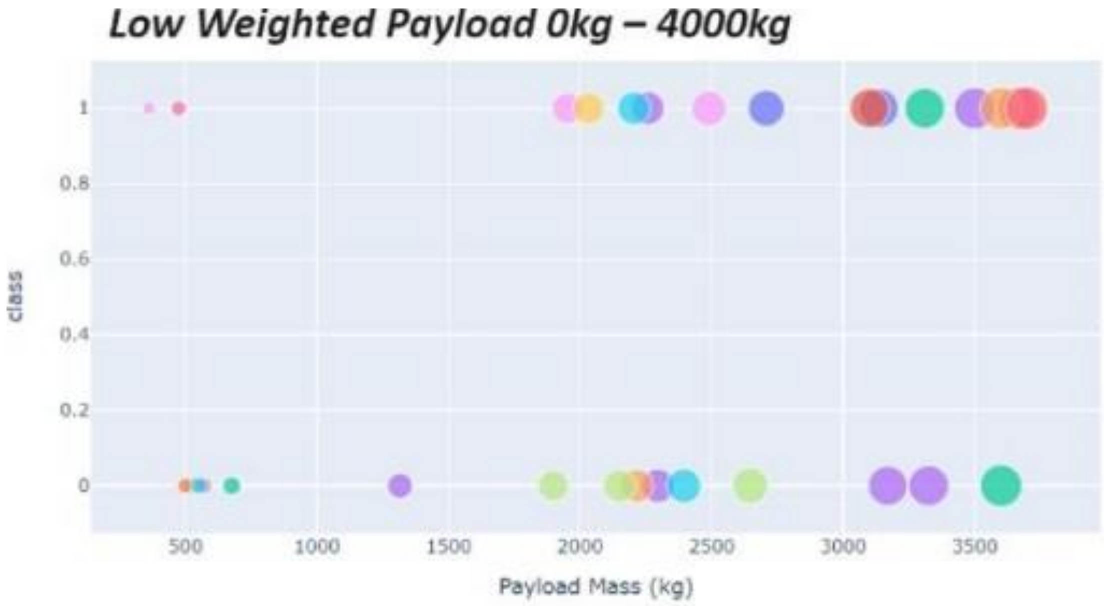
We can see that KSC LC-39A had the most successful launches from all the sites

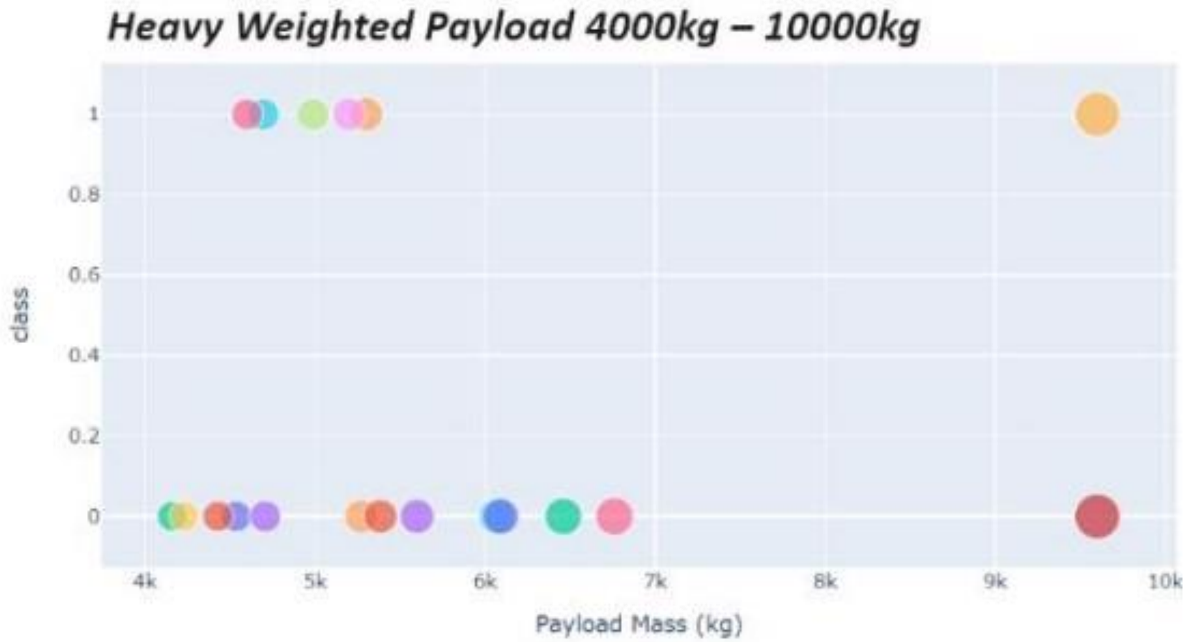
Pie Chart Displaying Success % by Launch Site

Launch Site with the Highest Launch Success Ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Payload vs Launch Outcome for All Sites

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier has the highest classification accuracy.

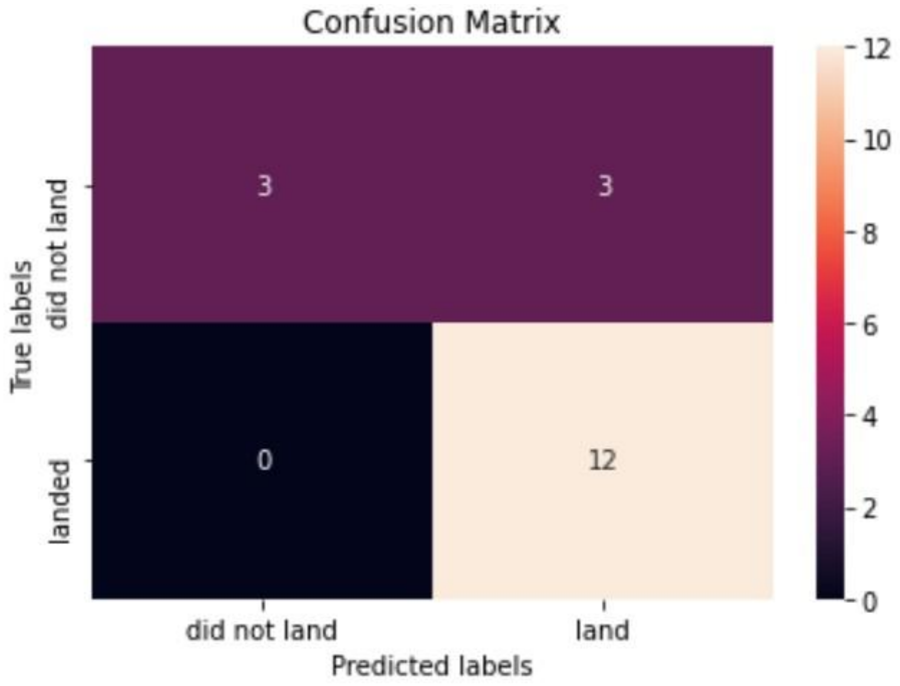
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix shows that the classifier can distinguish between the different classes.



Conclusions

- Launch success rate started to increase from 2013 to 2020.
- The larger the flight payload at a site, the greater the success rate at the launch site.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the highest success rate.
- KSC LC-39A had the most successful launches at any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!