IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Celine Gooljar

Outline

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

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- Data Analysis : EDA and SQL, Data Visualization
- Data Visualization with Folium
- Ploty Dash Dashboard
- Classification- Predictive Analysis

Summary of all results

- EDA with visualization results
- EDA with SQL results
- interactive map with Folium
- Plotly Dash dashboard
- predictive analysis (classification)

Introduction

Project background and context

The successful landing of the Flacon 9 rocket by SpaceX.

SpaceX suggest that the launch can cost up to 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. If the successful landing of the first stage can be determined the cost of a launch can be determined. This information can be useful when other companies want to bid against SpaceX for the launch.

Problems that require answers

- What variables are needed for a successful launch ?
- How does each variable affect the success rate of the launch ?
- What conditions of the variables are needed for the successful launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - Cleaning up the requested data collected
- Perform data wrangling
  - Extract the Falcon 9 launch records from Wikipedia
  - Parse the HTML table and convert it into a Pandas data frame.
  - Transform data for Machine Learning

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting graphs to show the relationship of the variables and patterns
- Perform interactive visual analytics using Folium and Plotly Dash
  - Plot maps to find geographical patterns about launch sites
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Filter

Data Collection - Scraping

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
]: # use requests.get() method with the provided static_url
   # assign the response to a object
   response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
   soup = BeautifulSoup(response.text, "html.parser")
```

```
18]: column_names = []

     table = soup.find_all('th')
     for x in range(len(table)):
         try:
          name = extract_column_from_header(table[x])
          if (name is not None and len(name) > 0):
             column_names.append(name)
         except:
          pass
```

Check the extracted column names

Create a data frame by parsing the launch HTML tables

Data Wrangling

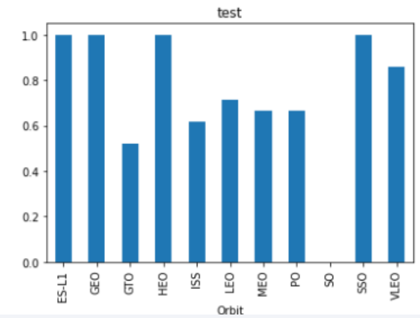
- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

EDA with Data Visualization

Bar Graph : **Success Rate vs Orbit**

- Categorical comparison for different orbit types

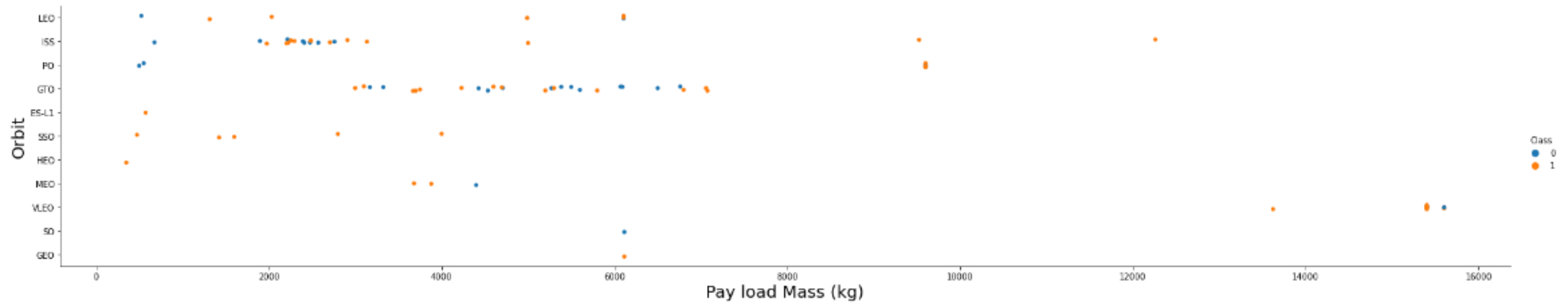
```
In [7]: # HINT use groupby method on Orbit column and get the mean of Class column

ClassC = df.groupby(["Orbit"])["Class"].mean()
ClassC.plot(kind="bar", title="test")
plt.show()
```



Scatter Plot : **Orbit vs Payload Mass**

- **Flight Number vs Payload Mass**
- **Flight Number vs Launch Site**
- **Payload vs Launch Site**
- **Orbit vs Flight Number**
- **Payload vs Orbit Type**
- Shows how one variable affect the other
- 2 variables only

```
# 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("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

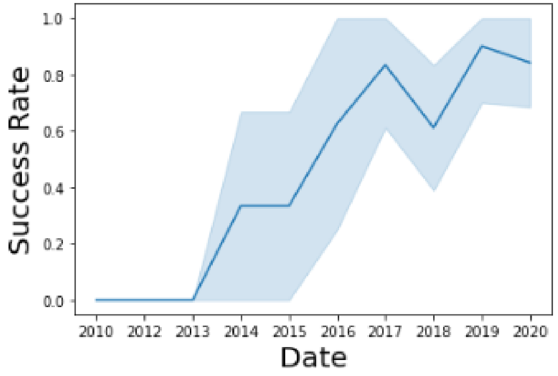


EDA with Data Visualization

Line Graph: **Success Rate vs Year**

- Shows linear trends

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
year=[]
df1 = df.copy()
year = Extract_year('date')
df1["Date"] = year
df1.head()
sns.lineplot(data=df1, x="Date", y="Class")
plt.xlabel("Date",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



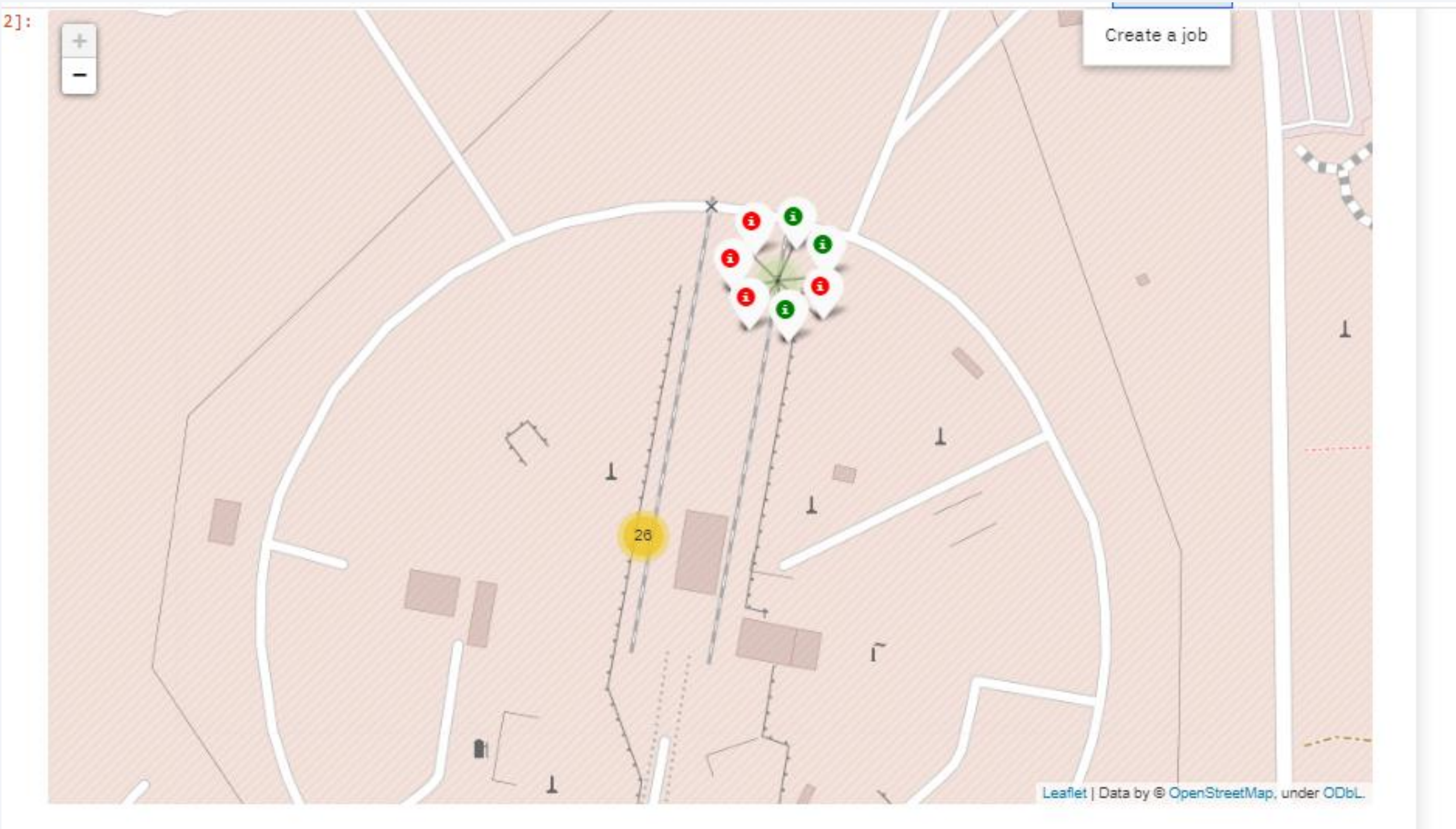
EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

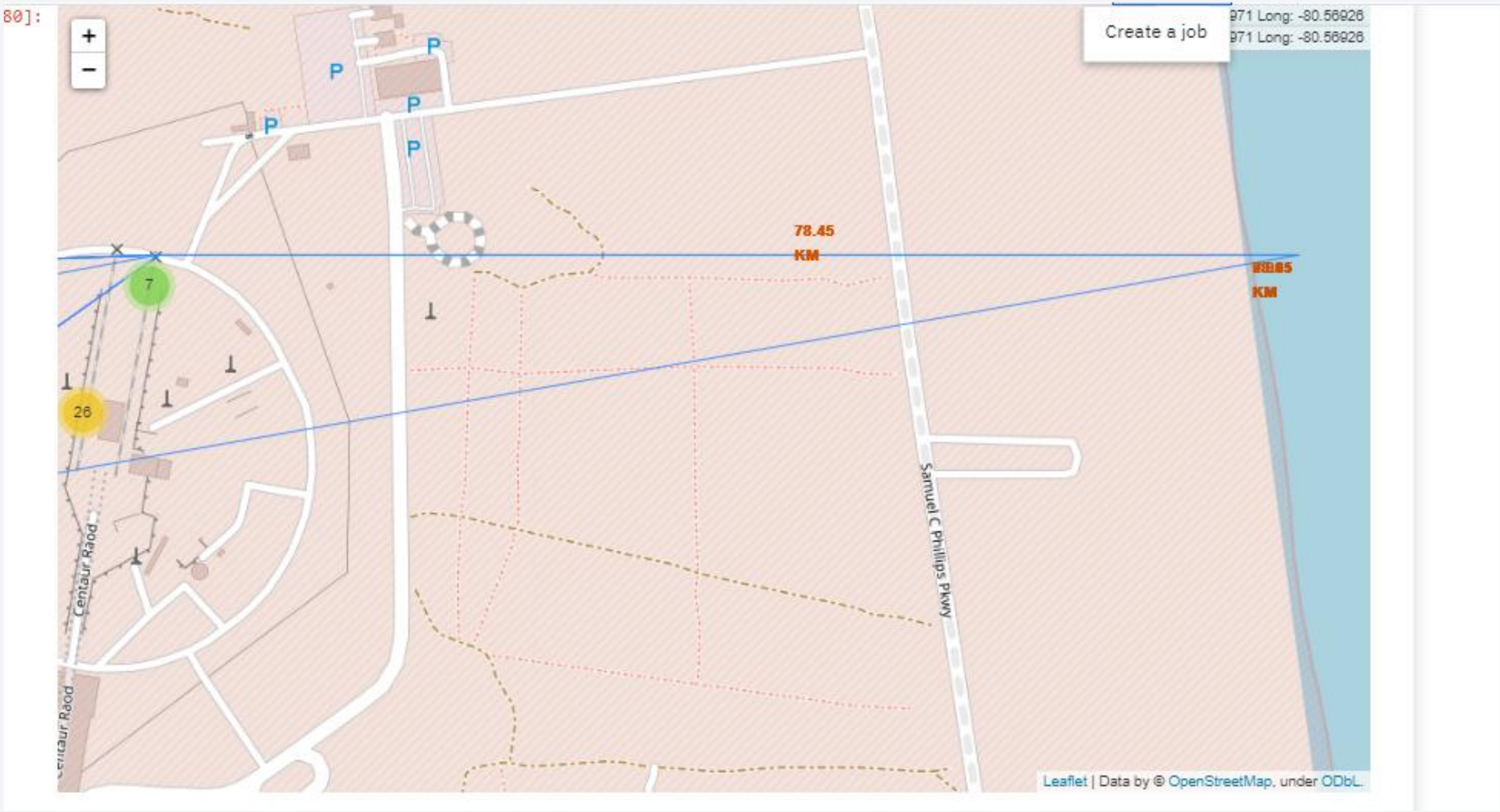
Build an Interactive Map with Folium

- add the launch site's location on a map using site's latitude and longitude coordinates
- Add a circle marker around each site
- If a launch was successful then we use a green marker and if a launch was failed, we use a red marker
- Calculate the distances between a launch site to its proximities to find trends about what is around the launch site

Build a Dashboard with Plotly Dash



Build a Dashboard with Plotly Dash



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

Results

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

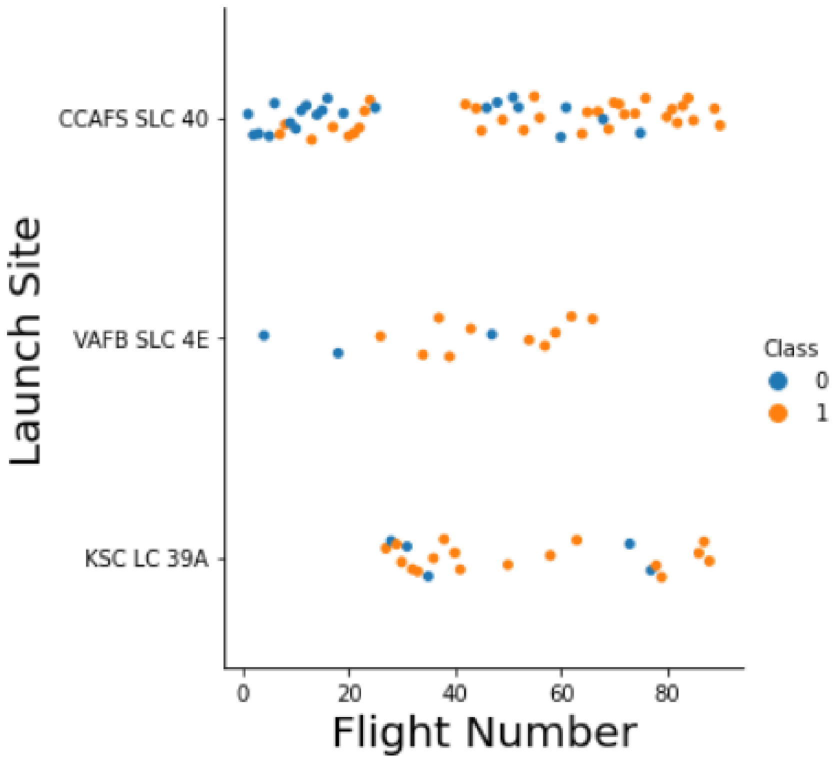
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

In [5]:

```
# 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 value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df,)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

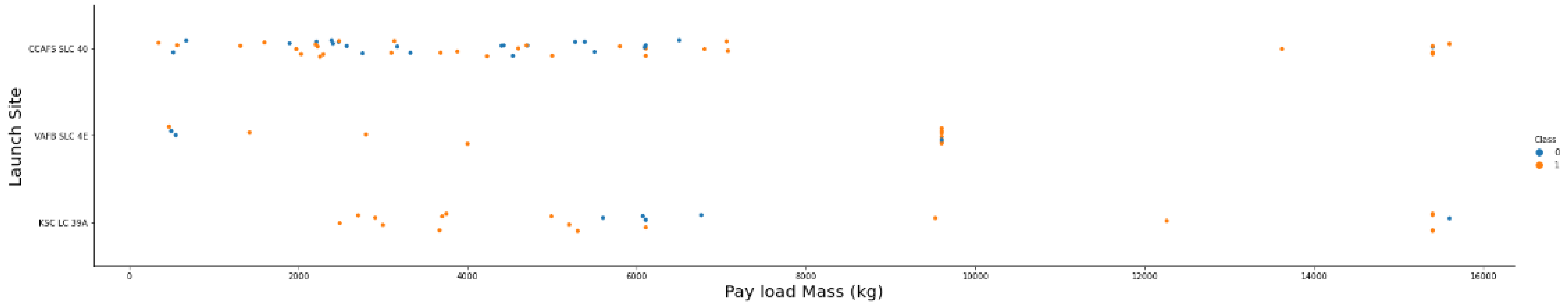


Payload vs. Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

[6]:

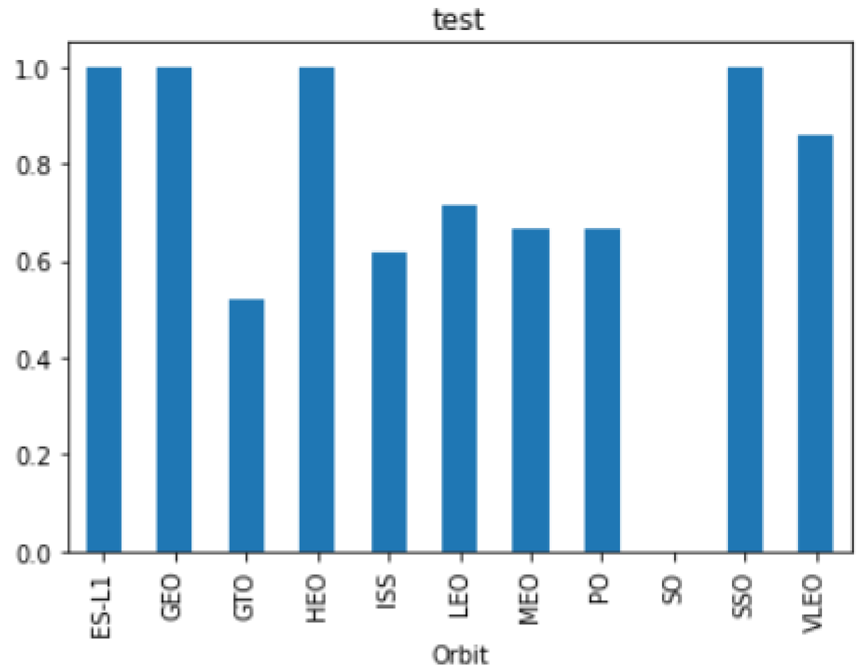
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

- Sho succ type
- Sho scat

```
ClassC = df.groupby(["Orbit"])["Class"].mean()
ClassC.plot(kind="bar", title="test")
plt.show()
```

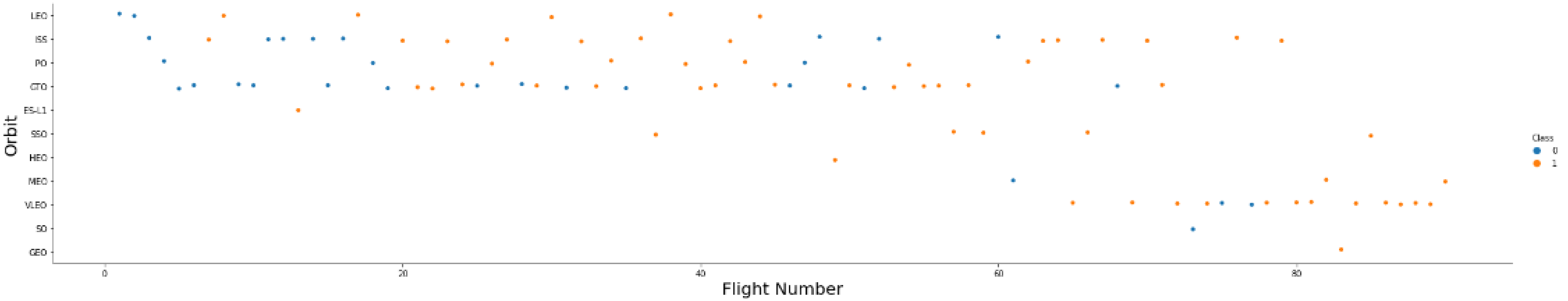


Flight Number vs. Orbit Type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

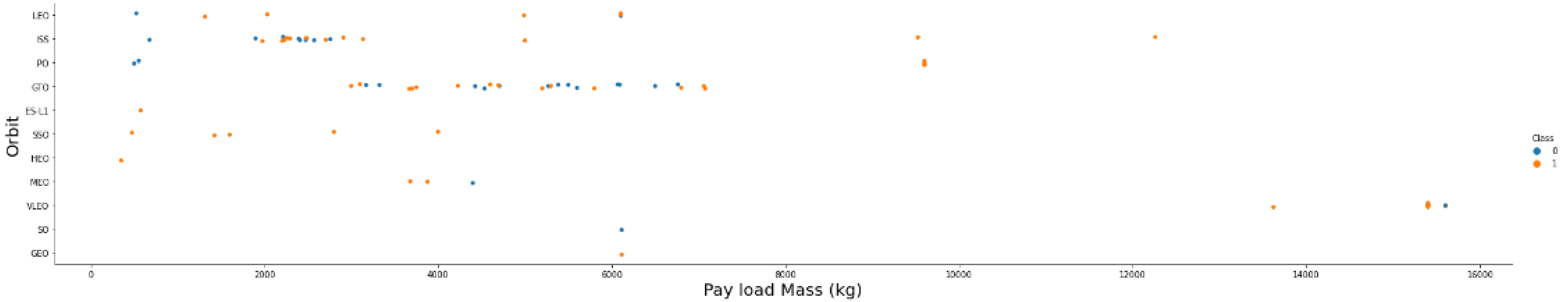
n [8]:

```
# 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()
```



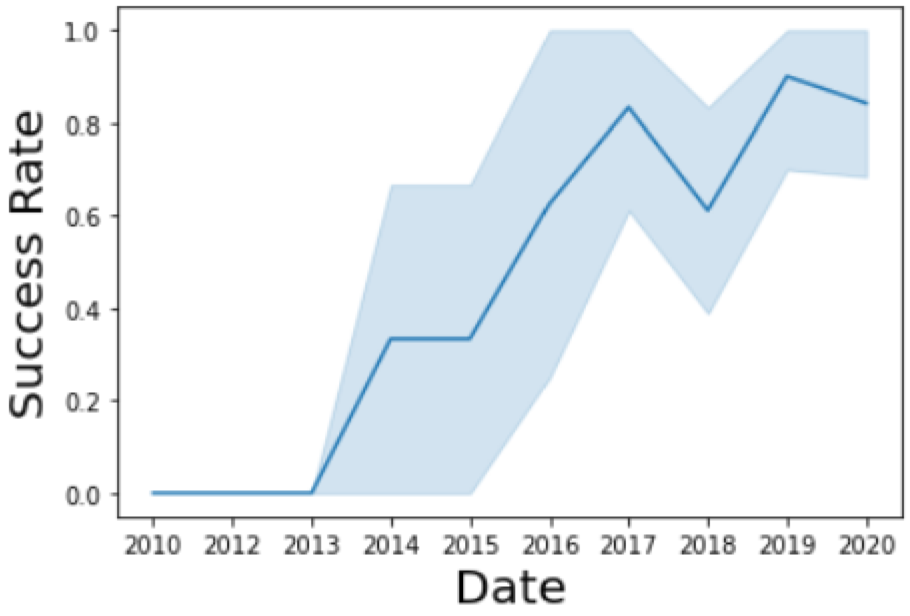
Payload vs. Orbit Type

```
]:  # 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("Pay load Mass (kg)",fontsize=20)
    plt.ylabel("Orbit",fontsize=20)
    plt.show()
```



Launch Success Yearly Trend

```
# Plot a line chart with x axis to be the extracted year and y axis t
year=[]
df1 = df.copy()
year = Extract_year('date')
df1["Date"] = year
df1.head()
sns.lineplot(data=df1, x="Date", y="Class")
plt.xlabel("Date",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

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

- Present your query result with a short explanation here

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 Launch Records

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

- Present your query result with a short explanation here

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

- 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

- Present your query result with a short explanation here

Section 4

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- Explain the important elements and findings on the screenshot

<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 5

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy

Confusion Matrix

- Tree algorithm produced the best results as it performed the best.

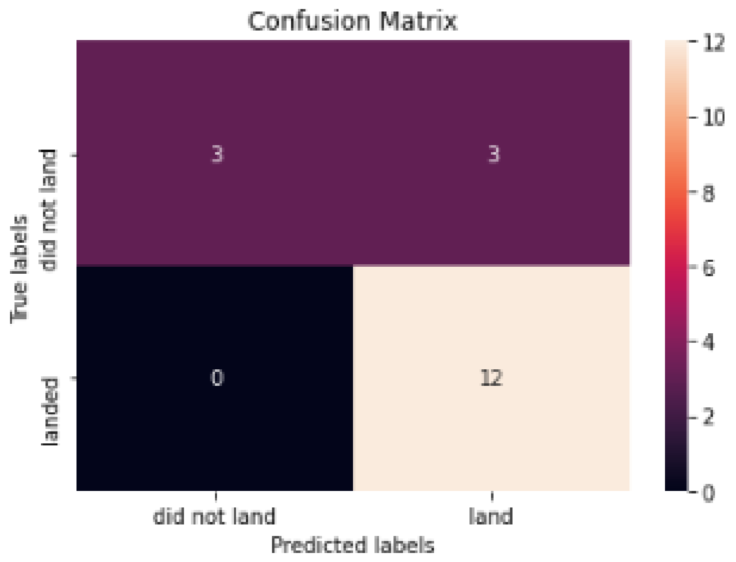
Find the method performs best:

```
In [51]: predictors = [knn_cv, svm_cv, logreg_cv, tree_cv]
         best_predictor = [knn_cv, svm_cv, logreg_cv, tree_cv]
         best_result = 0
         predictor.score(X_test, Y_test)

Out[51]: 0.7222222222222222
```

```
In [29]: tree_cv.score(X_test, Y_test)

Out[29]: 0.7222222222222222
```

We can plot the confusion matrix

```
In [30]: yhat = svm_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The tree algorithm is best for this data in Machine Learning

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!