IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniel R. Morales
4-17-2023

Outline

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

Executive Summary

- The aim of this project is to predict the success of Falcon 9 first stage landings using clean data obtained from the SpaceX API. To achieve this, I begin by performing some descriptive analyses of the data using SQL and data visualizations with Plotly and Dash. Next, I split the data into a train and test set to build and evaluate prediction models.
- One key finding is that the probability of successful launches is high, particularly for SSO, HEO, GEO, and ES-L1 orbit launches, which have a 100% success rate. This is a promising development in the ongoing quest to conquer space, as we gain insights into the factors that contribute to successful rocket launches.
- Overall, the results suggest that continued investment in future rocket launches is worthwhile, as the data indicates a high likelihood of success. By leveraging advanced analytics and predictive modeling, we can help advance the exploration and commercialization of space, and pave the way for future breakthroughs in space technology.

Introduction

- Our client has informed us that SpaceX advertises its Falcon 9 rocket launches on its website for $62 million, while other providers charge upwards of $165 million for each launch. Much of the cost savings can be attributed to SpaceX's ability to reuse the first stage of the rocket. However, a crucial factor in determining the cost of a launch is whether or not the first stage will successfully land. This information is vital if an alternate company wants to compete with SpaceX for a rocket launch.
- To achieve this, we must first collect and ensure that the data obtained from an API is in the correct format. Then, we employ data science techniques such as data analysis, data visualization, queries, and machine learning to accurately classify whether future launches or investments will be successful and worthwhile. By doing so, we can provide valuable insights for companies competing in the space industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- First
  - I requested and parsed the SpaceX launch data using the GET request using this url
    - spacex_url=https://api.spacexdata.com/v4/launches/past
    - response = requests.get(spacex_url)
    - data = pd.json_normalize(response.json())
- Second, after some manipulation and applied functions, I constructed the dataset by combining the columns into a **dictionary**
  - 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}

Data Collection

- Third
  - I set a data frame using pandas
    - df = pd.DataFrame(launch_dict)

n [26]:
```
# Show the head of the dataframe
df.head()
```

ut[26]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28

Data Collection – SpaceX API

- You can check in https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/jupyter-labs-spacex-data-collection-api.ipynb the process and specific input codes and outputs

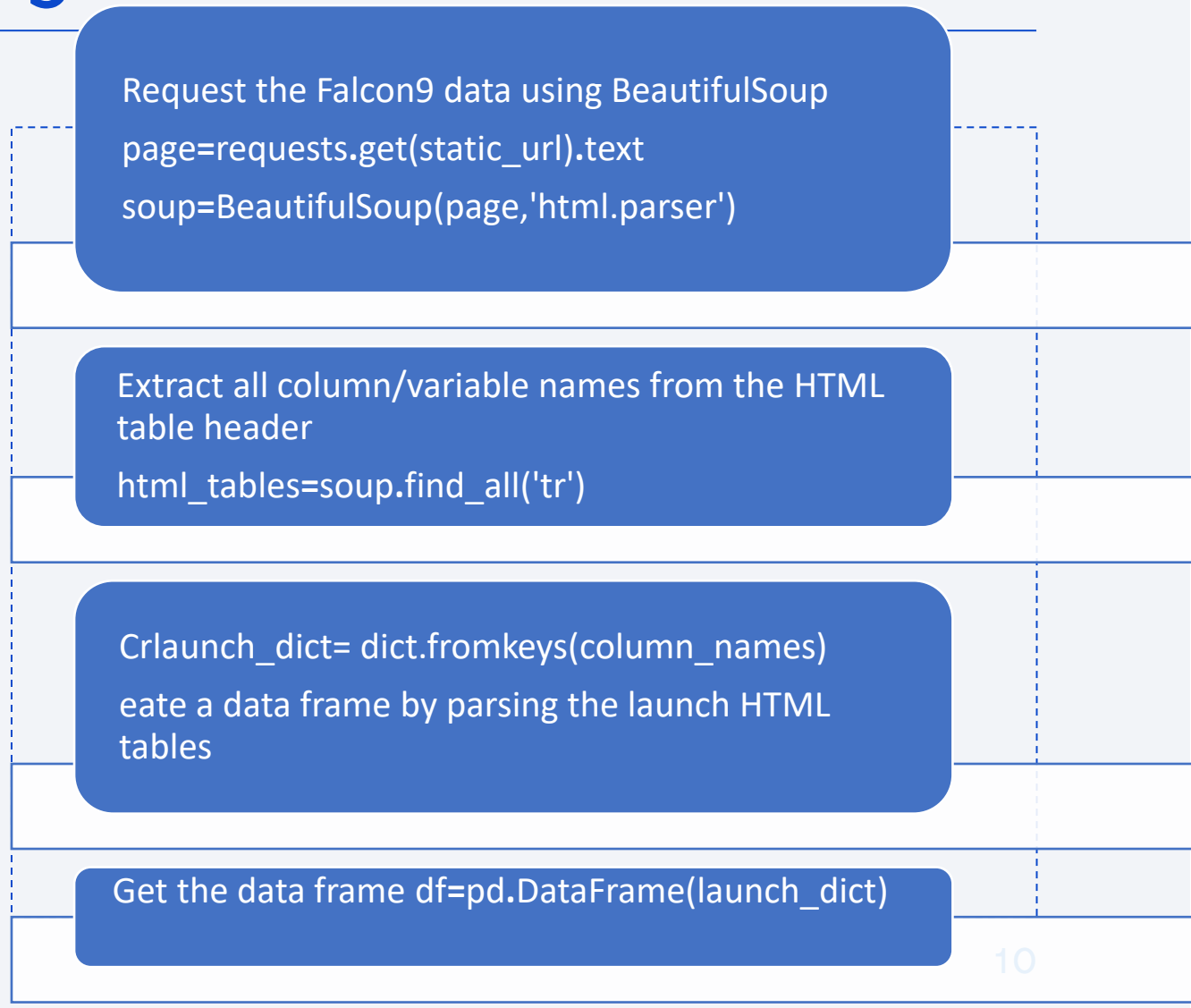
Request and parse the SpaceX launch data using the GET request

Build the dataset by combining the columns into a dictionary

Set a data frame using pandas

Data Collection - Scraping

- I used BeautifulSoup and Requests with an url with Spacex data
  - static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- You can check in https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/jupyter-labs-webscraping2.ipynb the process and specific input codes and outputs



Data Wrangling

- Data wrangling process
  - 1. filter the dataframe to only include Falcon 9 launches
  - 2. Reset the FlightNumber column
  - 3. Check for null values
  - 4. Dealing with Missing Values
- You can check in https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/jupyter-labs-spacex-data-collection-api.ipynb the process and specific input codes and outputs

Filter the dataframe to only include Falcon 9 launches

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

Reset the FlightNumber columna

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

Check for null values

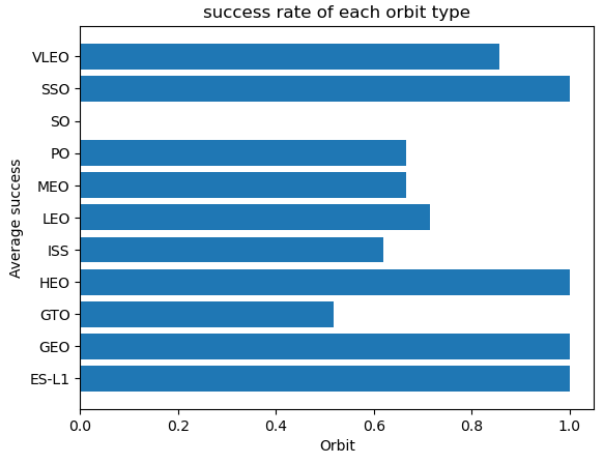
```
data_falcon9.isnull().sum()
```

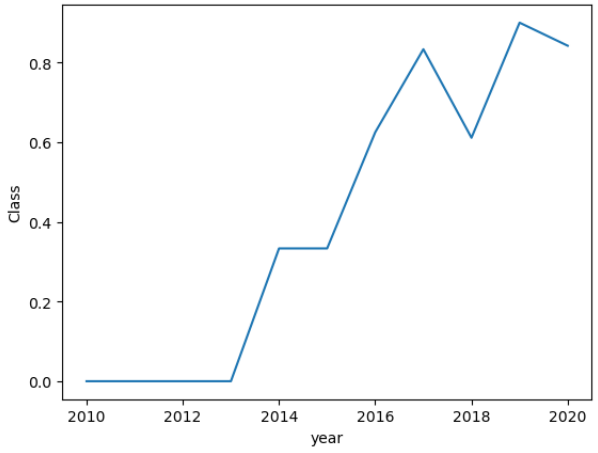
Dealing with Missing Values

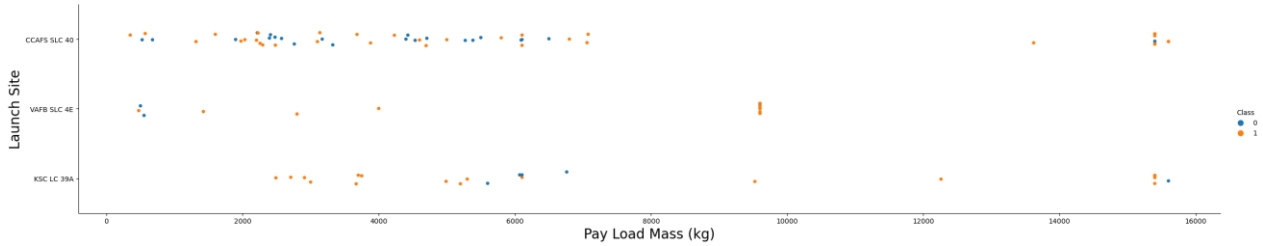
```
mean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].replace(np.nan, mean, inplace=True)
```

EDA with Data Visualization

- The rate of success has been increasing over time
- There are differences in the rate of success depending on orbit type
- In location VAFB SLC 4E there is a clear positive relation between success and Pay Load Mass
- Check in https://github.com/danielmoralesr/CapstopneIBMProject/blob/master/jupyter-labs-eda-dataviz2.ipynb







EDA with SQL

- Using Jupyter Notebook, and after I connect with IBM Cloud using this code %sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=SSL, I performed several queries
  - To display the names of the unique launch sites in the space mission
    - %sql SELECT UNIQUE(Launch_Site) FROM Spacex;
  - To display 5 records where launch sites begin with the string 'CCA'
    - %sql SELECT * FROM Spacex WHERE launch_site LIKE '%CCA%' LIMIT 5;
  - And so on. I add some output to the appendix.
- You can check https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/jupyter-labs-eda-sql-coursera2.ipynb, but additional properties are not allowed ('id' was unexpected)

Build an Interactive Map with Folium

- To build and interactive map
  - First, I map Nasa Jonson Space Center with
    - marker = folium.map.Marker(nasa_coordinate)
  - Second, I map each launch site using a marker_cluster
  - Third, I calculate and map the distance between a launch site to its proximities using PolyLine

- I added those objects in order to check the properties of the launch sites
- https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The dashboard has a dropdown menu with
  - a site-dropdown to plot a pie chart and
  - a payload-slider to display a scatter plot
- I added those plots and interactions in order to check visual success rate by launch sites and paylod mass range
- https://github.com/danielmoralesr/CapstopnelBMProject/blob/master/jupyter_app_cap10_dmr.ipynb

Predictive Analysis (Classification)

- First, I transformed features matrix X and response vector Y into a numpy arrays. Also I transformed X with SatndardScaler
- I used the function train_test_split to split the data X and Y into training and test data.
- I instantiated LogisticRegression, SVM, DecisionTreeClassifier, KNeiborsClasifier.
- For each model, I fitted the training data, and predict paramns and scores.
- I computed confusion matrices and compere the models performance
- https://github.com/danielmoralesr/CapstopneIBMProject/blob/master/alternative_week4_jupyternotebook_pdf.pdf
  - Note: I had to upload in pdf because https://github.com/danielmoralesr/CapstopneIBMProject/blob/master/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb says Invalid Notebook, Additional properties are not allowed ('id' was unexpected), Using nbformat v5.8.0 and nbconvert v7.2.7

Transform X and Y into a numpy arrays, and X with SatndardScaler

Apply the function train_test_split to split the data X and Y into training and test data.

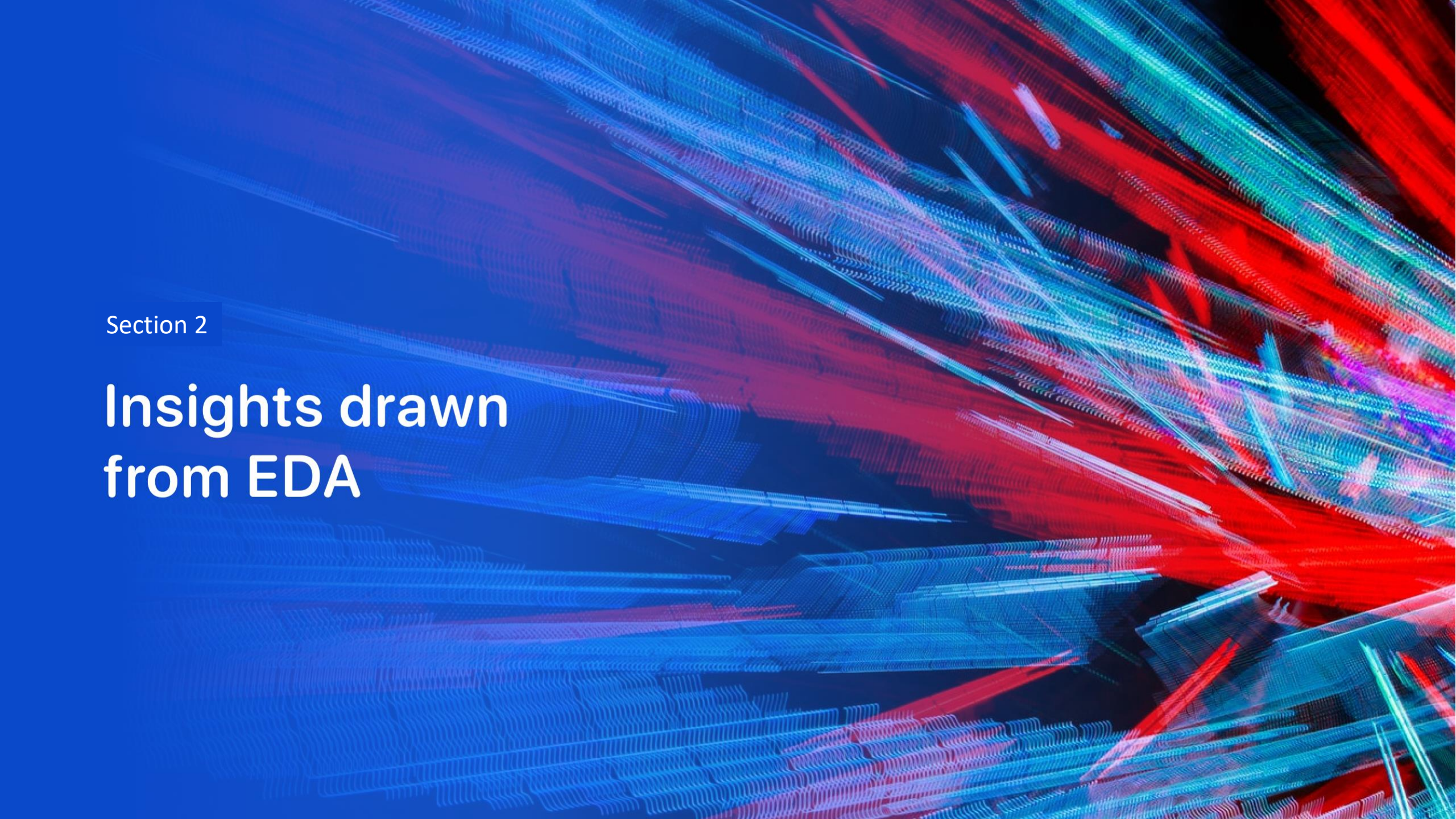
Instantiat LogisticRegression, SVM, DecisionTreeClassifier, KNeiborsClasifier models

For each model, fit the training data, and predict params and scores

Compute confusion matrices and compere the models performance

Results

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


Section 2
Insights drawn from EDA

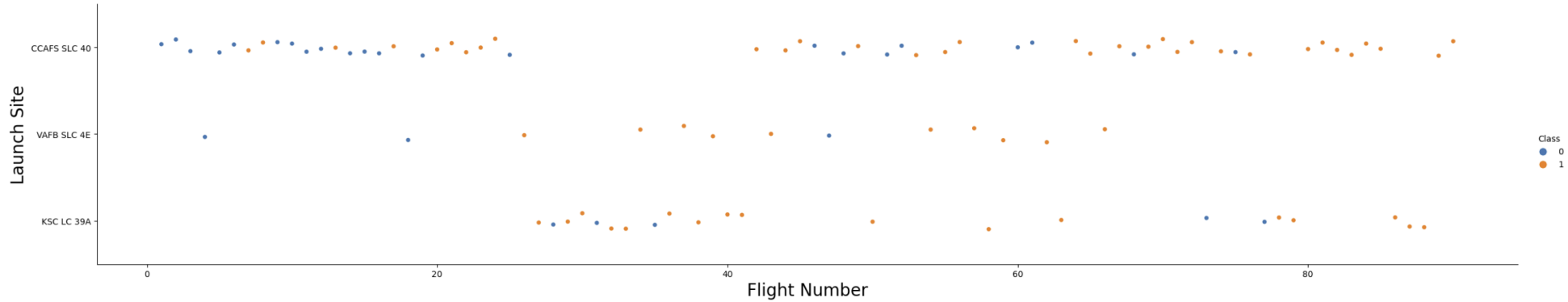
Flight Number vs. Launch Site

- The scatter plot presents the distribution of successful and unsuccessful launches for each Launch Site.
- It can be seen that the Flight Number is positively related to the prevalence of success

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

[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 value
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()
```



Payload vs. Launch Site

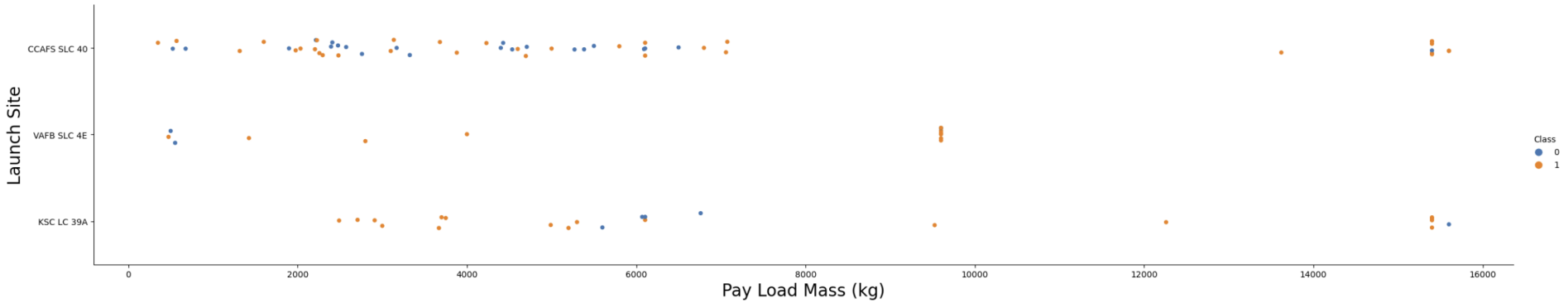
- This scatter plot shows that for the Launch Site VAFB, the precedence of successful launches is clearly related to Pay Load Mass.

TASK 2: Visualize the relationship between Payload and Launch Site

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

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 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

- There is a clear 100% launch success rate for SSO, HEO, GEO and ES-L1 orbit launches.
- Then with an 80% success rate the launches with VLEO orbit.
- Otherwise, the other orbits have a probability of success of less than 70%.

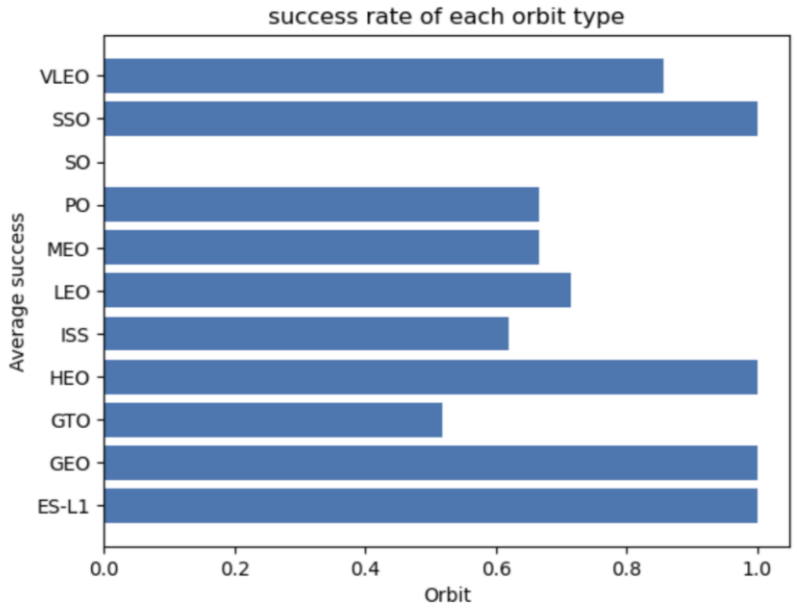
TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

In [8]:

```
# HINT use groupby method on Orbit column and get the mean of Class column
grouped_data = df.groupby('Orbit')['Class'].mean()
plt.barh(grouped_data.index, grouped_data.values)
plt.title('success rate of each orbit type')
plt.xlabel('Orbit')
plt.ylabel('Average success')
plt.show()
```



Flight Number vs. Orbit Type

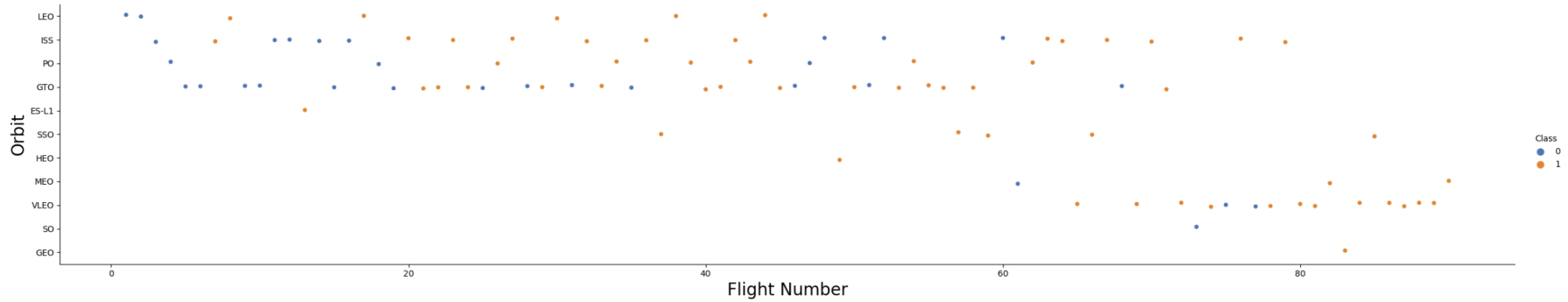
- Only for the SSO orbit is there a strong relationship with the flight number.
- Also for the LE orbit, although the first two flight numbers were unsuccessful.

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and 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()
```



Payload vs. Orbit Type

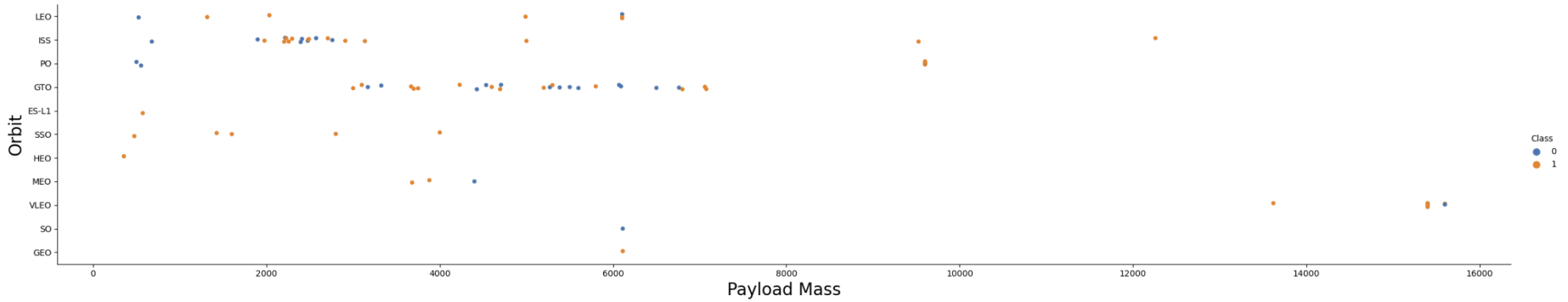
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

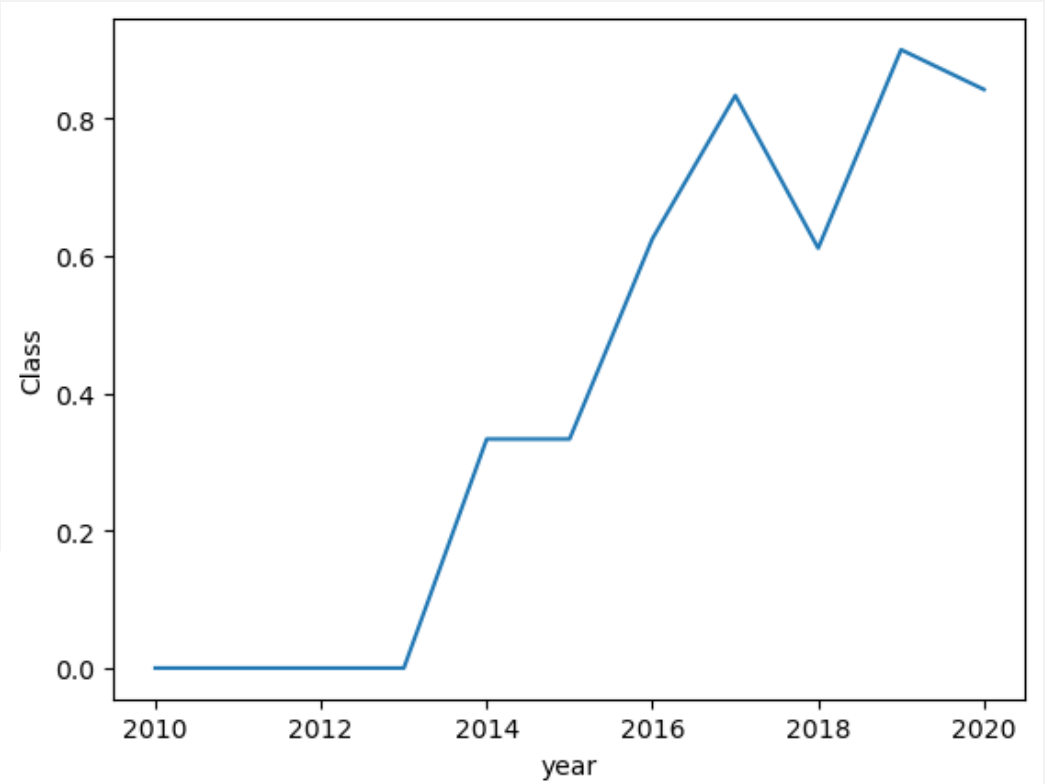
[n [10]]:

```
# 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("Payload Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- Query
  - %sql SELECT UNIQUE(Launch_Site) FROM Spacex;
- This query display the unique launch sites

Task 1

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

```
In [5]: %sql SELECT UNIQUE(Launch_Site) FROM Spacex;
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

- Using this query %sql SELECT * FROM Spacex WHERE launch_site LIKE '%CCA%' LIMIT 5;
- I can display 5 records where launch sites begin with the string 'CCA'

Task 2

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

In [13]:
```
%sql SELECT * FROM Spacex WHERE launch_site LIKE '%CCA%' LIMIT 5;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[13]:

DATE	time_utc	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA is 45596
- It was query using this code
  - %sql SELECT SUM(payload_mass_kg_) FROM Spacex WHERE customer = 'NASA (CRS)';

Task 3

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

In [14]:
```
%sql SELECT SUM(payload_mass_kg_) FROM Spacex WHERE customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[14]:

1
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1is 2534
- It was query using this code
  - %sql SELECT AVG(payload_mass_kg_) FROM Spacex WHERE booster_version LIKE '%F9 v1.1%';

Task 4

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

```
In [20]: %sql SELECT AVG(payload_mass_kg_) FROM Spacex WHERE booster_version LIKE '%F9 v1.1%';
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[20]:

1
2534

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad was 2015-12-22
- It was calculated using this code %sql SELECT MIN(DATE) FROM Spacex WHERE landing_outcome='Success (ground pad)';

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

Hint:Use min function

In [22]:
```
%sql SELECT UNIQUE(landing_outcome) FROM Spacex;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[22]:

landing_outcome
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

In [24]:
```
%sql SELECT MIN(DATE) FROM Spacex WHERE landing_outcome='Success (ground pad)';
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[24]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are in the picture bellow
- They were obtenined using this code
  - %sql SELECT payload FROM Spacex WHERE landing_outcome = 'Success (drone ship)' AND payload_mass_kg_ > 4000 AND payload_mass_kg_ < 6000;

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [25]:
```
%sql SELECT payload FROM Spacex WHERE landing_outcome = 'Success (drone ship)' AND payload_mass_kg_ > 4000 AND payload_mass_k
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[25]:

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes are in the picture bellow
- The query that was used to obtain was %sql SELECT mission_outcome, COUNT(*) as count FROM Spacex GROUP BY mission_outcome;

Task 7

List the total number of successful and failure mission outcomes

In [26]:
```
%sql SELECT UNIQUE(mission_outcome) FROM Spacex;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[26]:

mission_outcome
Failure (in flight)
Success
Success (payload status unclear)

In [29]:
```
%sql SELECT mission_outcome, COUNT(*) as count FROM Spacex GROUP BY mission_outcome;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[29]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass are bellow
- It was obtained using this code %sql SELECT booster_version, payload From Spacex WHERE payload_mass_kg_ = (SELECT MAX(payload_mass_kg_) FROM Spacex);

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [34]:
```
%sql SELECT booster_version, payload From Spacex WHERE payload_mass_kg_ = (SELECT MAX(payload_mass_kg_) FROM Spacex);
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[34]:

booster_version	payload
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21

2015 Launch Records

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 is bellow
- The query that was used is
  - %%sql SELECT landing_outcome, booster_version, launch_site
  - FROM Spacex
  - WHERE landing_outcome LIKE '%Failure%' AND landing_outcome LIKE '%drone ship%' AND YEAR(DATE)=2015;

Task 9

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

In [48]:
```
%%sql
SELECT landing_outcome, booster_version, launch_site
FROM Spacex
WHERE landing_outcome LIKE '%Failure%' AND landing_outcome LIKE '%drone ship%' AND YEAR(DATE)=2015;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[48]:

landing_outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Task 10

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

In [49]:
```
%%sql
SELECT landing_outcome, COUNT(*) as count
FROM Spacex
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing_outcome
ORDER BY count DESC;
```

```
 * ibm_db_sa://svh12272:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/blu
Done.
```

Out[49]:

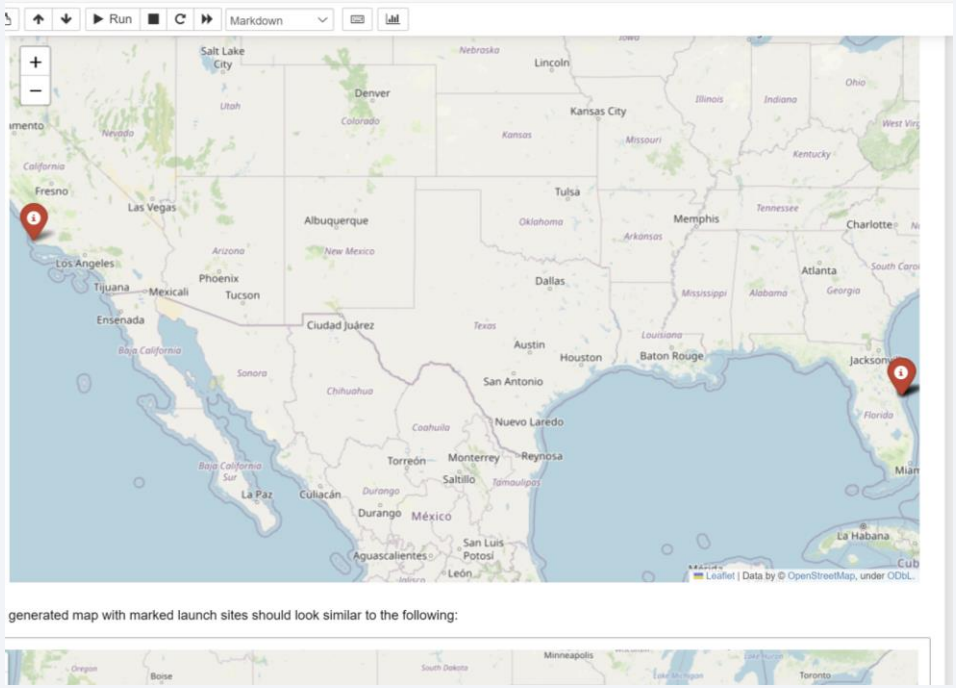
landing_outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

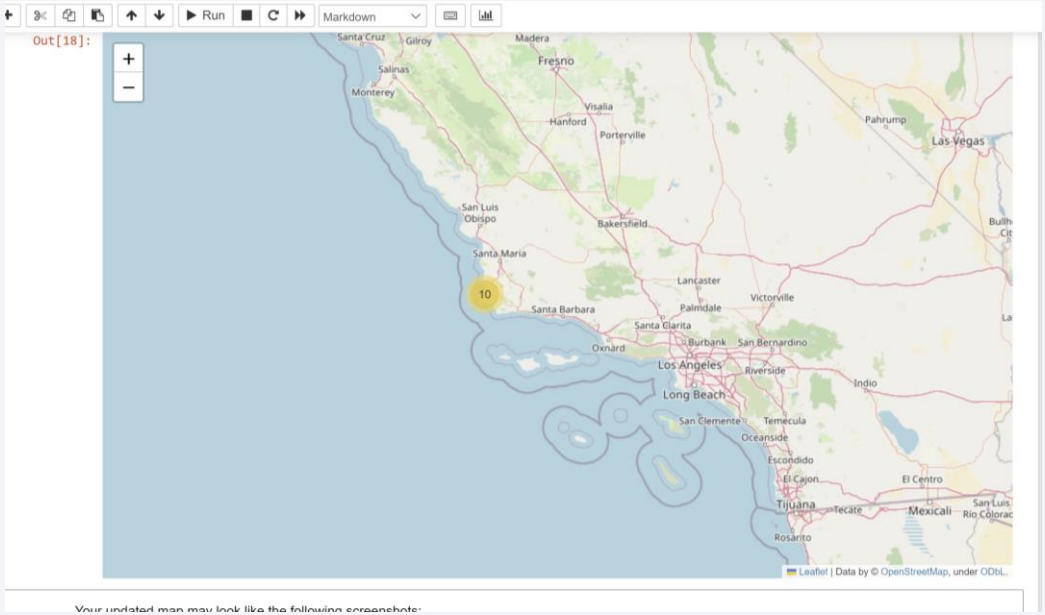
Lauch Sites

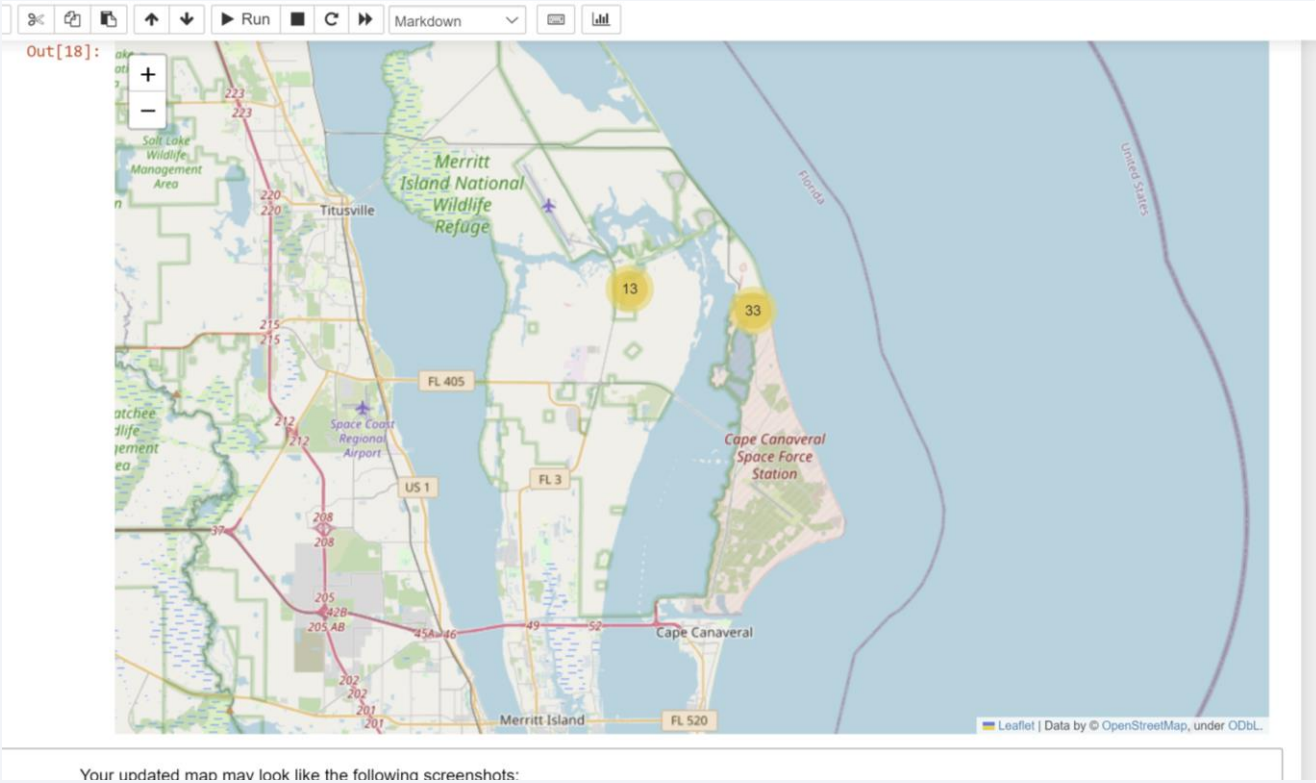
- You can see the launch sites that SpaceX used to launch their rockets



Number of successful launch per location

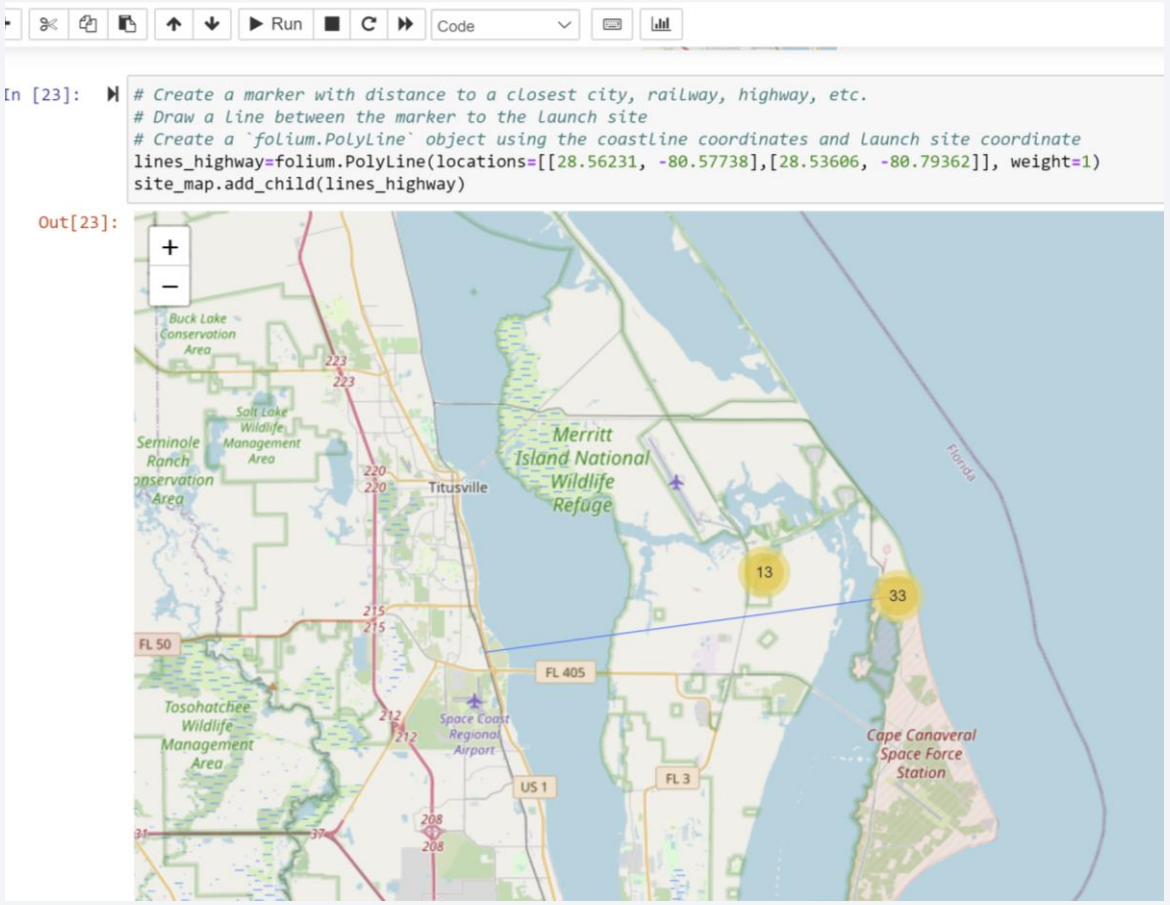
- In this graph you can appreciate geographically the success launches per site

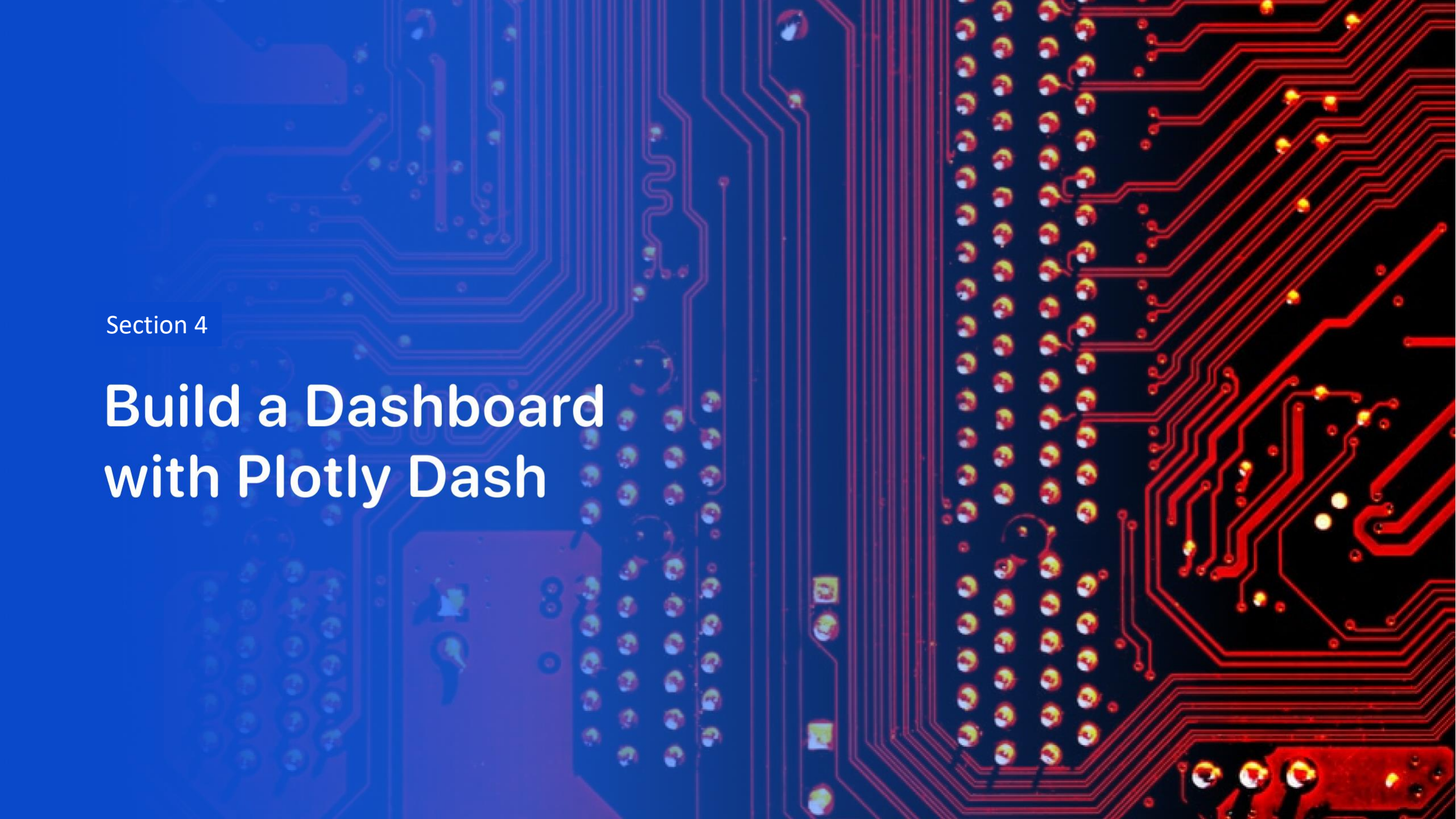




Distance to proximities

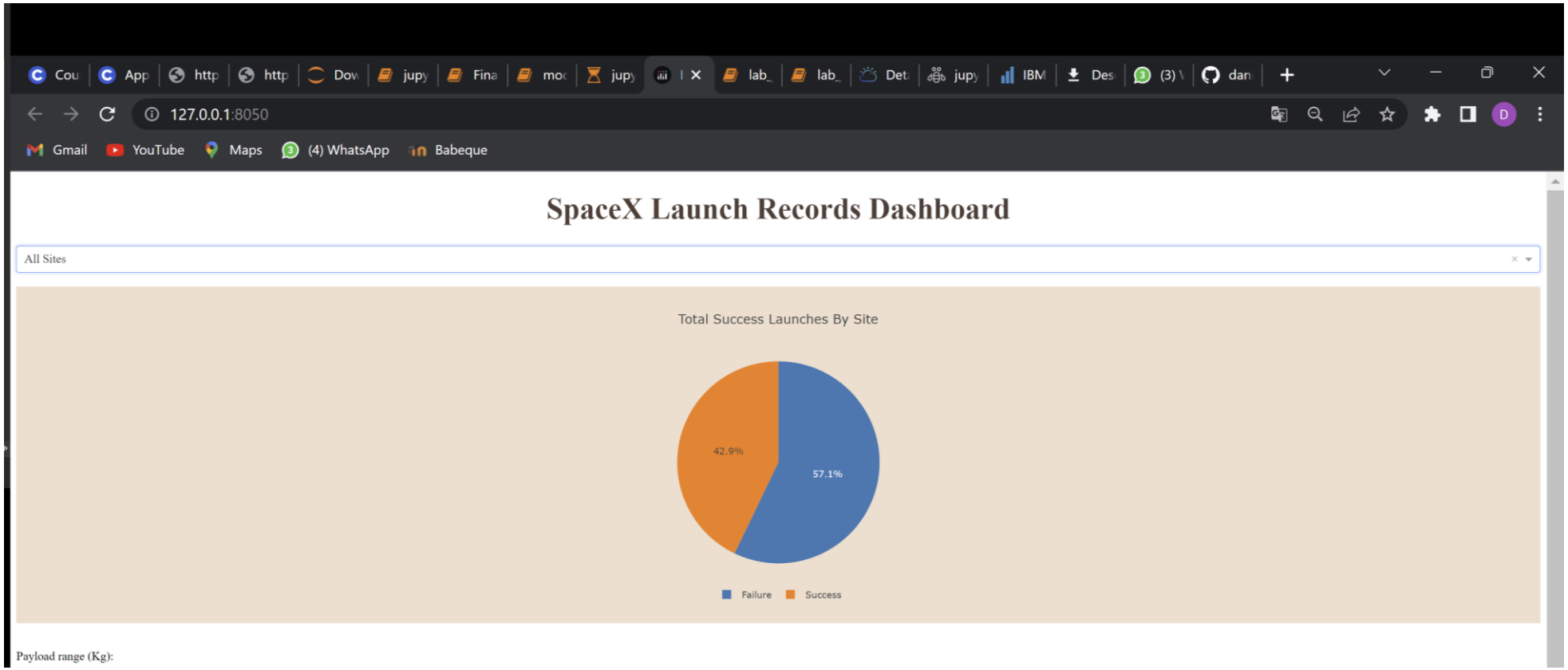
- The graph shows the distance to proximities between launch sites and railroads and so on.




Section 4
Build a Dashboard with Plotly Dash

Total Success Launches By Site

- The pie chart shows that for all sites the success rate is 57.1.
- You can drop-down bay site



Booster Version Category

- Depending on the Payload range, you can plot Booster Version Category by class

Booster Version Category by Payload range

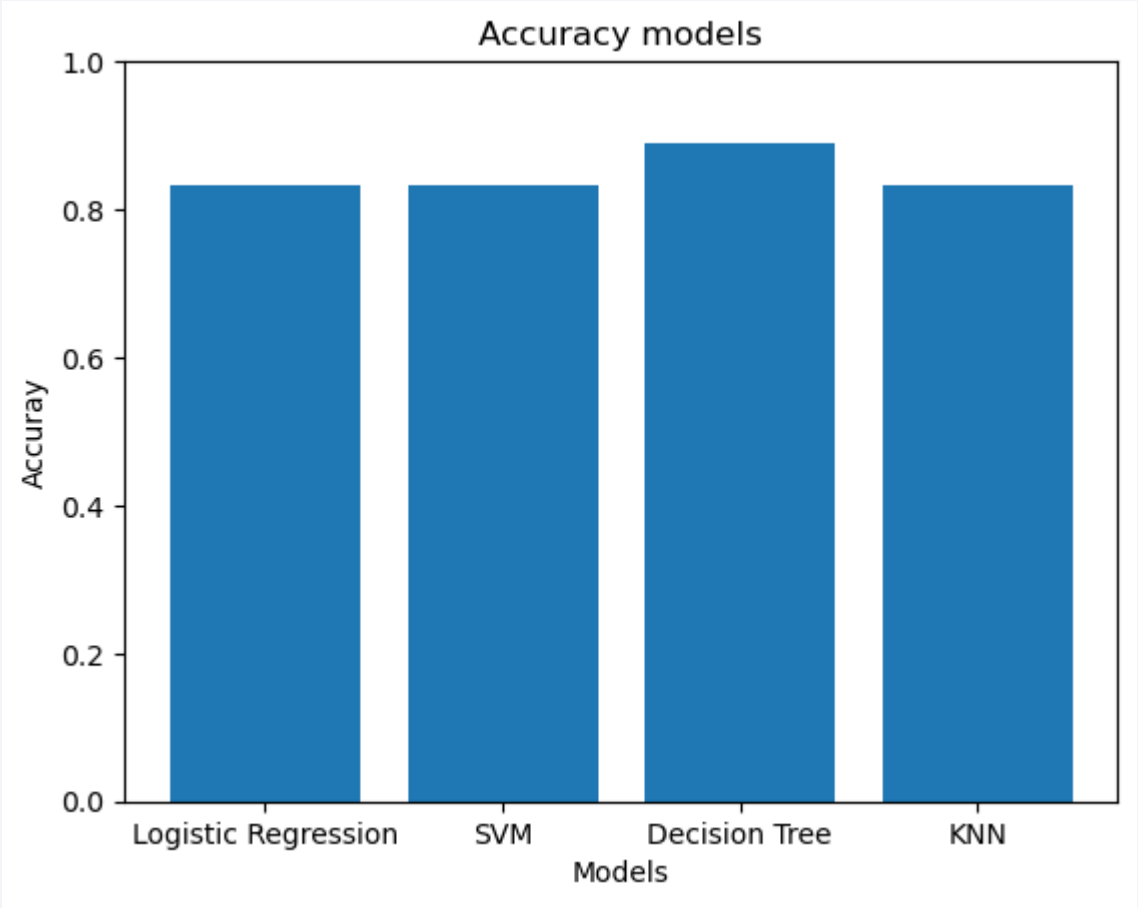
- You can select the payload range and see the booster version by class

Section 5

Predictive Analysis (Classification)

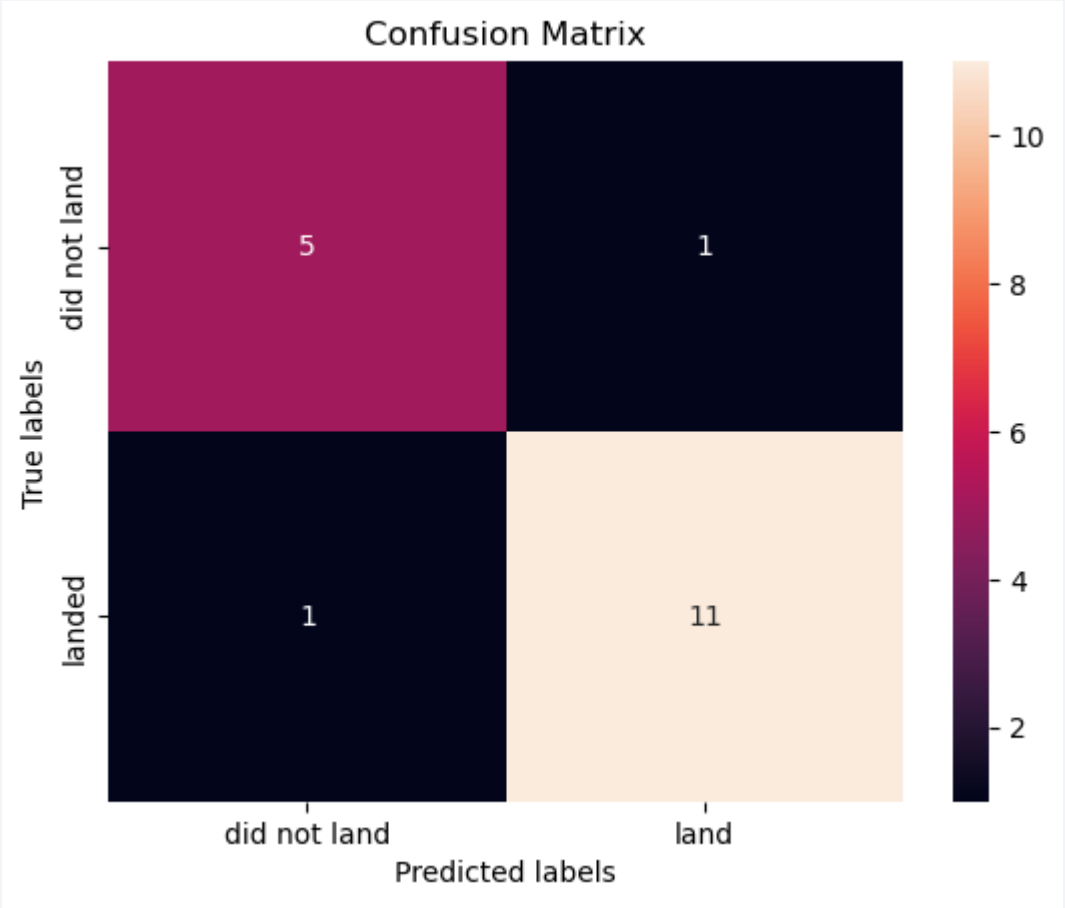
Classification Accuracy

- Accuracy of tested models
  - Logistic Regression Accuracy: 0.833333333333334
  - SVM Accuracy: 0.8333333333333334
  - Decision Tree Accuracy: 0.888888888888888
  - K-NN Accuracy: 0.8333333333333334



Confusion Matrix

- The confusion matrix of the best-performing model was the decision tree classifier
- With this model, the accuracy value was the highest
  - Just two records were miss classified and 88.88% of the testing sample were classified

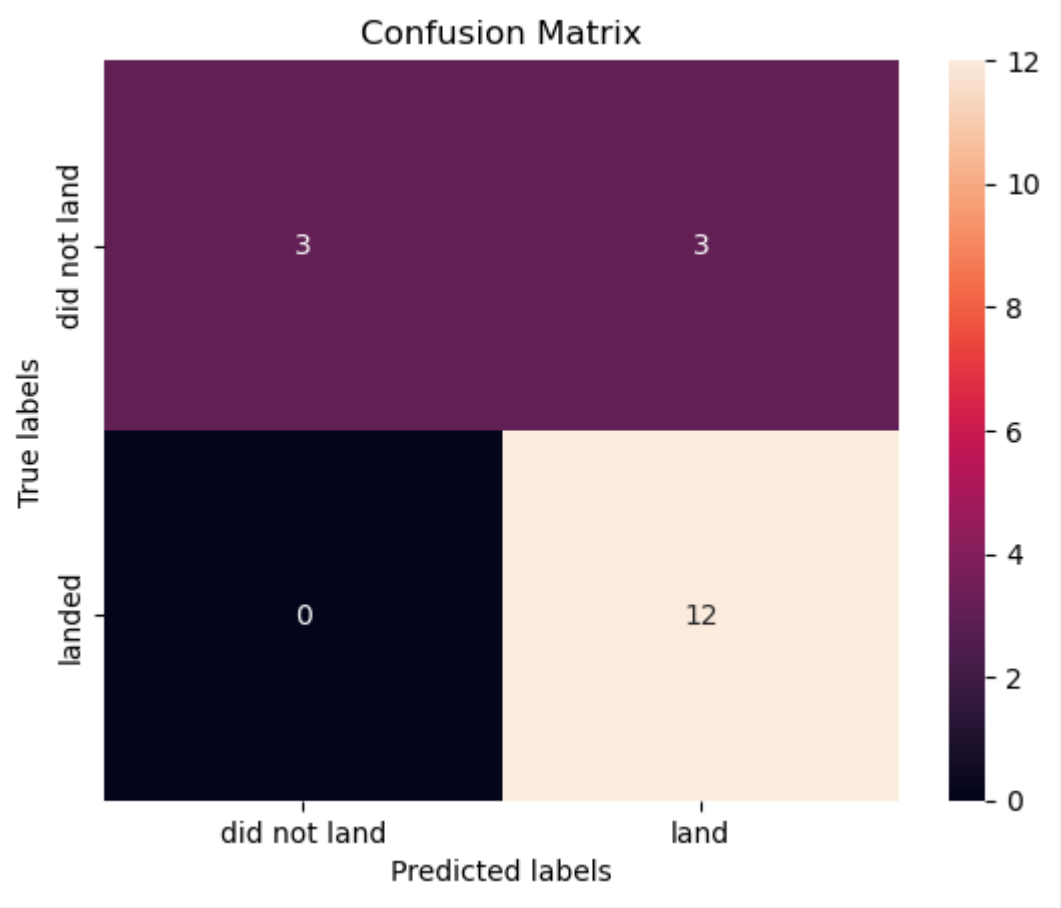


Conclusions

- Future rocket launches likely to be successful
- There is a clear 100% launch success rate for SSO, HEO, GEO and ES-L1 orbit launches.
- The race to conquer space is reaching an interesting point as we learn more about the factors that make a successful rocket launch.Point 4
- Over all, it is worth continuing to invest in future rocket launches

Appendix

- Confusion Matrix of Knn nearest neighbor



Thank you!