IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Eduardo Pessina>
<2022-05-24>

Outline

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

Executive Summary

- Summary of methodologies
- I try to use a data driven approach to find and predict whether a particular rocket will successfully re-land after its launch. I try different machine learning models as LogisticRegression, SVC, Decision Tree, KNN and concluded that decision tree best suits for this application as I show you in the results.

- Summary of all results
- The Decision Tree gave training accuracy of 0.89 while giving a test accuracy of 0.88

Introduction

- Project background and context
- SpaceX is the first company to reuse the first stage for future launches.
- By using the historical data of SpaceX‘s rocket launches I try to find if a future rocket’s first stage can re-land safely using Artificial Intelligence.
- The main objective of the Project is to come up with a model that takes in future rocket launches initial data and predicts if it lands safely back to earth. Using this model, I can decide how Much we need to invest in future launches.

- Problems you want to find answers
- Predicts the success of a future rockets re-landing

Section 1

Methodology

Methodology

Executive Summary

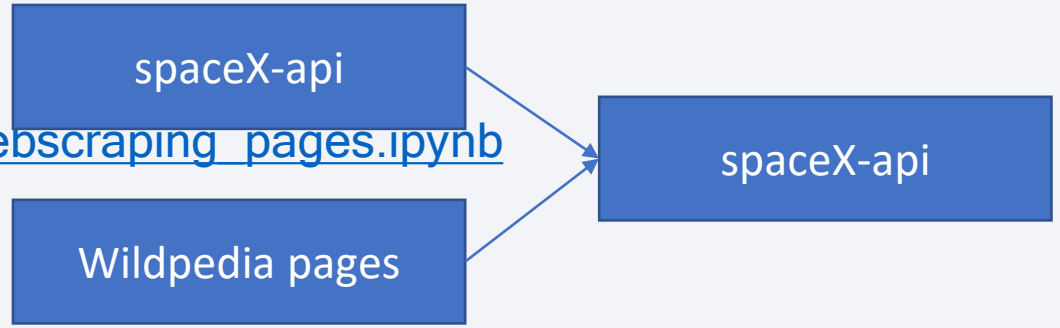
- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- This is the main phase in any Data Science task.
- We used two approaches for the data collection.

1. Input : spacex-api-data-collection.ipynb : https://github.com/eduofpess/ibm/blob/main/spacex-api-data-collection.ipynb

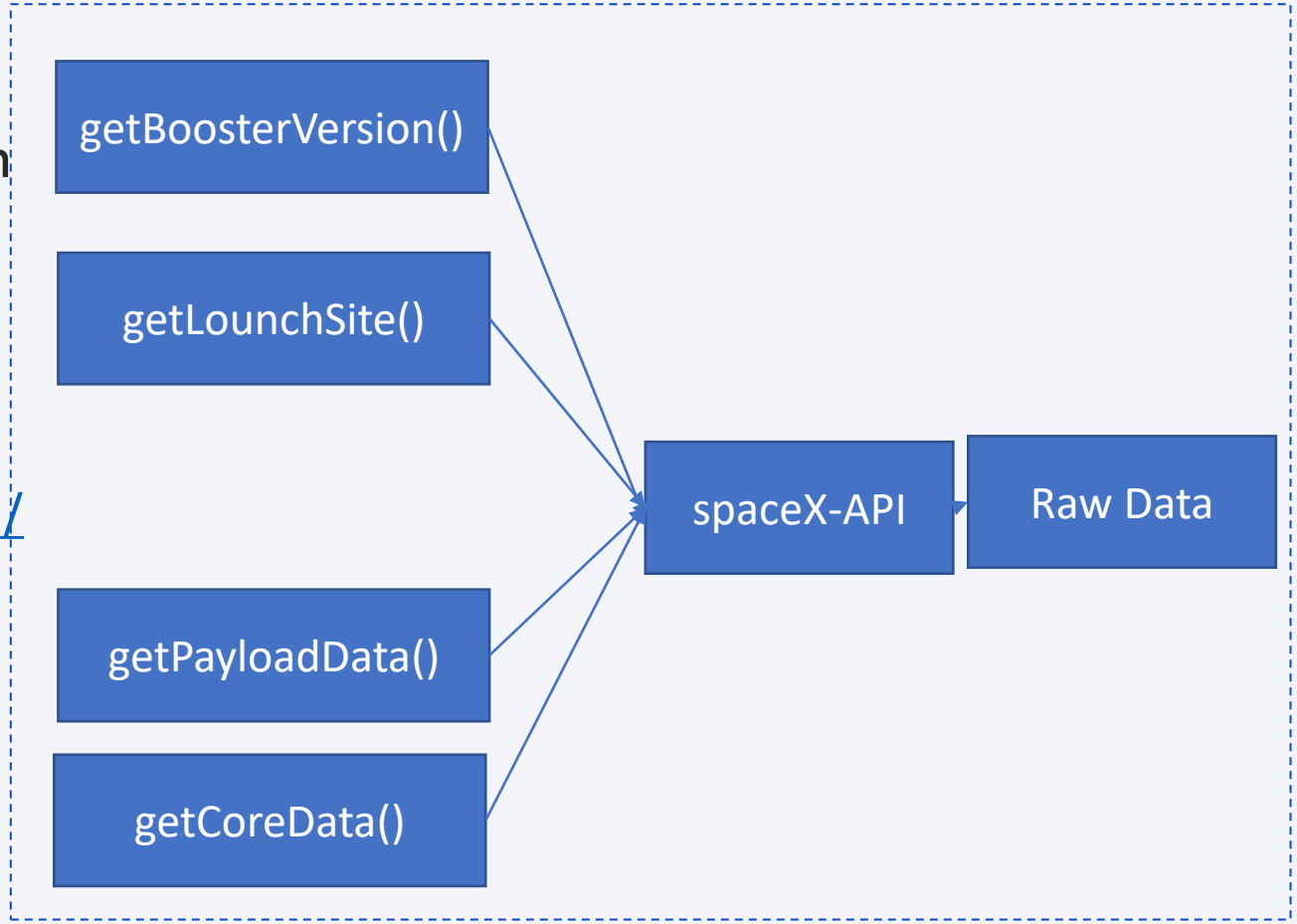
2. Input : webscraping_pages.ipynb : https://github.com/eduofpess/ibm/blob/main/webscraping_pages.ipynb



3. Output : Raw Data

Data Collection – SpaceX API

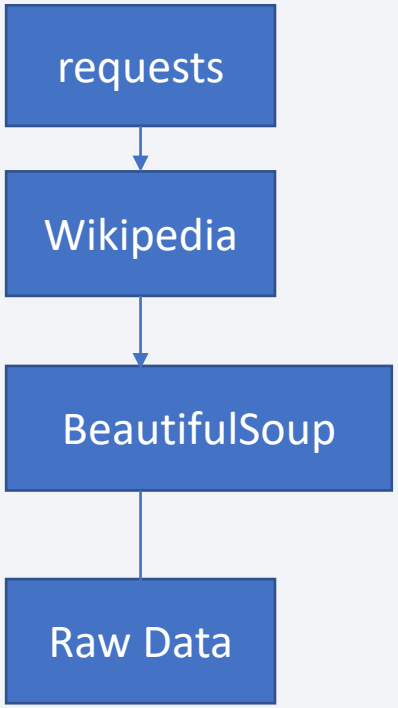
- SpaceX's API : collect booster versions, payload, core and launch site (payload id, launch id as parameters for my API)
- spacex-api-data-collection.ipynb:
- https://github.com/eduofpess/ibm/blob/main/spacex-api-data-collection.ipynb



Data Collection - Scraping

- Wikipedia tables to collect some data abaou SpaceX's previous lanches

- webscraping_pages.ipynb :
- https://github.com/eduofpess/ibm/blob/main/webscraping_pages.ipynb



Data Wrangling

- ETL data to machine readable;
- Missing data in PayloadMass column and replace with mean();
- Replaced (false,none) in landing_outcomes to (0,1)
- data-wrangling.ipynb:
- https://github.com/eduofpess/ibm/blob/main/data-wrangling.ipynb

EDA with Data Visualization

- I use those plots to analyse the data
  - Bar chart to find success rate fo different orbits;
  - Scatter plot of flight_numbers, launch_site and payload in orbit;
  - Line plot for yearly success rate of total launches;
  - Cat plot of flight_numbers, payload mass;

  - eda-data_visualization.ipynb:
  - https://github.com/eduofpess/ibm/blob/main/eda-data_visialization.ipynb

EDA with SQL

- find unique launch sites:

Select UNIQUE(LAUNCH_SITE) from SPACEXDATASET

- print first 5 records with launch site starting with‘CCA’:

Select*fromSPACEXDATASETWHERELAUNCH_SITELIKE‘CCA%’LIMIT5

- find avgpayload masscarried by booster F9:

Select AVG(PAYLOAD_MASSKG_) from SPACEXDATASET WHERE BOOSTER_VERSION=‘F9v1.1’

- ...
- eda-sql.ipynb :
- https://github.com/eduofpess/ibm/blob/main/eda-sql.ipynb

Build an Interactive Map with Folium

- Plotted launch sites available in our dataset based on lat, long;
- Used markers_cluster to mark ass success/failed launces in each site;
- Find the distance of those sites from cities, sea shores , …
- Find if a launch be successful based on their proximities

- interative_map_folium.ipynb:
- https://github.com/eduofpess/ibm/blob/main/interative_map_folium.ipynb

Build a Dashboard with Plotly Dash

- Based on launch site and payload mass;
- Plotted a pie chart of success rate on launch site;
- Plotted scatter plot payload effects th success rate of a launch

- dash.ipynb:
- https://github.com/eduofpess/ibm/blob/main/dash.ipynb

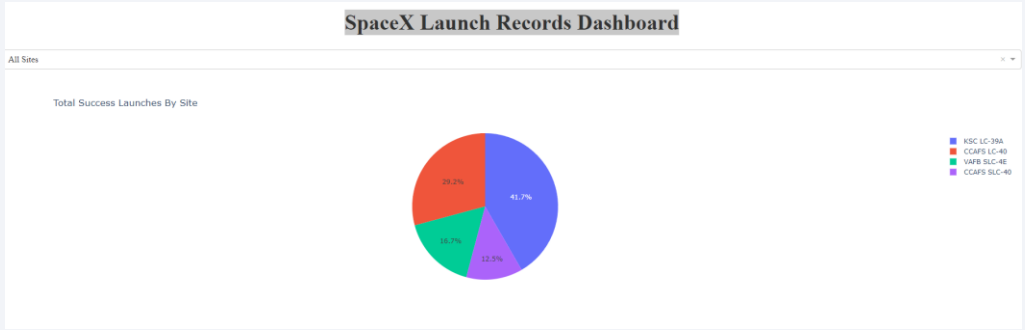
Predictive Analysis (Classification)

- logistic regression,SVC,DecisionTree,KNNasourmodels;
- Load the data, perform some standardization.
- Divide data in trainn and test sets.
- Use training set to iteratively find the best parameters for that data and finalized the parameters.
- Find the best model of the four modes based on test set performance.

- classification-predictive_analysis.ipynb:
- https://github.com/eduofpess/ibm/blob/main/classification-predictive_analysis.ipynb
- https://github.com/eduofpess/ibm/blob/main/prediction.ipynb

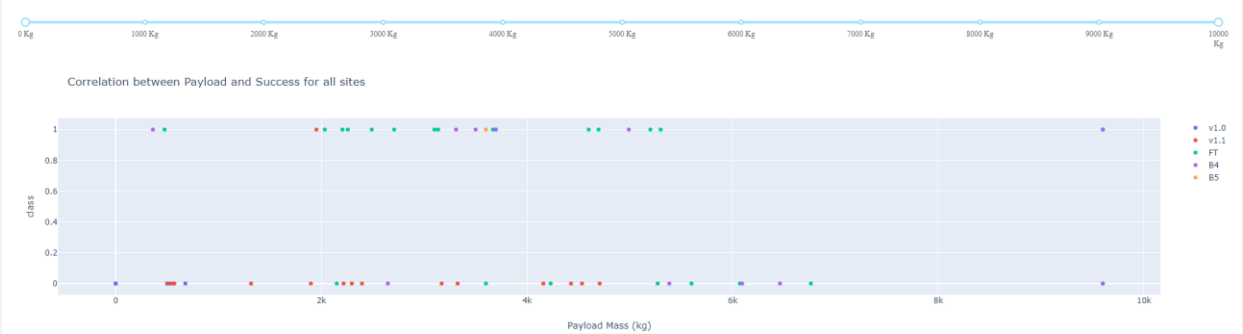
Results

- Exploratory data analysis results
  - 2013 there is significant increase in the success rate;
  - There is 99% of successful mission outcome;
  - KSCLC-39A launch site has highest successful landing rate.
- Interactive analytics demo in screenshots

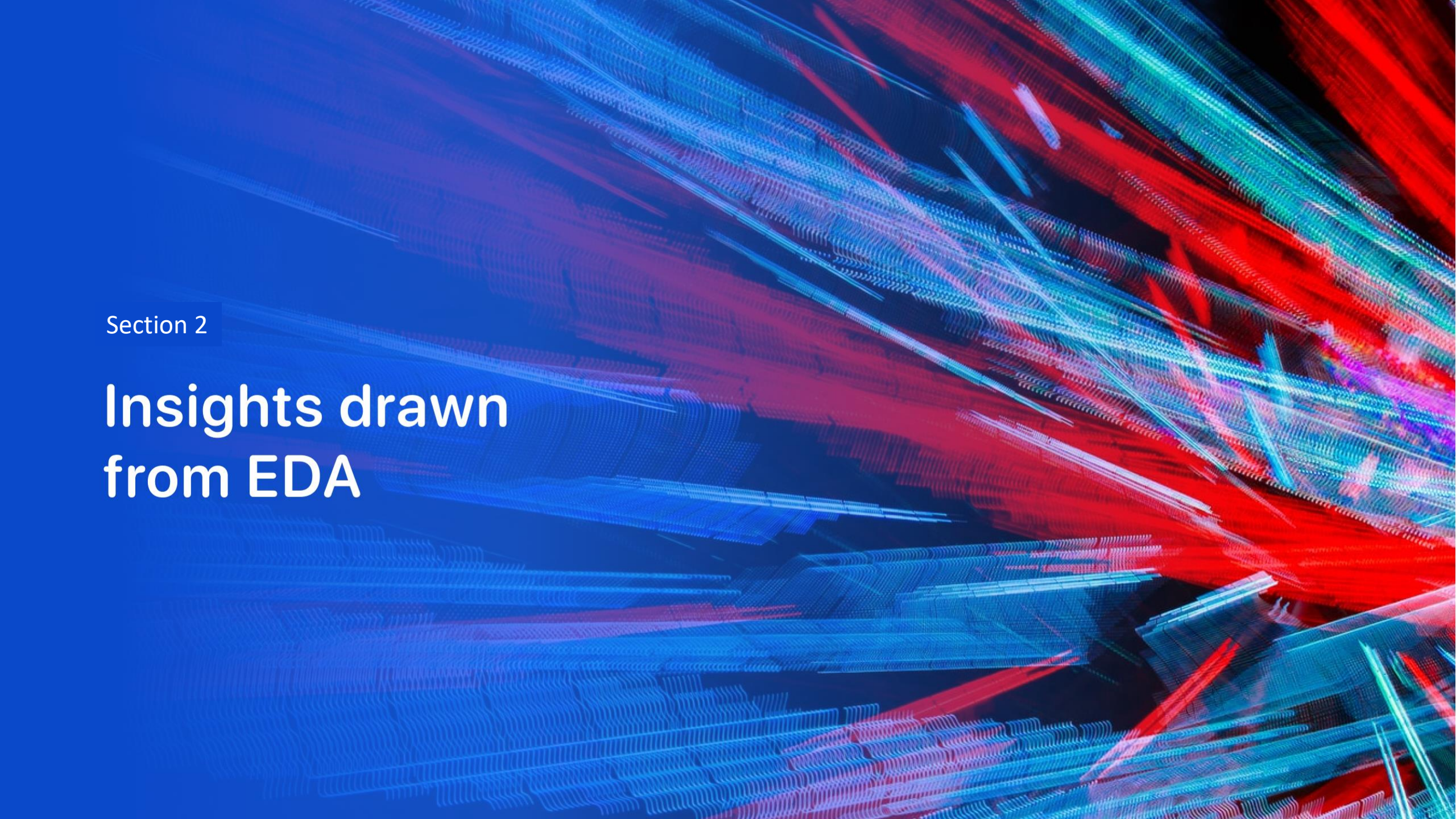


- Predictive analysis results

Decision Tree	0.88
Logistic regression	0.83
SVM	0.83
KNN	0.83



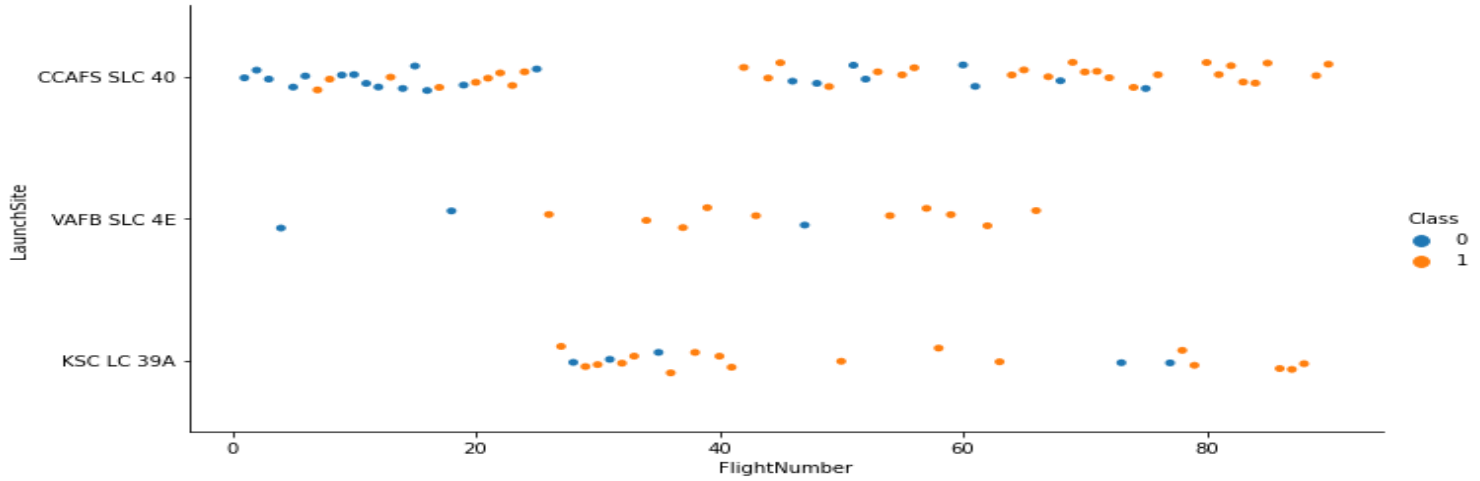
https://github.com/eduofpess/ibm/blob/main/prediction.ipynb


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

```
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 value
        sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=2)
Out[4]: <seaborn.axisgrid.FacetGrid at 0x2114d023d60>
```



Payload vs. Launch Site

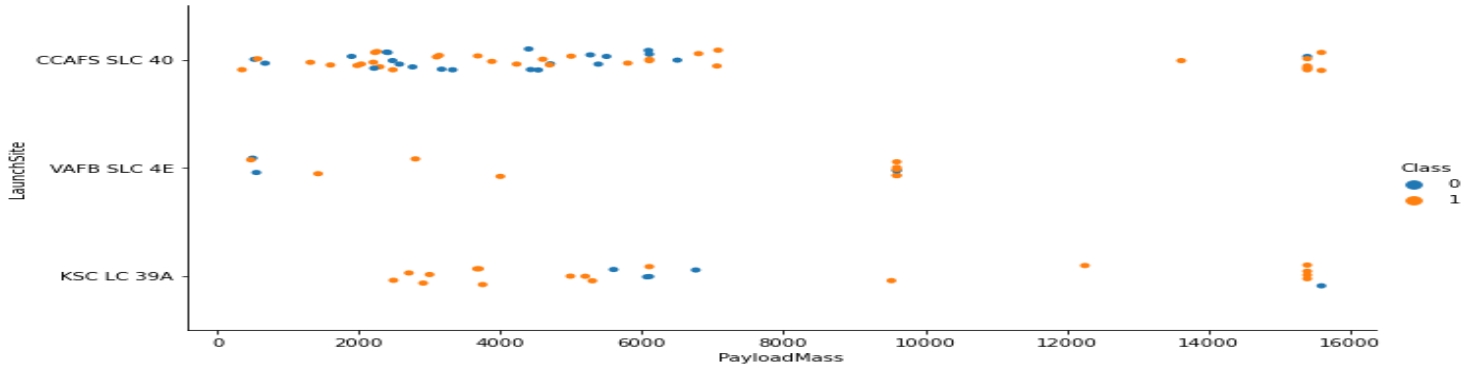
Payload	Sucess Rate
Less 8000 kg	+- 50%
More 8000 kg	+- 100%

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.

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 class val
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect=2)
```

Out[5]:
```
<seaborn.axisgrid.FacetGrid at 0x2114804d550>
```



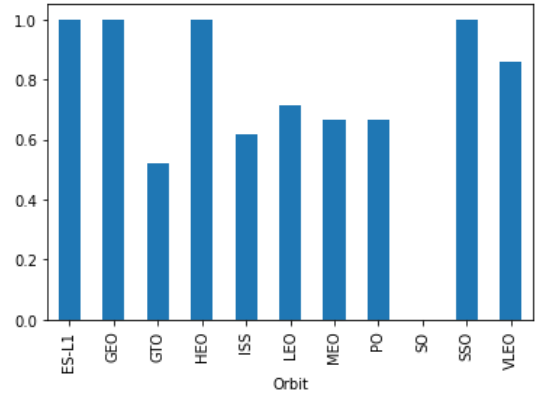
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

- SO problem
- GTO , LEO, Pom GTO – some problem
- Others no problem at all

```
In [6]: # HINT use groupby method on Orbit column and get the mean of Class column
        df_orbit = df.groupby('Orbit').mean()['Class']
        df_orbit.plot(kind='bar')

Out[6]: <AxesSubplot:xlabel='Orbit'>
```



Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type

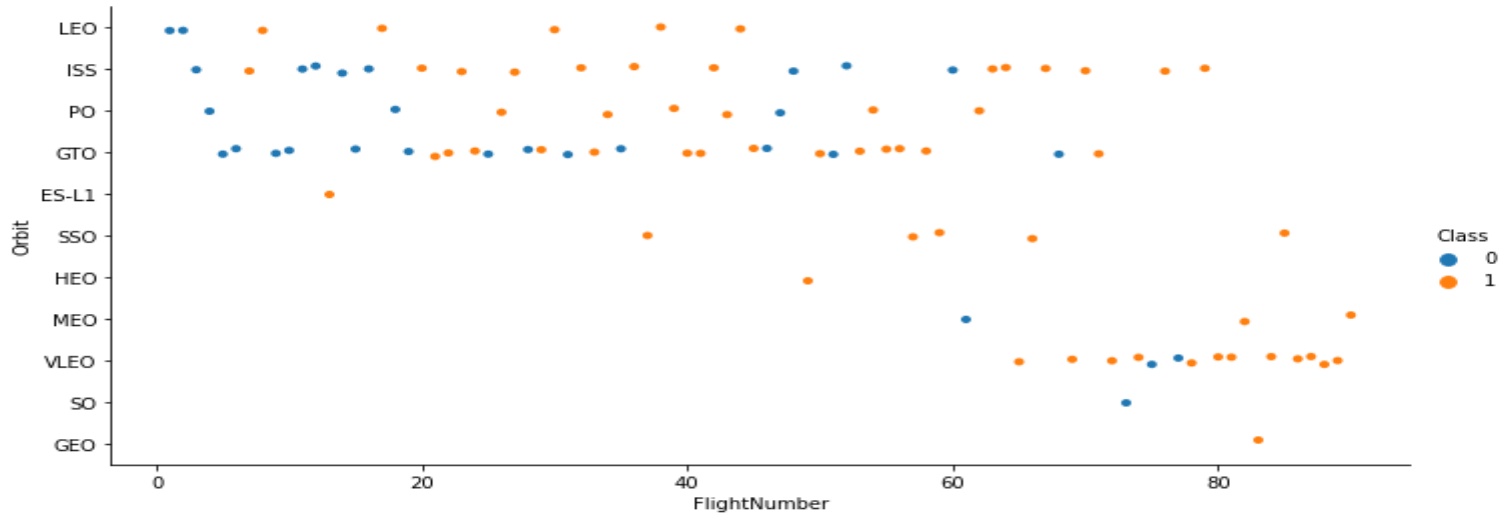
- Above 80 are all successful

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 [7]:
```
# 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=2)
```

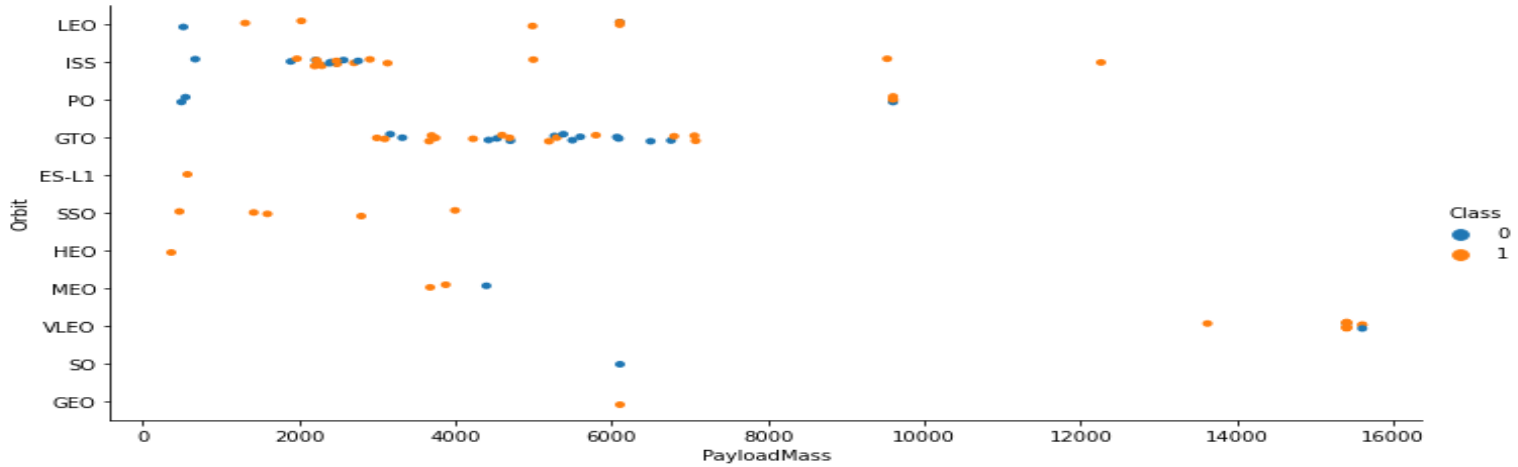
Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x21147fcea30>
```



Payload vs. Orbit Type

- LEO above 2000kg are 100% sccessful;
- SSO 100% successful

```
In [8]: # 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=2)
Out[8]: <seaborn.axisgrid.FacetGrid at 0x2114df682b0>
```



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

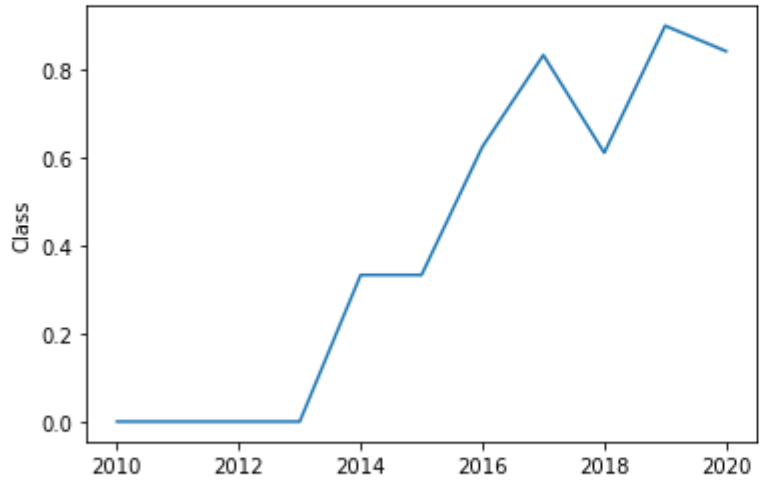
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- After 2018 highest success rate are achieve

```
In [10]: # Plot a line chart with x axis to be the extracted year and y axis to be the success rate
         sns.lineplot(x=df['Year'].unique(), y=df.groupby(['Year'])['Class'].mean())
```

```
Out[10]: <AxesSubplot:ylabel='Class'>
```



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

All Launch Site Names

Query : Select UNIQUE (LAUNCH_SITE) from SPACEXDATASET
Pandas :

```
In [9]: for i in df['Launch_Site'].unique():
            print (i)

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Query : Select * from SPACEXDATASET WHERE LAUNCH_SITE LIKE ‘CCA%’ LIMIT5
- Pandas : df[df.Launch_Site.str.contains('CCA',case=True)][:5]

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Query : Select MAX(PAYLOAD_MASSKG_) from SPACEXDATASET WHERE CUSTOMER='NASA(CRS)
- Pandas : df[df.Customer=='NASA (CRS)'].PAYLOAD_MASS__KG_.max()

```
Out[28]: 3310
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Query : SELECT AVG(PAYLOAD_MASS_KG_) FROM SPACEXDATASET WHERE BOOSTER_VERSION='F9 v1.1'
- Pandas :df[df.Booster_Version=='F9 v1.1'].PAYLOAD_MASS__KG_.mean()

```
Out[36]: 2928.4
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- QUERY : SELECT MIN(DATE) FROM SPACEXDATASET WHERE LANDING_OUTCOME= ‘Success (ground pad)’
- Pandas : df[df["Landing _Outcome"]=='Success (ground pad)'].Date.min()

```
Out[62]: '01-05-2017'
```

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
- Query : SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE LANDING_OUTCOME= ‘Success (drone ship)’ and PAYLOAD_MASS_KG_>4000 AND PAYLOAD_MASS_KG_<6000
- PANDAS :
- df1 = df[df["Landing _Outcome"]=='Success (drone ship)']
- df1[(df1.PAYLOAD_MASS__KG_ >4000) & (df1.PAYLOAD_MASS__KG_ <6000)].Booster_Version

```
Out[103]: 23       F9 FT B1022
          27       F9 FT B1026
          31    F9 FT  B1021.2
          42    F9 FT  B1031.2
          Name: Booster_Version, dtype: object
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Query :
- Select COUNT(*) from SPACEXDATASET WHERE MISSION OUTCOMELIKE 'Success%’

```
Out[120]: 100
```

- Select COUNT(*) rom SPACEXDATASET WHERE MISSION_OUTCOMELIKE 'Failure%’

```
Out[118]: 1
```

- Pandas :
  df[df.Mission_Outcome.str.contains('Success')==True].Mission_Outcome.count(
)
- df[df.Mission_Outcome.str.contains('Failure')==True].Mission_Outcome.count()

Boosters Carried Maximum Payload

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

```
select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASSKG_= (select
MAX(PAYLOAD_MASSKG_) from  SPACEXDATASET)
```

- Pandas :

```
df.loc[df['PAYLOAD_MASS__KG_'] == df['PAYLOAD_MASS__KG_'].max()].Booster_Version
```

```
Out[135]: 74     F9 B5 B1048.4
          77     F9 B5 B1049.4
          79     F9 B5 B1051.3
          80     F9 B5 B1056.4
          82     F9 B5 B1048.5
          83     F9 B5 B1051.4
          85     F9 B5 B1049.5
          92    F9 B5 B1060.2
          93    F9 B5 B1058.3
          94     F9 B5 B1051.6
          95     F9 B5 B1060.3
          99    F9 B5 B1049.7
          Name: Booster_Version, dtype: object
```

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Query :
- SELECT LANDINGOUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATASET WHERE LANDINGOUTCOME LIKE 'Failure(droneship)' and DATE LIKE '2015%'
- Pandas : select = ["Landing _Outcome","Booster_Version", "Launch_Site"]
- df[(df["Landing _Outcome"]=='Failure (drone ship)') & (df.Date.str.contains('2015'))][select]

Out[141]:

	Landing _Outcome	Booster_Version	Launch_Site
13	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
16	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
- Query : select LANDINGOUTCOME , COUNT(LANDINGOUTCOME) as count from SPACEXDATASET GROUPBY LANDINGOUTCOME ORDERBY count DESC
- Pandas :df.groupby("Landing _Outcome")["Landing _Outcome"].count().reset_index(name='count').sort_values(['count'], ascending=False)

Out[180]:

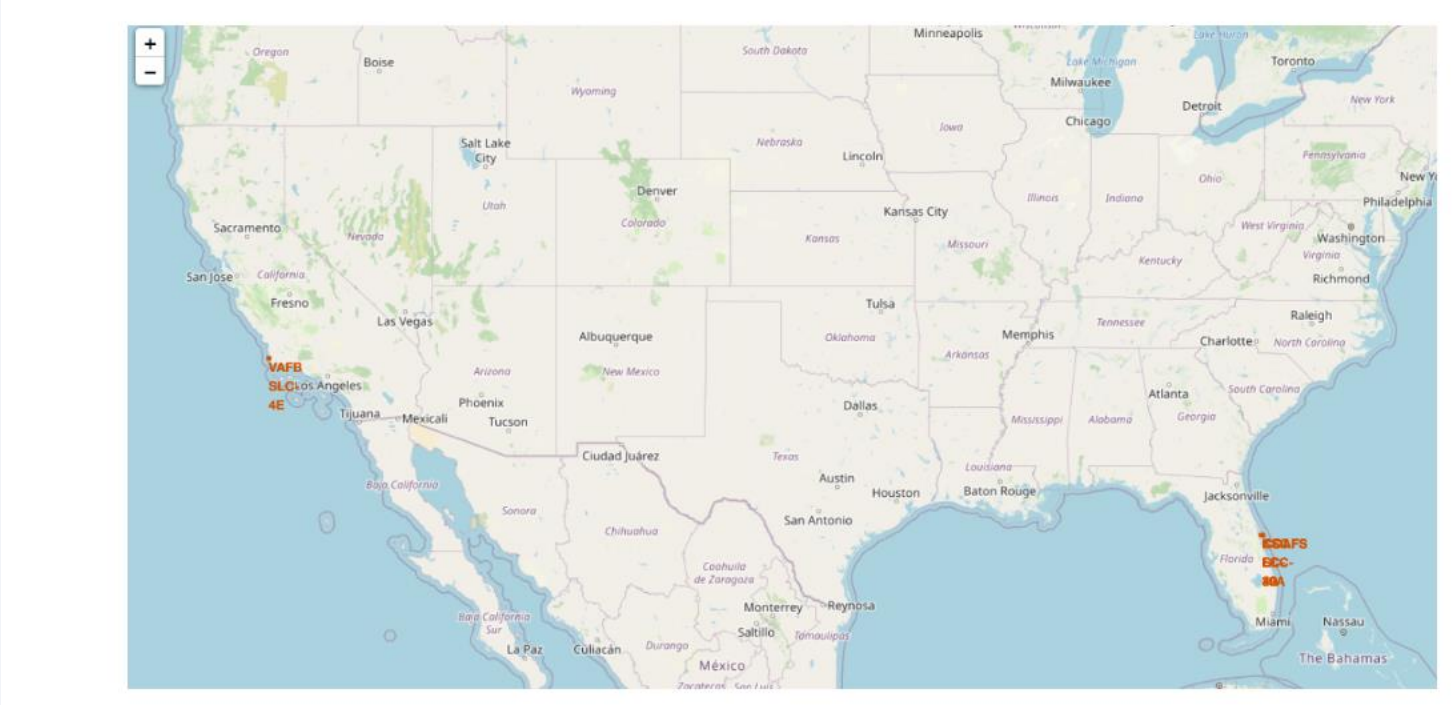
	Landing _Outcome	count
7	Success	38
4	No attempt	21
8	Success (drone ship)	14
9	Success (ground pad)	9
0	Controlled (ocean)	5
2	Failure (drone ship)	5
1	Failure	3
3	Failure (parachute)	2
10	Uncontrolled (ocean)	2
5	No attempt	1
6	Precluded (drone ship)	1

Section 3

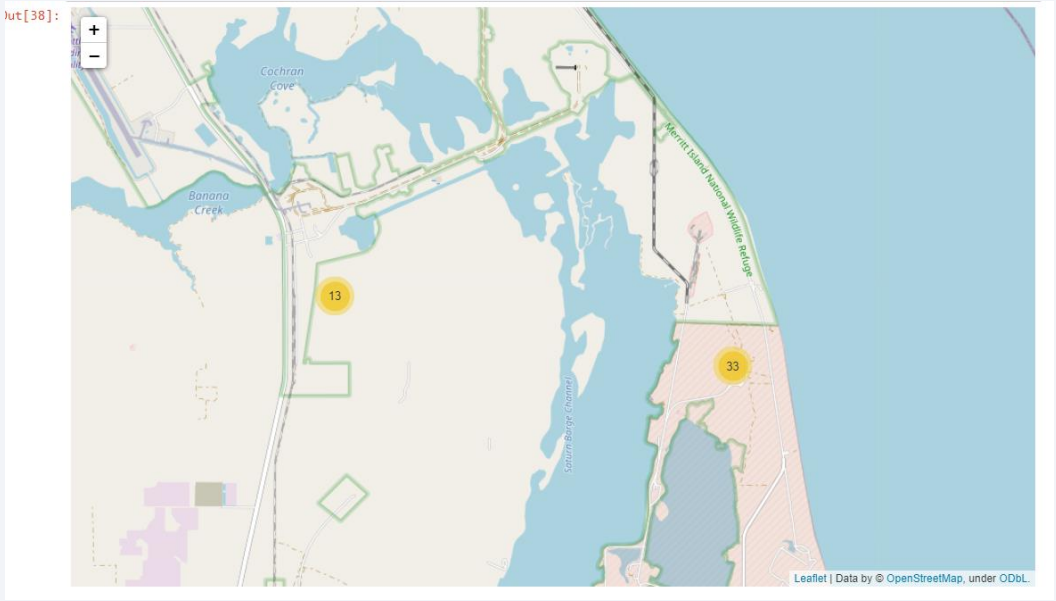
Launch Sites Proximities Analysis

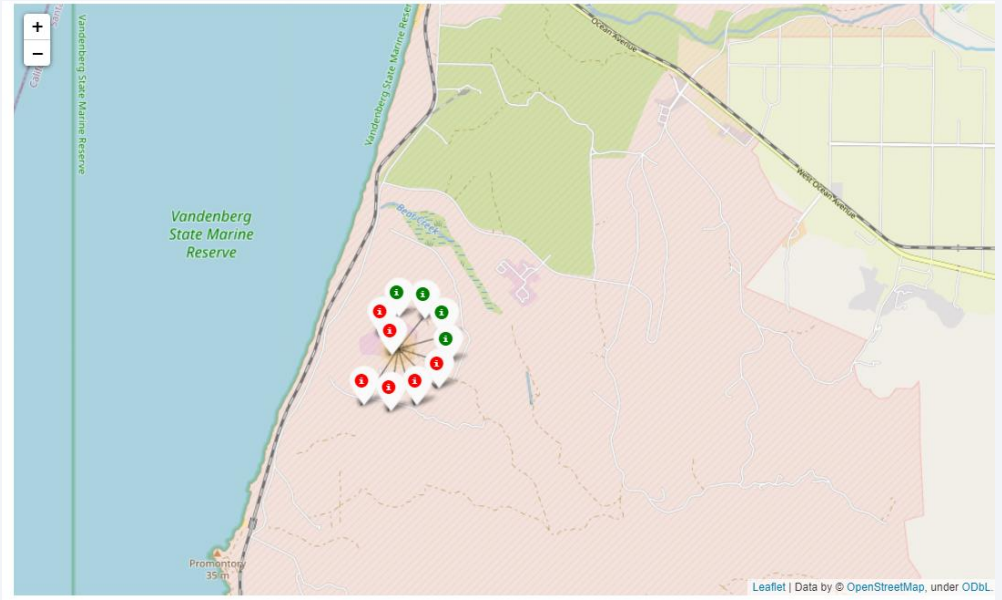
<Launch Sites of SpaceX>

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



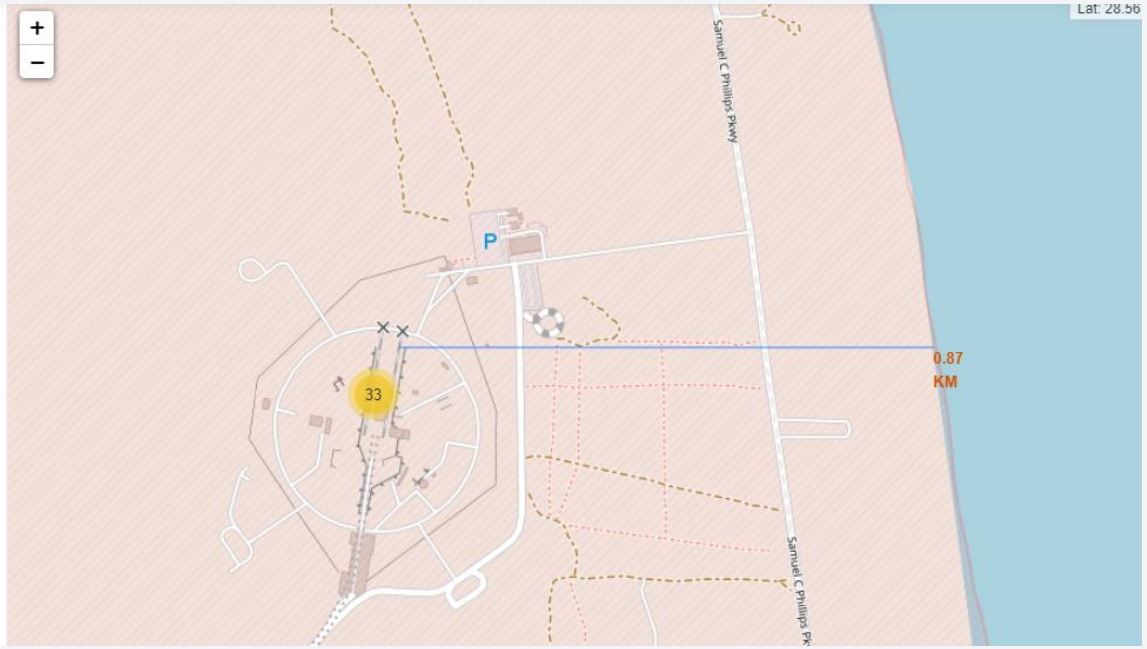
Different Launch Locations Clusters

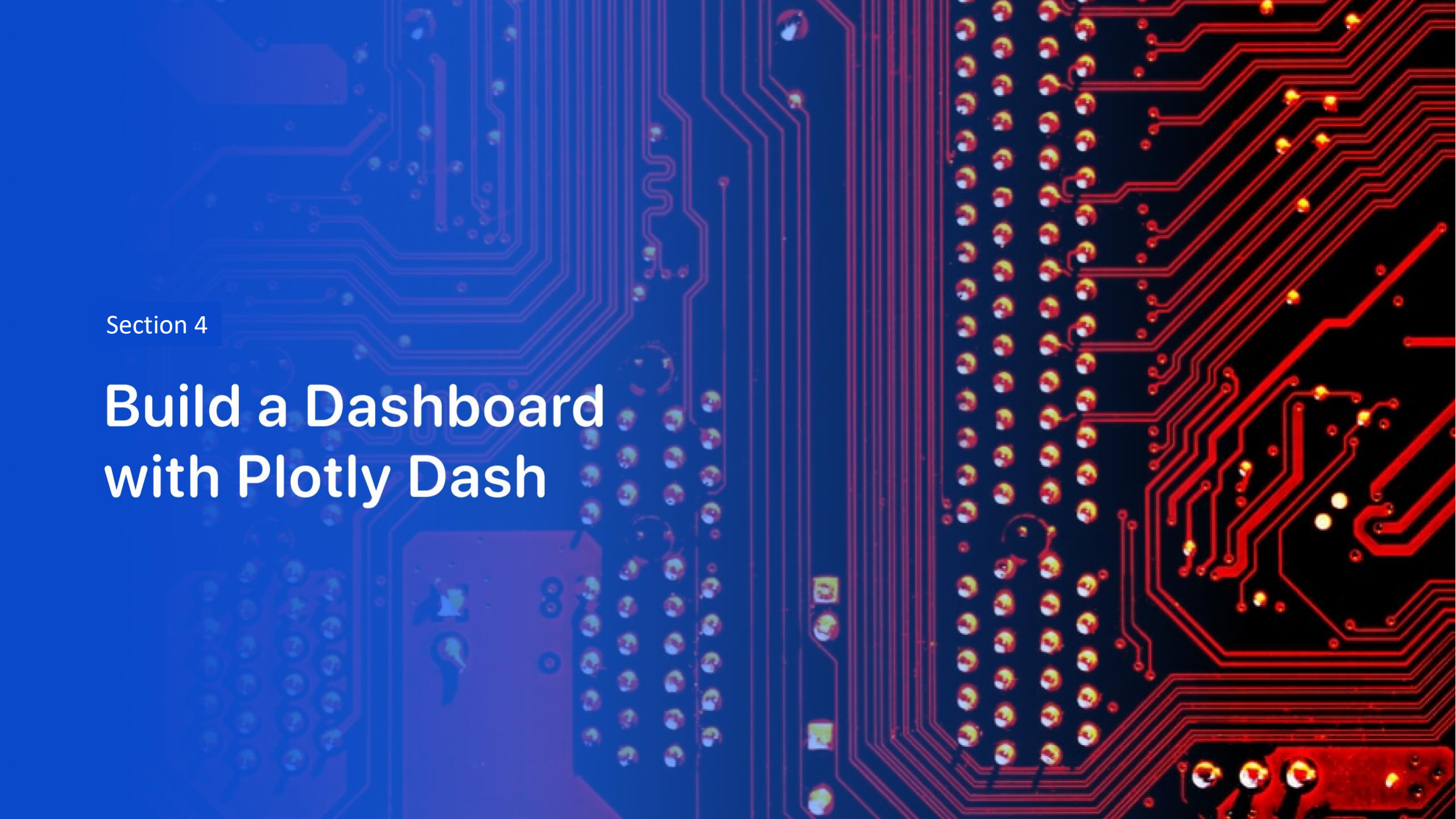




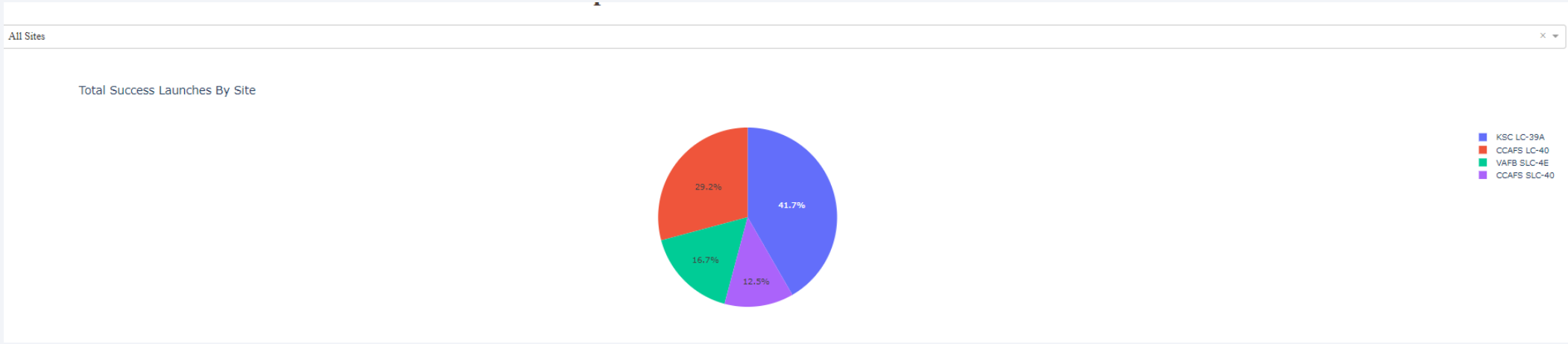
Distance between proximities

Out[43]:

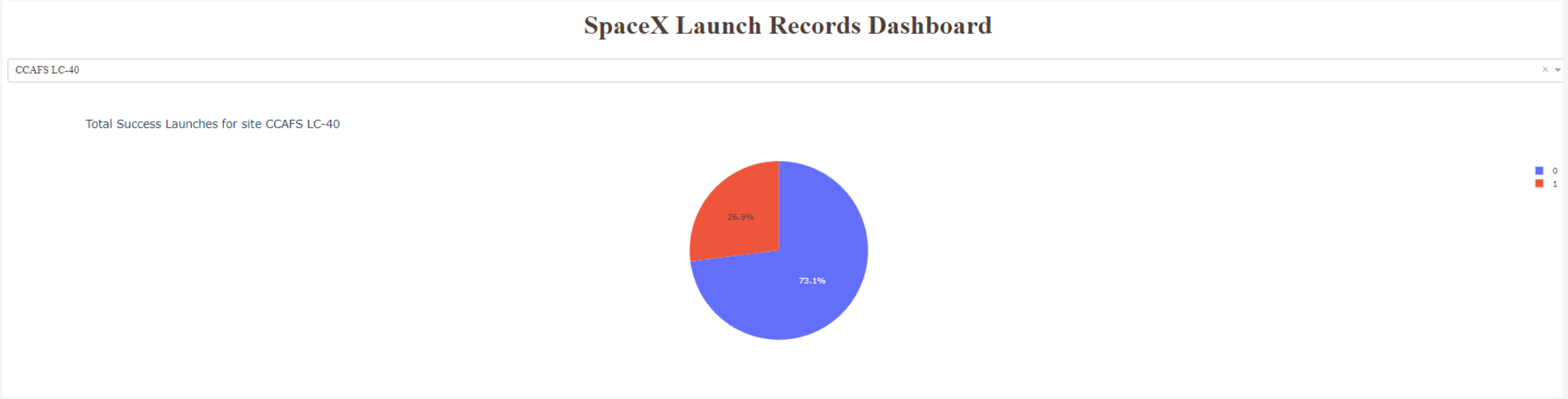



Section 4
Build a Dashboard with Plotly Dash

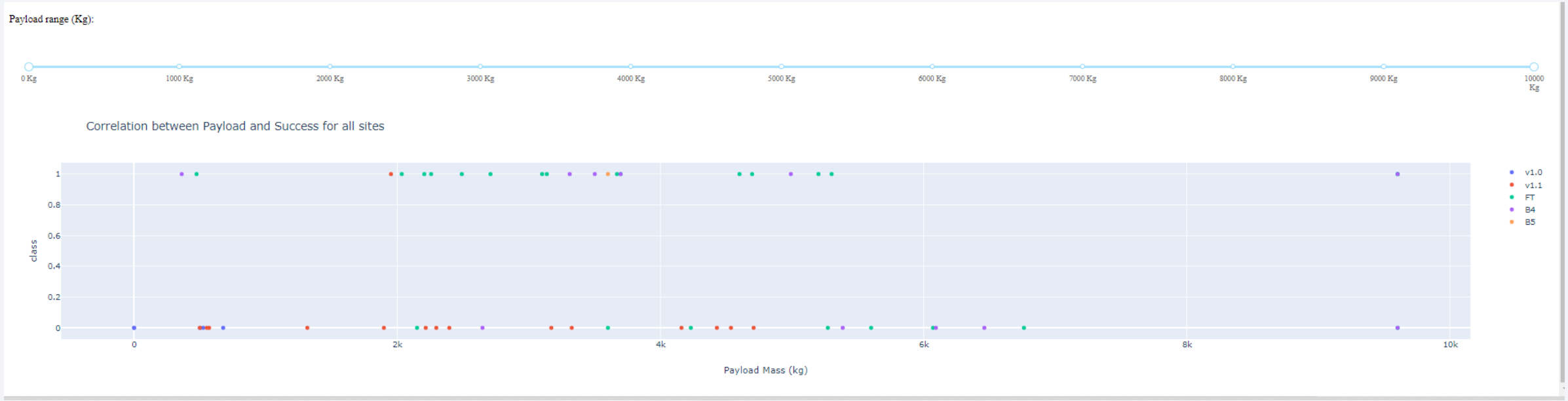
<Pie Chart for all launch sites>



Pie chart with launch site with highest launch success ratio



Payload vs. Launch Outcome scatter plot for all sites

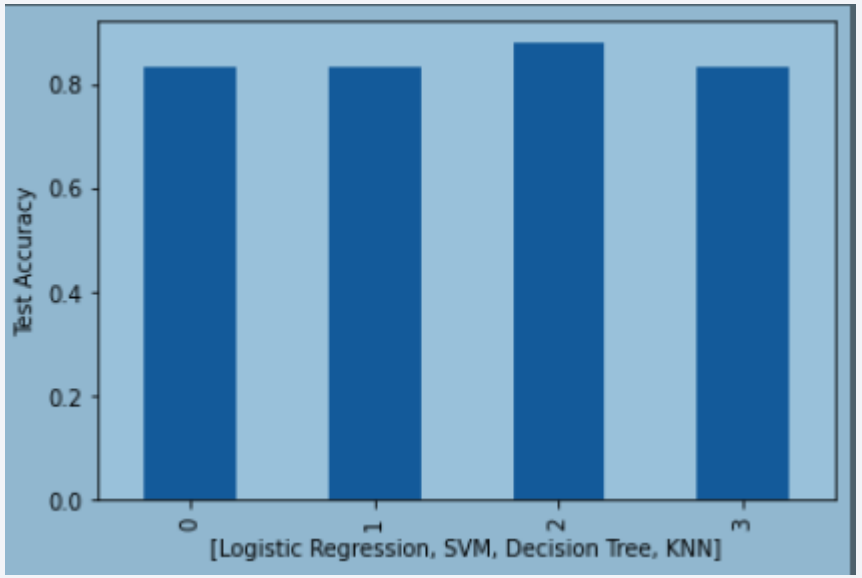


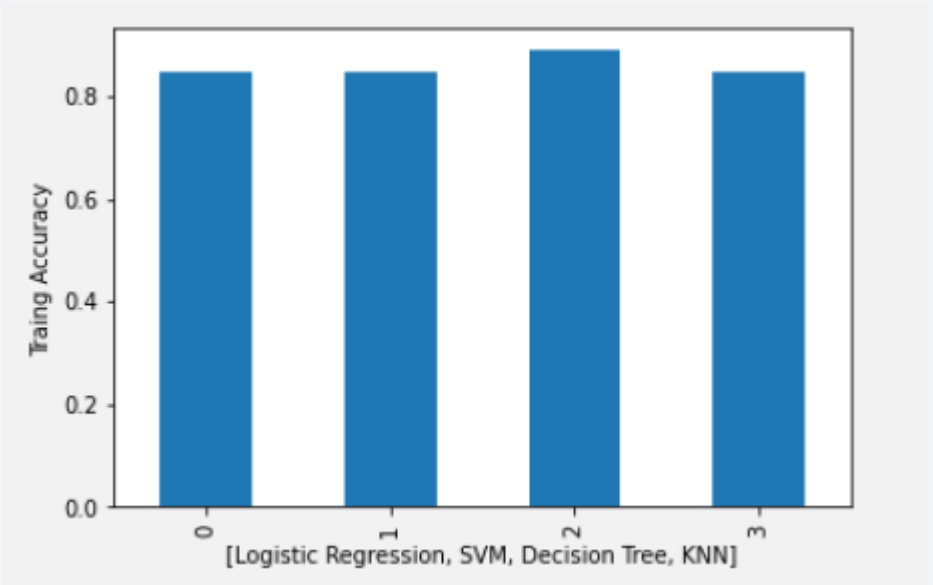
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Decision tree has the best accurancy and Test accurancy

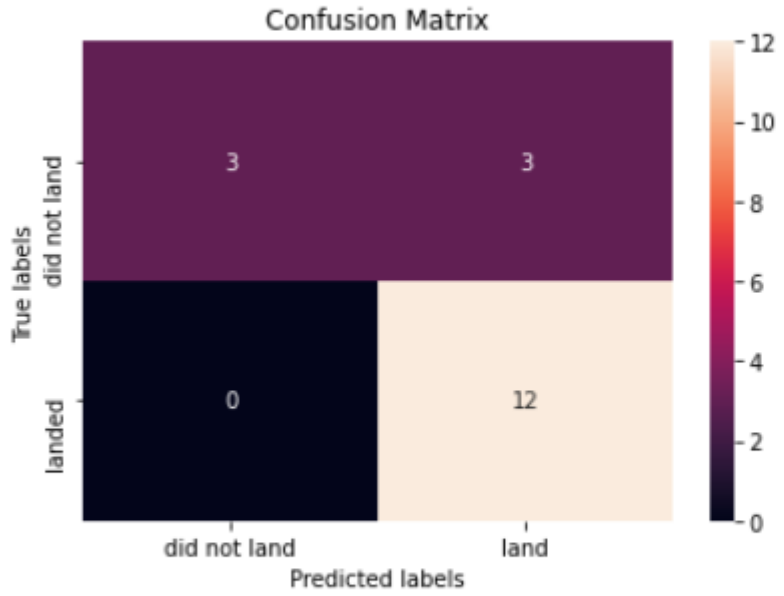




Confusion Matrix

3/ 6 are correctly predicted for negative class and 12/12 are corretly predicted for positive class

In [62]:
```
yhat = logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```



Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

Conclusion

- Decision tree with gini as criterion, max_deaph of 8, Auto as max features, randopm splitter, min_samples_leaf=4 and min_samples_split = 10 gives the best result

Appendix

1. https://github.com/eduofpess/ibm/blob/main/Spacex.csv
2. https://github.com/eduofpess/ibm/blob/main/classification-predictive_analysis.ipynb
3. https://github.com/eduofpess/ibm/blob/main/dash.ipynb
4. https://github.com/eduofpess/ibm/blob/main/data-wrangling.ipynb
5. https://github.com/eduofpess/ibm/blob/main/eda-data_visialization.ipynb
6. https://github.com/eduofpess/ibm/blob/main/eda-sql.ipynb
7. https://github.com/eduofpess/ibm/blob/main/interative_map_folium.ipynb
8. https://github.com/eduofpess/ibm/blob/main/prediction.ipynb
9. https://github.com/eduofpess/ibm/blob/main/spacex-api-data-collection.ipynb
10. https://github.com/eduofpess/ibm/blob/main/spacex_launch_dash.csv
11. https://github.com/eduofpess/ibm/blob/main/webscraping_pages.ipynb

Thank you!