Winning Space Race with Data Science

Bhavya S Devarakonda
10/04/2022

Outline

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

Executive Summary

Space X advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; while other providers cost upward of 165 million dollars. Space X cost is relatively less as it reuses the first stage. Our aim in this project is to predict the first stage launch outcome, that helps determine the cost of a launch.

In order to perform the research analysis, we have followed all the standard Data Science methodologies. The business purpose is to determine if an alternate company wants to bid against SpaceX for a rocket launch. Raw Data was collected from the Space X API and Wikipedia using Web Scraping technique (BeautifulSoup package) and further performed Wrangling and EDA using SQL. Performed Predictive Analysis to build and determine the model accuracy and prediction.

In conclusion, we have observed that the landing success rate got better over time and for certain payloads and Launch Sites, the successful landing outcome is more.

Introduction

Space X advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; while other providers cost upward of165 million dollars. Space X cost is relatively less as it reuses the first stage. Our aim in this project is to predict the first stage launch outcome, that helps determine the cost of a launch.

We want to identify the if the first stage of the launch is successful and determine the cost based on this observation.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Used SpaceX API and BeautifulSoup Python package for webscrapping the data from Wikipedia.
- Perform data wrangling
  - Identified several cases where the booster did not land successfully and converted those to 0-failure or 1-successful and added it as a column 'Class' to the launch DataFrame.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbor algorithms to perform the predictive analysis

Data Collection

- Data Collection has been done using SpaceX API endpoints and also from Wikipedia by using Web Scrapping technique via the BeautifulSoup Python package.
- Following templates explain in detail steps.

Data Collection – SpaceX API

- GitHub: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb

Imported libraries: requests, numpy, pandas, datetime

Defined Auxillary functions that will help us use the API to extract information using the Identification numbers in the launch data.

Request and parse the SpaceX launch data using the GET request

Normalized the JSON response, converted the response to a Dataframe and exported it to CSV.

Data Collection - Scraping

- GitHub URL: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb

Imported libraries: requests, numpy, pandas, datetime

Extract a Falcon 9 launch records HTML table from Wikipedia using requests.get() method

Created a BeautifulSoup Object from HTML response

Parse the table and convert it into a Pandas data frame

Exported the results to CSV.

Data Wrangling

- Loaded the Data from CSV and performed Data Analysis.
  - Identified and calculated the percentage of missing values in each attribute.
  - Identify the columns that are numerical and categorial
- Determined the number of Launch Sites as each launch aims to a dedicated orbit. Review GITURL for the orbit information
- Used value_counts() method to determine the number of occurrence of each orbit, occurrence of landing outcome per orbit type
- Created a landing outcome column based on the outcome of the landing, 0 if bad, 1 if landing is successful
- Git Repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Charts plotted:
  - CatPlot
  - Scatter Plot
  - Bar Chart
- These chart have been plotted to visualize and obtain insights from the data and to perform feature engineering
- Applied on hot encoding to create dummy variables to the categorical columns
- Git repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Performed the following SQL commands:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- Git repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Used circles and markers to each launch site and lines to determine proximity limits for each site from coast, railroad, highway and cities
- We wanted to determine if launch success rate is depended on the location and the proximities of a launch site i.e., the initial position of the rocket trajectories. We've observed that all launch sites are closed to the coast.
- Git repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Used dropdown for launch sites and slider for Payload Mass range
- Used pie chart to display the success rate for all launch sites and also specific sites.
- Plotted scatter plot using Payload range slider and site selected from the dropdown to visualize payload mass vs launch success.
- Git repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/spacex_dash_app.py

Predictive Analysis (Classification)

- Imported Libraries and Defined Auxillary functions, loaded dataframe, created a class column with 0s and 1s
- Normalized the data for better analysis, split test and train data sets. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function GridSearchCV.
- Performed Analysis using estimators, parameters for different models – we used Logistic Regression, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbors.
- Plotted confusion matrix to compare the predicted vs actual values.
- Git repo: https://github.com/bhavyasd/IBM-DataScience-Certificate/blob/main/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Exploratory Data Analysis Results

•Observed that the success rate kept increasing from 2013 to 2020

TASK 6: Visualize the launch success yearly trend

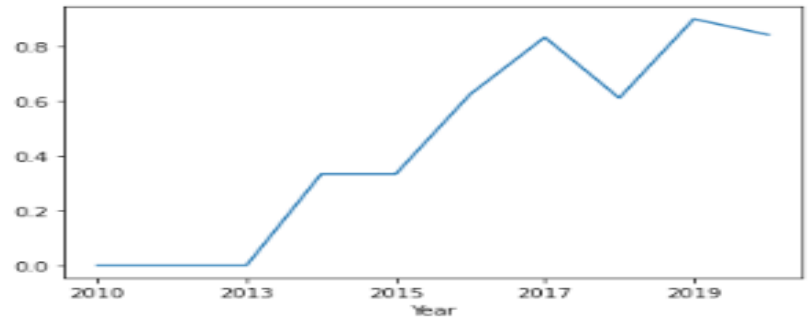
You can plot a line chart with x axis to be `Year` and y axis to be average success rate, to get the average launch success trend.

The function will help you get the year from the date:

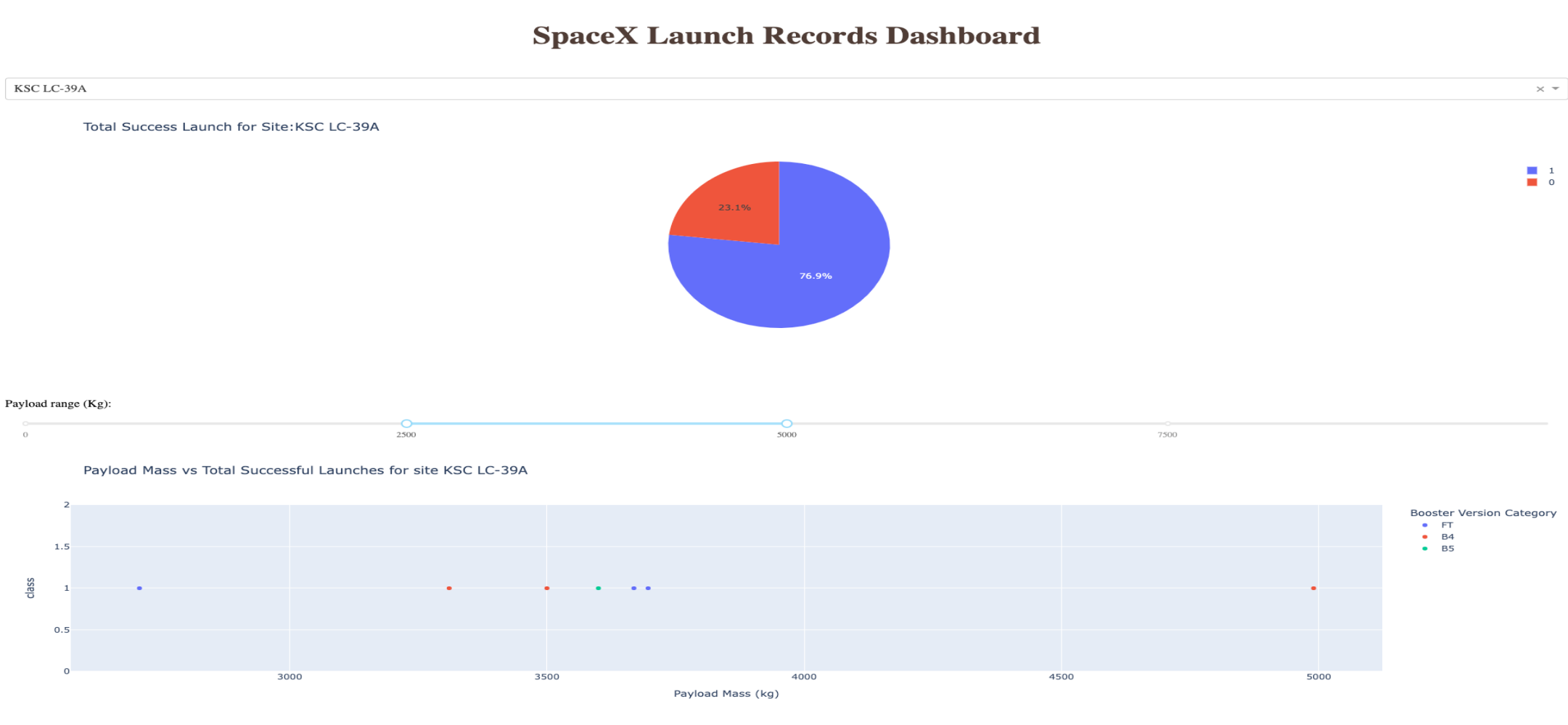
In [9]:
```
# A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

In [41]:
```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate

df1 = pd.DataFrame(Extract_year(df['Date']), columns=['Year'])
df1['Class'] = df['Class']

df1.groupby('Year')['Class'].mean().plot(kind='line')
```

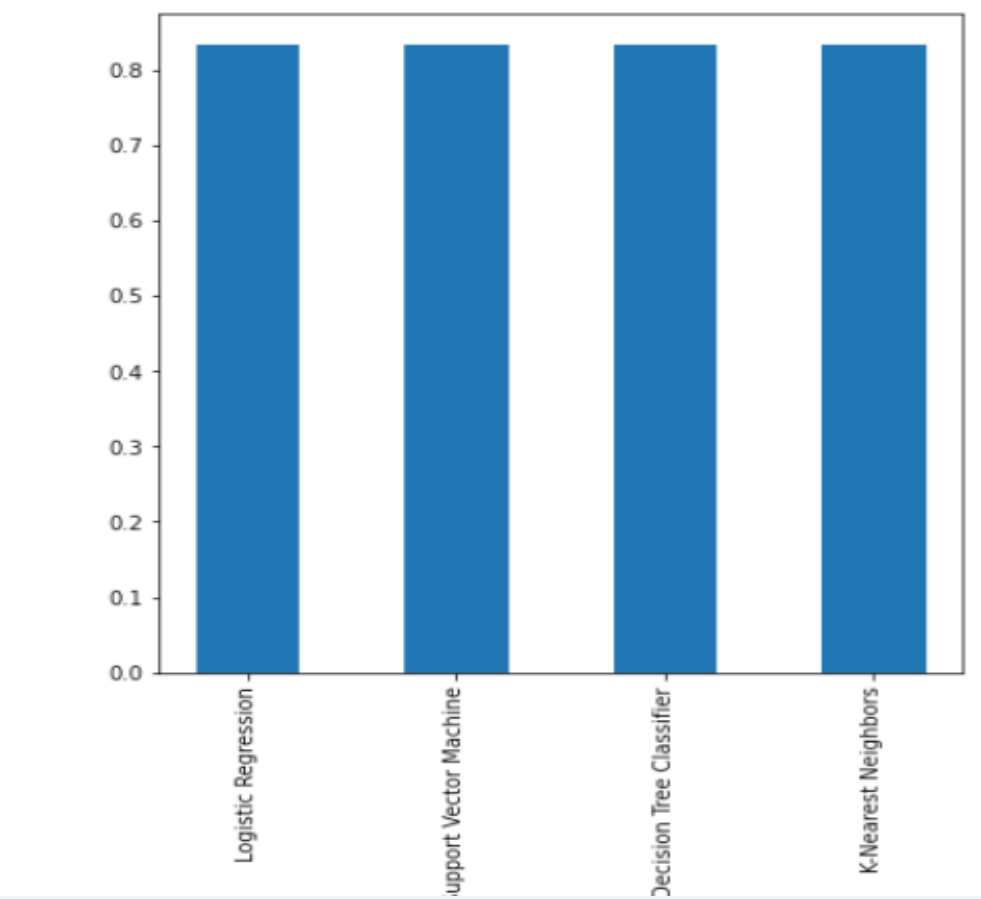
Out[41]:
```
<AxesSubplot:xlabel='Year'>
```

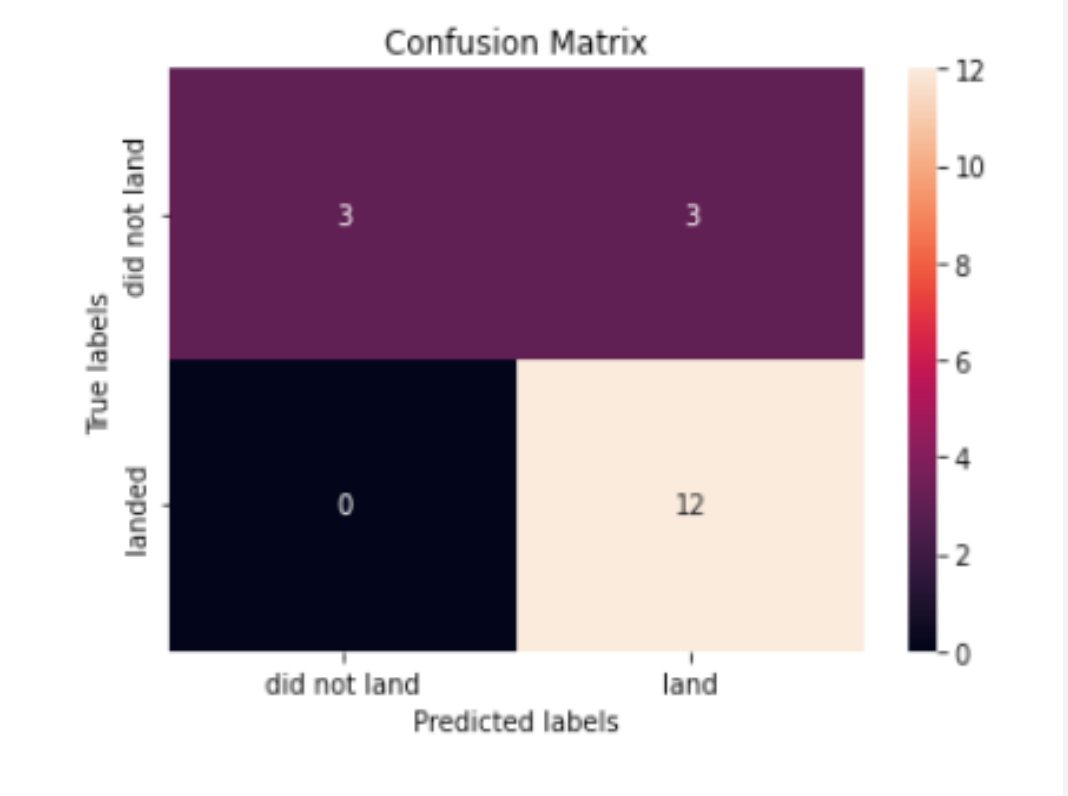


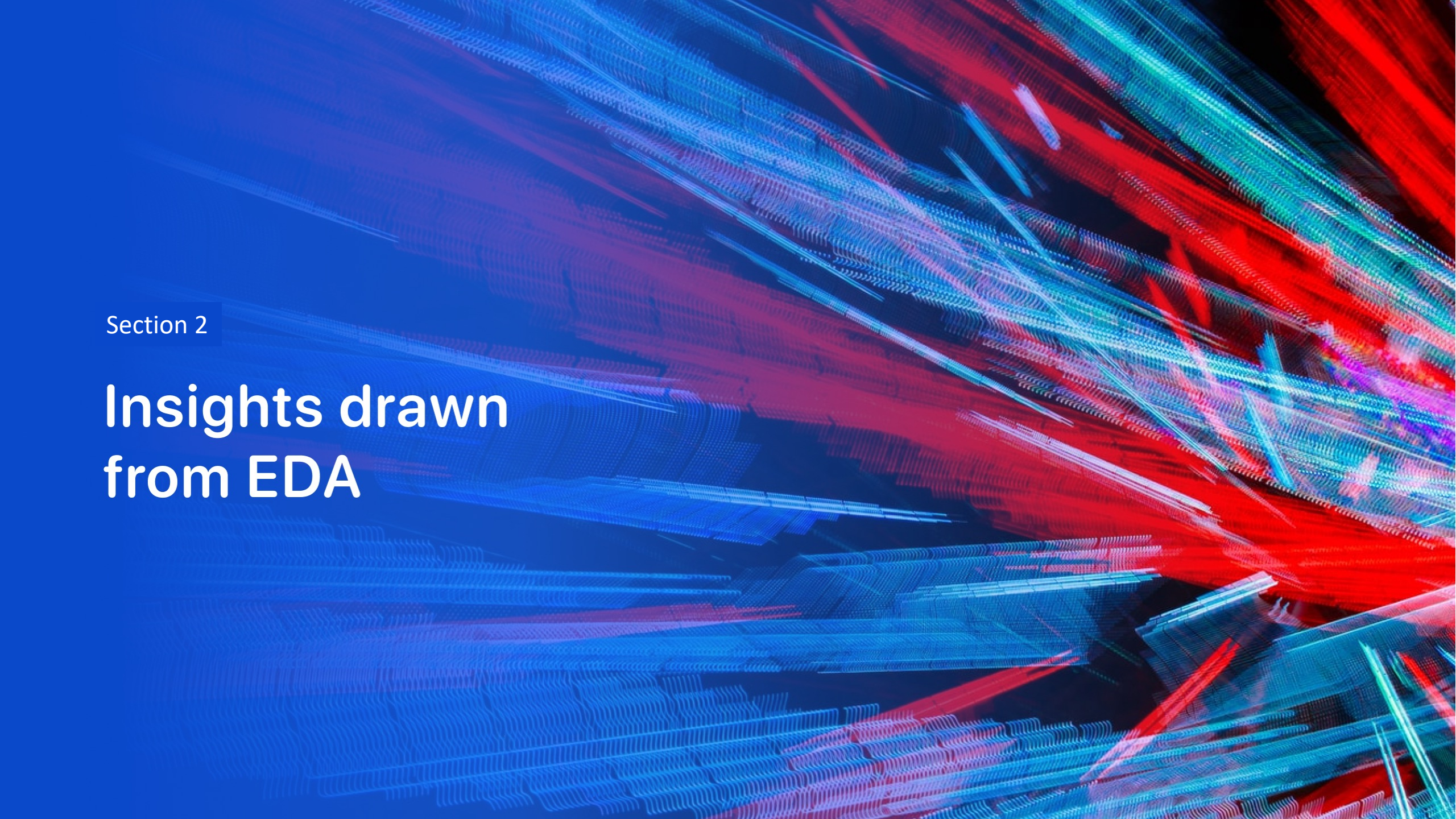
Interactive Analytics Results



Predictive Analysis Results





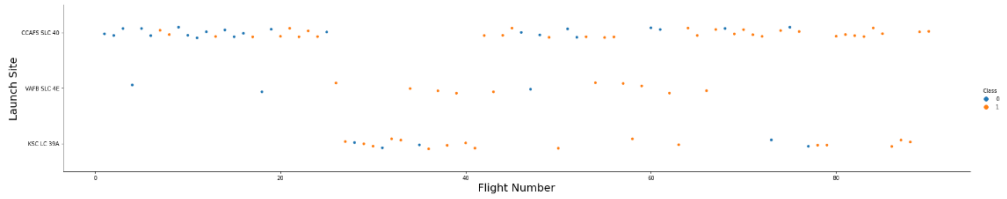

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- We can observe that the Launch Site CCAFS SLC-40 has more launches compared to other sites. We observe the launches to be more successful as the flight number increases.
- Launch Site VAFB-SLC has relatively less failure compared to the other two

In [4]:

```
# Plot a scatter point chart with x axis to be
sns.catplot(y="LaunchSite", x="FlightNumber", h
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```

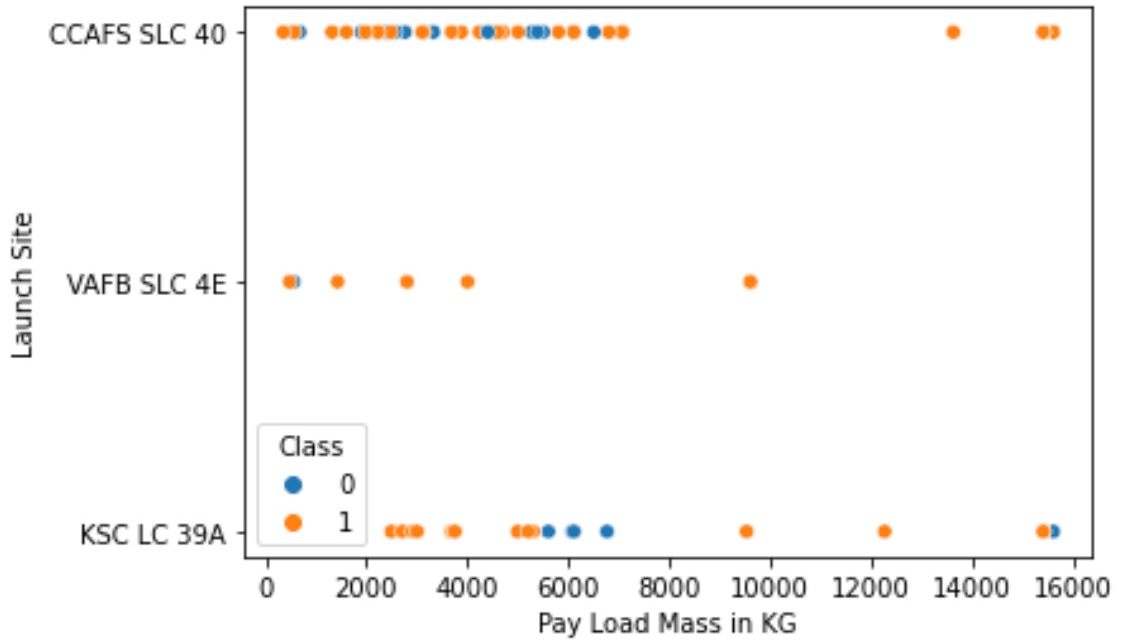


Payload vs. Launch Site

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

5]:

```
# Plot a scatter point chart with x axis to be
sns.scatterplot(y="LaunchSite", x="PayloadMass"
plt.xlabel("Pay Load Mass in KG")
plt.ylabel("Launch Site")
plt.show()
```

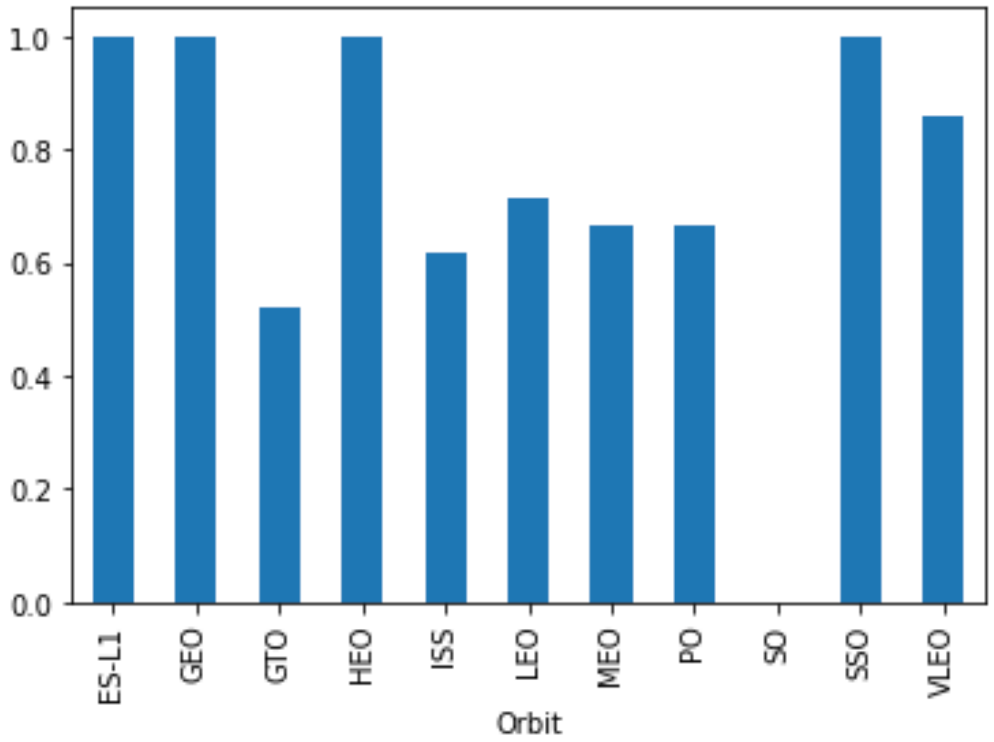


Success Rate vs. Orbit Type

- From the bar chart we observe that the orbits ES-L1, GEO, HEO, SSO, VLEO have a highest success rate.

In [6]:
```
# HINT use groupby method on Orbit column and g
df.groupby('Orbit').mean()['Class'].plot(kind="
```

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

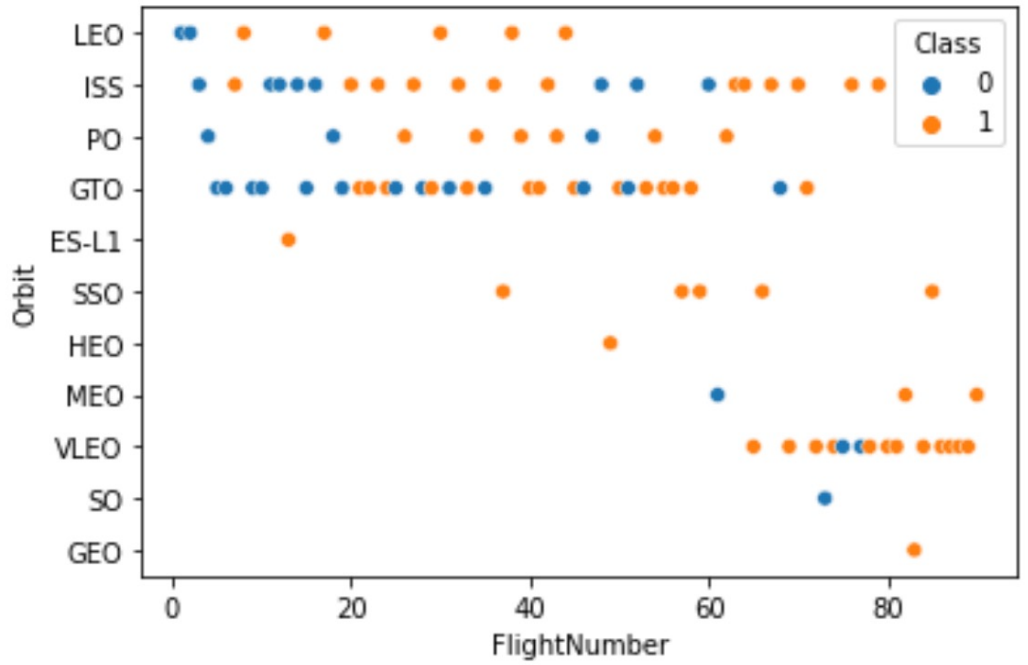


Flight Number vs. Orbit Type

- We observe that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

In [7]:
```
# Plot a scatter point chart with x axis to be
sns.scatterplot(y="Orbit", x="FlightNumber", hu
```

Out[7]:
```
<AxesSubplot:xlabel='FlightNumber', ylabel='Orb
it'>
```

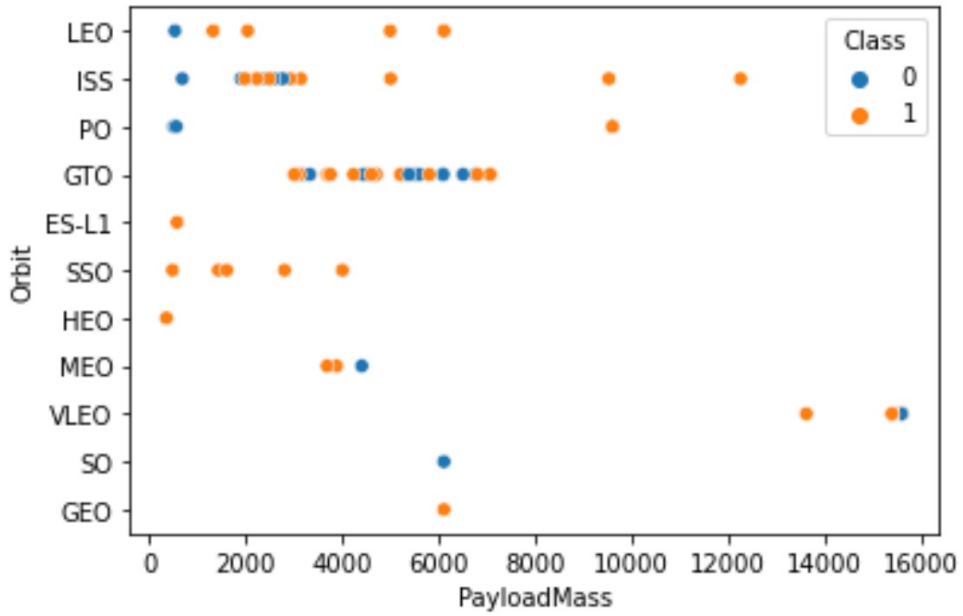


Payload vs. Orbit Type

- We observe that with heavy payloads the successful landing or positive landing rate is 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

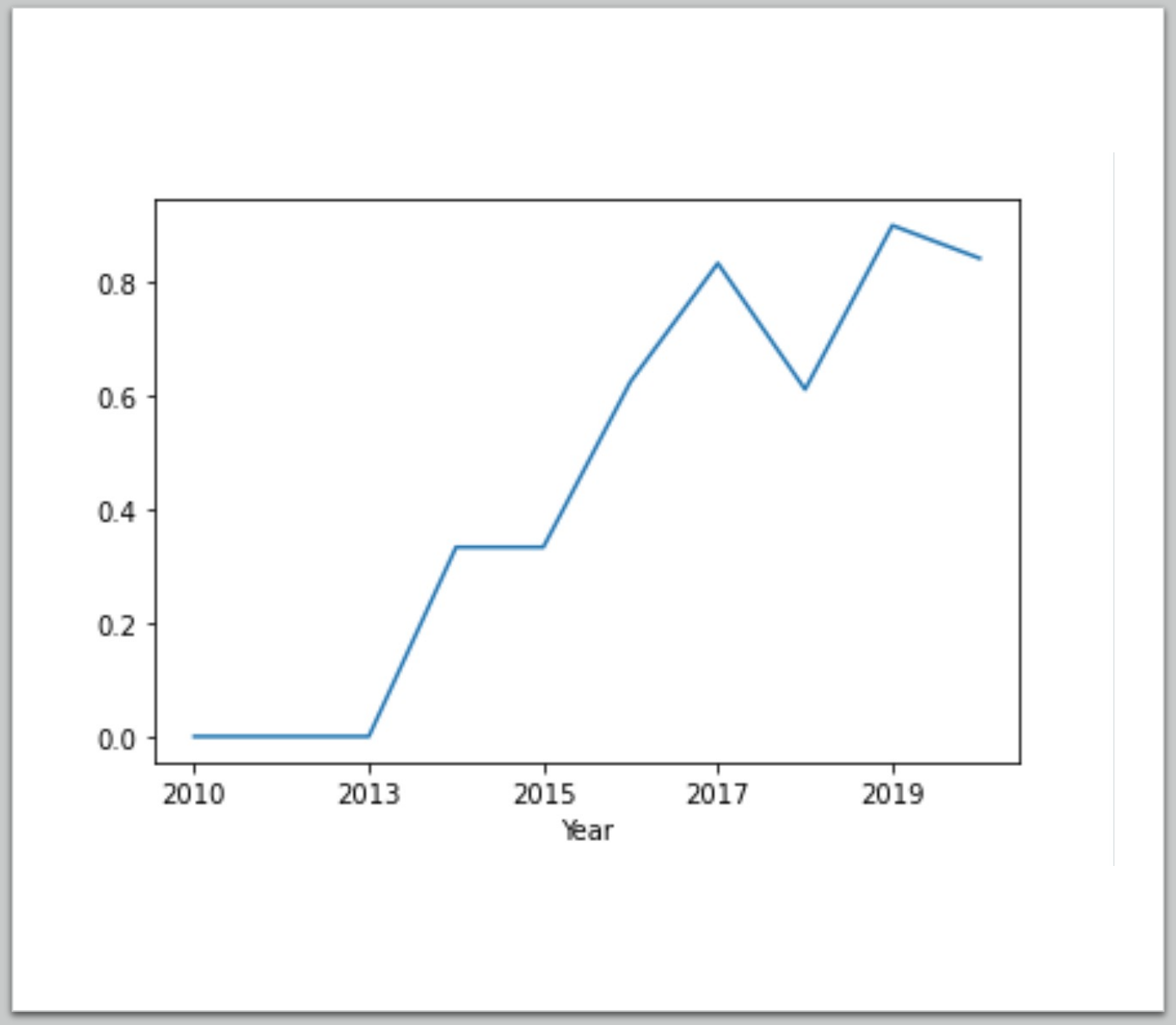
In [8]:
```
# Plot a scatter point chart with x axis to be
sns.scatterplot(y="Orbit", x="PayloadMass", hue
```

Out[8]:
```
<AxesSubplot:xlabel='PayloadMass', ylabel='Orbi
t'>
```



Launch Success Yearly Trend

- We see that the launch success kept increasing since year 2013, with a dip in year 2018.



All Launch Site Names

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

Task 1

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

In [154…

```
%sql select distinct(Launch_Site) from SPACEXTBL
```

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

Out[154…

Launch_Site
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`
- Present your query result with a short explanation here

Task 2

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

[9]:
```
%%sql
select * from SPACEXTBL
where "Launch_Site" like "%CCA%"
limit 5
```

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

[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PA
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

Total Payload Mass

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

Task 3

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

In [10]:
```
%%sql

select sum(PAYLOAD_MASS__KG_) from SPACEXTBL
where customer = "NASA (CRS)"
```

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

Out[10]:

sum(PAYLOAD_MASS__KG_)
45596

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

Task 4

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

In [11]:

```
%%sql

select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL
where Booster_Version like '%F9 V1.1%'
```

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

Out[11]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

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

Task 5

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

Hint:Use min function

In [41]:

```
%%sql

Select min(substr(Date,7,4) || substr(Date,4,2) ||
from SPACEXTBL
where "Landing _Outcome" = 'Success (ground pad)'

--select Date
--from SPACEXTBL
--where "Landing _Outcome" like '%Success (ground p
--ORDER BY Date Desc
--Limit 1
```

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

Out[41]:

min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2))
20151222

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

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 [53]:

```
%%sql

select Booster_Version, PAYLOAD_MASS__KG_
from SpaceXTBL
where "Landing _Outcome" = 'Success (drone ship)'
and (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)
```

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

Out[53]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

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

Task 7

List the total number of successful and failure mission outcomes

In [112…

```
%%sql

Select Count(Mission_Outcome), Mission_Outcome
from SPACEXTBL
where (Mission_Outcome like '%Success%' or Mission_
group by Mission_Outcome
```

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

Out[112…

Count(Mission_Outcome)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

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

Task 8

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

In [116…

```
%%sql
select Booster_Version
from spacextbl
where (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

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

Out[116…

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014
F9 v1.1 B1015
F9 v1.1 B1016
F9 v1.1 B1018
F9 FT B1019
F9 v1.1 B1017
F9 FT B1020
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1024
F9 FT B1025.1
F9 FT B1026
F9 FT B1029.1

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

30…
```
%%sql

select substr(date,4,2) as "Month", "landing _outcome", booster_version, launch_site from
where substr(Date,7,4) = '2015'
and "Landing _Outcome" = 'Failure (drone ship)'
```

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

30…

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- We can observe that there are a total of 8 successful landing outcomes between 2010 and 2017, where 5 boosters landed on drone ship and 3 on ground pad

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

n [151…

```
%%sql

SELECT "DATE","Landing _Outcome",count("Landing _Outcome")as LANDING_OUTCOME_COUNT
from SPACEXTBL
where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604'
and '20170320'
and "landing _Outcome" like '%Success%'
group by "Landing _Outcome"
order by count("Landing _Outcome") desc
```

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

ut[151…

Date	Landing _Outcome	LANDING_OUTCOME_COUNT
08-04-2016	Success (drone ship)	5
22-12-2015	Success (ground pad)	3


Section 3
Launch Sites
Proximities Analysis

All Launch Sites Map

- ‘All launch sites’ location markers on a global map
- The launch sites are above the equator line and closer to the coast
- 4 launch sites are marked in red
  - VAFB on the west coast
  - CCA, CCAS, KSLC on the southern coast

Launch Site Markers/Labels

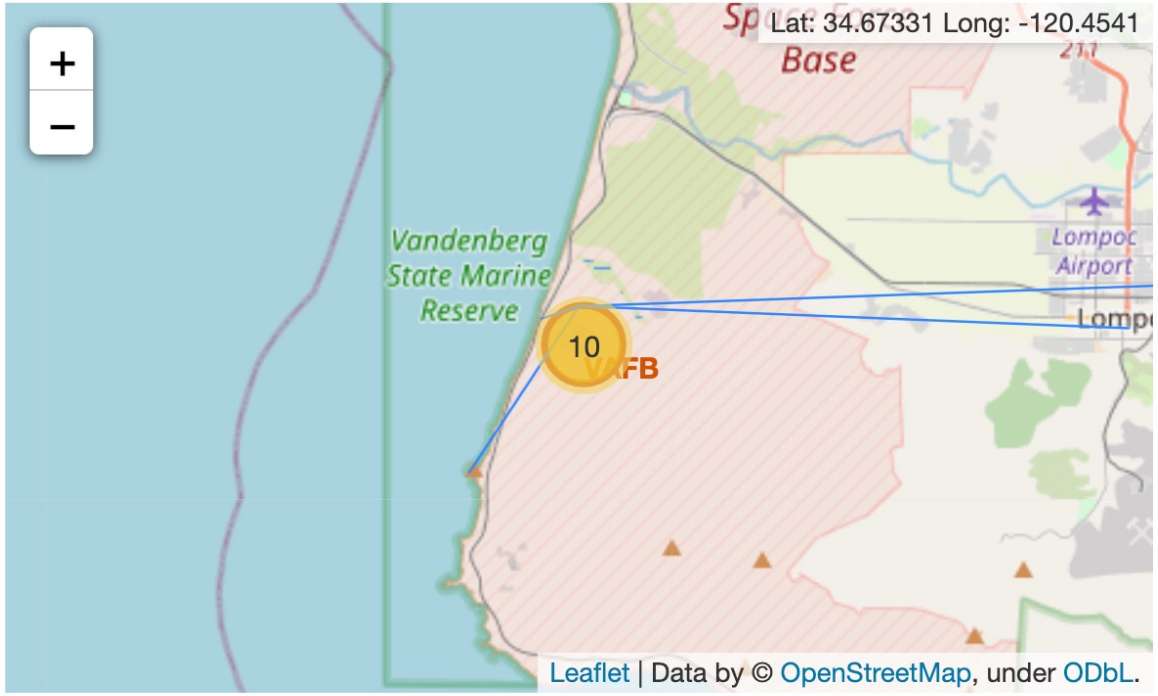
- We observer 10 launch outcomes from west coast launch site and 46 outcomes from the southern coast displayed as a cluster.
- If we zoom in further, we can also see the launch outcomes

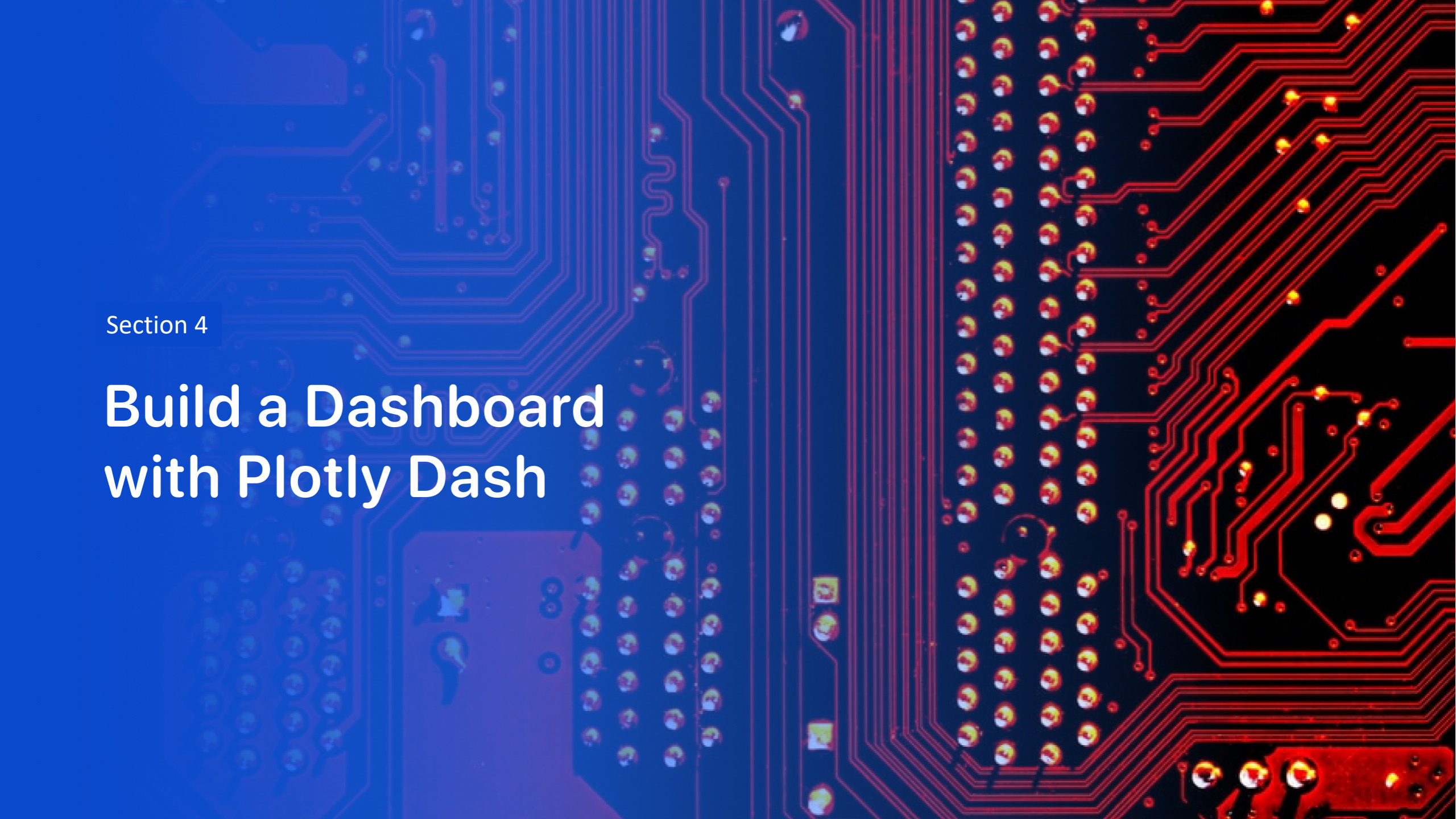
[15]:

Launch Site Proximities

- We observe that Launch sites are closest to the coast and then the railroads
- The launch sites are little further from a city and a highway

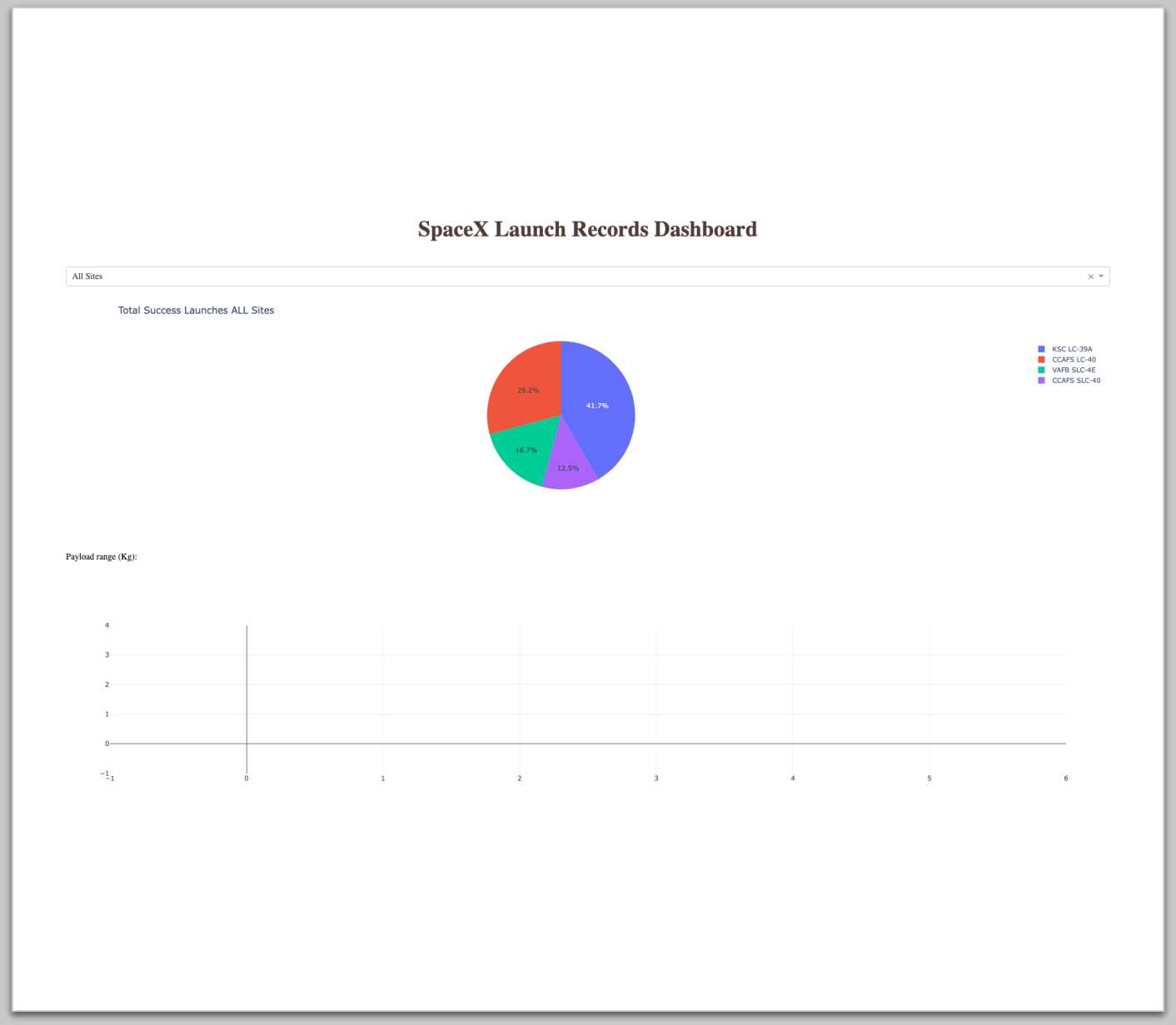
3]:




Section 4
Build a Dashboard with Plotly Dash

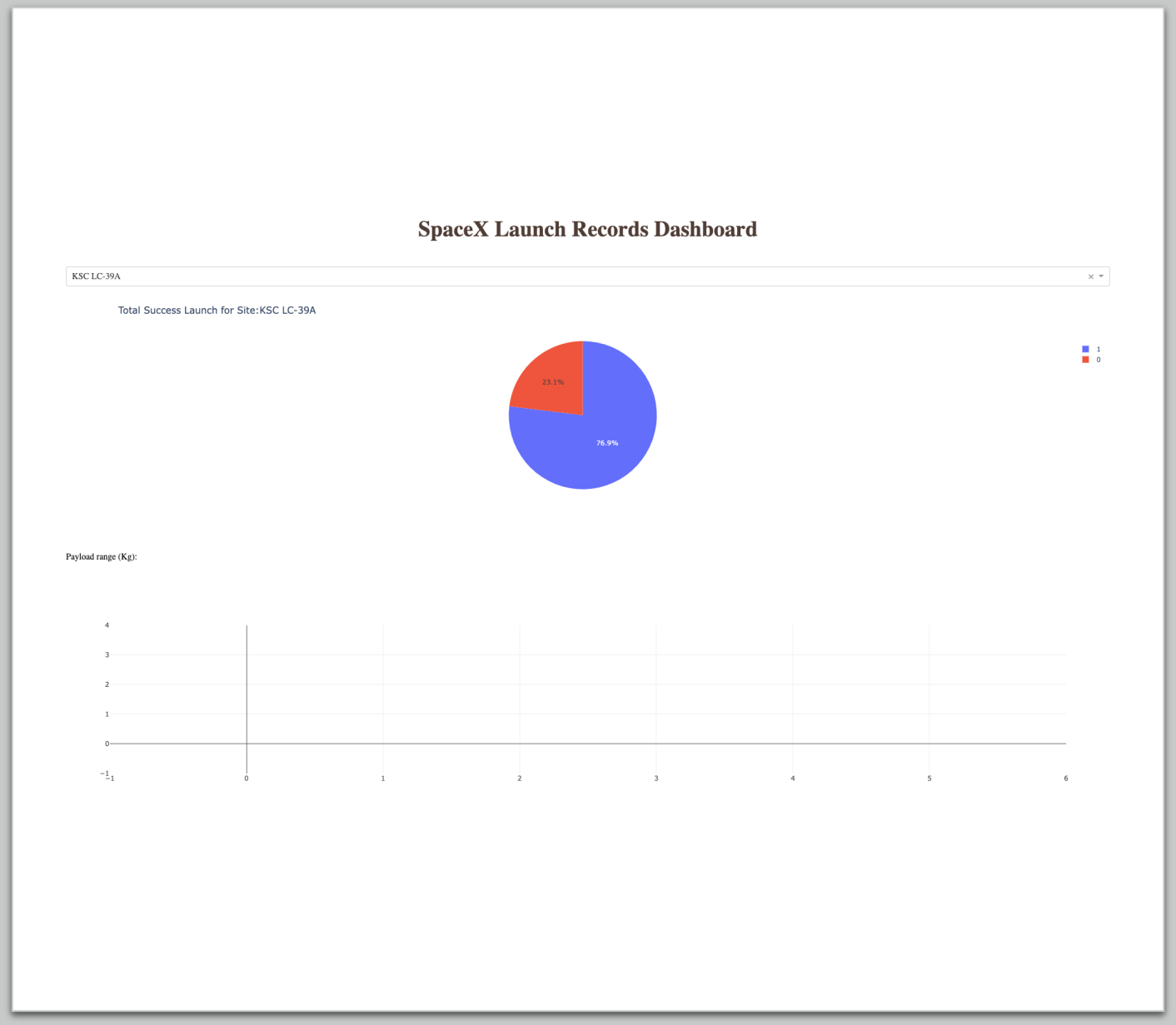
Launch Success for All sites

- Pie chart displaying the success rate for all launch sites
- We observe that KSC LC-39A has the maximum success rate and CCAFS SLC-40 has the least



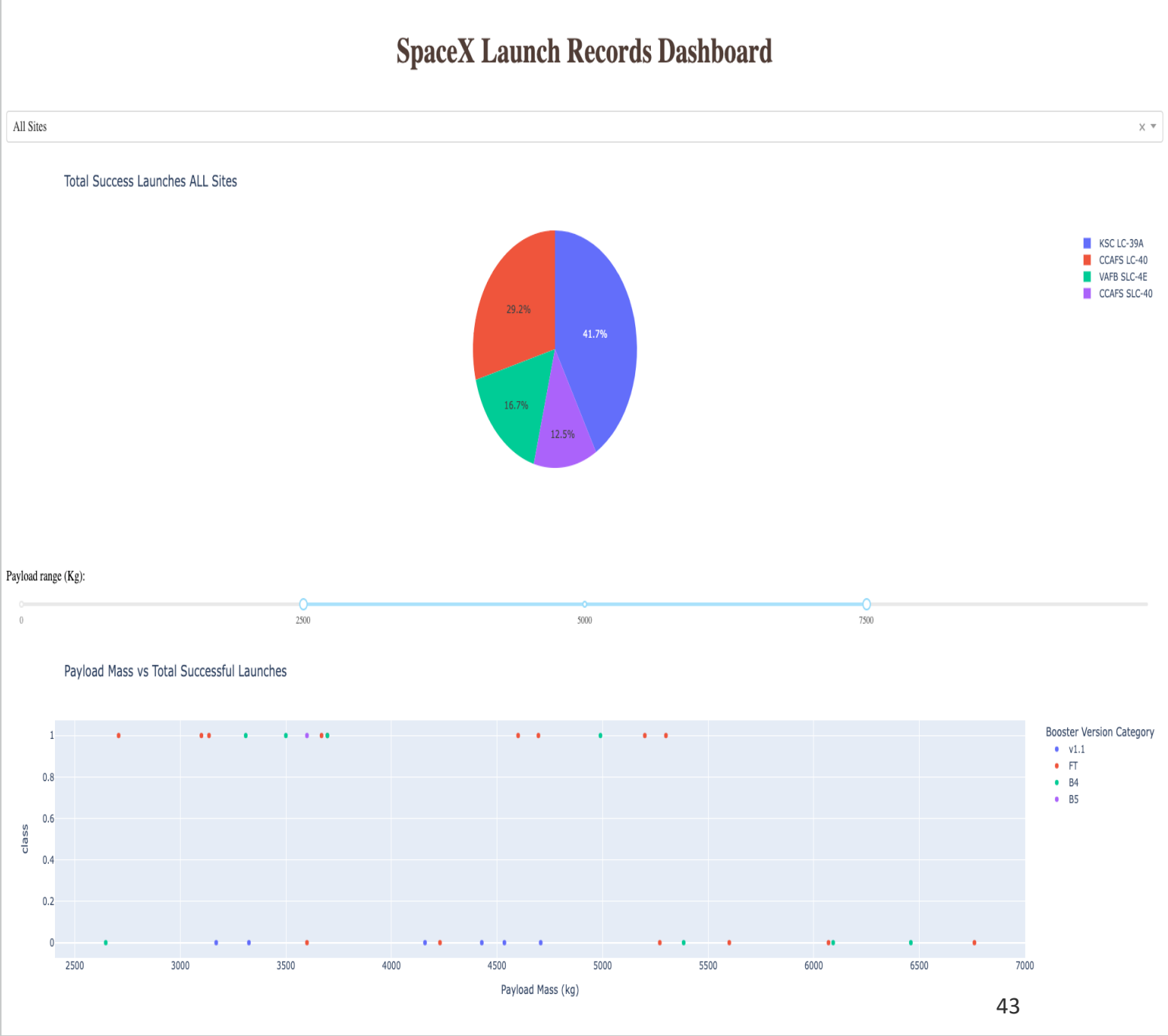
Launch Site with Highest Success

- KSC LC-39A has the highest success



Payload Vs Launches

- We observe the KSC LC-39A has a success rate of 41.7%
- For the payload range of 2500 to 7500 Kgs, the boosters FT have the highest success rates and the booster v1.1 have the lowest



Section 5

Predictive Analysis (Classification)

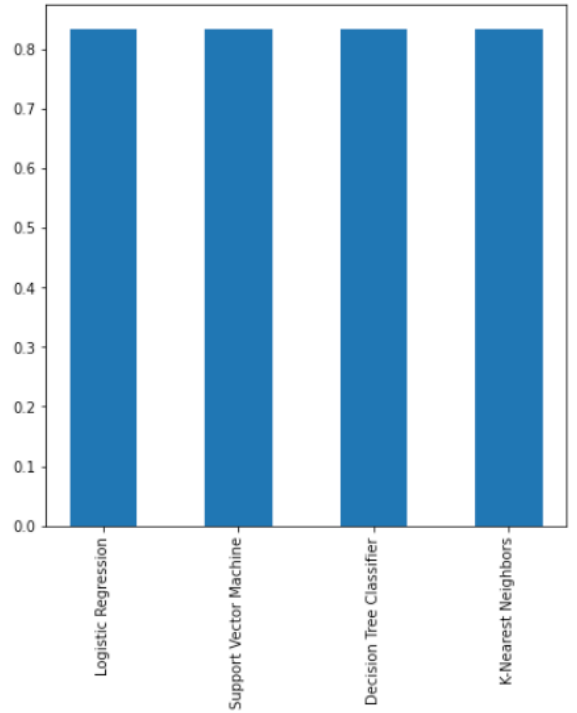
Classification Accuracy

- Bar Chart for all models calculating accuracy
- All models have same accuracy due to the small dataset

TASK 12

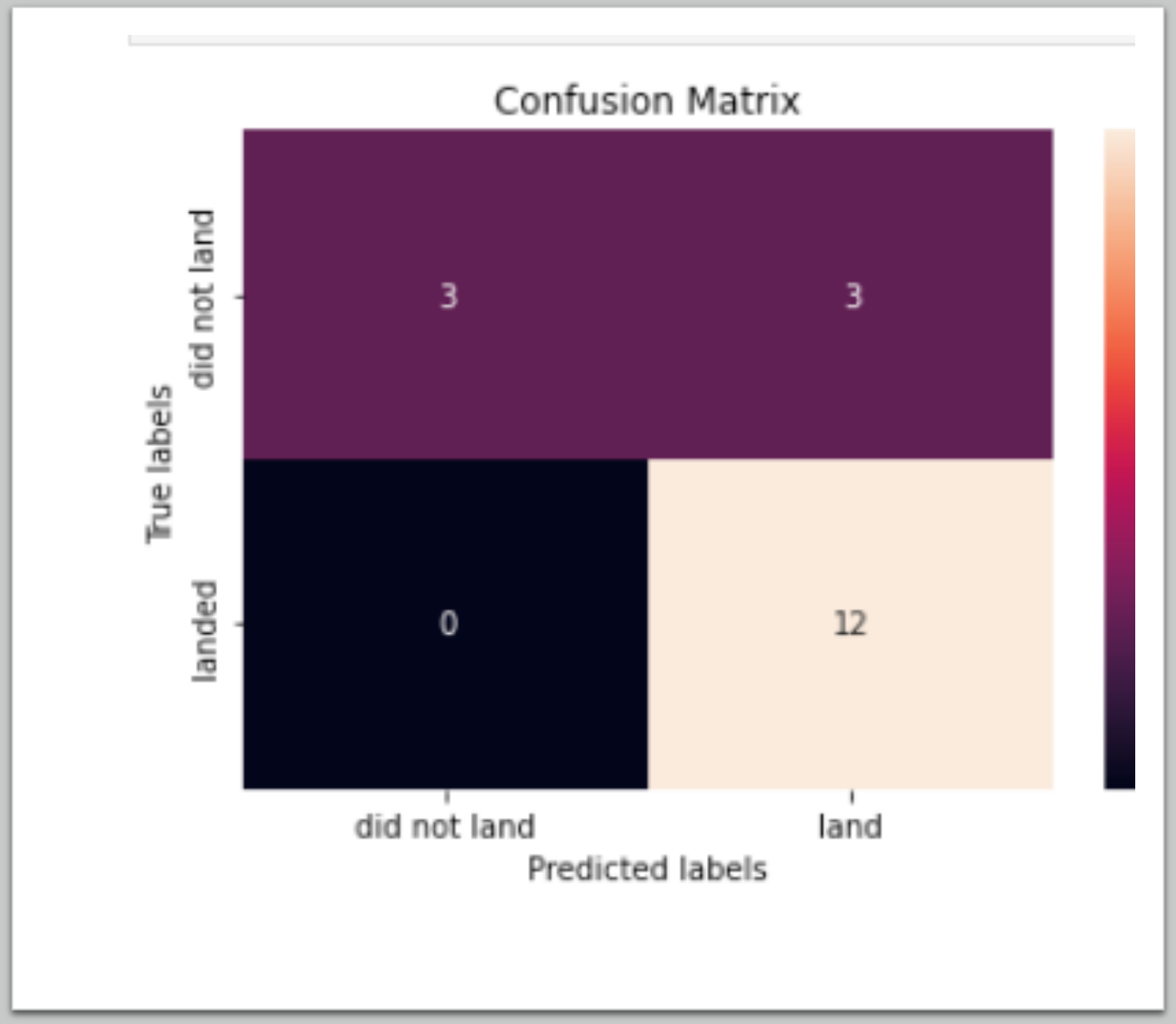
Find the method performs best:

[65]:
```
#Since, we have a small dataset, all the models are practically same and generated the same matrix.

fig = plt.figure(figsize=[5,5])
ax = fig.add_axes([0,1,1,1])
x = ["Logistic Regression","Support Vector Machine","Decision Tree Classifier","K-Nearest Neighbors"]
y = [logrev_cv_acc,svm_cv_acc,tree_cv_acc,knn_cv_acc]
ax.bar(x,y,width=0.5)
plt.xticks(rotation=90)
plt.show()
```



Confusion Matrix

- Confusion matrix for landing outcomes displaying True positives, True negatives, False positives and False negatives
- We notice that False positives (12) are more in this matrix



Conclusions

- From our analysis we can conclude the following from the SpaceX Data:
  - Launch Sites are in closer proximities to coasts.
  - Successful Launches have kept increasing since 2013
  - A total of 8 successful landing outcomes between 2010 and 2017, where 5 boosters landed on drone ship and 3 on ground pad
  - Booster Version FT have more success rates for certain payloads and v1.1 have the least success rates
  - Launch Site KSC LC-39A has the highest success rate.
  - Machine learning models we trained have an accuracy of 83.3%, which can be considered a good model, all models performed the same due to the small dataset

Appendix

- Sample Dash Plotly Code for Callback functions
- GitHub Repo Link for the complete capstone: https://github.com/bhavyasd/IBM-DataScience-Certificate/tree/main/Applied%20Data%20Science%20Capstone

```
p.callback(Output(component_id='success-pie-chart', component_property='figure'),
           Input(component_id='site-dropdown', component_property='value'))
 get_pie_chart(entered_site):
 filtered_df = spacex_df
 if entered_site == 'ALL':
     fig = px.pie(filtered_df, values='class',
                  names='Launch Site',
                  title='Total Success Launches ALL Sites')
     return fig
 else:
     filtered_df = spacex_df[spacex_df['Launch Site'] == entered_site]
     filtered_df1 = filtered_df.groupby(['Launch Site', 'class']).size().reset_index(name='class count')
     fig = px.pie(filtered_df1, values='class count', names='class',
                  title=f'Total Success Launch for Site:' + entered_site)
     return fig

ASK 4:
dd a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as ou
p.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
           [Input(component_id='site-dropdown', component_property='value'),Input(component_id="payload-slider",
 get_scatter_plot(entered_site,payload_slider):
 if entered_site == 'ALL':
     data_section = spacex_df[(payload_slider[0]<spacex_df['Payload Mass (kg)']) & (spacex_df['Payload Mass (kg)
     fig = px.scatter(data_section, x='Payload Mass (kg)', y='class',
     color="Booster Version Category",title='Payload Mass vs Total Successful Launches')

     return fig
 else:
     # return the outcomes piechart for a selected site
     filtered_df_site = spacex_df[spacex_df['Launch Site'] == entered_site]
     data_section = filtered_df_site[(payload_slider[0]<filtered_df_site['Payload Mass (kg)']) & (filtered_df_si
     fig = px.scatter(data_section, x='Payload Mass (kg)', y='class',
     color="Booster Version Category",title=f'Payload Mass vs Total Successful Launches for site {entered_site}'
     return fig
```

Thank you!