Winning Space Race with Data Science

Subhransu Sekhar Mohanty

30/01/2022

Outline

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

Executive Summary

- Data was collected using APIs and webscrapping. Then datasets were cleaned and went through exploratory data analysis to extract valuable insights from them. Then datasets were feature engineered to be fed into ML classifers. Train models performed decently and produced somewhat accurate results.

- We observed that missions are more likely to be successful each upcoming year due to newer booster technologies. Missions to orbits that are farther to earth have higher success rate. Proximity of the launchsite to the coastline is a factor for mission success. Payload mass positively correlates with success rate.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select mission_outcome,count(mission_outcome)
from spacex
group by mission_outcome;
```

```
 * postgres://postgres:***@localhost:5432/expdb
4 rows affected.
```

mission_outcome	count
Success (payload status unclear)	1
Success	98
Success	1
Failure (in flight)	1

This sql query showed the total value counts of each mission outcomes.

Introduction

This project is created to help create a new space startup named SpaceY which will compete SpaceX. We are thus using past launch records of Falcon-9 rocket to get insights from for our new startup.

We are finding which factors affect mission success in which way. We also want to find out long term trends in the data.

Section 1

Methodology

Methodology

- Data collection
  - API and Web-scrapping
- Perform data wrangling
  - Cleaning, Imputation, Feature Engineering
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning, evaluating classification models

Data Collection

Some data sets were collected by web-scrapping. We used BeautifulSoup

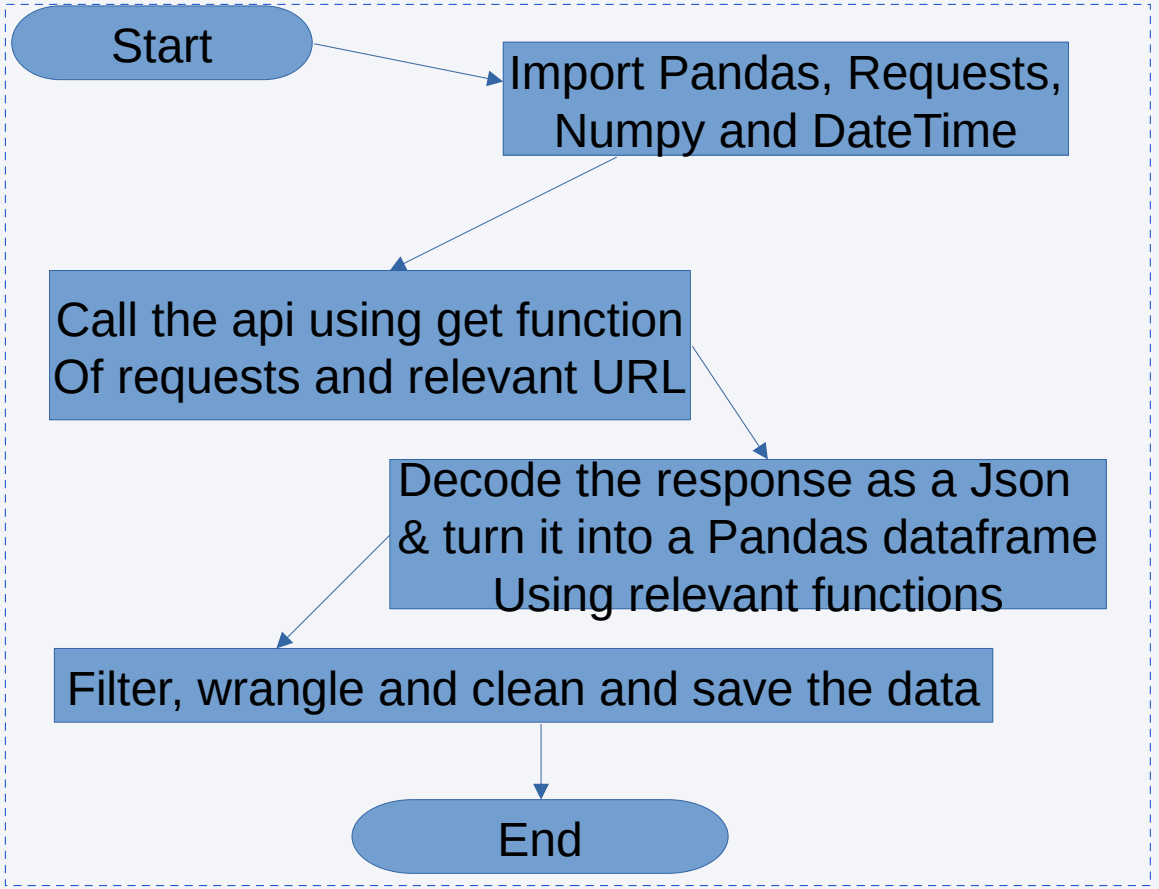
Package to parse the HTML contents which were downloaded using the get function of requests library.

Other datasets were collected using SpaceX APIs.

Now let’s move on to see how datasets were collected using both the processes.

Data Collection – SpaceX API

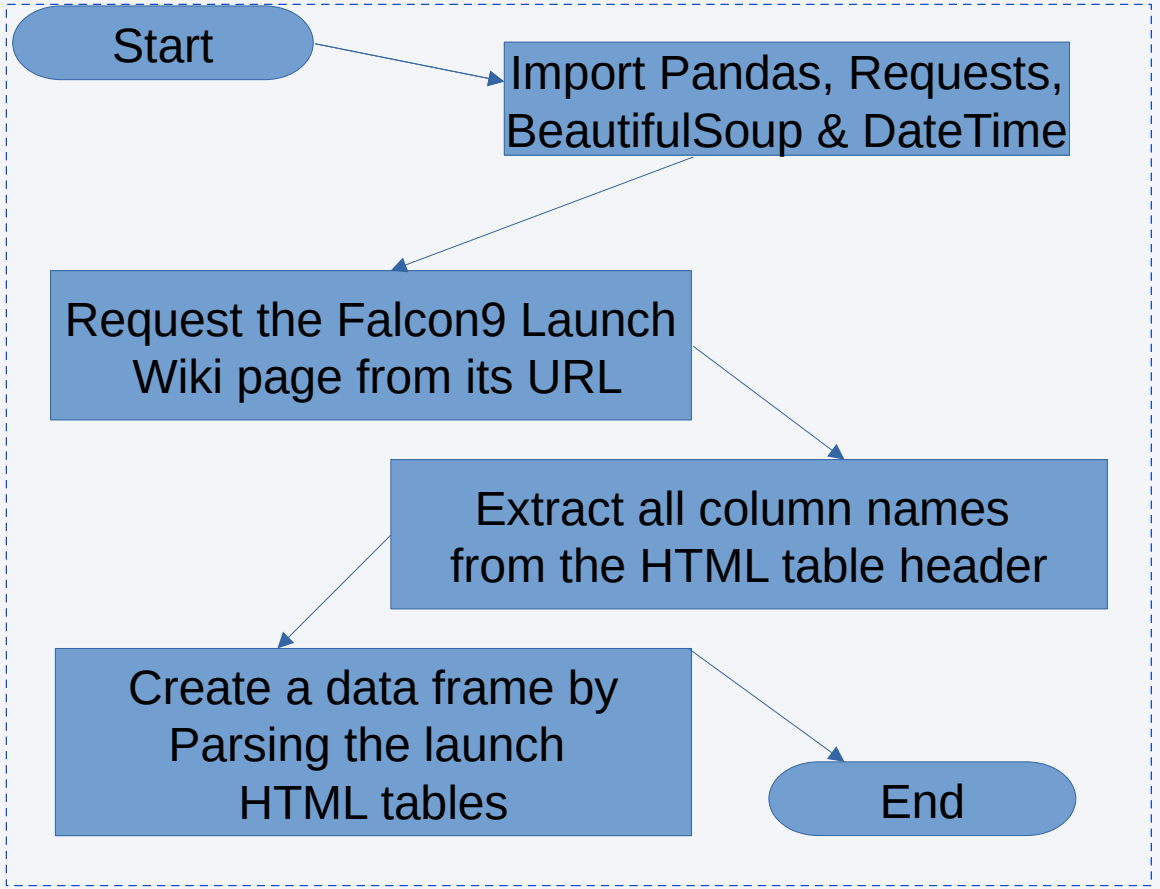
- It is the visualized data collection process with SpaceX REST API.
- This is the GitHub URL of the completed SpaceX API calls notebook.



Data Collection - Scraping

- This flowchart represents how we scrapped the wikipedia webpage and got the desired dataframe.

- This is the GitHub URL of my web scraping notebook.

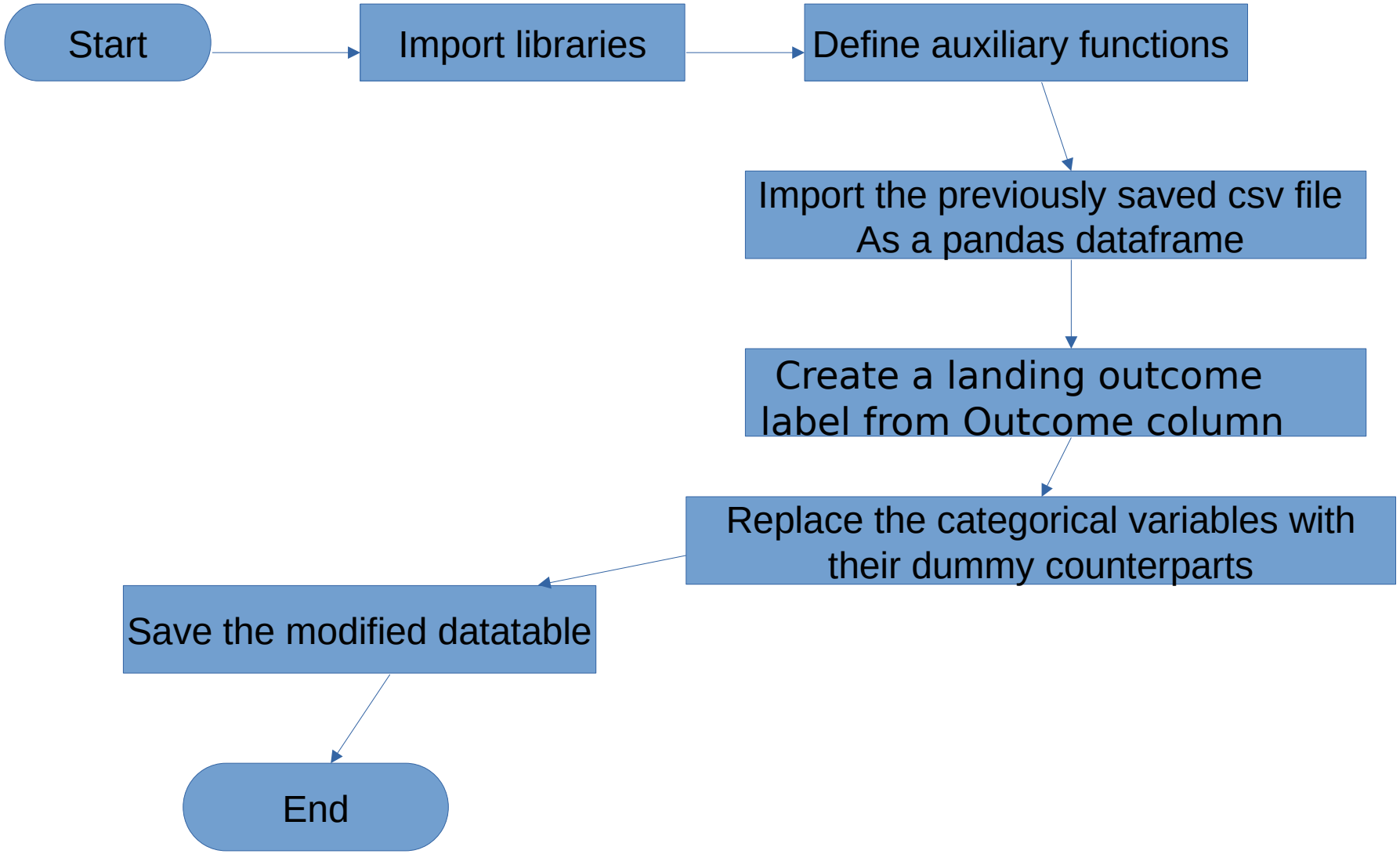


Data Wrangling

- Import Libraries and Define Auxiliary Functions
- Import the data-frame from the previously saved csv file
- Identify Column names and their types
- Create a landing outcome label from Outcome column
- Determine the average success rate from the mean of
- landing outcome column
- Create dummy variables from categorical variables and
- remove the original categorical columns
- Save the modified dataframe

This is the GitHub URL of my completed data wrangling notebooks as an external reference.

Data Wrangling Flowchart



EDA with Data Visualization

There are 2 previously plotted scatterplots in the notebook. The significance of drawn graphs are given below.

The first scatterplot visualizes the relationship between flight number and launch sites for both successful and failed missions.

The second scatterplot depicts the relationship between payload mass and launch sites for both successful and failed missions.

The column graph shows the relationship between success rate of each orbit type.

The third scatterplot represents the relationship between flightNumber and orbit type.

The fifth scatterplot objectifies the relationship between payload mass and orbit type.

The sixth scatterplot picturizes the yearly trend of the launch success.

This is the GitHub URL of my completed EDA with data visualization notebook.

EDA with SQL

```
select unique(launch_site) from spacex;

select * from spacex where launch_site  like 'CCA%' limit 5;

select sum(payload_mass__kg_) from spacex where customer = 'NASA (CRS)';

select avg(payload_mass__kg_) from spacex where booster_version='F9 v1.1';

select min(date) from spacex where landing__outcome='Success (ground pad)';

select unique(booster_version) from spacex where landing__outcome='Success (drone ship)' and payload_mass__kg_ between 4000
and 6000;

select mission_outcome,count(mission_outcome) from spacex group by mission_outcome;

select unique(booster_version) from spacex where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex);

select booster_version,launch_site from spacex where year(date)=2015 and landing__outcome='Failure (drone ship)';

select landing__outcome,count(landing__outcome) as counts from spacex where date between '2010-06-04' and '2017-03-20' group
by landing__outcome order by counts desc
```

This is the link to the notebook containing SQL queries and results.

Build an Interactive Map with Folium

We added circles and markers on all launch sites on the world map using the coordinates of launch-sites.

For each launches in the data frame,then we, added a folium.Marker to marker cluster and we distinguished the mission success of launch records by using green markers for successful mission outcomes and red ones for failed missions.

Then we marked down a point on the closest coastline using '**MousePosition**' and calculated the *distance* between the coastline point and the launch site.

After obtaining its coordinate, create a folium.Marker to show the distance and then we drew a PolyLine between a launch site to the selected coastline point.

This is the link to the notebook containing the codes and result of interactive map with folium.

Building a Dashboard with Plotly Dash

We added a piechart and a scatter plot to our **dashboard** along with a dropdown and a range slider.

The dropdown menu contained options to whether the select the data of all launch sites or just of a single individual site. The range slider lets the user to select the range of payload mass to filter the data.

The piechart shows the total successful launches by site when “ALL” option is selected in the dropdown and total successful and failed launches for a site when a particular site in selected in the menu.

The scatter plot shows the relationship between payload mass and mission success of different booster version categories for selected payload mass range and lauch site(s).

This is the GitHub URL of my completed Plotly Dash lab.

Predictive Analysis (Classification)

Scikit-Learn library was used to create ML classifier models for this capstone.

We used the previously wrangled version of data(cleaned and feature engineered) to train our models.

Then we use a standard scaler to normalize all the variables and their values.

To avoiding over-fitting we first split the data to test and train data with random state of 2 and the test to train ratio of 0.2.

We used K-nearest neighbor, decision tree, support vector machine and logistic regression models in this predictive analysis.

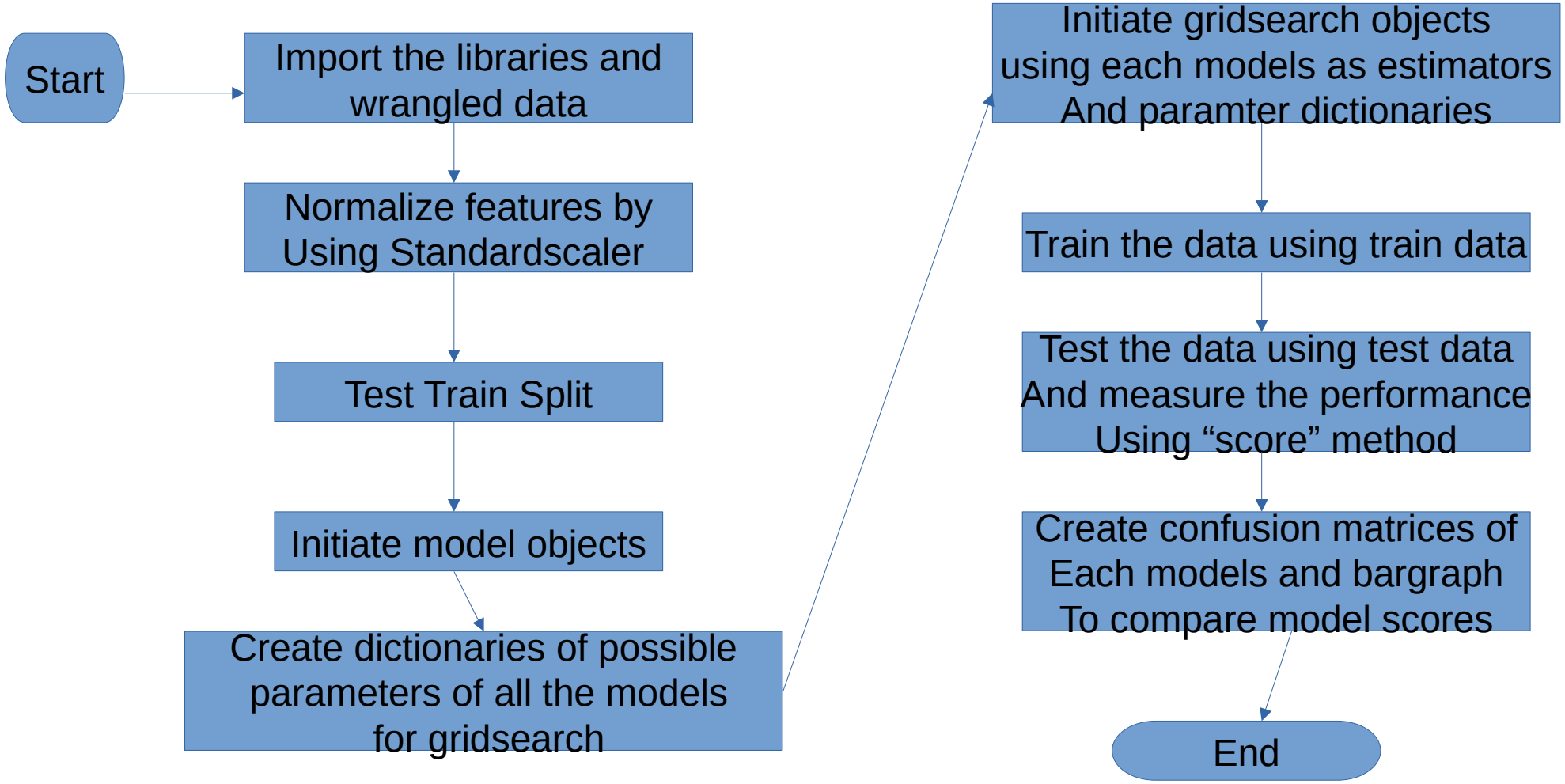
We created grid searches for each model to find best parameters for them.

We used train data to train the models and test data to get the accuracy which was the performance metric.

Upon testing we found that all the 4 models performed equally well when using the particular test set and sets of parameters.

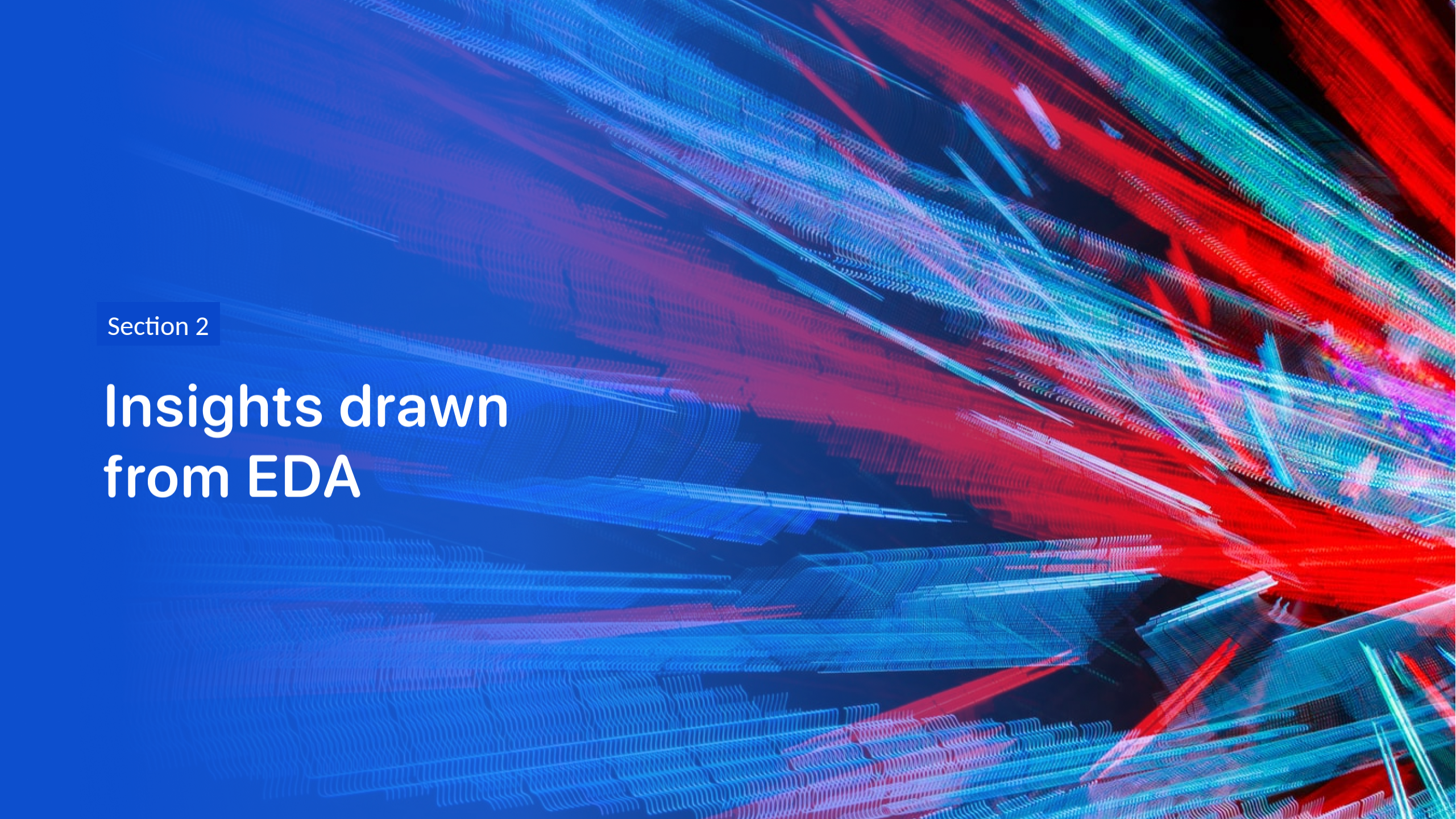
This the link to my notebook in my project capstone repository in Github containing ML codes and results and texts.

Predictive Analysis Flowchart

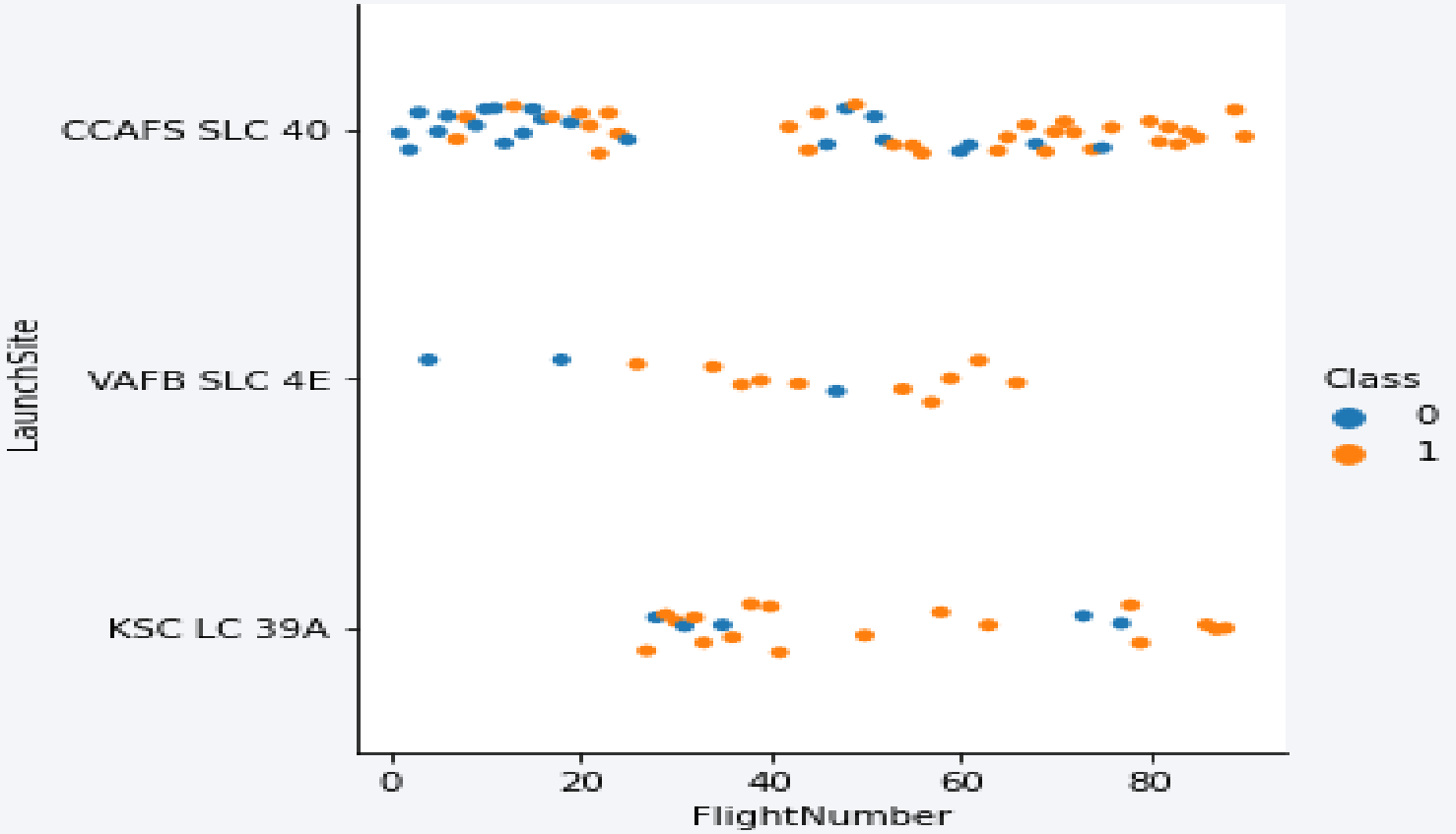


Results

- We explored the data using SQL queries and visualization and found out the trends and patterns hidden in the data.
- We have created interactive dashboard to view the effect of various factors on the mission success.
- We created 4 ML classifier models that can predict the probability of mission success upon providing the relevant features. These models perform adequately to roughly estimate the mission success for our startup.

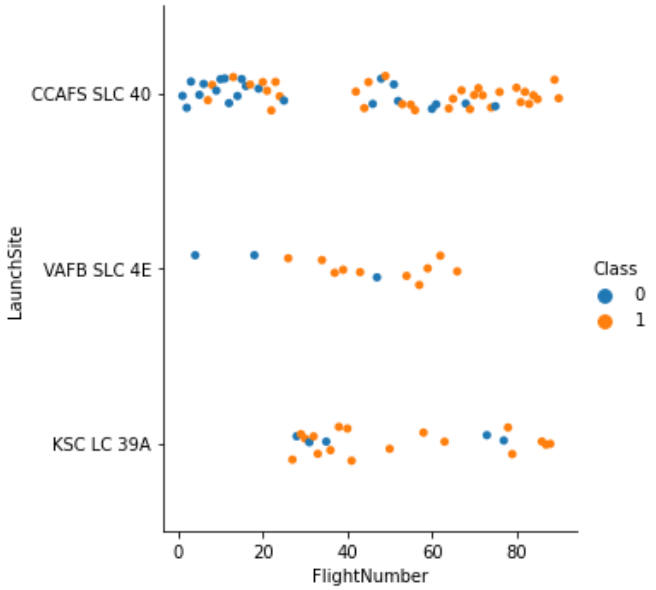

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



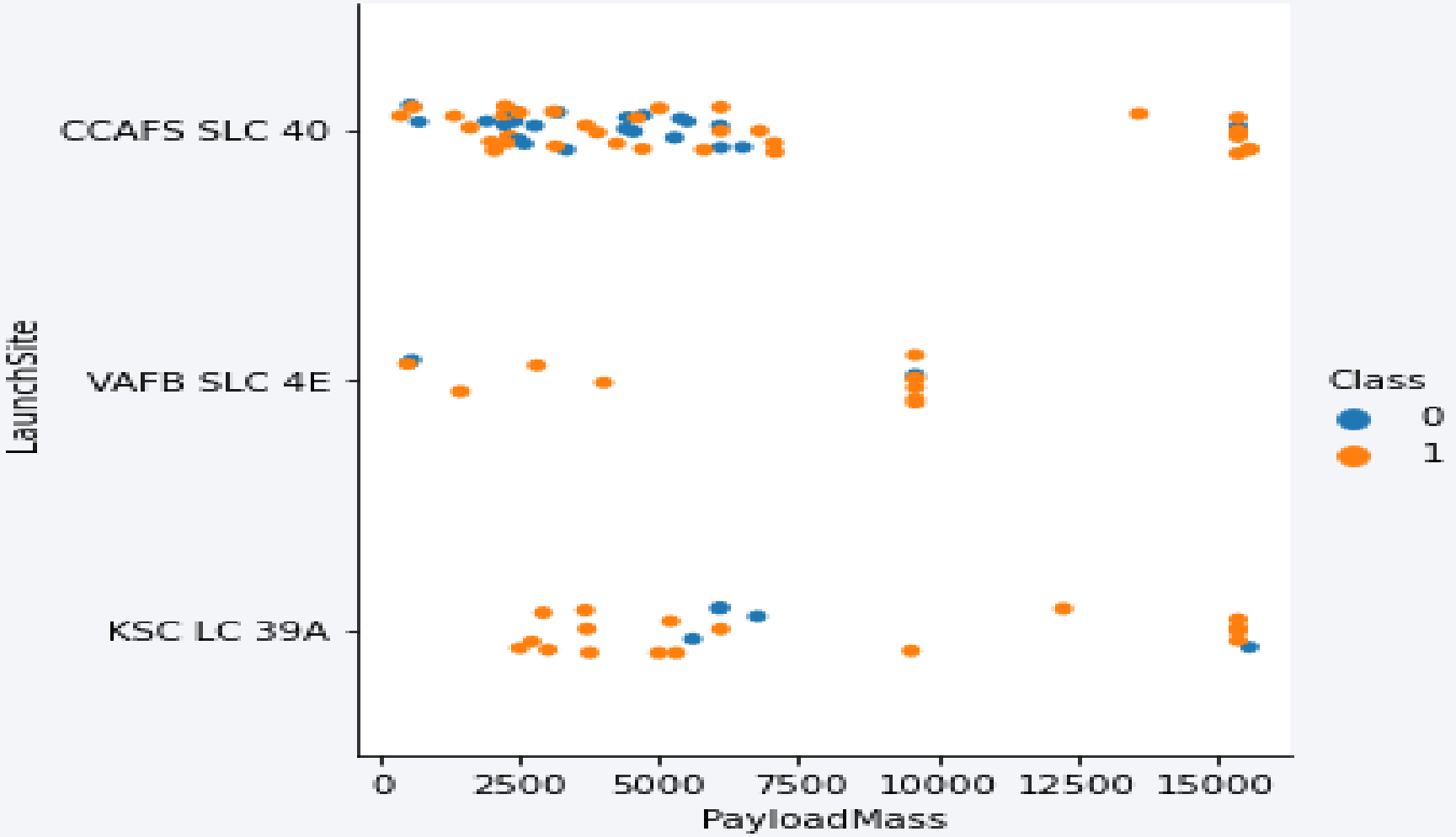
Flight Number vs. Launch Site

```
n [5]:
# Plot a scatter point chart with x axis to be Flight Number and y axis to b
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df)#, aspect
plt.show()
```



- This scatter plot shows the relationship between flight number and launch site for both successful and failed missions.
- Datapoints of successful missions are orange in colour and datapoints of failed missions are blue in colour.
- The horizontal axis shows the flight number.
- The vertical axis shows the launch site.

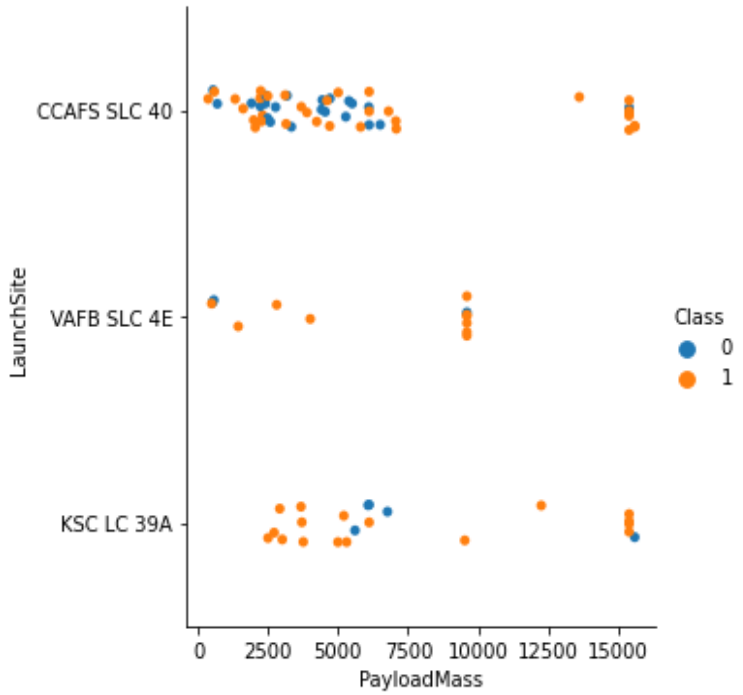
Payload vs. Launch Site



Payload vs. Launch Site

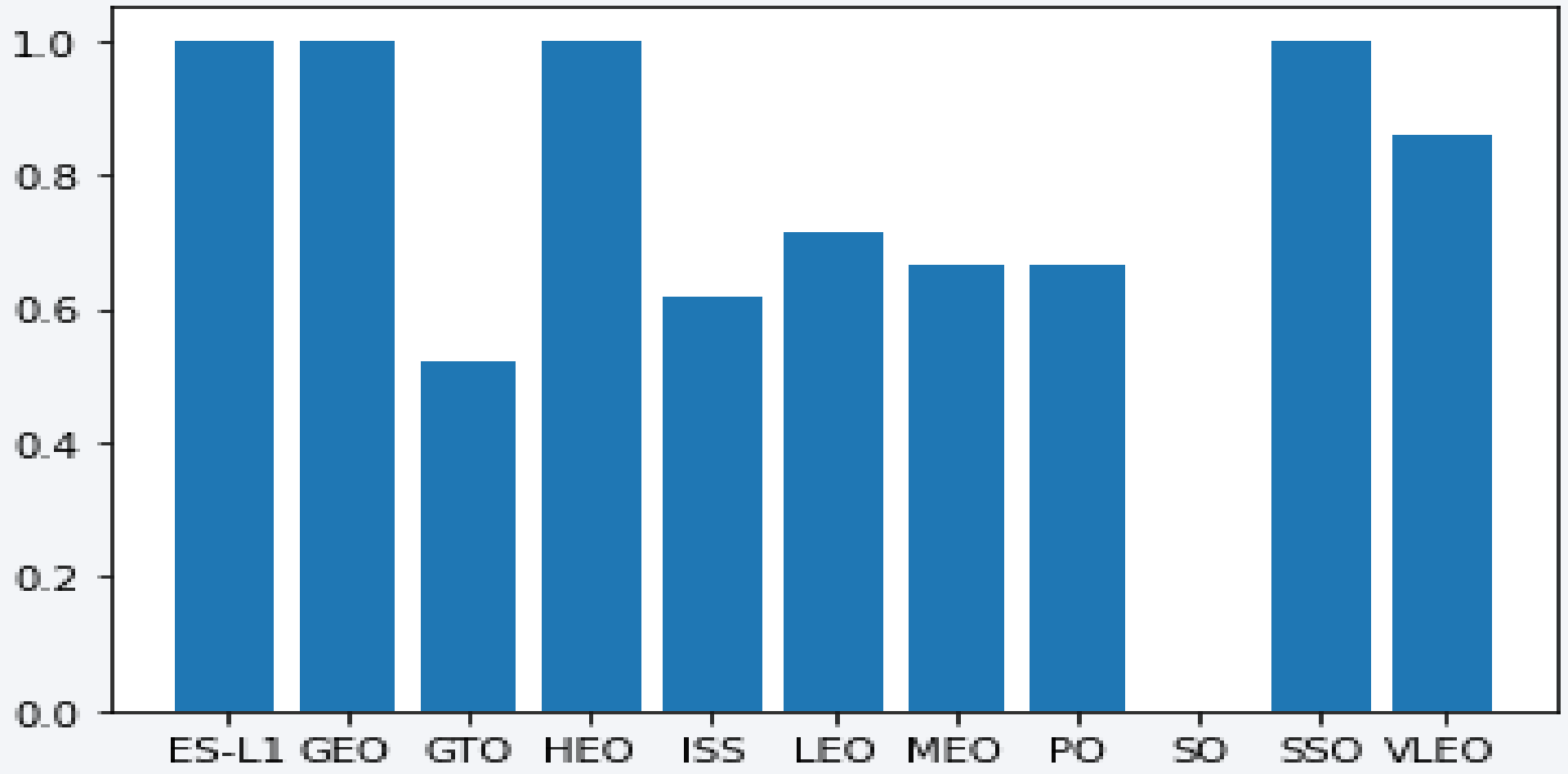
In [6]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg)
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df)
plt.show()
```



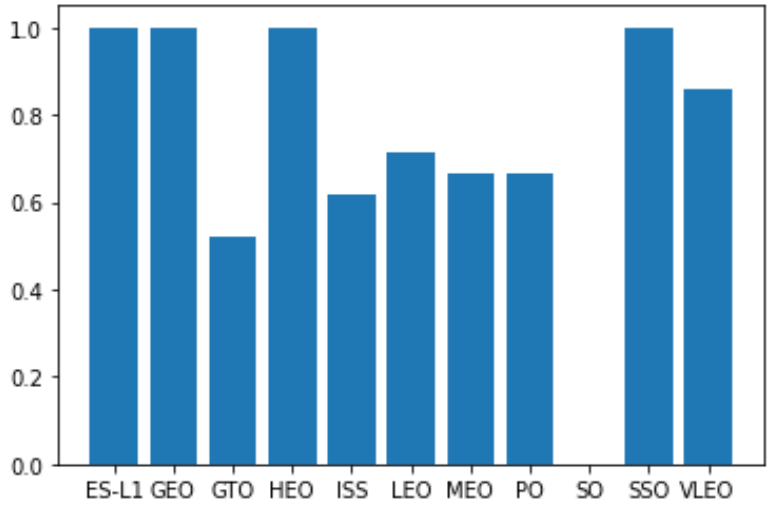
- This scatter plot shows the relationship between payload mass and launch site for both successful and failed missions.
- Datapoints of successful missions are orange in colour and datapoints of failed missions are blue in colour.
- The horizontal axis shows the payload mass(in KG).
- The vertical axis shows the launch site.

Success Rate vs. Orbit Type



Success Rate vs. Orbit Type

```
7]:
# HINT use groupby method on Orbit column and get the mean c
dfb2 = df['Class'].groupby(df['Orbit']).mean()
dfb2=pd.DataFrame([dfb2.index.to_list(),list(dfb2.values)])
dfb2=dfb2.transpose()
plt.bar(x=dfb2[0],height=dfb2[1])
plt.show()
#dfb2.columns = ["Orbit","SuccessRate"]
```

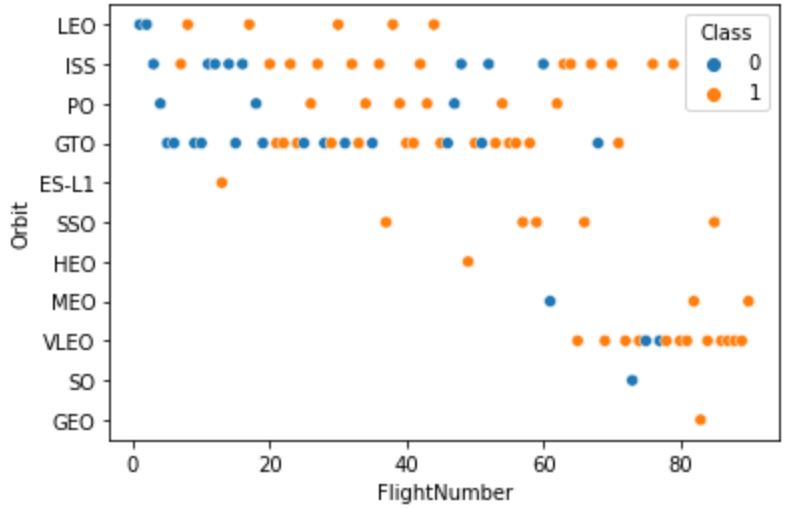


- This column chart shows the average success rate of space missions launched to different types of orbits.
- Those orbital targets are Sun Synchronous Orbit, Polar Orbit, International Space Station, (High/Low/Middle/Very-Low) Earth Orbit, Geostationary Orbit, Geostationary Transfer Orbit and First Lagrangian Point(ESA L1).

Flight Number vs. Orbit Type

```
[8]: # Plot a scatter point chart with x axis to be FlightNumber and
     sns.scatterplot(x='FlightNumber',y='Orbit',data=df,hue='Class')
```

```
[8]: <AxesSubplot:xlabel='FlightNumber', ylabel='Orbit'>
```



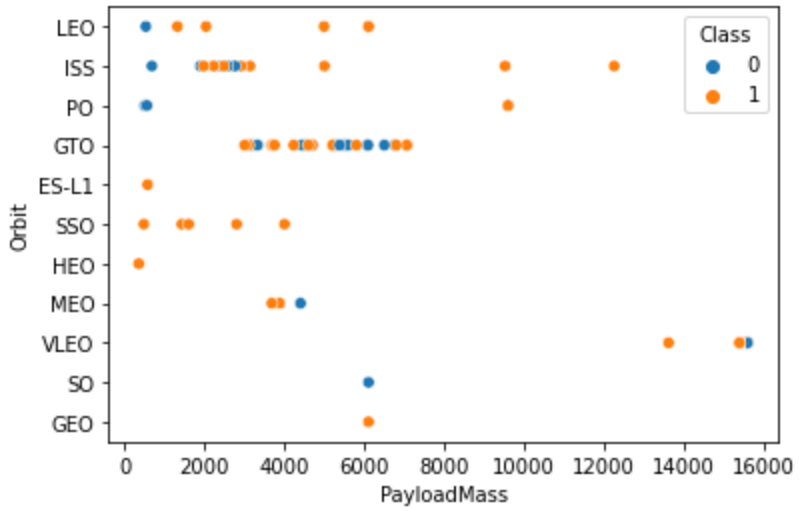
This scatter plot depicts the relationship between flight number and the orbit type.

Here orange data points represent successful missions while blue ones represent failed missions.

Payload vs. Orbit Type

```
[9]: # Plot a scatter point chart with x axis to be Payload and y ax.
     sns.scatterplot(x='PayloadMass',y='Orbit',data=df,hue='Class')
```

```
[9]: <AxesSubplot:xlabel='PayloadMass', ylabel='Orbit'>
```

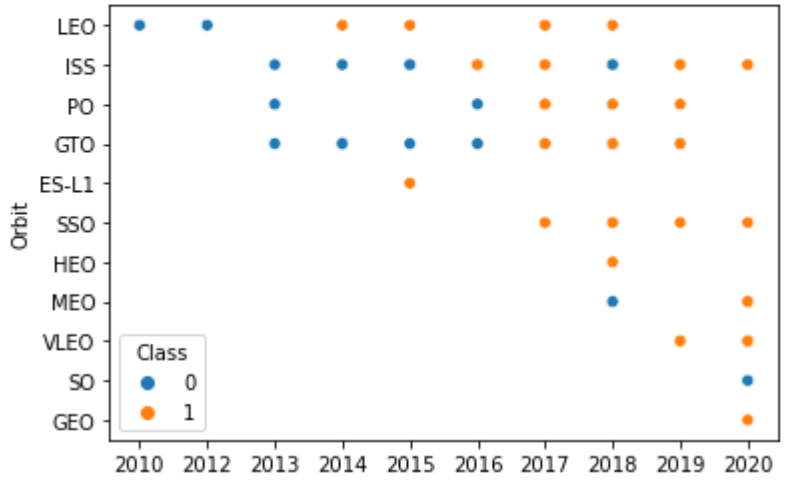


- This scatter plot depicts the relationship between pay load mass and the orbit type.
- Here orange data points represent successful missions while blue ones represent failed missions.
- The payload mass given here in the horizontal axis is in the unit of Killogram.
- We can see missions with heavier payloads are more likely to be successfull.
- In case of lighter payloads, 1st Lagrangian point, Sun Synchronous Orbit and High Earth Orbit mission are more likely be successful.

Launch Success Yearly Trend

```
12]:  # Plot a line chart with x axis to be the extracted year
      sns.scatterplot(x=extyr,y=df['Orbit'],hue=df['Class'])
```

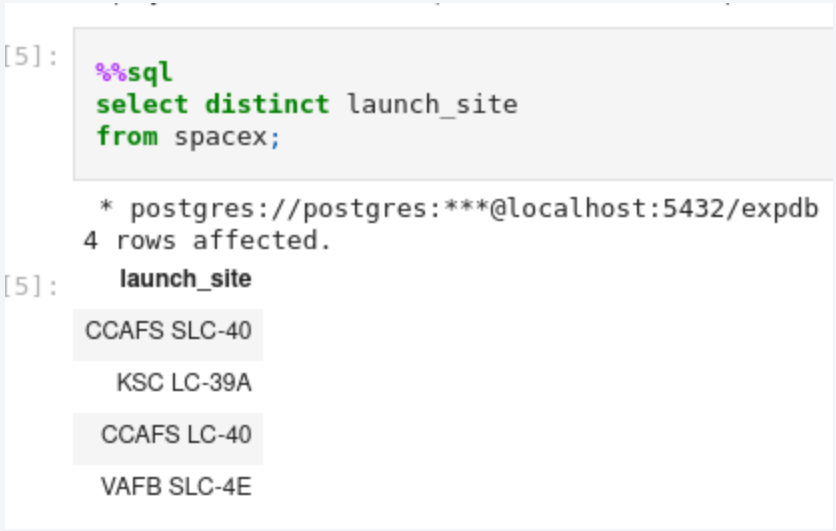
```
12]: <AxesSubplot:ylabel='Orbit'>
```



This graph shows the average success rate of different orbital destinations over the year.

We can see a positive trend here where space launches are more likely to be successful in each successive year than the previous one in all orbit types.

All Launch Site Names



```
[5]:
%%sql
select distinct launch_site
from spacex;
```

```
 * postgres://postgres:***@localhost:5432/expdb
4 rows affected.
```

[5]:

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

This query seeks the names of the unique launch sites in the space mission.
And we can see the 4 sites as result.

Launch Site Names Begin with 'CCA'

```
%%sql
select *
from spacex
where launch_site  like 'CCA%'
limit 5;
```

```
 * postgres://postgres:***@localhost:5432/expdb
5 rows affected.
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

This query seeks 5 records where launch sites begin with the string 'CCA' and we can see the result above.

Total Payload Mass

```
%%sql
select sum(payload_mass__kg_)
from spacex
where customer = 'NASA (CRS)';
```

```
 * postgres://postgres:***@localhost:5432/expdb
1 rows affected.
```

sum
45596

This query resulted in the total payload mass carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

```
]:
%%sql
select avg(payload_mass__kg_)
from spacex
where booster_version='F9 v1.1';
```

```
 * postgres://postgres:***@localhost:5432/expdb
1 rows affected.
```

]:

avg
2928.4000000000000000

This SQL query showed the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%%sql
select min("date_") as SucceedLanding1st
from spacex
where "Landing_Outcome"='Success (ground pad)';
```

```
 * postgres://postgres:***@localhost:5432/expdb
1 rows affected.
```

succeedlanding1st
2015-12-22

This query got us date when the first successful landing outcome in ground pad was acheived.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select distinct booster_version
from spacex
where "Landing_Outcome"='Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;
```

```
 * postgres://postgres:***@localhost:5432/expdb
4 rows affected.
```

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

This postgres query listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Boosters Carried Maximum Payload

```
%%sql
select distinct booster_version
from spacex
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex);
```

```
 * postgres://postgres:***@localhost:5432/expdb
12 rows affected.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

This query provides us the list of the booster_versions which have carried the maximum payload mass.

2015 Launch Records

```
%%sql
select booster_version,launch_site
from spacex
where extract(year from spacex."date_")=2015 and "Landing_Outcome"='Failure (drone ship)';
```

```
 * postgres://postgres:***@localhost:5432/expdb
2 rows affected.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

This query lists the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

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

```
%%sql
select "Landing_Outcome",count("Landing_Outcome") as counts
from spacex
where spacex.date_ between '2010-06-04' and '2017-03-20'
group by "Landing_Outcome"
order by count("Landing_Outcome") desc
```

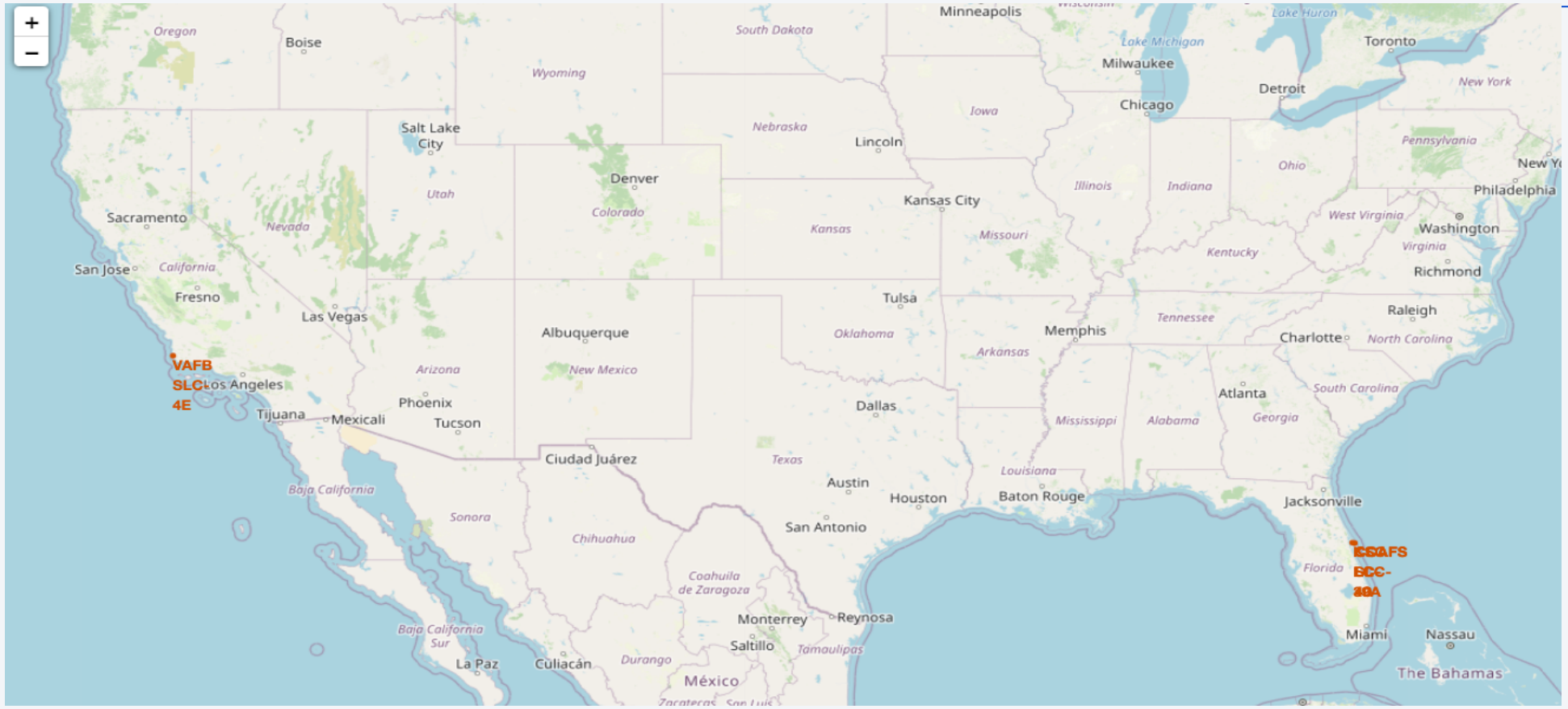
```
 * postgres://postgres:***@localhost:5432/expdb
8 rows affected.
```

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

This query ranks the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

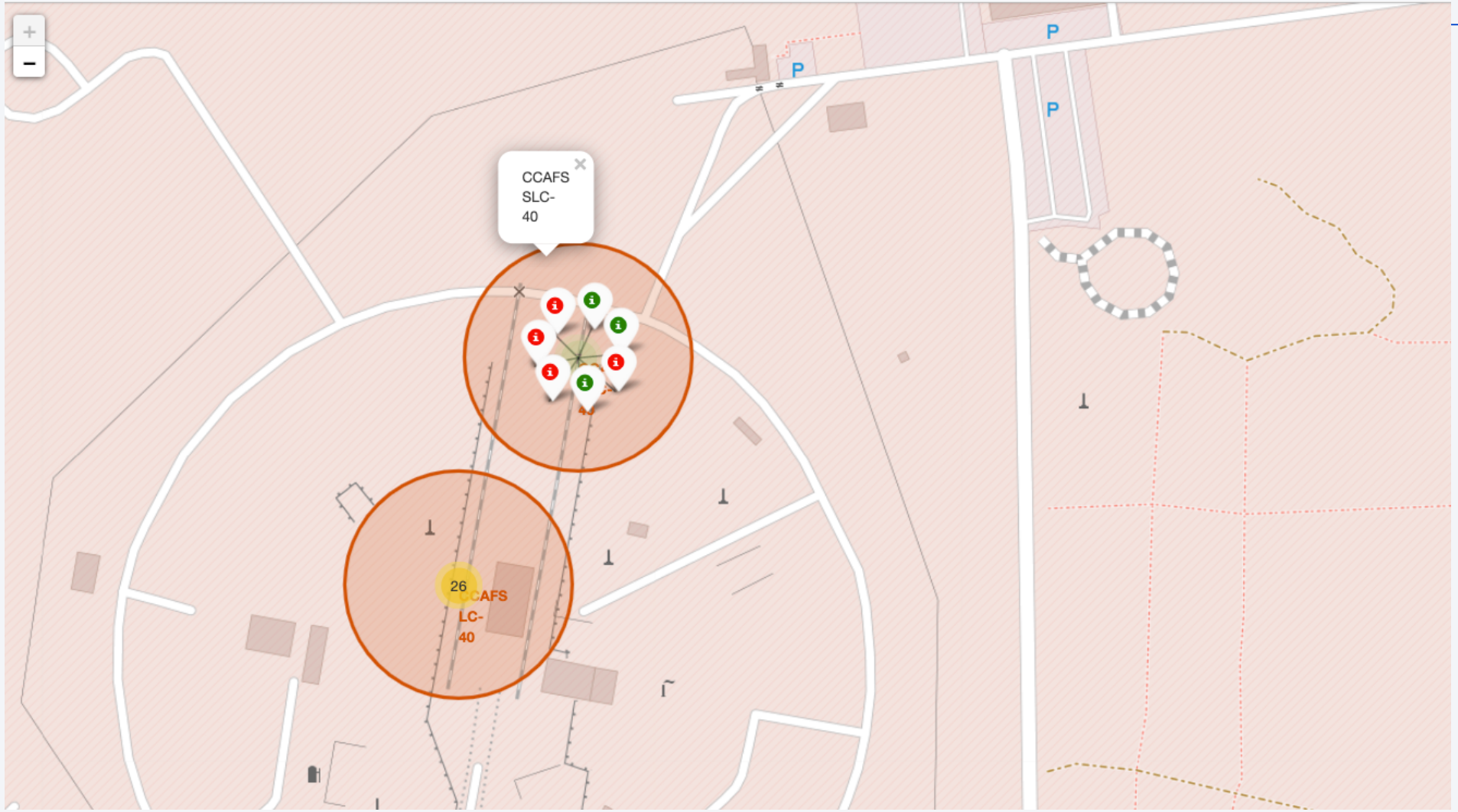

Section 3
Launch Sites
Proximities Analysis

Launch Sites of SpaceX(on Folium Map)



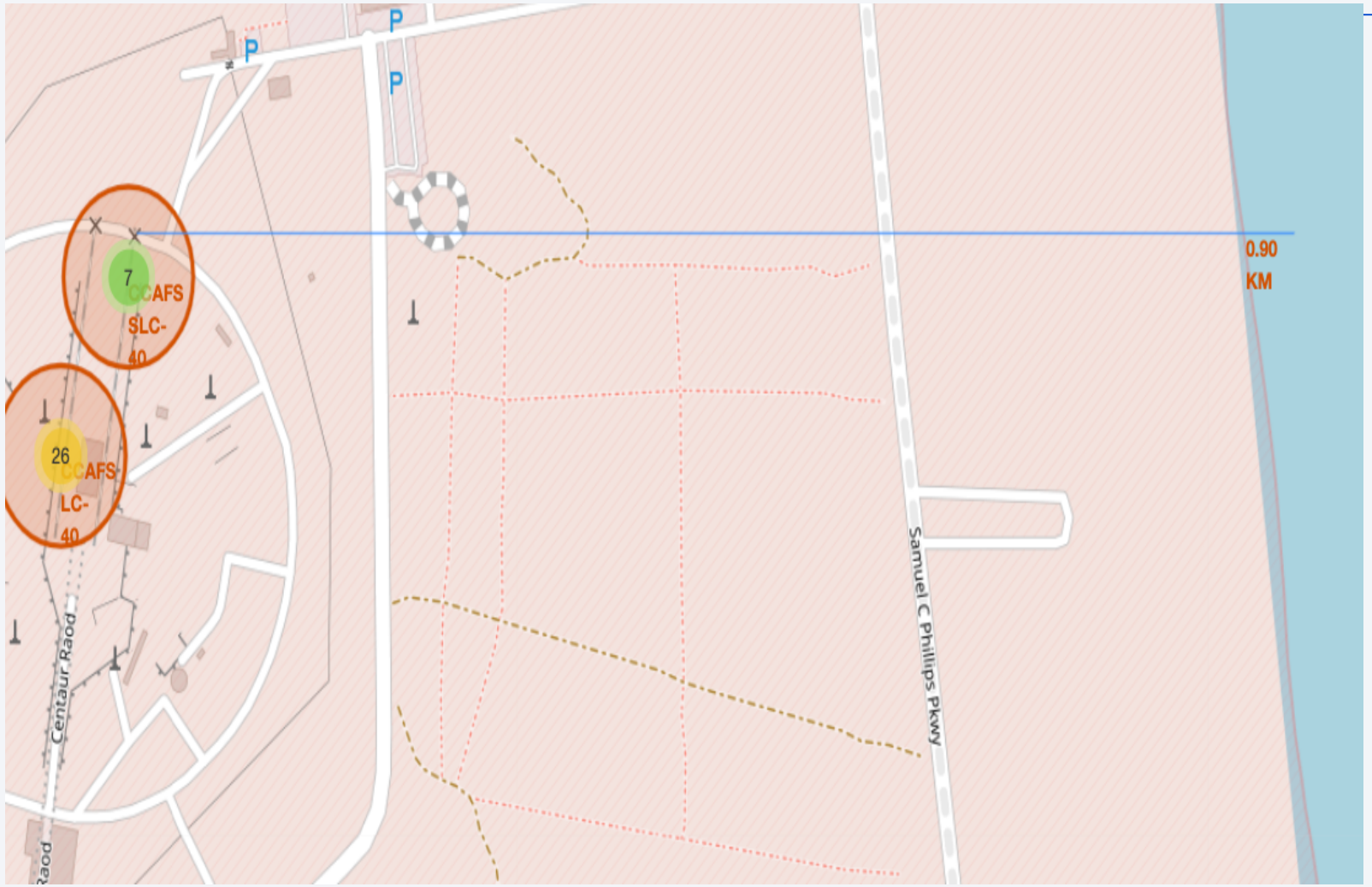
This folium map marks all launch sites

Successful and Failed launche(Folium)

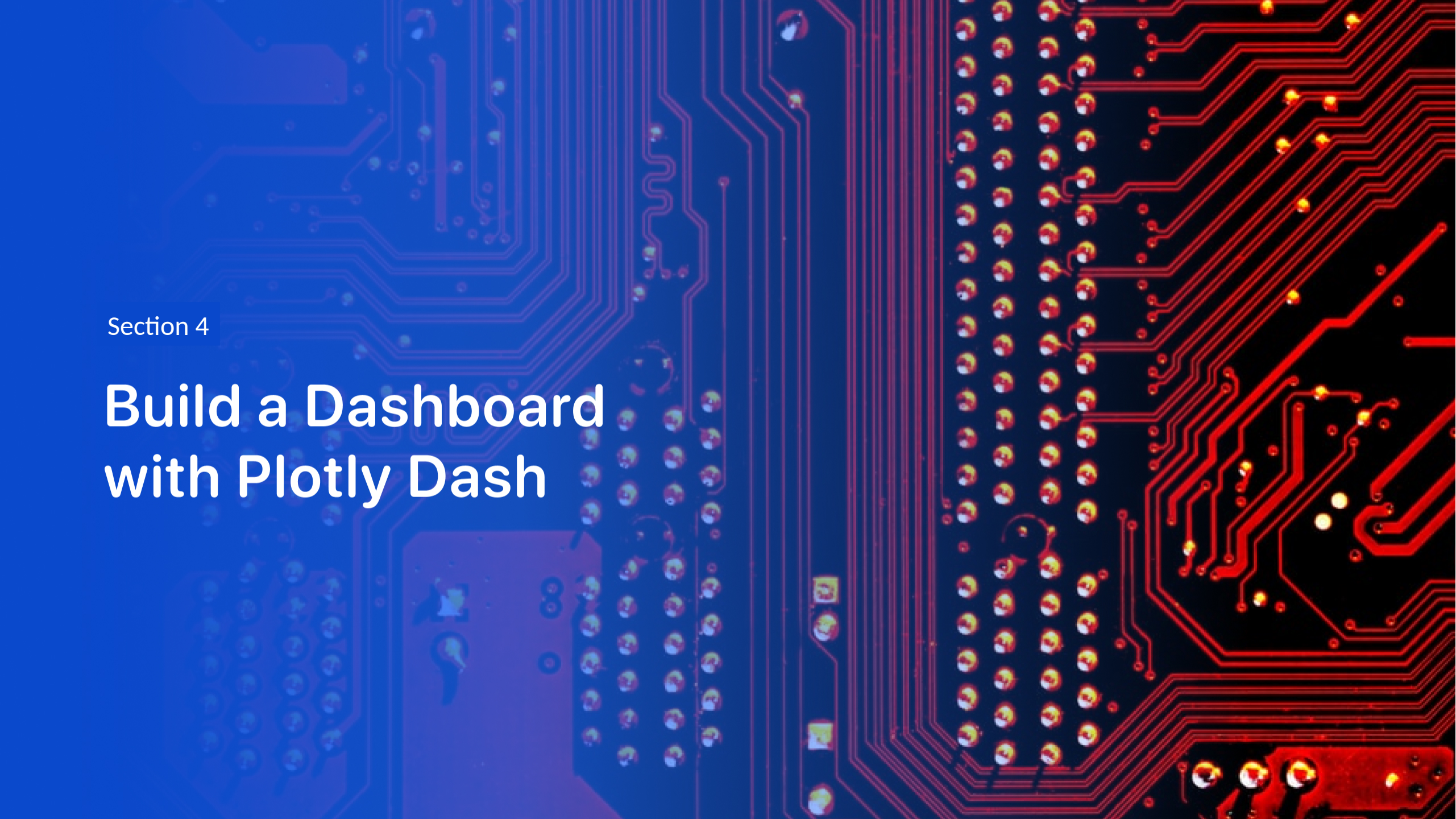


It shows the successful launches in green and failed launches in Red for CCAFB SLC-40 site.

CCAFS SLC-40 to coastline



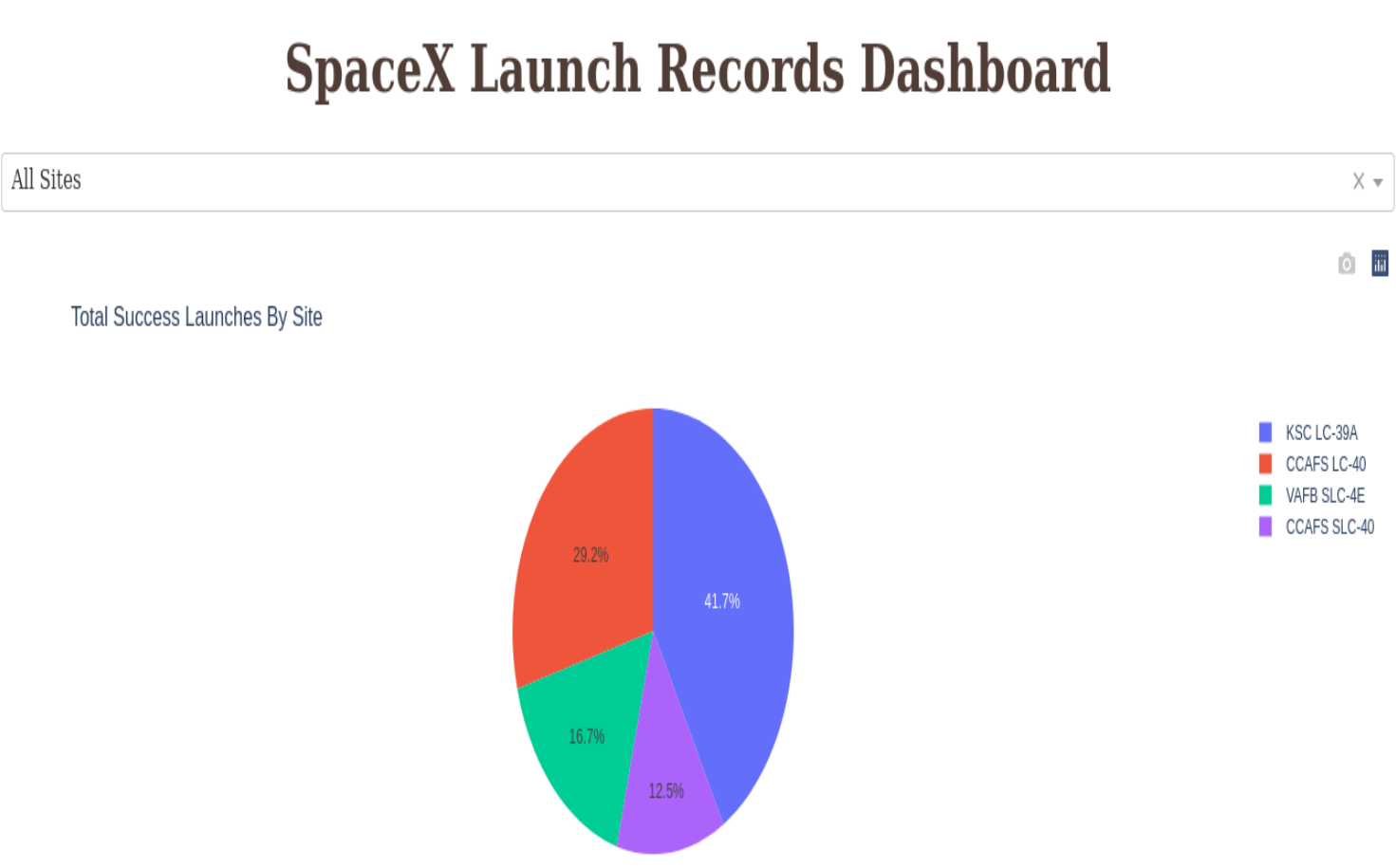
This folium map calulates the distance betSween launch site CCAFS SLC-40 to its nearest coastline point.



Section 4

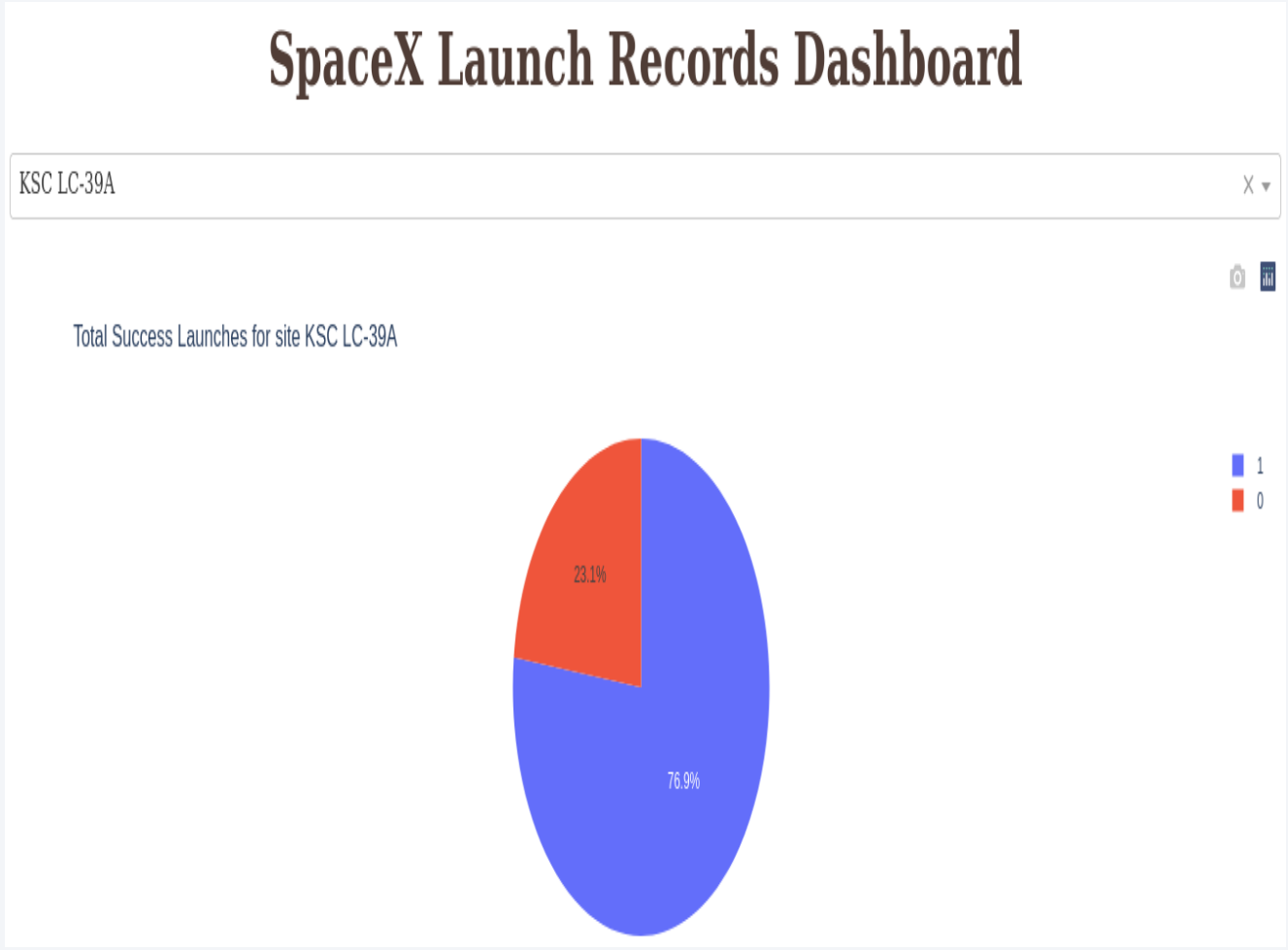
Build a Dashboard with Plotly Dash

Launch success rate of different launch sites



Here it is to be noted that this graph shows the number of successful launches from each launch sites not the success rate of the launch site.

Launch site with the highest success rate



KSC LC-39A has the highest success rate among all spaceX launchsites.

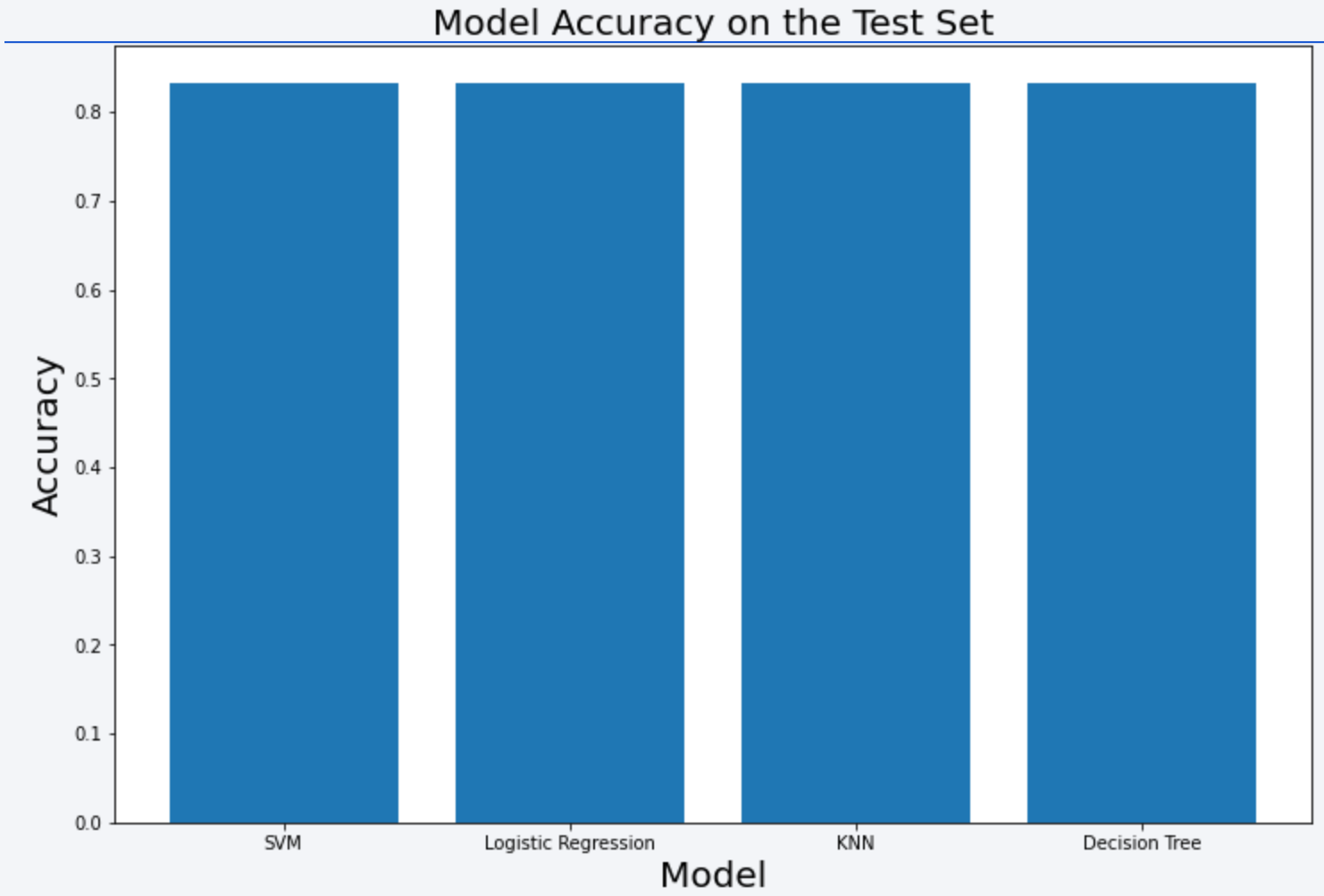
Payload vs Launch Outcome

We can see here that ‘FT’ booster version category has higher success rate.

Section 5

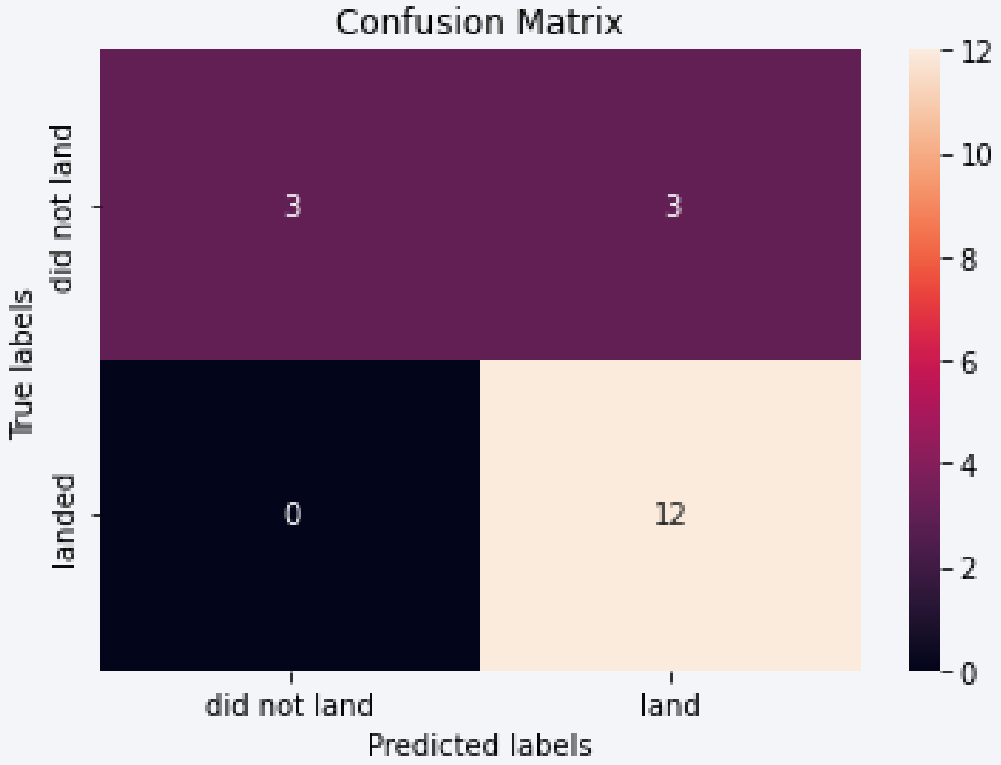
Predictive Analysis (Classification)

Classification Accuracy



As we can see, all the models perform equally well on test sets.

Confusion Matrix



This is the confusion matrix of KNN model as well as of others too since all of them performed equally upon being given the testset.

Conclusions

- Mission success rate is improving each year as newer and more advanced rocket technology
- coming to the market.
- Space missions to geostationary orbit, high earth orbit, sun sun synchronous orbit and 1st
- Lagrangian point are very likely to be successful.
- Proximity of the launchsite to the coastline is a factor for mission success.
- Newer version rocket boosters are very likely to make the mission successful.
- Heavier payloads are more likely to make a successful mission.

Appendix

This is the link to this project capstone repository of mine.

Links to the websites of the software packages used are given below.

https://scikit-learn.org/stable/

https://python-visualization.github.io/folium/

https://plotly.com/

https://pandas.pydata.org/

https://matplotlib.org/

https://seaborn.pydata.org/

https://www.libreoffice.org/

https://colab.research.google.com/

Thank you!