IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

OBINNA IBE

SEPTEMBER 29TH, 2022

Outline

1. Executive Summary
2. Introduction
    - Background of the study
    - Statement of problem
    - Question(s) to answer
3. Methodology
    - Data collection using API request to webscrapping and BeautifulSoup.
    - Data wrangling.
    - Exploration of data using SQL.
    - Visualization of data using matplotlib and pandas.
    - Interactive map built with folium and interactive dashboard built with plotly dash.
    Machine learning prediction using SVM, DecisionTree, KNN and Logistic Regression.

3. Results

4. Discussion

5. Findings & Implications

6. Conclusion

7. Appendix

Executive Summary

- SpaceX falcon9 has built a spaceship with two stages of which stage one (the main component) has a higher probability of being reusable. This reusable tendency when successful has made the launching of spaceship cheaper than it used to be.
- To bid against SpaceX in the space world, it is important to determine the successful landing of stage one, use the specification to build a Space Y spaceship which will be used to bid against SpaceX falcon9 for a better prize.
- To do these, many analytic process will be done, and they include;
- Data collection using API and webscrapping, wrangling of data to obtain a better comprehensive data, exploration of data using SQL, visualization of data using folium map and plotly dash.
- Prediction of successful landing using machine learning.
- This whole process will help in predicting successful landing keeping in consideration the features of the spaceship and how it affect landing success. This information will help in the building of Space Y spaceship which will be used to bid against Space X.

Introduction

- BACKGROUND

- The exploration of space has been ongoing since 1961 with the first cosmonaut in the person of Yuri Gagarin landing in space. Since then, exploration of space has continued with spaceships being destroyed after launching. These destructions has made landing in space very expensive with other spaceships announcing the use of $165 million for manufacturing and launching of spaceship, while Elon Musk is announcing $65million for reusable manufacturing of spaceships since the major component is already available. The reusable spaceship has made landing in space cheap. Bidding against SpaceX will require the manufacturing of reusable spaceships too.

- THE PROBLEM STATEMENT

- What are the factors that determine the landing success of the reusable component of Space X?
- Can a prediction of successful landing be made using SpaceX falcon 9 data?
- Is there a relationship between the features of spaceship and the landing success?

- PURPOSE OF RESEARCH

- To predict the successful landing of SpaceX falcon 9 spaceship to Build Space Y which will bid against Space X.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected with rest API and beautiful soup
- Perform data wrangling
  - Data was wrangled with python
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification was used in which the 4 methods of classification was used.

Data Collection

- The REST API was used to obtain target launch data Like launch specification payload delivery, rocket used, landing specification and landing outcome.
- The URL obtained was used to obtain past launch data.
- A get request from request library was used to obtain the launch data and the result was passed to .json(),
- The data was normalized into a dataframe using to_normalize
- Link to complete process: https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/SPACEX%20data%20colLection.ipynb

Data Collection -

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

```
response    requests.get(spacex url)
```

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

e shou l- see tha the request as s11..11ccessfu ll it the .200 sta s response code

```
response.status_code
```

200

o e decode t e iresponse con ent as a Json using .j son () and turn it into a Pa das dataframe us·ng .json_normalize()

```
Use ,ison_normali.z.e eet:hod t:o convert- i::he ..Json resul.t: int:o a da-t::afra e
f.a:sP. _..R<}n4sits_._·9_._j_ PJl. • .I!IP.'° r..it. so_n n_o['_l!Ja l_i :z_
}.._ p_or.:t • 9_1!1
```

```
j=response.json()
```

```
data=pd.json_normalize(j)
```

Data Collection - Scraping

Webscrapping from Wikipedia
was done using BeautifulSoup to obtain launch dictionary. The content of the table was extracted using launch_dict= dict.fromkeys(column_names). all the column names were gradually obtained and converted to pandas DataFrame and saved as a csv file.

- full process link:https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/spacex%20webscraping.ipynb

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables=soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

- The data obtained from Beautifulsoup was wrangled using python. The following were done.
- Finding the data type, sampling the data,
- getting the value count, including an outcome column to show if the landing was successful or not and dealing with null value to obtain a more comprehensive data which will be visualized, tested and be used for prediction.

- Link to complete process: https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/DATA%20WRANGLING%20.ipynb

EDA with Data Visualization

A bar chart was plotted to show different launch site and the relationship it was with successful outcome. A bar chat was used in order to show the levels in a more visible was since the difference wasn’t all that obvious.

Scattered plot was used to show the relationship between payload and success outcome, flight number and outcome.

- Full process link: https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/SPACEX%20DATA%20VIZUALIZATION.ipynb

EDA with SQL

- The data was queried to obtain the following information

- Distinct flight
- Payload average
- Launch site with CCA
- Average payload mass carried by booster version F9 v1.1
- The date when the first successful landing outcome in ground pad was achieved
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes
- The names of the booster_versions which have carried the maximum payload mass. Use a subquery
- The records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

LINK: https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/SPACE%20sql.ipynb

Build an Interactive Map with Folium

- A markerCluster was used to identify the number of clusters in the folium map
- The map indicated the launch site and distance between launch site from other launching place.
- folium.Circle was used to add a highlighted circle area with a text label on a specific coordinate.
- FoliumMarker was used to mark the location
- MousePosition was used to get coordinate for a mouse over a point on the map
- PolyLine was used to get a line between a launch site to the selected coastline point

- https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/MAP%20INTERACTIVE%20OF%20SPACEX%20.ipynb

Build a Dashboard with Plotly Dash

- An interactive pie chart and scatter plot was done with plotly dash.
- The pie chart showed the successful rate of each distinct flight.
- A plotly dash was done to show the relationship between payload and success rate.
- The plot were done to have a total understanding of the success outcome of each distinct flight, to also determine the relationship between each flight and their payload relationship with successful outcome.

https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/DASHBOARD%20SPACEX%20PLOTLY%20DASH.py

Predictive Analysis (Classification)

- All the necessary libraries needed for classification prediction was imported
- Data class was turned into numpy with .to_numpy()
- Data standardization and transformation was done with preprocessing.StandardScaler() and fited to the data X
- Data was trained with X_train and test, Y_train and test.
- Parameters were defined
- GridSearch was used to fit in the four classification namely SVM, DecisionTree, KNN and LogisticRegression.
- .score was used to calculate the accuracy score
- The best prediction calculation was calculated with max method

- https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(2).ipynb

Results

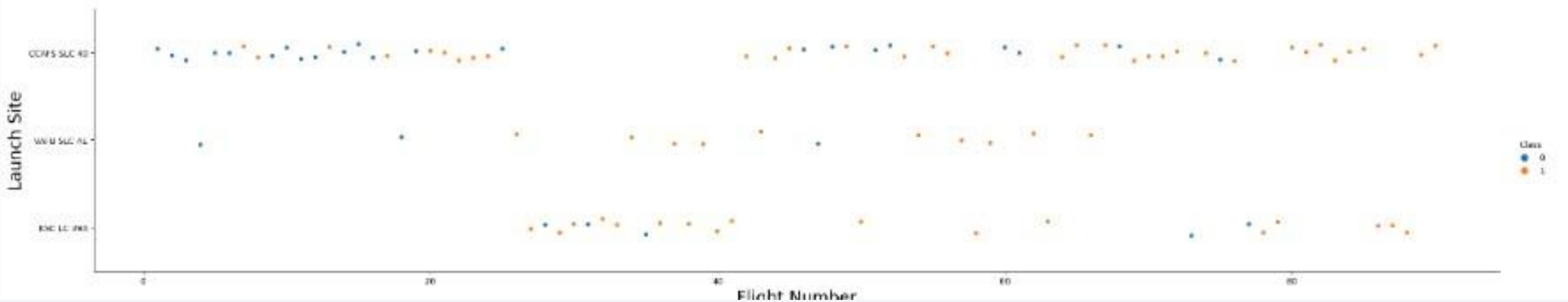
- We had more unsuccessful mission outcome than successful mission outcome.
- Some launch sites like HEO, SSO, GEO AND ES-L1 has a high success rate.
- Only LEO has a relationship between the success rate and flight number
- Heavy payload leads to increase in successful outcome rate in LEO, ISS and POLAR.
- From 2014, we had constant increase with successful launch.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

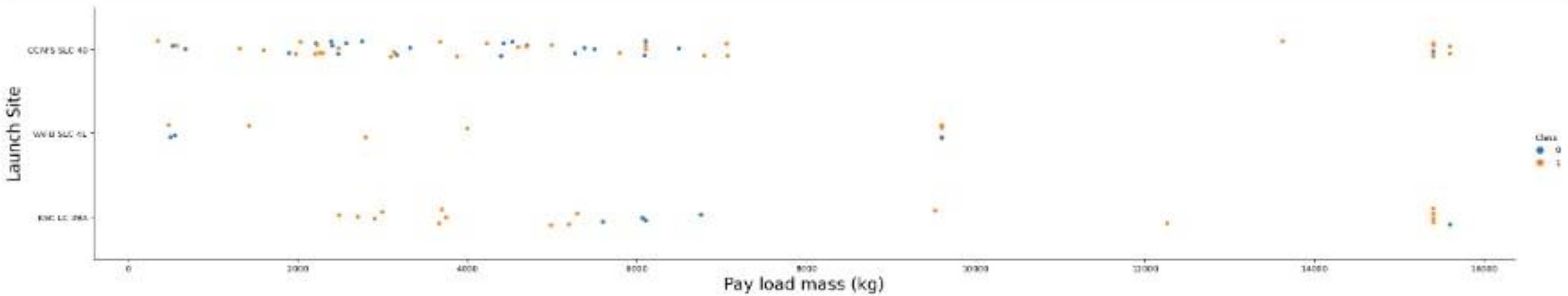
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class val
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



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%.

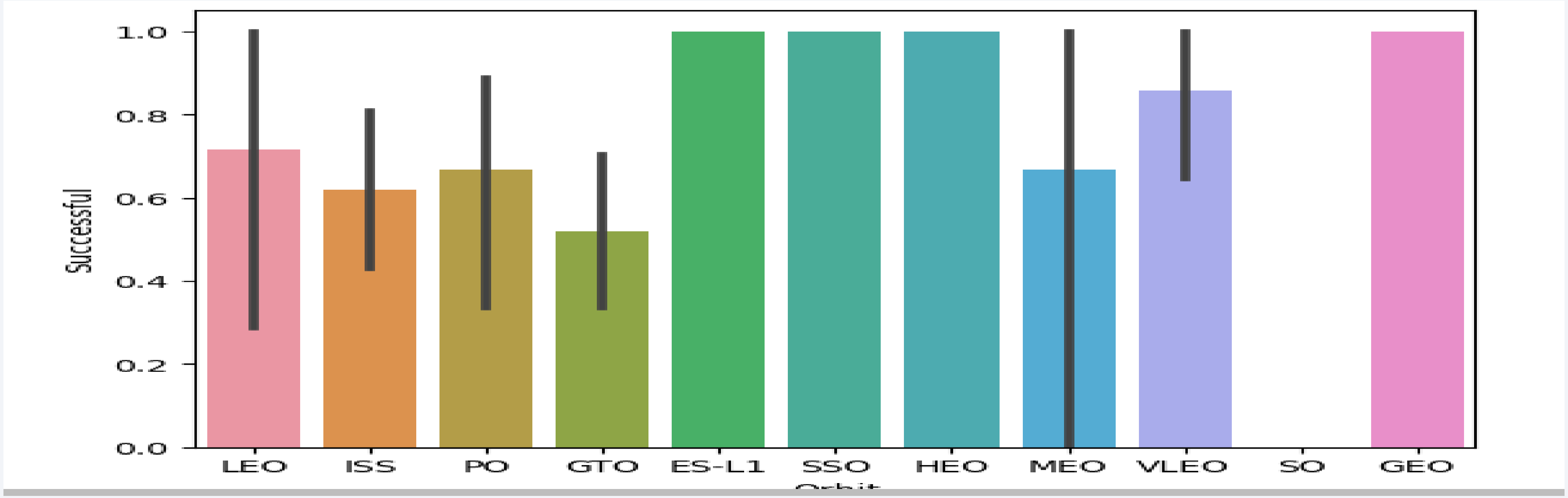
Payload vs. Launch Site

```
# 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 clas
sns.catplot(y='LaunchSite', x='PayloadMass', hue='Class', data=df, aspect=5)
plt.ylabel('Launch Site', fontsize=20)
plt.xlabel('Pay load mass (kg)', fontsize=20)
plt.show()
```



- Show a scatter plot of Payload vs. Launch Site
- Payload Vs. Launch Site scatter point chart shows that only in VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

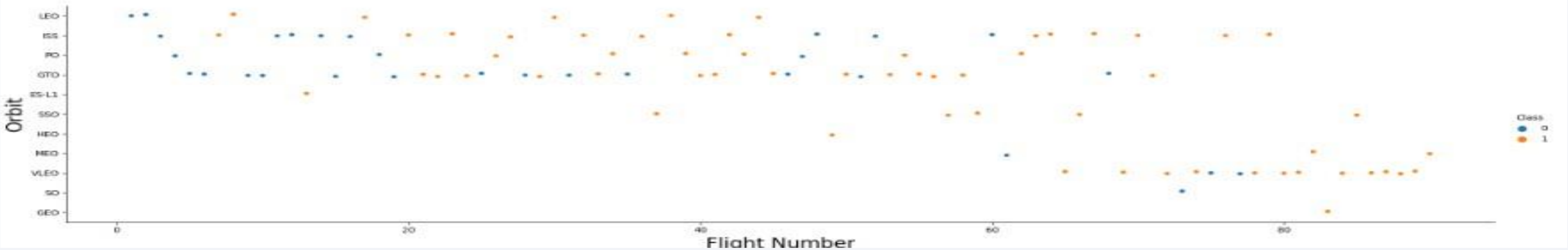
Success Rate vs. Orbit Type



HEO, GEO, SSO and ES-L1 has the highest success outcome when compared to other Orbits

Flight Number vs. Orbit Type

```python
# 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=4)
plt.ylabel('Orbit', fontsize=20)
plt.xlabel('Flight Number', fontsize=20)
plt.show()
```

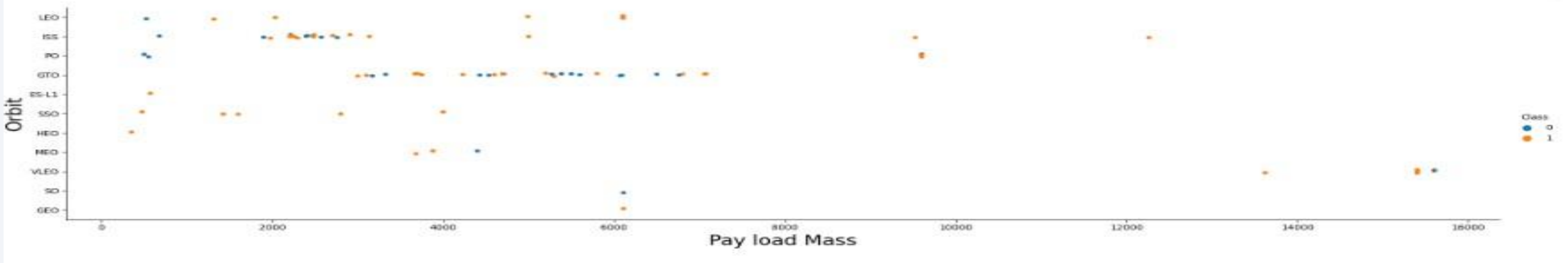


- Show a scatter point of Flight number vs. Orbit type

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

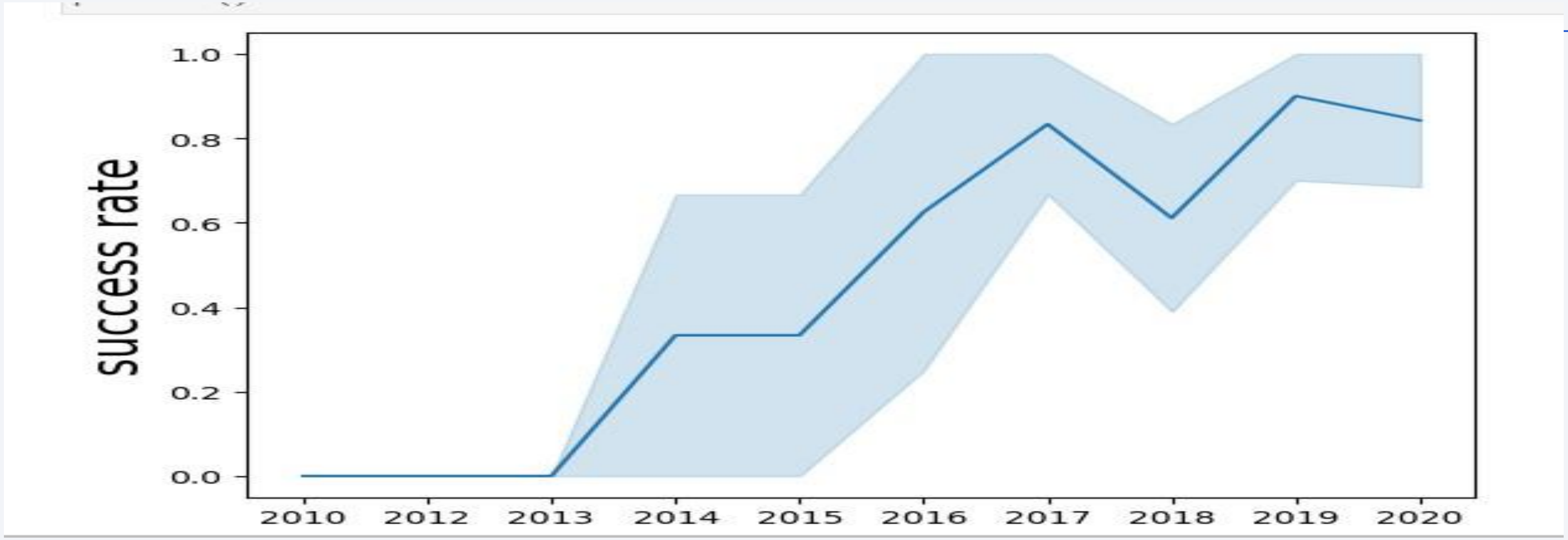
Payload vs. Orbit Type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y='Orbit', x='PayloadMass', hue='Class', data=df, aspect=4)
plt.ylabel('Orbit', fontsize=20)
plt.xlabel('Pay load Mass', fontsize=20)
plt.show()
```



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

Launch Success Yearly Trend



- More success was recorded from 2013 and above. The success increased as year increase except for 2018 which had a small decrease when compared to 2017

All Launch Site Names

```
[7]: %sql select DISTINCT LAUNCH_SITE from SPACEXTBL
```

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

[7]:

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

- The four distinct launch site was extracted using 'DISTINCT'

Launch Site Names Begin with 'CCA'

```
[8]: %sql select * from SPACEXTBL where LAUNCH_SITE  LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
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
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt

- The launch site with CCA were extracted using 'limit' to view only 5

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER LIKE 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

- The total pay load of NASA was calculated using 'sum' and 'like' function
- The result came out as 45596kg total

Average Payload Mass by F9 v1.1

Task 4

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

```
]: %sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Booster_Version ='F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

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

First Successful Ground Landing Date

- Using the function 'min', the date of the first success launch was gotten to be 01-03-2017

Successful Drone Ship Landing with Payload between 4000 and 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

```
%sql select "Booster_Version" from SPACEXTBL where "Landing _Outcome" like 'success (drone ship)' and PAYLOAD_MASS__KG_ be
```

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

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

- The successful payload landing of 4000 to 6000kg was gotten to be 5. operators like 'and' and 'like' was used.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
5]: %sql select count("Mission_Outcome") as successfulmission from SPACEXTBL where "Mission_Outcome" like 'success%'
```

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

5]:

successfulmission
100

```
6]: %sql select count ('Mission_Outcome') as failedmission from SPACEXTBL where "Mission_Outcome" like 'failure%'
```

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

6]:

failedmission
1

- There were 100 successful mission and one failed mission
- The 'like' operator was used to obtain this information

Boosters Carried Maximum Payload

```
%sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTBL WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- The minimum payload was 15600 and about 15 flight had that laod size.

2015 Launch Records

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.

```
%sql select "Booster_Version", "Launch_Site","Landing _Outcome", "Date"  from SPACEXTBL where "Landing _Outcome" like 'fai
```

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

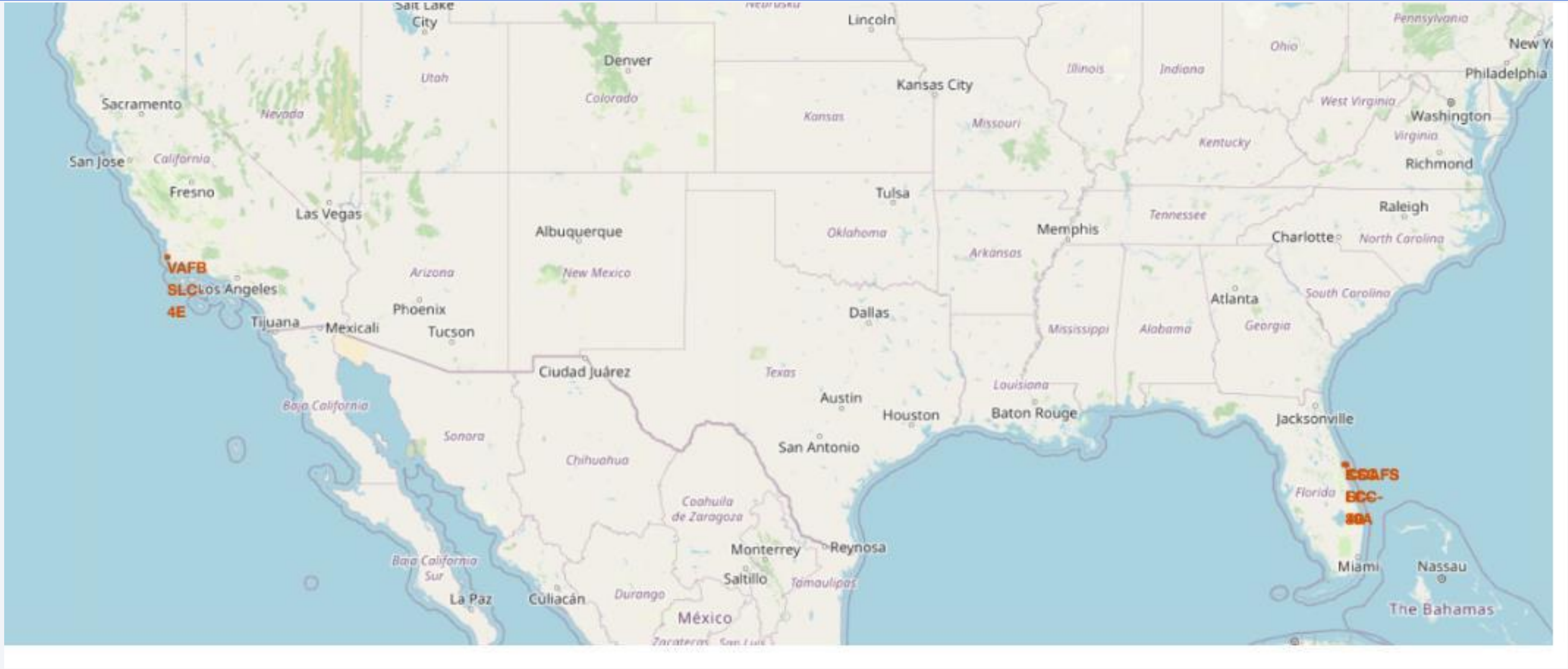
Booster_Version	Launch_Site	Landing _Outcome	Date
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	10-01-2015
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	14-04-2015

- There only two failed launch record for 2015

Section 3

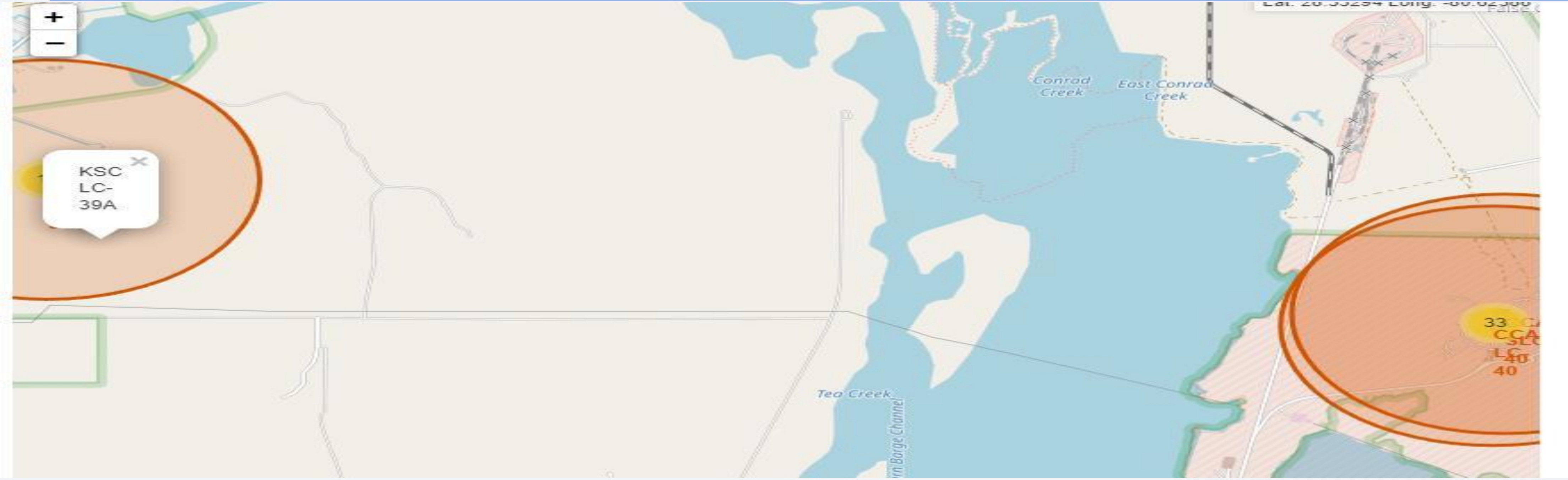
Launch Sites Proximities Analysis

LAUNCH SITE MAP LOCATION



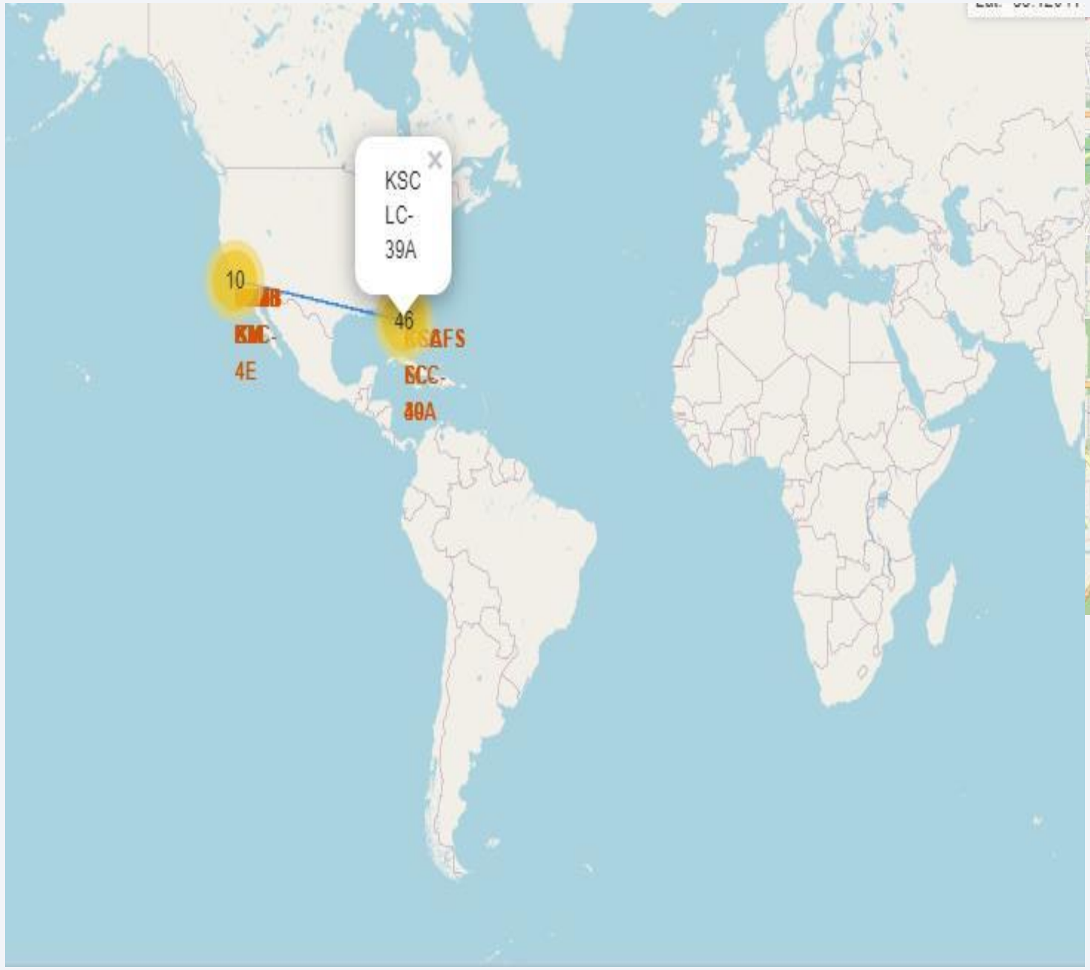
- The launch sites are mostly located far away to end of city. Two sites are close to each other hence, doubling themselves on the map

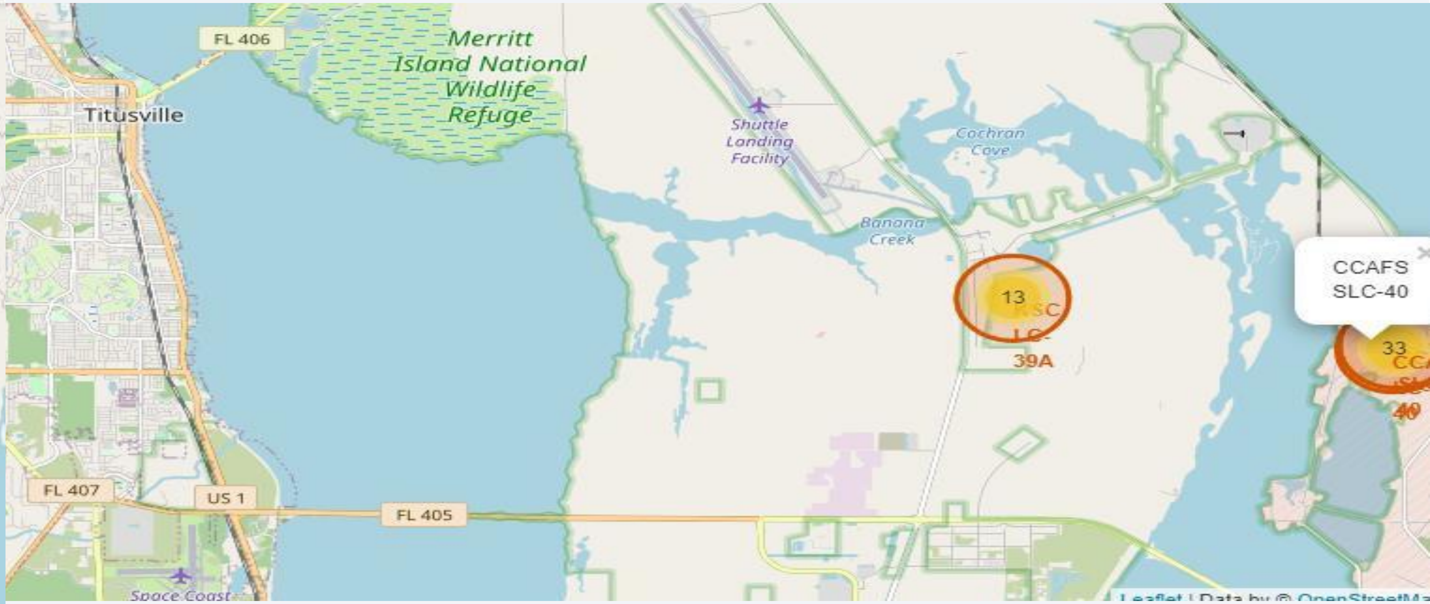
COLOUR IDENTIFICATION OF LAUNCH SITE ON MAP



- Color was added to the map to identify it position. The bobble helped make the launch site position more visible.

PROXIMITY OF LAUNCH SITE TO ESSENTIAL FACILITIES.



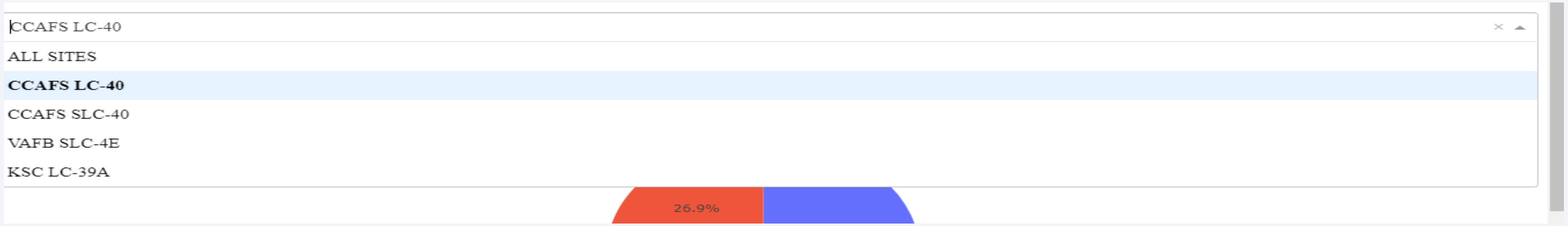


- The proximity was located and the distance between launch site and essential facilities were displayed using a line to trace the distance.

Section 4

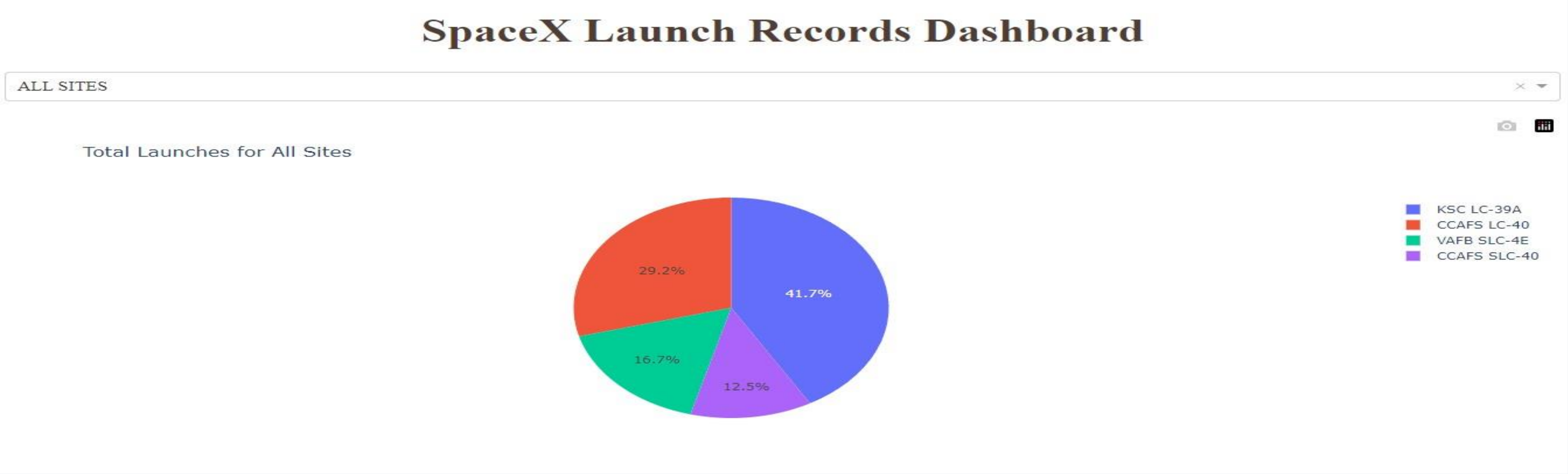
Build a Dashboard with Plotly Dash

INTERACTIVE VISUAL OF ALL LAUNCH SITE



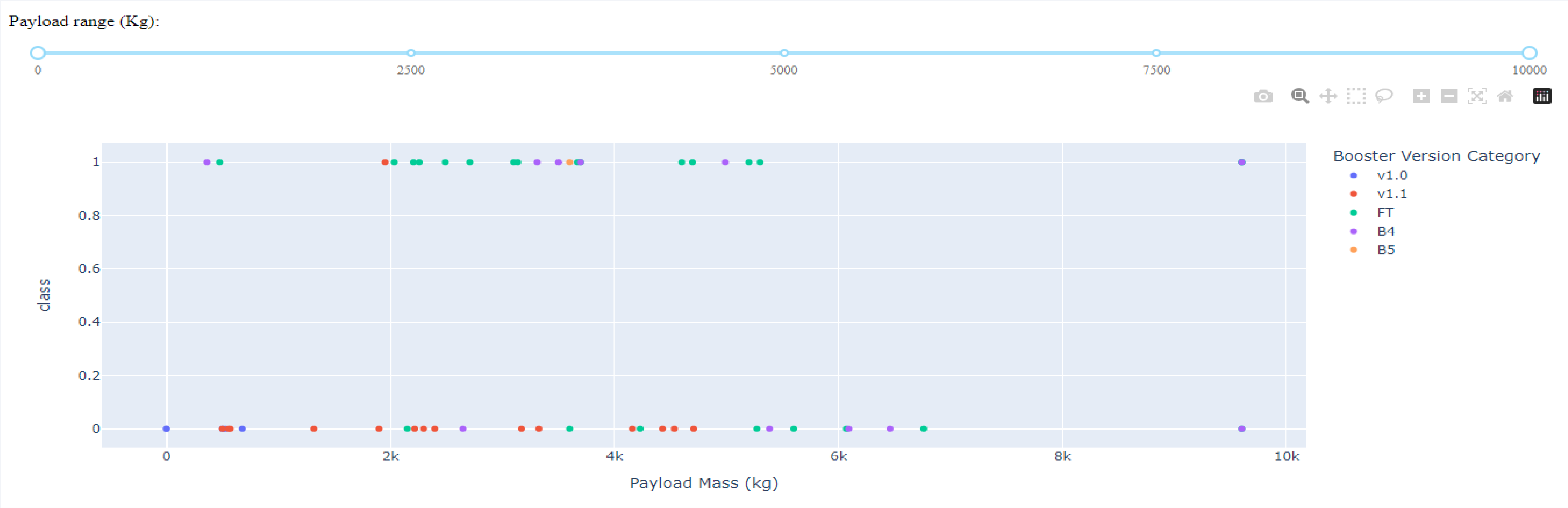
- An interactive visual of all the launch site was done to compare the success rate and payload.

LAUNCH SITE SUCCESS RATE



- With a success rate of 41,7% KSC launch site had the highest success rate compared to the other sites

INTERACTIVE VISUAL OF PAYLOAD (SCATTER PLOT)



- An interactive scatter plot was done to check the payload of all site and individual site.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

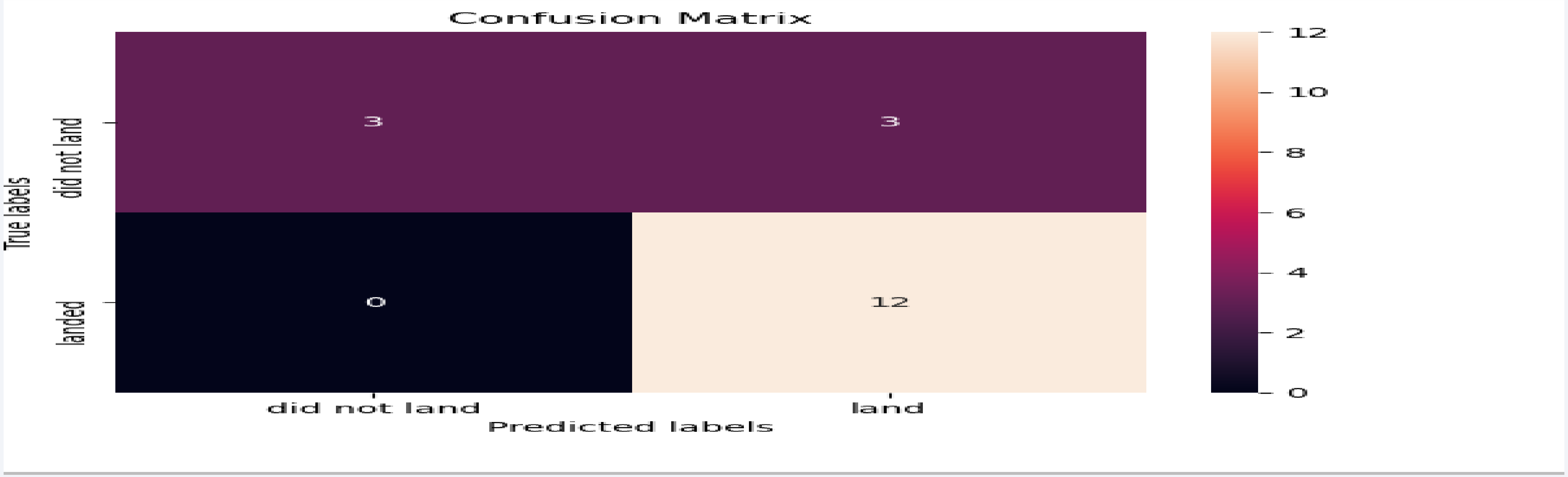
Find the method performs best:

```
models={'KNeighbors':knn_cv.best_score_,
        'DecisionTree':tree_cv.best_score_,
        'LogisticRegression':logreg_cv.best_score_,
        'SupportVector':svm_cv.best_score_}
best_algorithm = max(models, key=models.get)
#print('the best method is :',best_algorithm, 'and its score is:', models[best_algorithm])
if best_algorithm=='DecisionTree':
        print('best is DecisionTree')
if best_algorithm=='KNeighbors':
        print('best is KNeighbors')
if best_algorithm=='LogisticRegression':
        print('best is LogisticRegression')
if best_algorithm=='SupportVector':
        print('best is SupportVector')
```

```
best is DecisionTree
```

- Most of the accuracy result came out same with 83.33% but DecisionTree is the best method with accuracy score of 83.33334%

Confusion Matrix



- The confusion matrix showed that we had a good result but a little issue(3) with false positive meaning unsuccessful that showed successful

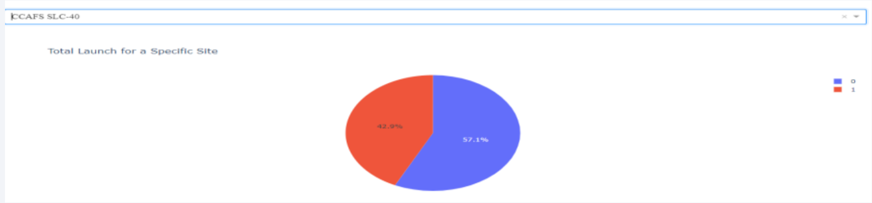
Conclusions

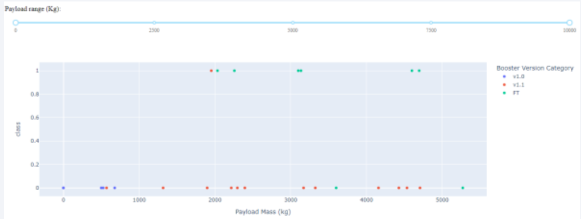
- There is no significant relationship between payload and successful launching.
- Higher launch amount increases the tendency of having more successful outcome.
- HEO, GEO, ES-L1 are the highest successful orbit.
- Success rate started to increase from 2013.
- KSC LC-39A is the highest success launch site.
- Decision tree with accuracy of 83.34% is the best method to classify.

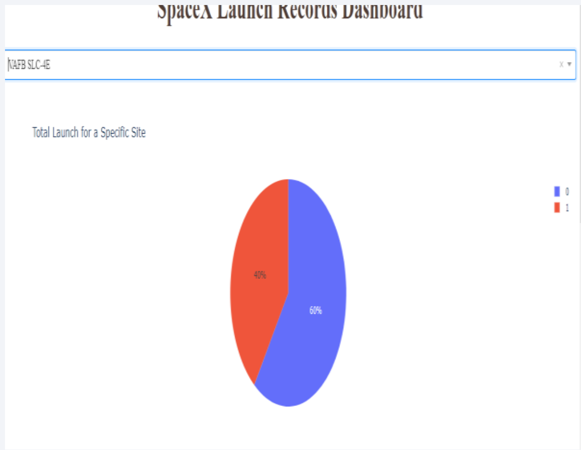
Appendix

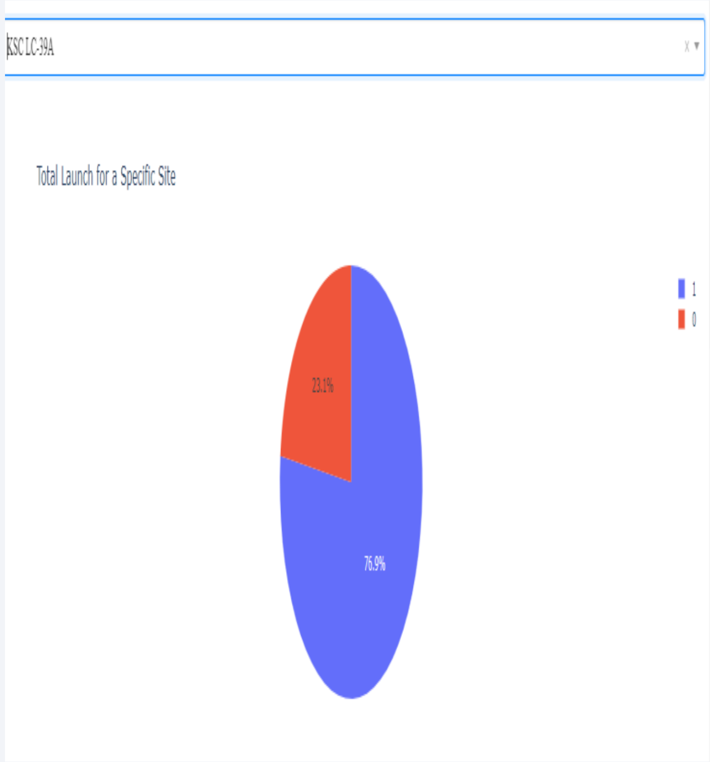
- Link for the data: https://github.com/ujunwa-DS/SPACEX-FALCON-9-CAPSTONE-PROJECT/blob/main/Spacex.csv

- SUCCESS VERSUS FAILURE OUTCOME OF EACH LAUNCH SITE










Thank you!