IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

M Homam Arjomandi
Dec 20, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection: The data been collected using the API and Web scraping using Request and BeautifulSoup Python libraries
  - Data Wrangling: Understanding of the data/pattern and a “Class” column added to represent the success and failure of booster landing using Pandas and NumPy
  - Exploratory Data Analysis (EDA): exploring the features using SQL and Visualization
  - Interactive Visual Analytics and Dashboards: using Folium and Plotly Dash
  - Predictive Analysis (Classification): using Hyperparameter for SVM, Classification Trees, and Logistic Regression
- Summary of all results
  - EDA results
  - Interactive Visual Analytics and Dashboards results
  - Predictive Analysis results

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Problems you want to find answers
  - Determine the cost of a launch by determination **if the first stage will land**.
  - Predict if the Falcon 9 first stage will land successfully

Section 1

Methodology

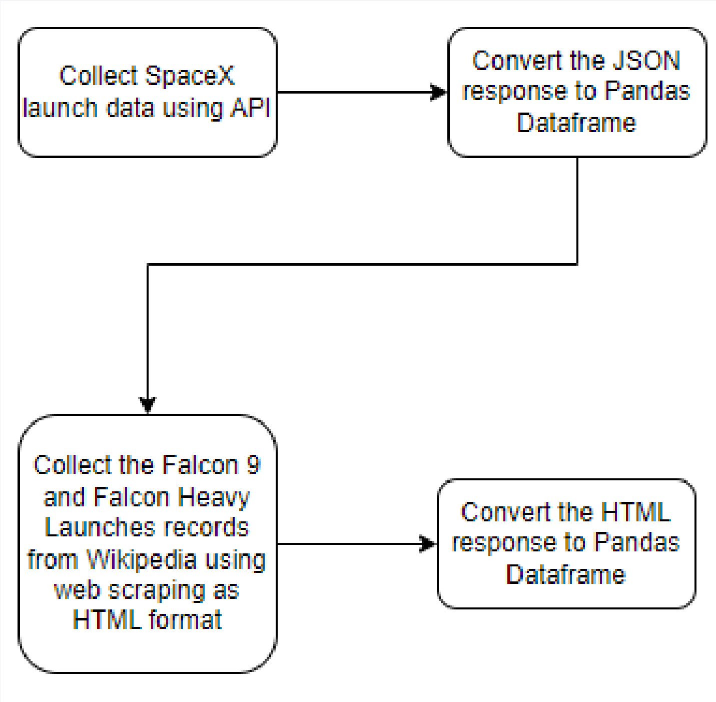
Methodology

Executive Summary

- Data collection methodology:
  - The data been collected using the API in form of JSON objects and Web scraping from Wikipedia in form of HTML and transferred into Panda Dataframes and saved as CSVs
- Perform data wrangling:
  - Handling data rows with missing values.
  - Adding the landing outcomes as Classes. (either 0 or 1). 0 is the booster did not land. 1 s, the booster did land.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Split the data into training testing data, train different classification models, optimize the Hyperparameter grid search, and, finally, find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression

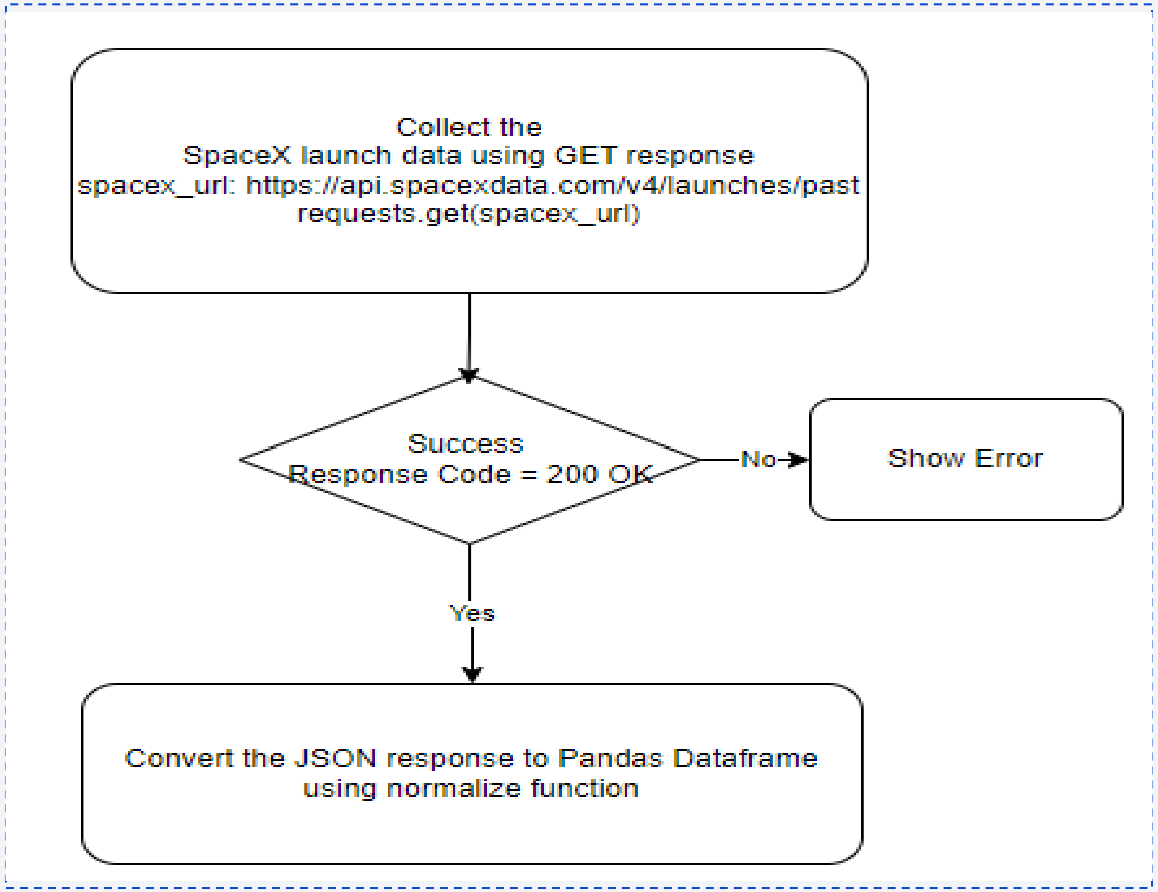
Data Collection

- Data sets were collected.
  - SpaceX launch data
  - Falcon 9 and Falcon Heavy Launches data
- Data collection process:
  - Collect the SpaceX launch data using API as JSON format
  - Collect the Falcon 9 and Falcon Heavy Launches records from Wikipedia using web scraping as HTML format



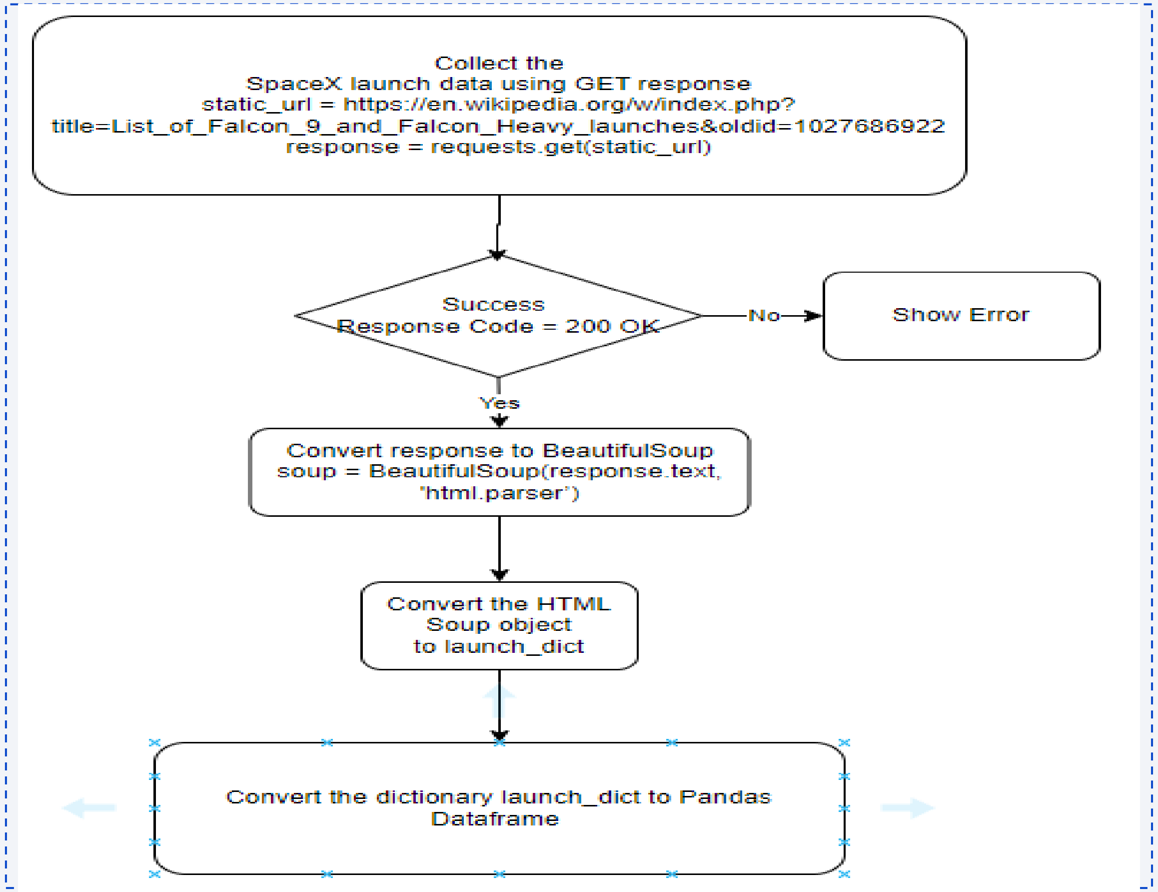
Data Collection – SpaceX API

- spacex_url=https://api.spacexdata.com/v4/launches/past
- response = requests.get(spacex_url)
- data = pd.json_normalize(response.json())
- GitHub URL of the completed SpaceX API calls notebook https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- response = requests.get(static_url)
- soup = BeautifulSoup(response.text, 'html.parser')
- launch_dict= dict.fromkeys(column_names)
- Fill in the launch_dict from the HTML data
- df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
- GitHub URL of the completed web scraping notebook: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- How data were processed
- In the API module the JSON data is:
  - Normalized to convert to DataFrame
  - Subset of columns selected 'rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc'
  - remove rows with multiple cores
  - In Cores, Payload columns extract the first single value and update the same column with extracted value
  - Convert the Date column from date_utc to a datetime datatype
  - Restrict the dates of the launches <= 2020, 11, 13
  - Apply the BoosterVersion function to create the launch_dict which is converted to launch_data dataframe
  - Filter the launch_data to have only ‘Falcon 9’
  - Update all null data in column PayloadMass with the mean value of the same column
- In Web Scraping module:
  - Load all the HTML values to the launch_dict dictionary
  - Convert the launch_dict to DataFrame
- GitHub URL of completed data wrangling related notebooks:
- API module: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb,
- Web Scraping Module: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-webscraping.ipynb

EDA with Data Visualization

- Charts were plotted and why we used those charts
  - Categorical Plot out the FlightNumber vs. PayloadMass to visualize the relationship between Flight Number and Payload Mass
  - Categorical Plot out the FlightNumber vs. LaunchSite to visualize the relationship between Flight Number and Launch Site
  - Categorical Plot out the Payload vs. LaunchSite to visualize the relationship between Payload and Launch Site
  - Bar Chart to represent the relationship between Success Rate of each Orbit type
  - Categorical Plot out the FlightNumber vs. Orbit to visualize the relationship between Flight Number and Orbit
  - Categorical Plot out the PayLoad vs. Orbit to visualize the relationship between PayLoad and Orbit
  - Line Plot out the Success Rate vs. Year to visualize the relationship between Success Rate and Year
- GitHub URL of completed EDA with data visualization notebook: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- The SQL queries you performed
  - 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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL of the completed EDA with SQL notebook: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

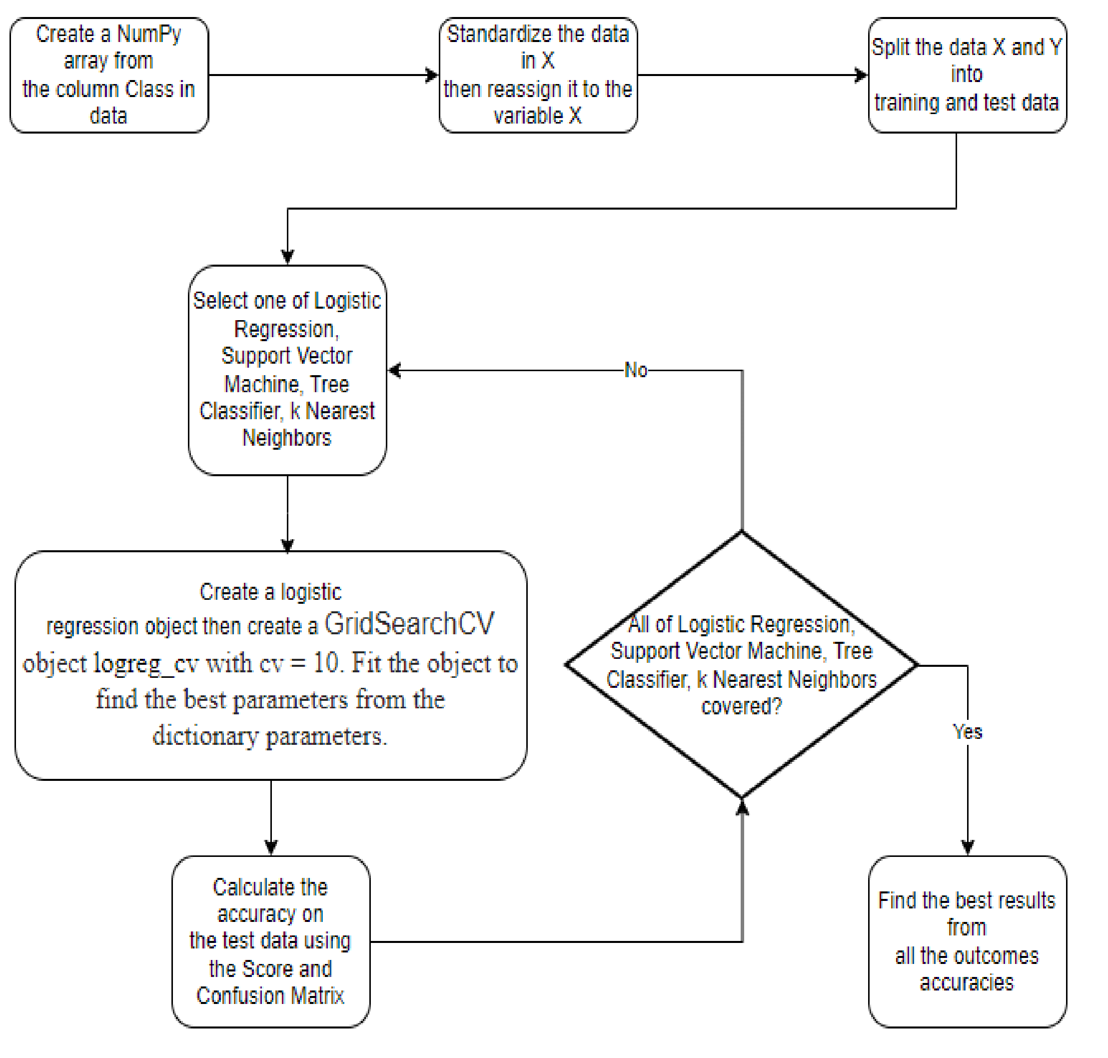
- Map objects such as markers, circles, lines, etc. created and added to a folium map
  - folium.Circle to add a highlighted circle area with a text label on a NASA Johnson Space Center and launch sites
  - folium.Marker to add a marker with a text label on a NASA Johnson Space Center and launch sites
  - folium. MarkerCluster to map containing many markers for each launch for the same site with green for success and red for failed ones
  - folium.PolyLine to draw a line between the site and the ocean coast
- GitHub URL of the completed interactive map with Folium map: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions you have added to a dashboard:
  - Pie chart graph to show the total success launches (i.e., the total count of class column) for All Sites
  - Pie chart graph for a selected site to show the success (class=1) count and failed (class=0) count for the selected site.
  - Scatter plot with the x axis to be the payload and the y axis to be the launch outcome (i.e., class column). As such, to visually observe how payload may be correlated with mission outcomes for selected site(s).
  - In addition, color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters.
- GitHub URL of the completed Plotly Dash lab: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Model development:
  - Create a NumPy array from the column Class in data
  - Standardize the data in X then reassign it to the variable X
  - Split the data X and Y into training and test data
  - For Logistic Regression, Support Vector Machine, Tree Classifier, k Nearest Neighbors do the following
    1. Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
    2. Calculate the accuracy on the test data using the Score and Confusion Matrix
  - Find the best results from all the outcomes accuracies
- GitHub URL of the completed predictive analysis lab: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Results

- Most of the flight numbers are covered by CCAFS SLC 40 Launch Site
- More success flight numbers over 20
- VAFB SLC 4E has the least flight numbers
- VAFB-SLC launch site no rockets launched for heavy payload mass(greater than 10000).
- CCAFS SLC 40 covered all payloads except the ones between 8000 and 14000.
- VAFB-SLC launch site covers the 10000 payload launches.
- RS-L1, GEO, HEO, SSO orbits have the highest success rate.
- GTO has the lowest success rate
- SO has no success
- Most of the flight numbers are covered in VLEO, SSO, GTO, PO, ISS, LEO
- GEO has only one success launch
- SO has only one failed launch
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites with 76.9% Success rate.
- The Decision Tree Classifier is the best machine learning algorithm for this task with accuracy is 0.833%.
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- GTO both positive landing rate and negative landing(unsuccessful mission) are both there here.
- The success rate since 2013 kept increasing till 2020

Section 2

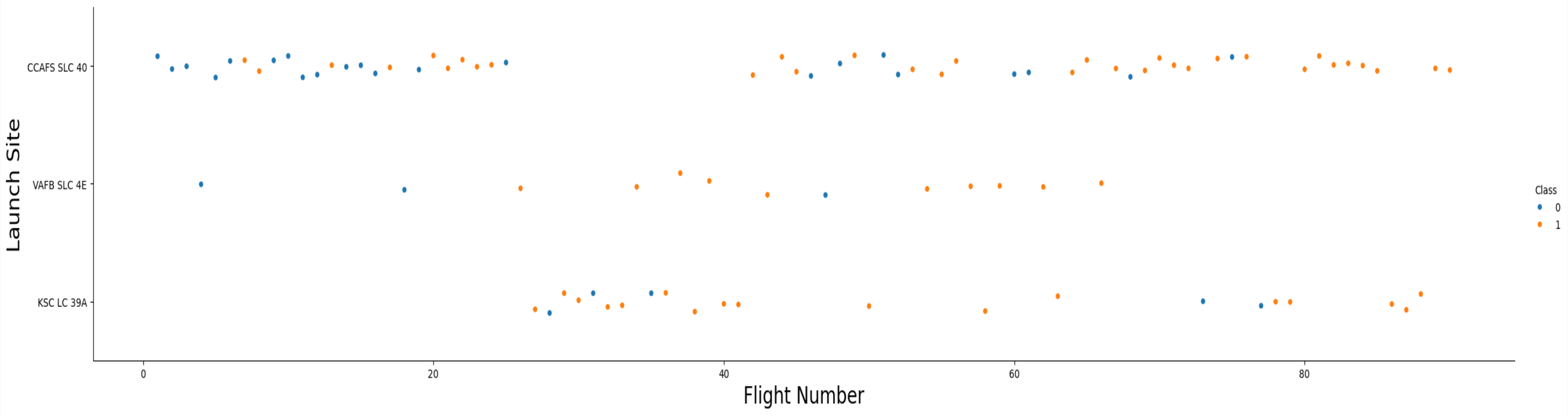
Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations



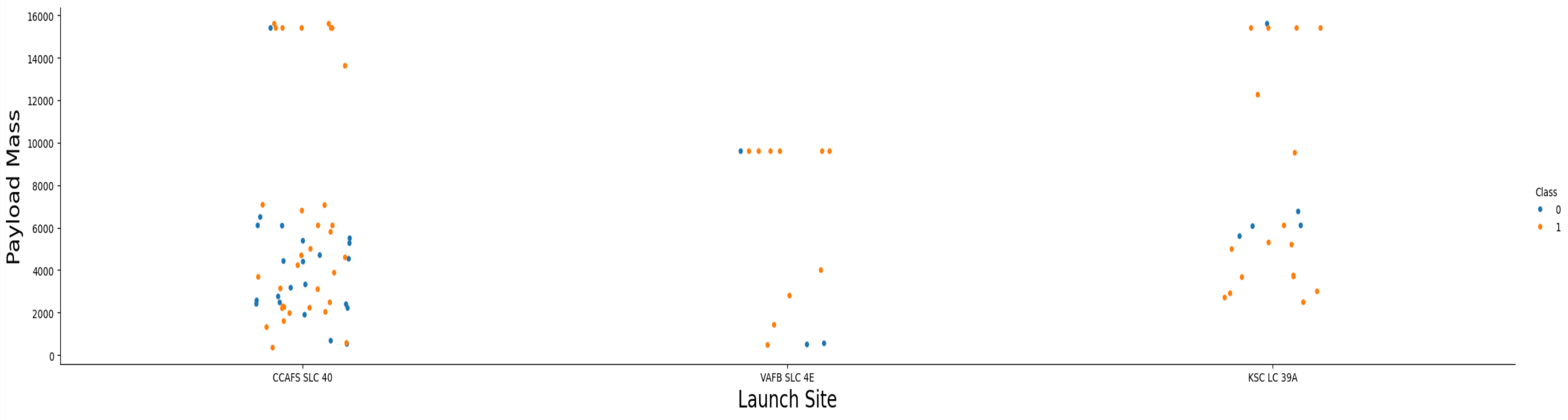
- Most of the flight numbers are covered by CCAFS SLC 40 Launch Site
- More success flight numbers over 20
- .VAFB SLC 4E has the least flight numbers

Payload vs. Launch Site

- Show a scatter plot of 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 class value
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Payload Mass",fontsize=20)
plt.show()
```

- Show the screenshot of the scatter plot with explanations



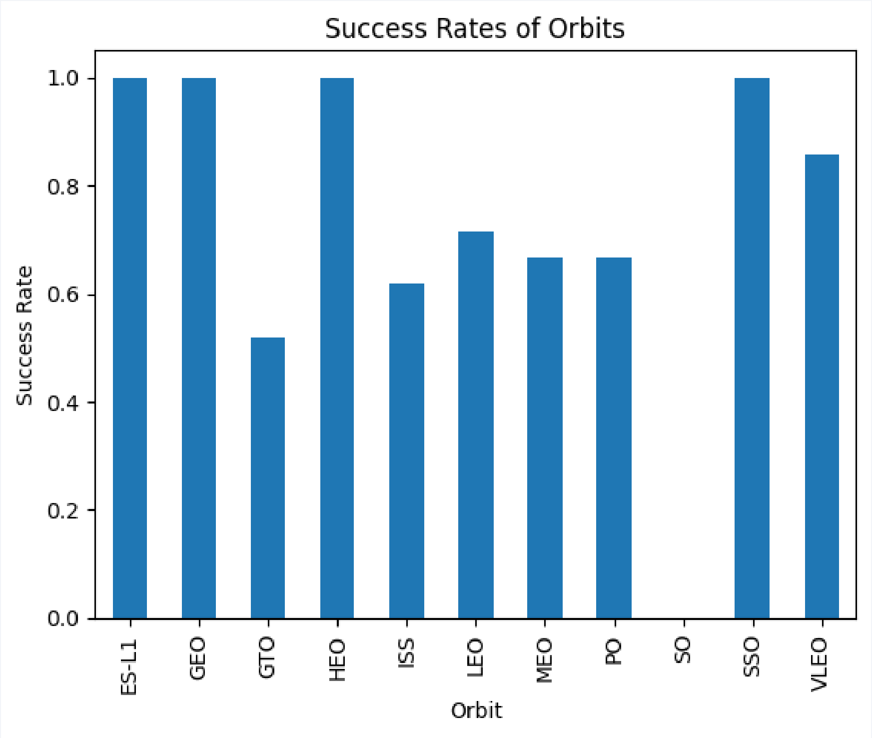
- VAFB-SLC launch site no rockets launched for heavy payload mass(greater than 10000).
- CCAFS SLC 40 covered all payloads except the ones between 8000 and 14000.
- VAFB-SLC launch site covers the 10000 payload launches.

Success Rate vs. Orbit Type

```
orbitg=df.groupby('Orbit')['Class'].mean()
# orbitg

orbitg.plot(kind='bar', title='Success Rates of Orbits', ylabel='Success Rate', xlabel='Orbit')
```

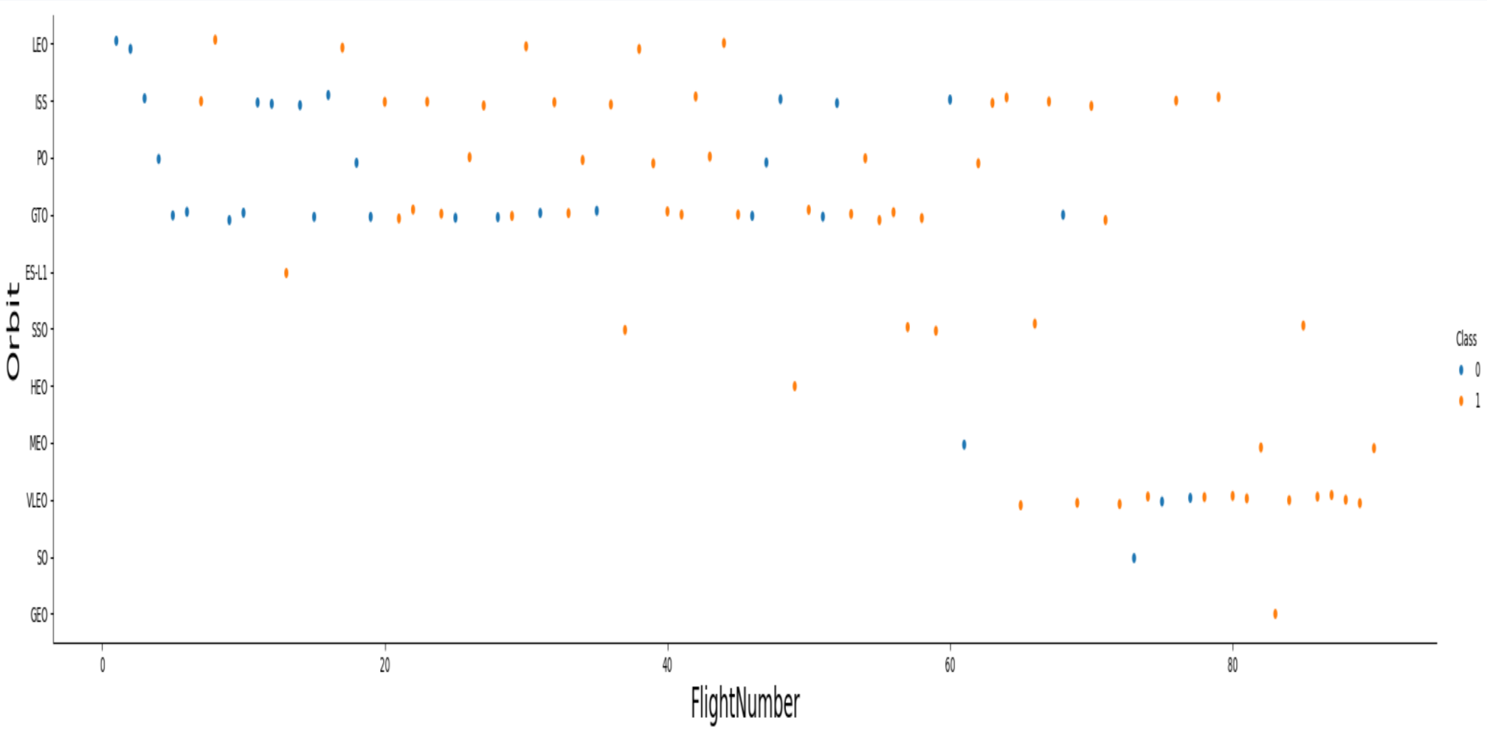
- RS-L1, GEO, HEO, SSO orbits have the highest success rate.
- GTO has the lowest success rate
- SO has no success



Flight Number vs. Orbit Type

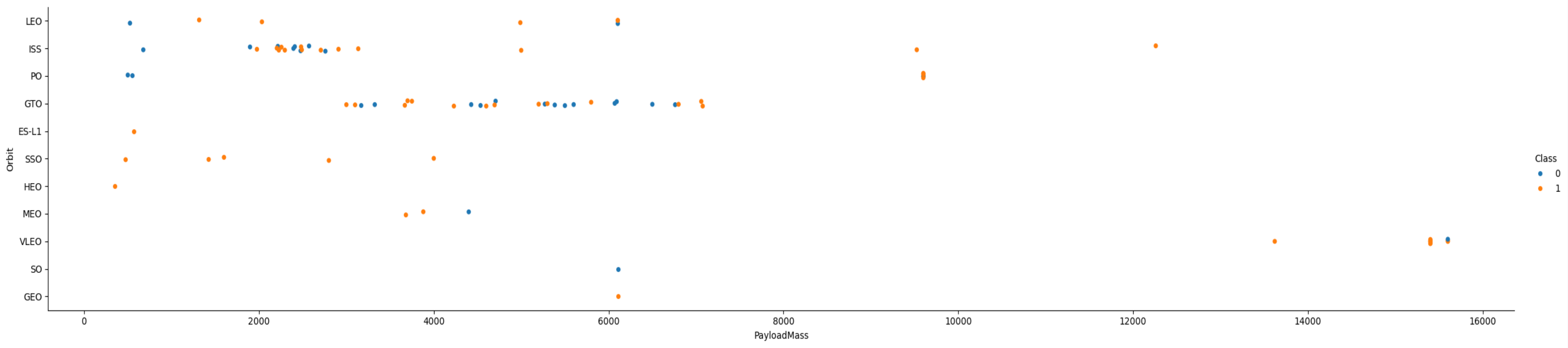
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

- 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.
- Most of the flight numbers are covered in VLEO, SSO, GTO, PO, ISS, LEO
- GEO has only one success
- SO has only one failed



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 = 5)
plt.xlabel("PayloadMass")
plt.ylabel("Orbit")
plt.show()
```

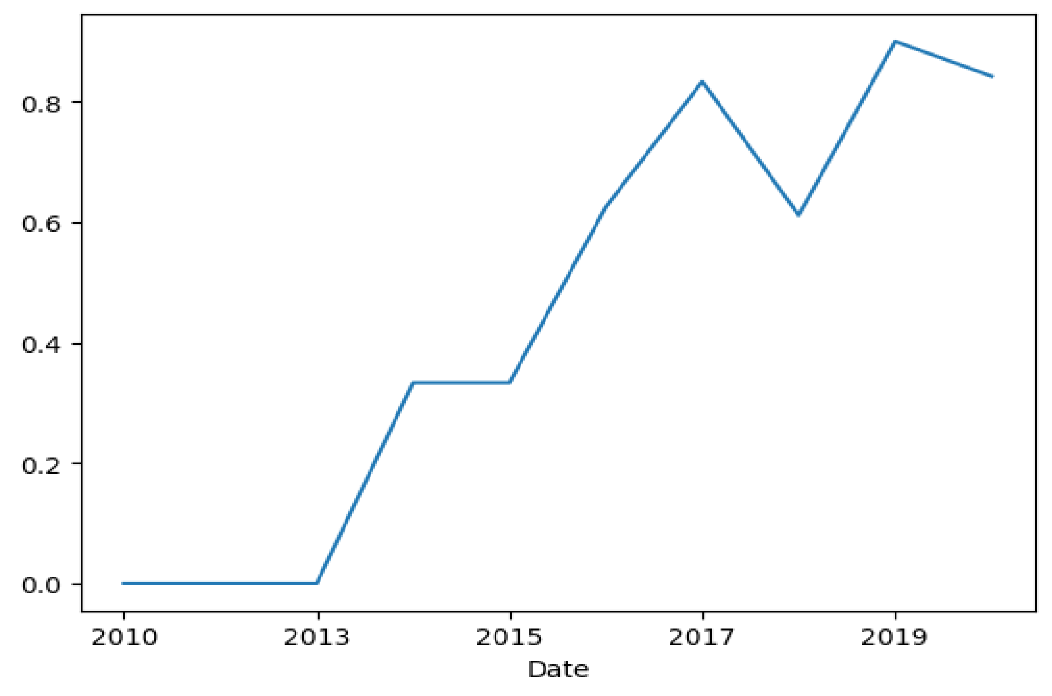


- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- GTO both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

```python
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df.groupby("Date")["Class"].mean().plot(kind="line")
```

- The success rate since 2013 kept increasing till 2020



All Launch Site Names

- Names of the unique launch sites using DISTINCT

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

```
[11]: %sql  select distinct Launch_Site from SPACEXTABLE

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

[11]:

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

Launch Site Names Begin with 'CCA'

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

```
%sql  select * from SPACEXTABLE where Launch_Site like'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Find 5 records where launch sites begin with `CCA` using LIKE and LIMIT

Total Payload Mass

- Calculate the total payload carried by boosters from NASA using SUM and WHERE on Customer name

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer ='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1 using AVG and WHERE on Booster Version

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version ='F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the date of the first successful landing outcome on ground pad using MIN with WHERE on Landing Outcome

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

Hint:Use min function

```
%sql  select min(date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

min(date)
2015-12-22

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 using WHERE on Landing Outcome and Greater Than/Less Than Operators

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,PAYLOAD_MASS__KG_ from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000  and PAYLOAD_MASS__KG_< 6000
```

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

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 using Group By Mission Outcome

List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) from SPACEXTABLE group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass using Sub Query to get Max Payload

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

```
%sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ in (Select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 using SUBSTR to extract Year and filter Landing Outcome to be 'Failure (drone ship)'

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
4]: %sql select substr(Date, 6,2),* from SPACEXTABLE where substr(Date,0,5)='2015' and Landing_Outcome='Failure (drone ship)'
```

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

4]:

substr(Date, 6,2)	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
01	2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
04	2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

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 using WHERE to filter dates and Group By to have the count of each landing Out Come.

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.

```
%sql select Landing_Outcome, Count(Landing_Outcome) from SPACEXTABLE where Date BETWEEN '2010-06-04' AND '2017-03-20' group by Landing_Outcome order by Count(Landing_Outcome) desc
```

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

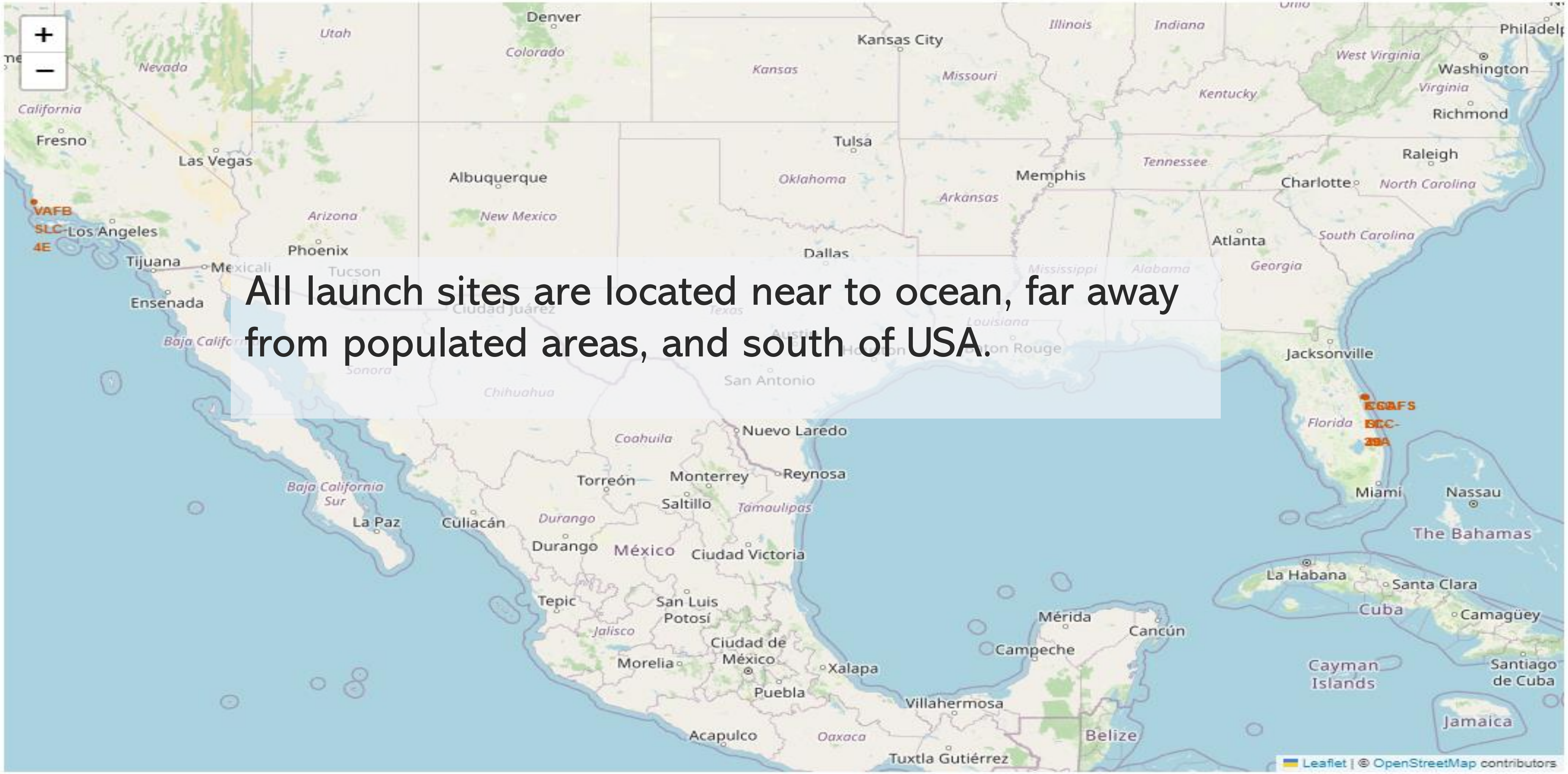
Landing_Outcome	Count(Landing_Outcome)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

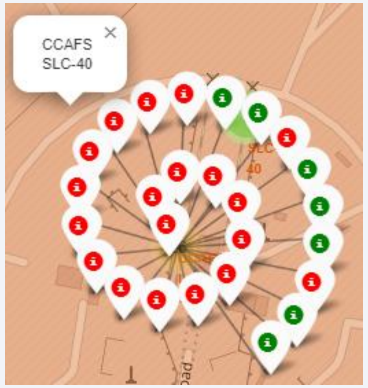
Launch Sites Locations Analysis with Folium

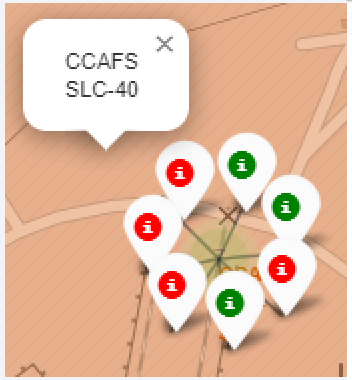
[7]:

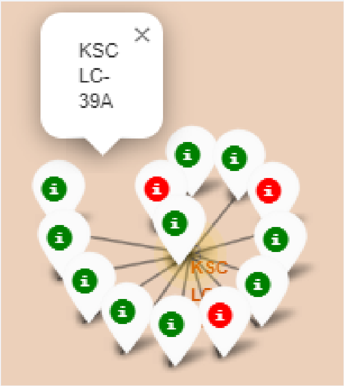


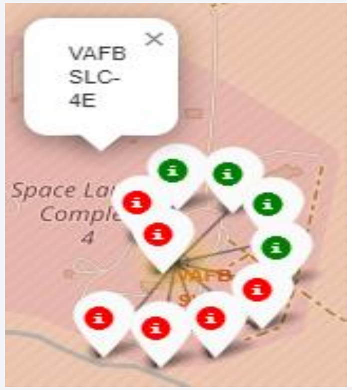
All launch sites are located near to ocean, far away from populated areas, and south of USA.

Mark the success/failed launches for each site on the map



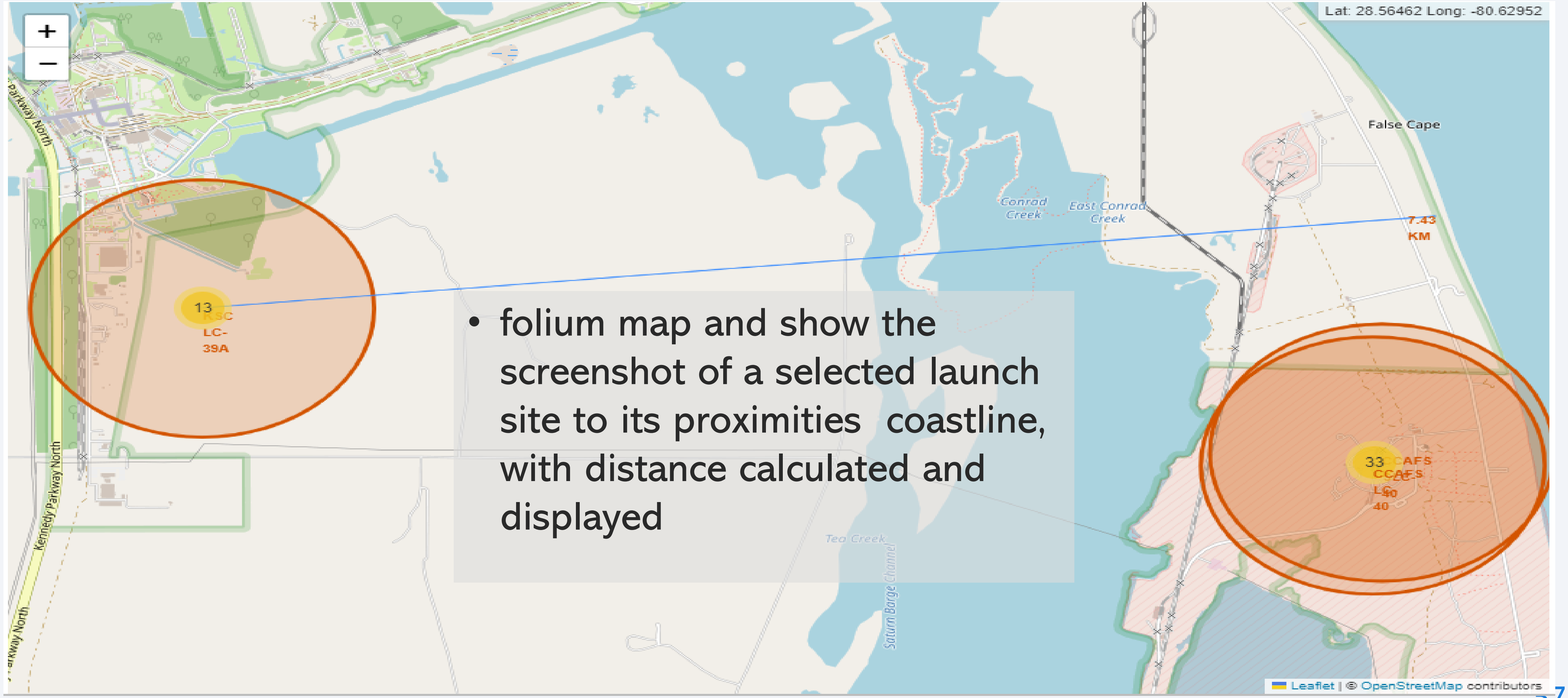






- CCAFS LC-40 has 7 Success of 26 launches
- CCAFS SLC-40 has 3 Success of 7 launches
- KSS SLC-39A has 10 Success of 13 launches
- VAFB SLC-4E has 4 Success of 10 launches
- Total launches is 56, 24 Success, 32 Failed

KSC LC-39A Launch site to a selected coastline point

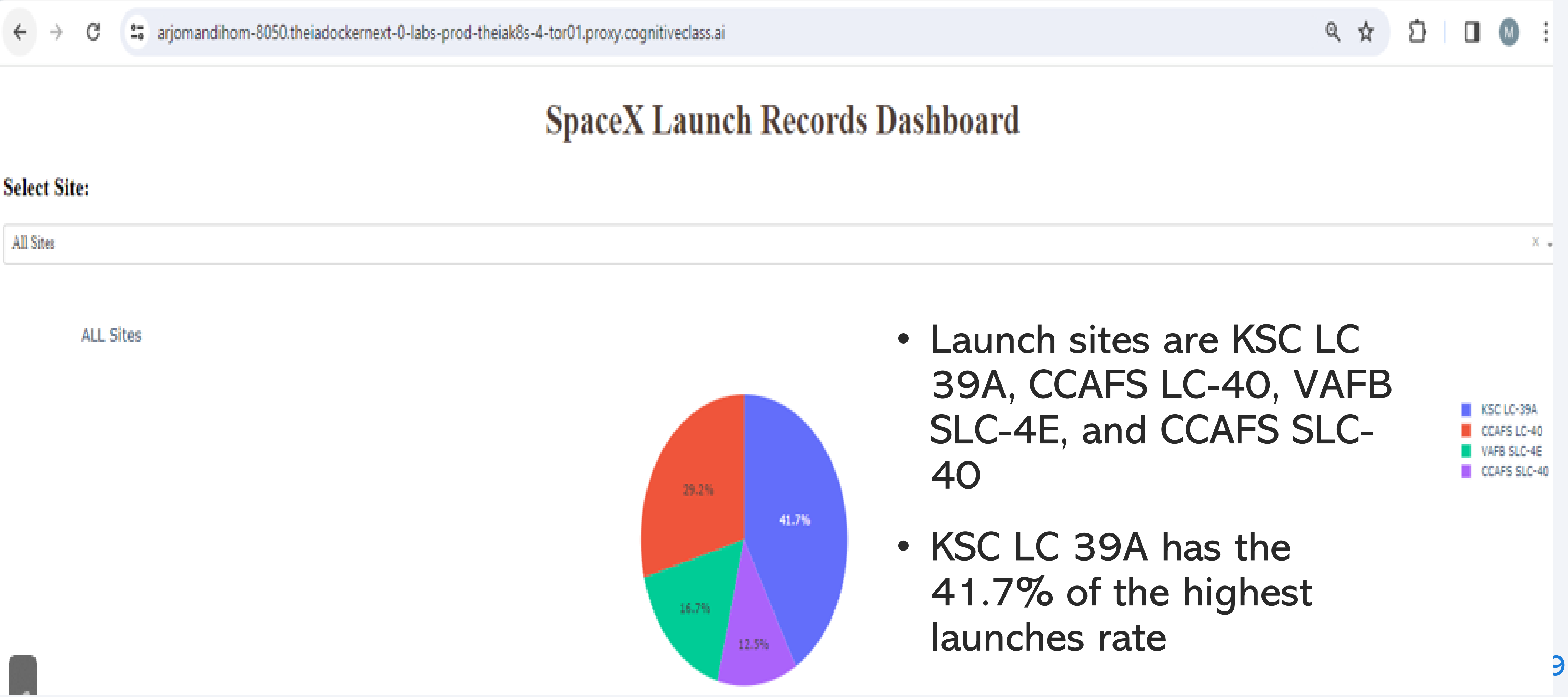


- folium map and show the screenshot of a selected launch site to its proximities coastline, with distance calculated and displayed

Section 4

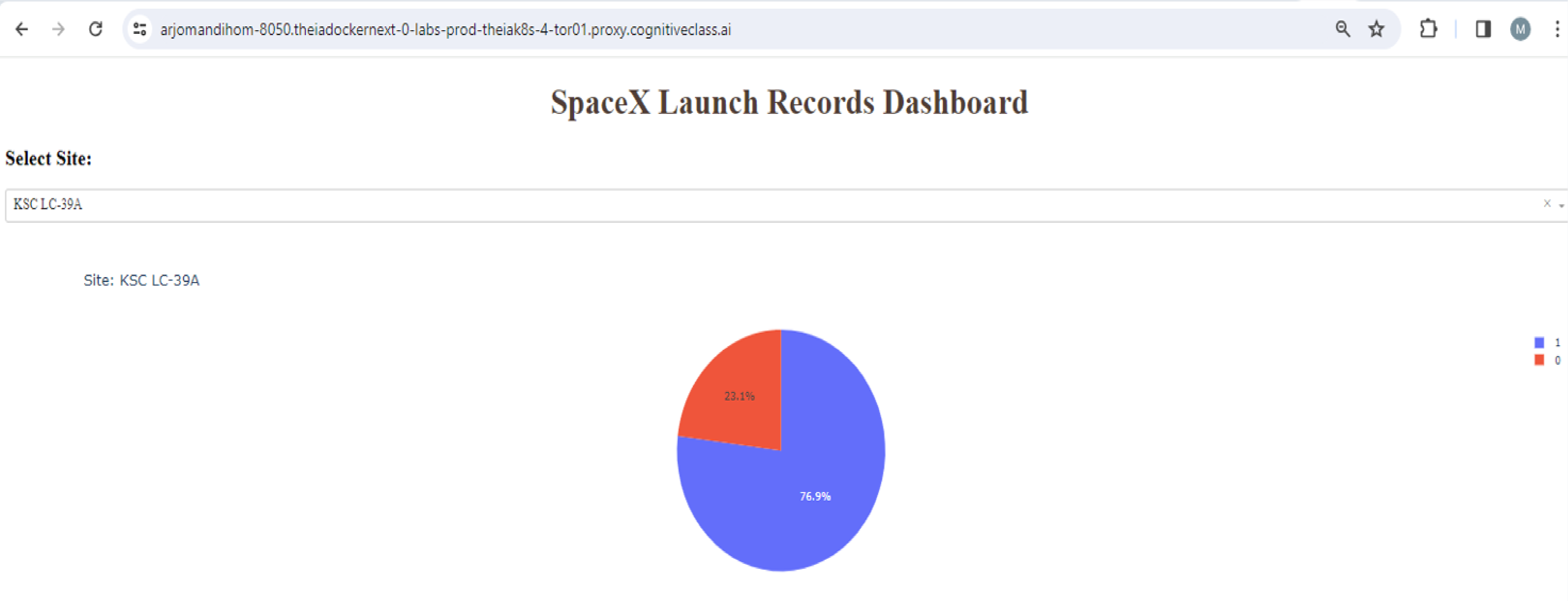
Build a Dashboard with Plotly Dash

Pie Chart: SpaceX Launch Records for ALL Sites



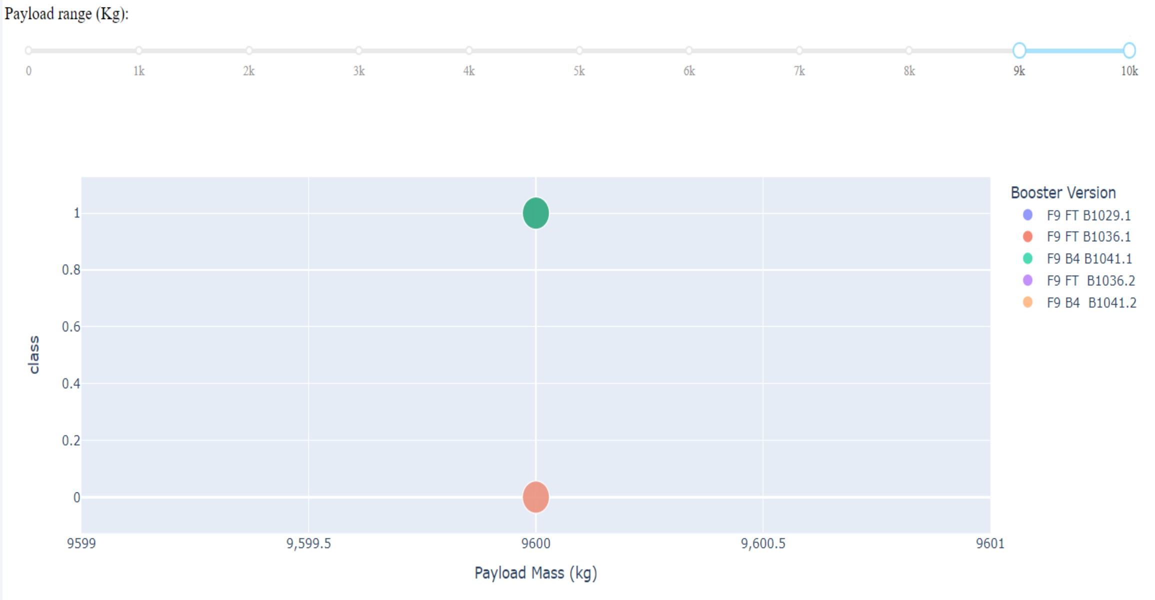
- Launch sites are KSC LC 39A, CCAFS LC-40, VAFB SLC-4E, and CCAFS SLC-40
- KSC LC 39A has the 41.7% of the highest launches rate

Pie Chart: SpaceX Launch Records for Site KSC LC-39A



Site KSC LC-39A has the highest launch success ratio

Scatter plot for the Payload and the Launch Outcome



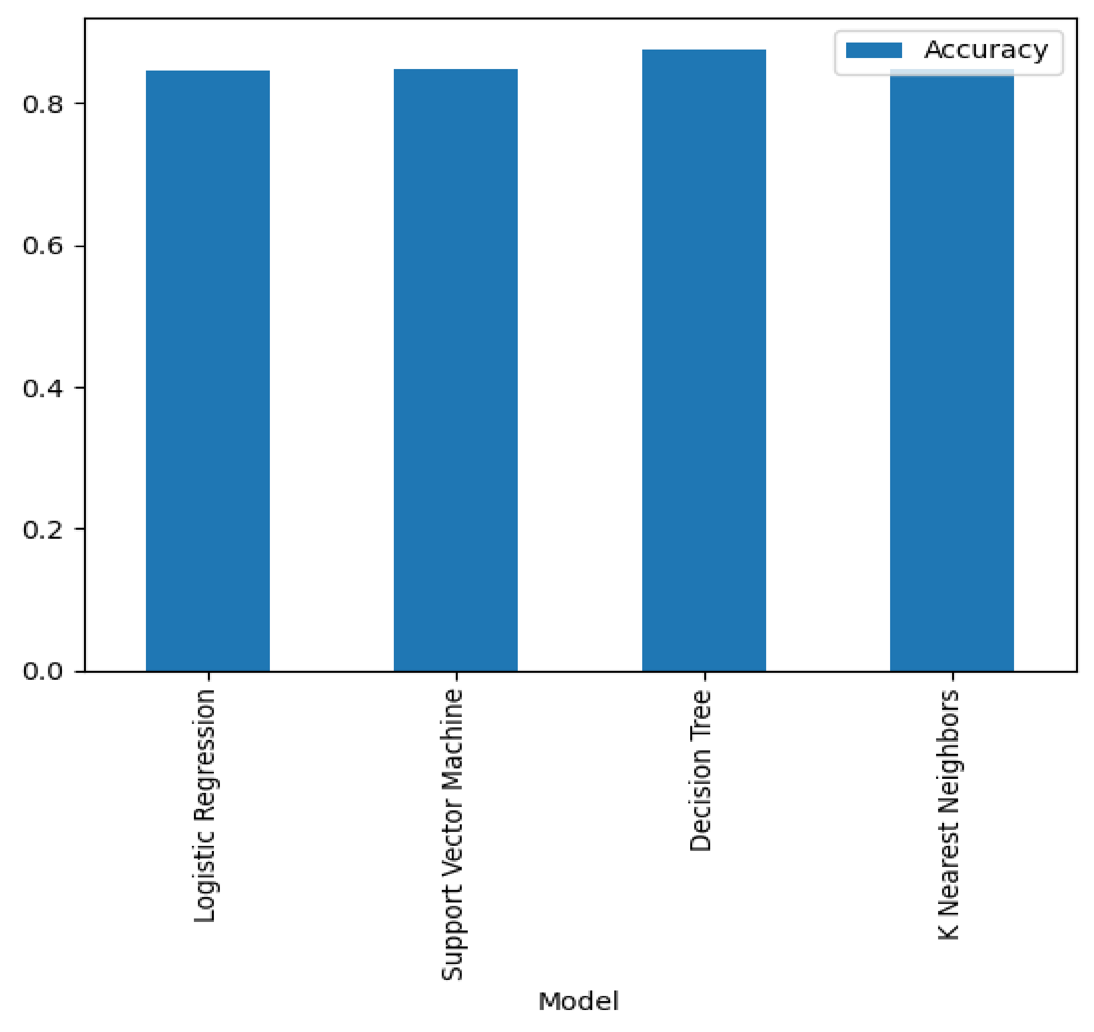
- Min Successful Payload is F9 B4 B1045.1
- Min Failed Payload are F9 v1.1 B0006, F9 v1.1 B1003
- Max Successful Payload are F9 FT B1029.1, F9 FT B1036.1, F9 B4 B1041.1
- Max Failed Payloads are F9 FT B1036.2, F9 B4 B1041.2

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree model has the highest classification accuracy (0.833)



Confusion Matrix

- Tree Classifier model's confusion matrix is the best performing model.
- Accuracy is 0.833%

TASK 9

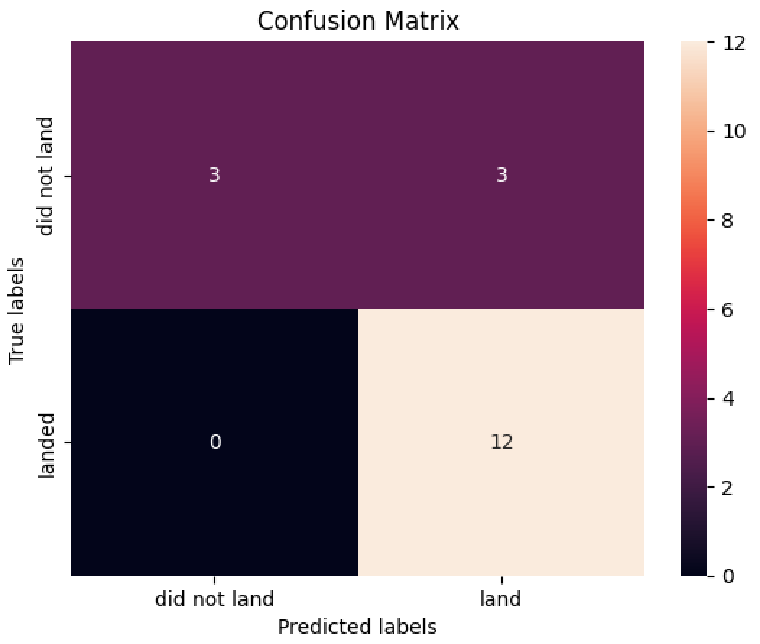
Calculate the accuracy of tree_cv on the test data using the method `score` :

```
[40]: yhat_tree=tree_cv.predict(X_test)
      metrics.accuracy_score(Y_test,yhat_tree)
```

```
[40]: 0.8333333333333334
```

We can plot the confusion matrix

```
[41]: plot_confusion_matrix(Y_test,yhat_tree)
```



Conclusions

- Most of the flight numbers are covered by CCAFS SLC 40 Launch Site
- More success flight numbers over 20
- VAFB SLC 4E has the least flight numbers
- VAFB-SLC launch site no rockets launched for heavy payload mass(greater than 10000).
- CCAFS SLC 40 covered all payloads except the ones between 8000 and 14000.
- VAFB-SLC launch site covers the 10000 payload launches.
- RS-L1, GEO, HEO, SSO orbits have the highest success rate.
- GTO has the lowest success rate
- SO has no success
- Most of the flight numbers are covered in VLEO, SSO, GTO, PO, ISS, LEO
- GEO has only one success launch
- SO has only one failed launch
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites with 76.9% Success rate.
- The Decision Tree Classifier is the best machine learning algorithm for this task with accuracy is 0.833%.
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- GTO both positive landing rate and negative landing(unsuccessful mission) are both there here.
- The success rate since 2013 kept increasing till 2020

Appendix

- Data sets created during this project:
  - https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/dataset_part_1.csv
  - https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/dataset_part_2.csv
  - https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/dataset_part_3.csv
  - https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/spacex_launch_dash.csv
  - https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation/blob/main/spacex_web_scraped.csv
- GitHub main folder that have all files URL: https://github.com/mharjomandi/IBM-Data-Science-Applied-Data-Science-Capstone-Presentation

Thank you!