IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

Summary of methodologies

- Collecting the data
- Data wrangling
- Exploring and Preparing Data
- EDA with SQL
- Launch Sites Locations Analysis with Folium
- Building a Dashboard with Plotly Dash
- Machine Learning Prediction

Summary of all results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

Project background and context

Falcon 9 is a two-stage reusable rocket designed and manufactured by SpaceX in the United State. It is reliable and safe transport of people and payloads into Earth orbit. We predicted SpaceX Falcon 9 First Stage will Land successfully or not. It cost 62 million dollars and other providers cost upward of 165 million dollars each. It saves lots of money because it can reuse the first stage. We can predict the cost of a launch if we can predict the first stage will land.

Problems you want to find answers

- What effects if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
- What circumstances does SpaceX have to achieve the best results and confirm the best rocket success landing rate?

Section 1

Methodology

Methodology

Data collection methodology

- Import Libraries and Define Auxiliary Functions
- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches
- remove the Falcon 1 launches keeping only the Falcon 9 launches
- Calculate the mean value of PayloadMass column
- export data_falcon9 to a CSV

Methodology

Perform data wrangling

- Put Space X dataset into a dataframe
- Identify and calculate the percentage of the missing values
- Identify which columns are numerical and categorical
- Calculate the number of launches on each site(use value_counts()method)
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- determine the sucess rate
- export dataframe to a CSV for EDA

Methodology

Perform exploratory data analysis (EDA) using visualization and SQL

- read the SpaceX dataset into a Pandas dataframe and print its summary
- Plot the FlightNumber vs. PayloadMassand overlay the outcome of the launch
- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type
- plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type
- Visualize the launch success yearly trend
- Create dummy variables to categorical columns
- Cast all numeric columns to float64
- export features_one_hot dataframe to a CSV

Methodology

- Perform interactive visual analytics using Folium and Plotly Dash

- Download and read the `spacex_launch_geo.csv`
- create a folium Map object with an initial center location to be NASA Johnson Space Center at Houston, Texas
- Create and add folium.Circle and folium.Marker for each launch site on the site map
- Mark the success/failed launches for each site on the map
- create markers for all launch records
- Calculate the distances between a launch site to its proximities
- calculate the distance between two points on the map based on their Lat and Long values
- Mark down a point on the closest railway using MousePosition and calculate the distance between the railway point to the launch site
- Draw a PolyLine between a launch site to the selected

Methodology

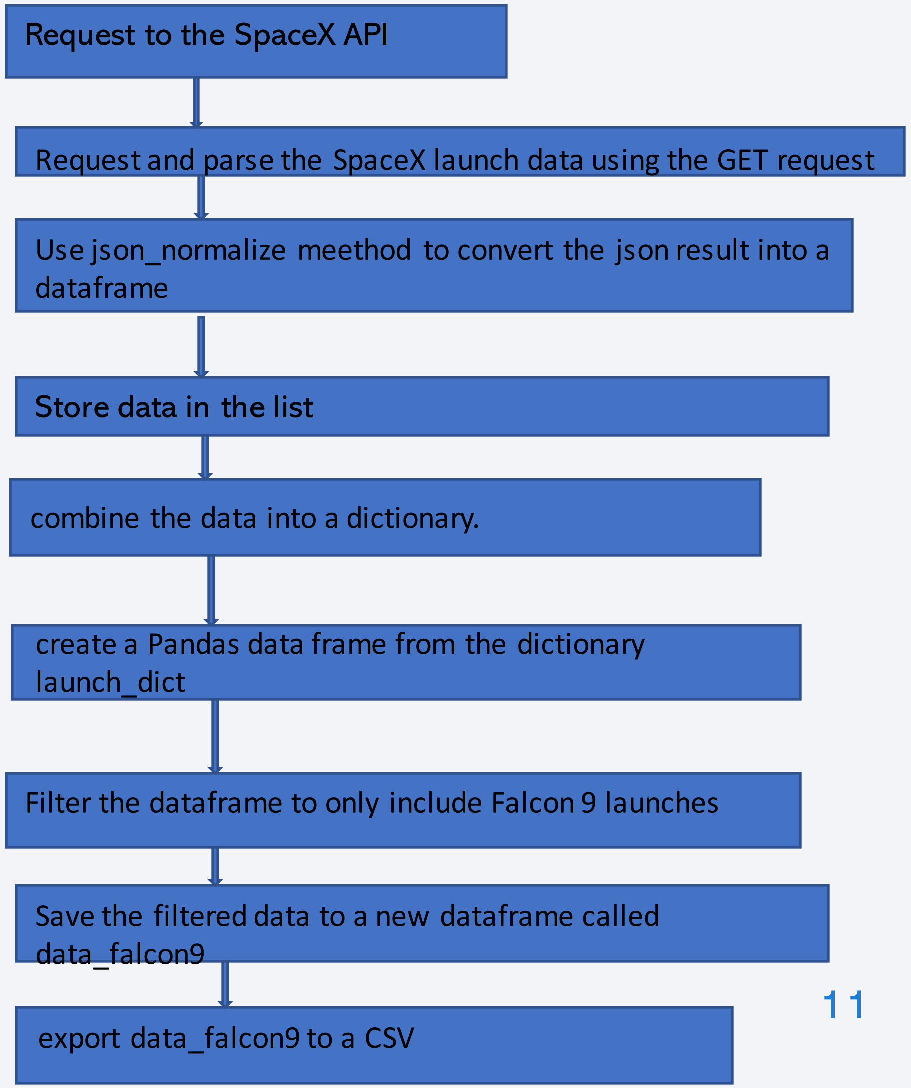
Perform predictive analysis using classification models

- Create a NumPy array from the column Class in data, by applying the method to_numpy()
- Standardize the data in X then reassign it to the variable X using the transform
- Use the function train_test_split to split the data X and Y into training and test data
- Create a logistic regression object using then create a GridSearchCV object logreg_cv with cv = 10
- Calculate the accuracy of logreg_cv on the test data using the method score
- Create a support vector machine object then create a GridSearchCV object svm_cv with cv – 10
- Calculate the accuracy of svm_cv on the test data using the method score
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10
- Calculate the accuracy of tree_cv on the test data using the method score
- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10
- Calculate the accuracy of tree_cv on the test data using the method score
- Find the method performs best

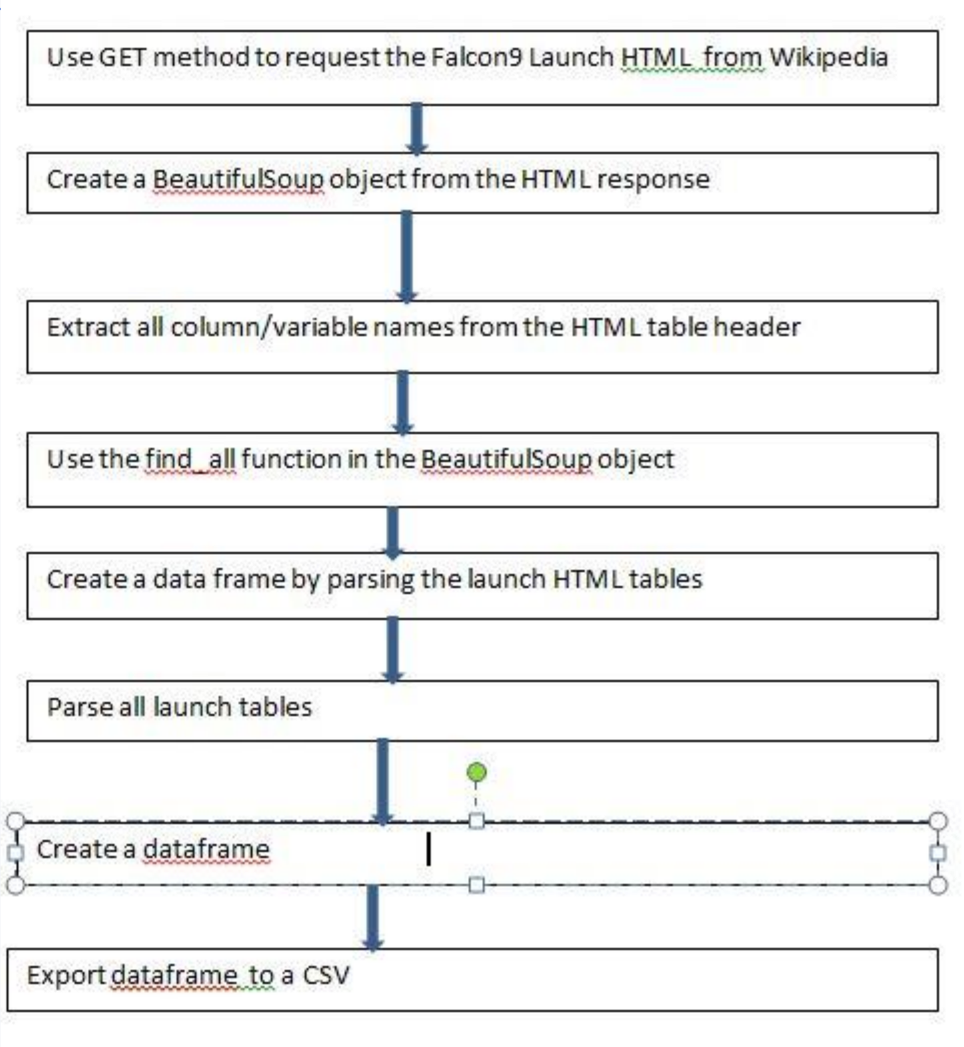
Data Collection

Describe how data sets were collected.

- Request to the SpaceX API
- Taking the dataset and using the rocket column, uses the launchpad column, the payloads column
- And the cores column to call the API and append the data to the lists
- Request and parse the SpaceX launch data using the GET request
- Use json_normalize meethod to convert the json result into a dataframe
- Store data in the list
- combine the columns or data into a dictionary.
- create a Pandas data frame from the dictionary launch_dict.
- Filter the dataframe to only include Falcon 9 launches
- Save the filtered data to a new dataframe called data_falcon9.
- export data_falcon9 to a CSV



Data Collection - Scraping



https://github.com/ibmlab6/testrepo/tree/master

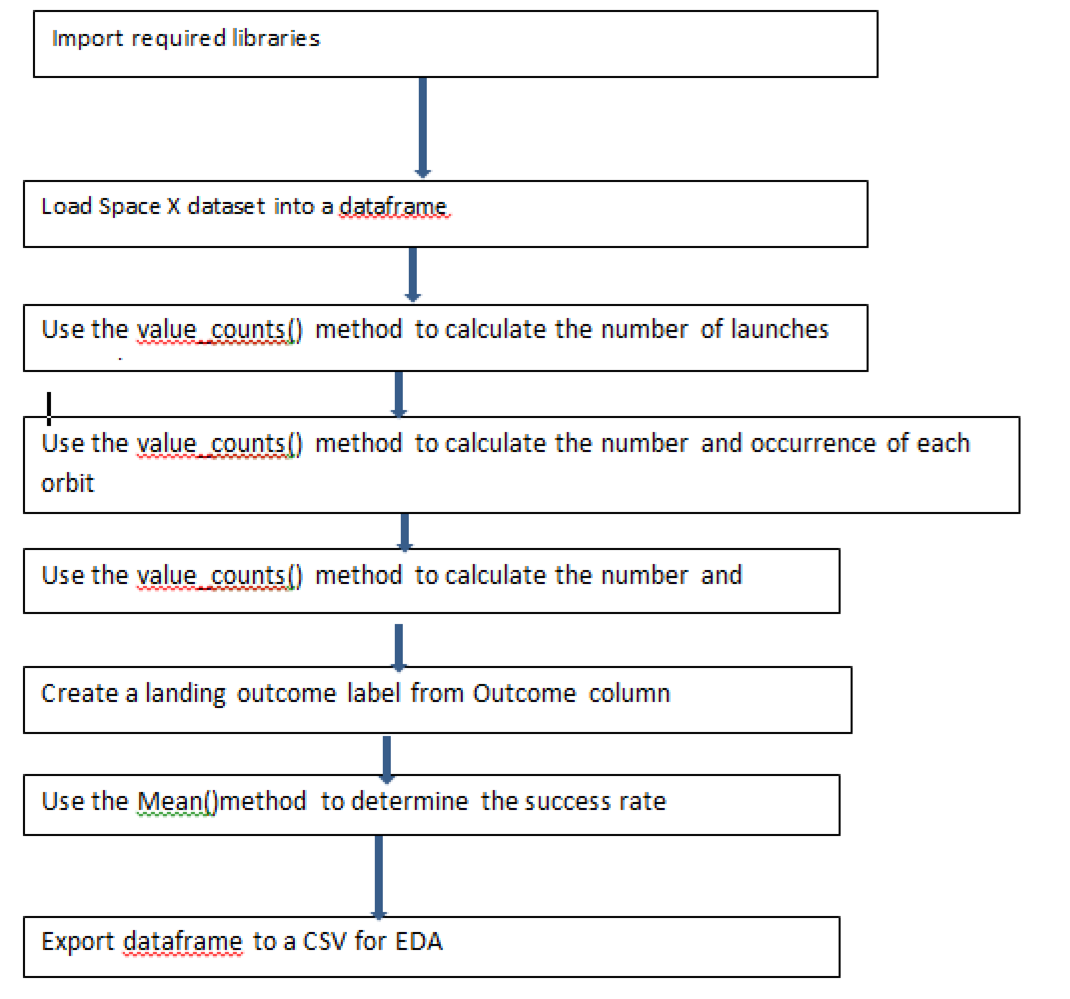
Data Wrangling

- Describe how data were processed

- Import required libaries
- Put Space X dataset into a dataframe
- Identify and calculate the percentage of the missing values
- Identify which columns are numerical and categorical
- Calculate the number of launches on each site(use value_counts()method)
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- determine the sucess rate
- export dataframe to a CSV for EDA

GitHub URL

https://github.com/ibmlab6/testrepo/tree/master



EDA with Data Visualization

Summarize what charts were plotted and why you used those charts

scatter point charts

- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit Type
- Orbit VS. Payload Mas

We used scatter point charts to visualize the relationship between two variables.

GitHub URL

https://github.com/ibmlab6/testrepo/tree/master

Bar Graph

success rate VS. orbit type

To visualize any relationship between success rate and orbit type, we create a bar chart for the success rate of each orbit

Line Graph

Success Rate VS. Year

To visualize the launch success yearly trend and to get the average launch success trend, we create a line chart with x axis to be Year and y axis to be average success rate

EDA with SQL

Using bullet point format, summarize the SQL queries you performed

- Displaying 5 records where launch sites begin with the string 'CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was acheived
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for the in year 2015
- Ranking 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

https://github.com/ibmlab6/testrepo/tree/master

Build an Interactive Map with Folium

- To visualize launch site locations, we create a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas and a blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name.
- We create and add folium.Circle and folium.Marker for each launch site on the site map and mark the success/failed launches for each site on the map.
- We create markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
- Marker clusters can be a good way to simplify a map containing many markers having the same coordinate.

We calculate the distance between two points on the map based on their Lat and Long values using the following method:

- Mark down a point on the closest railway using MousePosition and calculate the distance between the railway point to the launch site.
- After obtained its coordinate, create a folium.Marker to show the distance
- Draw a PolyLine between a launch site to the selected
- We also draw a line between a launch site to its closest city, coastline, highway.

After we plot distance lines to the proximities, we can answer the following questions easily:

- Are launch sites in close proximity to railways? N0
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

GitHub URL

https://github.com/ibmlab6/testrepo/tree/master

Build a Dashboard with Plotly Dash

Summarize what plots/graphs and interactions you have added to a dashboard

- Add a Launch Site Drop-down Input Component
- Add a callback function to render success-pie-chart based on selected site dropdown
- Add a Range Slider to Select Payload
- Add a callback function to render the success-payload-scatter-chart scatter plot

Explain why you added those plots and interactions

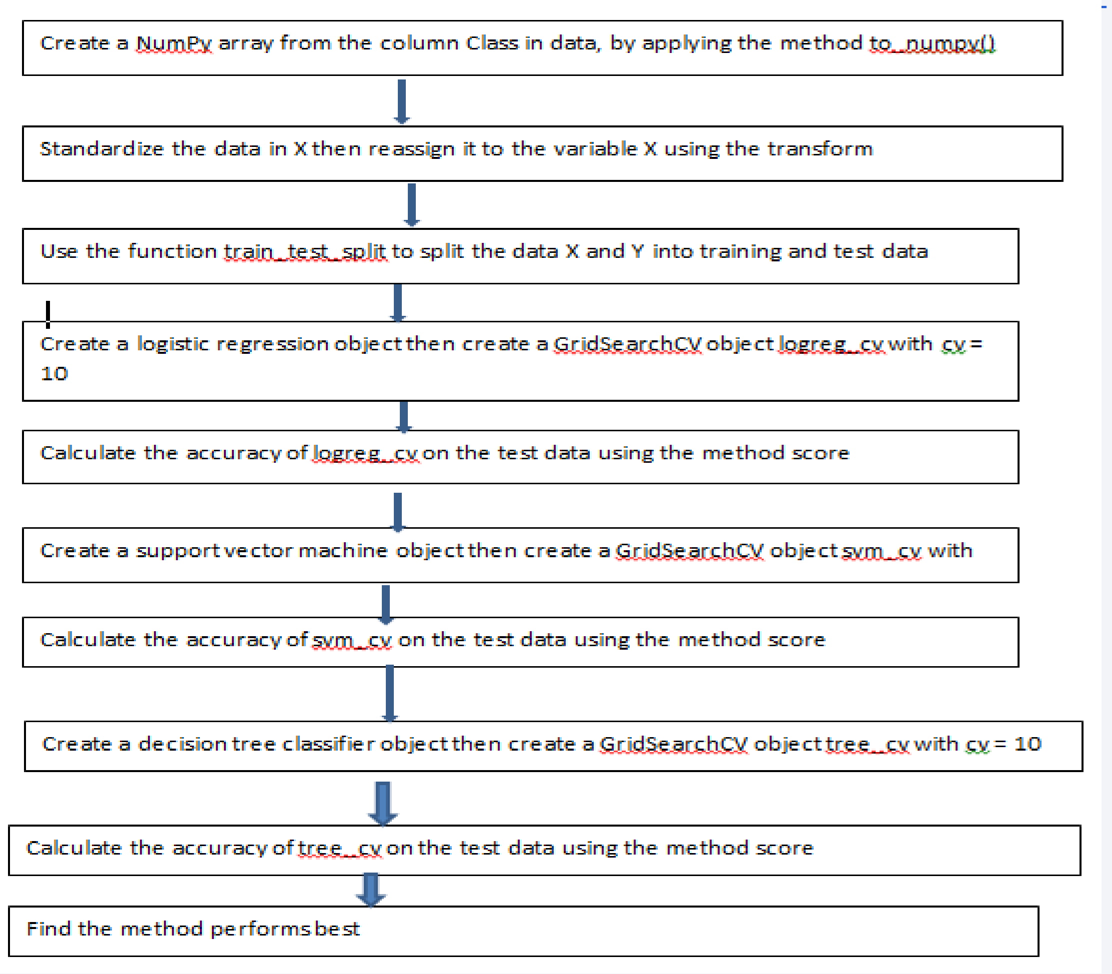
- We built a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time
- We add pie chart to the dashboard to visualize launch success counts
- We add a Range Slider to Select Payload to the dashboard to find if variable payload is correlated to mission outcome and to select different payload range and to identify some visual patterns.
- We add payload scatter plot to the dashboard to observe how payload may be correlated with mission outcomes for selected site(s).

Predictive Analysis (Classification)

- **Perform predictive analysis using classification models**
- Create a NumPy array from the column Class in data, by applying the method to_numpy()
- Standardize the data in X then reassign it to the variable X using the transform
- Use the function train_test_split to split the data X and Y into training and test data
- Create a logistic regression object using then create a GridSearchCV object logreg_cv with cv = 10
- Calculate the accuracy of logreg_cv on the test data using the method score
- Create a support vector machine object then create a GridSearchCV object svm_cv with cv – 10
- Calculate the accuracy of svm_cv on the test data using the method score
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10
- Calculate the accuracy of tree_cv on the test data using the method score
- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10
- Calculate the accuracy of tree_cv on the test data using the method score
- Find the method performs best

GitHub URL

https://github.com/ibmlab6/testrepo/tree/master



Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

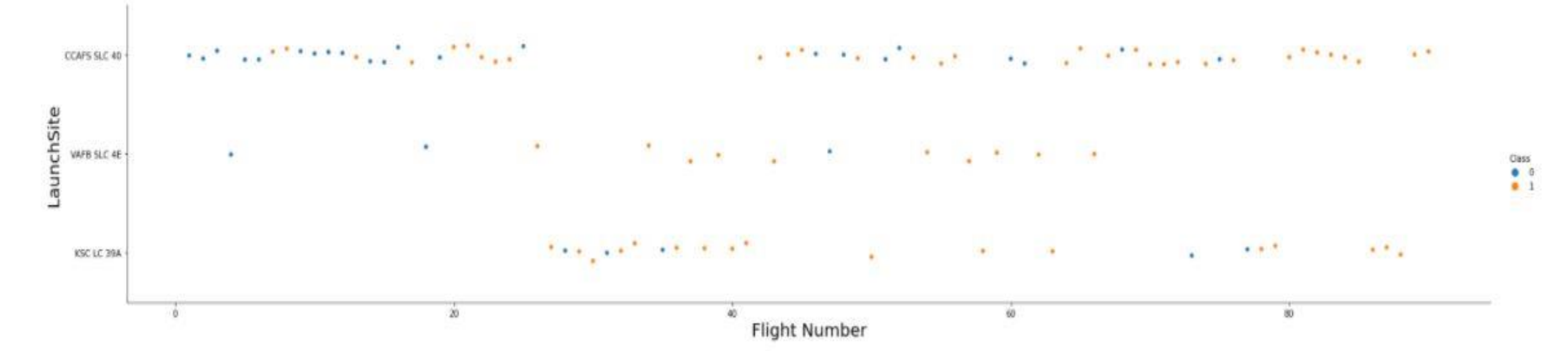
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

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

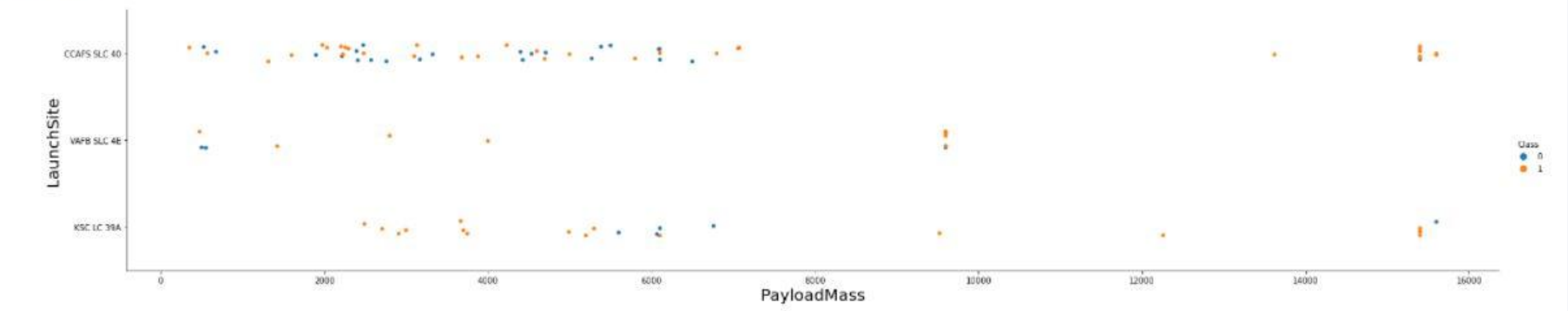
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

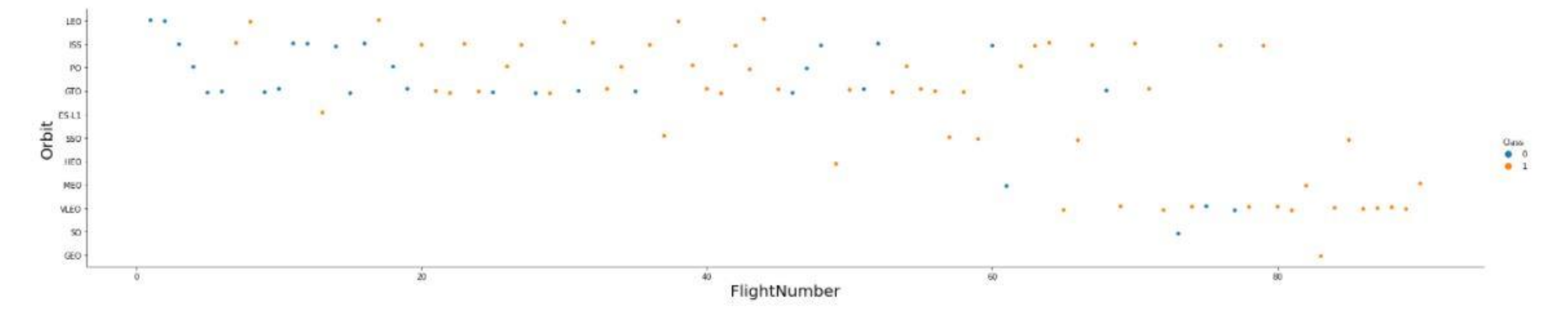
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

- Show a scatter point of 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 = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

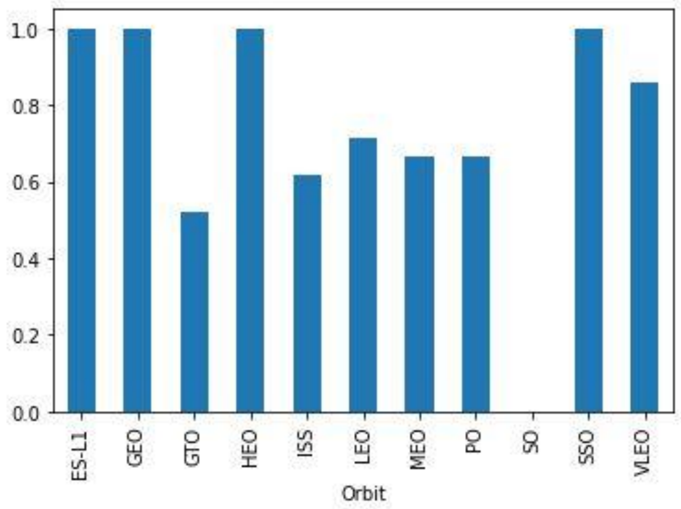


Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

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

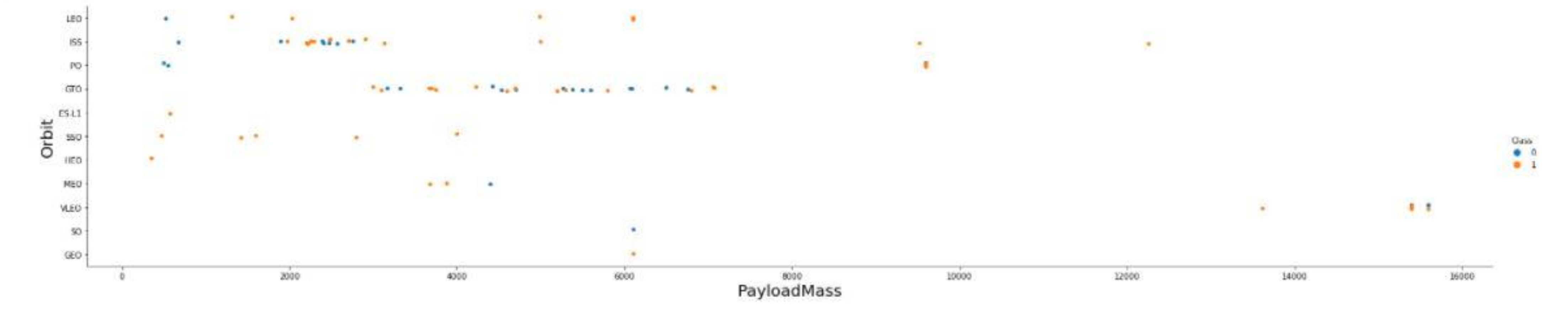
```
]: <AxesSubplot:xlabel='Orbit'>
```



Payload vs. Orbit Type

- Show a scatter point of 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",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

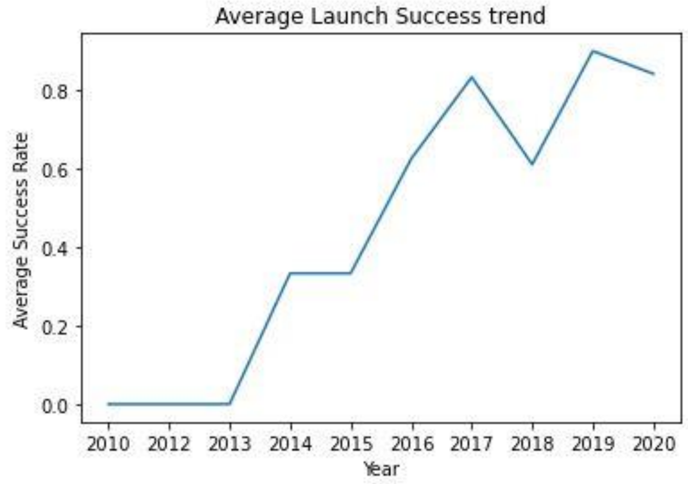


Launch Success Yearly Trend

- Show a line chart of yearly average success rate

```
df1 = pd.DataFrame(Extract_year(df['Date']), columns=['year'])
df1['Class']=df['Class']
sns.lineplot(x=np.unique(Extract_year(df['Date'])), y=df1.groupby('year')['Class'].mean())

plt.title('Average Launch Success trend')
plt.xlabel('Year')
plt.ylabel('Average Success Rate')
plt.show()
```



All Launch Site Names

- Find the names of the unique launch sites

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

We found the names of unique launch sites by using a query with select ,distinct column name from SPACXTBL.

```
%sql SELECT distinct launch_site from SPACEXTBL
```

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Present your query result with a short explanation here

We found 5 records where launch sites begin with `CCA` by using like ‘CCA%’ by using a query with (like ‘CCA%’ , limit 5).

```
%sql Select LAUNCH_SITE from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

total_payload_mass
45596

Present your query result with a short explanation here

We calculate the total payload carried by boosters from NASA by using a query with sum(PAYLOAD_MASS_KG_)

```
%sql Select Sum(PAYLOAD_MASS__KG_)as total_payload_mass from SPACEXTBL where CUSTOMER='NASA (CRS)'
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

avg_payload
2928

- Present your query result with a short explanation here

We calculate the average payload mass carried by booster version F9 v1.1 by using a query with Avg(PAYLOAD_MASS_KG_),

(where)condition

```
%sql Select Avg(PAYLOAD_MASS__KG_) as avg_payload from SPACEXTBL where BOOSTER_VERSION='F9 v1.1'
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

DATE	landing__outcome
2015-12-22	Success (ground pad)

- Present your query result with a short explanation here

We found the dates of the first successful landing outcome on ground pad by using a query with Min(date),where(condition),Group by

```
%%sql Select MIN(Date) as date,Landing__Outcome from SPACEXTBL where Landing__Outcome='Success (ground pad)' Group by Landing__Outcome
```

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

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- Present your query result with a short explanation here

We list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 by using a query

With select, where , and ,<,>

```
%sql Select BOOSTER_VERSION,Landing__Outcome,PAYLOAD_MASS__KG_ from SPACEXTBL
where Landing__Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_< 6000
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

sucess_mission_outcome	failure_mission_outcome
100	1

- Present your query result with a short explanation here

We calculate the total number of successful and failure mission outcomes by using a query with distinct, select ,as

```
%sql Select distinct(Select count (MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME LIKE '%Success%') as Sucess_Mission_Outcome,
(Select count (MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME LIKE '%Failure%')as Failure_Mission_Outcome from SPACEXTBL
```

Boosters Carried Maximum Payload

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

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

We list the names of the booster which have carried the maximum payload mass by using a sub query with where PAYLOAD_MASS__KG_ =(select Max (PAYLOAD_MASS__KG_) from SPACEXTBL)

```
%sql Select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ =( select Max (PAYLOAD_MASS__KG_) from SPACEXTBL)
```

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

We list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 by using a query with

where Landing__Outcome='Failure (drone ship)' and Year(Date)='2015'

```
%sql Select Landing__Outcome,BOOSTER_VERSION,LAUNCH_SITE,date from SPACEXTBL where Landing__Outcome='Failure (drone ship)' and Year(Date)='2015';
```

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

landing__outcome	RANK
Uncontrolled (ocean)	2
Success (ground pad)	3
Success (drone ship)	5
recluded (drone ship)	1
No attempt	10
Failure (parachute)	2
Failure (drone ship)	5
Controlled (ocean)	3

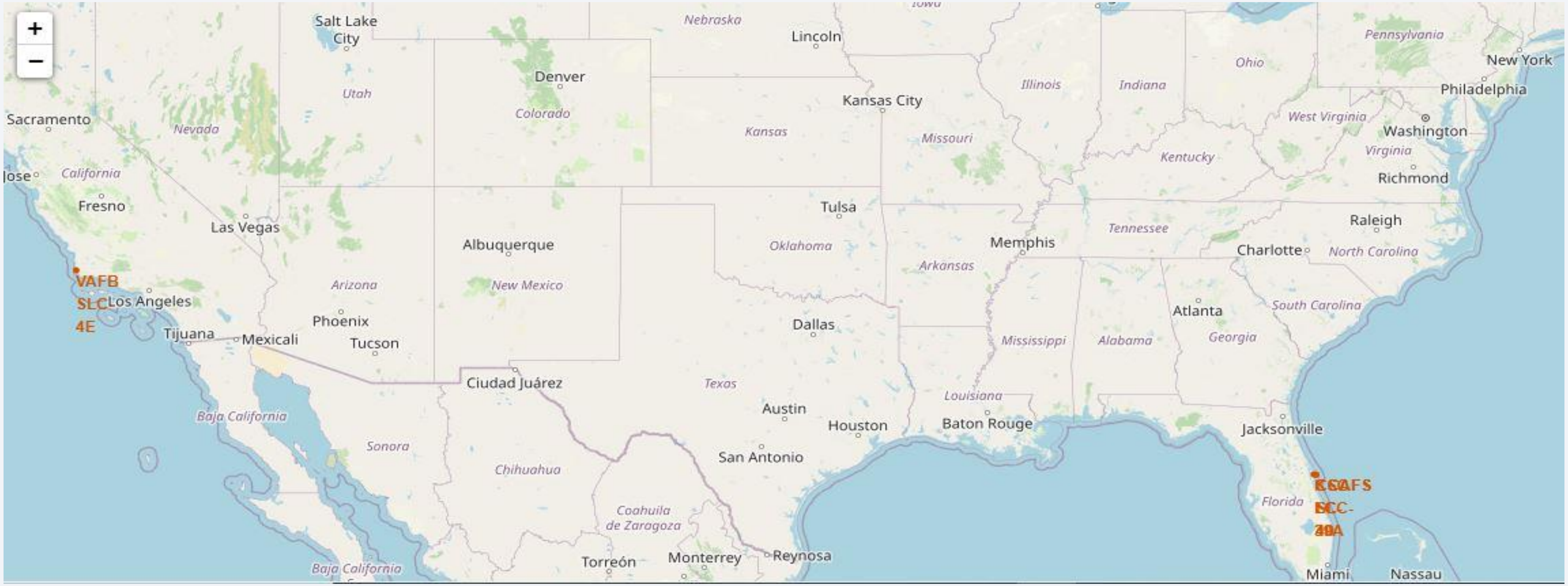
We rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order by using a query with count(Landing__Outcome)as Rank, between and,group by,order by desc

```
%sql Select Landing__Outcome, count(Landing__Outcome)as Rank from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20' group by Landing__Outcome order by Landing__Outcome desc;
```

Section 4

Launch Sites Proximities Analysis

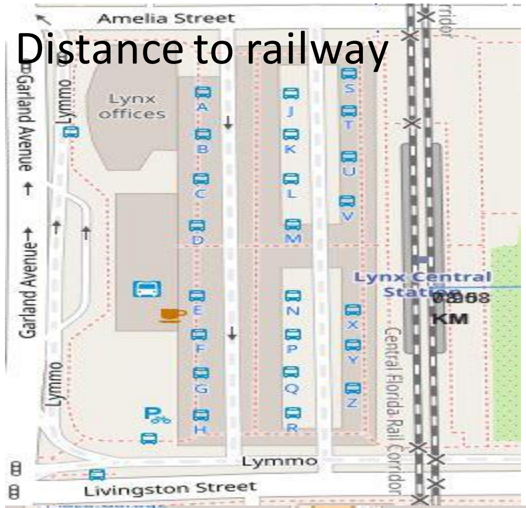
All launch sites' location markers on a global map



We can see that all spaceX launch sites are in Florida and California coasts in USA.

Color-labeled launch outcomes on the map

Green markers show successful Launch sites and red markers show failure Launch sites a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



After we plot distance lines to the proximities,

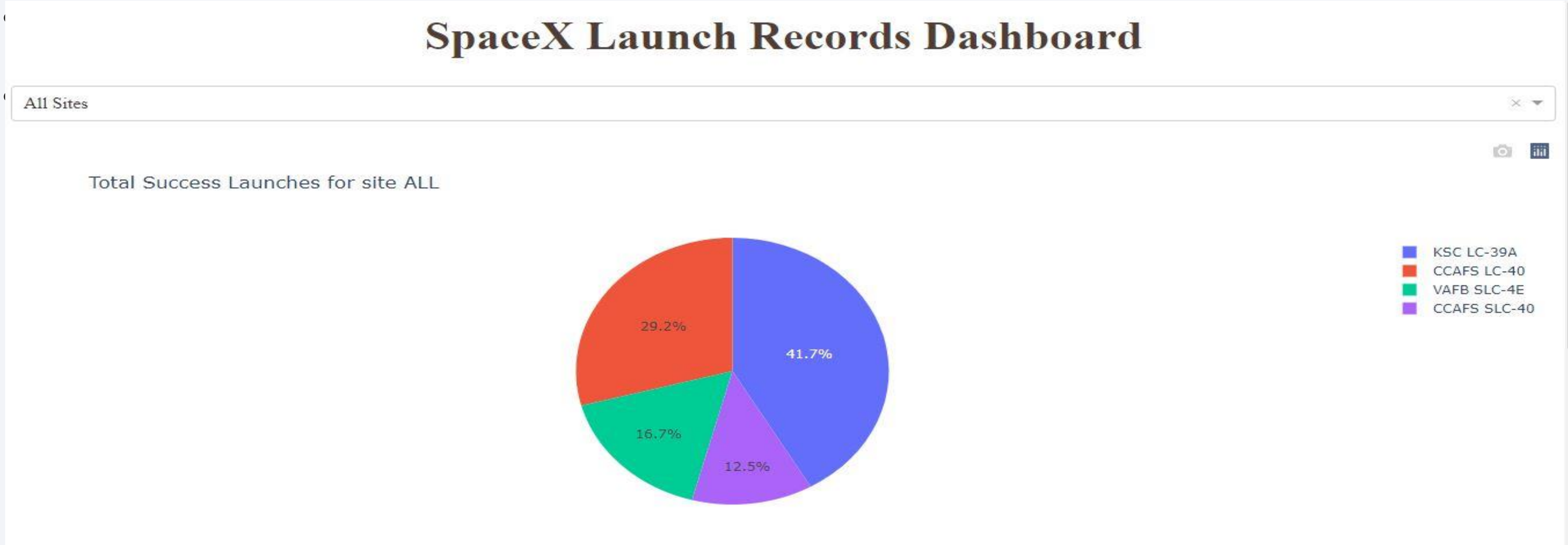
* Are launch sites in close proximity to railways? No
* Are launch sites in close proximity to highways? No
* Are launch sites in close proximity to coastline? Yes
* Do launch sites keep certain distance away from cities? Yes

Section 5

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

Total Success Launches for site ALL

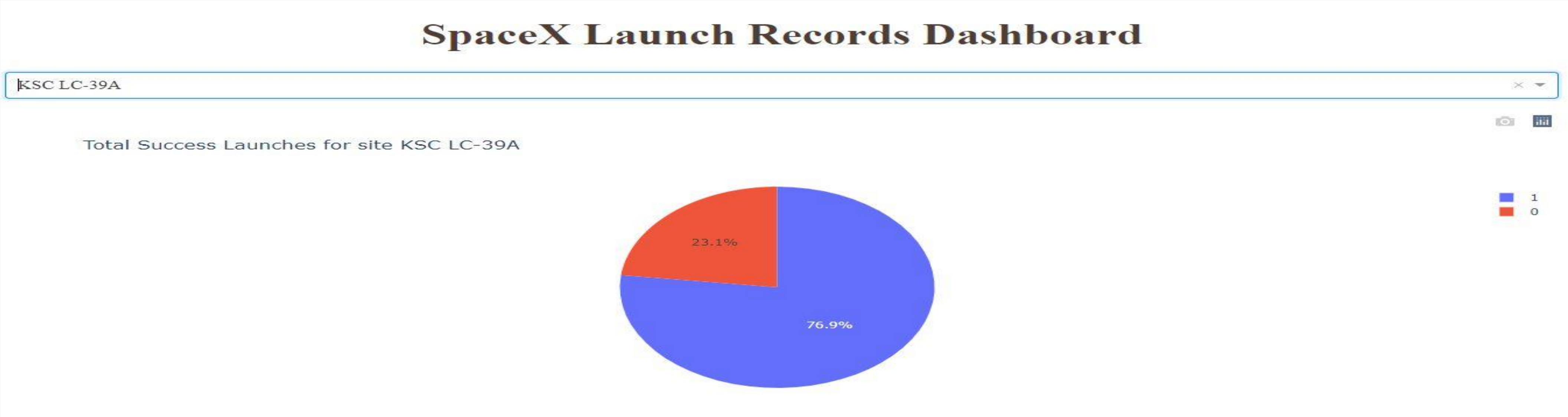


KSC LC-39A has the largest successful launches rates from all the launch sites.

SpaceX Launch Records Dashboard

Total Success Launches for site KSC LC-39A

Show the screenshot of the piechart for the launch site with highest launch success ratio

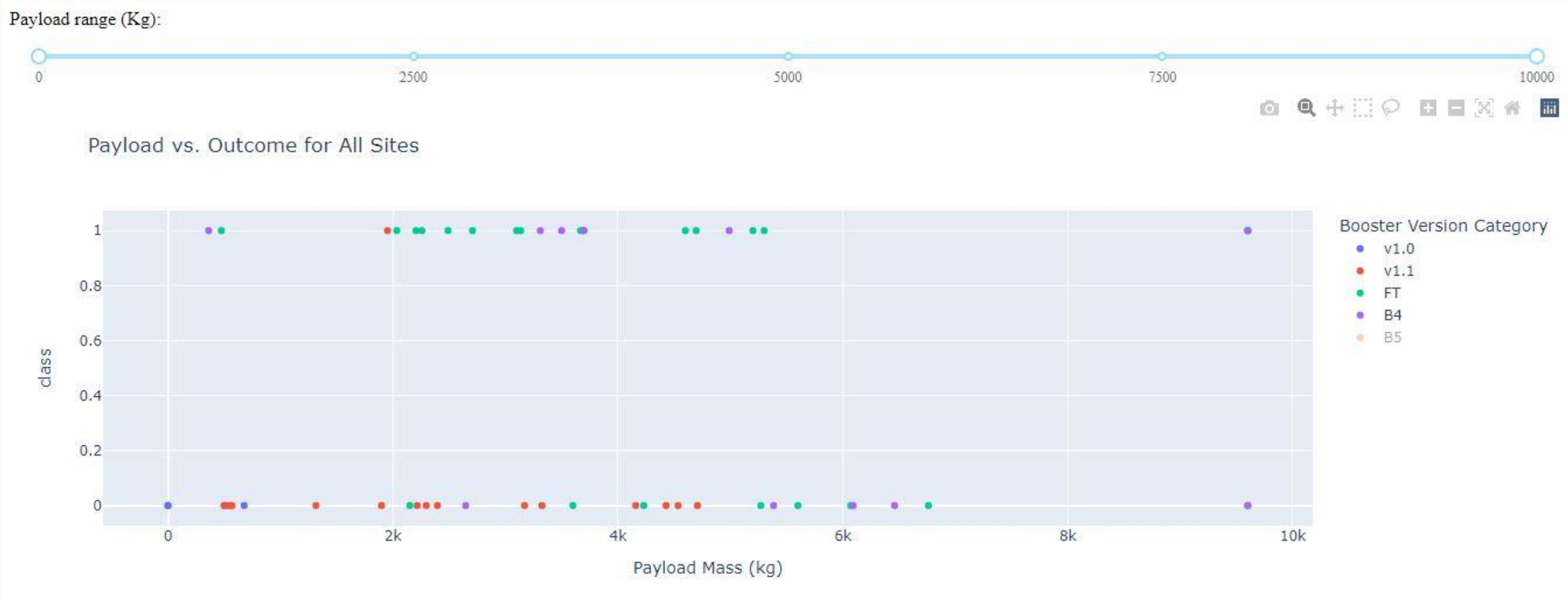


We found that KSC LC-39A has 76.9% success rate and 23.1% failure rate

SpaceX Launch Records Dashboard

Payload vs. Outcome for All Sites

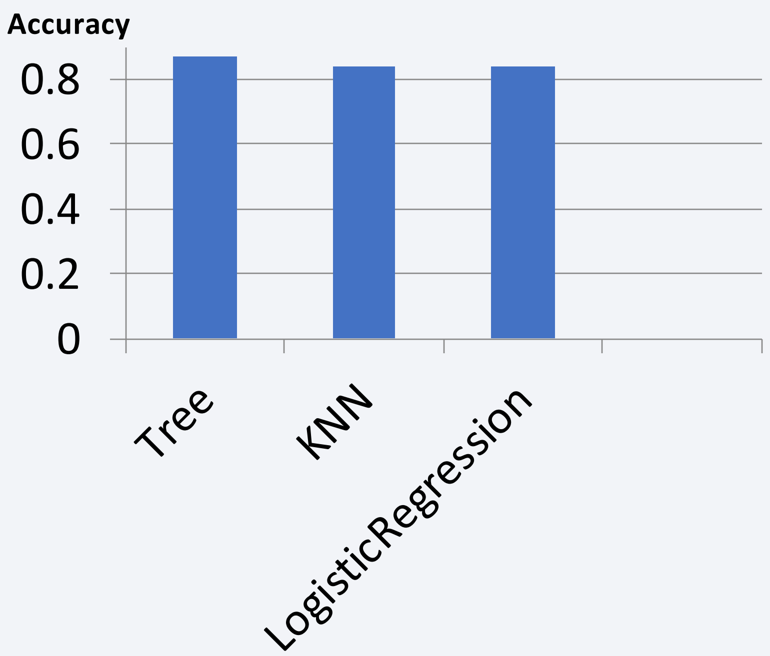
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
-



Section 6

Predictive Analysis (Classification)

Classification Accuracy

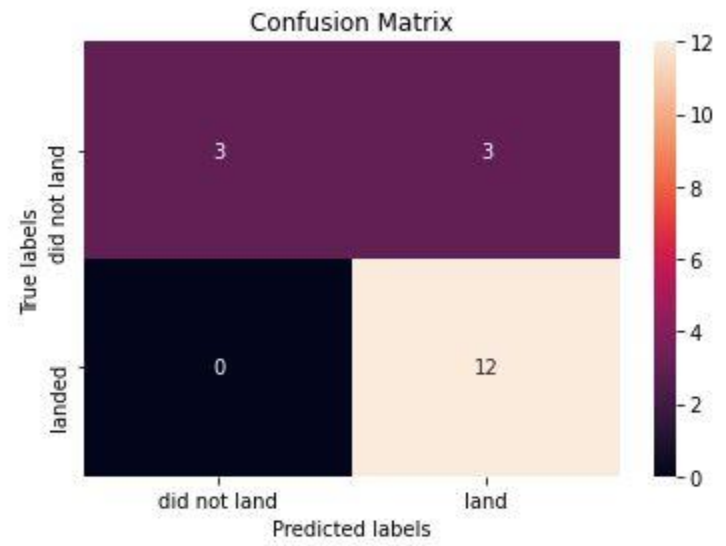



```
Best Algorithm is Tree with a score of 0.875
Best Params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2,
'splitter': 'best'}
```

After selecting the best hyperparameters for the decision tree classifier, we achieved 0.875 accuracy on the test data

Confusion Matrix

The confusion matrix of the best performing model

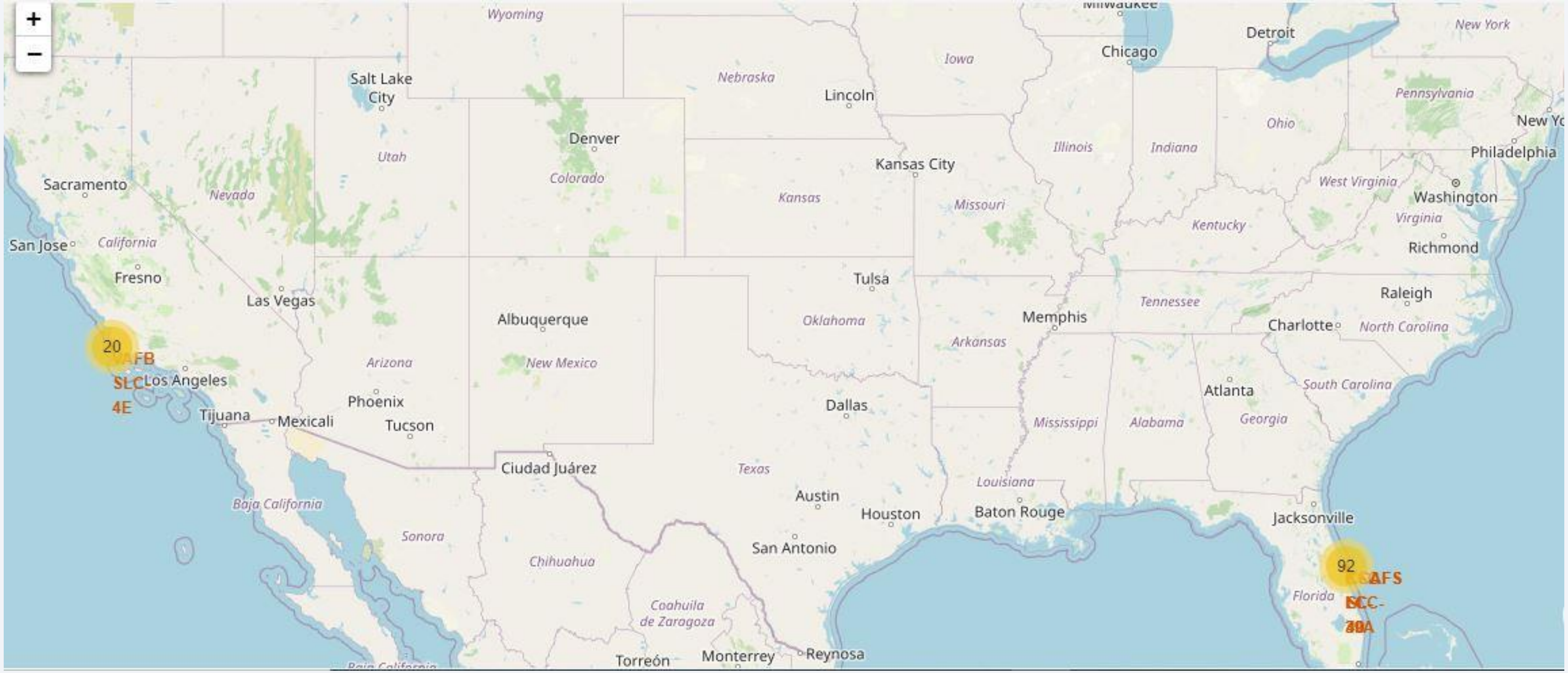


Conclusions

- The decision tree classifier Algorithm is the best for Space X Falcon 9 First Stage Landing Prediction
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX Falcon 9 are directly proportional time in years they will eventually perfect the launches
- From all the launch sites, KSC LC-39A had the most successful launch
- Orbit GEO,HEO,SSO,ES-L1 have the best success rate

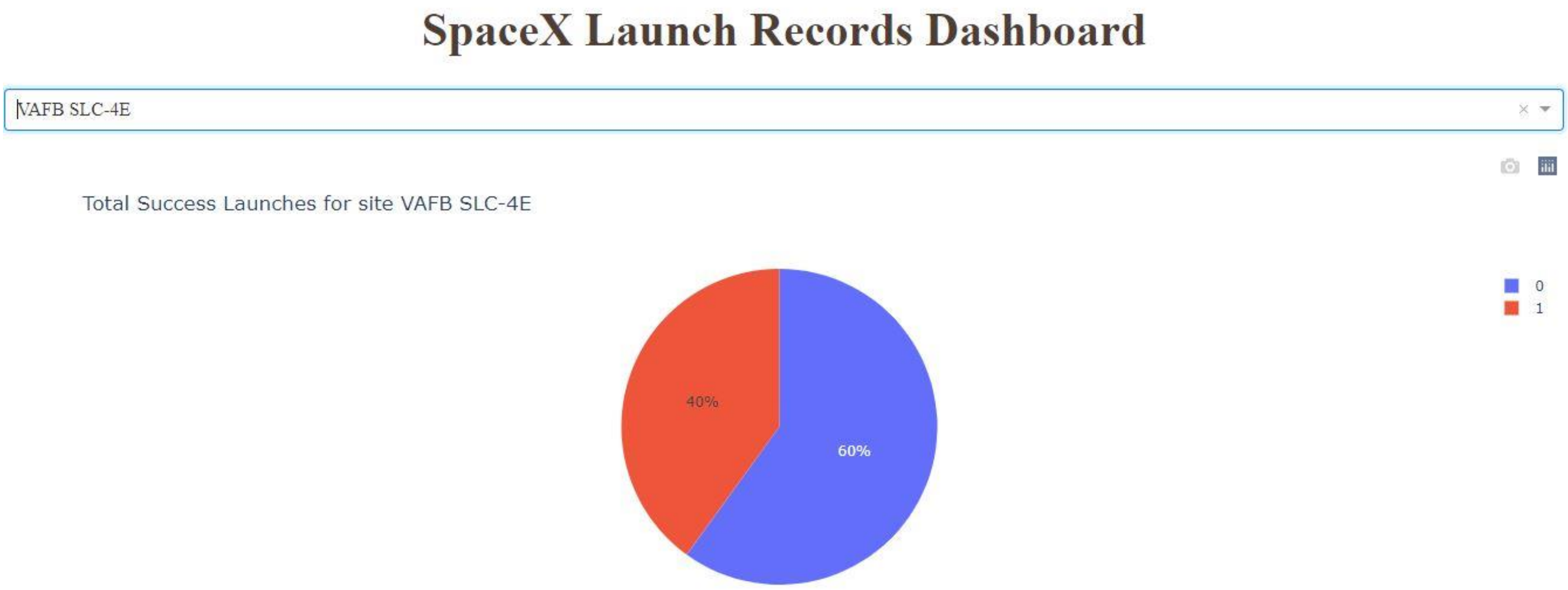
Appendix

folium map with marker from slide 38

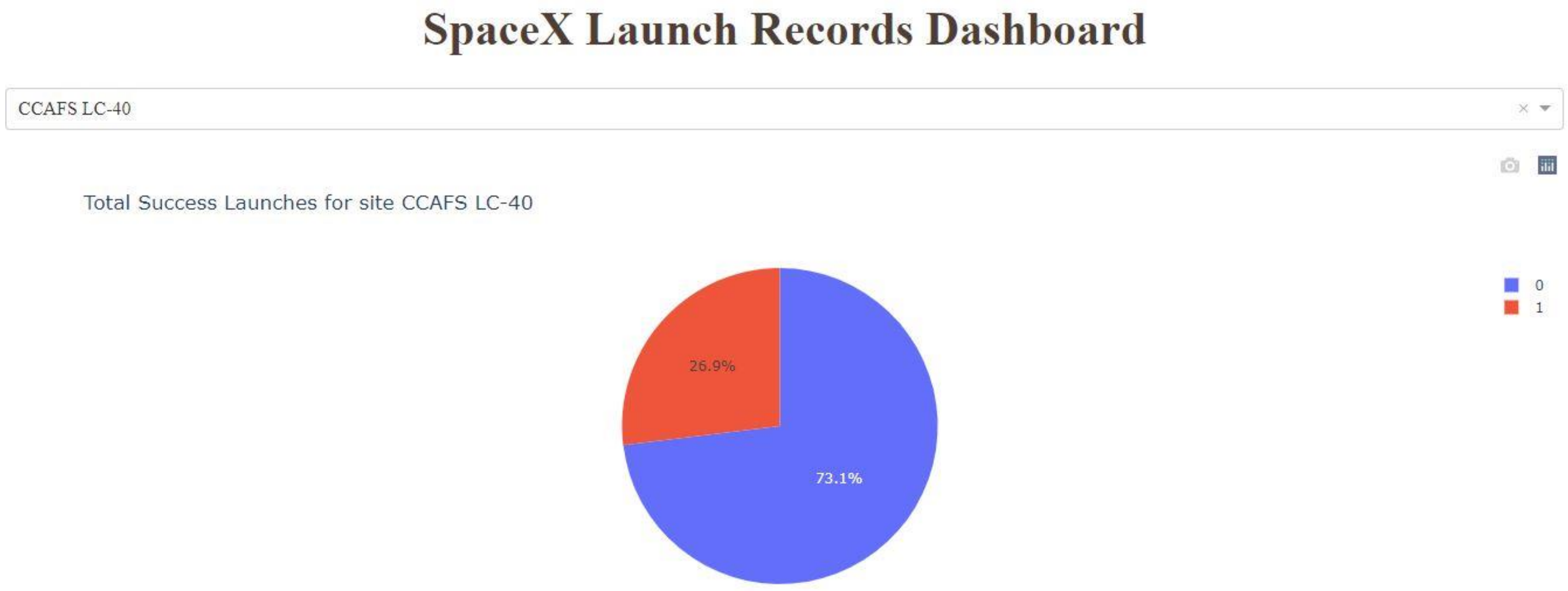


Appendix

From Section 5



Appendix


SpaceX Launch Records Dashboard
CCAFS LC-40
Total Success Launches for site CCAFS LC-40
0
1
26.9%
73.1%

Thank you!