IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Prajapati Swapnilkumar>
<03-11-2021>

Outline

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

Executive Summary

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we have determine the cost of a launch.
- Afterwards, we analyze geo spatial data using folium and found out launch sites and then used machine learning to predict if the 1st stage will be successful or not.

Introduction

- Space X 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 Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we have determine the cost of a launch.

- In following scenario where we want to predict whether the SpaceX will reuse their 1st stage and also determine the cost of the launch by the gathered data
- Several methodologies has been used to achieve this task such as geo spatial analysis and visualizations and finally machine learning.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data has been collected two ways: 1) Web scrapping 2)Through SpaceX API
- Perform data wrangling
    - Data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- EDA with data visualization
- EDA with SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- The SpaceX dataset was collected through both API and web scraping using Beautiful soup library.
- The API used were following:

1. Request data from rockets api

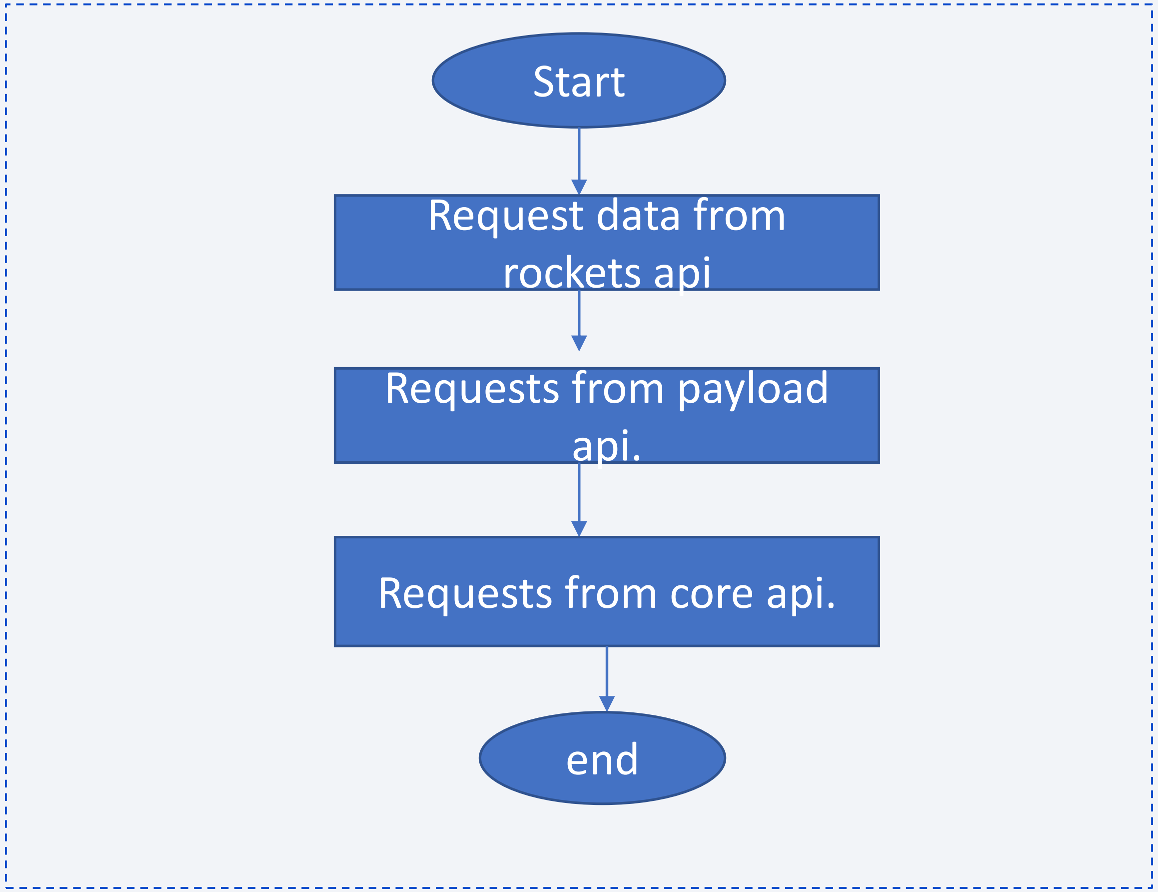
2.Requests from payload api.

3. Requests from core api.

Data Collection - SpaceX API

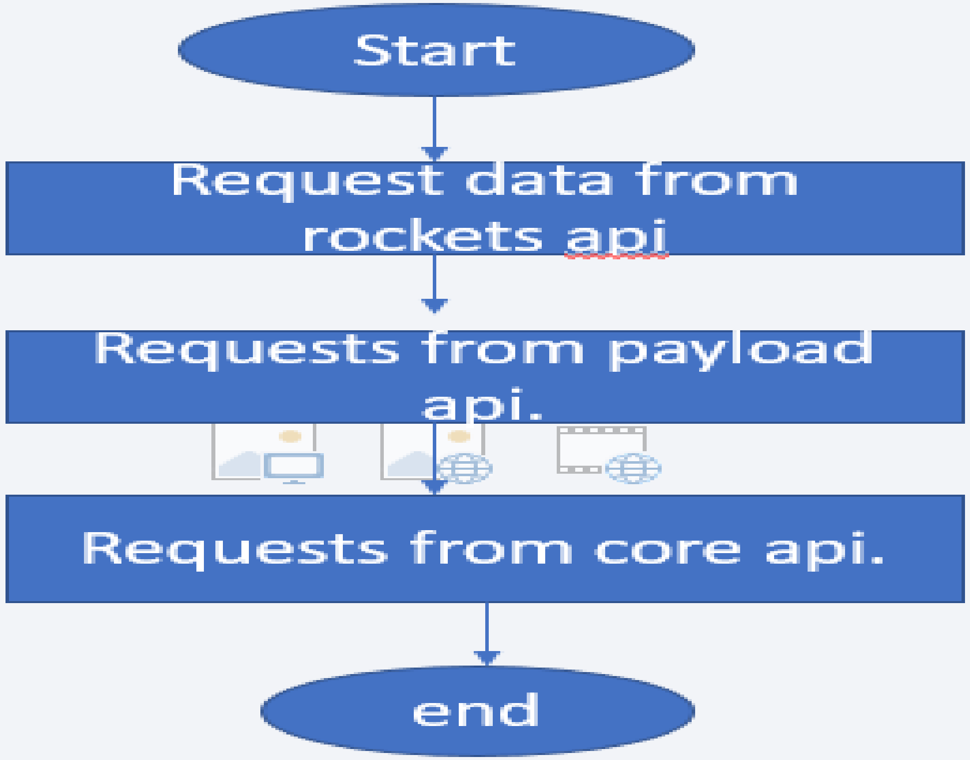
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (Data collection API), as an external reference and peer-review purpose



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- [Data collecting web-scrapping]()



Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- data wrangling

EDA with Data Visualization

- Mostly the Scatter plots and catplots charts were used to represent the relationship between the features and their correlation.
- Eg. the FlightNumber vs. PayloadMassand overlay the outcome of the launch. We see that as the flight number increases, the first stage is more likely to land successfully.
- EDA with data visualization notebook

EDA with SQL

- 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 acheived.
- 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 failed landing_outcomes in drone ship, their booster versions, and launch

Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
- interactive map with Folium map.

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y, make sure the output is a Pandas series (only one bracket df['name of column']). Then, Standardize the data in X then devide them into train and test data and used GridSearchCV to find best parameters for each algorithm.
- The machine learning algorithms used were Logistic , SVM, Decision tree and KNN. In which Tree classifier performed best with 88.88% accuracy.
- predictive analysis

Results

- Exploratory data analysis results:Exploratory Data Analysis, Preparing Data Feature Engineering

- Interactive analytics demo in screenshots:

- Predictive analysis results

Find the method performs best:

```
scores = [lr_score,svm_score,tree_score,knn_score]
print(scores)
print(scores.index(max(scores)))
```

```
[0.8333333333333334, 0.8333333333333334, 0.8888888888888888, 0.8333333333333334]
2
```

Section 2

Insights drawn from EDA

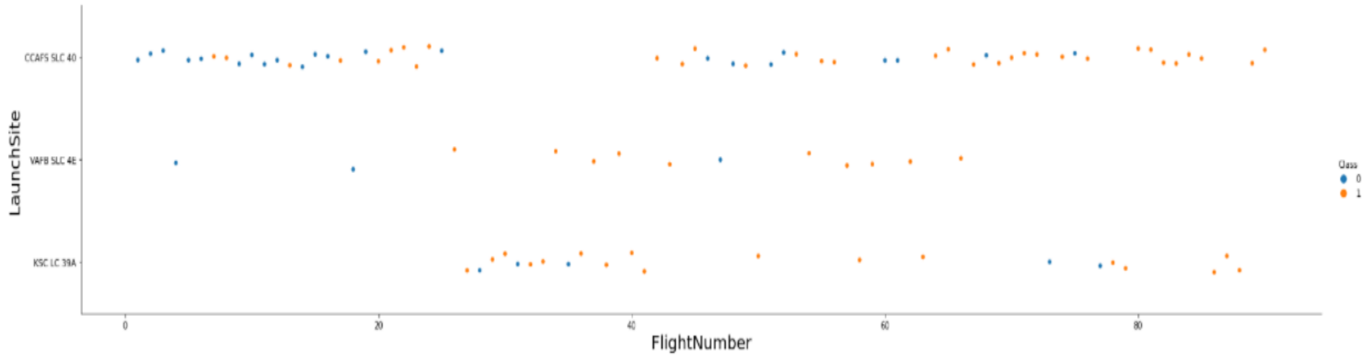
Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter x parameter to `FlightNumber`,set the y to `Launch Site` and set the parameter hue to `'class'`

```
# 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(x = "FlightNumber", y="LaunchSite", hue = "Class", data= df, aspect=5)
plt.xlabel("FlightNumber", fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

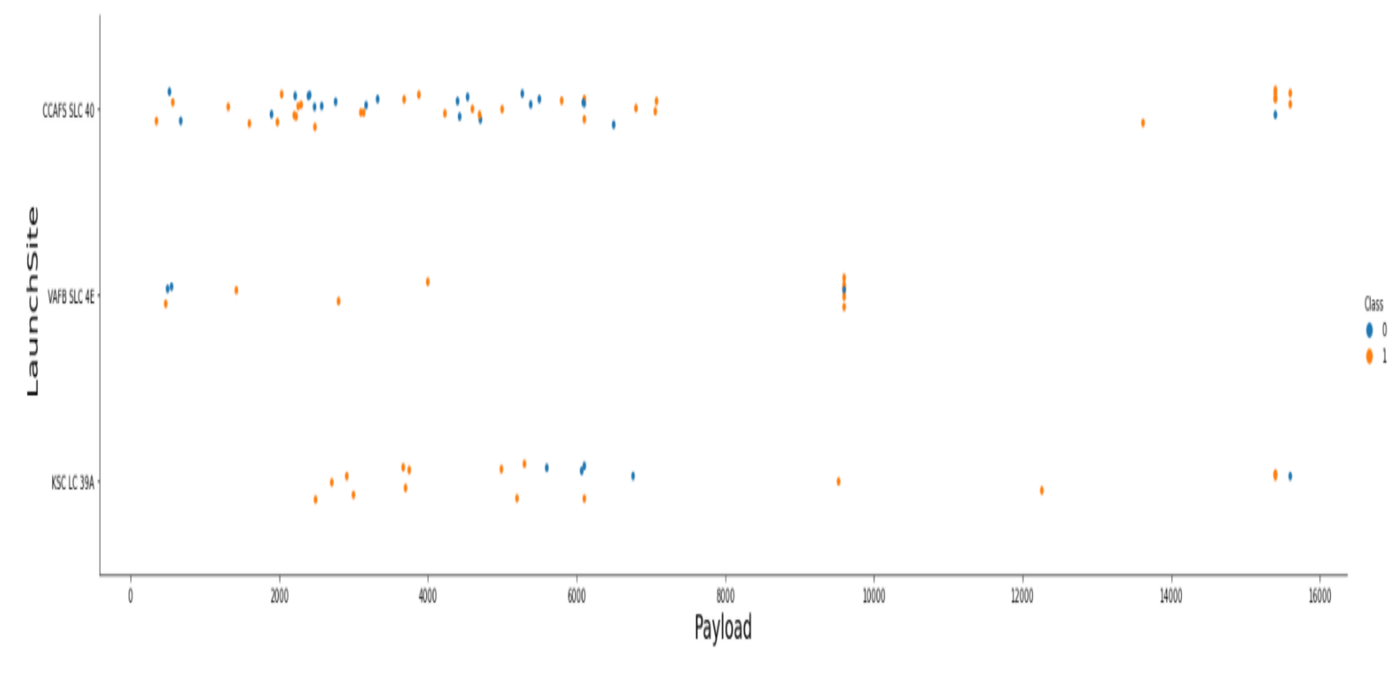


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

```
sns.catplot(x = "PayloadMass", y="LaunchSite", hue = "Class", data= df, aspect=5)
plt.xlabel("Payload", fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

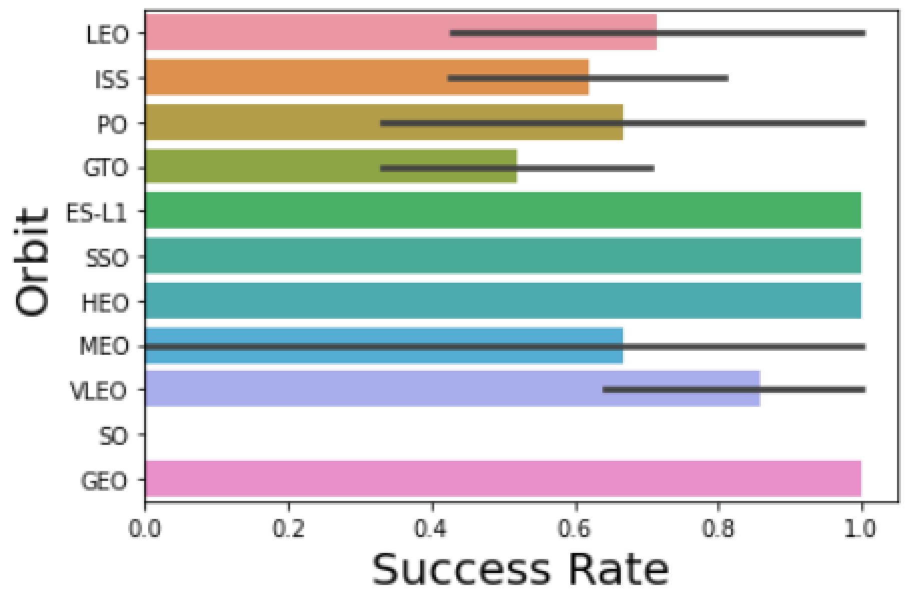


Success Rate vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations

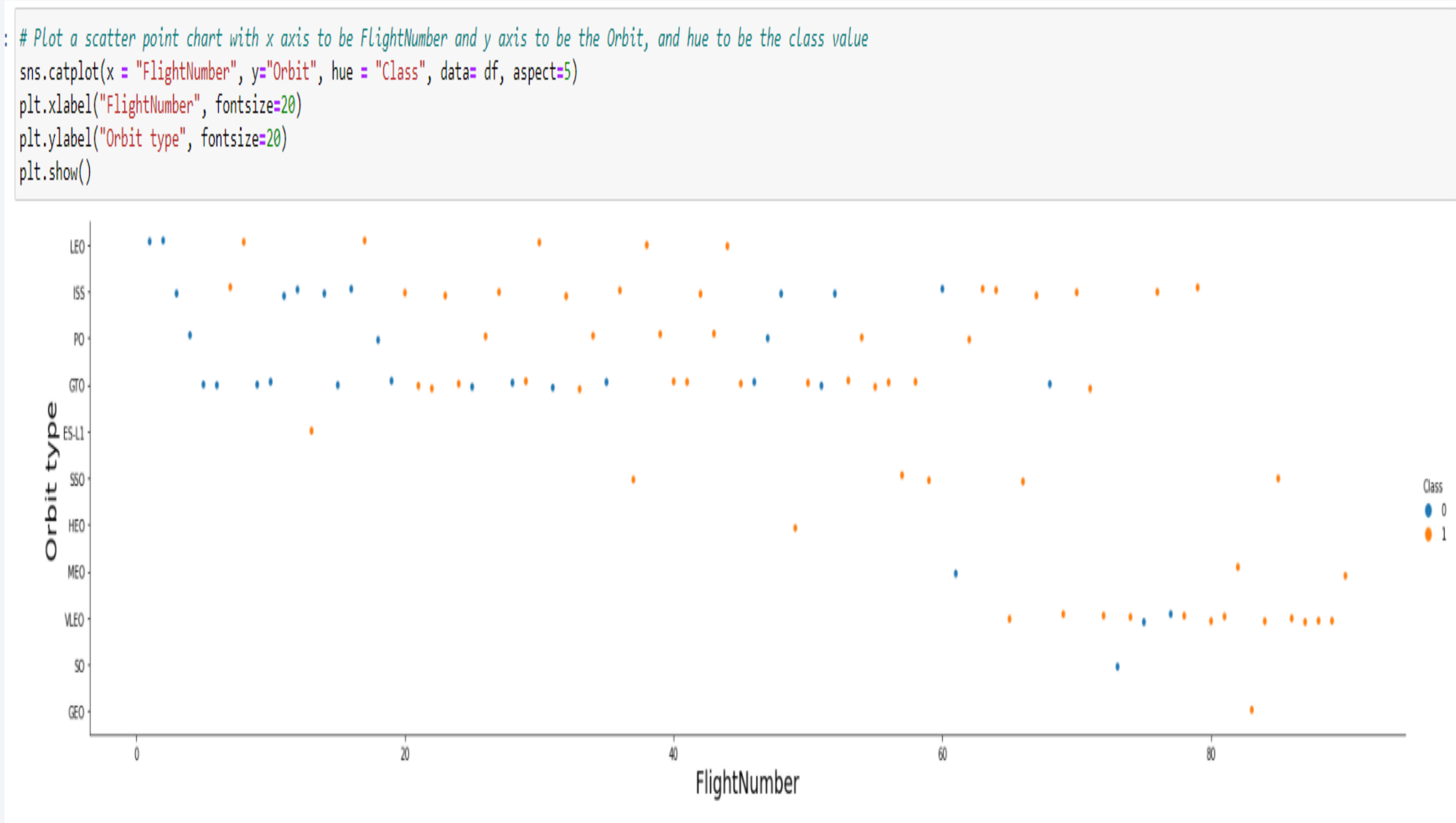
```
sns.barplot(x = "Class", y="Orbit", data= df)
plt.xlabel("Success Rate", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

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

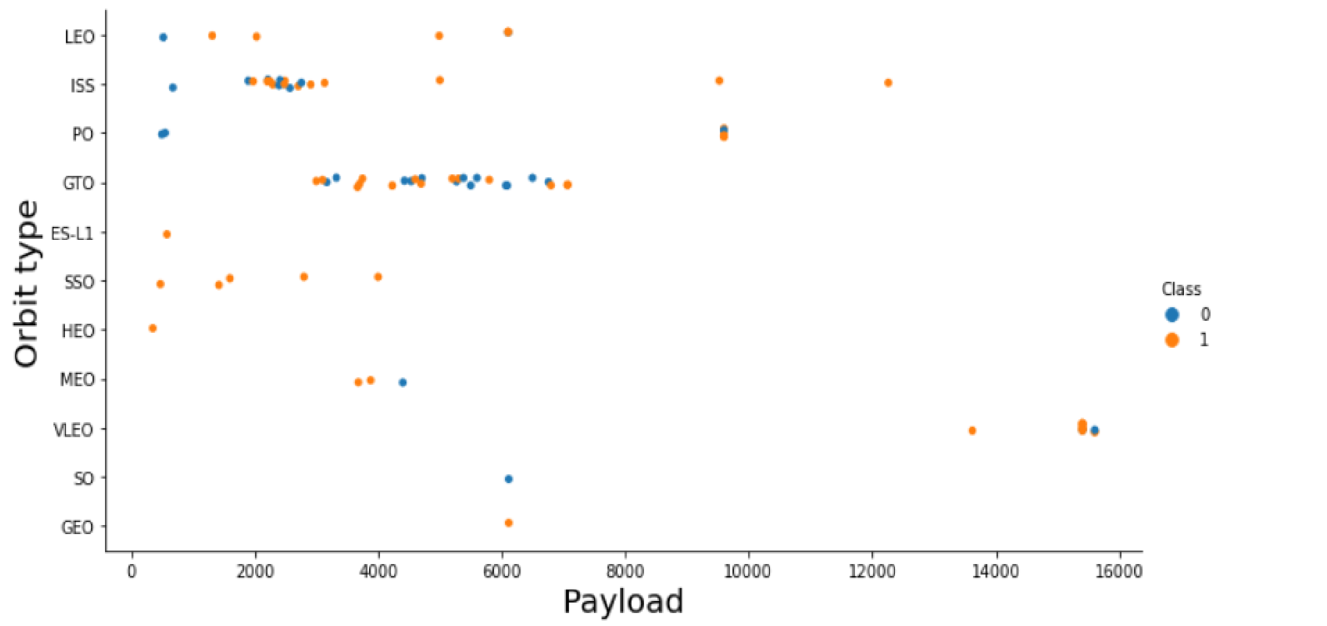
- Show the screenshot of the scatter plot with explanations



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

```
# 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(x = "PayloadMass", y="Orbit", hue = "Class", data= df, aspect=2)
plt.xlabel("Payload", fontsize=20)
plt.ylabel("Orbit type", fontsize=20)
plt.show()
```



Launch Success Yearly Trend

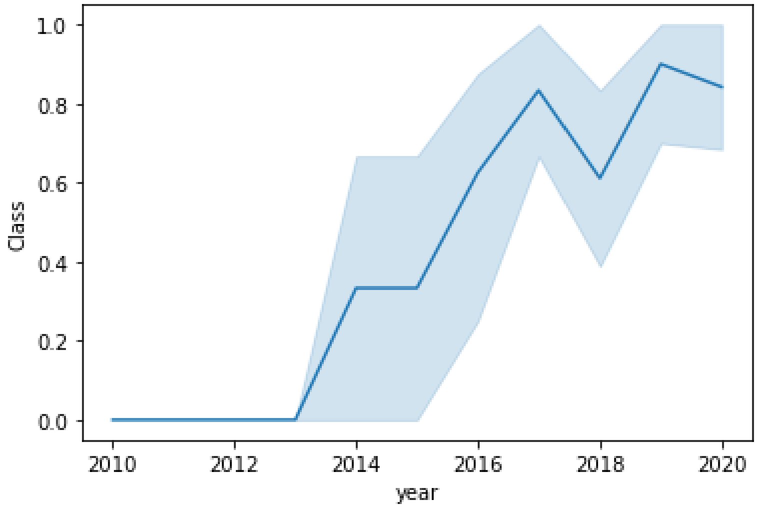
- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

```
# Plot a line chart with x axis to be the extracted year and y axis to be the succ

sns.lineplot(x="year", y="Class", data=df)
```

```
5]: <AxesSubplot:xlabel='year', ylabel='Class'>
```



All Launch Site Names

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

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

```
%%sql
SELECT Unique(LAUNCH_SITE) from Spacextbl
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9
Done.
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

```
%%sql
select LAUNCH_SITE from Spacextbl where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0t
Done.
```

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

Total Payload Mass

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

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

```
%%sql
select sum(PAYLOAD_MASS__KG_) as Total_payload_mass from SPACEXTBL
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj
Done.
```

total_payload_mass
619967

Average Payload Mass by F9 v1.1

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

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

```
%%sql
select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1og
Done.
```

payloadmass
6138

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Task 5

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

Hint:Use min function

```
%%sql
select min(DATE) from SPACEXTBL where LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu
Done.
```

1
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

- Present your query result with a short explanation here

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.a
Done.
```

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

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

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

boosterversion
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

- Present your query result with a short explanation here

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

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015'
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

- Present your query result with a short explanation here

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 FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC
%sql select max(DATE) from SPACEXTBL
```

```
 * ibm_db_sa://mvl71963:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

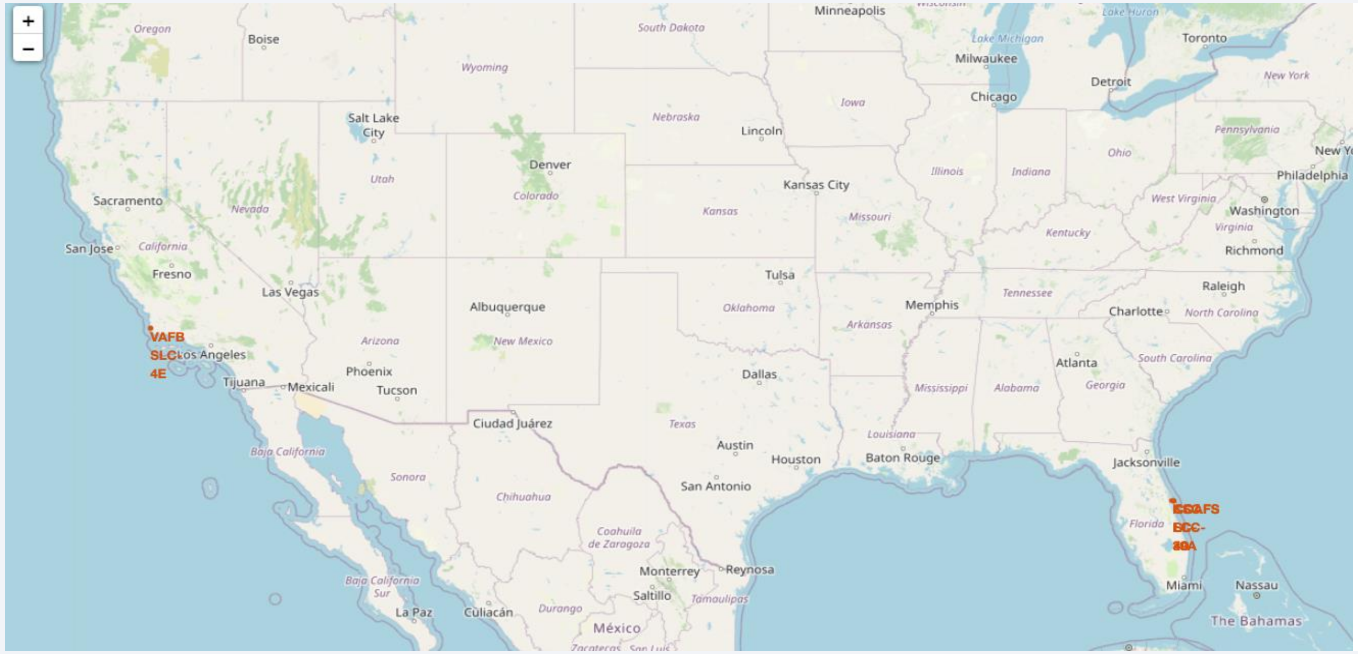
1
2020-12-06

Section 4

Launch Sites Proximities Analysis

Launch Sites from site maps

- Launch Sites from site maps



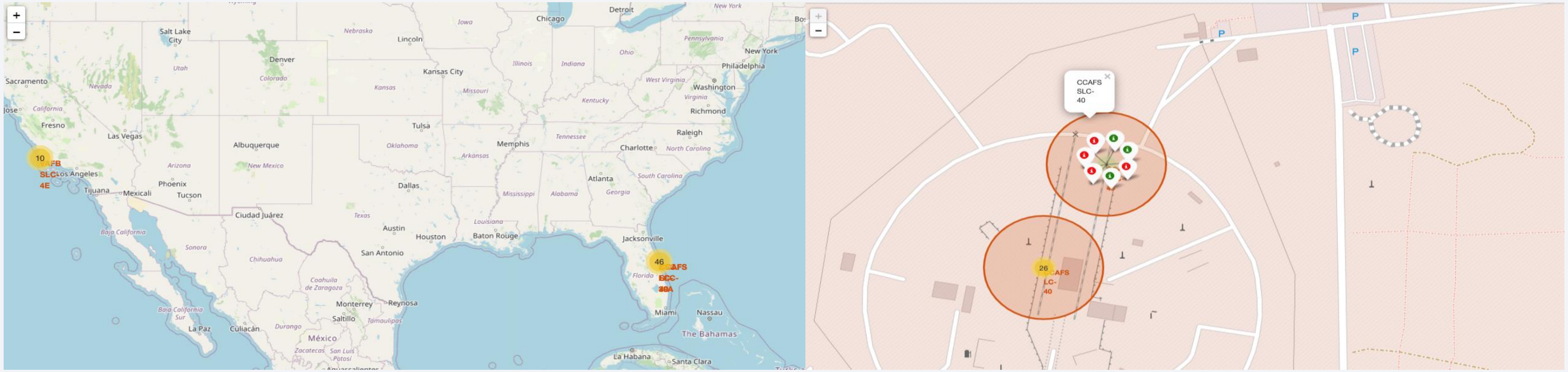

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

- This indicates that not all launch sites are on equator.
- Additionally, Launch sites are close to the coastal areas.

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

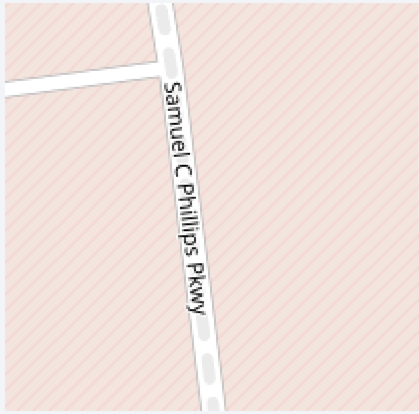
- Mark the success/failed launches for each site on the map

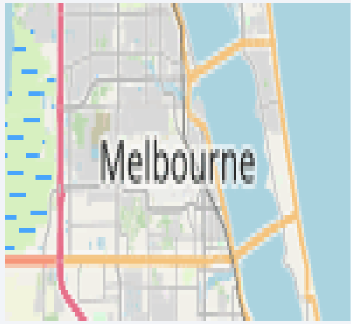


- We can clearly see that markers have specific colors, RED shows the unsuccessful launches and GREEN shows the successful launches on the site map.

Calculate the distances between a launch site to its proximities

- Calculate the distances between a launch site to its proximities:





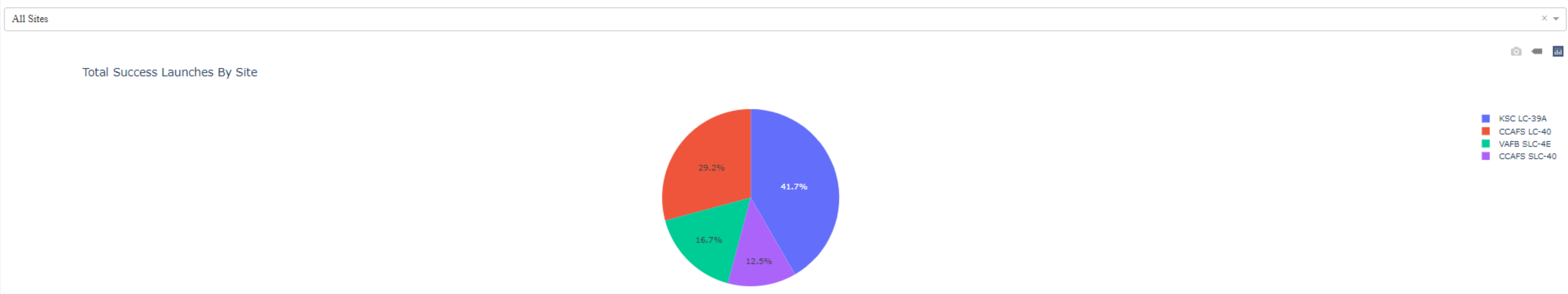
- First image is the distance from launch to coast
- Second image is A railway map symbol
- Third is A highway map symbol
- Last one is
- A city map symbol

Section 5

Build a Dashboard with Plotly Dash

Pie chart for all sites

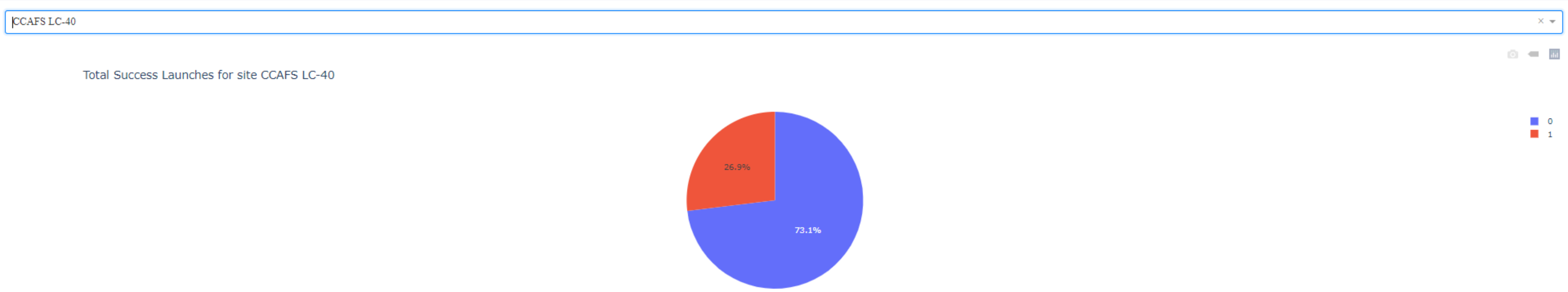
- Total success launch by site:



- It is easily noticeable that KSC LC-39A has the highest success rate of 41.7% while CCAF SLC-14 has the lowest success rate of 12.5%
- CCAFS LC-14 has 29.2% and VAFB SLC-40 has 16.7% success rate

Site with Highest success and Launch rate

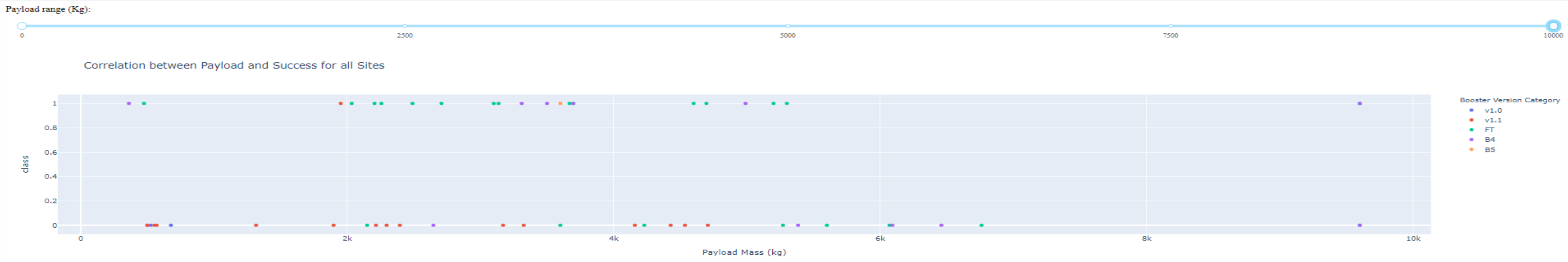
- Launch Site with highest launch and success rate:



- In terms of highest launch and success rate CCAFS LC-40 has the best average than the other launch sites.

Payload vs. Launch Outcome

- Payload vs. Launch Outcome scatter plot for all sites



- For different values of payload booster has different success rate but as we set payload to 1000kg the booster which has most success rate is FT.

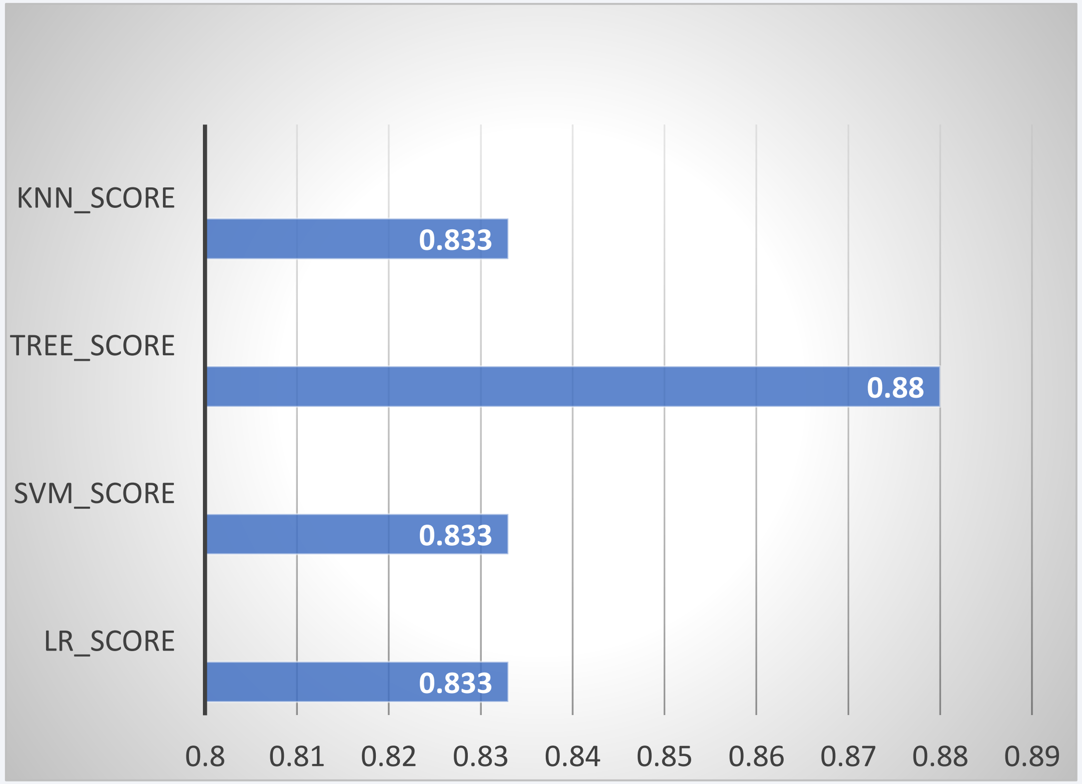
Section 6

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

Find the method performs best:

```
scores = [lr_score,svm_score,tree_score,knn_score]
print(scores)
print(scores.index(max(scores)))
```

```
[0.8333333333333334, 0.8333333333333334, 0.8888888888888888, 0.8333333333333334]
2
```

- Among all of the clsassifier tree_score has the highest accuracy of 88.88%

Conclusions

- In terms of highest launch and success rate CCAFS LC-40 has the best average than the other launch sites
- The highest success rate of booster was FT (at 1000kg payload)
- Models and their performances:
- Logistic Regression : 83.33%
- SVM : 83.33%
- Tree classifier: 88.88% (Best performance)
- KNN score: 83.33%

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!