



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Olga Yurlova
12/5/2022

https://github.com/OkaYurl/My_IBM_Final_Project



Outline

- Executive Summary.....3
- Introduction.....4
- Methodology.....5
- Results.....17
- Conclusion.....46
- Appendix.....47

Executive Summary

- The Winning Space Race with Data Science Project (DS Project) goal is to determine the cost of each launch by training a machine learning model based on the public information to predict if SpaceX will reuse the first stage. The DS Project collects Data with API and with web scraping related Wiki pages. Data wrangling includes: Wrangling Data using an API, Sampling Data, and Dealing with Nulls. Applied exploratory data analysis (EDA) (with visualization and SQL) and interactive visual analytics (with Folium and Plotly Dash) allows to understand data and find the reasonable Training Labels for the Machine Learning models. For prediction were used the following classifiers: Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors.
- The DS Project applied the mentioned above methodologies to the SpaceX dataset and got some results:
 - The launch success yearly trend demonstrates that the success rate since 2013 kept increasing till 2020. The plotted bar chart Success Rate vs. Orbit Type shows that orbits ES-L1, GEO, NEO and SSO have highest success rate.
 - Interactive analytics with Folium showed that launch site VAFB SLC-4E is in very close proximity to the coast of Pacific Ocean and CCAFS SLC-40 site has a relatively highest number of flights. Plotly Dash was used for building an application for users to perform interactive visual analytics on SpaceX launch data in real-time. Pie chart of the launch site VAFB SLC-4E demonstrated highest launch success ratio.
 - Predictive analysis results determined the same level of accuracy about 83% for all models. This is because the dataset is small and has a lesser values.

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone. Perhaps the most successful is [SpaceX](#). SpaceX's accomplishments include: sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space. One reason SpaceX can do this is the rocket [launches are relatively inexpensive](#). SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can [reuse the first stage](#). Therefore, if it can be determined if the first stage will land, it can be determined the cost of a launch.
- SpaceX's [Falcon 9](#) launch like regular rockets. Unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage. Sometimes the first stage does not land. Sometimes it will crash. Other times, SpaceX will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.
- [The Winning Space Race with Data Science Project \(DS Project\)](#) goal is to determine the cost of each launch. It is done by gathering information about SpaceX and creating dashboards for the project. It is also determined if SpaceX will reuse the first stage by training a machine learning model and by using public information to predict if SpaceX will reuse the first stage. Instead of using rocket science to predict if the first stage will land successfully, a [machine learning model will be trained](#) using public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary:

- The DS Project **collects Data with API**. The Data collection is started with the SpaceX REST API endpoints, or URL, api.spacexdata.com/v4/launches/past. Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages.
- **Data wrangling**: to transform the raw data into a clean dataset which provides meaningful data the following procedure is applied: Wrangling Data using an API, Sampling Data, and Dealing with Nulls.
- Performing **exploratory data analysis** (EDA) using **visualization** and **SQL**.
- Performing interactive visual analytics using **Folium** and **Plotly Dash**.
- Performing predictive analysis using the following classification models: **Logistic Regression**, **Support Vector machines**, **Decision Tree Classifier**, and **K-nearest neighbors**.

Data Collection

- SpaceX launch data is gathered from an API, specifically the [SpaceX REST API](#).
- The DS Project starts with the SpaceX REST API endpoints, or URL, `api.spacexdata.com/v4/launches/past`. This URL is used to target a specific endpoint of the API to get past launch data by request. The result can be viewed by calling the `.json()` [method](#).
- The response will be in the form of a JSON, specifically a list of JSON objects which each represent a launch. To convert this JSON to a dataframe, the [json_normalize function](#) is used.
- The [Python BeautifulSoup](#) package is used to web scrape some HTML tables that contain valuable Falcon 9 launch records.
- Then the data from those tables are parsed and converted into a [Pandas data frame](#) for further visualization and analysis.

Data Collection – SpaceX API

- SpaceX launch data is gathered from an API, specifically the SpaceX REST API. The DS Project starts with the SpaceX REST API endpoints, or URL, `api.spacexdata.com/v4/launches/past`.
- This API provides data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- As a response from API the DS Project will get a list of JSON objects which each represent a launch. To convert this JSON to a dataframe, it is used the `json_normalize` function. This function allows to “normalize” the structured JSON data into a flat table.

<https://github.com/OkaYurl/My IBM Final Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

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

```
response =  
requests.get(spacex_url)
```

```
data =  
pd.json_normalize(response.json())
```


Data Collection - Scraping

- Another data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages.
- The Python BeautifulSoup package is used to web scrape some HTML tables that contain valuable Falcon 9 launch records.
- The filtered launch data for the Falcon 9 have to be included only.

[https://github.com/OkaYurl/My IBM Final Project/blob/main/jupyter-labs-webscraping\(2\).ipynb](https://github.com/OkaYurl/My IBM Final Project/blob/main/jupyter-labs-webscraping(2).ipynb)

```
static_soup =  
BeautifulSoup(response.text,  
"html.parser")
```

```
launch_df =  
pd.DataFrame(launch_dict)
```

```
data_falcon9 =  
launch_df[launch_df['BoosterVers  
ion']!= 'Falcon 1']
```

Data Wrangling

- To transform the raw data into a clean dataset which provides meaningful data the following procedure is applied: [Wrangling Data using an API](#), [Sampling Data](#), and [Dealing with Nulls](#).
- In some of the columns, there is an identification number, not actual data. The API is used again targeting another endpoint to gather specific data for each ID number. These functions are used: [getBoosterVersion\(data\)](#), [getLaunchSite\(data\)](#), [getPayloadData\(data\)](#), and [getCoreData\(data\)](#).
- The data are stored in lists and are used to create the dataset. In order to make the dataset viable for analysis it is necessary to deal with [NULL values](#) (by replacing with the mean ([mean_value = data_falcon9\['PayloadMass'\].mean\(\)](#)) or using one hot encoding ([data_falcon9\['PayloadMass'\].fillna\(value=mean_value, inplace=True\)](#))):
- The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 VAFB SLC 4E , Vandenberg Air Force Base Space Launch Complex 4E (SLC-4E), Kennedy Space Center Launch Complex 39A KSC LC 39A. The location of each Launch is placed in the column LaunchSite.
- The launch outcomes were converted into the [column 'Class'](#) with 1 means the booster successfully landed 0 means it was unsuccessful. The average success rate of Falcon 9 launch is about 67%.

EDA

- **Exploratory Data Analysis (EDA)** is the **first step of any data science project** to find some patterns in the data and determine what would be the label for training supervised models.
- 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. During the Data Wrangling stage those outcomes were converted into **Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful**.
- The **categorical variables** are converted using **one hot encoding**, preparing the data for a machine learning model that predicts if the first stage will successfully land.

EDA with Data Visualization

The DS Project performs EDA with Data visualization by plotting the charts showing the relationships between different data like the following:

- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Success Rate vs. Orbit Type
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type
- Launch Success Yearly Trend

[https://github.com/OkaYurl/My IBM Final Project/blob/main/jupyter-labs-eda-dataviz\(2\)\(3\)\(2\).ipynb](https://github.com/OkaYurl/My IBM Final Project/blob/main/jupyter-labs-eda-dataviz(2)(3)(2).ipynb)

EDA with SQL

The performed SQL queries:

- 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)**.
- 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**.
- 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 **successful landing_outcomes** between the **date 04-06-2010 and 20-03-2017** in descending order.

Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, as well as 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 also involves many factors by analyzing the existing launch site locations with more interactive visual analytics using [Folium](#).
- Some geographical patterns about launch sites were created and added to the Folium Map:
 - [all launch sites](#) on a map;
 - the [success/failed launches](#) for each site on the map;
 - the [distances](#) between a launch site to its proximities.

[https://github.com/OkaYurl/My IBM Final Project/blob/main/lab_jupyter_launch_site_location_done\(2\).ipynb](https://github.com/OkaYurl/My IBM Final Project/blob/main/lab_jupyter_launch_site_location_done(2).ipynb)

Build a Dashboard with Plotly Dash

- Interactive visual analytics like [Dashboard](#) usually are built for stakeholders. Interactive visual analytics enables users to explore and manipulate data in an interactive and real-time way. Common interactions including filter, search, and link. With interactive visual analytics, users could find visual patterns faster and more effectively.
- The DS Project performed interactive visual analytics on SpaceX launch data in real-time. This dashboard application contains input components such as a [dropdown list](#) (to select one specific site) and a [range slider to interact with a pie chart](#) (visualizing launch success counts for all or with different boosters) and a [scatter point chart](#) (observe how payload may be correlated with mission outcomes for selected site(s) for all or with different boosters).

https://github.com/OkaYurl/My IBM Final Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- The DS Projects performs **exploratory Data Analysis** and determines Training Labels by following steps:
 - create a **column for the class**
 - **standardize** the data
 - split into **training data and test data**
- The model was trained and was performed with **Grid Search**, allowing to find the hyperparameters that allows a given algorithm to perform best. **Logistic Regression**, **Support Vector Machines**, **Decision Tree Classifier**, and **K-nearest neighbors** models were tested. The **confusion matrixes** were built. Using the best hyperparameter values, the model with the best **accuracy** was determined using the training data.

[https://github.com/OkaYurl/My IBM Final Project/blob/main/SpaceX Machine%20Learning%20Prediction Part 5 done\(2\).ipynb](https://github.com/OkaYurl/My IBM Final Project/blob/main/SpaceX Machine%20Learning%20Prediction Part 5 done(2).ipynb)

Results

- The DS Project performs EDA with Data visualization by plotting the charts showing the relationships between different data. The scatter plot [Flight Number vs. Launch Site](#) shows that [CCAFS SLC-40](#) site has a relatively [highest number of flights](#). The plotted bar chart [Success Rate vs. Orbit Type](#) shows that orbits [ES-L1](#), [GEO](#), [NEO](#) and [SSO](#) have highest success rate. The [launch success yearly trend](#) demonstrates that the success rate since 2013 kept [increasing](#) till 2020 with one not really successful 2018.
- Interactive analytics with [Folium](#) allowed to generate a map with each site's location using site's latitude and longitude coordinates. The map can show, for example, that launch site [VAFB SLC-4E](#) is in very [close proximity to the coast of Pacific Ocean](#); From the color-labeled markers in marker clusters, it should be able to easily identify which launch sites have relatively [high success rates](#) (like [KSC LC-39A](#)).
- [Plotly Dash](#) was used for building an application for users to perform interactive visual analytics on SpaceX launch data in real-time. [Pie chart](#) for the launch site VAFB SLC-4E with highest launch success ratio was selected. The [Booster version](#) on each scatter point on the [success-payload-scatter-chart](#) were color-labeled, so that it is possible to observe mission outcomes with different boosters.
- [Predictive analysis](#) used the following classification models: Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Predictive analysis results demonstrates the same level of [accuracy about 83% for all models](#). This is because [the dataset is small and has a lesser values](#).

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is dynamic and technological.

Section 2

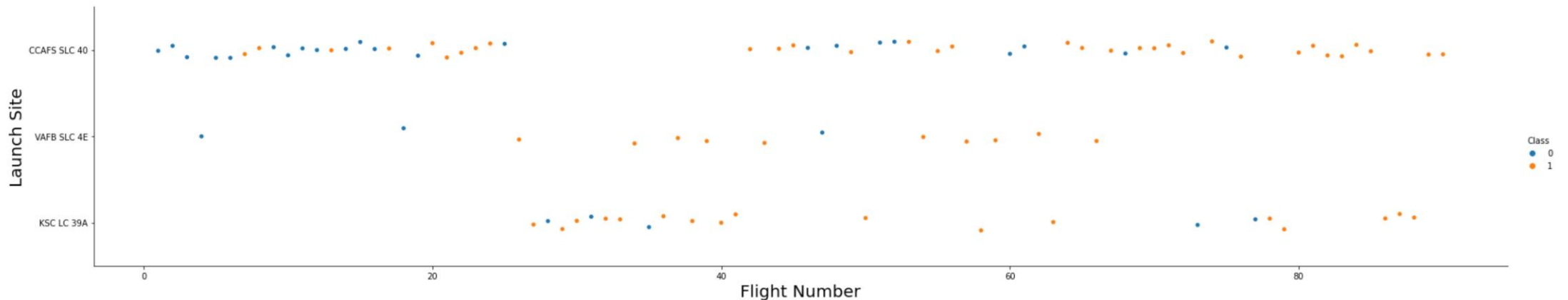
Insights drawn from EDA

Flight Number vs. Launch Site

TASK 1: Visualize the relationship between Flight Number and Launch Site

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



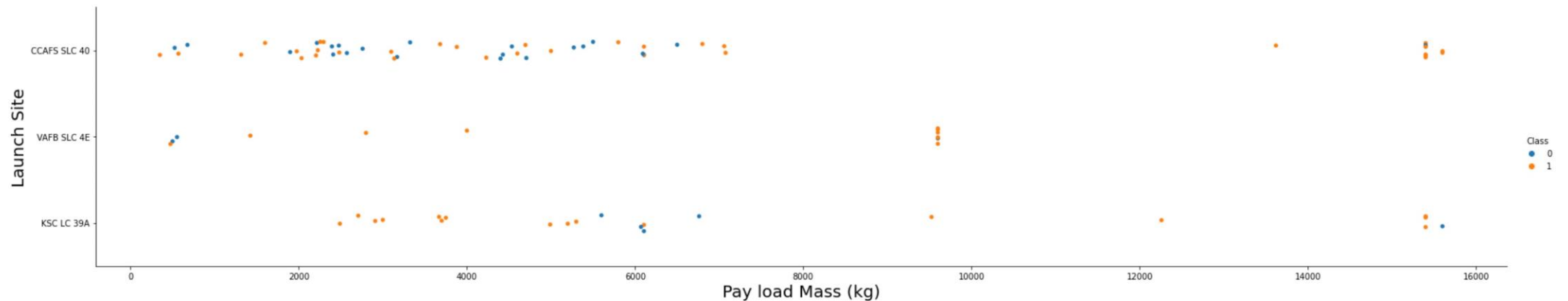
The scatter plot Flight Number vs. Launch Site shows that the different launch sites have different number of flights. **CCAFS SLC-40** site has a relatively **highest** number of flights.

Payload vs. Launch Site

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

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



The scatter plot of Payload vs. Launch Site can demonstrate that for the **VAFB-SLC 4E** launch site there are **no rockets** launched for heavy payload mass (**greater than 10000 kg**).

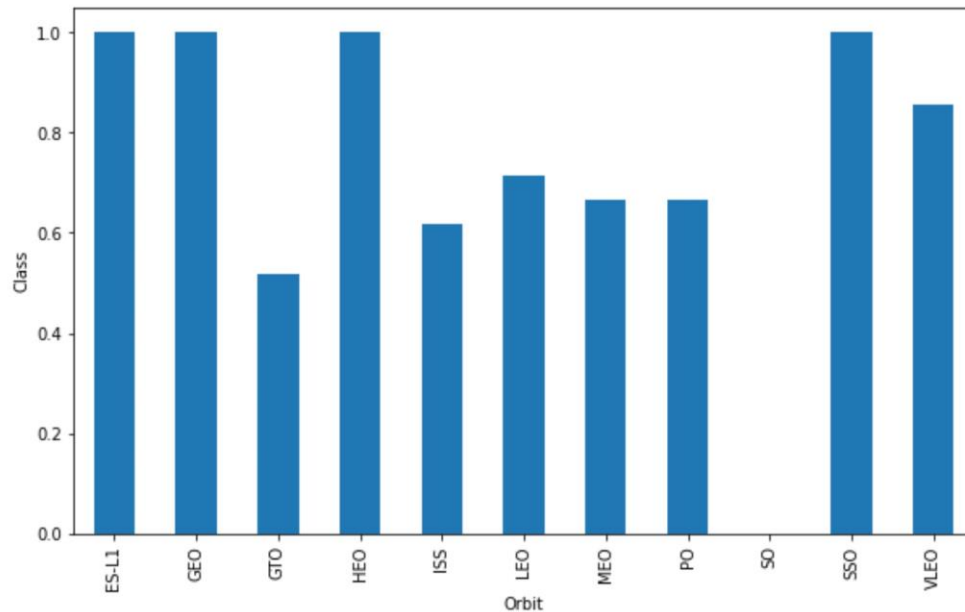
Success Rate vs. Orbit Type

TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar` chart for the success rate of each orbit

```
# HINT use groupby method on Orbit column and get the mean of Class column
df.groupby('Orbit')['Class'].mean().plot(kind='bar', figsize=(10, 6))
plt.xlabel('Orbit')
plt.ylabel('Class')
plt.show()
```



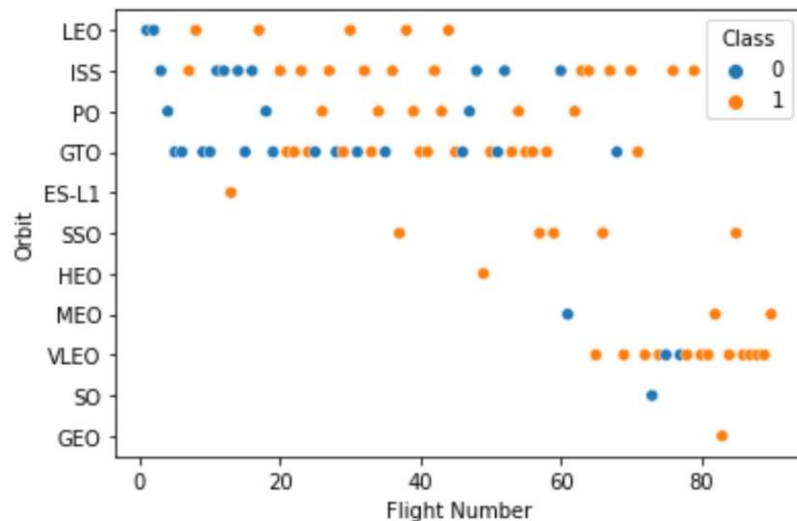
The plotted bar chart for the success rate of each orbit shows that orbits **ES-L1**, **GEO**, **NEO** and **SSO** have **highest success rate**.

Flight Number vs. Orbit Type

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and 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.scatterplot(data=df, x="FlightNumber", y="Orbit", hue="Class")
plt.xlabel("Flight Number")
plt.ylabel("Orbit")
plt.show()
```



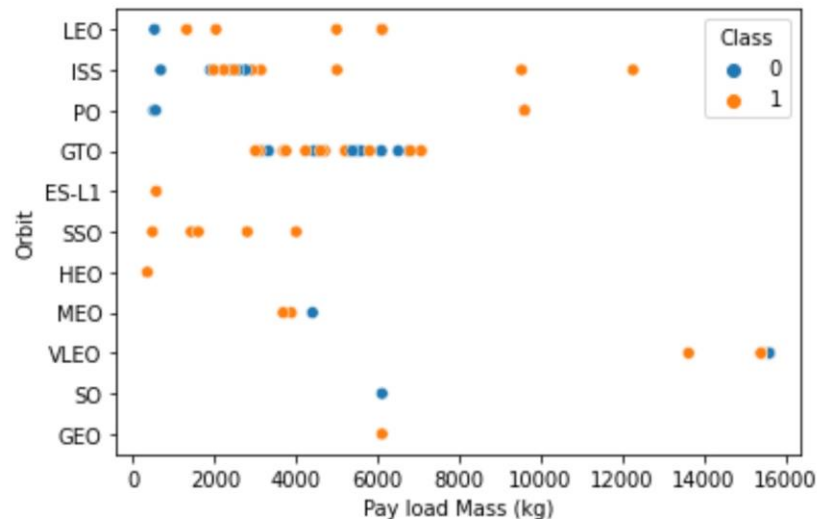
The scatter plot of Flight Number vs. Orbit Type can demonstrate that the **LEO** orbit the Success **appears related** to the number of flights; on the other hand, there seems to be **no relationship** between flight number when in **ISS** and **GTO** orbit.

Payload vs. Orbit Type

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and 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.scatterplot(data=df, x="PayloadMass", y="Orbit", hue="Class")
plt.xlabel("Pay load Mass (kg)")
plt.ylabel("Orbit")
plt.show()
```



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

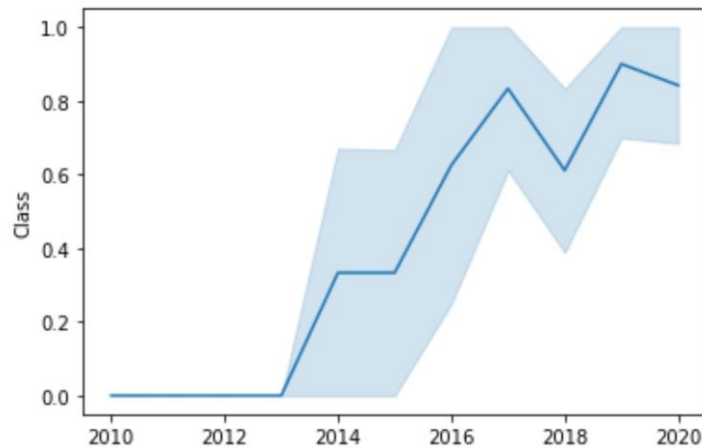
Launch Success Yearly Trend

TASK 6: Visualize the launch success yearly trend

You can plot a line chart with x axis to be `Year` and y axis to be average success rate, to get the average launch success trend.

The function will help you get the year from the date:

```
# Plot a Line chart with x axis to be the extracted year and y axis to be the success rate
df['Year'] = pd.DatetimeIndex(df['Date']).year
sns.lineplot(data=df, x="Year", y="Class")
plt.ylabel('Class')
plt.xlabel('Year')
plt.show()
```



The launch success yearly trend demonstrates that the success rate since 2013 kept increasing till 2020 with one not really successful 2018.

All Launch Site Names

SQL queries to display the names of the unique launch sites in the space mission:

```
%sql select distinct(Launch_Site) from SPACEXTBL
```

```
Launch_Site
```

```
CCAFS LC-40
```

```
VAFB SLC-4E
```

```
KSC LC-39A
```

```
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

SQL queries to display 5 records where launch sites begin with the string 'CCA':

```
%sql select * from SPACEXTBL where Launch_Site like('CCA%') limit 5
```

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
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

SQL queries to calculate the total payload mass carried by boosters launched by NASA (CRS):

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like('NASA (CRS)')
```

```
SUM(PAYLOAD_MASS__KG_)
```

```
45596
```

Average Payload Mass by F9 v1.1

SQL queries to calculate the average payload mass carried by booster version F9 v1.1:

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version  
like('F9 v1.1%')
```

```
AVG(PAYLOAD_MASS__KG_)
```

```
2534.6666666666665
```

First Successful Ground Landing Date

SQL queries to list the date when the first successful landing outcome in ground pad was achieved:

```
%%sql select min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) as MinDate  
from SPACEXTBL where "Landing _Outcome" like ('%Success (ground pad)%')
```

MinDate

20151222

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL queries to 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_ BETWEEN 4000 AND 6000
```

Booster_Version

F9 FT B1022

F9 FT B1026

F9 FT B1021.2

F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

SQL queries to list the total number of successful and failure mission outcomes:

```
%sql select COUNT(Mission_Outcome) AS Success from SPACEXTBL where  
Mission_Outcome like('%Success%')
```

Success

100

```
%sql select COUNT(Mission_Outcome) AS Failure from SPACEXTBL where  
Mission_Outcome not like('%Success%')
```

Failure

1

Boosters Carried Maximum Payload

SQL queries to list the names of the booster_versions which have carried the maximum payload mass:

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=(select  
max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

SQL queries to 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:

```
%%sql select substr(Date, 4, 2) as "Month", Booster_Version, Launch_Site  
from SPACEXTBL where "Landing _Outcome"='Failure (drone ship)' and  
substr(Date,7,4)='2015'
```

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

SQL queries to rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.:

```
%%sql select "Date", "Landing _Outcome", count ("Landing _Outcome")  
from SPACEXTBL where "Date" between '04-06-2010' and '20-03-2017'  
and "Landing _Outcome" like 'Success%' group by "Landing _Outcome" order by count  
("Landing _Outcome") desc
```

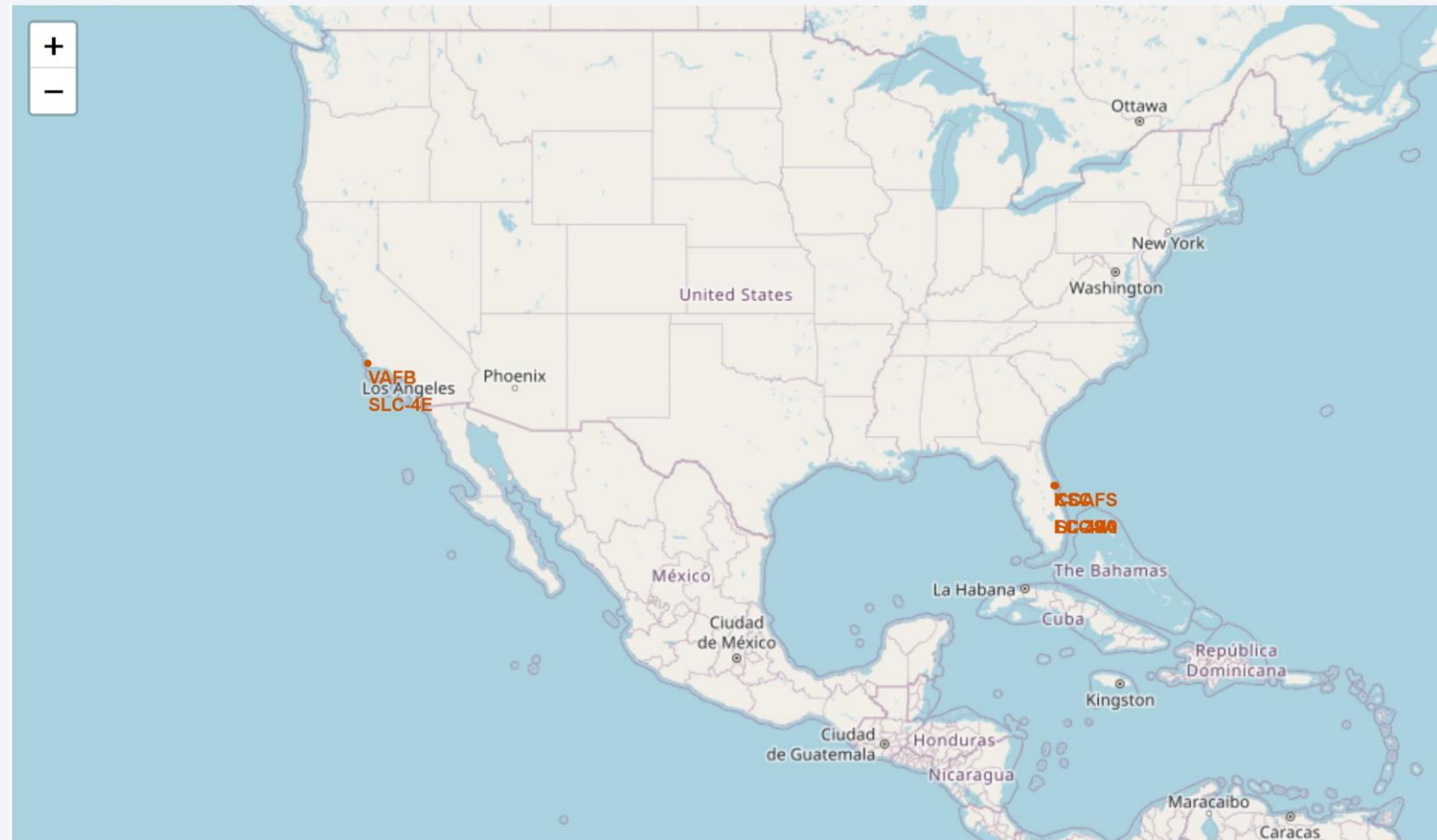
Date	Landing _Outcome	count ("Landing _Outcome")
07-08-2018	Success	20
08-04-2016	Success (drone ship)	8
18-07-2016	Success (ground pad)	6

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

Launch Sites Proximities Analysis

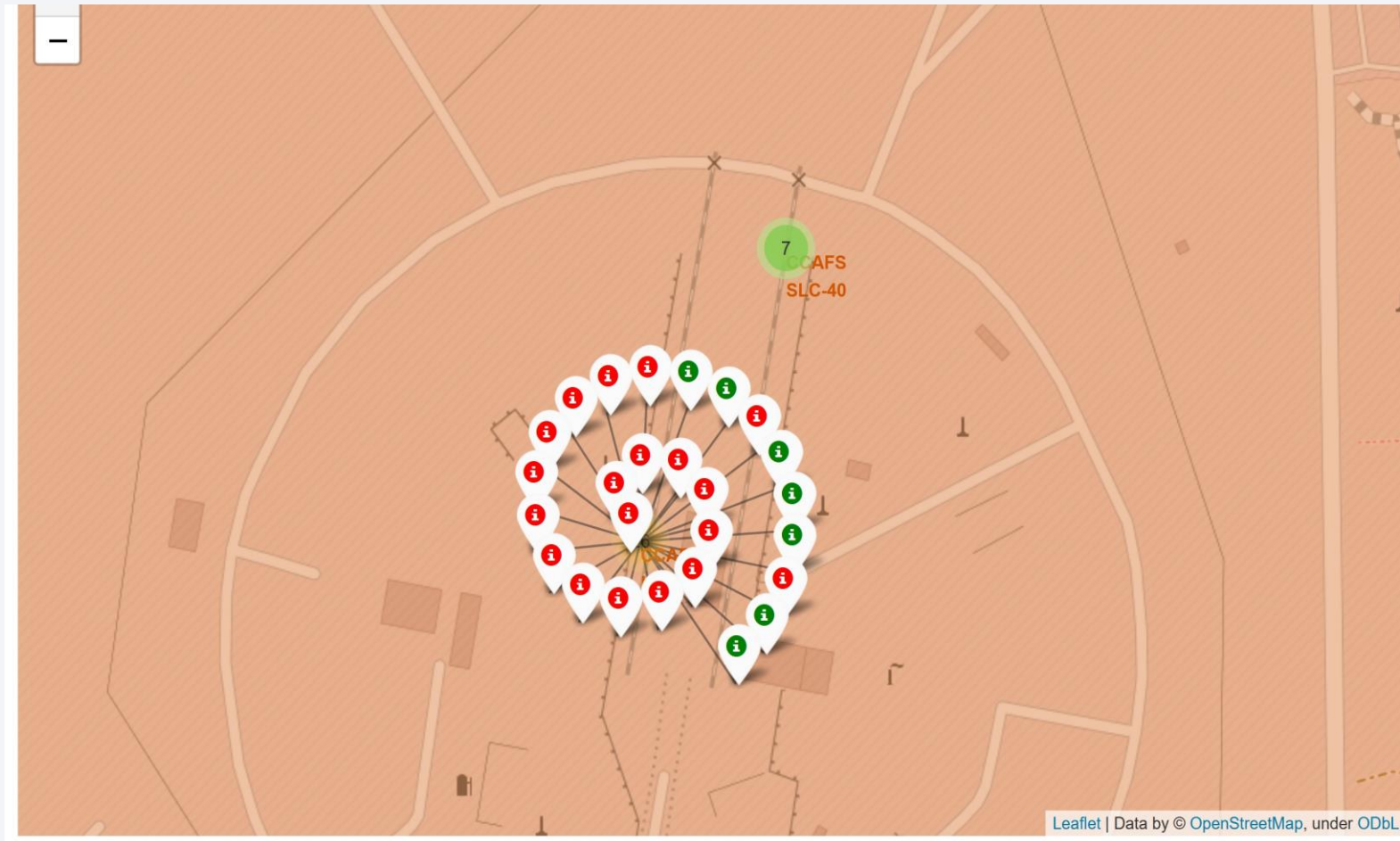
All launch sites on a map



- Folium was used to add each site's location on a map using site's latitude and longitude coordinates (find below). Folium includes functions to create and add `folium.Circle` and `folium.Marker` for each launch site on the site map.
- The generated map with marked launch sites allows to explore the map by zoom-in/out the marked areas: and to see, for example, that launch site VAFB SLC-4E is in very close proximity to the coast of Pacific Ocean.

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	-120.610745

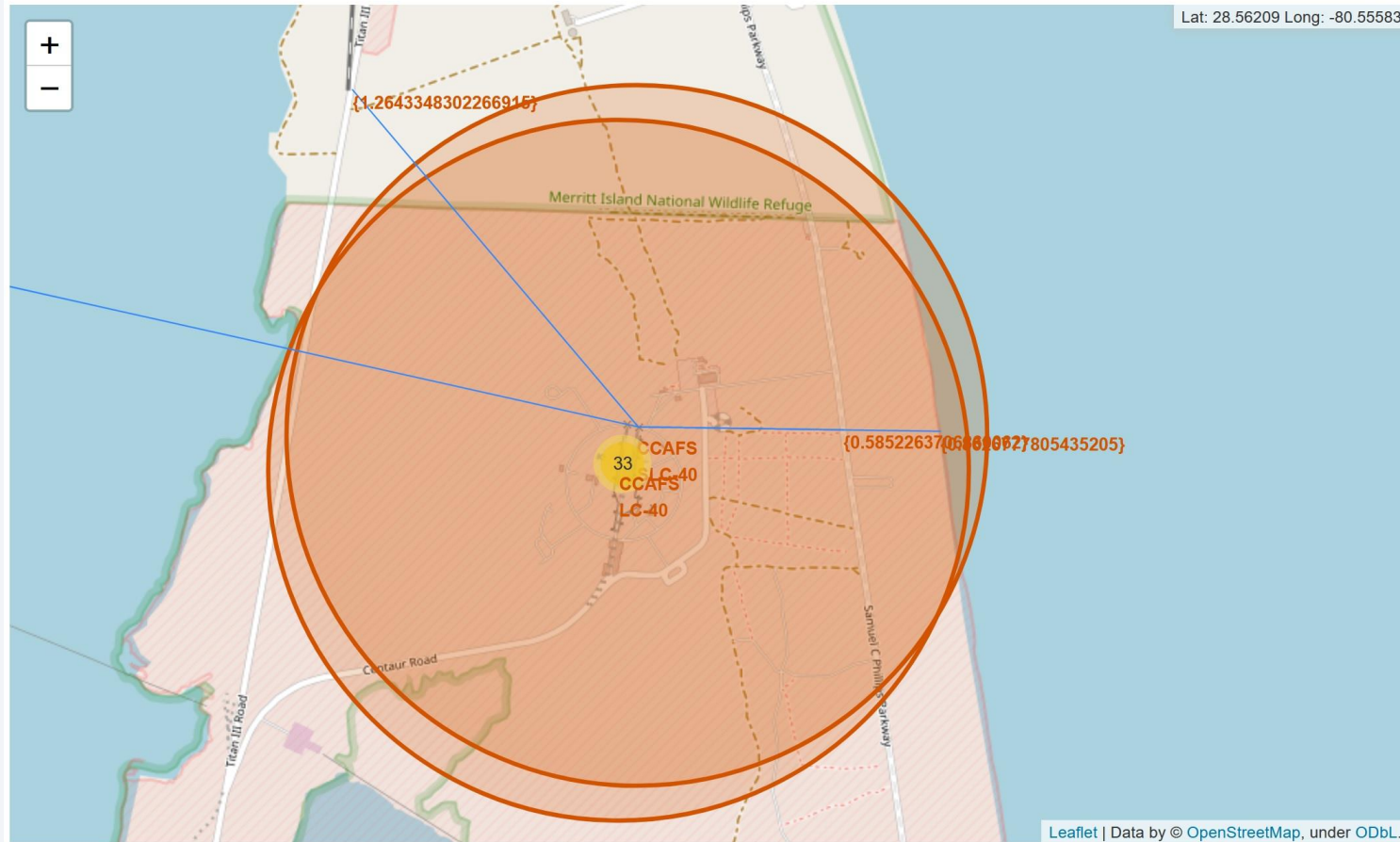
The success/failed launches for each site on the map



- **Marker clusters** was used to simplify a map containing many markers having the same coordinate and to show the color-labeled launch outcomes on the map.
- If a launch was successful (class=1), then was used a green marker and if a launch was failed, was used a red marker (class=0)
- From the color-labeled markers in marker clusters, it should be able to easily identify which launch sites have relatively **high success rates** (like **KSC LC-39A**).

The distances between a launch site to its proximities

distance_city_highway (km): 0.5852263706860062



- **Folium** package has the several instruments using to explore and analyze the proximities of launch sites:
- **MousePosition** allows to get coordinate for a mouse over a point on the map based on their Lat and Long values;
- After obtained its coordinate, a **folium.Marker** was created to show the distance;
- **Folium.PolyLine** draws a line between a launch site to its closest city, railway, highway, etc.
- For example, the proximities of launch site **CCAFS SLC-40** to its closest city Titusville, railway and highway.

The proximities of launch site CCAFS SLC-40 from the following points:

distance_city_Titusville (km): 23.262712044907804

distance_railway (km): 1.2643348302266915

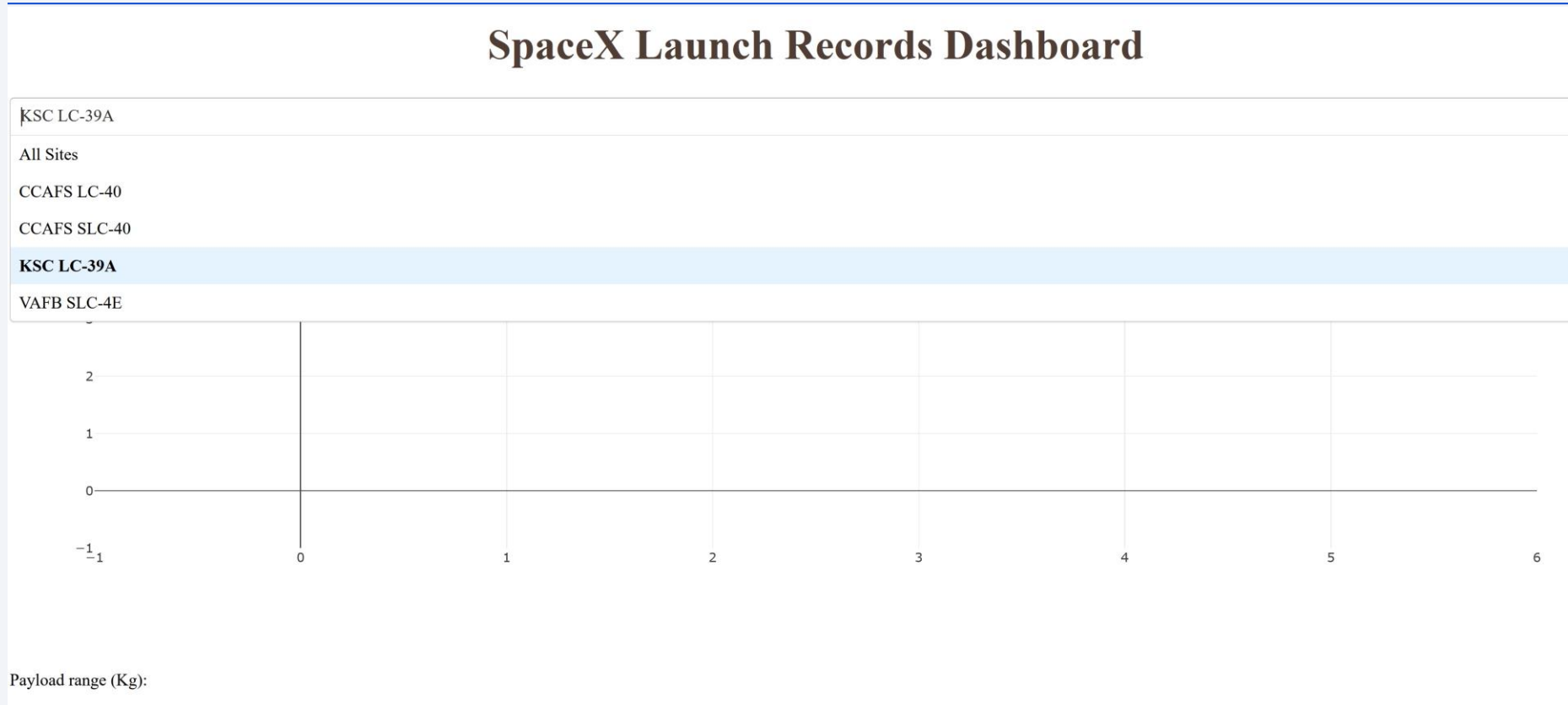
distance_city_highway (km): 0.5852263706860062



Section 4

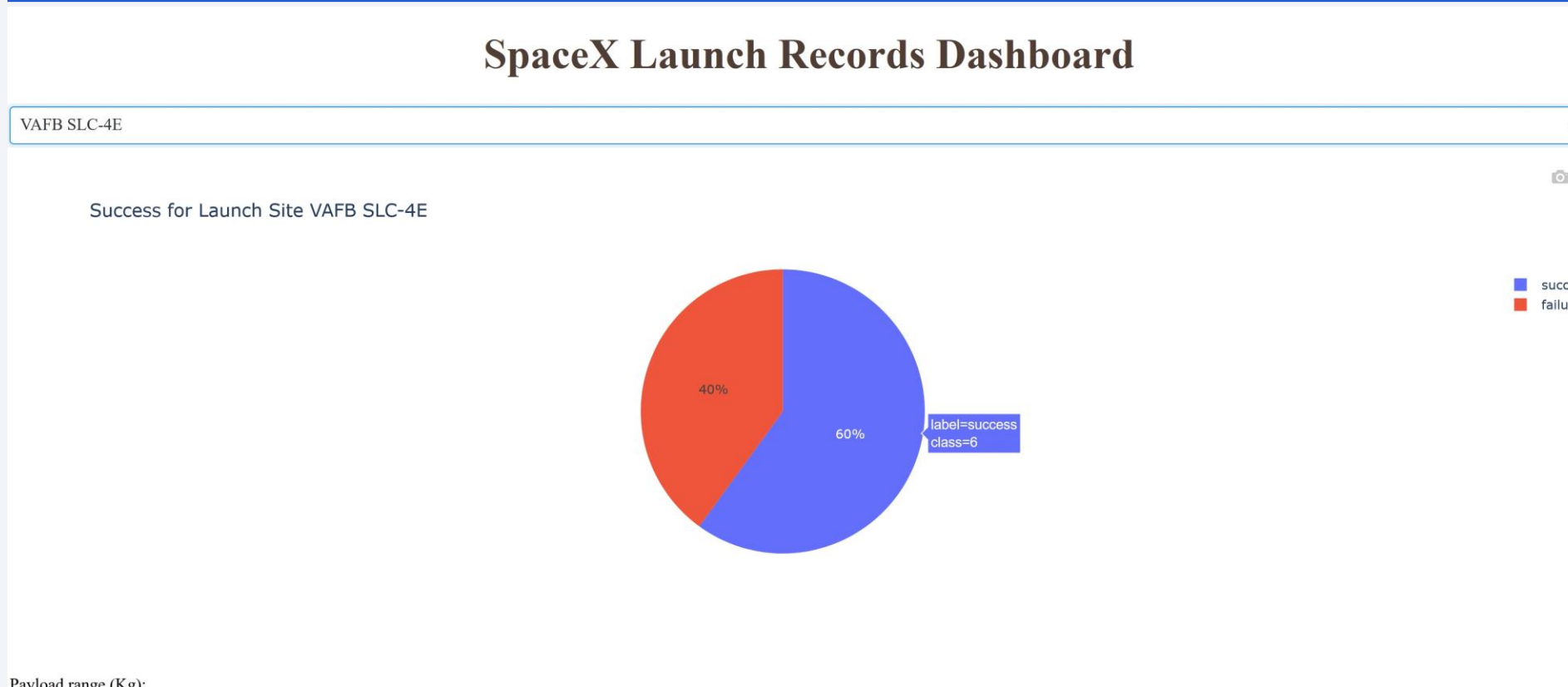
Build a Dashboard with Plotly Dash

Launch Site Drop-down Input Component



- [Plotly Dash](#) was used for building an application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- There are four different launch sites and the completed dropdown menu allows to select one specific site and check its detailed success rate (class=0 vs. class=1) and see which one has the largest success count.

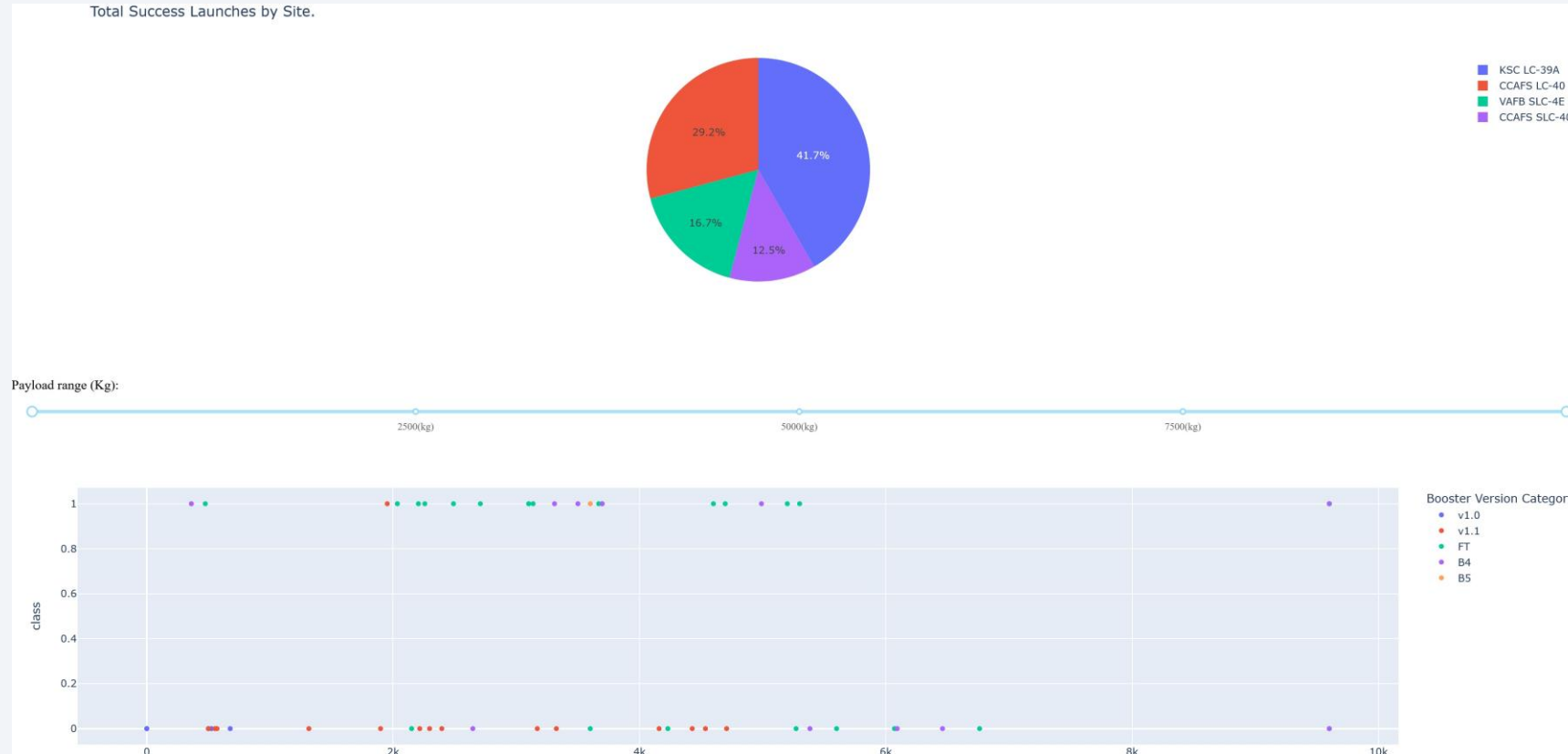
The pie-chart for the launch site with highest launch success ratio



Payload range (Kg):

- **Dash callback function** is a type of Python function which is automatically called by Dash whenever receiving an input component updates, such as a click or dropdown selecting event.
- The callback function is used to get the selected launch site from site-dropdown and render a pie chart visualizing launch success counts.
- Pie chart for the launch site VAFB SLC-4E with highest launch success ratio was selected.

Payload vs. Launch Outcome scatter plot for all sites

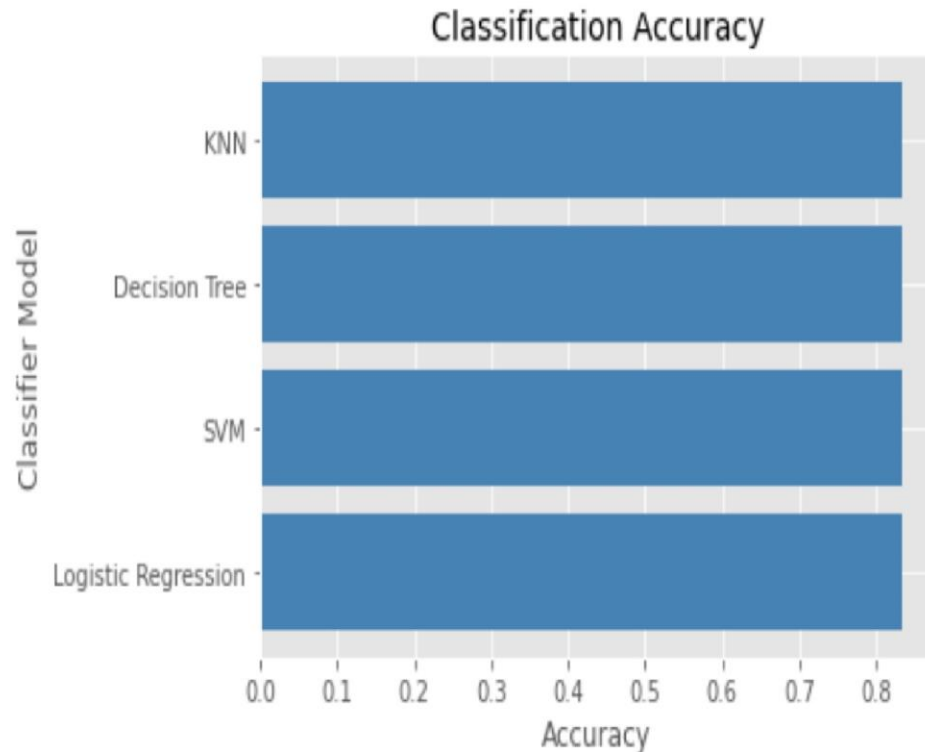


- A callback function renders [the success-payload-scatter-chart](#) scatter to plot a scatter plot with the x axis to be the payload and the y axis to be the launch outcome (i.e., class column). As such, it allows visually observe how payload may be correlated with mission outcomes for selected site(s). The Booster version on each scatter point were color-labeled, so that it is possible to observe mission outcomes with different boosters.
- Above a screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider.

Section 5

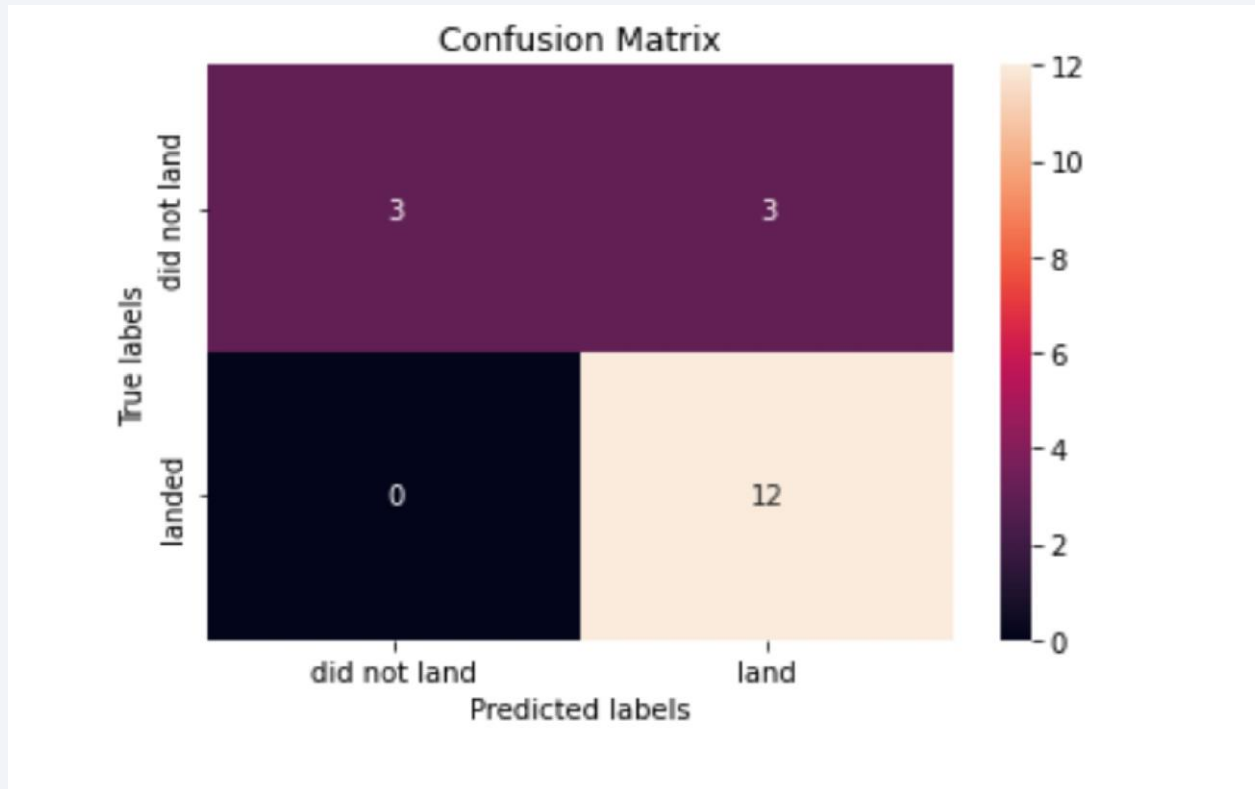
Predictive Analysis (Classification)

Classification Accuracy



- The DS Project created a machine learning pipeline to predict if the first stage will land given the prepared data.
- Predictive analysis used the following classification models: [Logistic Regression](#), [Support Vector machines](#), [Decision Tree Classifier](#), and [K-nearest neighbors](#).
- The results of the score are practically the same: 0.8333, and the confusion matrix looks [the same for all classifier models](#). This is because the [dataset is small and has lesser values](#).

Confusion Matrix



- The confusion matrix of the Logistic Regression Model shows that logistic regression can distinguish between the different classes – the classifier could predict 12 successful landed cases but the major problem is false positives.

Conclusions

- “The Winning Space Race with Data Science Project” (DS Project) goal was to determine the cost of each launch by training a machine learning model based on the public information to predict if SpaceX will reuse the first stage.
- The DS Project collected Data with API and with web scraping related Wiki pages and performed EDA with Data visualization by plotting the charts showing the relationships between different data. It showed that the launch success yearly trend demonstrates that the success rate since 2013 kept increasing till 2020.
- The interactive analytic with Folium could defined that the launch site KSC LC-39A has relatively high success rate. The built with Plotly Dashboard application helped users to perform interactive visual analytics on SpaceX launch data in real-time.
- Predictive analysis results demonstrates the same level of accuracy about 83% for all applied classifier models. This is because the dataset is small and has a lesser values.
- “The Winning Space Race with Data Science Project” could prove the positive perspective of using the Falcon 9 Rocket series whose launch progress can prove a reasonable cost. The results of the DS Project can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Appendix

- Data Collection Result

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004

Appendix

- Data Wrangling Result

er	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

We can use the following line of code to determine the success rate:

```
df["Class"].mean()
```

```
0.6666666666666666
```

Thank you!

