IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shuang Tian
Jan. 6 2023

Outline

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

Executive Summary

- Summary of methodologies
- SpaceX API collected data
- Data wrangling filter Falcon 9
- SQL Database query data
- Folium spatial data visualizing
- Seaborn EDA data
- 4 ways machine learning predict
- Summary of all results
- collected Falcon 9 90 observations
- EDA Falcon 9 dataset. Explored the relationship class with launch site, payload, orbit, booster version. Booster catalog.
- Machine learning analysis with KNN, SVM, Decision Tree, Logistic Regression,
- The Decision Tree method have highest accuracy 0.833~0.944.
- Use Decision Tree model to predict the launch result. Base on the Site, payload mass, orbit etc.

Introduction

- Project background and context
- SpaceX has gained worldwide attention for a series of historic milestones.
- It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas 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 can determine the cost of a launch.
- This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- This dataset includes a record for each payload carried during a SpaceX mission into outer space.

- Problems you want to find answers
- Is the observations of Falcon 9 a little bit low? Only 90 observation.
- How to evaluate the accuracy of the model is better? For this case, use different method we can get different result, even to the Final decision tree. Using Grid_search is 0.833, but using F1_score or Metrics is 0.944.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - In this case we use spacex api to collect the dataset.
  - Request and parse the SpaceX launch data using the GET request
- Perform data wrangling
  - Using mean fill na of payloadmass,
  - Convert landing pad successful outcome into Training label 1, others with 0.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built 4 different model, KNN, SVM, Decision Tree, LR. Use Accuracy metrics to evaluate classification models, the Decision Tree is highest accuracy.

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

1. spacex_url=https://api.spacexdata.com/v4/launches/past
2. data = pd.read_json(static_json_url)
3. Set subset and seletct the featuers we only need.
4. data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
5. Use series function to obtain the variables we need.

   launch site, payload mass, date, longitude, latitude, orbit, booster version, etc.

6. From Booster version filter only Falcon 9 version

   data_falcon9=data.loc[data['BoosterVersion']=='Falcon 9']

7 the data_faclon9 shape is 90x7

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 (must include completed code cell and outcome cell), as an external reference and peer-review purpose

1 Request and parse the SpaceX launch data using the GET request¶

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

response = requests.get(spacex_url)
```

2 Filter the dataframe to only include Falcon 9 launches

```
data_falcon9=data.loc[data['BoosterVersion']=='Falcon 9']
```

3 GitHub URL of Space link

Data Collection - Scraping

- Present web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

- a

1. spacex_url=https://api.spacexdata.com/v4/launches/past
2. `data = requests.get(url).text`
3. `soup = BeautifulSoup(data,"html.parser")`

4 GitHub URL of Space link

Data Wrangling

1. Describe how data were processed
2. Used pandas read url to obtain datafram. Used value_count to find out the landing pad data, caculate the launch site, orbit, mission outcome, landing outcome, create the failure and success with 0 and 1 as class variable to classify the Landing outcome to prepare the machine learn data.
3. You need to present your data wrangling process using key phrases and flowcharts
4. df['LaunchSite'].value_counts()
5. df['Orbit'].value_counts()
6. landing_outcomes=df['Outcome'].value_counts()
7. df['Class']=landing_class

- Add the GitHub URL of your completed data wrangling related notebooks,

EDA with Data Visualization

- **Summarize what charts were plotted and why you used those charts**

1. plot out 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. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
2. plot out the FlightNumber vs. launch site, figure out different launch site has different launch success rates.
3. plot out the payload mass vs. launch site, We also want to observe if there is any relationship between launch sites and their payload mass.
4. create a bar chart for the sucess rate of each orbit, we can observe the different orbit with different success rates.
5. plot out the FlightNumber vs. orbit, we can see that in the LEO orbit the Success appears related to the number of flights; no relationship between flight number when in GTO orbit.
6. 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.

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

1. Understand the Spacex DataSet
2. Load the dataset into the corresponding table in a slqlite database
3. Execute SQL queries to answer assignment questions
4. Query the site, payload, minimum, maximum payload mass, booster version, mission outcome. Launch outcome.

- Add the GitHub URL of your completed EDA with SQL notebook,

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map

1. Mark all launch sites on a map
2. Mark the success/failed launches for each site on the map
3. Calculate the distances between a launch site to its proximities

- Explain why you added those objects

- visualize those locations by pinning them on a map.
- intuitive insights about where are those launch sites
- enhance the map by adding the launch outcomes for each site, and see which sites have high success rates
- mark down their coordinates (shown on the top-left) in order to the distance to the launch site.

- The GitHub URL of your completed interactive map with Folium map

Build a Dashboard with Plotly Dash

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

1. Dropdown list for 4 launch site,
2. Payload mass slider bar
3. Callback function

- Explain why you added those plots and interactions
- With the 4 launch sites and payload as parameter to show the launch success rates.
- Interactive dashboard can display what we want to know by selecting launch site and payload mass.
- the GitHub URL of your completed Plotly Dash lab

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

1. create a column for the class Standardize the data Split into training data and test data -Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
2. eate a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
3. Using the GridSearchCV object with the best parameter to build the predict model.
4. You need present your model development process using key phrases and flowchart

```
# define hyperparameters to tune
parameters_tree = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

# define the model
tree = DecisionTreeClassifier(random_state = 12345)

# define the grid search object
grid_search_tree = GridSearchCV(
    estimator = tree,
    param_grid = parameters_tree,
    scoring = 'accuracy',
    cv = 10
)
# execute search
tree_model = grid_search_tree.fit(X_train, Y_train)
```

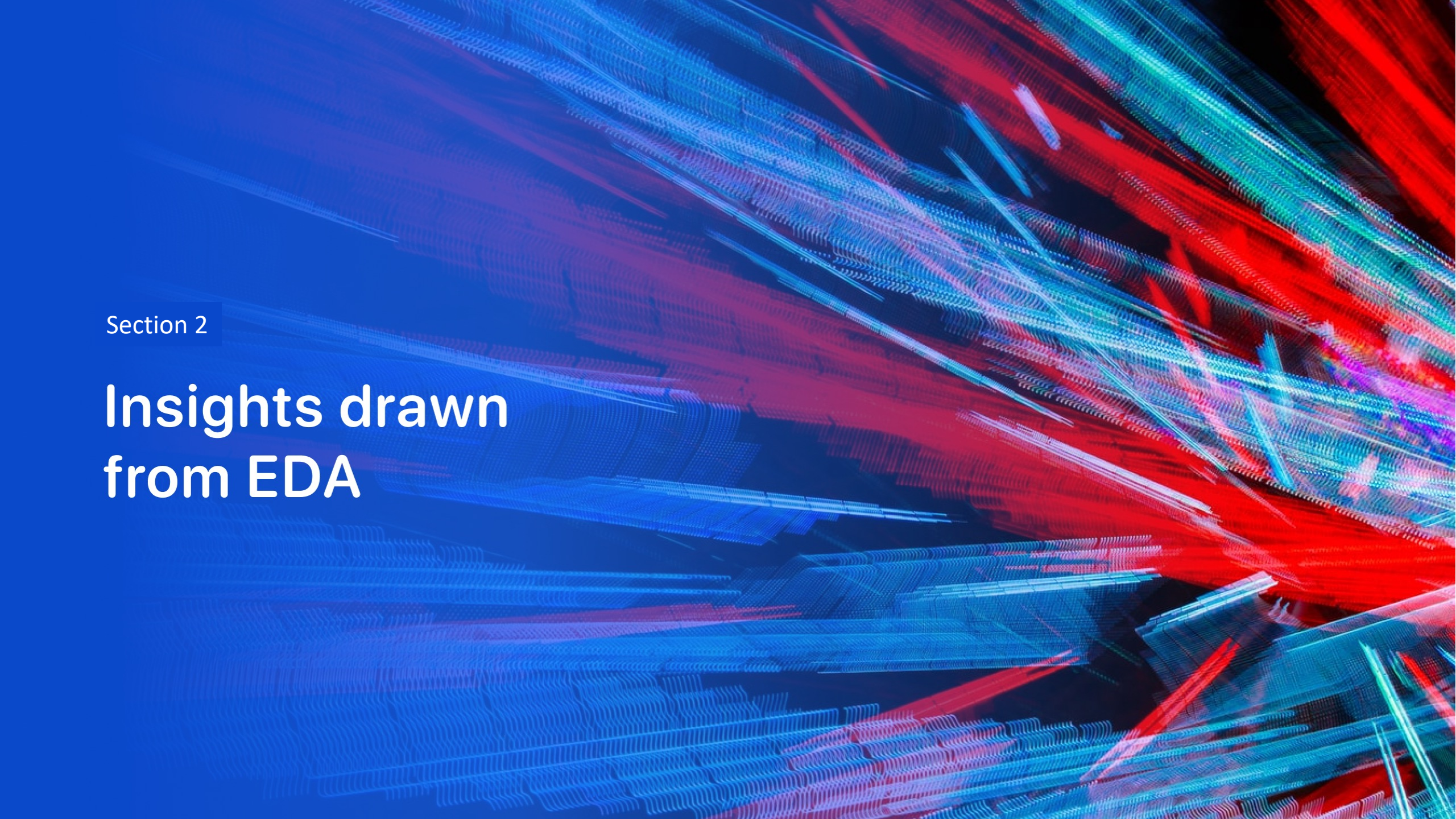
```
print("tuned hyperparameters :(best parameters) ",tree_model.best_params_)
print("accuracy :",tree_model.best_score_)
```

```
tuned hyperparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_
samples_split': 5, 'splitter': 'random'}
accuracy : 0.873214285714285 6
```

- the GitHub URL of your completed predictive analysis lab

Results

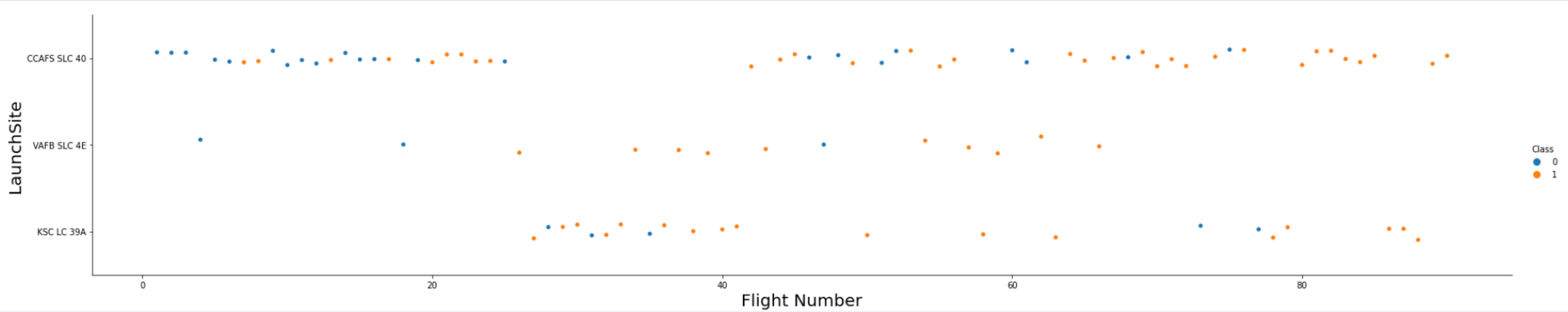
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

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

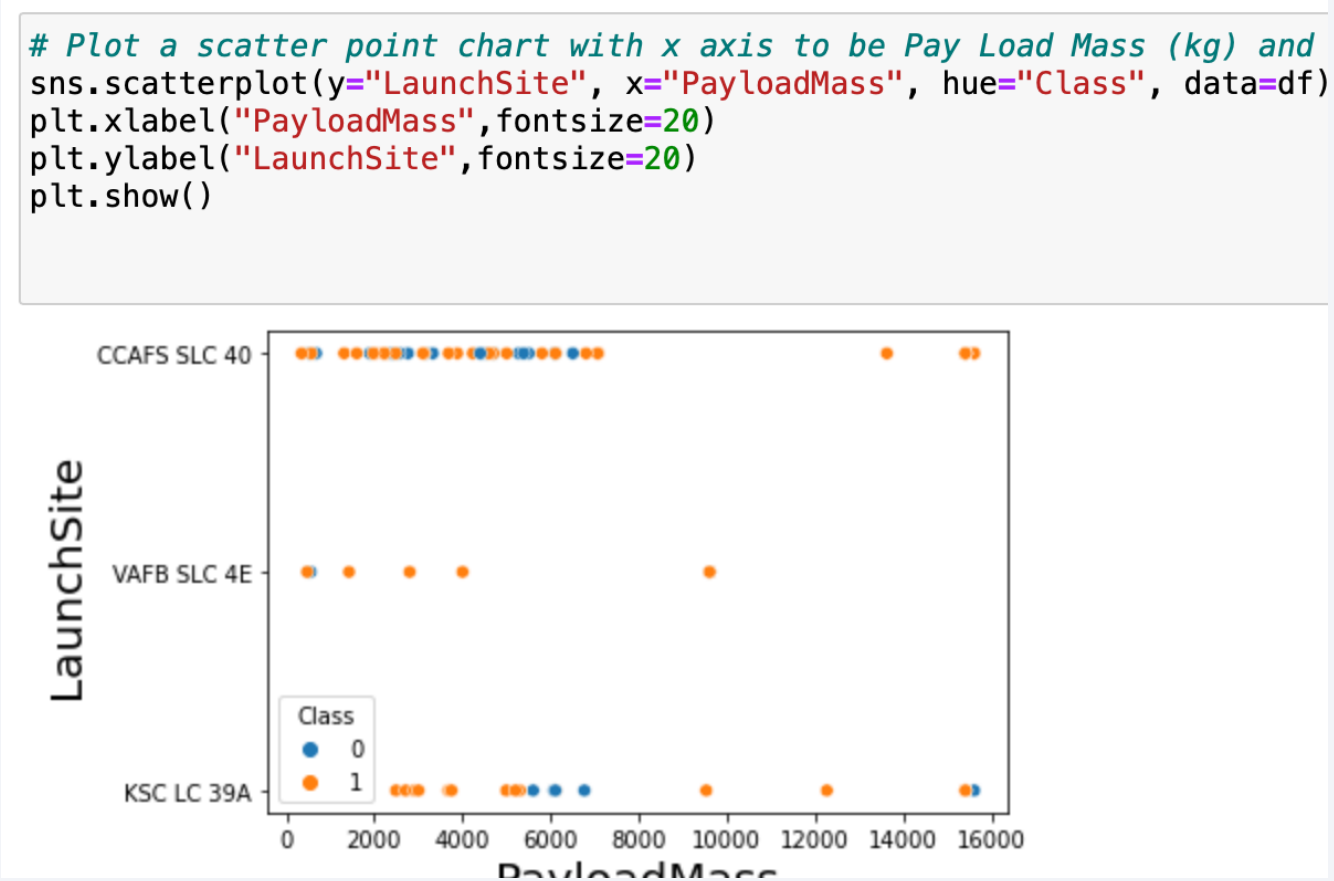
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

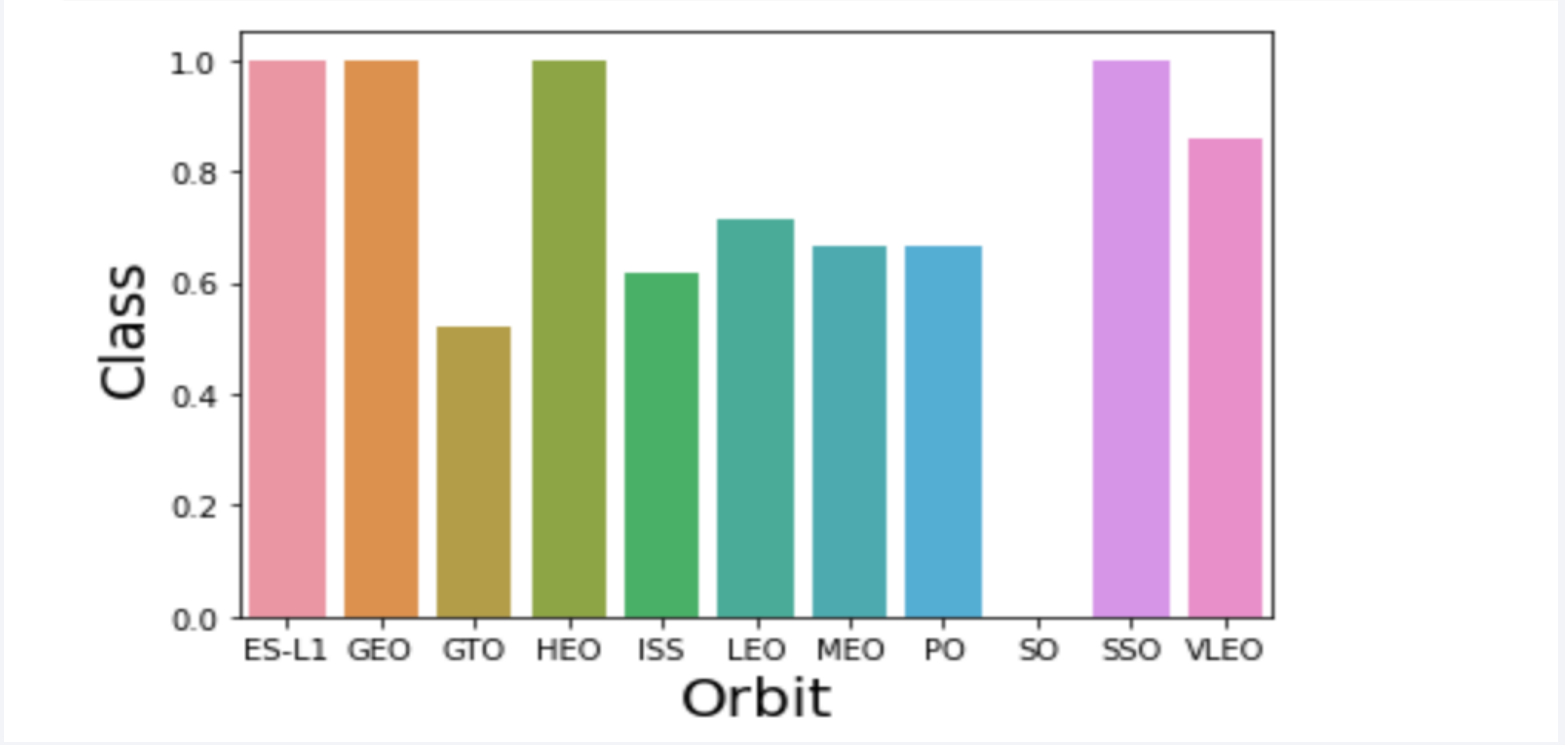
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and
sns.scatterplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

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

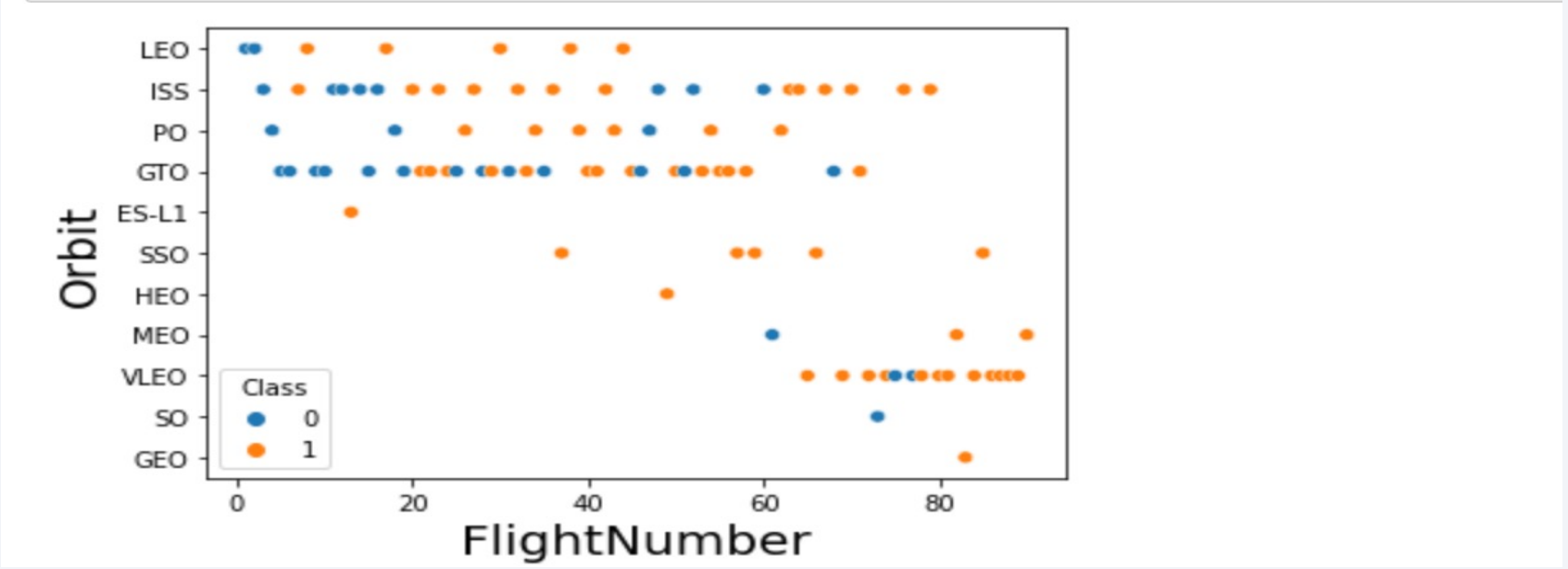
```
sns.barplot(y="Class", x="Orbit", data=df3)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.show()
```

- 

Flight Number vs. Orbit Type

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

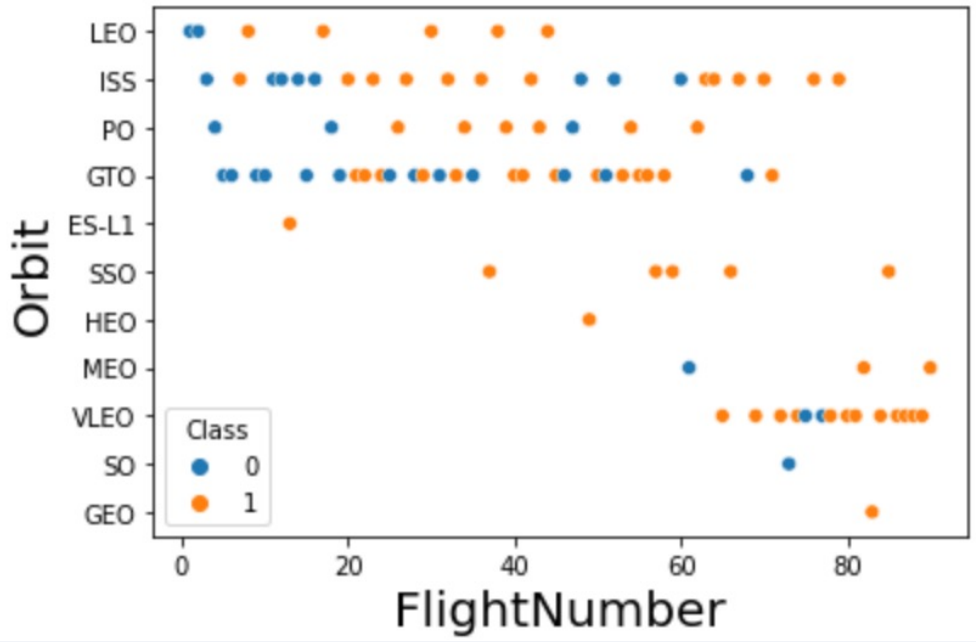
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis
ax=sns.scatterplot(y="Orbit", x="FlightNumber", hue="Class", data=df)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
sns.move_legend(ax,"lower left")
plt.show()
```



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

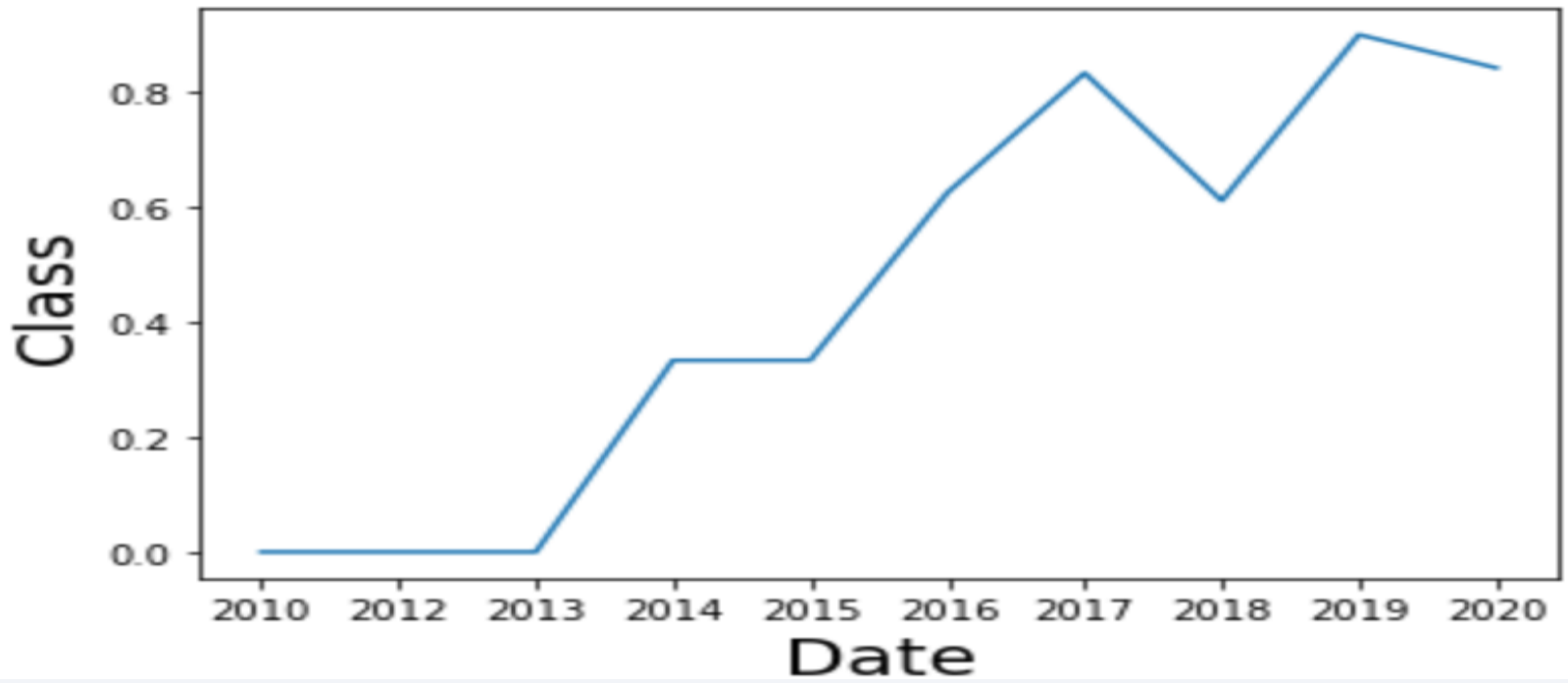
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis
ax=sns.scatterplot(y="Orbit", x="FlightNumber", hue="Class", data=df)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
sns.move_legend(ax,"lower left")
plt.show()
```



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
-

```
# Plot a line chart with x axis to be the extracte
ax=sns.lineplot(y="Class", x="Date", data=df6)
plt.xlabel("Date",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.show()
```



All Launch Site Names

- Find the names of the unique launch

```
%%sql
select distinct(launch_site) from spacextbl
```

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

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

Launch Site Names Begin with 'CCA'

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

```
%%sql
select * from spacextbl where launch_site like 'CCA%' limit 5
```

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

Date	Time	Booster_Version	Launch_Site	Payload	Payload_Mass	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

- Calculate the total payload carried by boosters from NASA

```
%%sql
select sum(PAYLOAD_MASS) from spacextbl where Customer like 'nasa%'
```

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

sum(PAYLOAD_MASS)
99980

Average Payload Mass by F9 v1.1

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

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

```
%%sql
select AVG(PAYLOAD_MASS) from spacextbl where Booster_Version='F9 v1.1'
```

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

AVG(PAYLOAD_MASS)
2928.4

First Successful Ground Landing Date

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

```
%%sql
select min(date) from spacextbl where Mission_Outcome='Success'
```

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

Date
04-06-2010

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

```
%%sql
select Booster_Version from spacextbl
where Landing_Outcome='Success (drone ship)'
and payload_mass between 4000 and 6000
```

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

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

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

Mission_Outcome	count(*)
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

```
%%sql
select booster_version from spacextbl where PAYLOAD_MASS= (select max(PAYLOAD_MASS) from spacextbl)
```

```
 * 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

```
%%sql
select booster_version, launch_site,date from spacextbl
where substr(Date,7,4)='2015'
and Landing_outcome='Failure (drone ship)'
```

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

Booster_Version	Launch_Site	Date
F9 v1.1 B1012	CCAFS LC-40	10-01-2015
F9 v1.1 B1015	CCAFS LC-40	14-04-2015

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

```
%%sql
select landing_outcome,count(*) from spacextbl
where Landing_Outcome like 'Success%'
and Date between '04-06-2010' and '20-03-2017'
group by landing_outcome
```

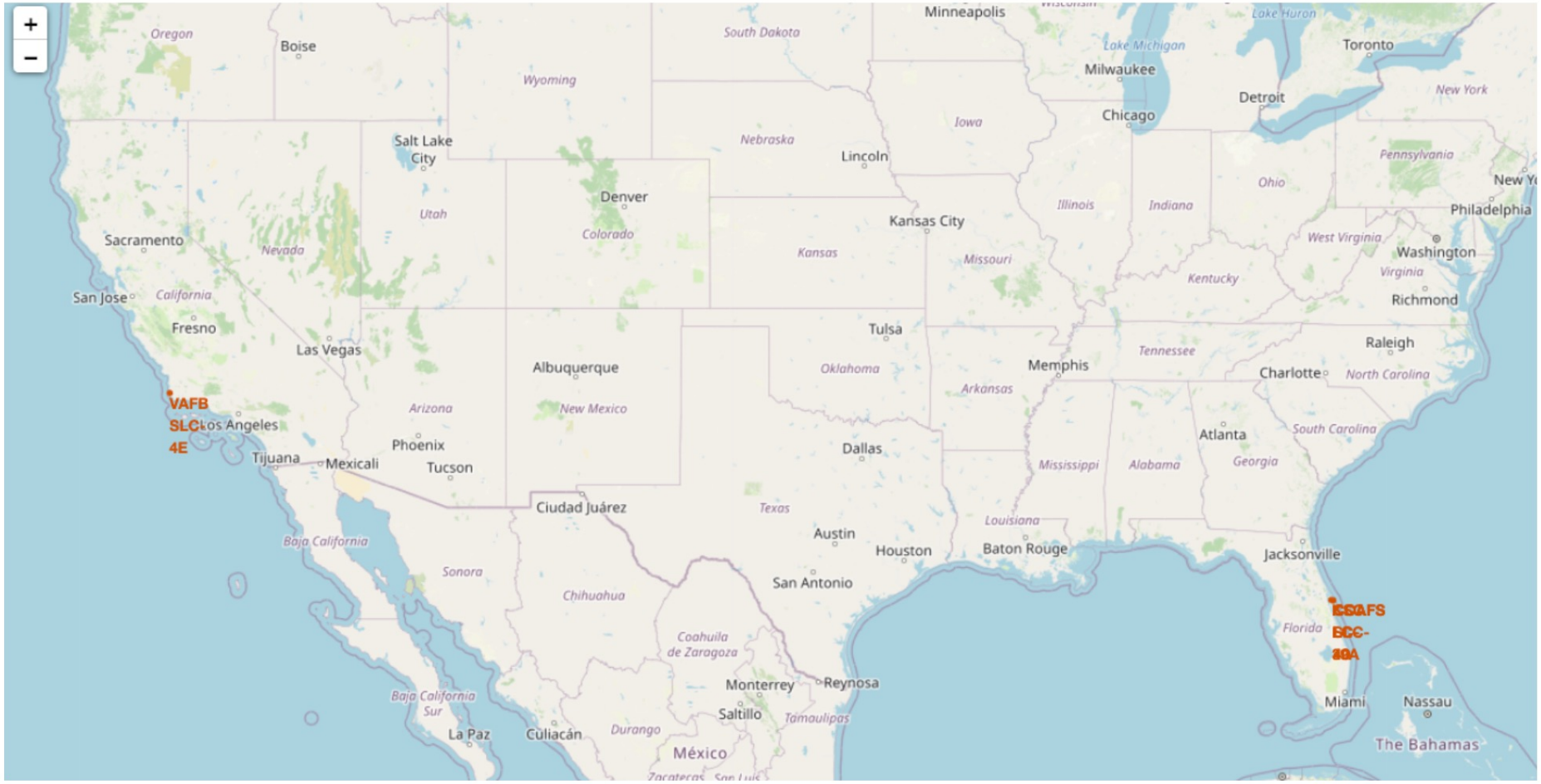
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count(*)
Success	20
Success (drone ship)	8
Success (ground pad)	6


Section 3
Launch Sites
Proximities Analysis

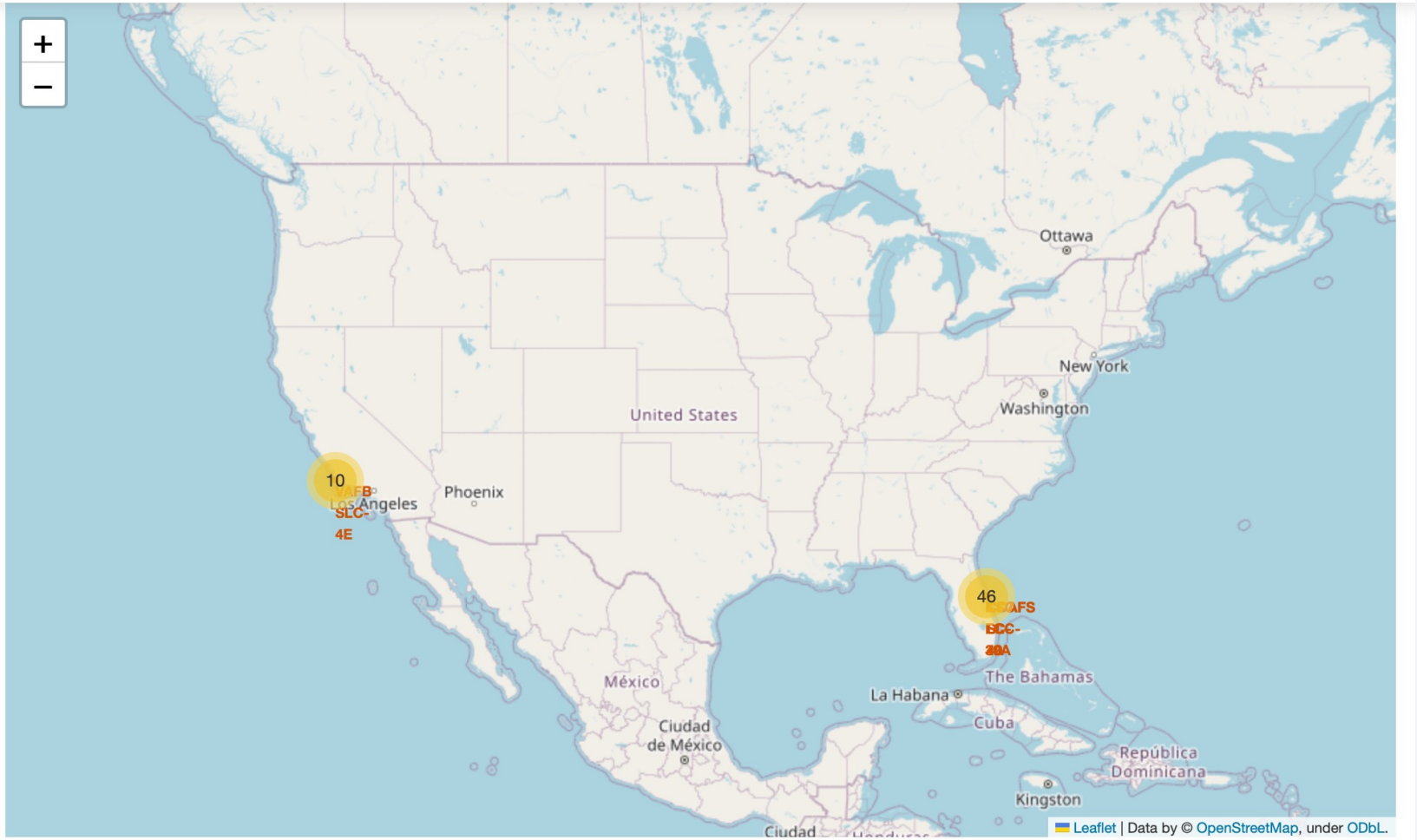
The generated map with marked launch sites

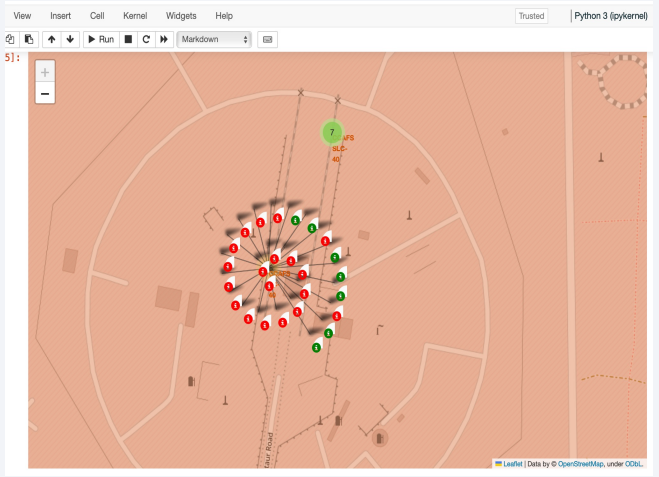
The generated map with marked launch sites



Launch site outcome

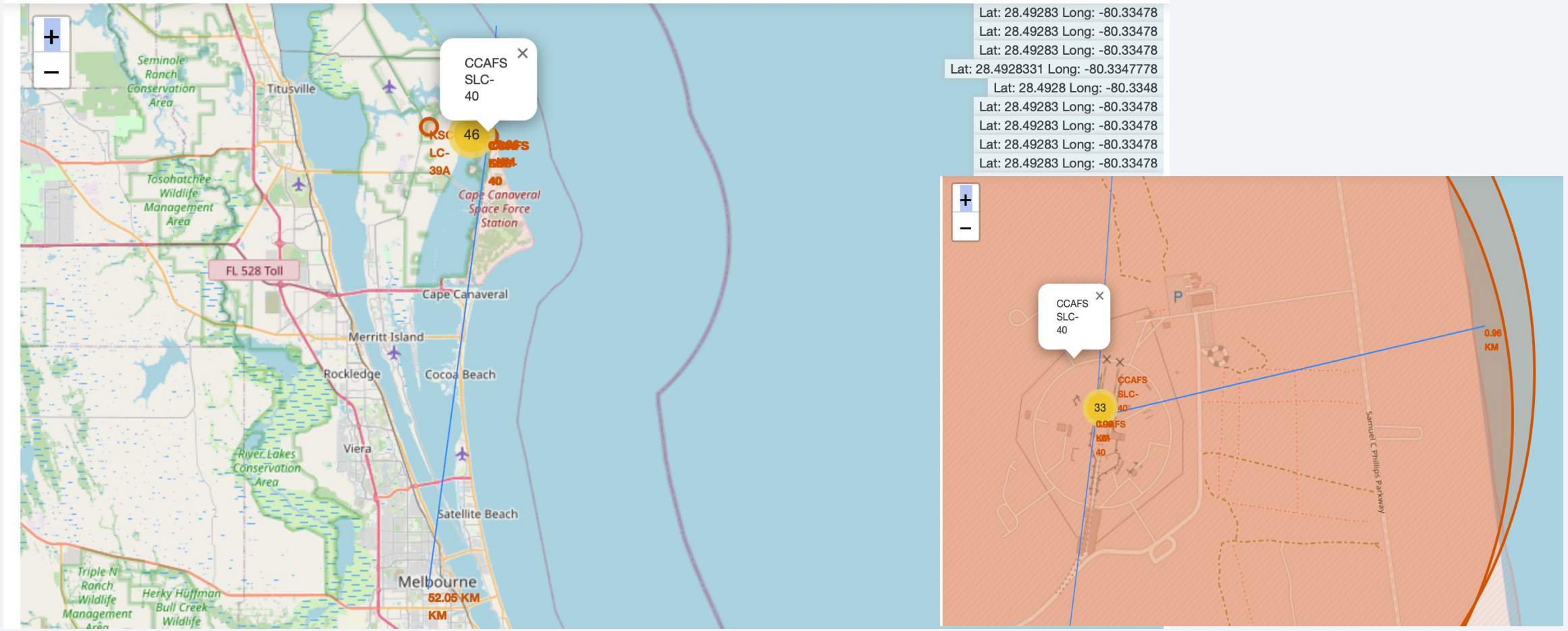
Launch site outcome

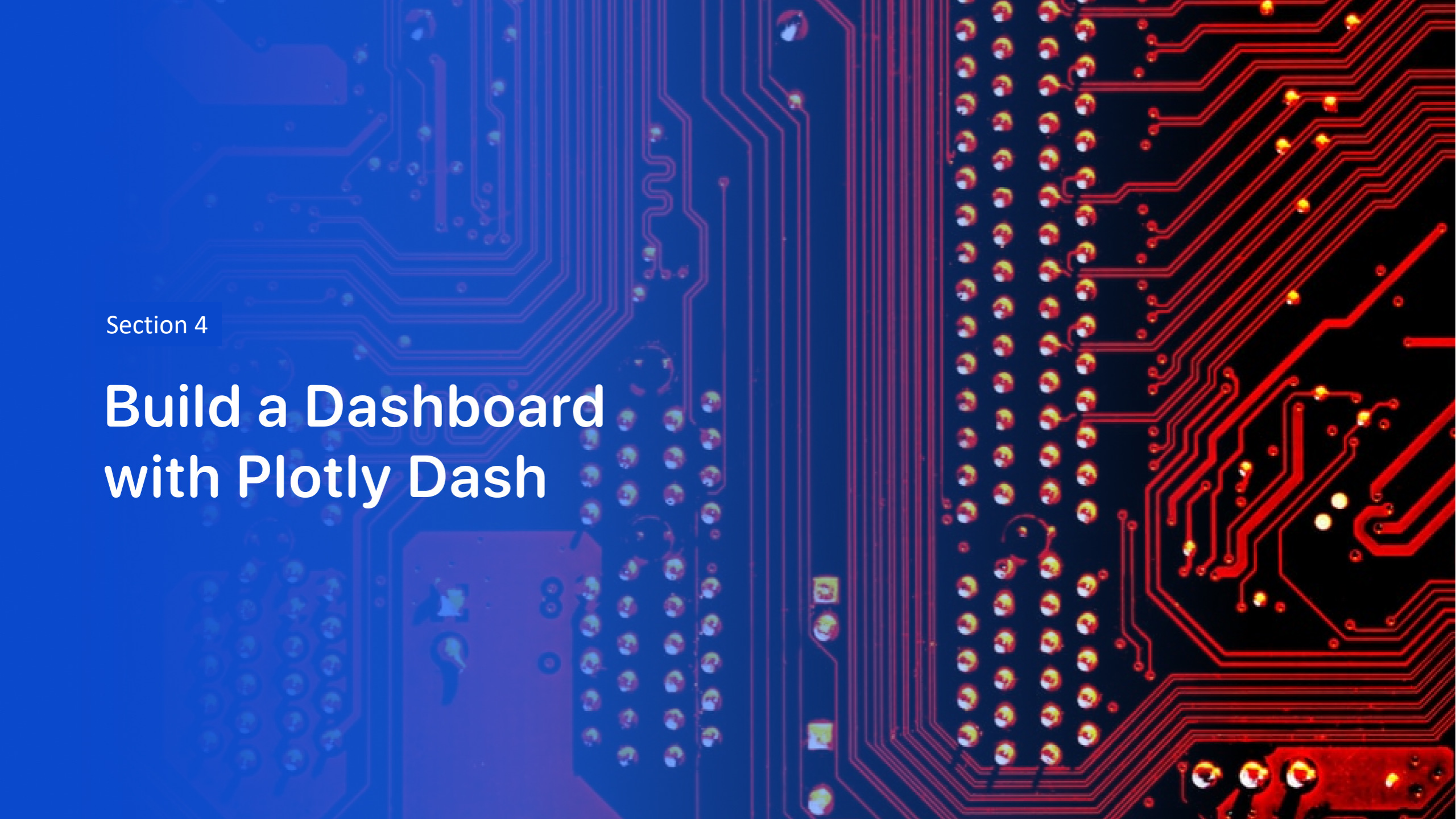




Launch site to facility distance

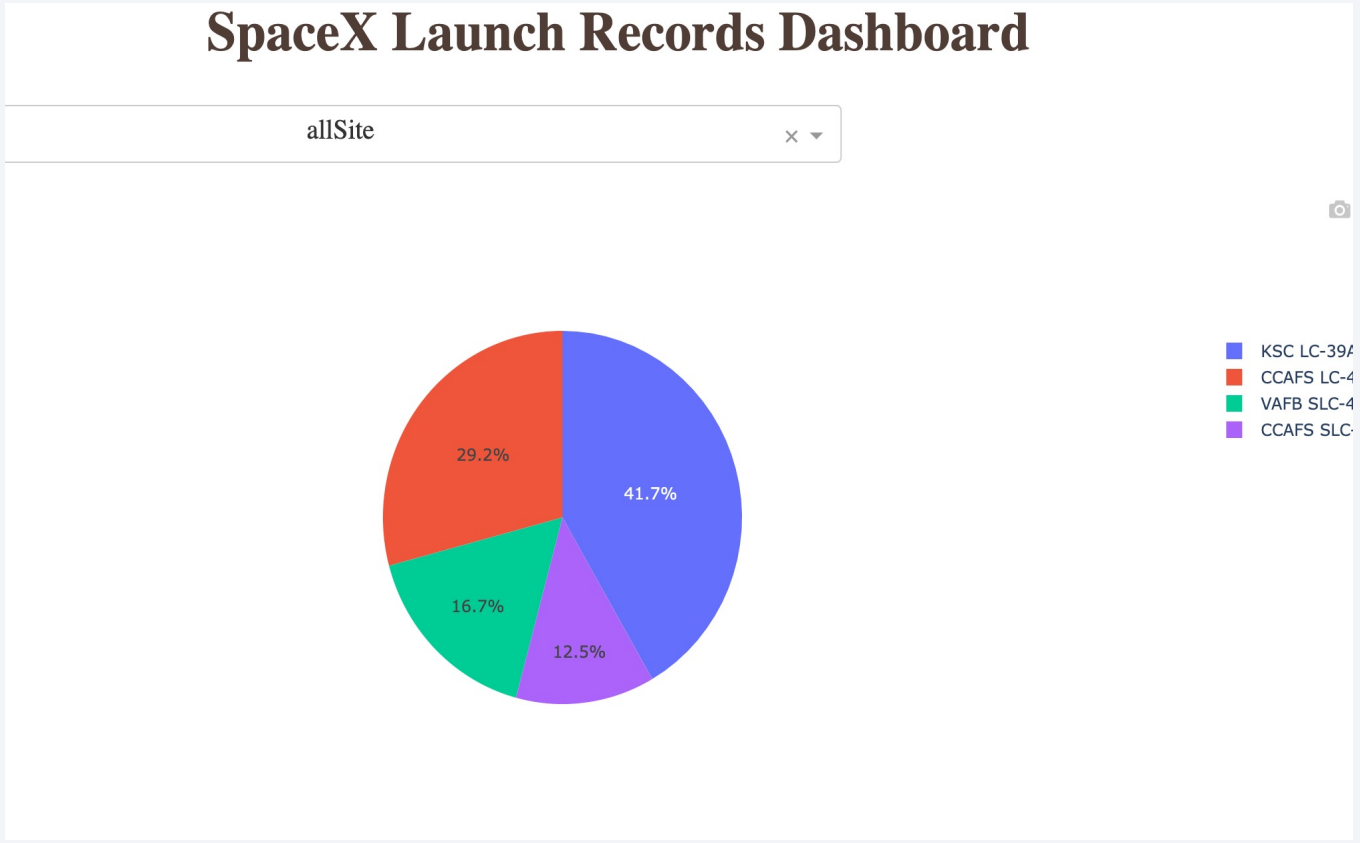
Launch site to facility distance




Section 4
Build a Dashboard with Plotly Dash

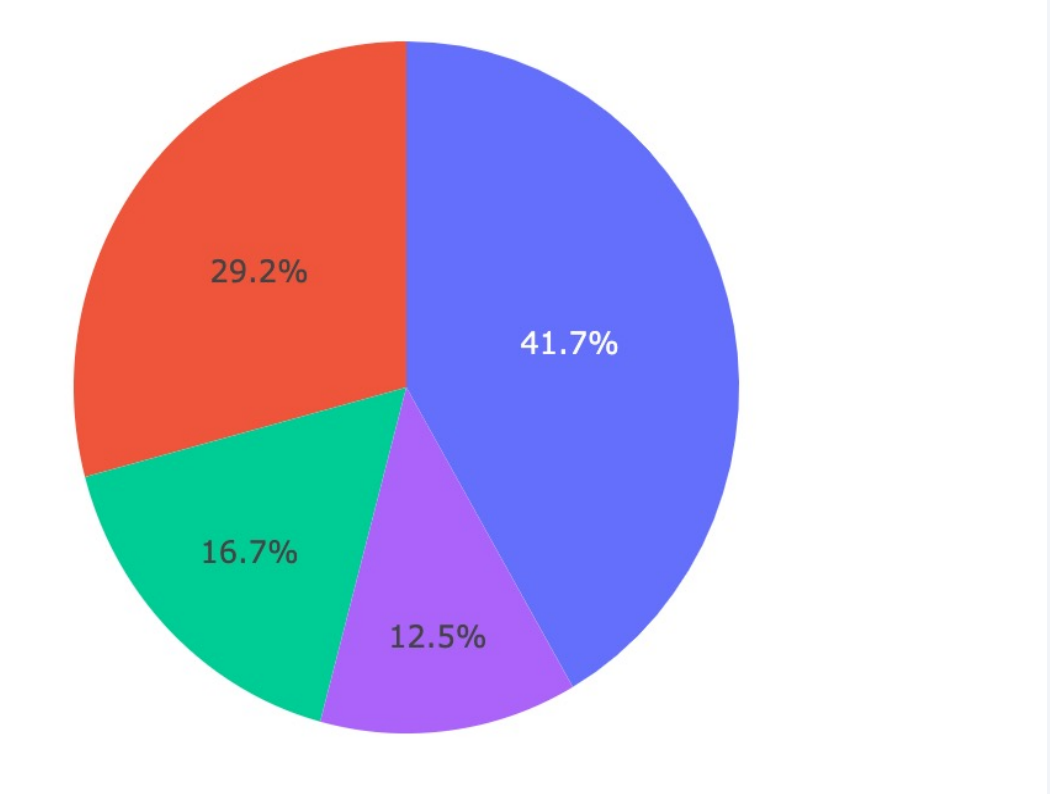
Each site success rate piechart

- Each site success rate piechart



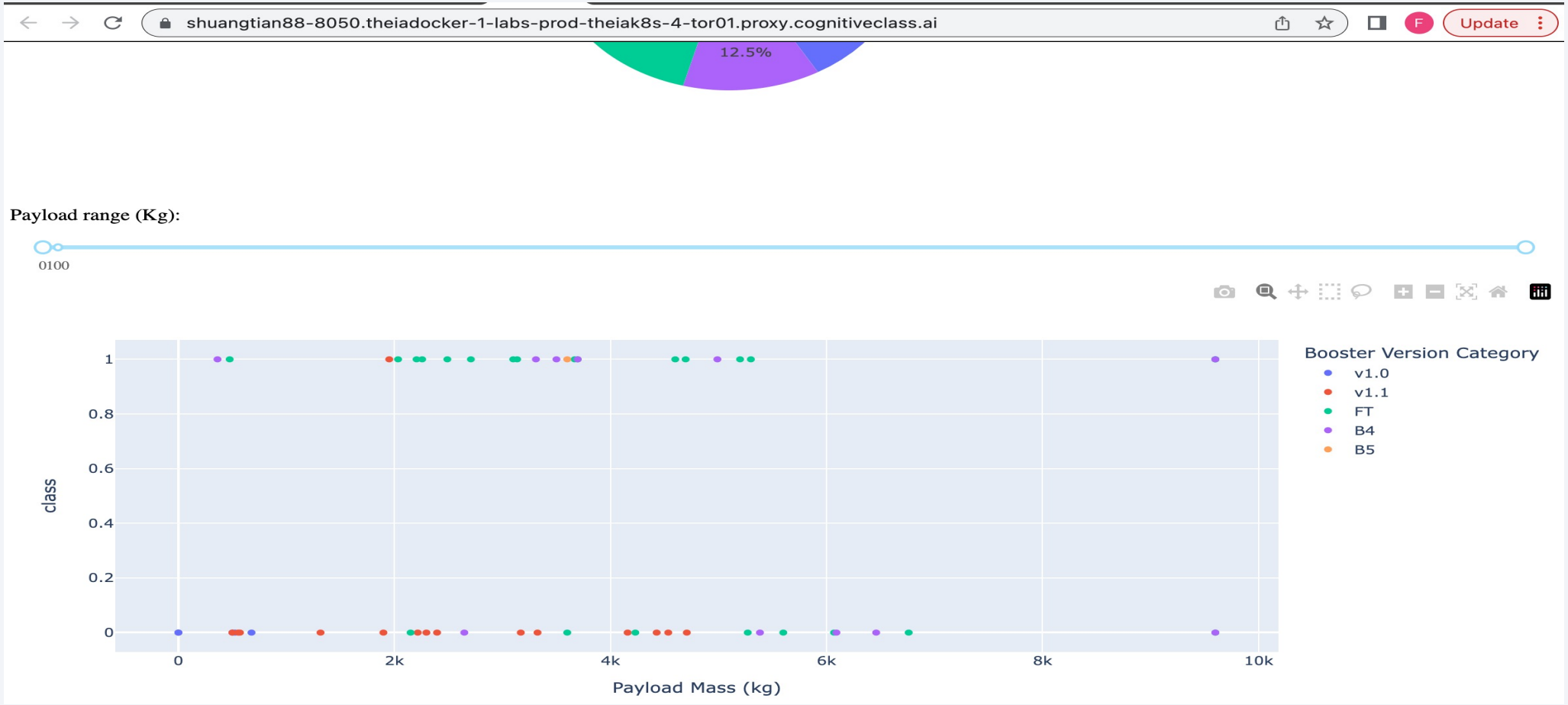
launch success ratio piechart

- launch success ratio piechart



Launch site & Payload launch outcome graph

- Launch site & Payload launch outcome graph



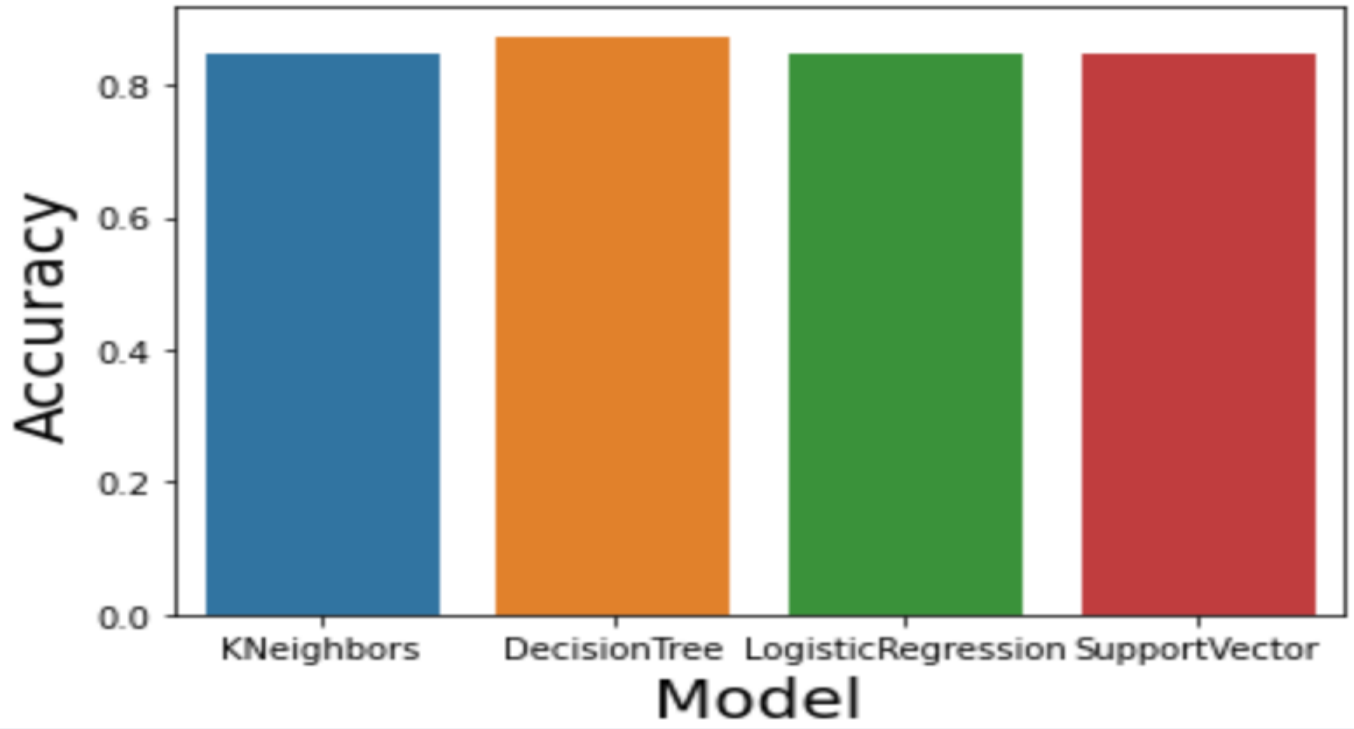
Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

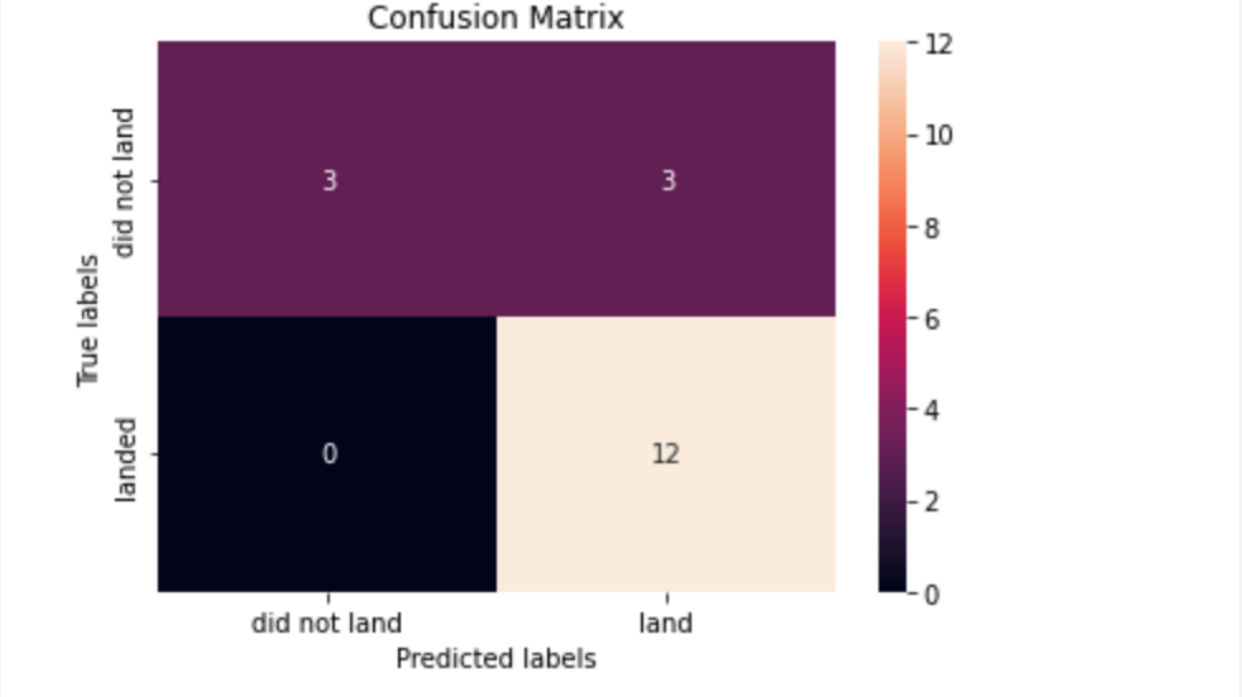
```
sns.barplot(y=1, x=0, data=df)
plt.xlabel("Model",fontsize=20)
plt.ylabel("Accuracy",fontsize=20)
plt.show()
```



Confusion Matrix

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

```
predTree= tree_model.predict(X_test)
plot_confusion_matrix(Y_test, predTree)
```



Conclusions

- Point 1 Decision Tree model is the highest accuracy model to predict if the first stage will land successfully.
- Point 2 The launch number or the date, launch site, payload, orbit, booster version are key factors to infect the first stage landing.
- Point 3 From 2010 to present, the spaceX more and more successful to reuse the first stage to reduce the launch costs.
- Point 4 Comparing the different classification model is necessary, metrics the accuracy to optimize the model.
- Point 5 exploratory data analysis and preparing data feature engineering are very useful.

Appendix

- 1 Sqlite can be used to set up database at local driver in this case, very helpful

```
In [2]: %load_ext sql

In [3]: import csv, sqlite3

        con = sqlite3.connect("my_data1.db")
        cur = con.cursor()

In [4]: !pip install -q pandas==1.1.5

In [5]: %sql sqlite:///my_data1.db
```

```
In [67]: import pandas as pd
         df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs
         df=df.rename({'Time (UTC)':'Time','Landing _Outcome':'Landing_Outcome','PAYLOAD_MASS__KG_':'Payload_Mass'},axis=1)
         df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

```
In [68]: df.columns
Out[68]: Index(['Date', 'Time', 'Booster_Version', 'Launch_Site', 'Payload',
                'Payload_Mass', 'Orbit', 'Customer', 'Mission_Outcome',
                'Landing_Outcome'],
               dtype='object')
```

```
In [69]: %%sql
         select * from spacextbl limit 2
 * sqlite:///my_data1.db
Done.
```

Out[69]:

Date	Time	Booster_Version	Launch_Site	Payload	Payload_Mass	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)

Thank you!