IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ADETUNJI PHILIP

22- 02-2022

Outline

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

Executive Summary

- **Summary of methodologies**

In this capstone, we were able to predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore we determine that the first stage will land, and that determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Summary of all results**

I was able to create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs. and it give me 81%.

Introduction

- Project background and context

In this capstone, we were able to predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore we determine that the first stage will land, and that determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
  - We were able to load the dataset, select some feature from it, we were able to out
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data Collection: The data has been collected by SpaceX API with the help of request library in python. Since the data was in JSON format, we have normalized it into a pandas data frame where we get the data in tabular form. There were 94 rows and 17 columns known as features. We only required the details of the Falcon 9 rocket launch, so we have removed all the rockets and kept only Falcon 9. We have observed that if there are any missing values, we replace them with the mean values as missing values don't give good results. To get an efficient outcome, we need to perform some

The data was collected from a wikipedia page titled list of falcon 9 and falcon heavy launches, the web scraping was done using beautiful soap

Data Collection – SpaceX API

1.Request to the SpaceX API

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

2. Clean the requested data

- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Data%20Collection%20and%20data%20cleaning1.ipynb

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Data%20Collection%20and%20data%20cleaning.ipynb

Data Wrangling

Data wrangling. Data Wrangling: After removing the unnecessary rows of rocket details and keeping only Falcon 9 rocket launch details, we got 90 rows and 17 columns. We have performed some analysis by calculating the number of launches on each site, and we observed that CCAFS SLC 40 has the maximum number of launches, i.e., 55, whereas launch site VAFB SLC 4E has the minimum, which is 13. Then we have calculated the number and occurrence of each orbit where GTO orbit has the maximum number, i.e., 27. We have also calculated the number and event of mission outcome per orbit type, and we observed that the number of successfully landed results is better than the unsuccessful ones.

https://github.com/Adetayo047/IBM-Data-Science/blob/master/Data%20Collection%20and%20data%20cleaning1.ipynb

EDA with Data Visualization

Exploratory Data Analysis: In this section, we determined the EDA with the help of matplotlib and seaborn library. We have performed so many relationships between the features, such as comparison between Flight Number and Launch Site, comparison between Payload and Launch Site, comparison between the success rate of each orbit type, comparison between FlightNumber and Orbit type, and the comparison between Payload and Orbit type. With the help of plotting graphs, we have observed that VAFB-SLC launch site, there are no rockets launched for heavy payload mass(greater than 10000) and LEO orbit the Success appears related to the number of flights. On the other hand, there seems to be no relationship between flight numbers and GTO orbit. Also, the success rate since 2013 kept increasing till 2020.

https://github.com/Adetayo047/IBM-Data-Science/blob/master/Complete%20the%20EDA%20with%20Visualization%20lab.ipynb

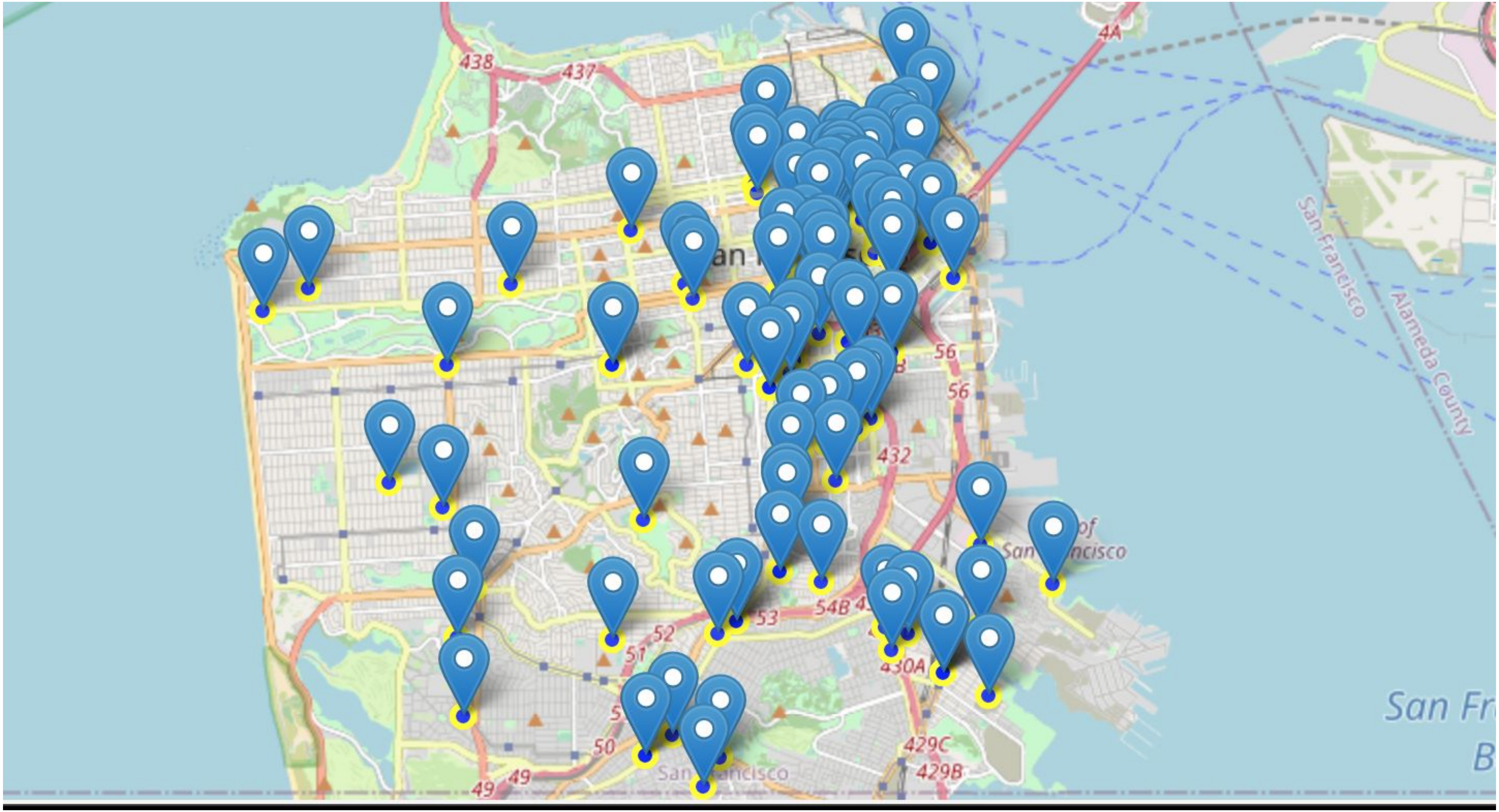
EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Display the names of the unique launch sites in the space mission
  - **%sql** select * from spacex
  - **%sql** select distinct launch_site from spacex
  - **%sql** select * from spacex where launch_site like 'CCA%'limit 5
  - **%sql** select sum(payload_mass__kg_) as sumpayload from spacex where customer = 'NASA (CRS)'
  - **%sql** SELECT AVG(payload_mass__kg_) FROM spacex WHERE booster_version='F9 v1.1'
  - **%sql** SELECT DATE FROM spacex WHERE landing__outcome = 'Success (ground pad)'
  - **%sql** SELECT booster_version from spacex WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ between 4000 AND 6000
  - **%sql** SELECT DATE FROM spacex WHERE landing__outcome = 'Success (ground pad)'
- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Complete%20the%20EDA%20with%20SQL%20lab.ipynb
-

Build an Interactive Map with Folium

- ❖ It help in customizing the world map by specifying the centre of the map, and the initial zoom level. Defining all locations on a map by their respective *Latitude* and *Longitude* values. Therefore I create a map and pass in a center of *Latitude* and *Longitude* values of **[0, 0]**.
- ❖ For a defined center, you can also define the initial zoom level into that location when the map is rendered. **The higher the zoom level the more the map is zoomed into the center** and also create a map centered around Canada and play with the zoom level to see how it affects the rendered map.

- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Generating%20Maps%20with%20Python.ipynb


438
437
4A
San Francisco
Alameda County
56
56
432
54B
53
52
51
50
49
49
430A
429C
429B
San Francisco

Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels

1. create a column for the class
2. Standardize the data
3. Split into training data and test data

Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

1. Find the method performs best using test data

https://github.com/Adetayo047/IBM-Data-Science/blob/master/Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

- **Exploratory data analysis results**

- **Interactive analytics demo in screenshots**

Predictive analysis results

- Best Algorithm is Tree with a score of 0.875

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbit_HEO	Orbit_ISS	...	Serial_B1058	Serial_B1
0	1.0	6104.959412	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
1	2.0	525.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
2	3.0	677.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0
3	4.0	500.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
4	5.0	3170.000000	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	86.0	15400.000000	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
86	87.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0
87	88.0	15400.000000	6.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
88	89.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
89	90.0	3681.000000	1.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0

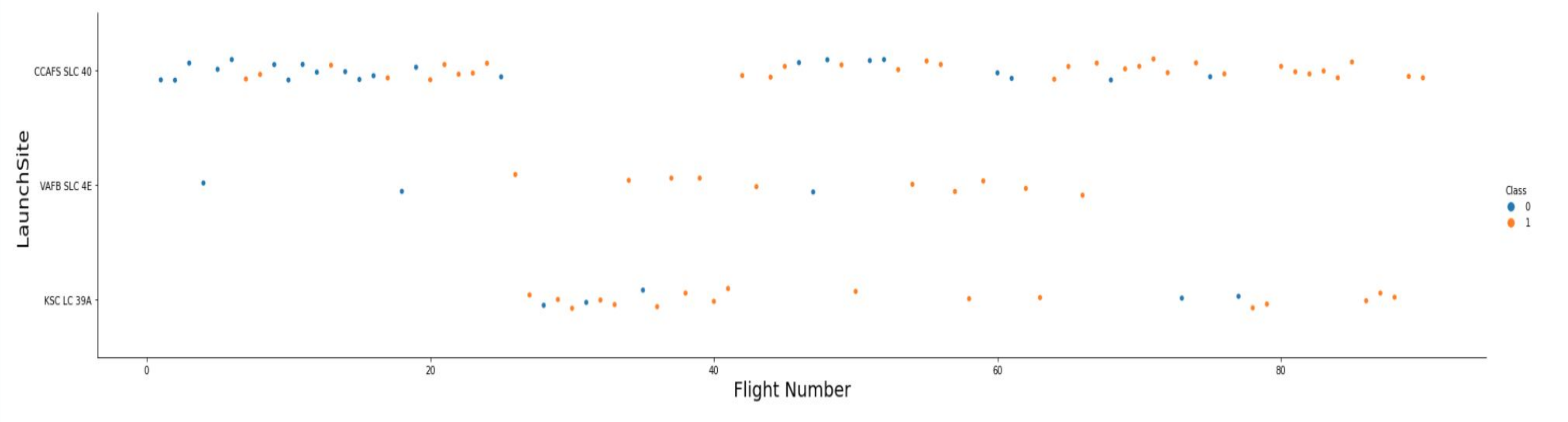
90 rows x 83 columns

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

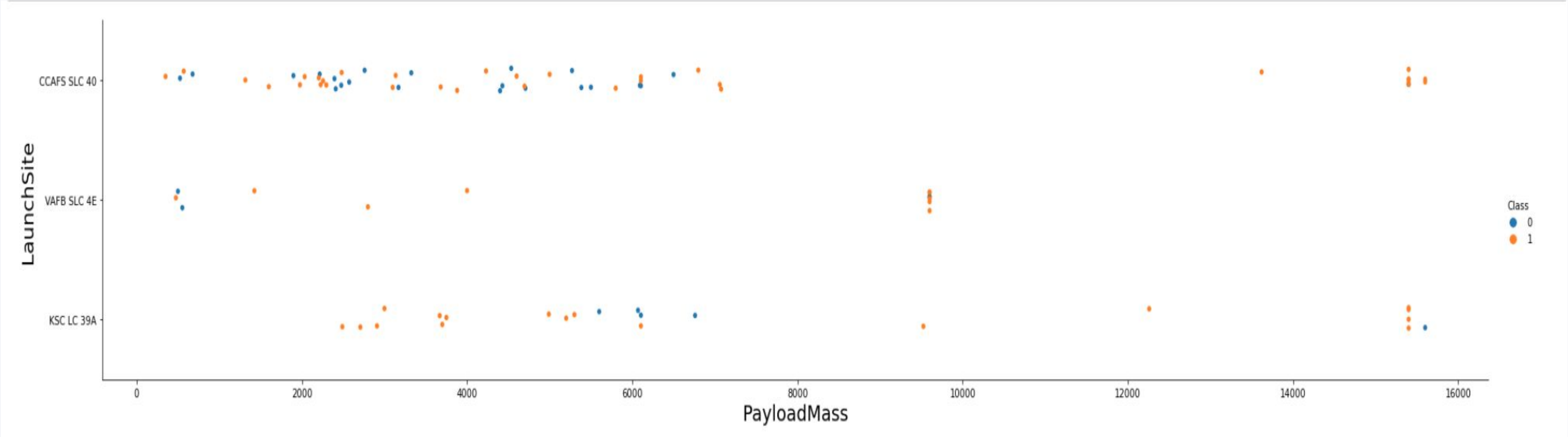
Show a scatter plot of Flight Number vs. Launch Site



We see that there is a no significant increase or decrease in the flight number especially for the launchSite VA FB SLC 4E, KSC LC 39A. Which show no relationship or a week relationship between Flight Number and LaunchSite

Payload vs. Launch Site

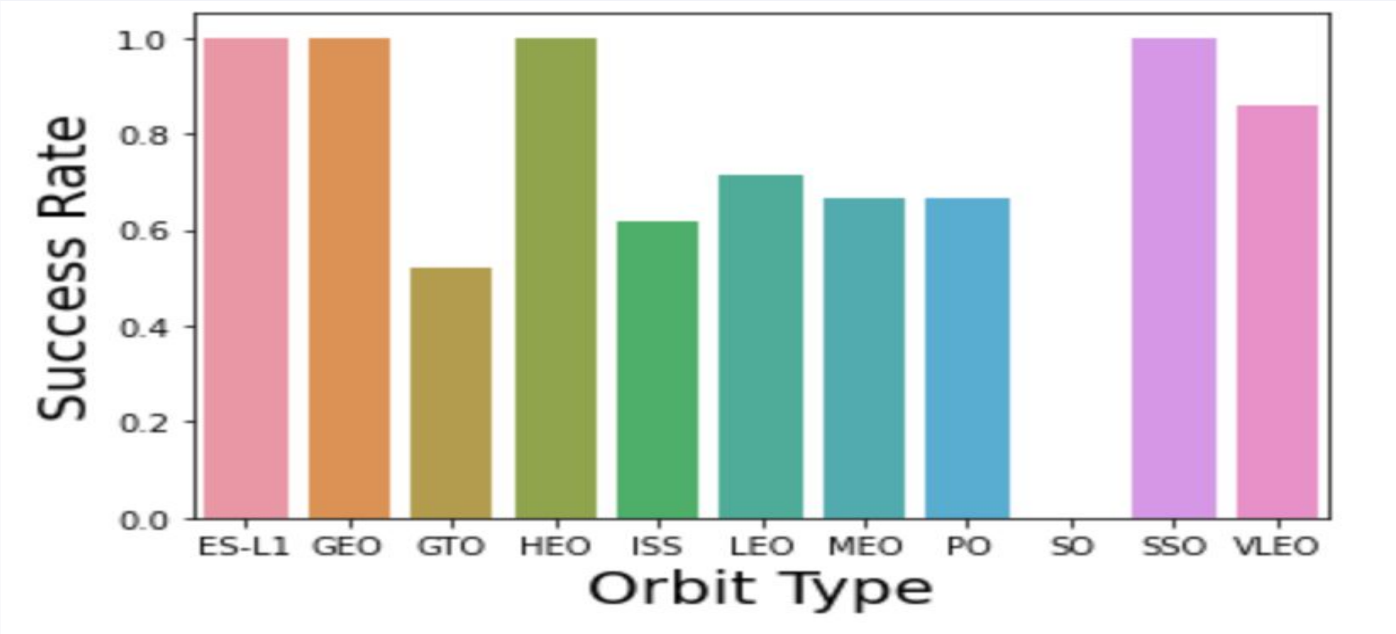
- Show a scatter plot of Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type

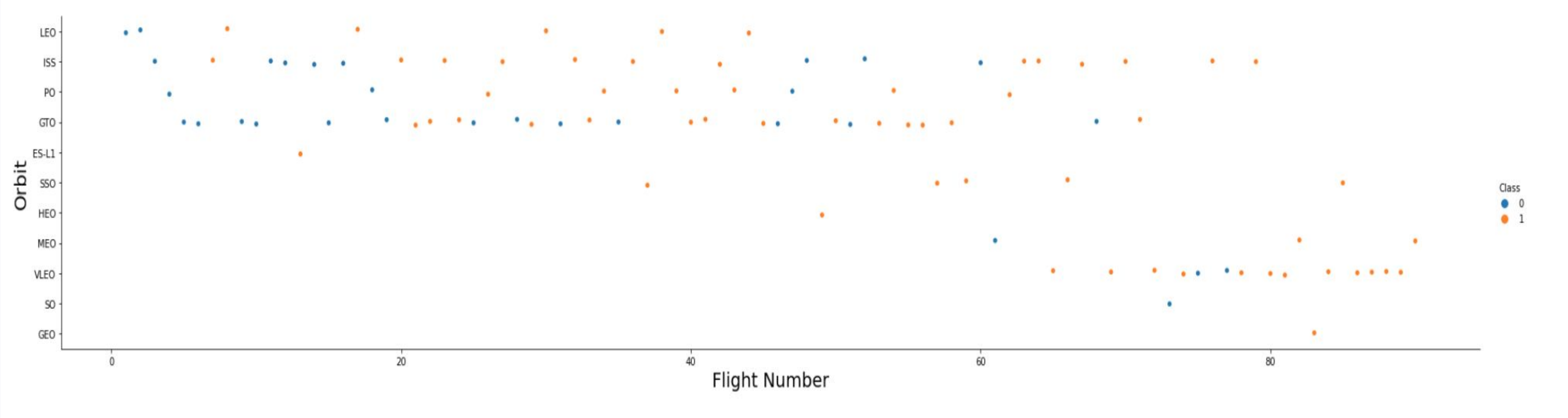
- Show a bar chart for the success rate of each orbit type



- We see from the chart that the orbit type that has the highest success rate are the ES-L1, GEO,HEO, and SSO but the orbit type with no success rate is SO.

Flight Number vs. Orbit Type

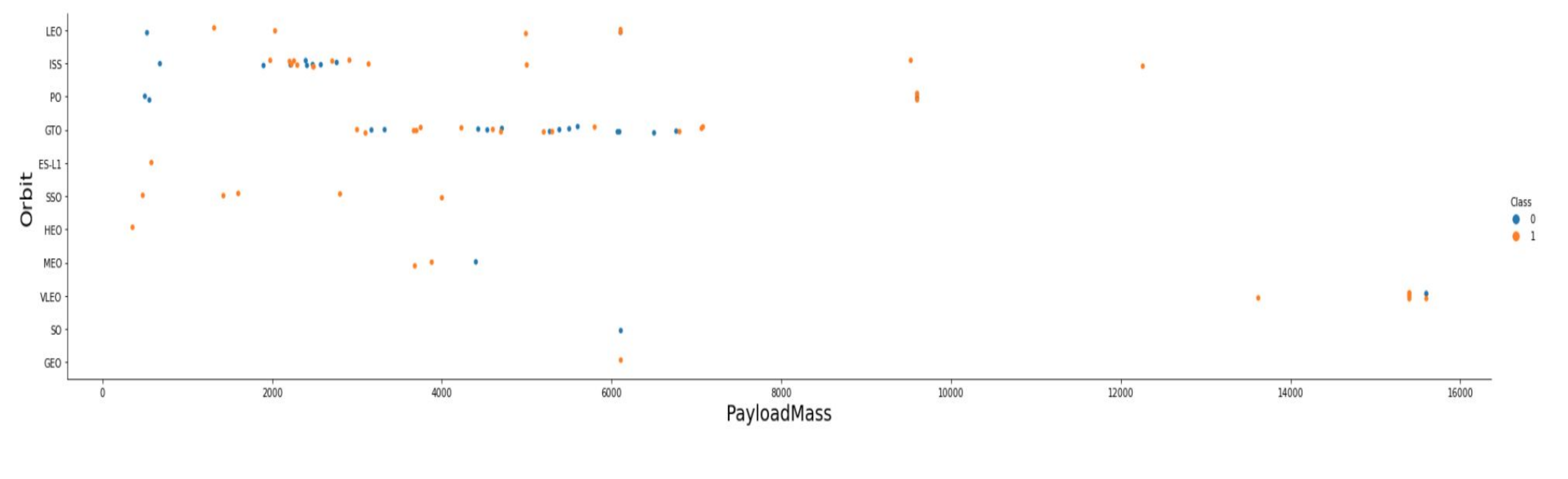
- Show a scatter point of Flight number vs. Orbit type



to some extent we can say there is a week relationship between Flight Number and orbit and You should see that in 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 GTO orbit

Payload vs. Orbit Type

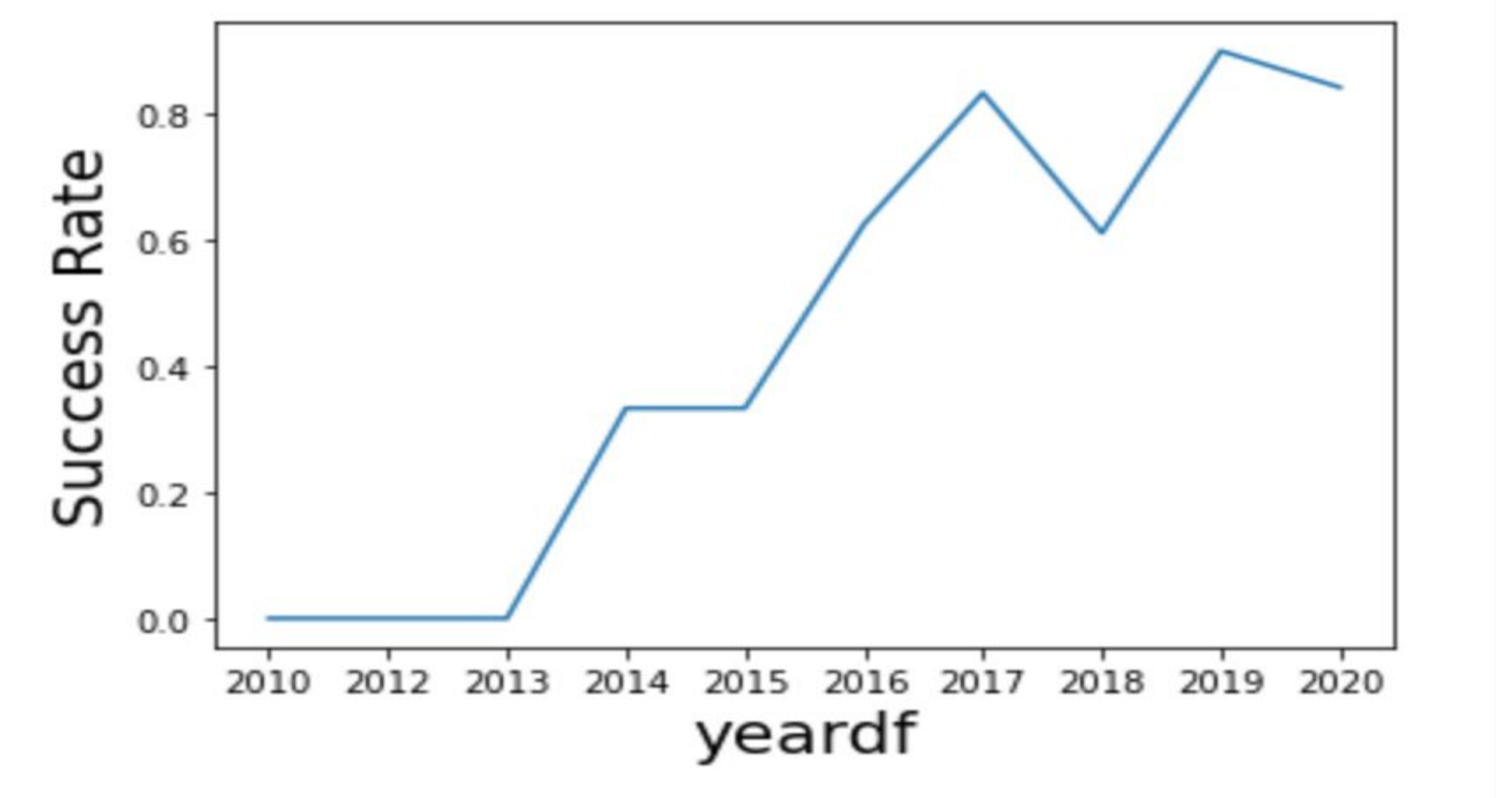
- Show a scatter point of payload vs. orbit type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



you can observe that the success rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites

```
]: %sql select distinct launch_site from spacex

 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4
4/BLUDB
Done.
```

]:

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

I was able to select the distinct or unique launch site from the dataset or table (spacex)

Launch Site Names Begin with 'CCA'

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

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

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3230
4/BLUDB
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Selecting all the records from table spacex that the launchsite column begins with ‘CCA’

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(payload_mass__kg_) as sumpayload from spacex where customer = 'NASA (CRS)'

 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.database
4/BLUDB
Done.
```

sumpayload
45596

With the help of SQL fetching out the total payload carried by boosters from NASA

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(payload_mass__kg_) FROM spacex WHERE booster_version='F9 v1.1'
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.database
4/BLUDB
Done.
```

1
2928

- this query is used to select the average payload mass carried by the booster version F9 v1.1

First Successful Ground Landing Date

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

```
%sql SELECT DATE FROM spacex WHERE landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.app
4/BLUDB
Done.
```

DATE
2015-12-22
2016-07-18
2017-02-19
2017-05-01
2017-06-03
2017-08-14
2017-09-07
2017-12-15
2018-01-08

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship

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 spacex WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ between
4000 AND 6000
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3230
4/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

```
%sql SELECT COUNT(*) FROM spacex WHERE mission_outcome LIKE '%Success%' OR mission_outcome LIKE '%Failure%'
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:323(
4/BLUDB
Done.
```

1
101

Boosters Carried Maximum Payload

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

```
%sql SELECT booster_version FROM spacex WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM spacex)
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:323
4/BLUDB
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,landing__outcome,launch_site from spacex where landing__outcome ='Failure (drone ship)' AN
D YEAR(DATE)=2015
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3230
4/BLUDB
Done.
```

booster_version	landing__outcome	launch_site
F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
F9 v1.1 B1015	Failure (drone ship)	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

```
]: %sql SELECT DATE,COUNT(landing__outcome) as landoutcount,YEAR(DATE) AS YEARS,MONTH(DATE) AS MONTH from spacex WHERE (l
anding__outcome LIKE '%Failure (drone ship)%' \
or landing__outcome LIKE '%Success (ground pad)%') and DATE between '2010-06-04' and '2017-03-20' group by DATE order
by landoutcount desc
```

```
 * ibm_db_sa://xkr78011:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3230
4/BLUDB
Done.
```

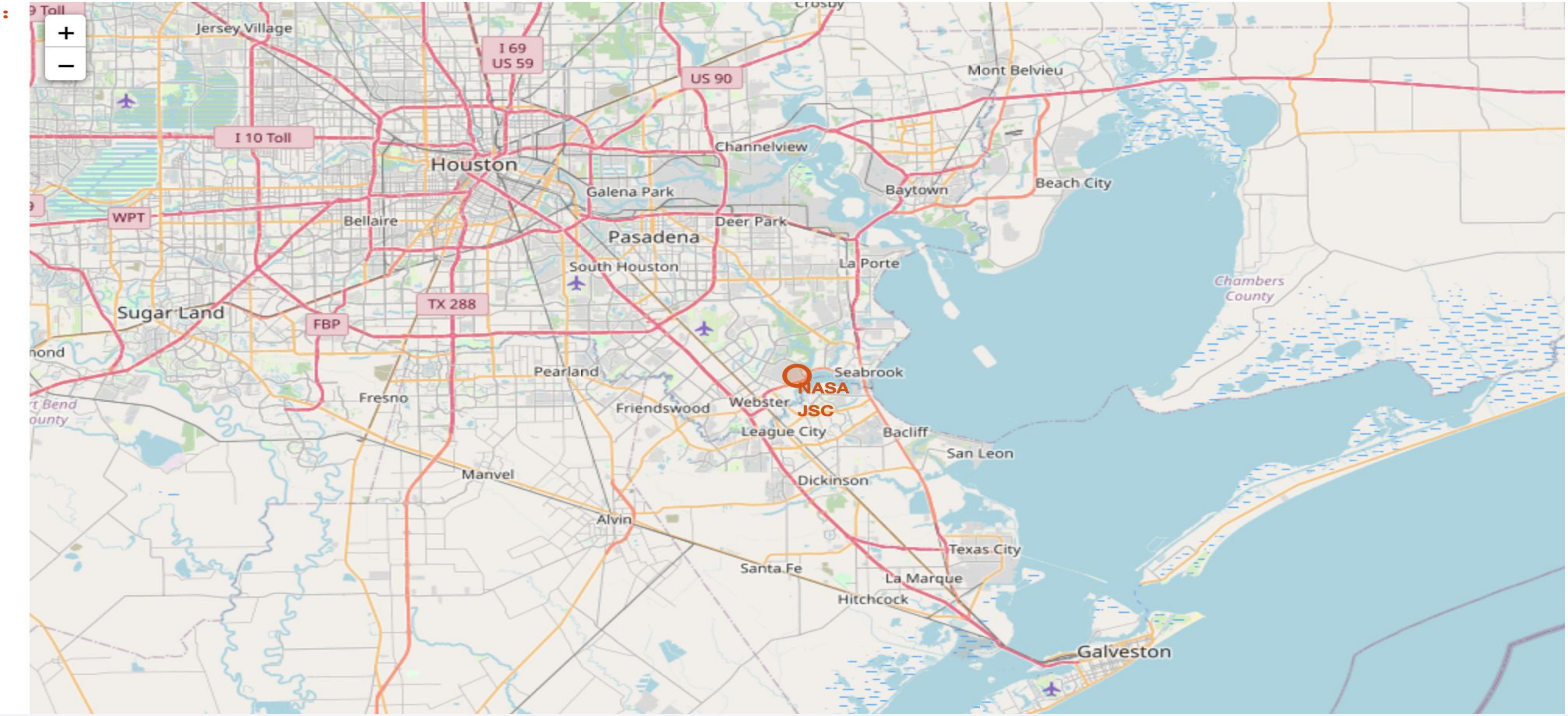
]:

DATE	landoutcount	YEARS	MONTH
2015-01-10	1	2015	1
2015-04-14	1	2015	4
2015-12-22	1	2015	12
2016-01-17	1	2016	1
2016-03-04	1	2016	3
2016-06-15	1	2016	6
2016-07-18	1	2016	7
2017-02-19	1	2017	2

Section 3

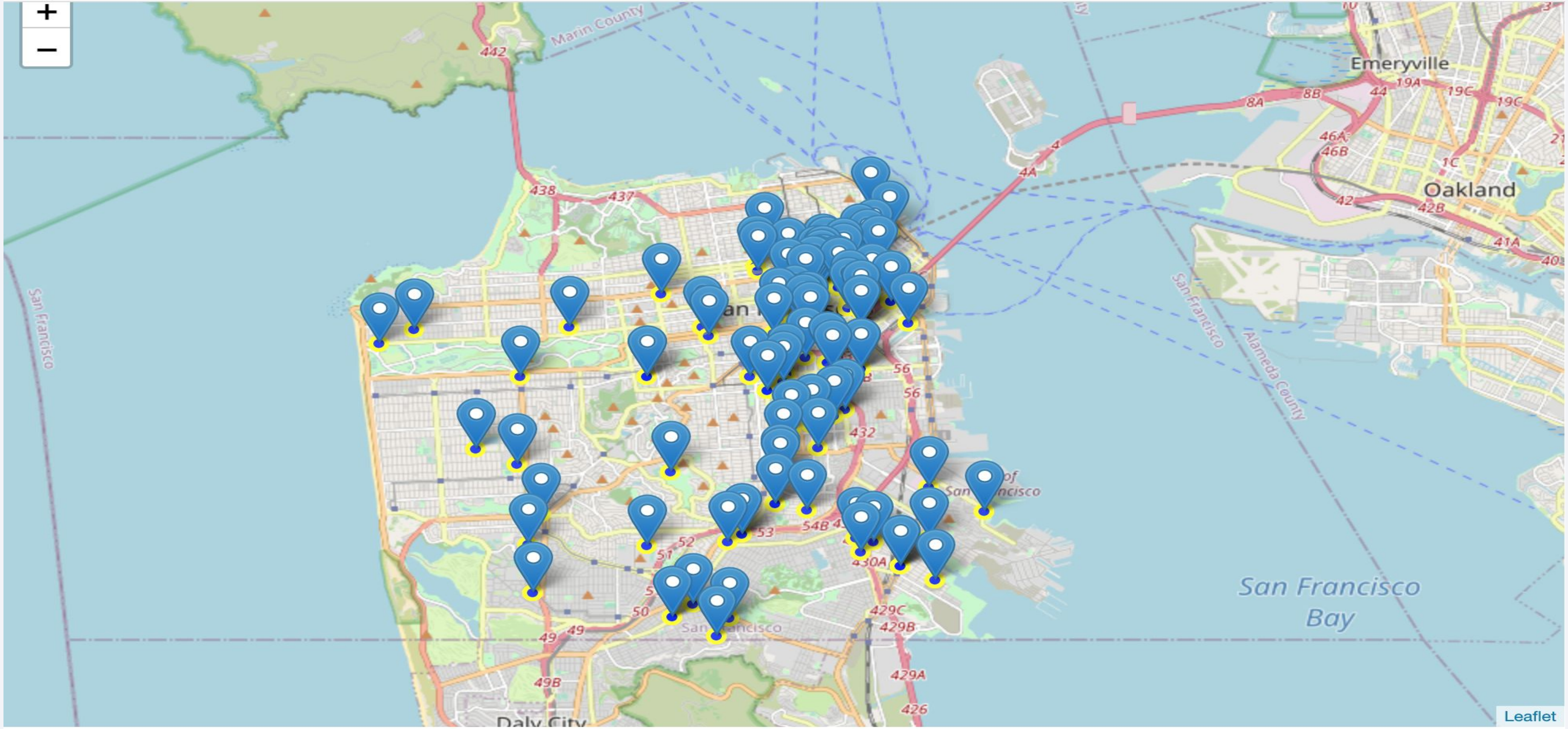
Launch Sites Proximities Analysis

Launch Sites Locations Analysis with Folium



Stamen Terrain Maps

Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Calculate the accuracy on the test data using the method score:

```
print('Accuracy=  ',logreg_cv.score(X_test,Y_test))
```

```
Accuracy=   0.8333333333333334
```

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.031622776601
accuracy : 0.8482142857142856
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_c
es_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.875
```

```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

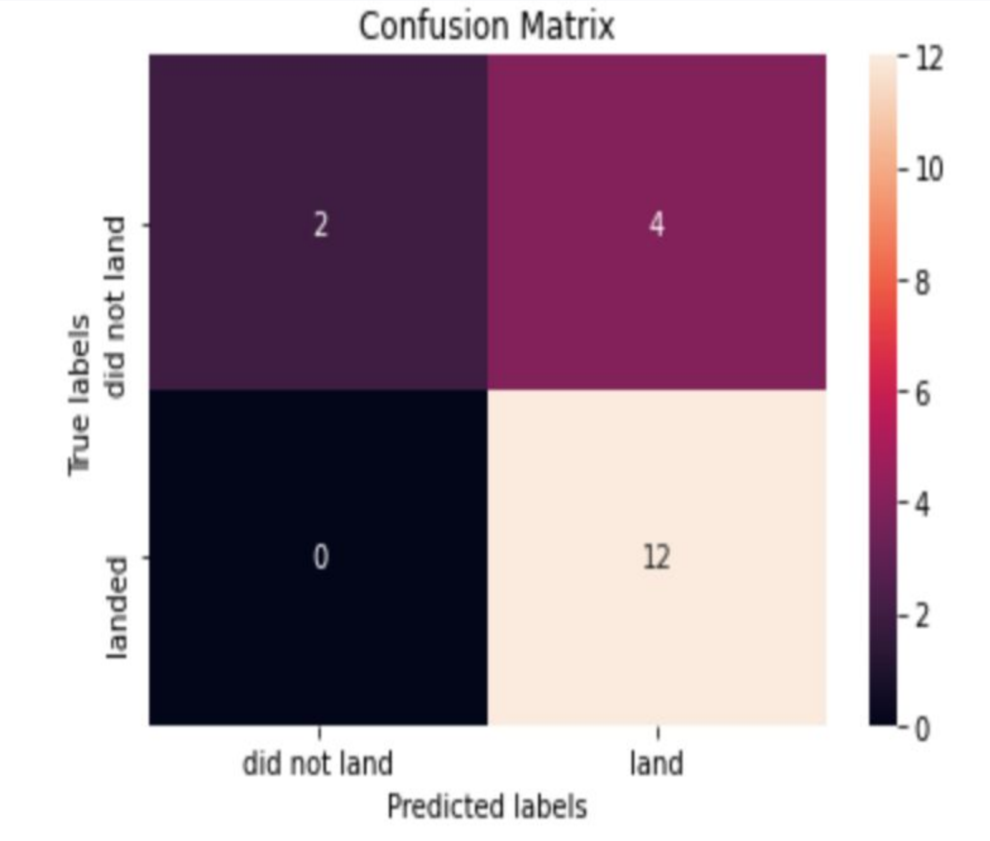
```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neig
accuracy : 0.8482142857142858
```

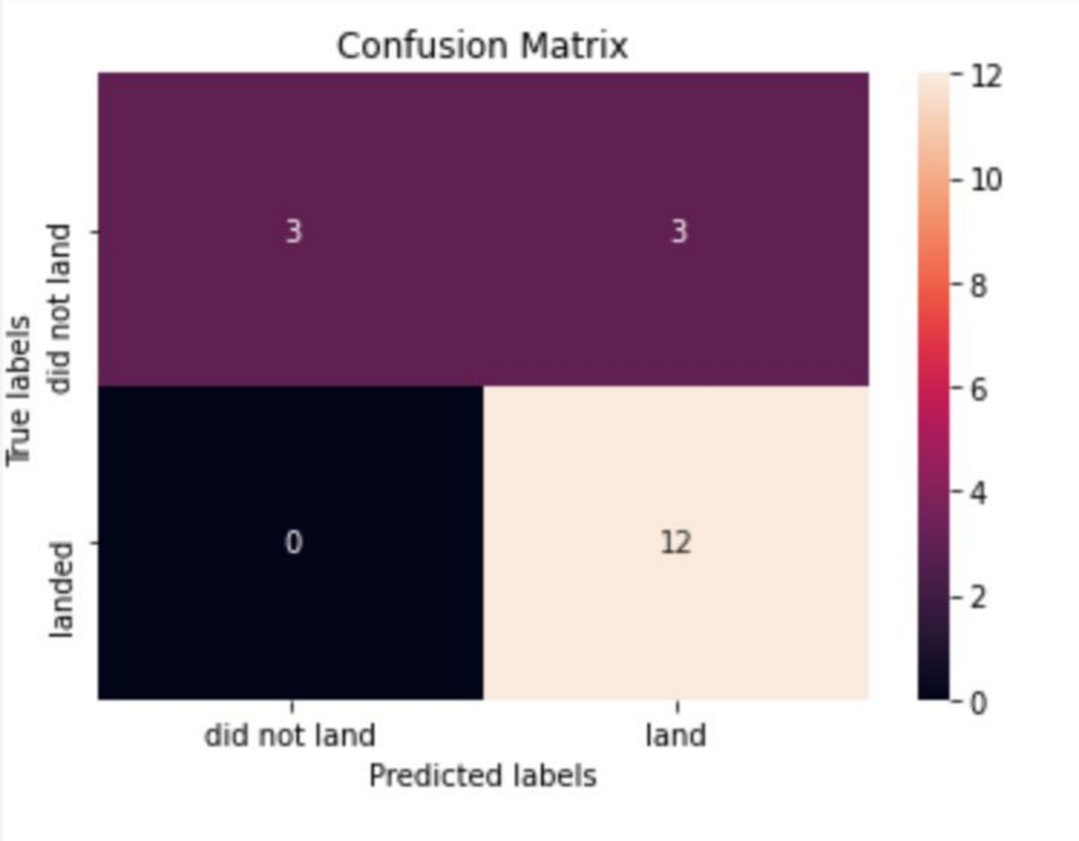
Best Algorithm is Tree with a score of 0.875

Best Params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}

Confusion Matrix

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





Conclusions

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.875
Best Params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples
_split': 5, 'splitter': 'random'}
```

After training our model, we get quite a good score with an accuracy of 87.50%. We can conclude that our model is performing well and giving us good results. Thus, our objective can be accomplished. SpaceX can minimize the cost by reusing the first stage.

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

- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Complete%20the%20EDA%20with%20SQL%20lab.ipynb
- https://github.com/Adetayo047/IBM-Data-Science/blob/master/Complete%20the%20EDA%20with%20Visualization%20lab.ipynb
- https://github.com/Adetayo047/IBM-Data-Science/commit/ed3f7e3a7deb9d17b74d630cd38370c279da2bb0
- https://github.com/Adetayo047/IBM-Data-Science/commit/13aad2084f8830c980e9cc648001769adbc22182
-

Thank you!