SpaceX Landing Prediction

Theodar A

27-10-2021

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- The goal of this project is to predict if Falcon 9 first stage will land successfully.
- To do this, we use the data posted by SpaceX in its website. A rocket's success depends on the following
  - Payload of the rocket
  - Location of the launch
  - Orbit of the rocket
- We take the historical data and analyze them. It involves data collection, wrangling, analysis and visualization.
- After that, we send them into Machine learning models and prepare a model that predicts it right.
- The prediction is submitted to the end user.

INTRODUCTION

- SpaceX posts the launch and outcomes of its rockets in its site.
- We take the details of the rocket and group it by several types and versions of the rockets.
- We create visualizations for better understanding of the relations between several factors of a launch and the success\failure of a rocket launch.
- We take into account, the orbit, payload and launch site of a launch to predict its outcome.
- Machine learning models are used to predict the outcome of the first launch

METHODOLOGY

- We do the following steps to analyze the data and provide a prediction
- Data Collection
- Data Wrangling
- Exploratory Data Analysis
- Map based on Landing Data
- Dashboard of landing sites and successes
- Machine learning model for prediction

Data Collection

We collect data at this stage using REST API. Below flow chart explains the process.

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

Normalize the data and store it in a data frame

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

Take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.

```
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
```

We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
Using the date we will restrict the dates of the launches

Create a data frame with only the required columns.

GitHub link : https://github.com/tamalada/Spacex-Project/blob/main/Spacex-Data%20Collection.ipynb

Data Collection – Web Scraping

We collect data at this stage using Beautiful Soup package from Wiki.

static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
We save the result in a response object

The response object text is passed to a beautiful soup object.
soup = BeautifulSoup(response.text)

Collect all the tables in the page.
html_tables = soup.find_all('table')

Create a data frame by parsing the launch html tables
launch_dict= dict.fromkeys(column_names)

The resultant dataframe is stored in a csv file.

GitHub link : https://github.com/tamalada/Spacex-Project/blob/main/SpaceX-Data%20collection%20using%20web%20scraping.ipynb

Data Wrangling Methodology

We are going to clean the data. Remove outliers and empty or wrong values.

Load the data set
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)

Identify and calculate the percentage of the missing values in each attribute
df.isnull().sum()/df.count()*100

Identify which columns are numerical and categorical
df.dtypes

Calculate number of launches in each site,orbit and orbit type. Example for launch site,
df['LaunchSite'].value_counts()

Create a outcome column

GitHub url : https://github.com/tamalada/Spacex-Project/blob/main/SpaceX-%20EDA%20and%20Data%20Wrangling.ipynb

EDA with Visualization

- We use scatter plot to find if there is a relation between Flight number and launch site. And it is found that,
  - Most successful launch site is KSC LC 39A.
  - Launch site with most failures is CCAFS SLC 40
- Scatter plot for Payload and launch site is created to find relation.
  - Launch is successful with lower payloads at KSC LC 39A and VAFB SLC 4E
  - Launches are not very successful even with lower payloads in CCAFS SLC 40
- Success rate of each launch site is plotted to find the site with highest success. They are,
  - SSO,HEO,GEO and ES-L1
- Relationship between Flight number and orbit type is plotted in scatter plot.
  - In LEO orbit, success is related to the number of flights.
  - In GTO it is not followed.
- Relationship between payload and orbit type is plotted in scatter plot
  - Heavy payloads have negative influence in GTO orbits and positive on Polar LEO orbits.
- Launch success yearly trend is plotted using line plot.
  - Launch was not very successful in the beginning, but reached its peak in around 2019.

Git Hub link: https://github.com/tamalada/Spacex-Project/blob/main/SpaceX-EDA%20with%20Visualization.ipynb

EDA using SQL

- Found the number of unique launch sites in the mission.
- Created search query to find 5 records where launch sites begin with string “CCA”.
- Found total payload mass carried by a customer. Here, NASA.
- Found average payload mass carried by booster version F9 v1.1.
- Found the date when first successful landing outcome in ground pad was achieved.
- Found the names of boosters that succeeded in lading with payload mass between 4000 and 6000
- Listed the total number of success and failure outcomes.
- Found the booster versions that carried maximum payload mass.
- Listed failed outcomes in drone ship.
- Ranked the landing outcomes between a particular date interval, here, 2010-06-04 and 2017-03-20.

All the above can be used as several search queries in the data model in future versions. For now, used for data analysis.

GitHub Link: https://github.com/tamalada/Spacex-Project/blob/main/SpaceX-EDA%20with%20SQL.ipynb

Interactive visual analytics

- Folium map is used to plot the various launch sites in world map.
- The success and failures are plotted respectively. Graphical representation gives better understanding.
- Also distance between a launch site and coastline, rail road, highway and city respectively were plotted.
- It was found that a launch site is close to a rail road and highway but far from a city and comparatively closer to a coastline.
- Plotly dash dashboard was created with the given data. A pie-chart for the launch sites and based on the launch site, the success of launch with respect to payload was plotted in a scatter plot.

GitHub link Folium map: https://github.com/tamalada/Spacex-Project/blob/main/SpaceX%20-%20Launch%20site%20location.ipynb

GitHub link Plotly dash app : https://github.com/tamalada/Spacex-Project/blob/main/SpaceX%20project%20Dashboard.py

Predictive Analysis Methodology

Load the data frame from source
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")

Create a label from the data set. Target variable.
Y = data["Class"].to_numpy()

Normalize the data set and assign it to X for prediction
X = preprocessing.StandardScaler().fit(X).transform(X)

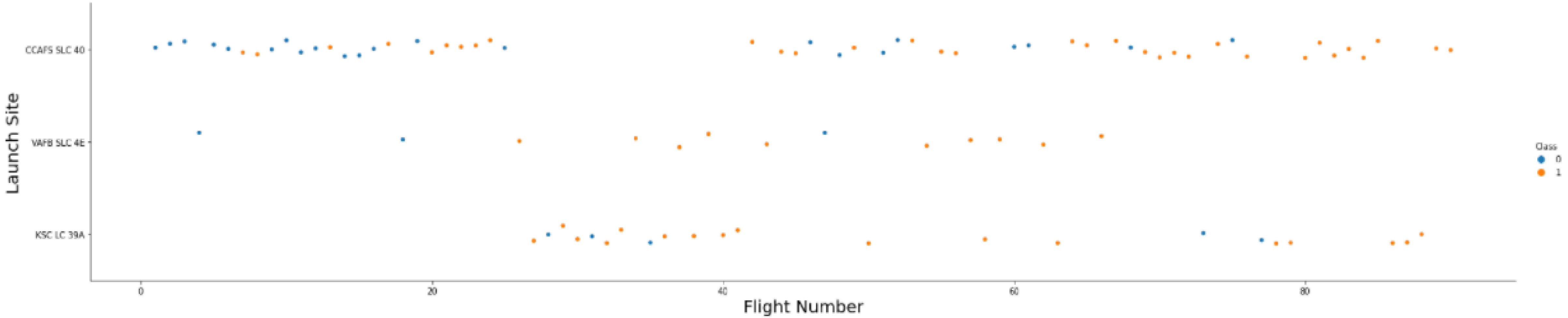
Split training and test data set
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

Run Logistic regression,SVM,Decision tree, KNN on the data and find the predictions

Score the predictions using score(), Jaccard index and F1 score methods.

GitHub Link: https://github.com/tamalada/Spacex-Project/blob/main/SpaceX%20project%20Prediction%20ML.ipynb

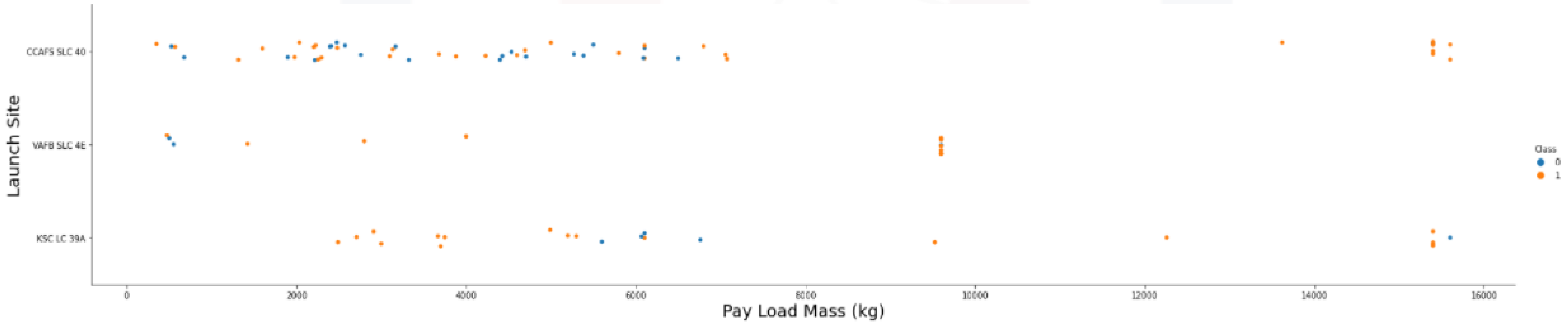
EDA Visualization Results - 1



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

Explanation: In CCAFS SLC 40 has most failures, and the most successful being KSC LC 39A based on the above plot.

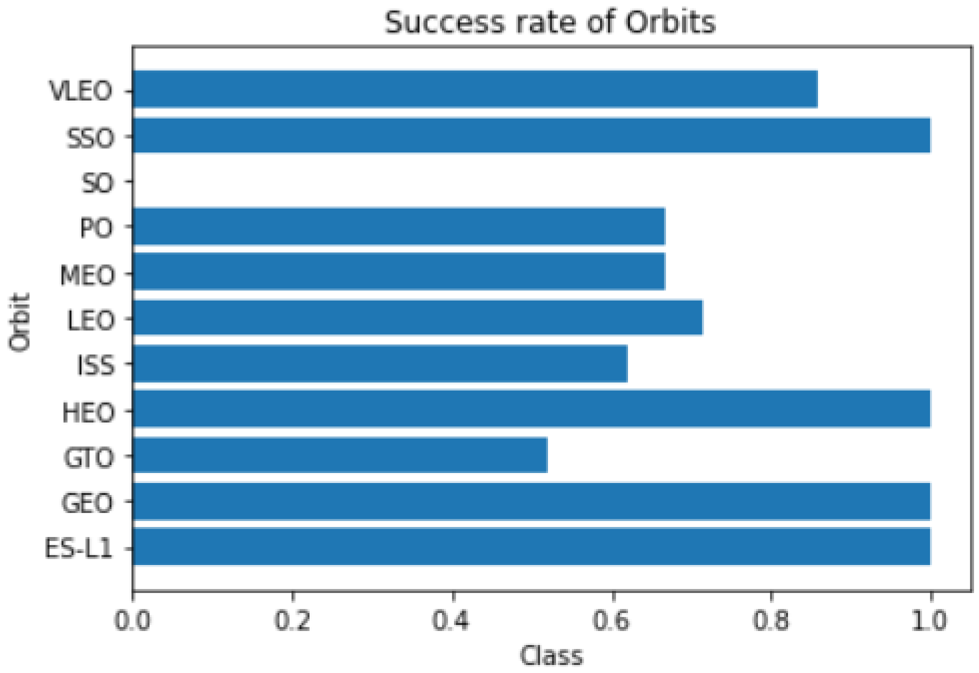
EDA Visualization Results - 2



Now try to explain any patterns you found in the Payload Vs. Launch Site scatter point chart.

Explanation: Smaller payload with KSC LC 39A has higher success outcomes.

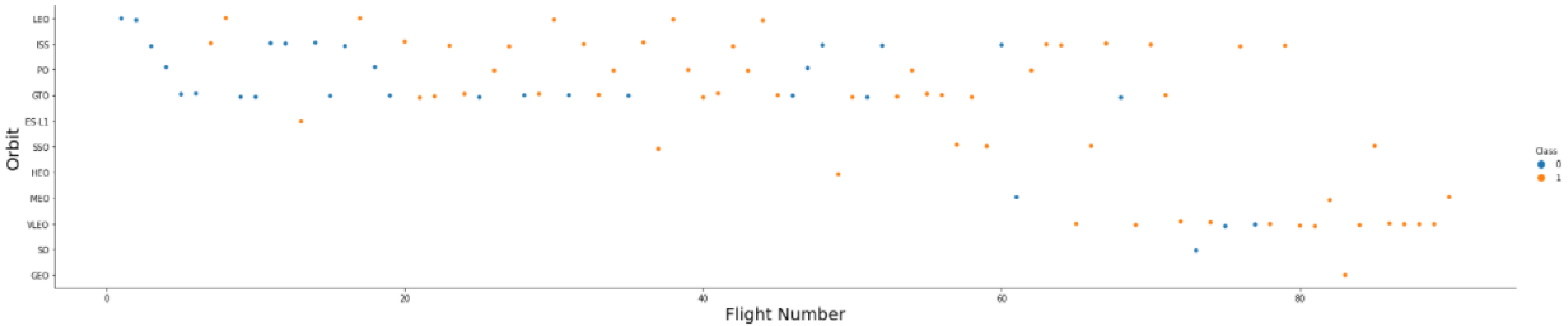
EDA Visualization Results - 3



Analyze the ploted bar chart try to find which orbits have high sucess rate.

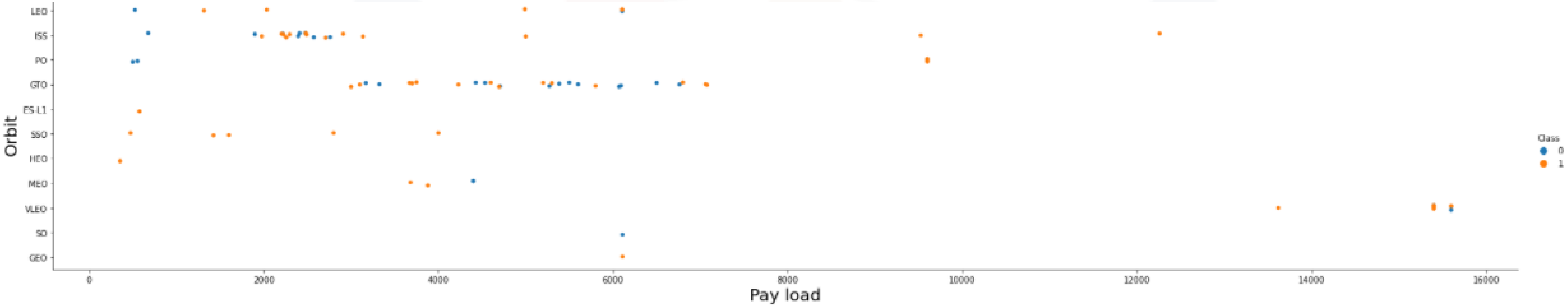
Answer : As the above chart shows, the orbits with high success rate are, SSO,HEO,GEO and ES-L1

EDA Visualization Results - 4



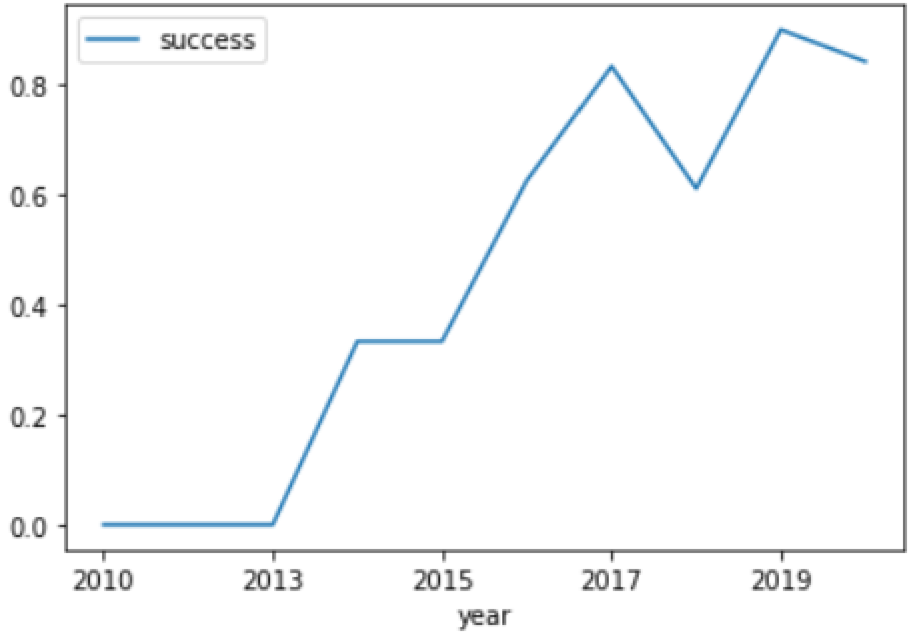
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

EDA Visualization Results - 5



Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

EDA Visualization Results - 6



Success rate is on a increasing trend from 2013 to 2020

EDA with SQL Results - 1

Task 1

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

In [7]:
```
%sql select distinct launch_site from spacextbl
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[7]:

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

EDA with SQL Results - 2

Task 2

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

In [10]:
```
%sql select * from spacextbl where launch_site like '%CCA%' limit 5
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[10]:

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

EDA with SQL Results - 3

Task 3

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

In [12]:
```
%sql select sum(payload_mass__kg_) as Total_Payload_Mass from spacextbl where customer='NASA (CRS)'
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[12]:

total_payload_mass
45596

EDA with SQL Results - 4

Task 4

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

In [16]:
```
%sql select AVG(payload_mass__kg_) as average_payload_mass from spacextbl where booster_version like '%F9 v1.1%'
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[16]:

average_payload_mass
2534

EDA with SQL Results - 5

Task 5

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

Hint:Use min function

In [22]:
```
%sql select MIN(YEAR(DATE)) as first_successful_landing from spacextbl where mission_outcome='Success'
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[22]:

first_successful_landing
2010

EDA with SQL Results - 6

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

In [28]:
```
%sql select * from spacextbl where landing__outcome='Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[28]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

EDA with SQL Results - 7

Task 7

List the total number of successful and failure mission outcomes

In [34]:
```
%sql select mission_outcome,count(mission_outcome) as count from spacextbl group by mission_outcome
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[34]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

EDA with SQL Results - 8

Task 8

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

In [36]:
```
%sql select booster_version,payload_mass__kg_ from spacextbl where payload_mass__kg_=(select MAX(payload_mass__kg_) from spacext
bl)
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[36]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

EDA with SQL Results - 9

Task 9

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

In [43]:
```
%sql select booster_version,launch_site,landing__outcome from spacextbl where landing__outcome='Failure (drone ship)' and YEAR(D
ATE)=2015
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[43]:

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

EDA with SQL Results - 10

Task 10

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

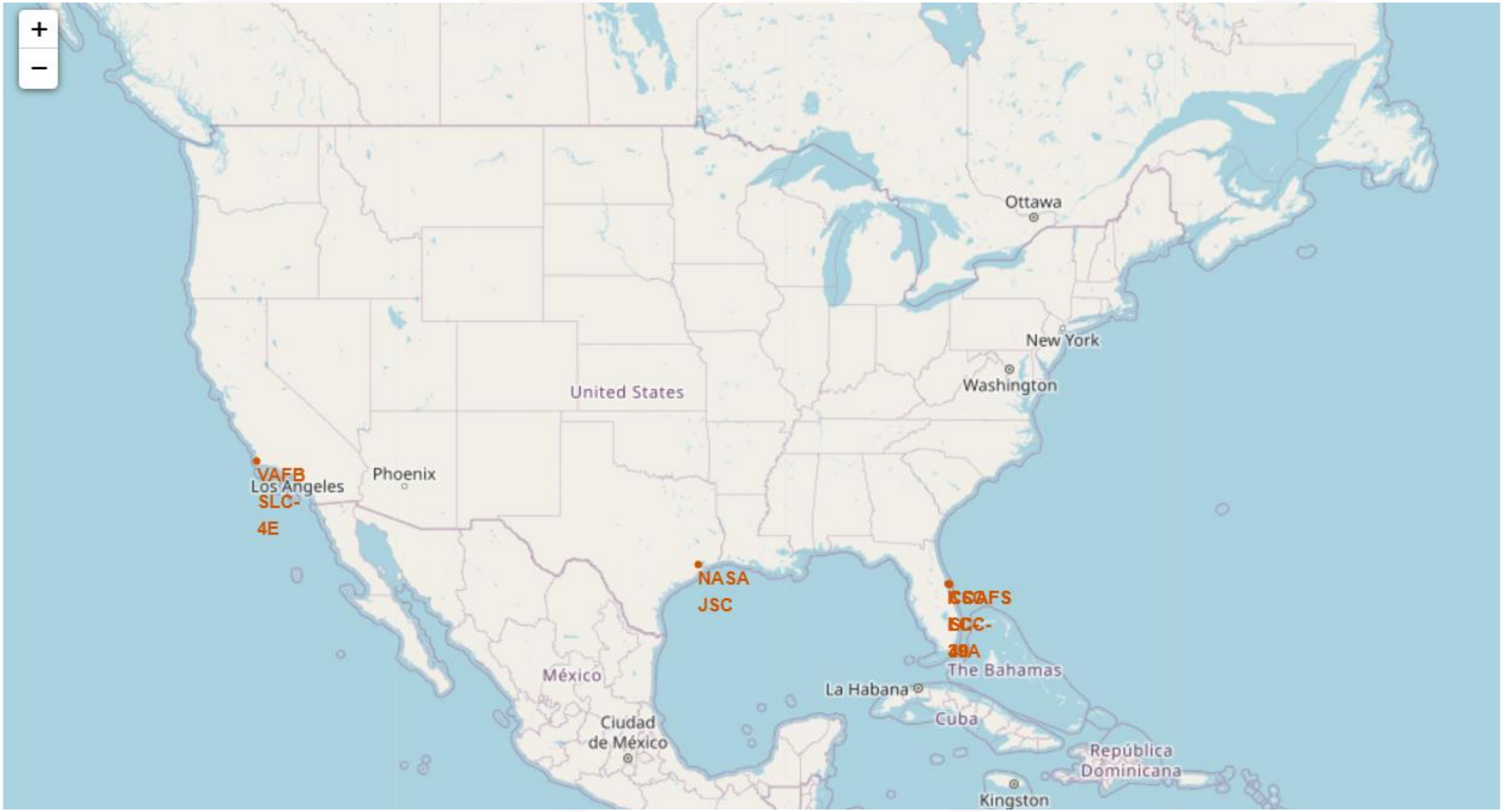
In [52]:
```
%sql select landing__outcome,count(landing__outcome) as count_landing_outcome from spacextbl where DATE between '2010-06-04' and
'2017-03-20' group by landing__outcome  order by count_landing_outcome desc
```

```
 * ibm_db_sa://qqd48118:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[52]:

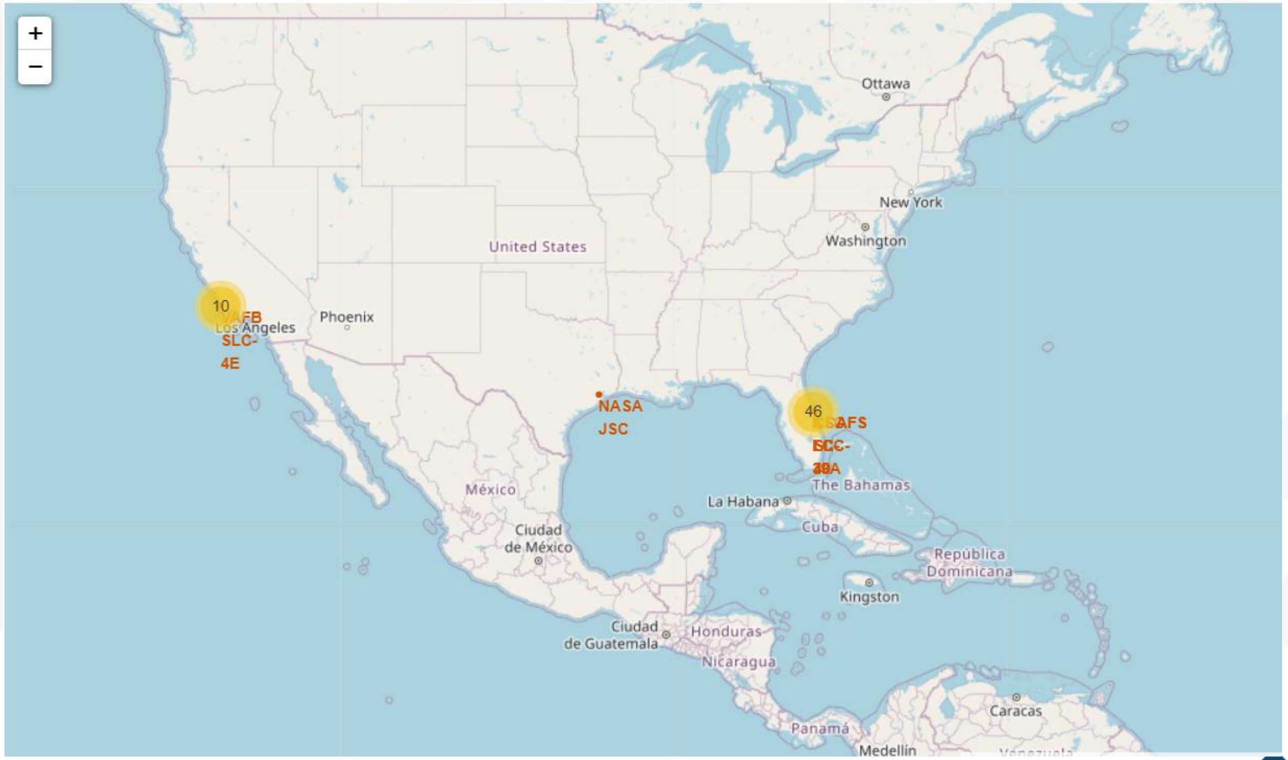
landing__outcome	count_landing_outcome
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Map with Folium - 1



Map with Folium – 2.1

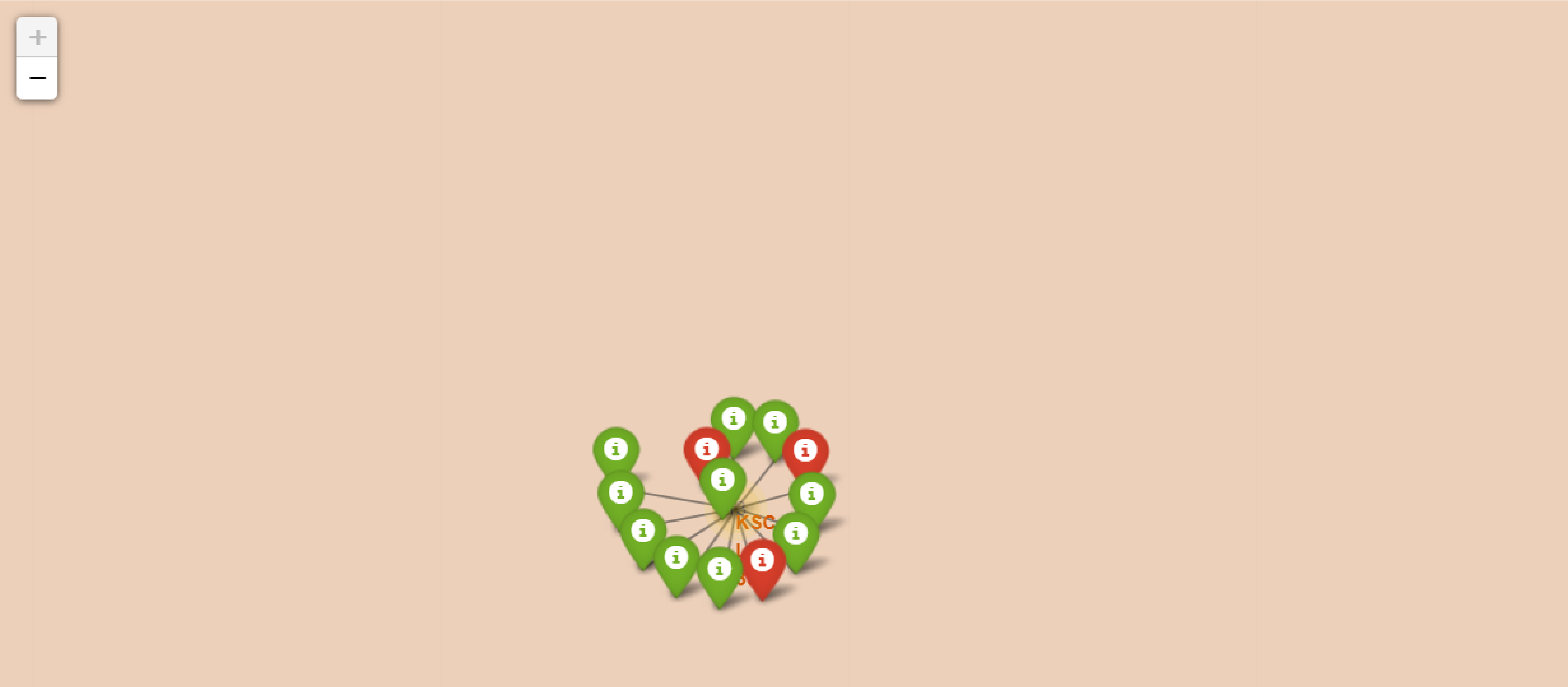
The launch sites are marked in a world map. The number of launches are displayed on the site. For detailed view, it needs to be zoomed in and clicked on.



Map with Folium – 2.2

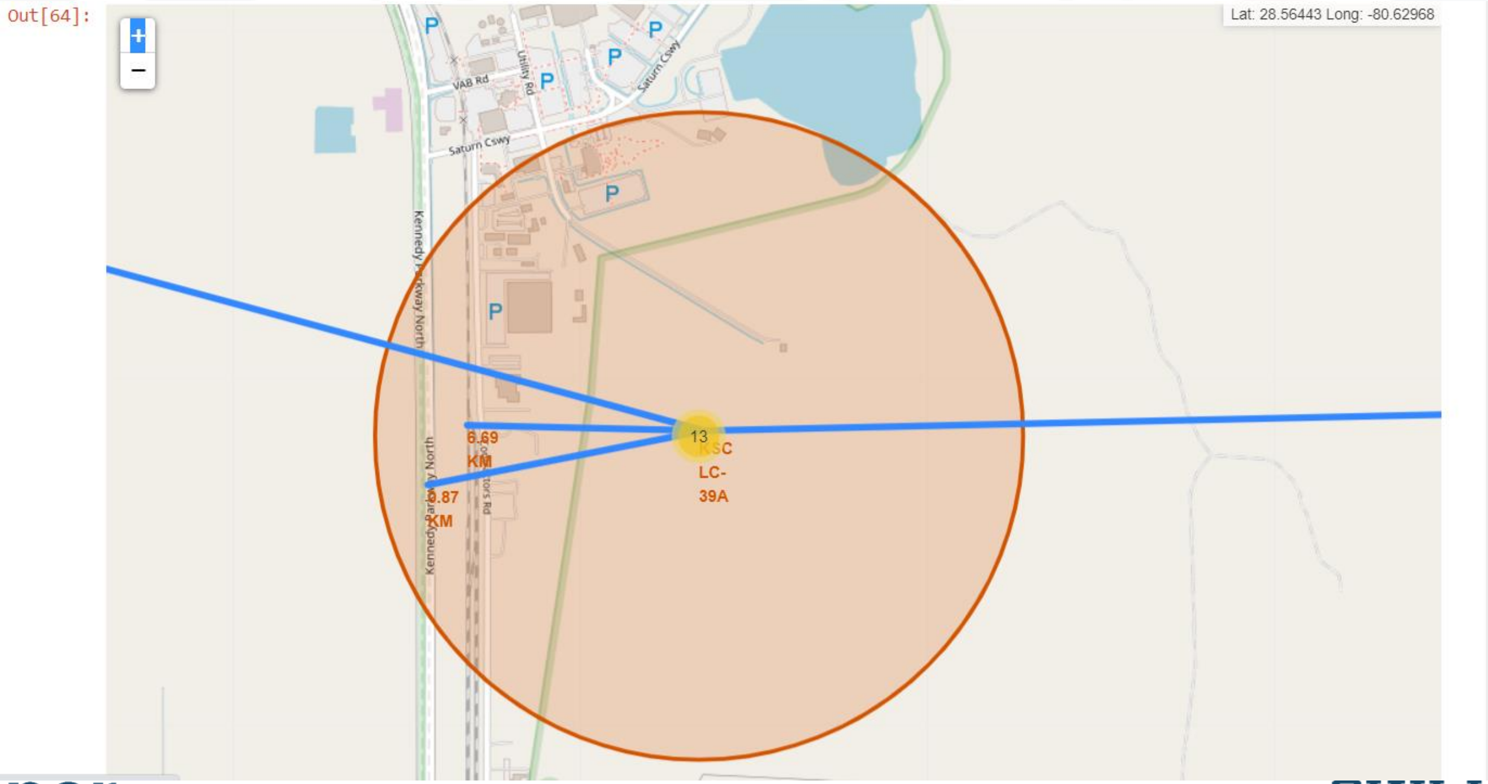
For example, in the site KSC LC 39A, the successful and failed attempts are marked as below.

Out[55]:



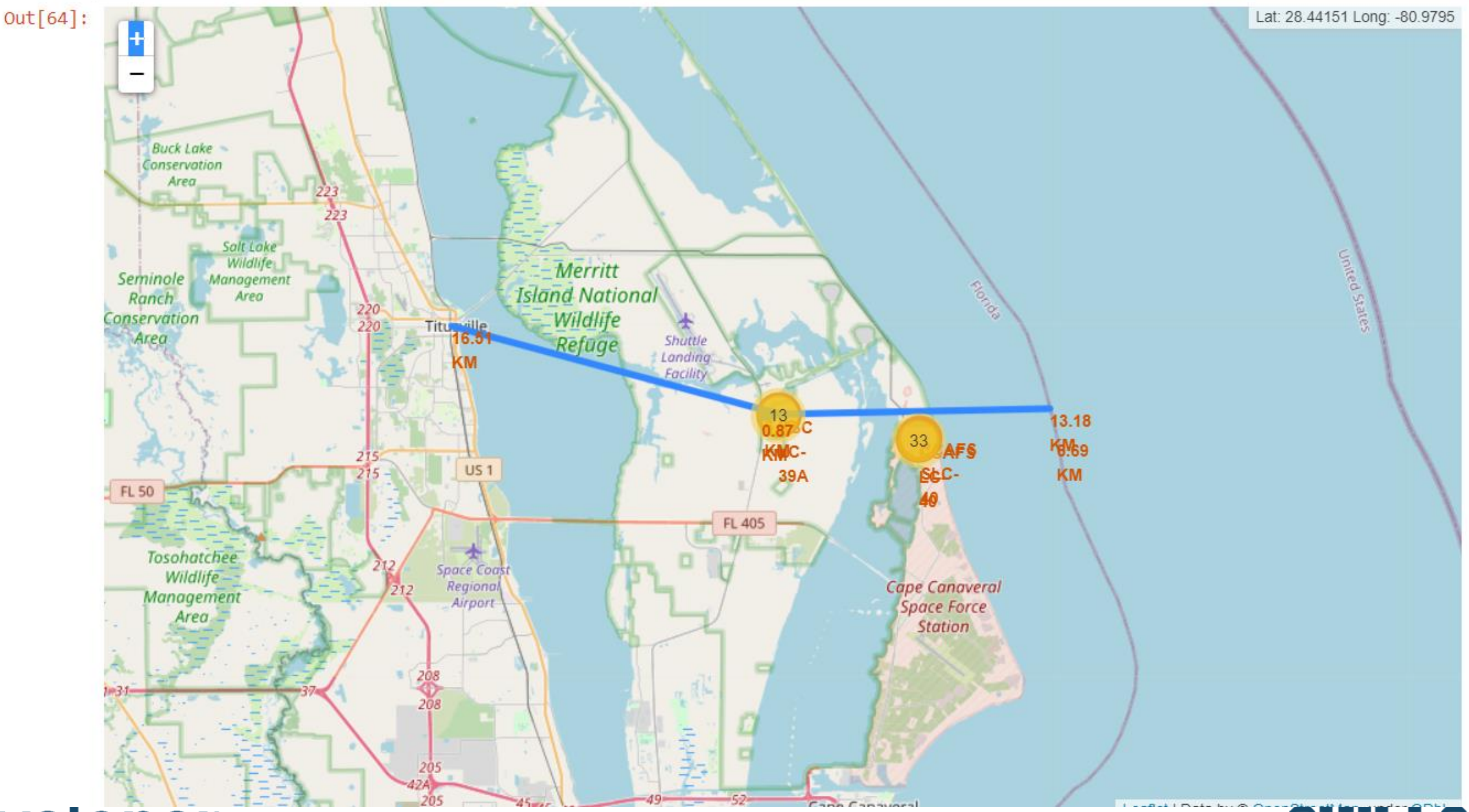
Map with Folium – 3.1

This map is to explain the proximity of locations. The launch site is closer to highway and railway as given in below screenshot.



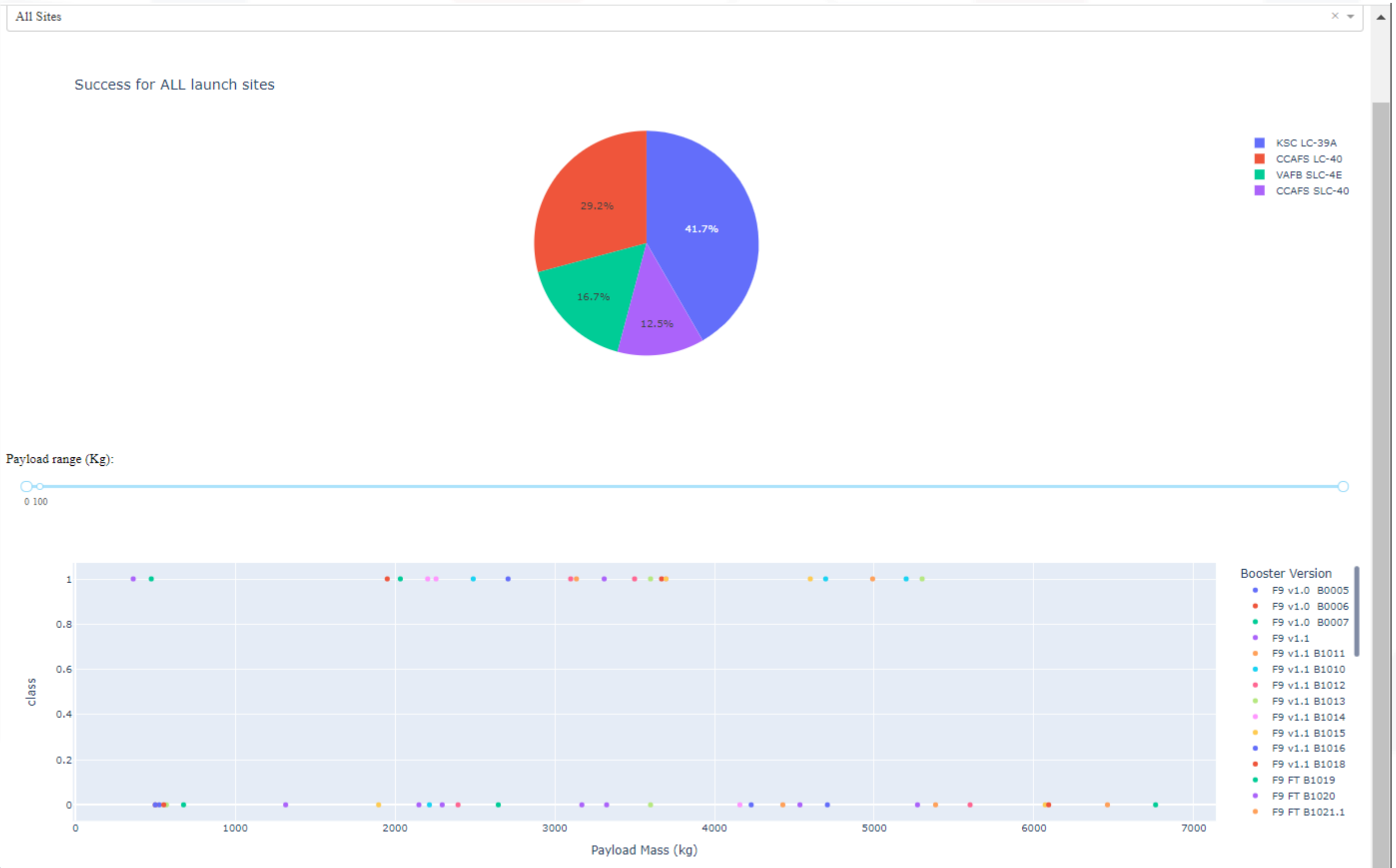
Map with Folium – 3.2

And the launch site is far from city and closer to coastlines. In case of a bad landing. As shown in below screenshot



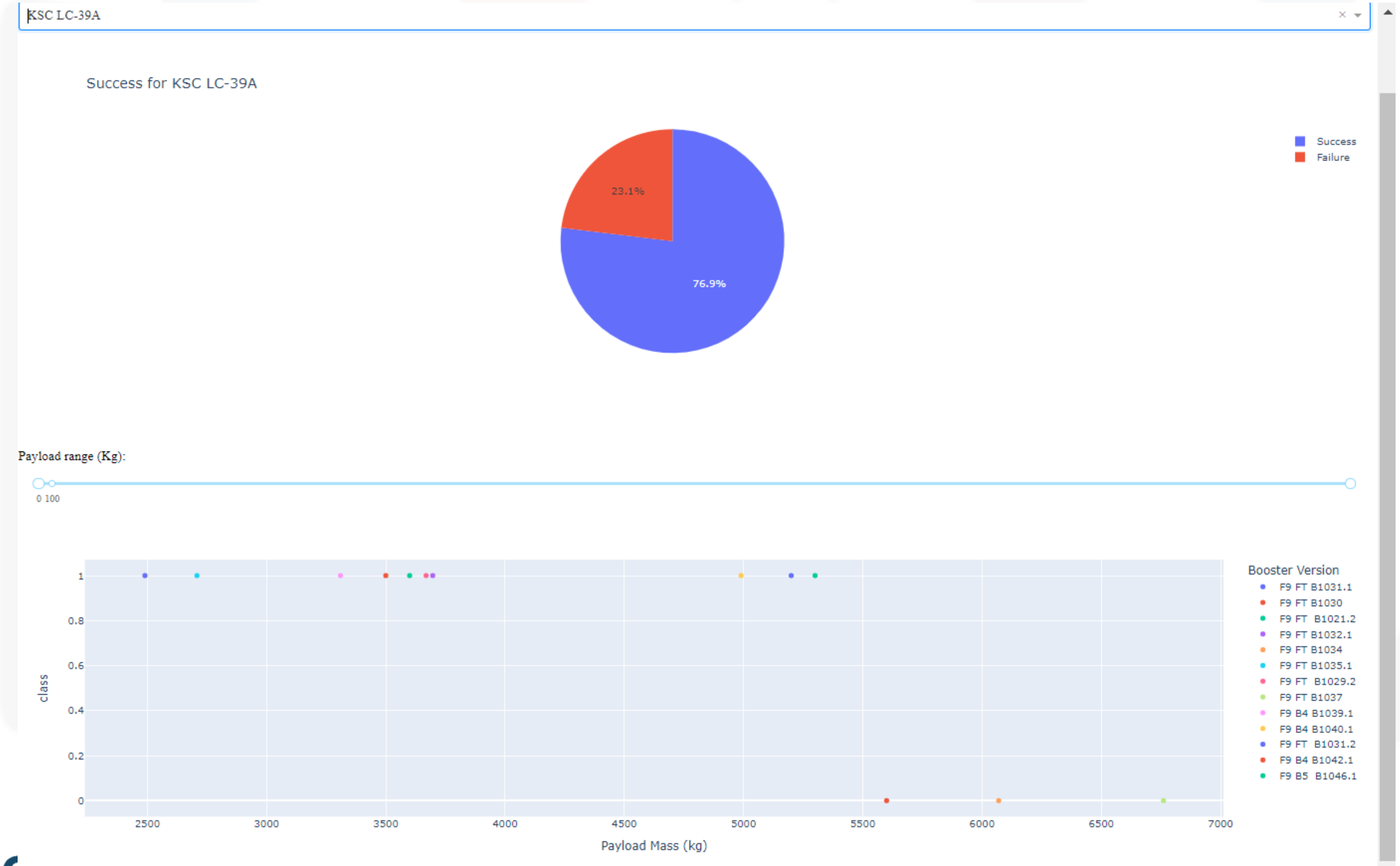
Interactive Dashboard-1

As shown, the highest success is for KSC LC 39A site.



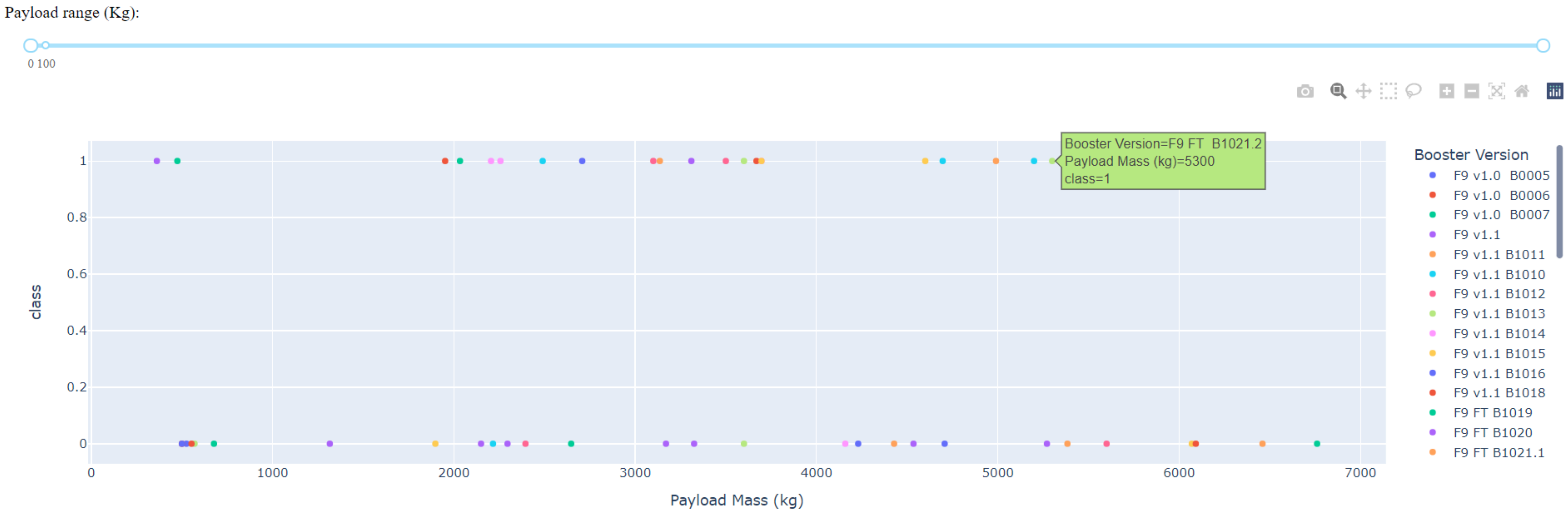
Interactive Dashboard-2

Success percentage is 76.9% and maximum possible payload is 5300kg.



Interactive Dashboard-3

The booster version that took that max. payload is F9 FT B1021.2. Combining this and the above two will give higher possibility for success.



KNN Result

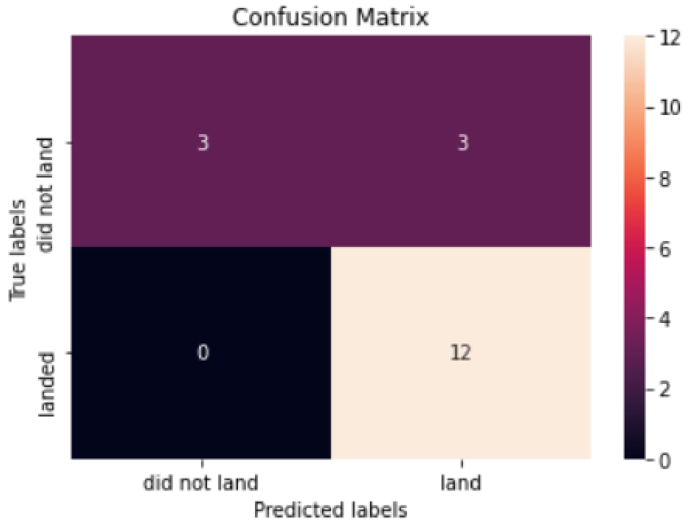
Calculate the accuracy of knn_cv on the test data using the method `score`:

In [120]:
```
knn_cv.score(X_test,Y_test)
```

Out[120]: 0.8333333333333334

We can plot the confusion matrix

In [121]:
```
knn_yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,knn_yhat)
```



Logistic Regression Result

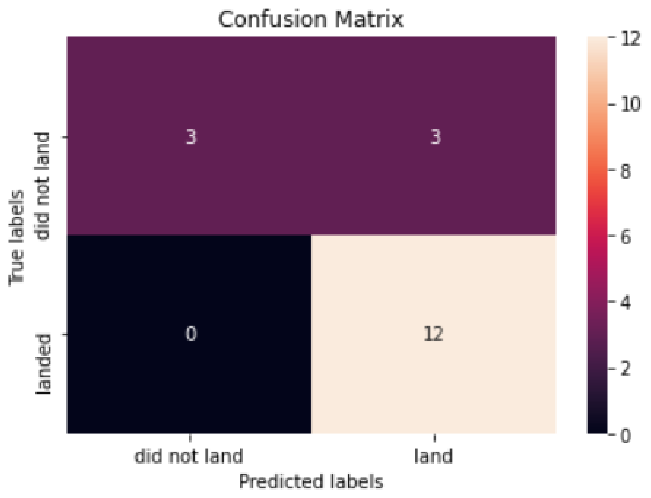
Calculate the accuracy on the test data using the method `score`:

```
In [105]: print("The score is ",logreg_cv.score(X_test,Y_test))
```

```
The score is  0.8333333333333334
```

Lets look at the confusion matrix:

```
In [106]: logreg_yhat=logreg_cv.predict(X_test)
          plot_confusion_matrix(Y_test,logreg_yhat)
```



Decision Tree Result

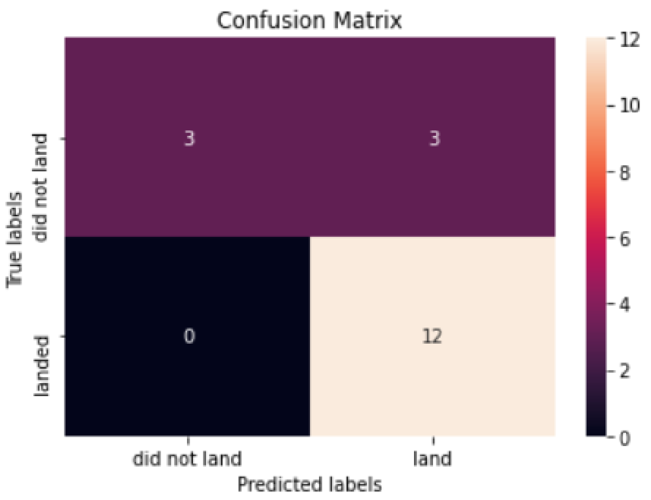
Calculate the accuracy of tree_cv on the test data using the method `score`:

In [115]:
```
tree_cv.score(X_test,Y_test)
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
accuracy : 0.875
```

We can plot the confusion matrix

In [116]:
```
tree_yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,tree_yhat)
```



SVM Result

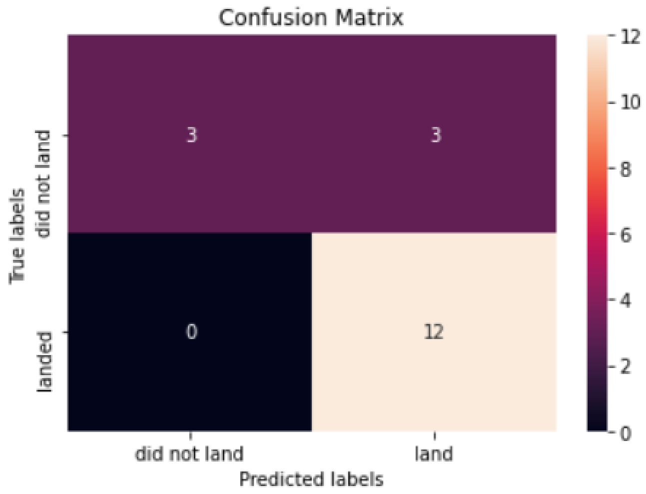
Calculate the accuracy on the test data using the method `score`:

In [110]:
```
print("svm score ",svm_cv.score(X_test,Y_test))
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
svm score  0.8333333333333334
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

We can plot the confusion matrix

In [111]:
```
svm_yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,svm_yhat)
```



Result Analysis

- Based on the below screenshot, all models perform the same. But, under score() method, decision tree scored more than the others on testing with test set(accuracy = 0.87) but others were less.
- So the recommended method is Decision Tree.

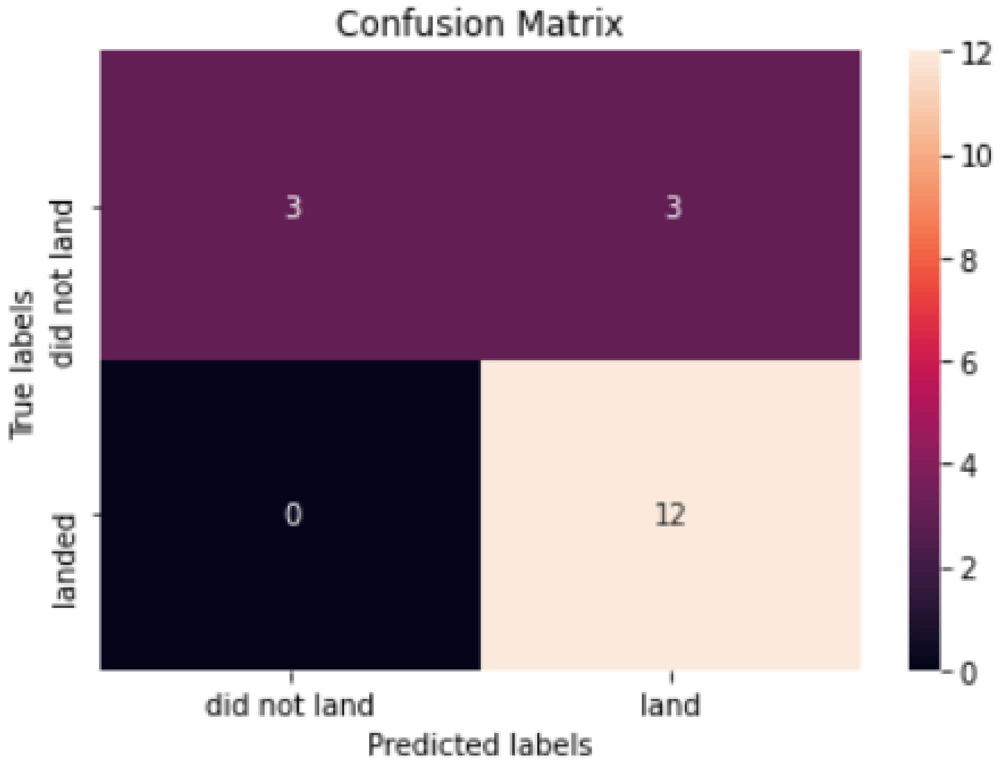
Find the method performs best:

In [122]:
```
from sklearn.metrics import jaccard_score
from sklearn.metrics import f1_score

print("F1 score for KNN : ",f1_score(Y_test, knn_yhat, average='weighted'))

print("F1 score for Decision Tree : ",f1_score(Y_test, tree_yhat, average='weighted'))

print("F1 score for SVM : ",f1_score(Y_test, svm_yhat, average='weighted'))

print("F1 score for LR : ",f1_score(Y_test, logreg_yhat, average='weighted'))


print("Jaccard index for KNN : ",metrics.jaccard_score(Y_test, knn_yhat,average='weighted'))
print("Jaccard index for Decision Tree : ",metrics.jaccard_score(Y_test, tree_yhat,average='weighted'))
print("Jaccard index for SVM : ",metrics.jaccard_score(Y_test, svm_yhat,average='weighted'))
print("Jaccard index for LR : ",metrics.jaccard_score(Y_test, logreg_yhat,average='weighted'))
```

```
F1 score for KNN :  0.8148148148148149
F1 score for Decision Tree :  0.8148148148148149
F1 score for SVM :  0.8148148148148149
F1 score for LR :  0.8148148148148149
Jaccard index for KNN :  0.7000000000000001
Jaccard index for Decision Tree :  0.7000000000000001
Jaccard index for SVM :  0.7000000000000001
Jaccard index for LR :  0.7000000000000001
```

Decision Tree’s confusion matrix

- The issue of false positives still remains, but it still could predict the positive landings correctly.
- Out of 18 attempts 15 were predicted correctly. So this could be used for prediction.



CONCLUSION

- EDA visualization recommends that, SSO,HEO,GEO and ES-L1 have better possibility of success
- EDA with SQL gives better searching capabilities that could be used.
- Interactive Dashboard recommends that using KSC LC 39A launch site with payload less than 5300kg gives a higher possibility of success.
- Classification model can be used to predict outcome with more combinations of the above given factors like orbit, payload, launch site and booster version.

APPENDIX

- https://github.com/tamalada/Spacex-Project