SpaceX Capstone Project

Fabien Mahieux

15-Aug-23

OUTLINE

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

EXECUTIVE SUMMARY

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive analysis (classification
- Summary of results
  - EDA results
  - Interactive analysis
  - Predictive analysis

INTRODUCTION

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62m; other providers cost upward of $165m each, much of the savings is due to the fact that SpaceX can reuse the first stage

- Project tasks

Predict if the first stage of the SpaceX Falcon 9 rocket will land successfully

METHODOLOGY

- Data collection methodology:
  - SpaceX REST API
  - Web-scrapping from Wikipedia
- Data wrangling
- Exploratory Data Analysis using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models

Data Collection

SpaceX REST API

- Request and Parse SpaceX launch data using GET request
- Filter the dataframe to only include Falcon 9 launches

https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Wikipedia Webscrapping (BeautifulSoup)

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML header
- Creating a data frame by parsing the launch HTML tables

https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Create a landing outcome label

https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Visualization

- Flight Number vs Payload Mass (Kg)
- Flight Number vs Launch Site
- Payload Mass (Kg) vs Launch Site
- Payload Mass (Kg) vs Orbit type

https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- List of the date when the first succesful landing outcome in ground pad was acheived
- List of the names of the boosters which have success in drone ship and have payload mass between 4000 and 6000
- List of the total number of successful and failure mission outcomes
- List of the names of the booster versions which have carried the maximum payload mass.
- List of the records displaying month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20 in descending order.

https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Interactive Visual Analytics with Folium

- Mark all launche sites on a map

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

- Calculate the distances between launch site to its proximities

- https://github.com/FabienMHX/IBM-Data-Science-Capstone-Project/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Dashboard with Plotly Dash

- Dropdown list with Launch Sites
- Pie chart showing successful launches
- Slider of Payload Mass Range
- Scatter Chart showing Payload Mass vs Success Rate by Booster version

Predictive Analysis

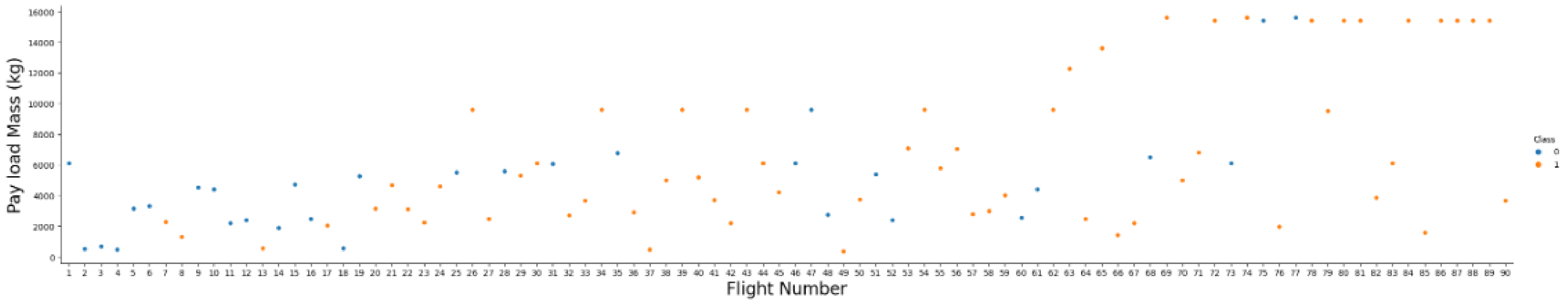
- Create a NumPy array from the Class column
- Standardize the data with StandardScaler, Fit and transform the data
- Split the data using train_test_split
- Create a GridSearchCV object
- Calculate the accuracy on the test data using score() method
- Assess the confusion matrix for all models
- Identify the best model using Jaccard_Score, F1_Score and accuracy

RESULTS SUMMARY

Exploratory Data Analysis

RESULTS - Visualization

Filght Number vs Payload Mass (Kg)

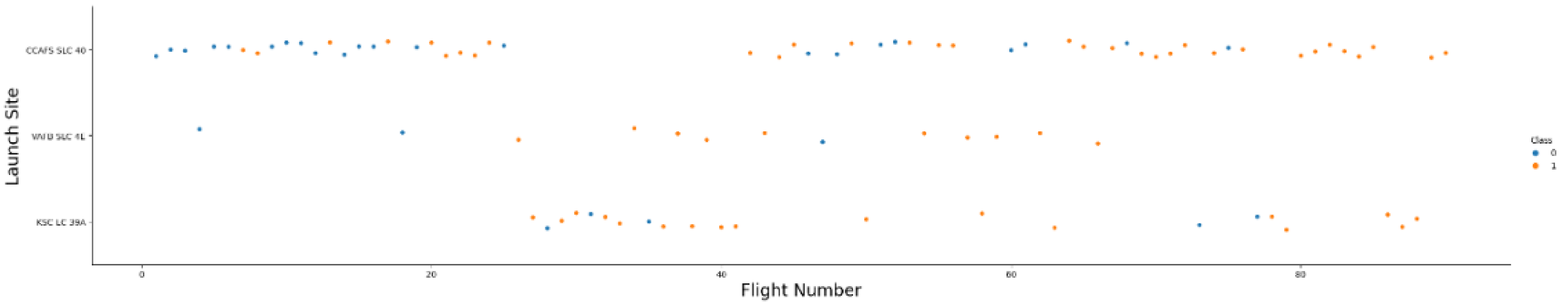


We see that as the flight number increases, the first stage is more likely to land successfully.

The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

RESULTS - Visualization

Filght Number vs Launch Site

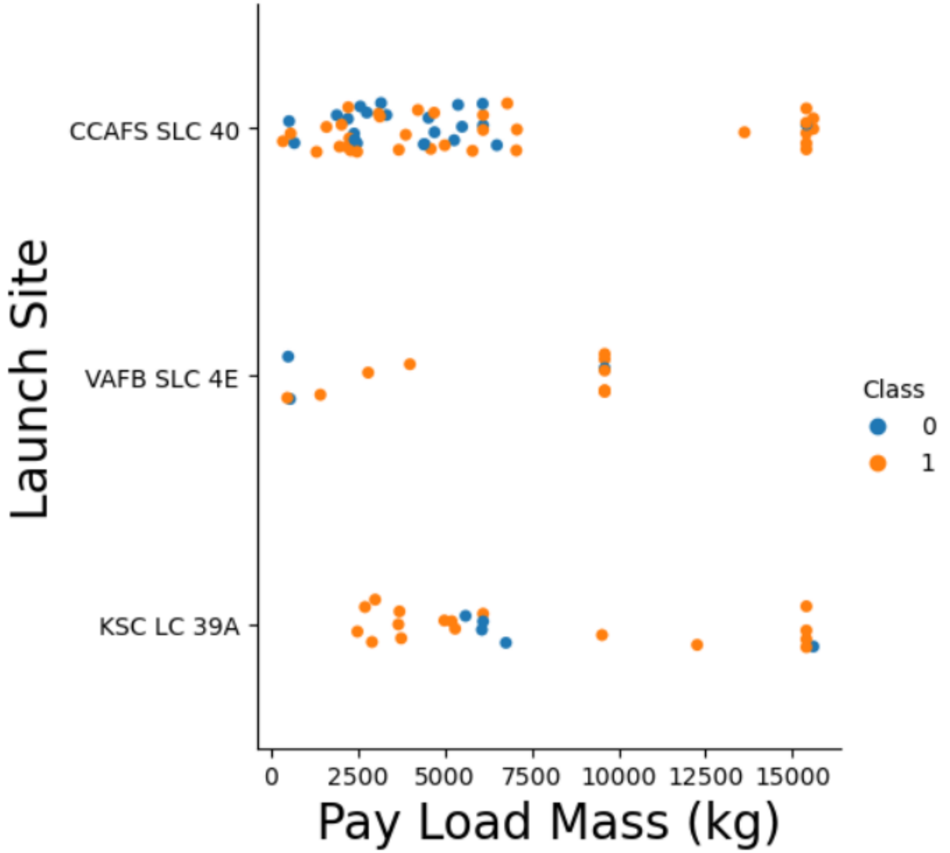


We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

RESULTS - Visualization

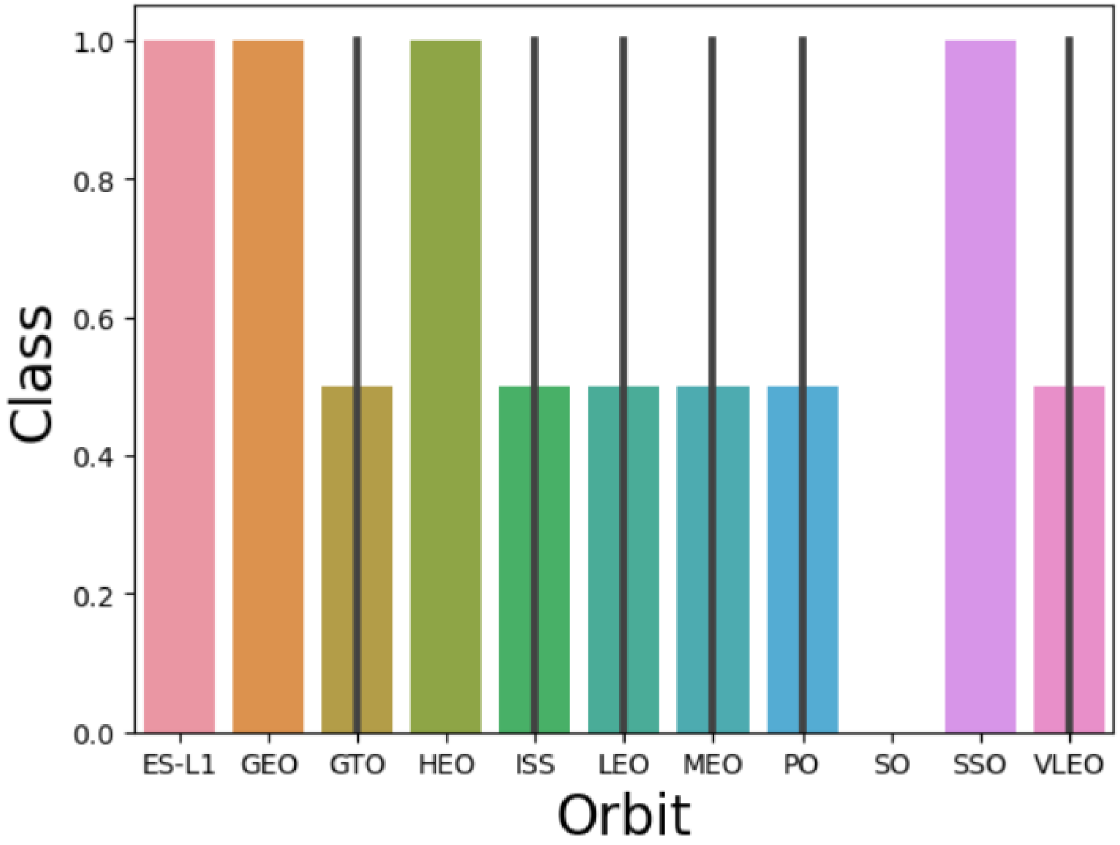
Launch Site vs Payload Mass (Kg)

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



RESULTS - Visualization

Success rate of each Orbit type



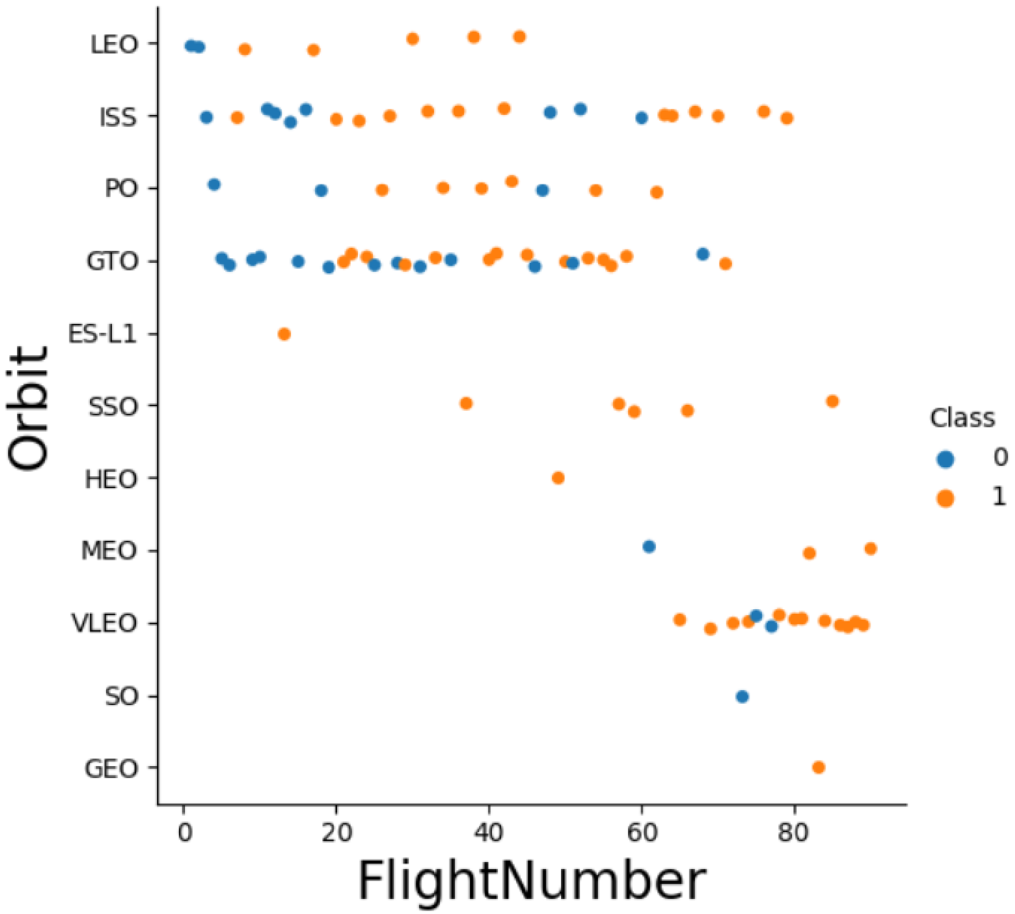
100% Success rate : ES-L1, GEO, HEO, SSO

50% Success rate : GTA, ISS, LEO, MEO, PO, VLEO

0% Success rate : SO

RESULTS - Visualization

Flight number vs Orbit



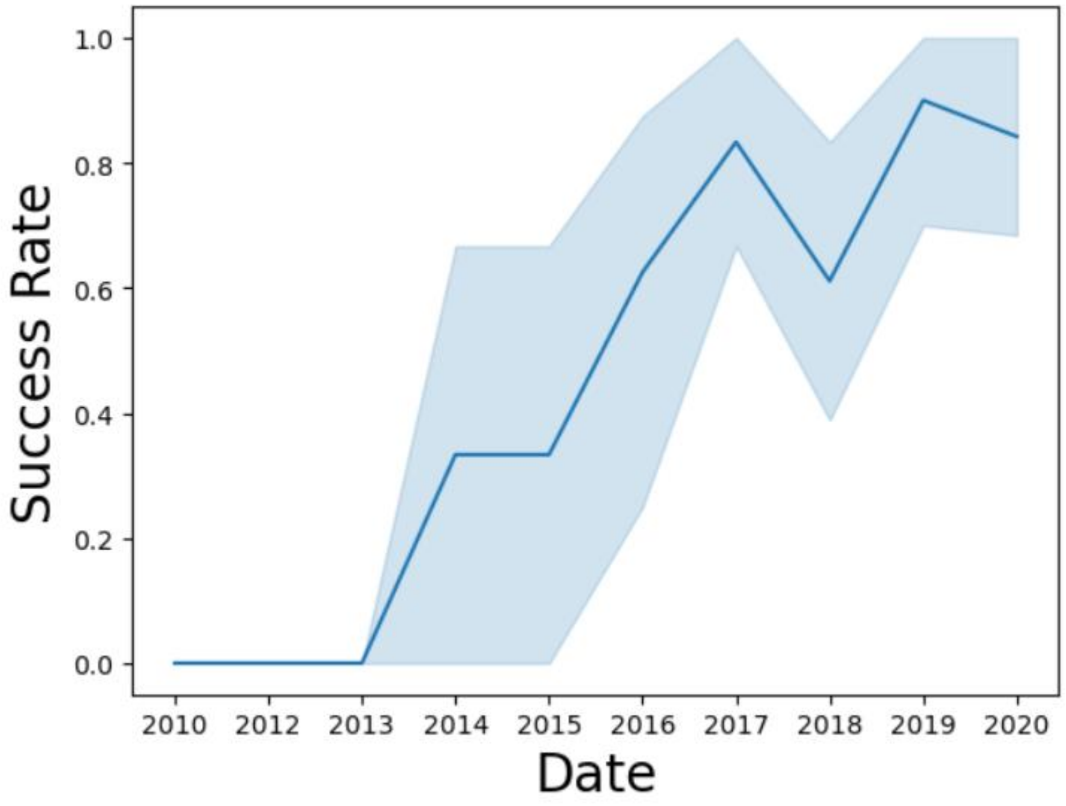
We can see that in the LEO orbit the Success appears related to the number of flights;

On the other hand, there seems to be no relationship between flight number when in GTO orbit.

RESULTS - Visualization

Launch Success over time

We can observe that the success rate since 2013 kept increasing until 2022



RESULTS - SQL

Unique Launch sites

```
In [9]: %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL

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

Out[9]:

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

RESULTS - SQL

5 records where launch sites begin with 'CCA'

Out[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

RESULTS - SQL

Total payload mass carried by boosters launched by NASA (CRS)

In [11]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

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

Out[11]:

SUM(PAYLOAD_MASS__KG_)
45596

Average payload mass carried by boosters version F9 v1.1

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

In [12]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

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

Out[12]:

AVG(PAYLOAD_MASS__KG_)
2928.4

RESULTS - SQL

Date when the first successful landing outcome in group pad was achieved

In [14]:
```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)'
```

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

Out[14]:

MIN(DATE)
2015-12-22

Name of the boosters which have success in drone ship and have payload mass between 4000 and 6000

In [16]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING_OUTCOME='Success (drone
```

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

Out[16]:

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

RESULTS - SQL

Total number of successful and failure mission ourcomes

In [17]:
```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

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

Out[17]:

COUNT(*)
101

Name of the booster versions which have carried the maximum payload mass

In [18]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Out[18]:

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

RESULTS - SQL

List of records displaying the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.

In [64]:
```
%sql SELECT DATE,substr(Date, 6, 2) AS MONTH, substr(Date, 1, 4) AS YEAR, BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME \
FROM SPACEXTBL \
WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND YEAR = '2015';
```

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

Out[64]:

Date	MONTH	YEAR	Booster_Version	Launch_Site	Landing_Outcome
2015-10-01	10	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

In [27]:
```
sql SELECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDINC
```

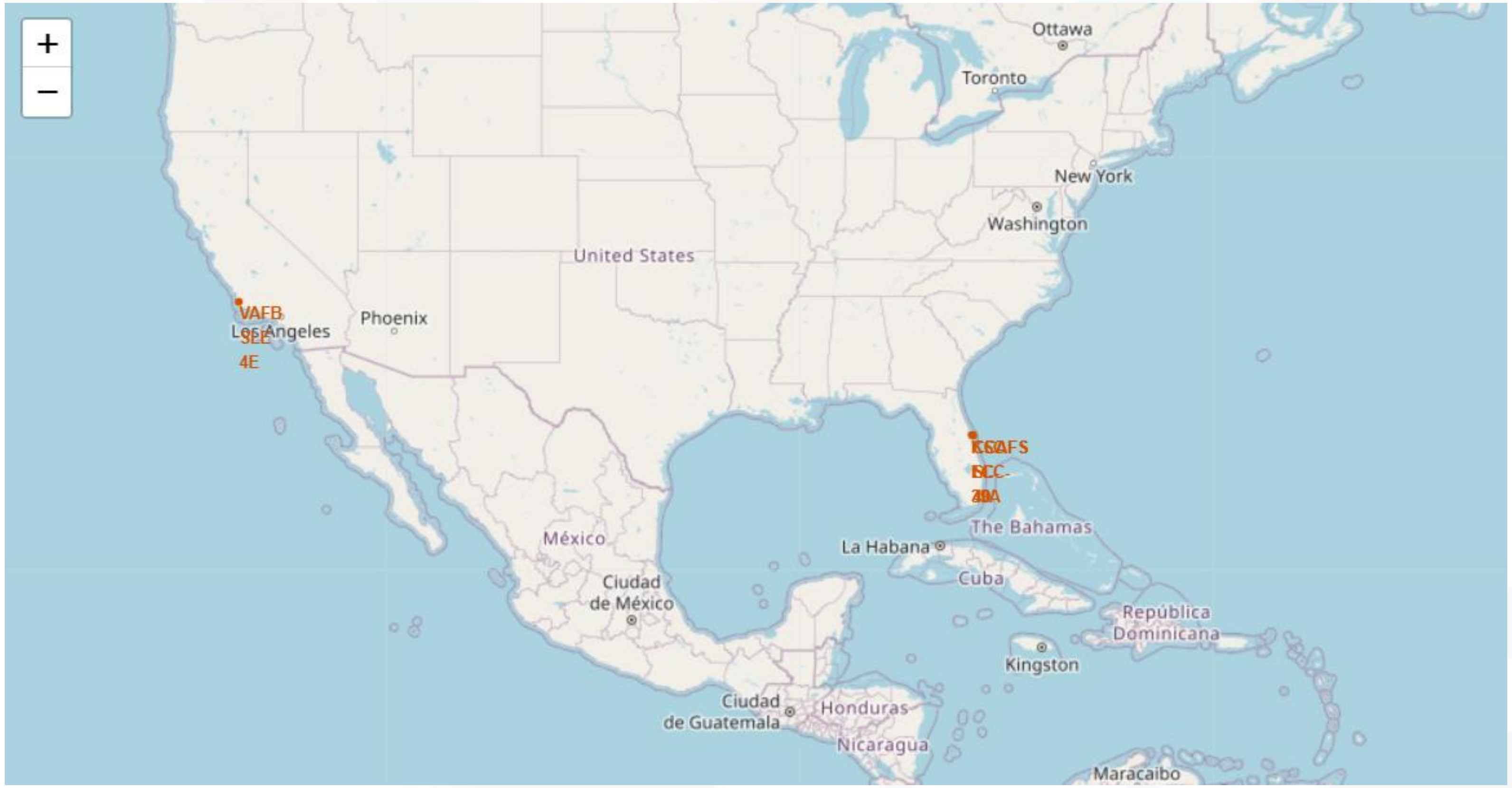
```
 * sqlite:///my_data1.db
Done.
```

Out[27]:

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

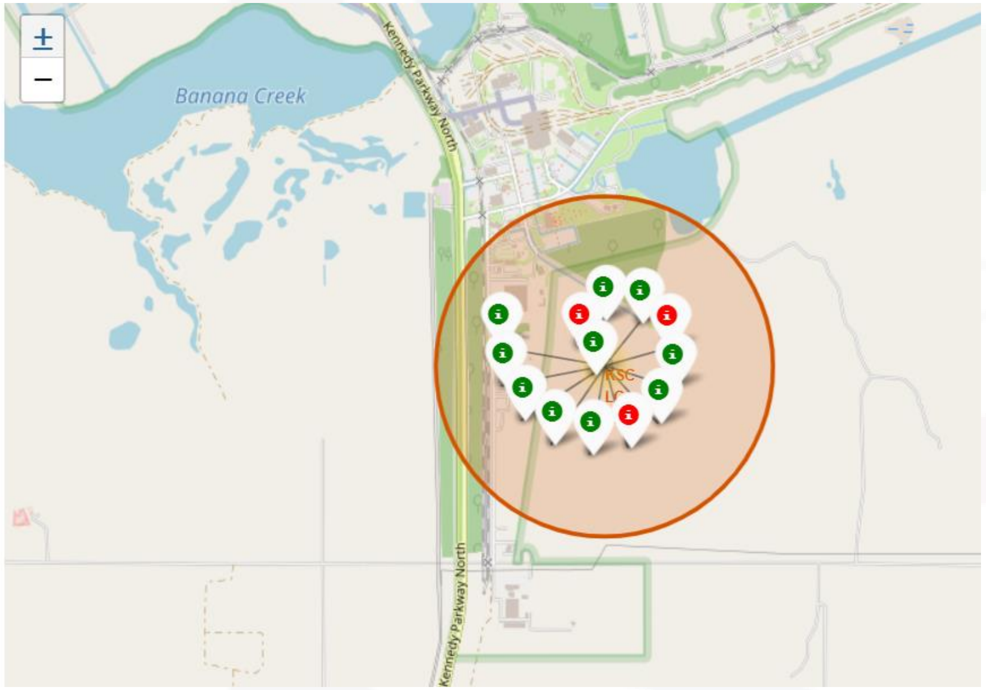
RESULTS – Interactive Visual Analytics with Folium Lab

Mark all lunch sites on map



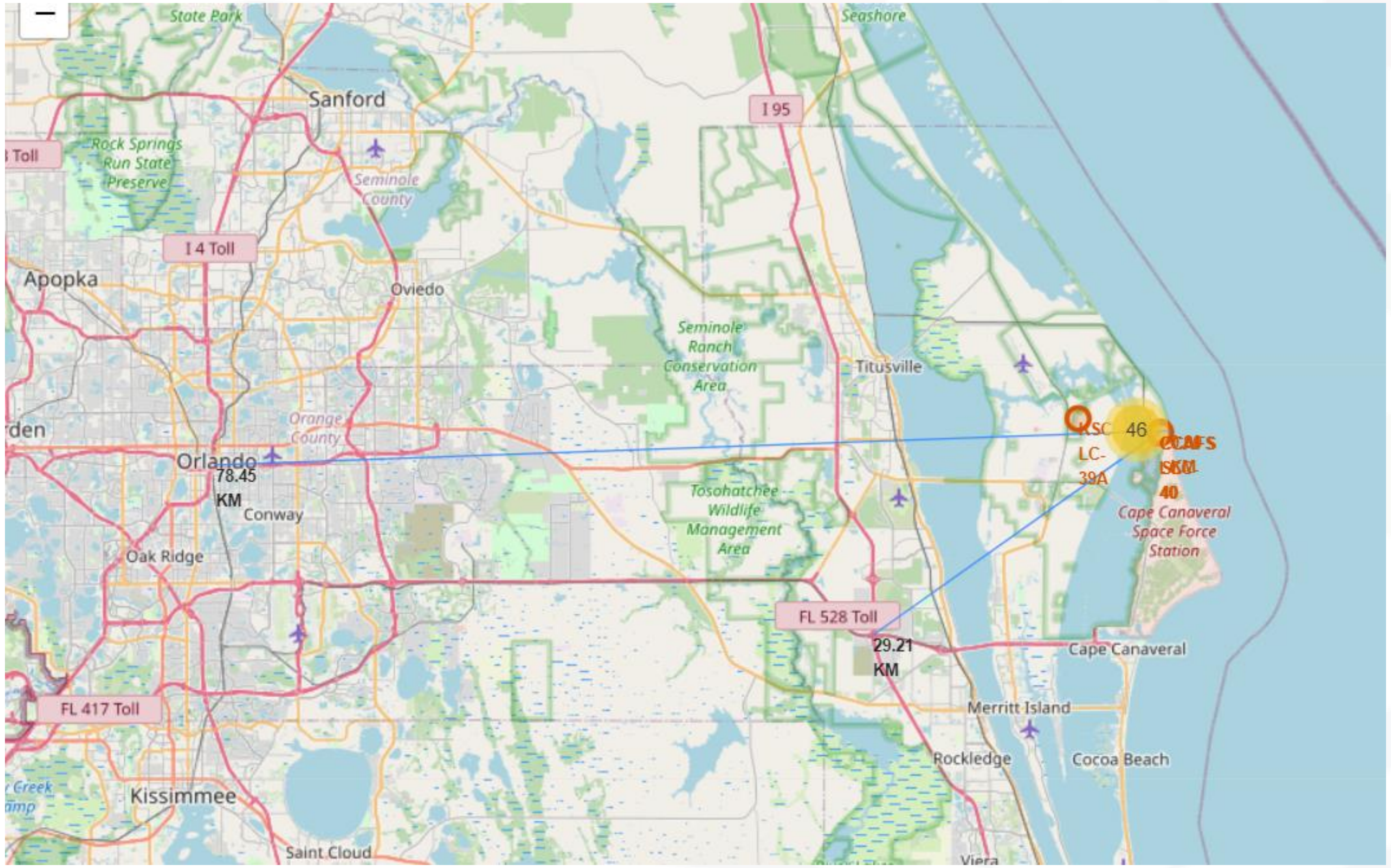
RESULTS – Interactive Visual Analytics with Folium Lab

Launch outcomes



RESULTS – Interactive Visual Analytics with Folium Lab

Distance to proximities

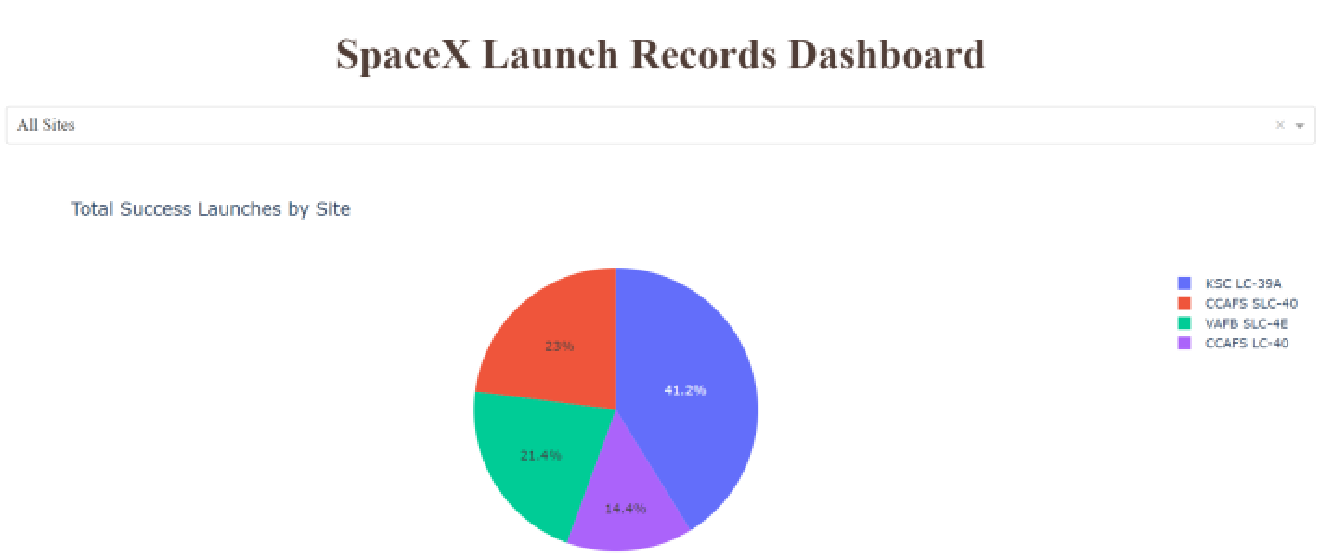


IBM Developer

SKILLS NETWORK

RESULTS – Dashboard with Plotly

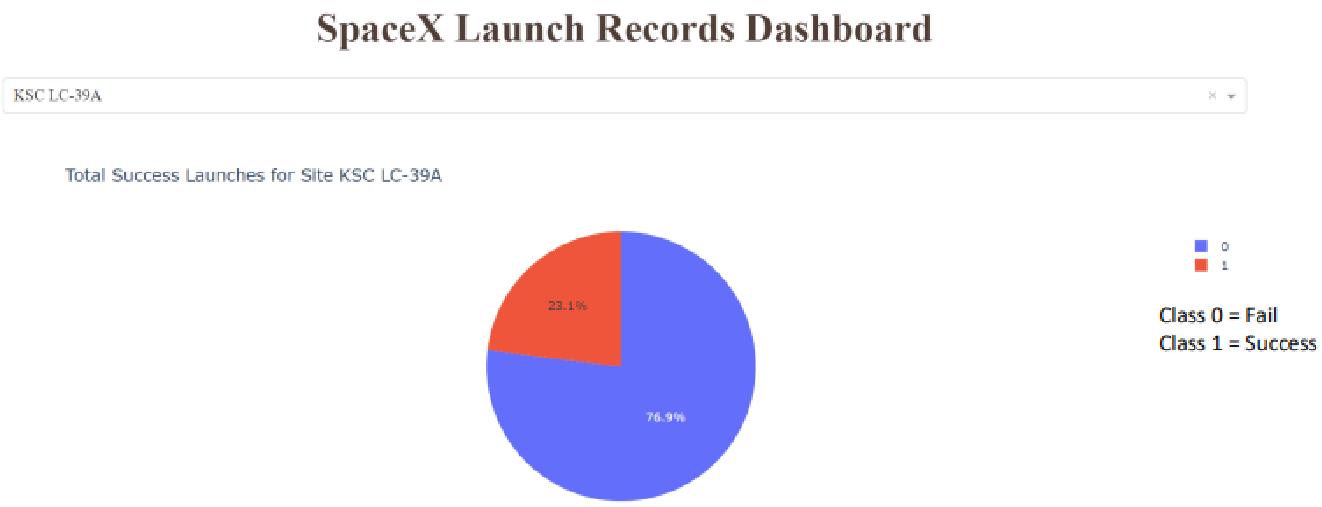
Launch success by sites



KSC LC-39A has the most successful launches (41.2%)

RESULTS – Dashboard with Plotly

Launch success KSC LC-39A



KSC LC-39A has the highest success rate (76.9%)

RESULTS – Dashboard with Plotly

Payload range and success

Payloads between 2,000 Kg and 5,000 Kg have the highest success rate

Predictive Analysis

Creating a NumPy array from the Class feature

```
In [8]:  Y = data['Class'].to_numpy()
         Y

Out[8]: array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
               1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
               1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1], dtype=int64)
```

Standardize and reassign data

```
In [13]:  # students get this
          transform = preprocessing.StandardScaler()
          X = transform.fit_transform(X)
          X

Out[13]: array([[-1.71291154e+00, -5.29526321e-17, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                [-1.67441914e+00, -1.19523159e+00, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                [-1.63592675e+00, -1.16267307e+00, -6.53912840e-01, ...,
                 -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
                ...,
                [ 1.63592675e+00,  1.99100483e+00,  3.49060516e+00, ...,
                  1.19684269e+00, -5.17306132e-01,  5.17306132e-01],
                [ 1.67441914e+00,  1.99100483e+00,  1.00389436e+00, ...,
                  1.19684269e+00, -5.17306132e-01,  5.17306132e-01],
                [ 1.71291154e+00, -5.19213966e-01, -6.53912840e-01, ...,
                 -8.35531692e-01, -5.17306132e-01,  5.17306132e-01]])
```

Predictive Analysis

Creating a training data set

`X_train, X_test, Y_train, Y_test`

In [14]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.2, random_state=2)
```

we can see we only have 18 test samples.

In [15]:
```
Y_test.shape
```

Out[15]:
```
(18,)
```

Creating a logistic regression

In [16]:
```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

In [17]:
```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

logreg_cv = GridSearchCV(lr, parameters, cv=10)

logreg_cv.fit(X_train, Y_train)
```

Out[17]:
```
GridSearchCV(cv=10, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']})
```

Predictive Analysis

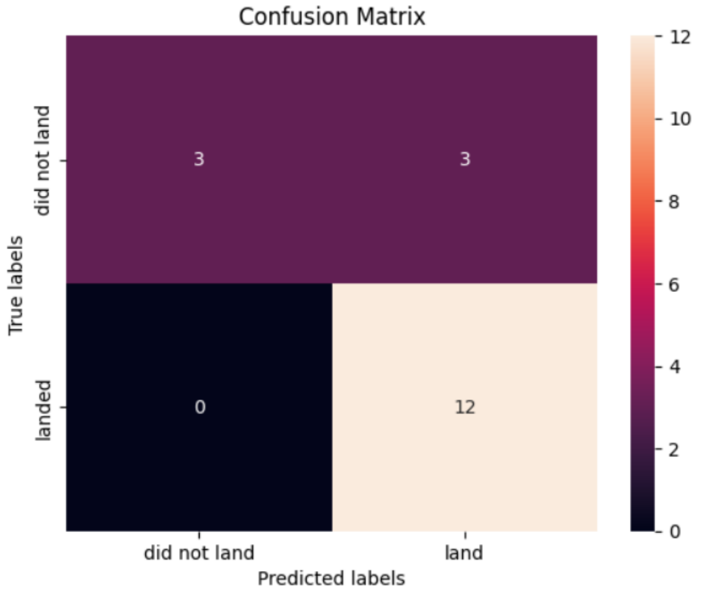
Accuracy test

In [19]:
```
logreg_cv.score(X_test, Y_test)
```

Out[19]: 0.8333333333333334

Confusion matrix

In [20]:
```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusion

Model performance :

The models performed similarly on the test set with the decision tree model slightly outperforming

Launch sites :

Most of the launches are closed to the equator for an additional natural boost

Launch success increased over time

KSC LC39-A has the highest success rate.

Payload mass : the higher the payload mass, the higher the success rate