Applied Data Science Capstone

Armando Morgado de Oliveira Neto

11/08/2022

IBM Developer

SKILLS NETWORK

OUTLINE

- Executive Summary
- Methodology and Results
  - SQL
  - Data Visualization
  - Folium
  - Plot Dash
  - Machine Learning
- Introduction
- Conclusion
- Link GitHub

IBM Developer

SKILLS NETWORK

EXECUTIVE SUMMARY

- Used Methodology:
  - Collecting the data
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Analysis with Pandas and Matplotlib
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Ploty Dash
  - Machine Learning Prediction
- Results
  - Data Analysis Results
  - Interactive Analytics Results
  - Machine Learning Results

INTRODUCTION

- Project Context

The objective of this project is simulate how a company, named SpaceY, will compete against the giant SpaceX. Using data to comprehend how SpaceX can launch a rocket costing just 62 millions of dollars while all others do it for around 165 millions.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - External factors influence the success rate?
  - We can do regression logistic with the information collected?

METHODOLOGY

- Start collecting the data
  - Extract relevant information from Wikipedia page about Space X as you can see below:

```
df.head()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing
0	1	CCSFS	Transporter-1	~5,000 kg	SSO	SpaceX	Success\n	F9 B5B1058.5	Success
1	2	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure
2	3	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure
3	4	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n
4	5	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt

METHODOLOGY

On this paper the following tools were used to scrap and model the data:

- Pandas
- BeaultifulSoup
- Numpy
- Folium
- SQL
- Matplotlib
- Seaborn
- Sklearn
- Requests
- Plotdash

METHODOLOGY

- At Data Wrangling stage, the data was modeled to calculate the launch for each launching site, calculate the number and occurrence of each orbit, calculate the number and occurence of mission outcome per orbit type and create a landing outcome label from Outcome column.

RESULTS - Data Wrangling

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	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	1
1	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	1
2	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	1
3	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	1
4	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	1
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1005	-80.577366	28.561857	1
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1006	-80.577366	28.561857	0
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1007	-80.577366	28.561857	0
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1008	-80.577366	28.561857	1
9	10	2014-09-07	Falcon 9	4428.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1011	-80.577366	28.561857	1
10	11	2014-09-21	Falcon 9	2216.000000	ISS	CCAFS SLC 40	False Ocean	1	False	False	False	NaN	1.0	0	B1010	-80.577366	28.561857	1
11	12	2015-01-10	Falcon 9	2395.000000	ISS	CCAFS SLC 40	False ASDS	1	True	False	True	5e9e3032383ecb761634e7cb	1.0	0	B1012	-80.577366	28.561857	1
12	13	2015-02-11	Falcon 9	570.000000	ES-L1	CCAFS SLC 40	True Ocean	1	True	False	True	NaN	1.0	0	B1013	-80.577366	28.561857	0
13	14	2015-04-14	Falcon 9	1898.000000	ISS	CCAFS SLC 40	False ASDS	1	True	False	True	5e9e3032383ecb761634e7cb	1.0	0	B1015	-80.577366	28.561857	1
14	15	2015-04-27	Falcon 9	4707.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1016	-80.577366	28.561857	1

METHODOLOGY - SQL

- Perform exploratory data analysis using visualization and SQL
  - We first stablish a conexion with the dataset and load SQL extension;
  - With SQL we can browse the dataset for useful information like names of unique launch sites, total payload mass carried by boosters launched by NASA (CRS), average payload mass carried by booster version F9 v1.1, first succesful landing outcome in ground pad was achieved, total number of successful and failure mission outcomes.

RESULTS - SQL

Task 1

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

In [83]:

```
%%sql

SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;
```

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

Out[83]:

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

Task 2

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

In [85]:

```
%%sql
SELECT * FROM SPACEXTBL
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

Out[85]:

Date	Time	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing
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

RESULTS - SQL

Task 3

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

In [92]:

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
```

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

Out[92]:

SUM(PAYLOAD_MASS__KG_)
45596

Task 4

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

In [91]:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version = 'F9 v1.1'
```

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

Out[91]:

AVG(PAYLOAD_MASS__KG_)
2928.4

RESULTS - SQL

Task 5

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

Hint:Use min function

In [97]:
```
%%sql
SELECT * FROM SPACEXTBL
WHERE Landing = 'Success (ground pad)'
Order by SPACEXTBL.Date ASC
Limit 1;
```

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

Out[97]:

Date	Time	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)

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 [101…
```
%%sql
SELECT Booster_Version FROM SPACEXTBL
where Landing = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

Out[101…

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

RESULTS - SQL

Task 7

List the total number of successful and failure mission outcomes

In [124…

```
%%sql
SELECT COUNT(Mission_Outcome) AS sucesso, teste AS falha
FROM (SELECT COUNT(Mission_Outcome) AS teste FROM SPACEXTBL WHERE Mission_Outcome Like '%Failure%' ),  SPACEXTBL
WHERE Mission_Outcome Like '%Success%'
```

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

Out[124…

sucesso	falha
100	1

Task 8

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

In [141…

```
%%sql
SELECT distinct(Booster_Version) FROM SPACEXTBL
where PAYLOAD_MASS__KG_ = (Select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Out[141…

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

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [158…

```
%%sql
SELECT SUBSTR(Date,4,2) as Mes, landing, Booster_Version, Launch_Site   FROM SPACEXTBL
where SUBSTR(Date,7,4) = '2015'
AND
landing = 'Failure (drone ship)'

--SUBSTR(Date,7,4)
```

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

Out[158…

Mes	Landing	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

In [170…

```
%%sql
SELECT DISTINCT(launch_site) FROM SPACEXTBL
```

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

Out[170…

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [167…

```
%%sql
SELECT landing, count(landing) FROM SPACEXTBL
WHERE SPACEXTBL.Date between '04-06-2010' and '20-03-2017'
AND landing LIKE '%Success%'
GROUP BY landing
Order by count(landing) desc
```

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

Out[167…

Landing	count(landing)
Success	20
Success (drone ship)	8
Success (ground pad)	6

METHODOLOGY – Data Visualization

- Data Analysis Exploratory with Visualization

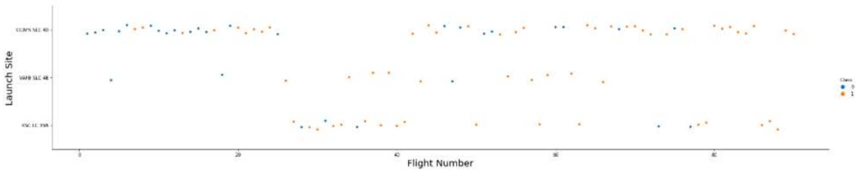
We now look for results using data visualization tools, we are now looking for a link between flight number and launch site using scatter chart. Another good point to look is launch sites and their payload mass, success rate of each orbit type, flight number and orbit type, etc. We basicaly use the vast amount of information we have and cross them together to get useful information for the stakeholders.

RESULTS - Data Visualization

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'`

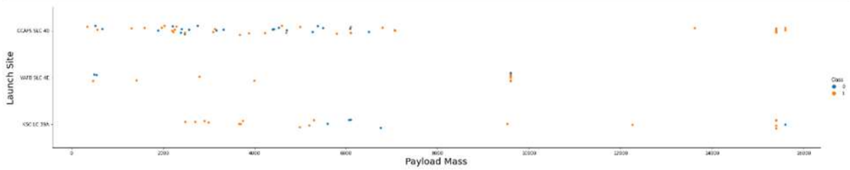
In [4]:
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



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.

In [5]:
```
# 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 b
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



IBM Developer

SKILLS NETWORK

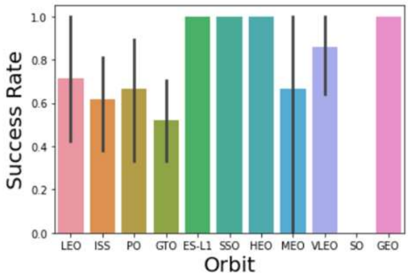
RESULTS - Data Visualization

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 sucess rate of each orbit

In [6]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
sns.barplot(y="Class", x="Orbit", data=df)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```

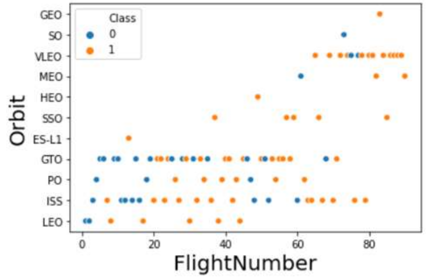


Analyze the ploted bar chart try to find which orbits have high sucess rate.

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.

In [7]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class
sns.scatterplot(y="Orbit", x="FlightNumber", hue="Class", data=df)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit" ,fontsize=20)
plt.show()
```



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.

IBM Developer

SKILLS NETWORK

RESULTS - Data Visualization

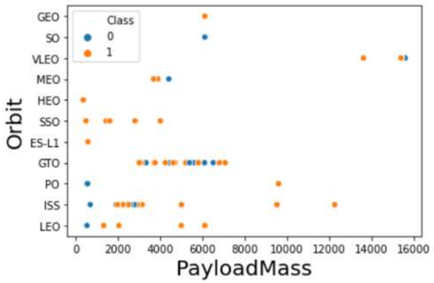
TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

In [8]:

```
# 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(y="Orbit", x="PayloadMass", hue="Class", data=df)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit" ,fontsize=20)
plt.show()
```



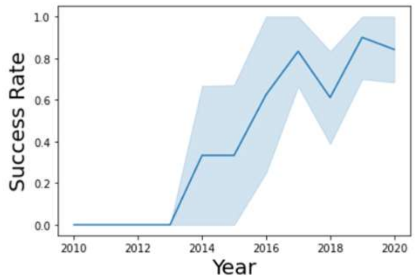
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

In [14]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate

sns.lineplot(y="Class", x="year", data=df)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate" ,fontsize=20)
plt.show()
```

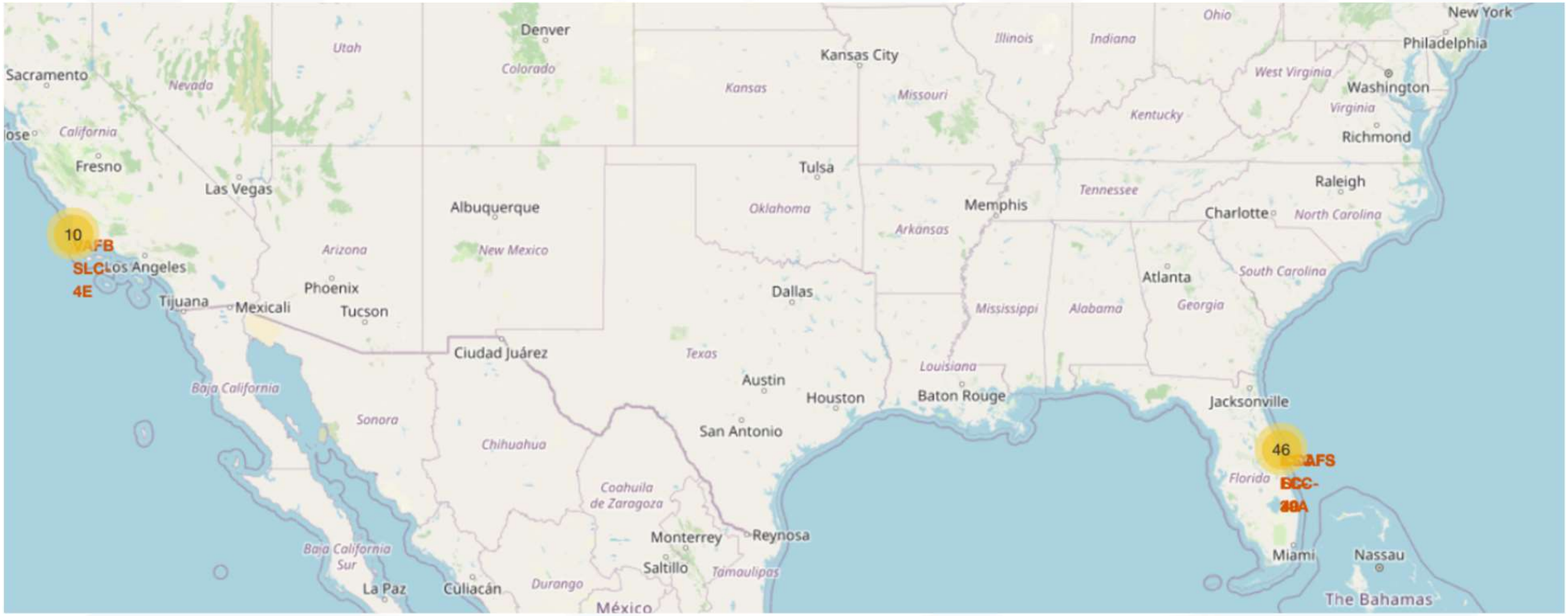


you can observe that the sucess rate since 2013 kept increasing till 2020

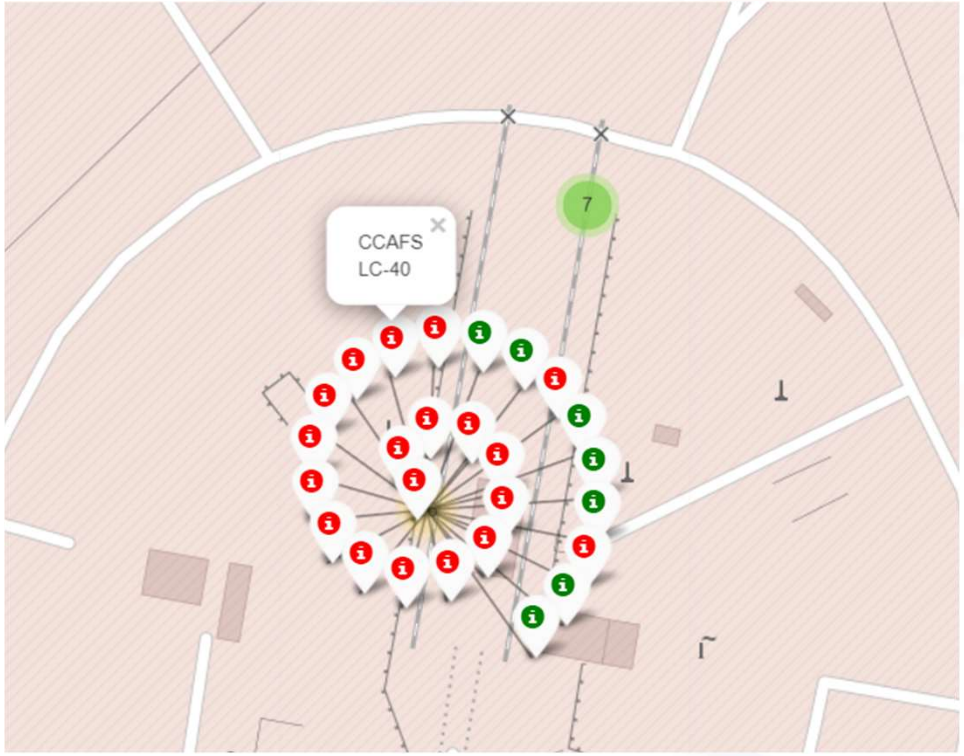
METHODOLOGY - Folium

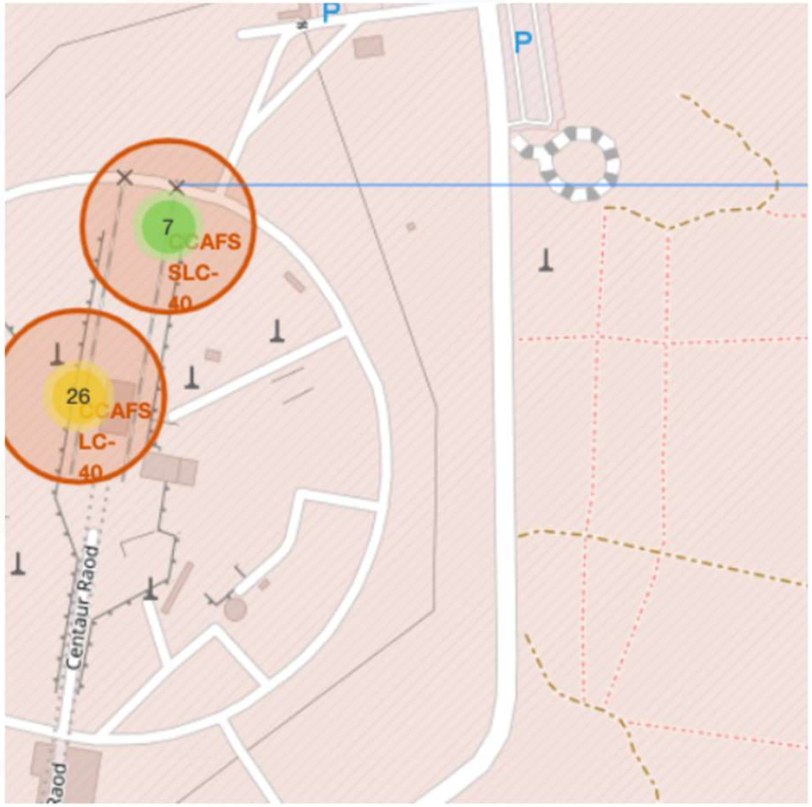
- Utilizing Folium to collect launch location data (lat and lng) and correctly marking on the map. We can now see useful information about the location of the launches utilizing the interactive map. Now we can transform all data and analyze it on the map.
- We already know the success rate depends on multiple factor and now we can assume location is also a valid indicator.

RESULTS - Folium

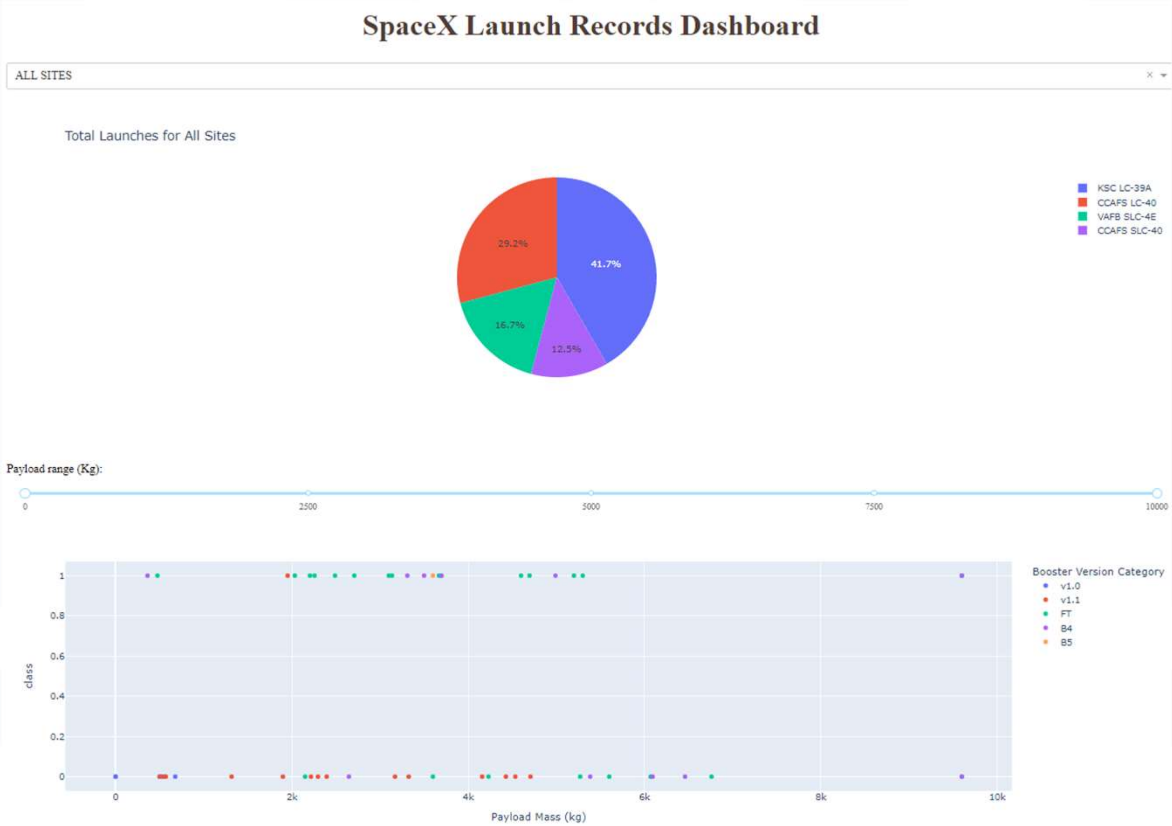


RESULTS - Folium





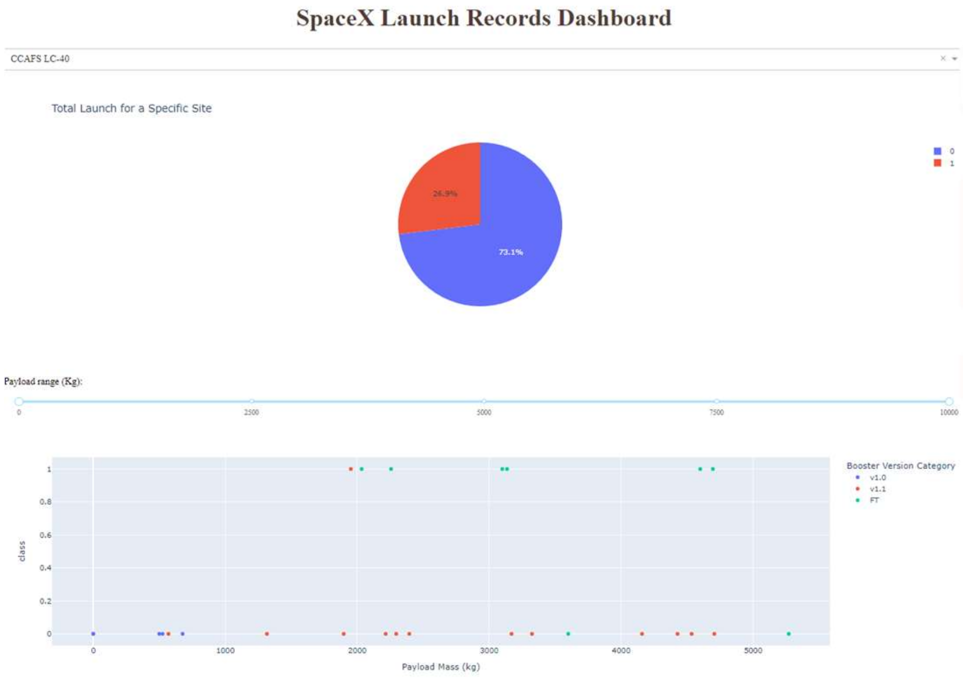
RESULTS – Plot Dash

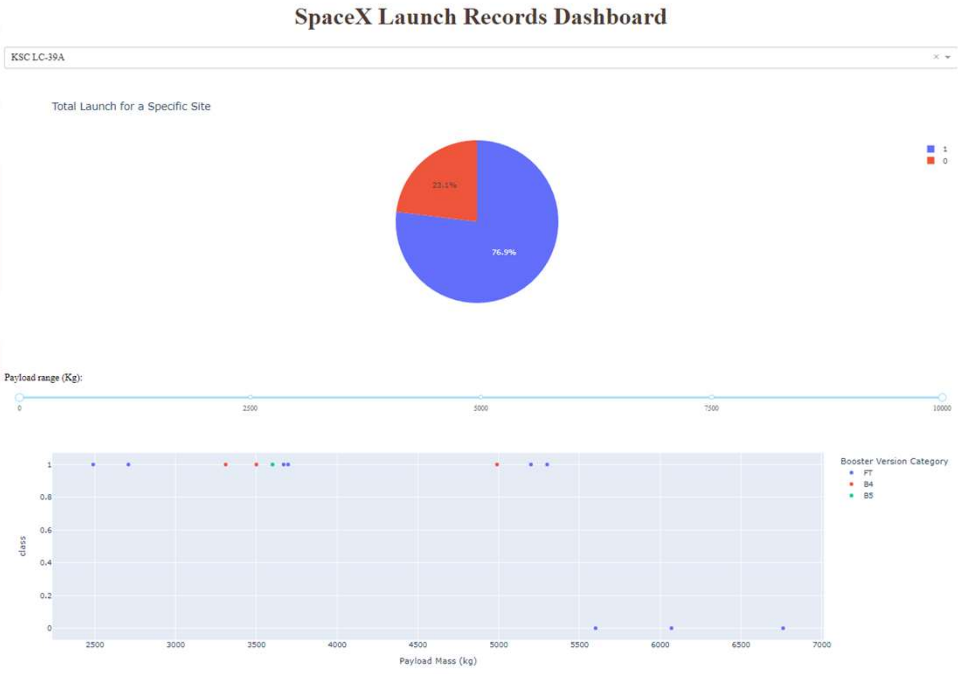


IBM Developer

SKILLS NETWORK

RESULTS – Plot Dash





IBM Developer

SKILLS NETWORK

METHODOLOGY - Machine Learning

- **Machine Learning Prediction**
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

RESULTS - Machine Learning

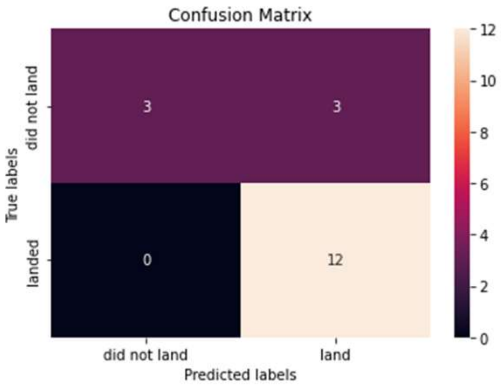
TASK 5

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

```
In [14]: logreg_cv.best_score_
Out[14]: 0.8196428571428571
```

Lets look at the confusion matrix:

```
In [15]: yhat=logreg_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



TASK 9

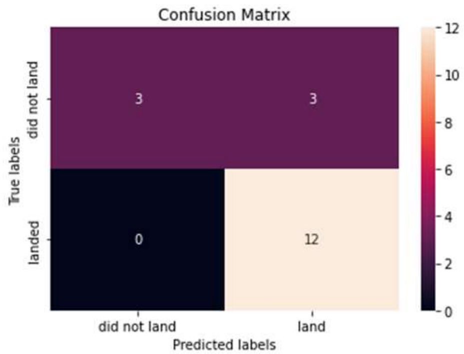
Calculate the accuracy of tree_cv on the test data using the method `score` :

```
In [31]: tree_cv.best_score_
Out[31]: 0.875
```

```
In [ ]:
```

We can plot the confusion matrix

```
In [20]: yhat_tree = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test, yhat_tree)
```



IBM Developer

SKILLS NETWORK

RESULTS - Machine Learning

TASK 9

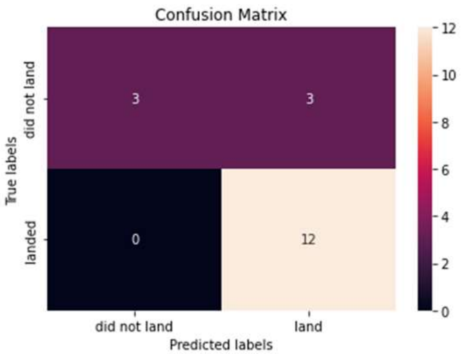
Calculate the accuracy of tree_cv on the test data using the method `score` :

```
In [31]: tree_cv.best_score_
Out[31]: 0.875
```

```
In [ ]:
```

We can plot the confusion matrix

```
In [20]: yhat_tree = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test, yhat_tree)
```



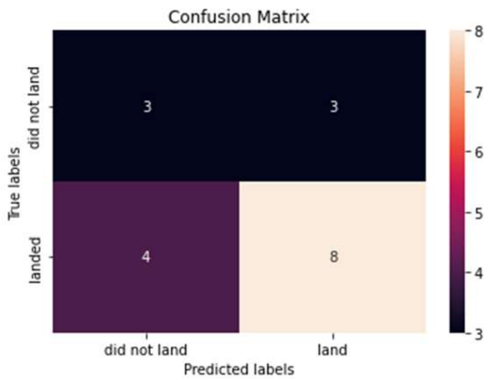
TASK 11

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

```
In [32]: knn_cv.best_score_
Out[32]: 0.6642857142857143
```

We can plot the confusion matrix

```
In [25]: yhat = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



RESULTS - Machine Learning

TASK 12

Find the method performs best:

In [36]:

```
print('Accuracy for Logistics Regression method:', logreg_cv.best_score_)
#print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.best_score_)
print('Accuracy for K nearsdt neighbors method:', knn_cv.best_score_)
```

```
Accuracy for Logistics Regression method: 0.8196428571428571
Accuracy for Decision tree method: 0.875
Accuracy for K nearsdt neighbors method: 0.6642857142857143
```

In [ ]:

CONCLUSION

Exploring the data through all the visualization presented we can conclude there was a positive evolution in the success rate on the landing of the SpaceX rockets. We can assume the most importants variables behind those success was rocket weight and orbit.

We could see KSC LC 39A was the most succesful lauch site when compared with his competitors. We could also see the best model of machine learning to use to determine the success rate is the Decision Tree with an accuracy of 87%.

Link GitHub

https://github.com/ArmandoOliveira/Applied-Data-Science-Capstone