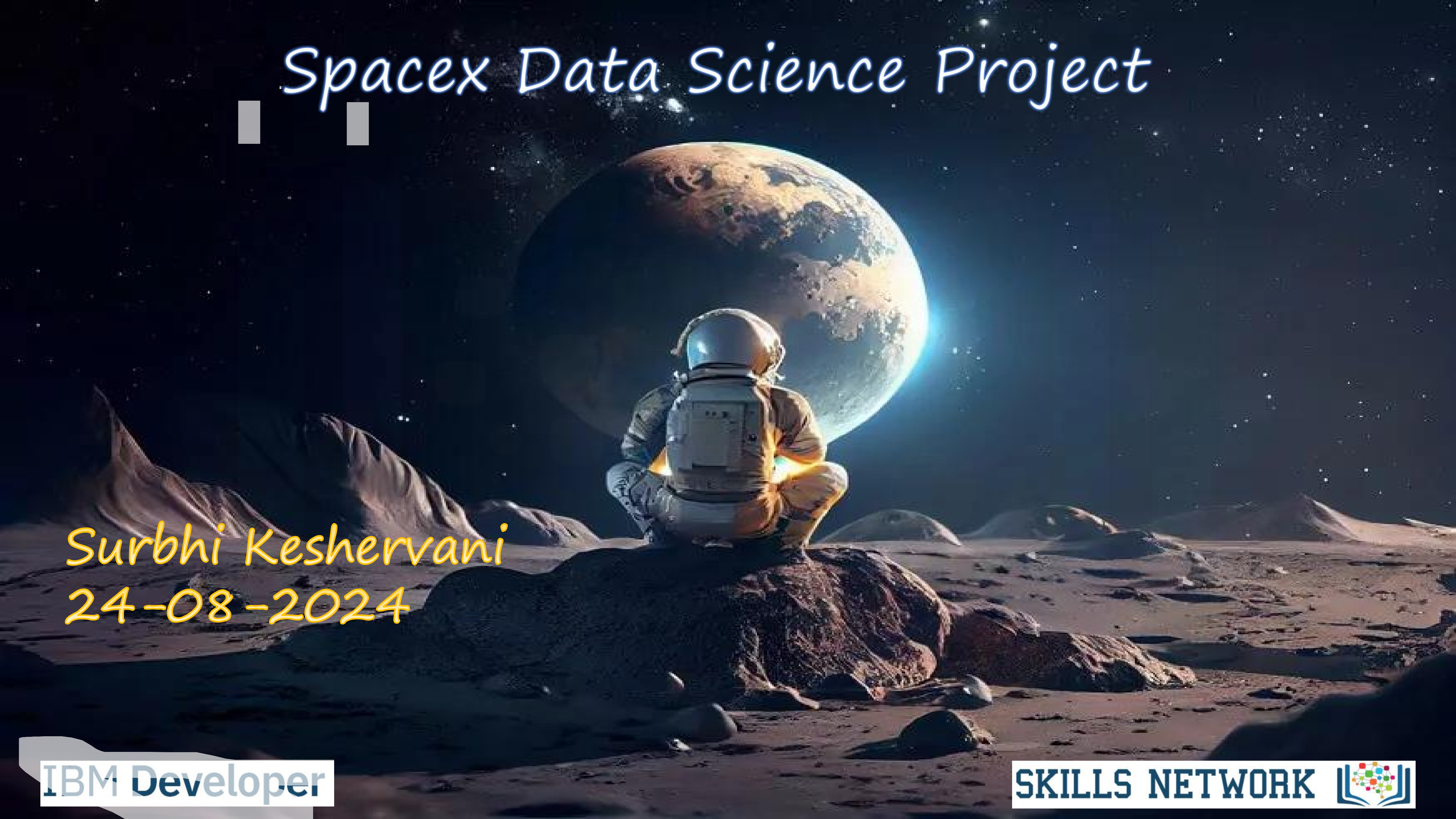
Spacex Data Science Project



Surbhi Keshervani
24-08-2024

IBM Developer

SKILLS NETWORK

OUTLINE

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

EXECUTIVE SUMMARY

- *Summary of Methodologies*
  - *Data Collection via API, web scraping*
  - *Data Wrangling*
  - *EDA with SQL*
  - *EDA using Python Pandas, Matplotlib*
  - *Building an interactive map with Folium*
  - *Buidling an interactive Dashboard with Plotly Dash*
  - *Predictive Analysis*
- *Summary of All Results*
  - *Exploratory Data Analysis Results*
  - *Interactive demo with screenshots*
  - *Predictive Analysis Results*

INTRODUCTION

- *Project Background and Context*
  - *The aim of this project is to predict if the Falcon 9 first stage will successfully land. SpaceX says on its website that the Falcon 9 rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By determining if the stage will land, we can determine the cost of a launch. This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.*
- *Project Background and Context*
  - *What are the main characteristics of a successful or failed landing ?*
  - *What are the effects of each relationship of the rocket variables on the success or failure of a landing ?*
  - *What are the conditions which will allow SpaceX to achieve the best landing success rate ?*

Methodology

IBM Developer

SKILLS NETWORK

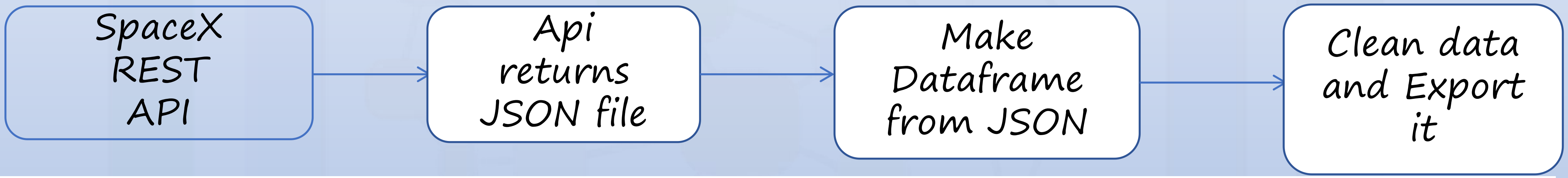
METHODOLOGY

EXECUTIVE SUMMARY

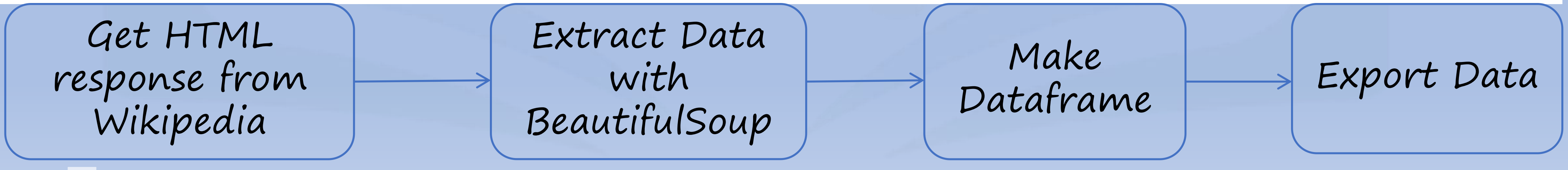
- Data Collection Methodology
  - SpaceX REST API
  - Web Scraping from Wikipedia
- Perform Data Wrangling
  - Dropping Necessary Columns
  - One-Hot Encoding for Classification Models
- Perform Exloratory Data Analysis using Visualization and SQL
- Perform Exploratory Data Analysis using Folium and Plotly Dash
- Perform Predictive Analysis using Classification Models
  - How to build, tune, evaluate classification models

DATA COLLECTION

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
- The information obtained by the API are rocket, launches, payload information.
- The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
- URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

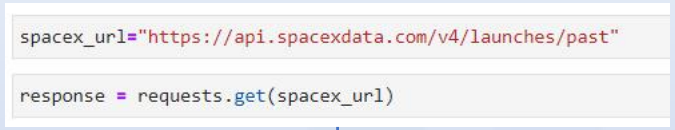

Get HTML response from Wikipedia
Extract Data with BeautifulSoup
Make Dataframe
Export Data

IBM Developer

SKILLS NETWORK

Data Collection - SpaceX API

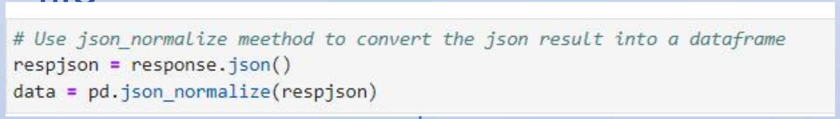
1. Getting response from API



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

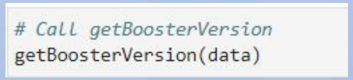
```
response = requests.get(spacex_url)
```

2. Convert Response from JSON file

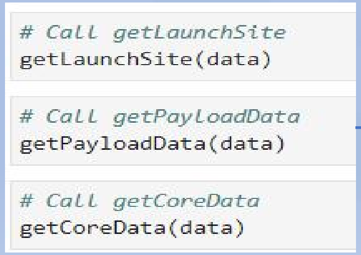


```
# Use json_normalize meethod to convert the json result into a dataframe
respjson = response.json()
data = pd.json_normalize(respjson)
```

3. Transform Data



```
# Call getBoosterVersion
getBoosterVersion(data)
```

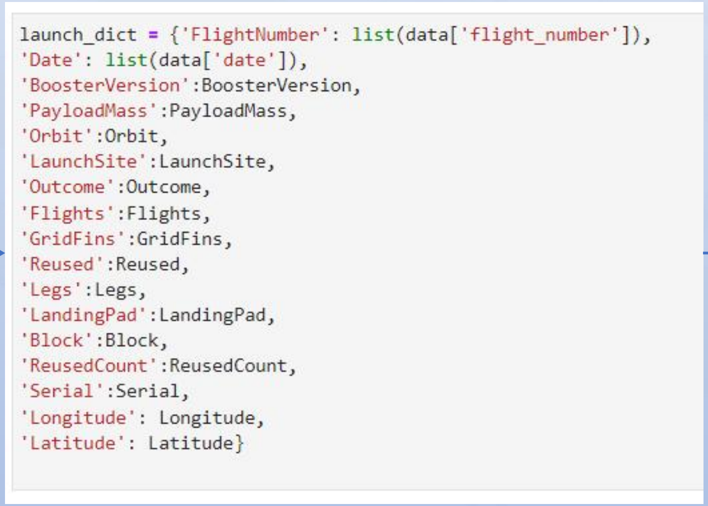


```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

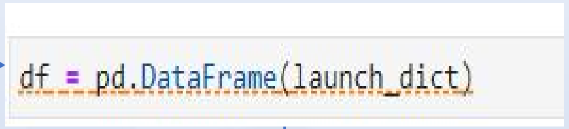
```
# Call getCoreData
getCoreData(data)
```

4. Create Dictionary



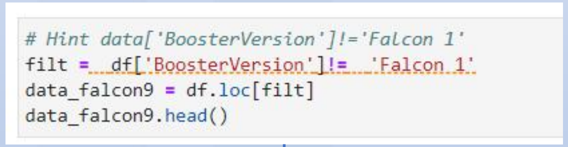
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create Dataframe



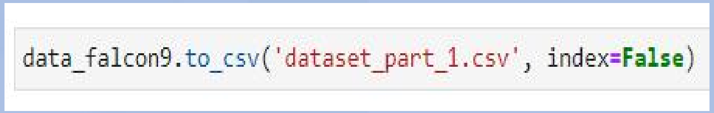
```
df = pd.DataFrame(launch_dict)
```

6. Filter Dataframe



```
# Hint data['BoosterVersion']!='Falcon 1'
filt = df['BoosterVersion']!= 'Falcon 1'
data_falcon9 = df.loc[filt]
data_falcon9.head()
```

7. Export to File



```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

IBM Developer

SKILLS NETWORK

Data Collection - Scraping

1. Getting response from HTML

```
response = requests.get(static_url)
```

2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.content, 'html.parser')
```

3. Find All Tables

```
html_tables = soup.find_all('table')
```

4. Get Column Names

```
column_names = []
table = first_launch_table.find_all('th')
for row in table:
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

5. Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add Data to Keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

See notebook for the rest of the code

6. Create Dataframe from the dictionary

```
df=pd.DataFrame(launch_dict)
```

7. Export to File

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

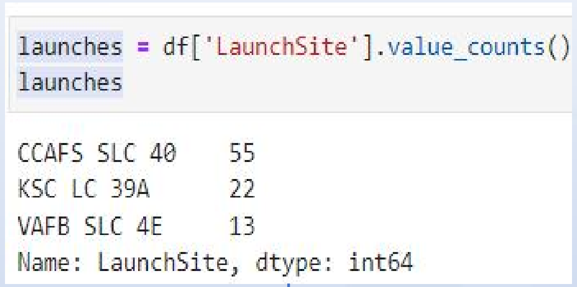
IBM Developer

SKILLS NETWORK

Data Wrangling

- In the dataset, there are several cases where booster did not landed successfully.
  1. True Ocean, True ASDS or True RTLS means the mission has been successful
  2. False Ocean, False ASDS or False RTLS means the mission has been a failure.
- We need to transform string variables into catoegorical variables where 1 means the mission has been successful and 0 means the mission has been a failure.

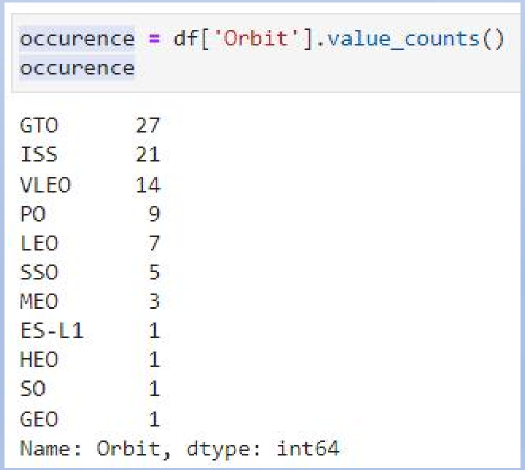
1. Calculate launches number for each site



```
launches = df['LaunchSite'].value_counts()
launches

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

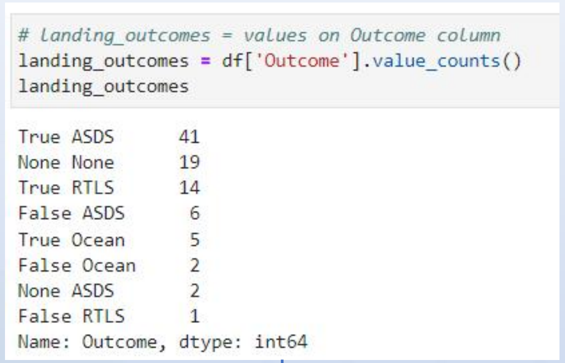
2. Calculate the Number and Occurence of each Orbit



```
occurence = df['Orbit'].value_counts()
occurence

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

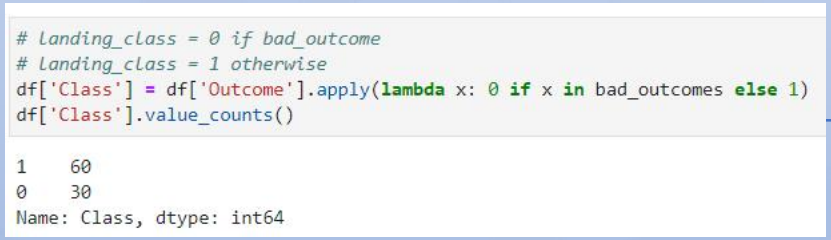
3. Calculate Number and Occurence of mission outcome per orbit type



```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

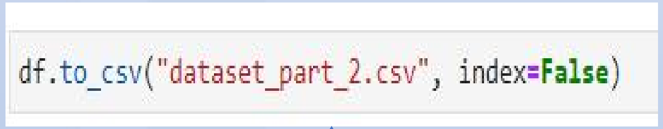
4. Creating Landing Outcome label from Outcome Column



```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()

1    60
0    30
Name: Class, dtype: int64
```

5. Export to File



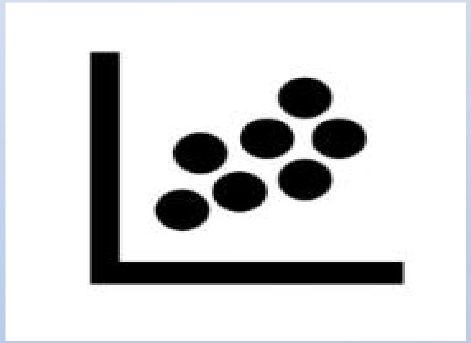
```
df.to_csv("dataset_part_2.csv", index=False)
```

IBM Developer

SKILLS NETWORK

EDA with Data Visualization

- Scatter graphs
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload vs Launch
  - Orbit vs Flight Number
  - Payload vs Orbit Type
  - Orbit vs Payload Mass



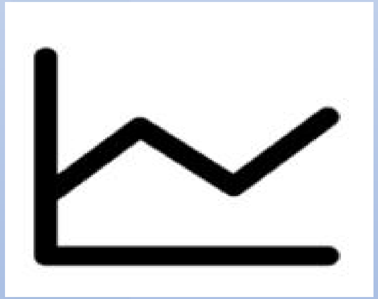
Scatter plots show relationship between Variables.

This relationship is called Correlation.

- Bar Chart
  - Success Rate vs Orbit

Bar Chart shows relationship between numerical and categorical variables.

- Line Graph
  - Success Rate vs Year



Line Graphs shows data variables and their trends. Line graphs can help to show global behaviour and make predictionsfor unseen data.

IBM Developer

SKILLS NETWORK

EDA with SQL

- **We performed SQL queries to gather and understand data from dataset.**
  - Displaying the names of the unique launch sites in the space mission.
  - Display 5 records where launch sites begin with the string ‘CCA’.
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display the average paylaod mass carried by booster version F9 v1.1
  - List the date when the first successful in the ground pad was achieved.
  - List the name of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the number of successful and failure mission outcomes.
  - List the number of booster_version whih have carried the maximum payload mass.
  - List the records whch will display the month names, the failure landing _outcomes in drone ship, booster versions, launch sites for the months in year 2015.
  - Rank the successful landing_outcome between the date 04-06-2010 and 20-03-2017 in descending order.

Build an Interactive Map with Folium

- **Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas.**
  - Red Circle at NASA Johnson Space Center coordinte with label showing its name. (folium.Circle, folium.map.Marker, folium.features.DivIcon)
  - Red Circles at each launch site coordiates with showing launch site names. (folium.Circle, folium.map.Marker)
  - The grouping of points in a cluster to display multiple and different information for the same coordinates. (folium..plugins.MarkerCluster)
  - Marker to suucessful and unsuccessful landings. Green for successful and Red for unsuccessful landings. (folium.map.Marker, folium.Icon)
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- **These objects are created in order to understand better the problem and the data. we can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.**

Build a Dashboard with Plotly Dash

- **Dashboard has dropdown, pie chart, rangeslider and scatter plot components.**
  - Dropdown allows a user to choose the launch site or all launch sites. (dash_core_components.Dropdown)
  - Pie Chart shows the total success and the total failure for the launch site chosen with the dropdown component. (plotly.express.pie)
  - Rangeslider allows a user to select a paylaod mass in a fixed range. (dash_core_components.RangeSlider)
  - Scatter chart shows the relationship between two variables, in particular Success vs Paylaod Mass. (plotly.express.Scatter)

Predictive Analysis (Classification)

- **Data Preparation**
  - Load Dataset.
  - Normalize Data.
  - Split data into training and testing datasets.
- **Model Preparation**
  - Selection of Machine Learning Algorithms.
  - Set parameter for each algorithm to GridSearchCV.
  - Training GridSearchModel models with training dataset.
- **Model Evaluation**
  - Get best hyperparameters for each type of model.
  - Computer Accuracy for each model with test dataset.
  - Plot Confusion Matrix.
- **Model Comparison**
  - Comparison of each model according to their accuracy.
  - The model with the best accuracy will be chosen. (see notebook for result)

RESULTS

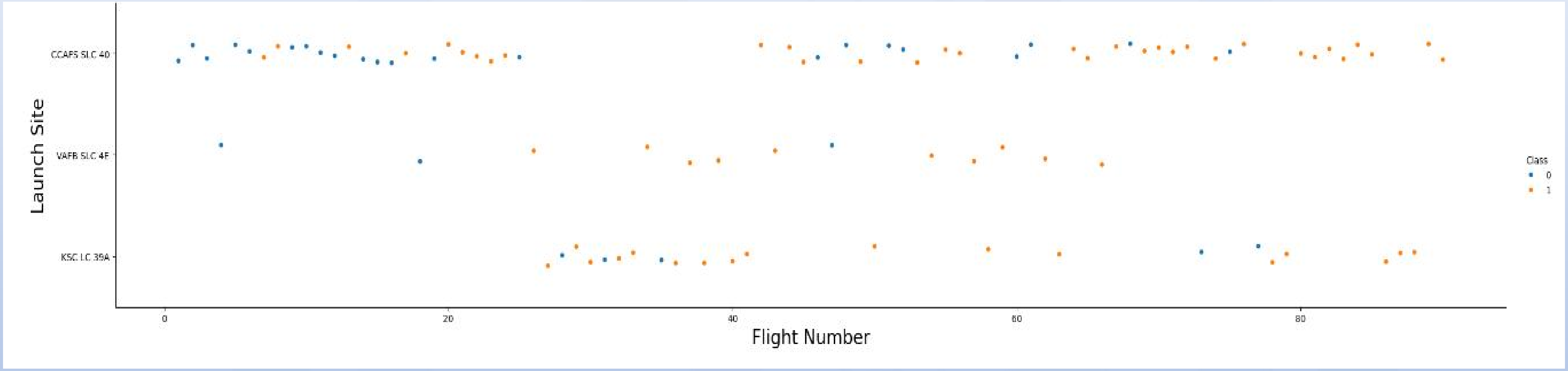
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Insights Drawn from EDA

IBM Developer

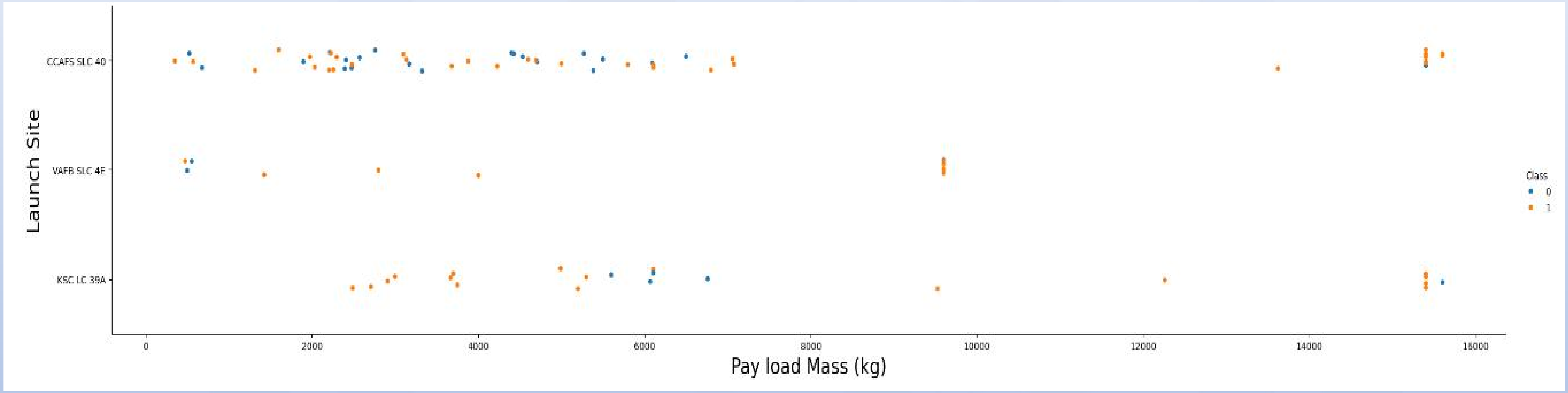
SKILLS NETWORK

Flight Number vs Launch Site



We observe that, for each site, the success rate is increasing

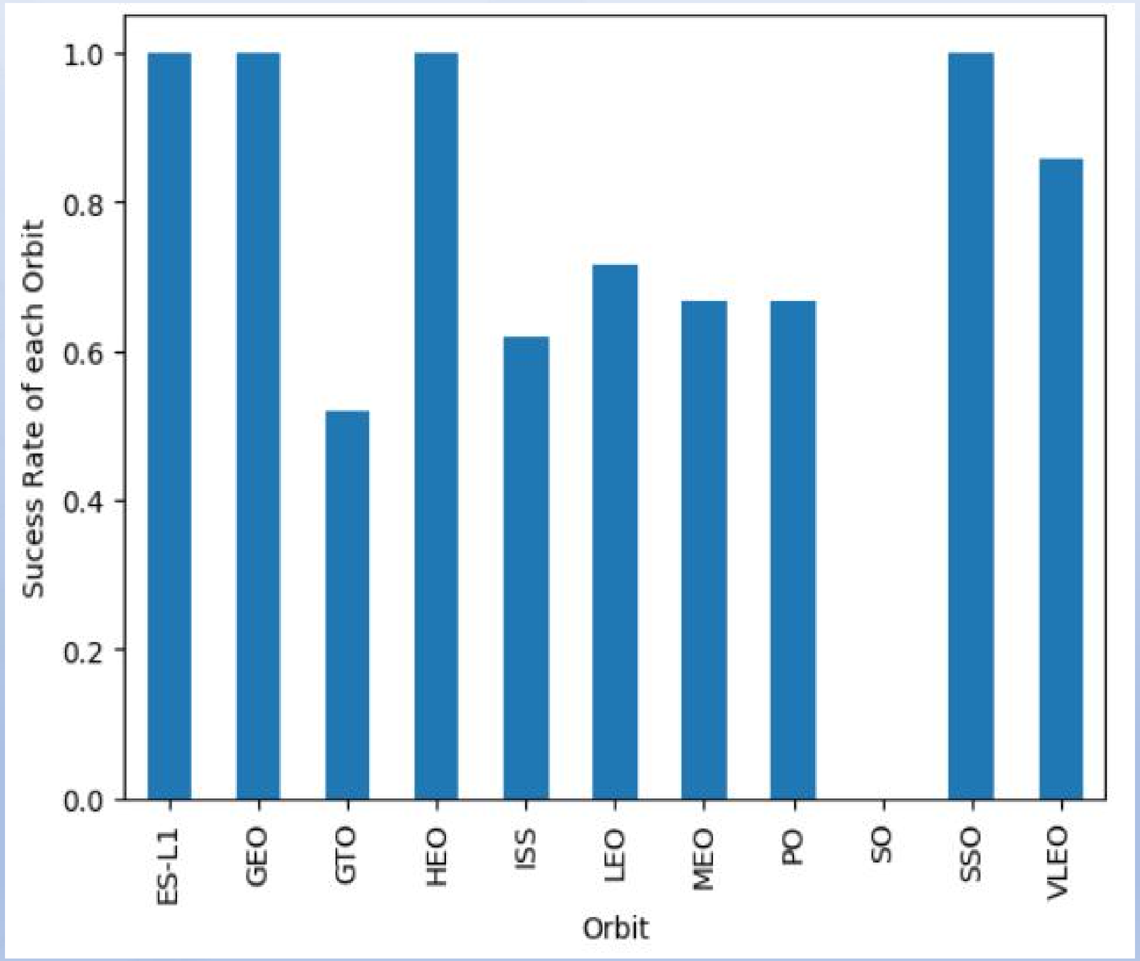
Payload vs Launch Site



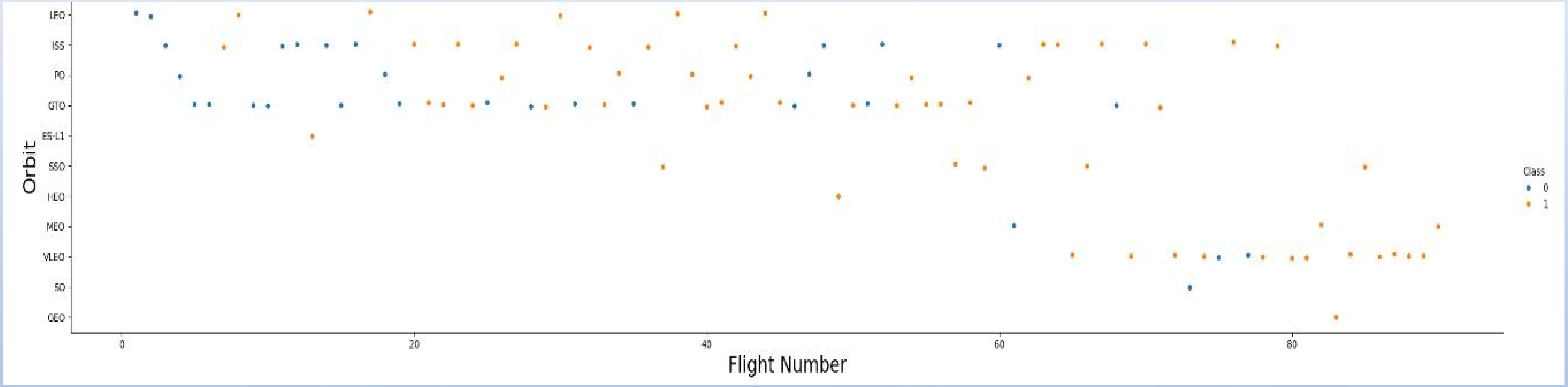
Depending on the launch site, a heavier payload may be a consideration for a successful landing. A too heavy payload mass can make a landing fail.

Success Rate vs Orbit Type

With this plot, we can success rate for different orbit types. we note that ES-L1, GEO, HEO, SSO have the best success rate.

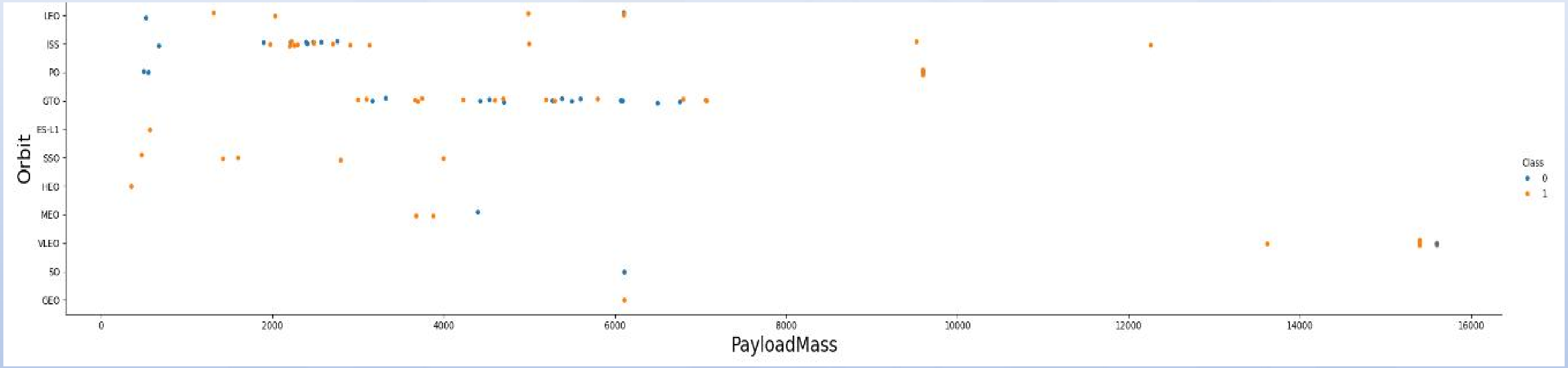


Flight Number vs Orbit Type



We notice that the success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights. But we can suppose that the high successvrate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.

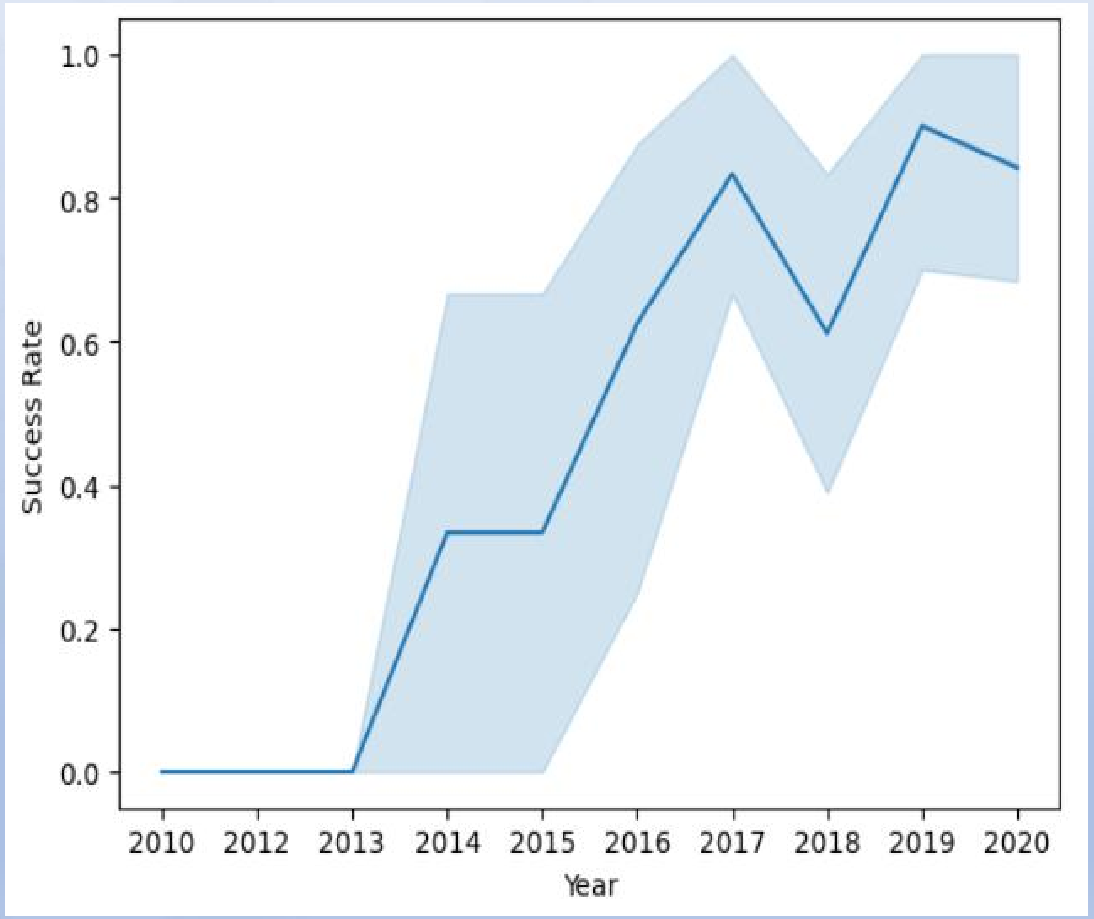
Payload vs Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for a GTO orbit improves the success of a launch.

Launch Success Yearly Trend

Since 2013,we can see an increase in the Space X Rocket success rate

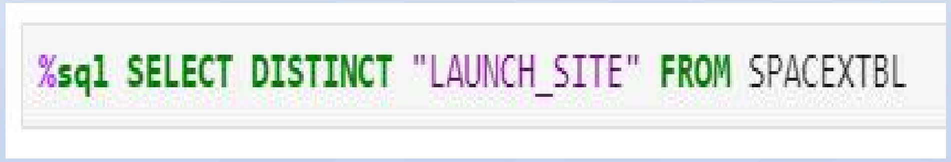


IBM Developer

SKILLS NETWORK

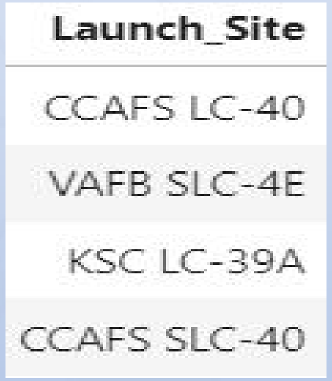
All Launch Site Names

SQL Query



```
%sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Results



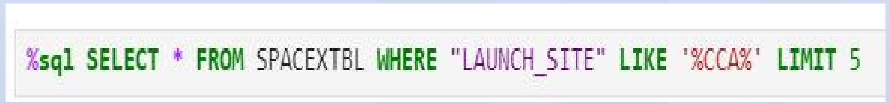
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Explanation

The use of DISTINCTin the query allows to remove duplicateLAUNCH_SITE.

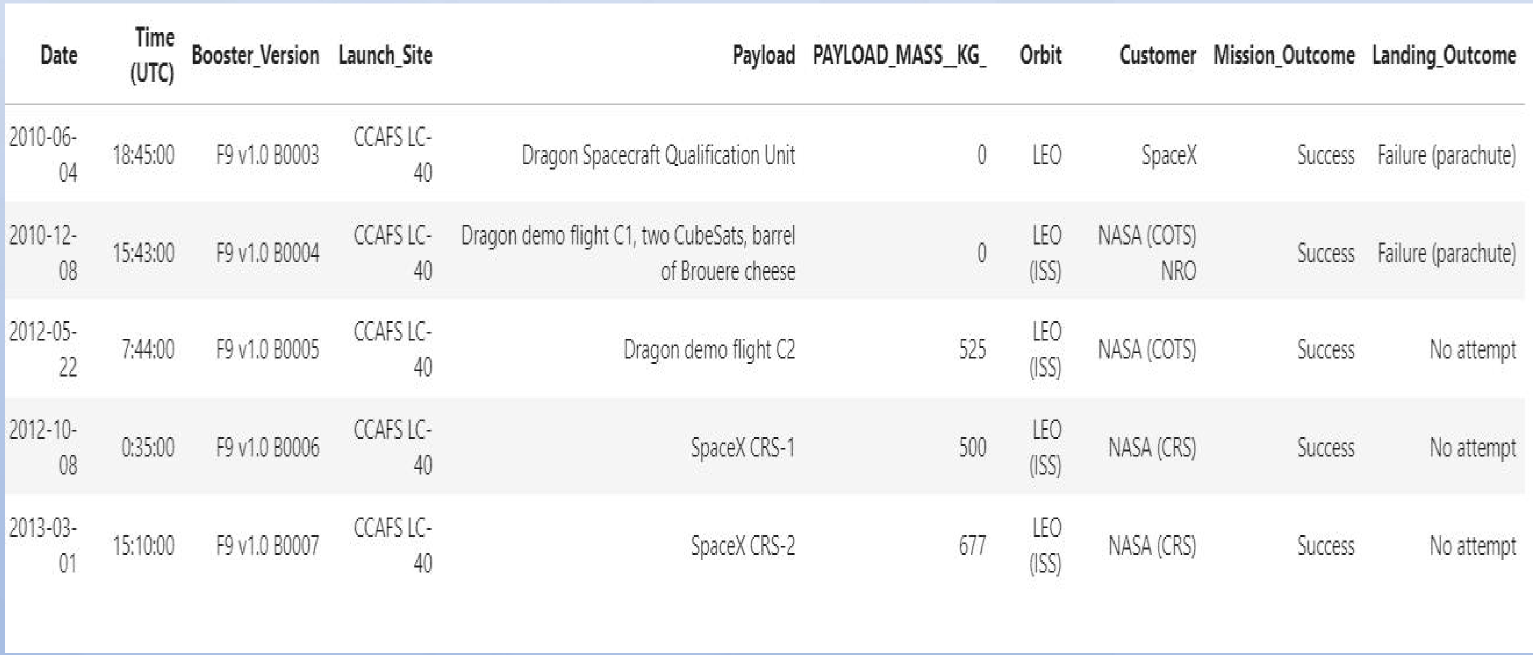
Launch Site Names Begin with 'CCA'

SQL Query



```
%sql SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Results



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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Explanation

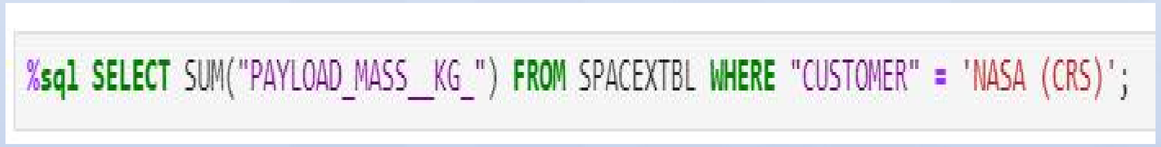
The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

IBM Developer

SKILLS NETWORK

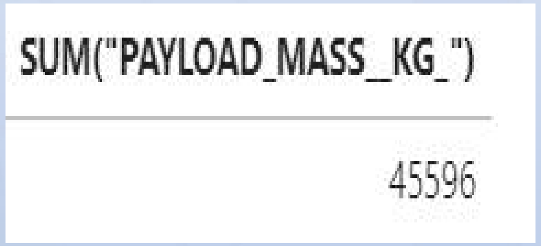
Total Payload Mass

SQL Query



```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)';
```

Results



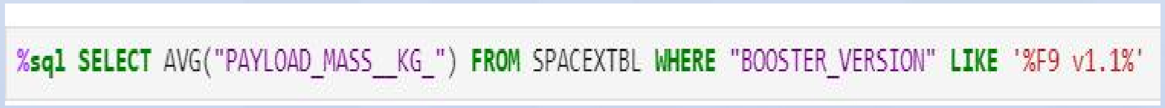
```
SUM("PAYLOAD_MASS__KG_")
                   45596
```

Explanation

This query returns the sum of all payloadmasses where the customeris NASA (CRS).

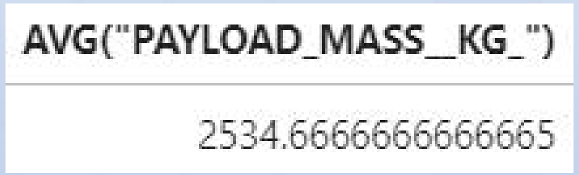
Average Payload Mass by F9 v1.1

SQL Query



```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results



AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

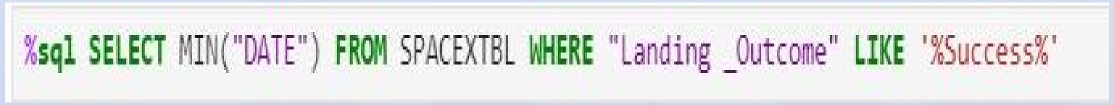
This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

IBM Developer

SKILLS NETWORK

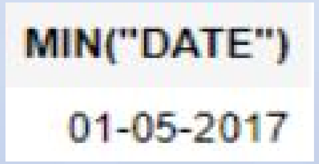
First Successful Ground Landing Date

SQL Query



```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results



MIN("DATE")
01-05-2017

Explanation

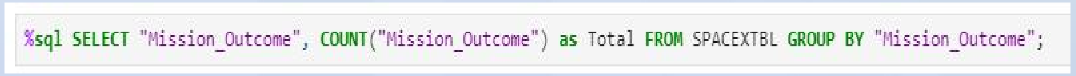
With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

IBM Developer

SKILLS NETWORK

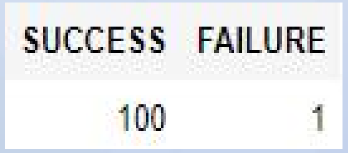
Total Number of Successful and Failure Mission Outcomes

SQL Query



```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

Results



SUCCESS	FAILURE
100	1

Explanation

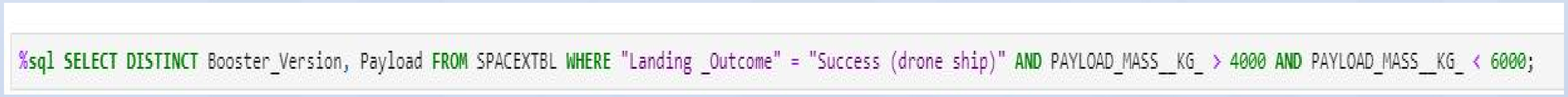
With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission.The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

IBM Developer

SKILLS NETWORK

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

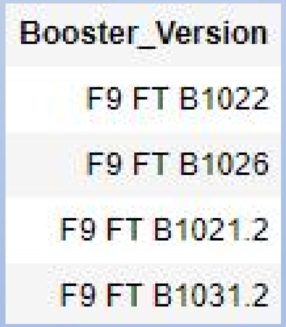


```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

Results



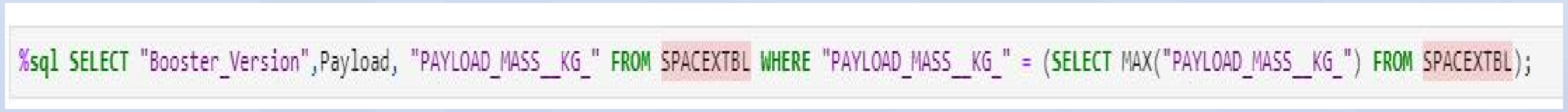
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

IBM Developer

SKILLS NETWORK

Boosters Carried Maximum Payload

SQL Query

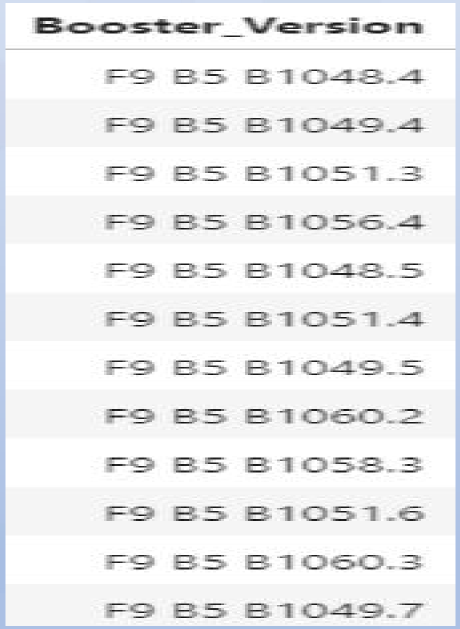


```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

Explanation

We used a subquery to filter data by returning only the heaviest payload mass with MAX function.The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

Results



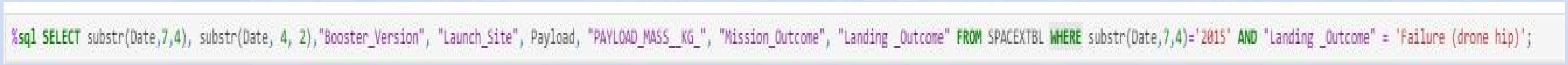
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

IBM Developer

SKILLS NETWORK

2015 Launch Records

SQL Query

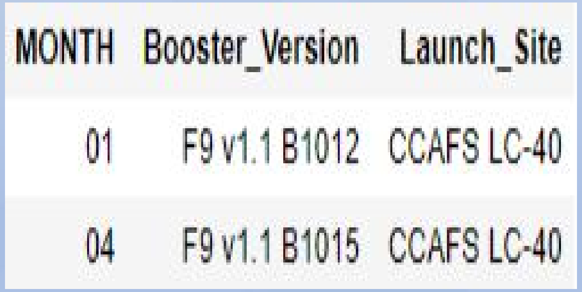


```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone hip)';
```

Explanation

This query returns month, booster version,launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7,4) showsyear.

Results



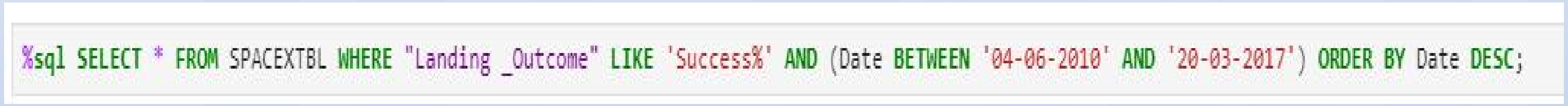
MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

IBM Developer

SKILLS NETWORK

Rank Landing Outcomes Between 04-06-2010 to 20-03-2017

SQL Query

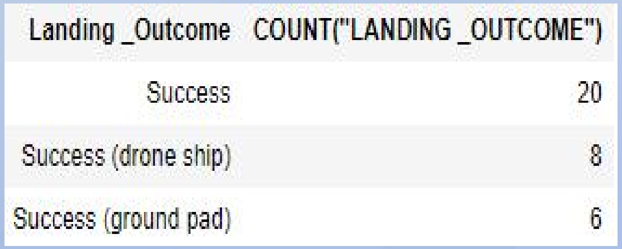


```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;
```

Explanation

This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Results



Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

IBM Developer

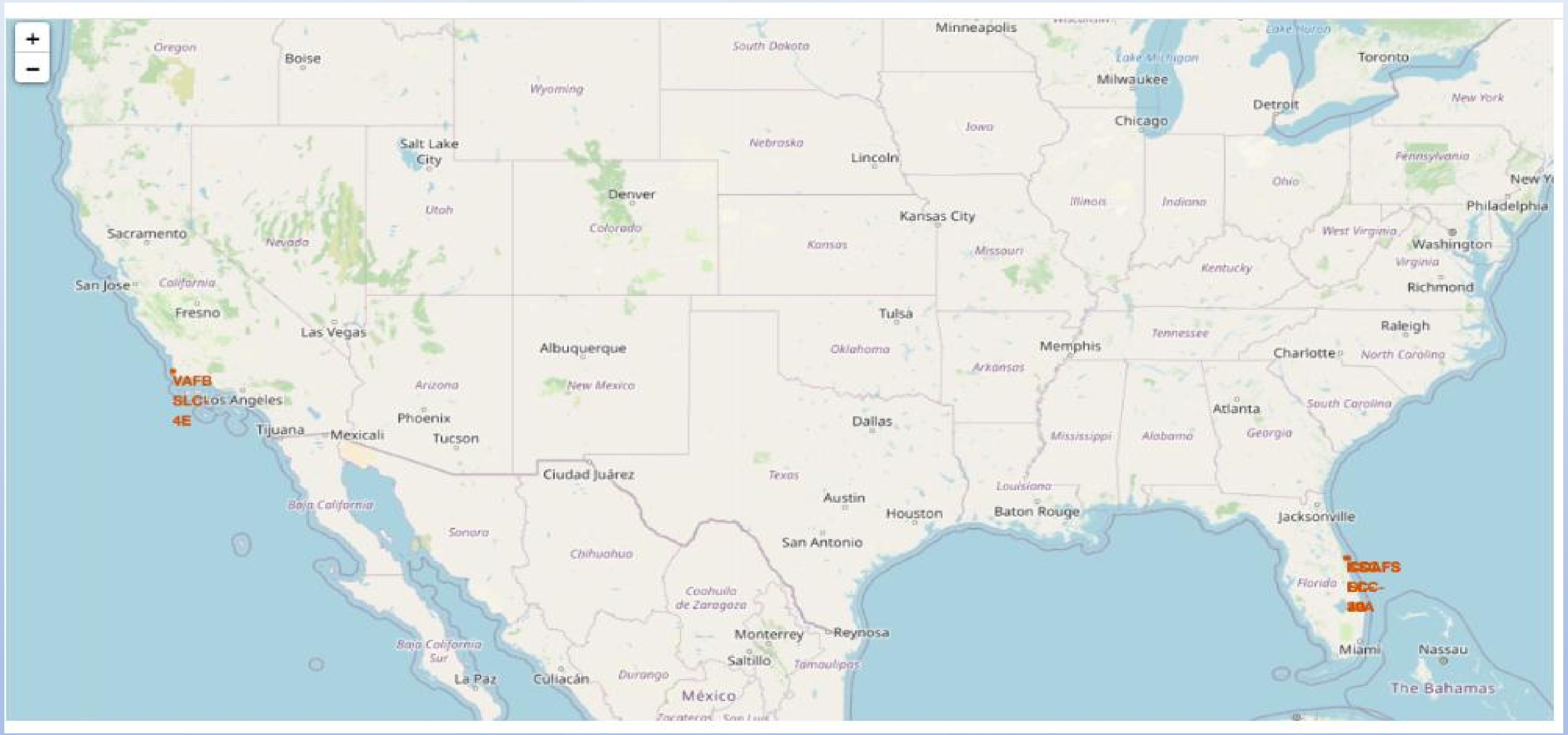
SKILLS NETWORK

Launch Sites Proximities Analysis

IBM Developer

SKILLS NETWORK

Folium Map - Ground Stations

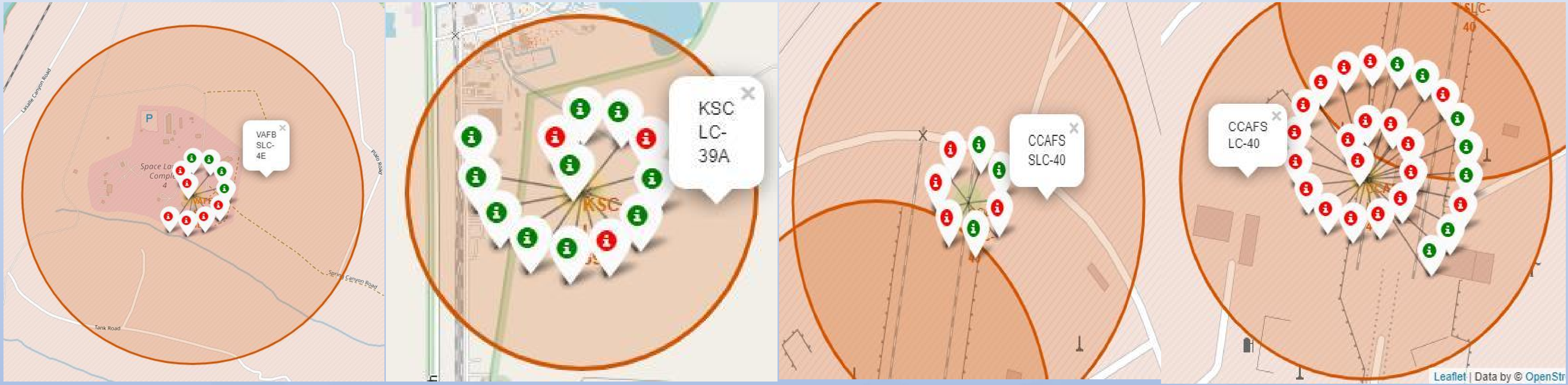


We see that SpaceX launch sites are located on the coast of the United States

IBM Developer

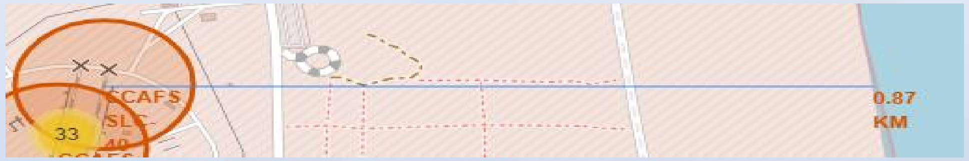
SKILLS NETWORK

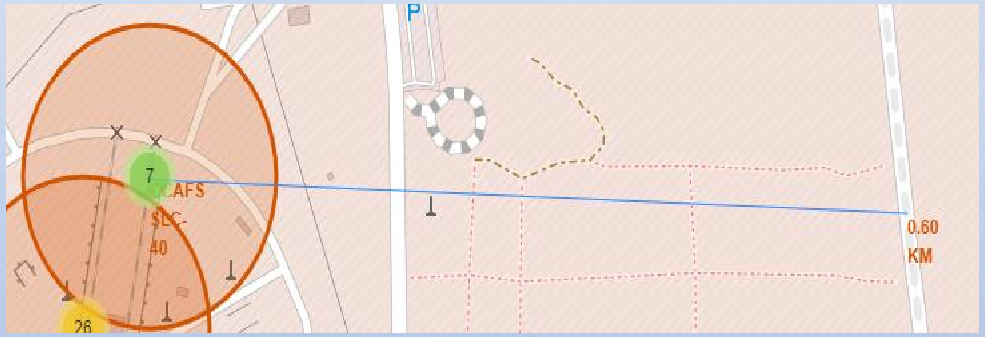
Folium Map - Color Labeled Markers

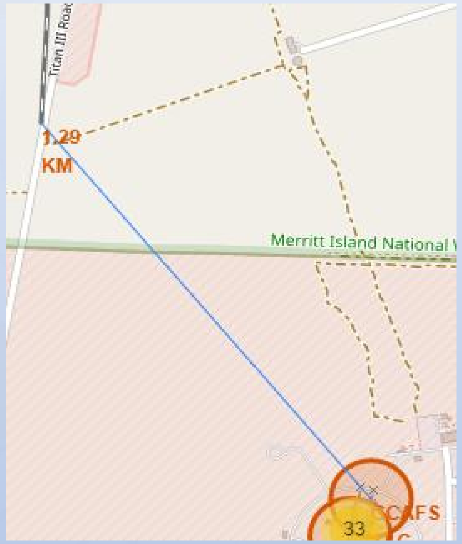


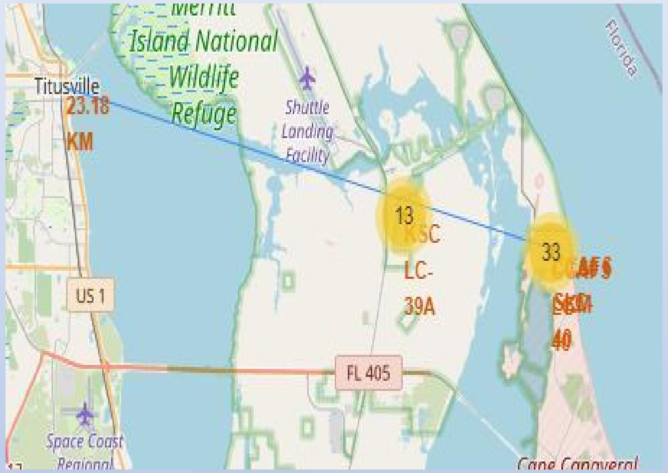
Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

Folium Map - Distances Between CCAFS SLC-40 and its proximities



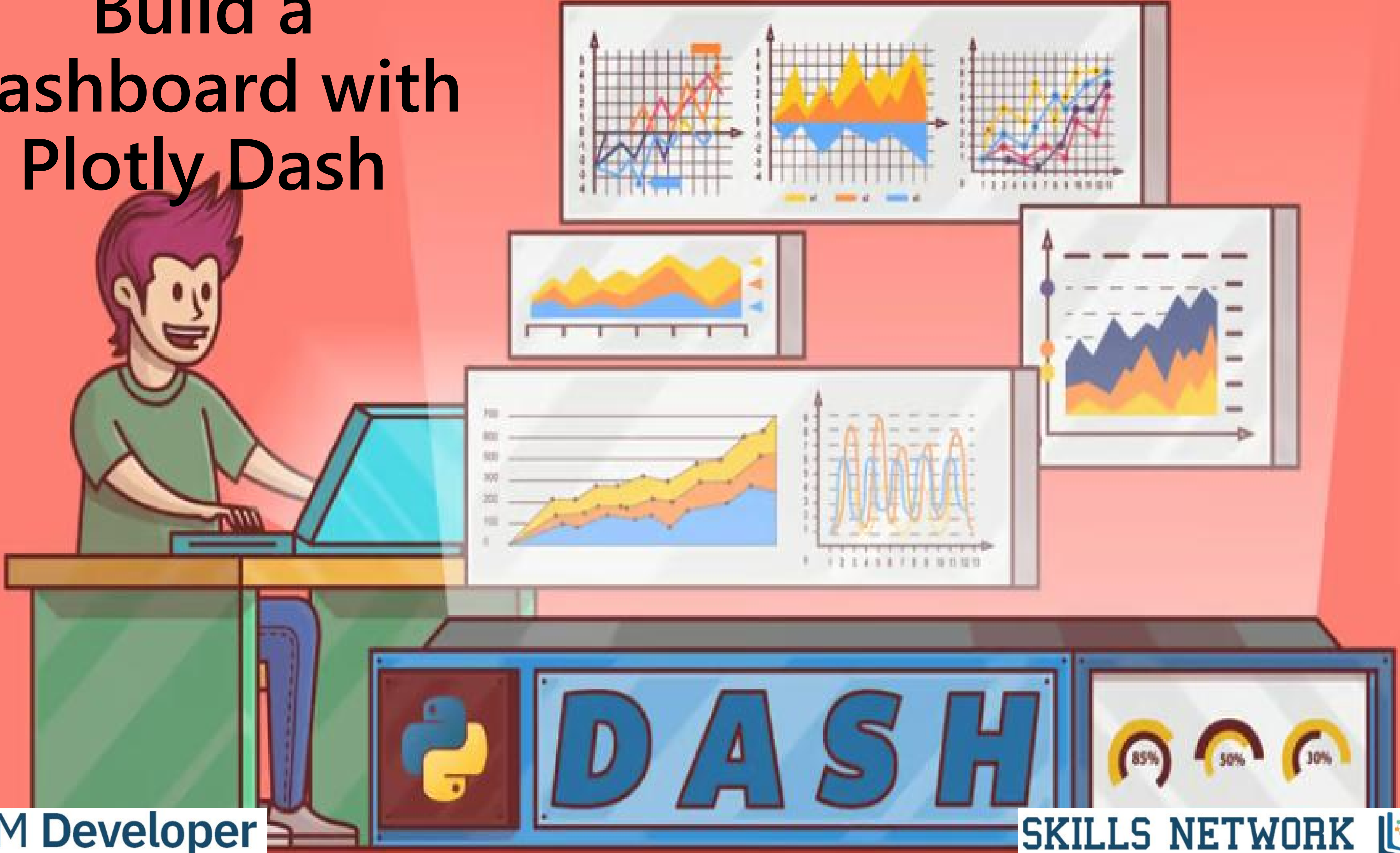






Is CCAFS SLC-40in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

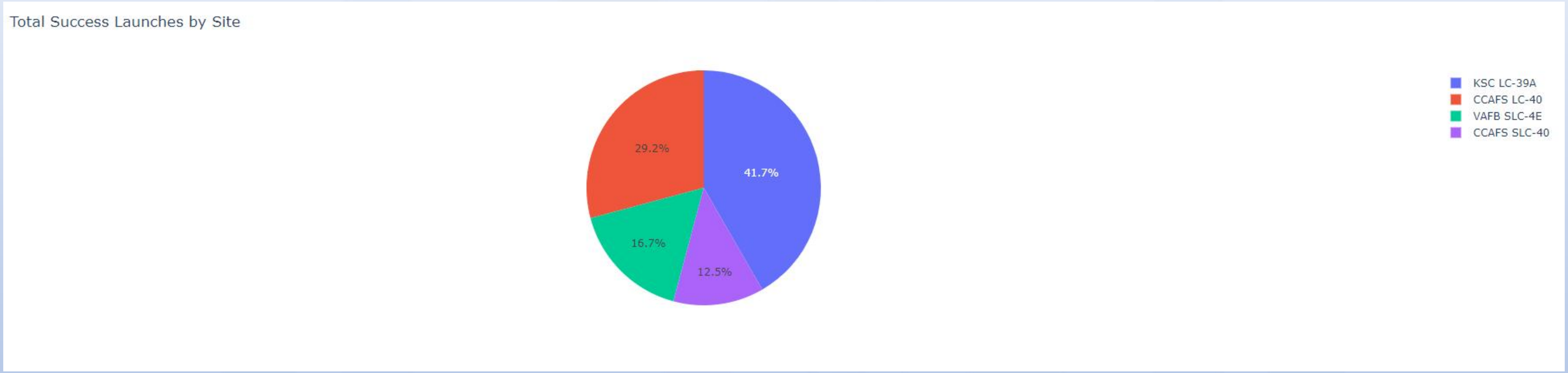
Build a Dashboard with Plotly Dash



IBM Developer

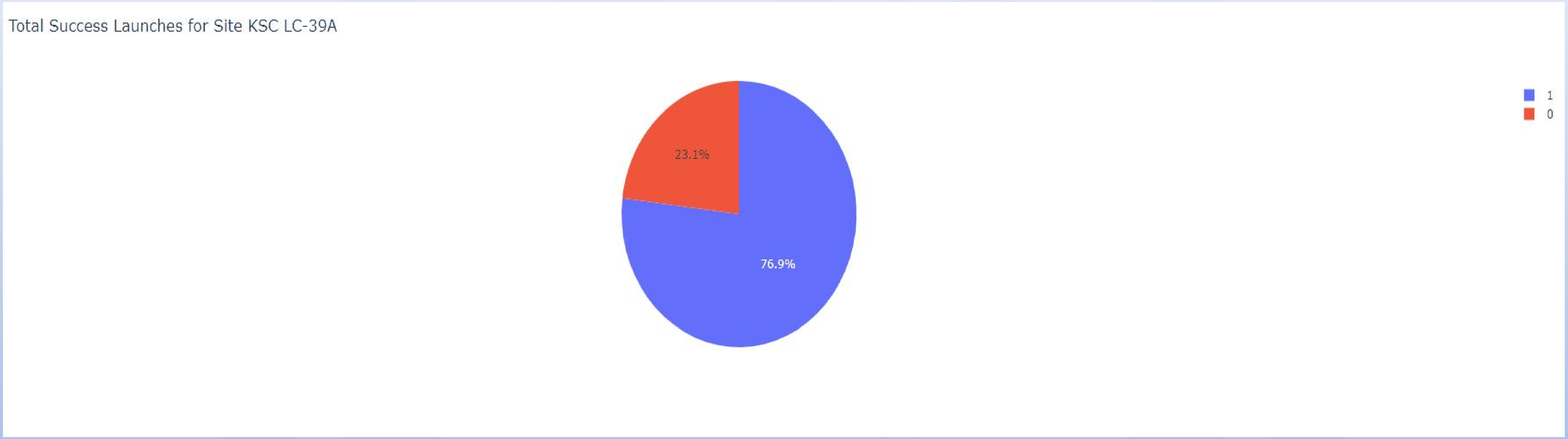
SKILLS NETWORK

Dashboard - Total Success by Site



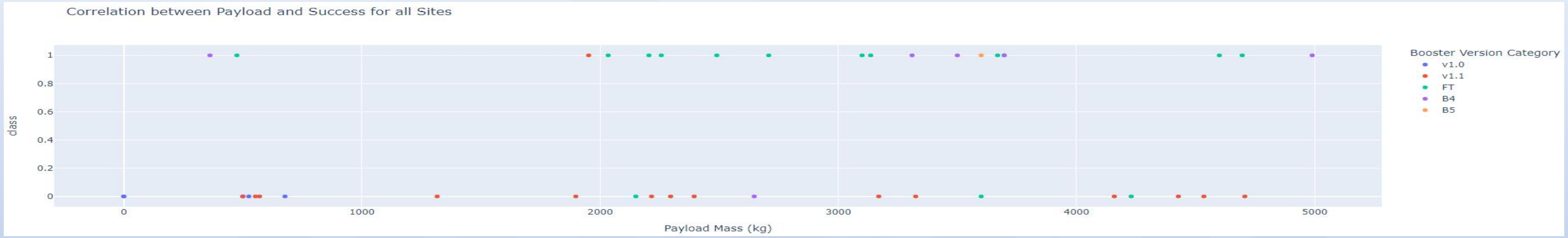
We see that KSC LC-39A has the best successrate of launches.

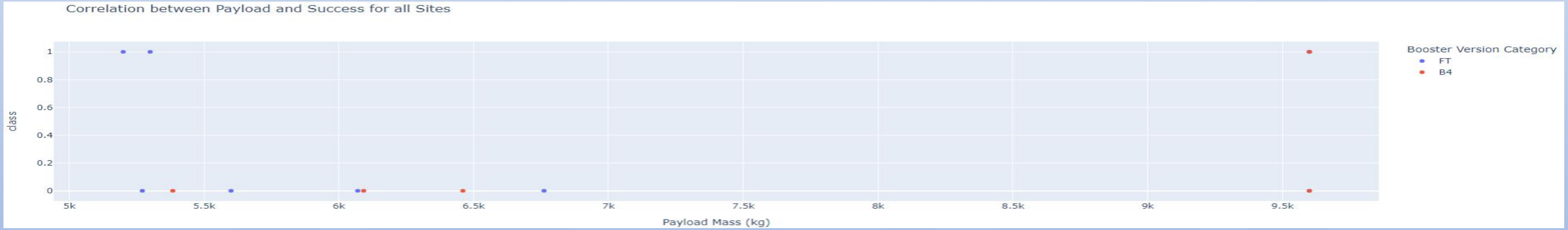
Dashboard - Total Success Launches for Site KSC LC-39A



We see that KSC LC-39A has achieved a 76.9%success ratewhile getting a 23.1% failure rate.

Dashboard - Payload Mass vs Outcome for All Sites with Different Payload Mass Selected





First graph is for Low Weighted Payloads which is 0-5000 kg and the second graph is for High Weighted Payloads which is for 5000-10000 kg.
Low Weighted Payloads hace better success rate than High Weighted Payloads

IBM Developer

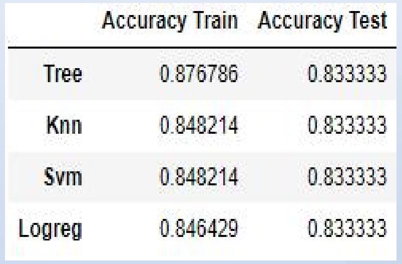
SKILLS NETWORK

Predictive Analytics

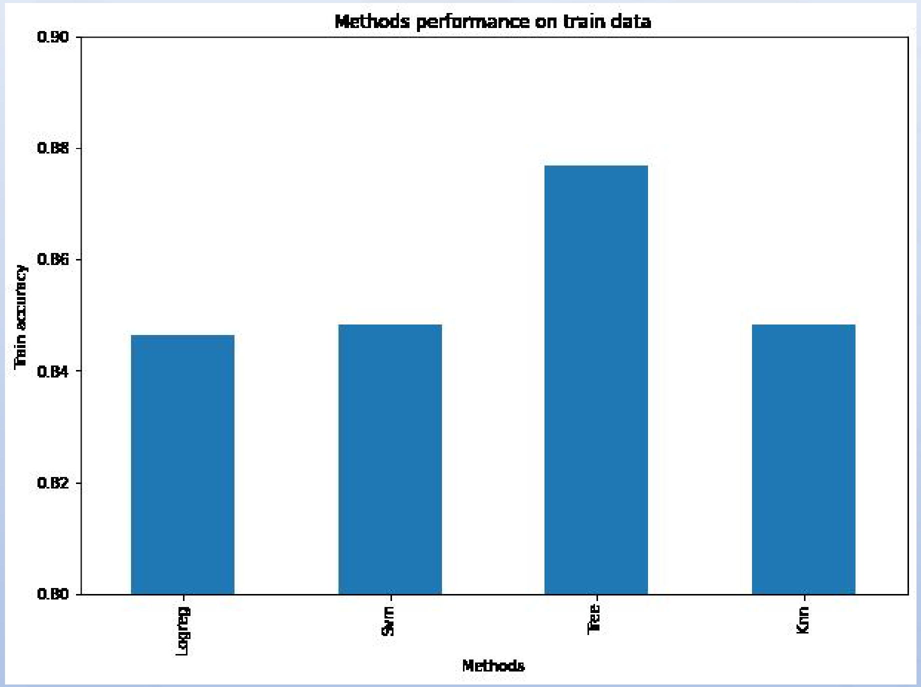
IBM Developer

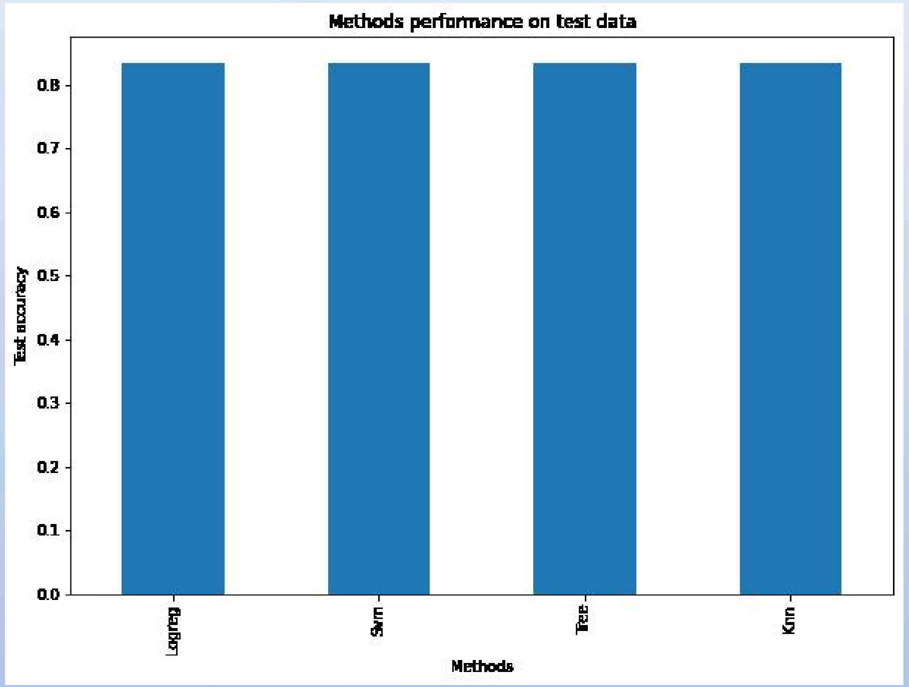
SKILLS NETWORK

Classification Accuracy



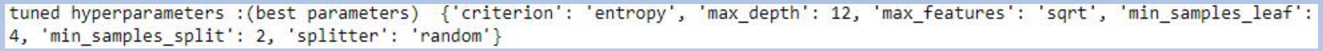
	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333





For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

Decision Tree Parameters

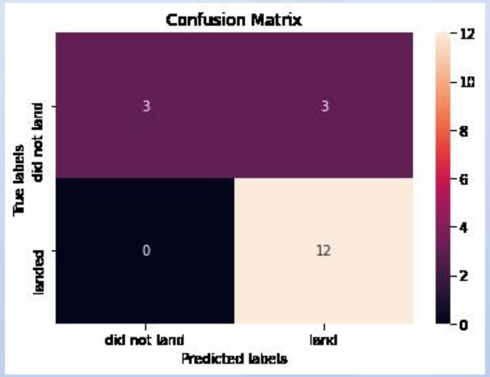


```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```

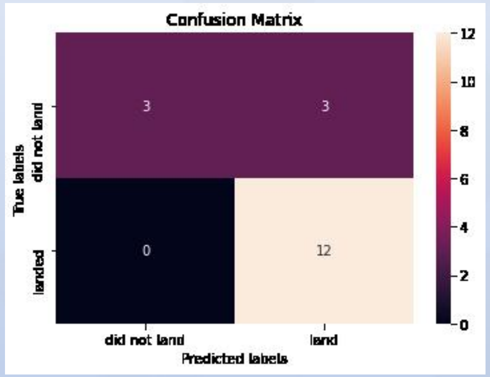
IBM Developer SKILLS NETWORK

Confusion Matrix

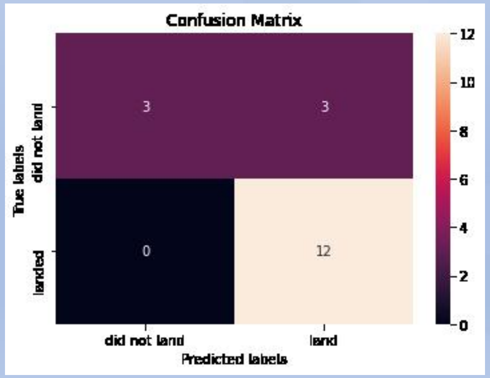
Logistic regression



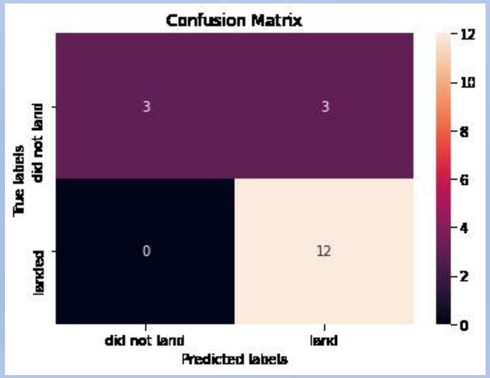
DecisionTree



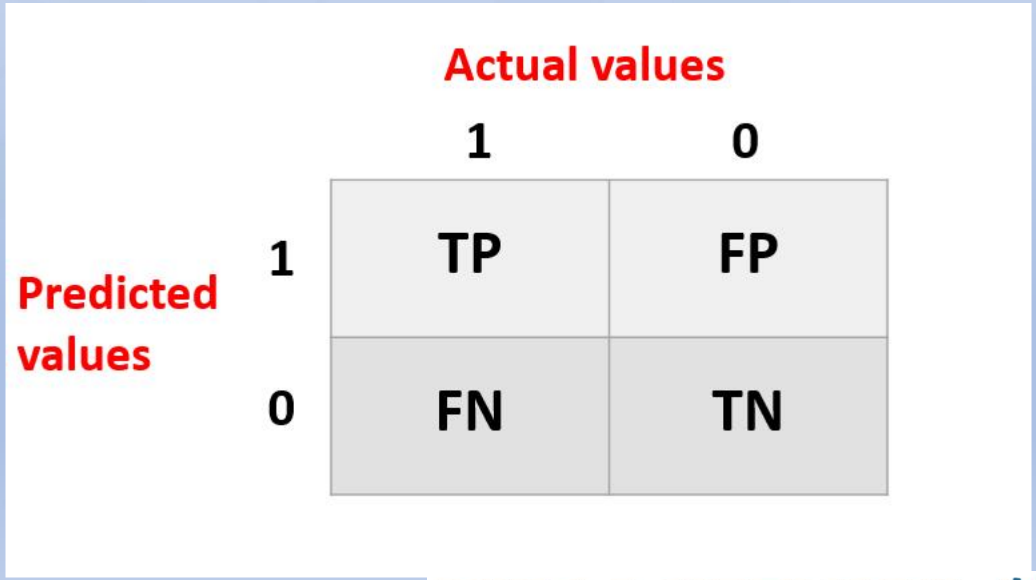
KNN



SVM



As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.



		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especiallythe number of previouslaunches.Indeed,we can assume that there has been a gain in knowledgebetweenlaunchesthat allowed to go from a launchfailure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the bestlaunchsite).To get an answer to this problem, we could obtain atmosphericor other relevant data.
- For this dataset, we choose the Decision Tree Algorithm a s the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

IBM Developer

SKILLS NETWORK