SPACEX FALCON 9 ANALYSIS

Brian Jalleh

02/08/2022

IBM Developer

SKILLS NETWORK

OUTLINE

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

EXECUTIVE SUMMARY

- Methodologies
  - Data Collection with API calls
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Map with Folium
  - Interactive Dashboard with Plotly Dash
  - Predictive Analysis (Classification)

- Results
  - Exploratory Data Analysis results
  - Interactive analytics results
  - Predictive Analysis results

INTRODUCTION

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

- Problems to be answered
  - The best model to be used to predict if the first stage will successfully
  - Conditions to ensure the best successful landing rate

METHODOLOGY

DATA COLLECTION WITH API CALLS

1) Requesting launch data from SpaceX API

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

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

2) Decoding response content as a Json and normalising

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

3) Cleaning the data

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

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

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

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

4) Combining the columns into a dictionary to construct our dataset

```
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}
```

```
# Create a data from launch_dict
launch_df = pd.DataFrame(launch_dict)
```

5) Replacing Missing Values with Mean

```
# Calculate the mean value of PayloadMass column
mean_payloadmass = data_falcon9["PayloadMass"].mean()
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].replace(np.nan, mean_payloadmass, inplace = True)
```

6) Exporting the dataset into a CSV file

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

DATA COLLECTION WITH WEB SCRAPING

1) Requesting Falcon 9 Launch Wiki page from its URL

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

2) Creating a BeautifulSoup object from the HTML response

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

3) Extracting all column names from the HTML table header

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

```
first_launch_table = html_tables[2]
```

```
column_names = []

for th in first_launch_table.find_all("th"):
    name = extract_column_from_header(th)
    if ((name != None) and (len(name) > 0)):
        column_names.append(name)
```

4) Creating a dictionary to store column values

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

5) Extracting the data in the table

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

6) Exporting the dataset into a CSV file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

IBM Developer

SKILLS NETWORK

DATA WRANGLING

1) Calculating the number of launches on each site

```
df["LaunchSite"].value_counts()
```

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

2) Calculating the number and occurrence of each orbit

```
df["Orbit"].value_counts()
```

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
HEO	1
GEO	1
ES-L1	1
SO	1

3) Calculating the number and occurrence of mission outcome per orbit type

```
df["Outcome"].value_counts()
```

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
None ASDS	2
False Ocean	2
False RTLS	1

4) Creating a set of outcomes for unsuccessful landings for the second stage

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

5) Creating a landing outcome label from Outcome column

```
landing_class = []

for i in df["Outcome"]:
    if (i in bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

6) Determining the success rate

```
df["Class"].mean()
```

```
0.6666666666666666
```

6) Exporting the dataset into a CSV file

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

EXPLORATORY DATA ANALYSIS WITH SQL

- Loading data into a DB2 instance and executing SQL queries to find answers to the following:
  - The names of the unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'KSC'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The date where the successful landing outcome in drone ship was achieved
  - The names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes
  - The names of the booster versions which have carried the maximum payload mass. Use a subquery
  - The records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017
  - The count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION

- Scatter graphs:
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit Type
  - Payload vs Orbit Type
- Bar graph:
  - Orbit Type vs Success Rate
- Line graph:
  - Yearly Launch Success

INTERACTIVE MAP WITH FOLIUM

- The following objects were added to the map
  - Markers for all launch sites on the map
  - Markers for the success/failed launches for each site on the map
  - The distances between a launch site to its proximities
- The following questions were answered
  - Are the launch sites in close proximity to railways? Yes
  - Are the launch sites in close proximity to highways? Yes
  - Are the launch sites in close proximity to the coastline? Yes
  - Do launch sites keep a certain distance away from cities? Yes

INTERACTIVE DASHBOARD WITH PLOTLY DASH

- The following objects were added to the dashboard
  - A Launch Site Drop-down Input Component
  - A pie chart for Total Successful Launches By Site
  - A Range Slider to Select Payload
  - A scatter graph for the correlation between the Selected Payload vs Success By Site
- The following questions were answered
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version has the highest launch success rate?

PREDICTIVE ANALYSIS (CLASSIFICATION)

1) Creating a column for the classification

```
Y = data["Class"].to_numpy()
```

2) Standardizing the data

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

3) Splitting the data into training data and test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 2)
```

4) The models are trained and hyperparameters are selected using the function GridSearchCV

5) A bar graph displaying the Test Accuracy vs Method

- The following question was answered
  - Find the best Hyperparameter for SVM, Classification Trees, and Logistic Regression
    - The method that performs the best using test data

The sample size of the test data is not large enough to distinguish the methods from each other as they all are shown to have the same test accuracy.

RESULTS

EDA WITH SQL

ALL UNIQUE LAUNCH SITES

Query

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

Result

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

- DISTINCT operator only selects unique sites

LAUNCH SITE NAMES BEGINNING WITH ‘KSC’

Query

```
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE "KSC%"
LIMIT 5
```

Result

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

- Only 5 results were shown due to the LIMIT operator
- The LIKE operator and % sign allows only names starting with ‘KSC’ to be called

TOTAL PAYLOAD MASS CARRIED BY NASA (CRS)

Query

```
SELECT SUM(PAYLOAD_MASS__KG_) as total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER == "NASA (CRS)"
```

Result

total_payload_mass_kg
45596

- The 'as' operator renames the column name to the assigned column name
- The SUM operator adds all the masses in the PAYLOAD_MASS__KG_ column

AVERAGE PAYLOAD MASS CARRIED BY BOOSTER VERSION F9 V1.1

Query

```
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE BOOSTER_VERSION == "F9 v1.1"
```

Result

AVG(PAYLOAD_MASS__KG_)
2928.4

- The WHERE operator selects the tuples which only have 'F9 v1.1' as their BOOSTER_VERSION
- The AVG operator finds the average of the masses in the PAYLOAD_MASS__KG_ column

FIRST SUCCESSFUL LANDING DATE

Query

```
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING_OUTCOME == 'Success (drone ship)'
```

Result

first_successful_landing_date
06-05-2016

- The WHERE Operator selects the tuples which only have ‘Success (drone ship)’ as their LANDING OUTCOME
- The MIN operator finds the earliest date in the DATE column

NAMES OF BOOSTERS WHICH HAVE SUCCESS IN THE GROUND PAD AND HAVE A PAYLOAD MASS OF BETWEEN 4000 AND 6000

Query

```
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE (LANDING_OUTCOME == "Success (ground pad)")
        AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

Result

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

- The BETWEEN operator selects the tuples which have a payload mass between 4000 and 6000

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

Query

```
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

Result

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- The GROUP BY

- The COUNT(*) counts each MISSION_OUTCOME

NAMES OF BOOSTER VERSIONS WHICH HAVE CARRIED THE MAXIMUM PAYLOAD MASS

Query

```
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_)
                                                FROM SPACEXTBL)
```

Result

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

- The IN operator compares two tables to each other and only returns the tuples which are present in both tables

SUCCESSFUL LANDING OUTCOMES IN 2017

Query

```
SELECT SUBSTR(DATE,4,2) AS 'Month', LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE (SUBSTR(DATE,7,4)='2017') AND (LANDING_OUTCOME = "Success (ground pad)")
```

Result

Month	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

- The SUBSTR operator obtains the substring of the attribute

SUCCESSFUL LANDING OUTCOMES BETWEEN 04-06-2010 AND 20-03-2017

Query

```
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS "Count"
FROM SPACEXTBL
WHERE (LANDING_OUTCOME LIKE "Success%") AND (DATE BETWEEN "04-06-2010" AND "20-03-2017")
GROUP BY LANDING_OUTCOME
ORDER BY Count DESC
```

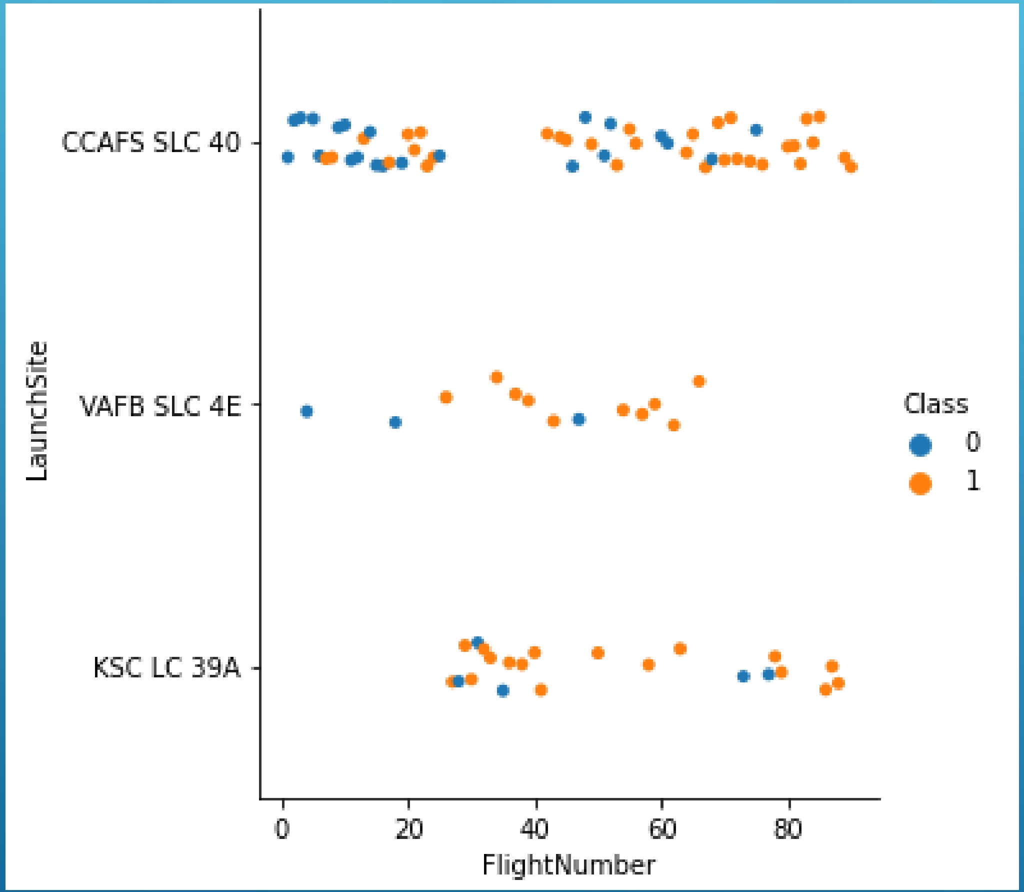
Result

Landing_Outcome	Count
Success	20
Success (drone ship)	8
Success (ground pad)	6

- The ORDER BY operator orders the output table by the Count of each Landing Outcome

EDA WITH VISUALIZATION

FLIGHT NUMBER VS LAUNCH SITE

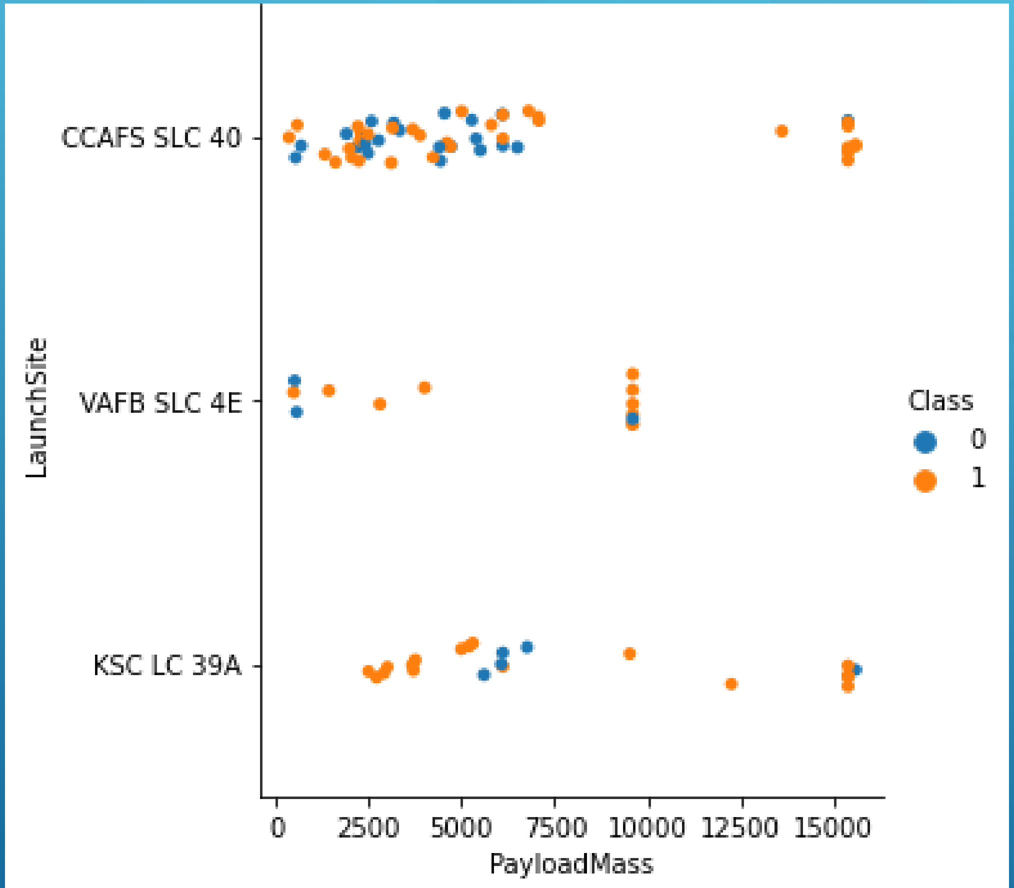


- The Blue (0) represents failed launches and the Orange (1) represents successful launches
- The graph describes an **increase in successful launches as the number of flights increases**
- The number of successful launches is shown to increase for all launch sites after the 30th flight

IBM Developer

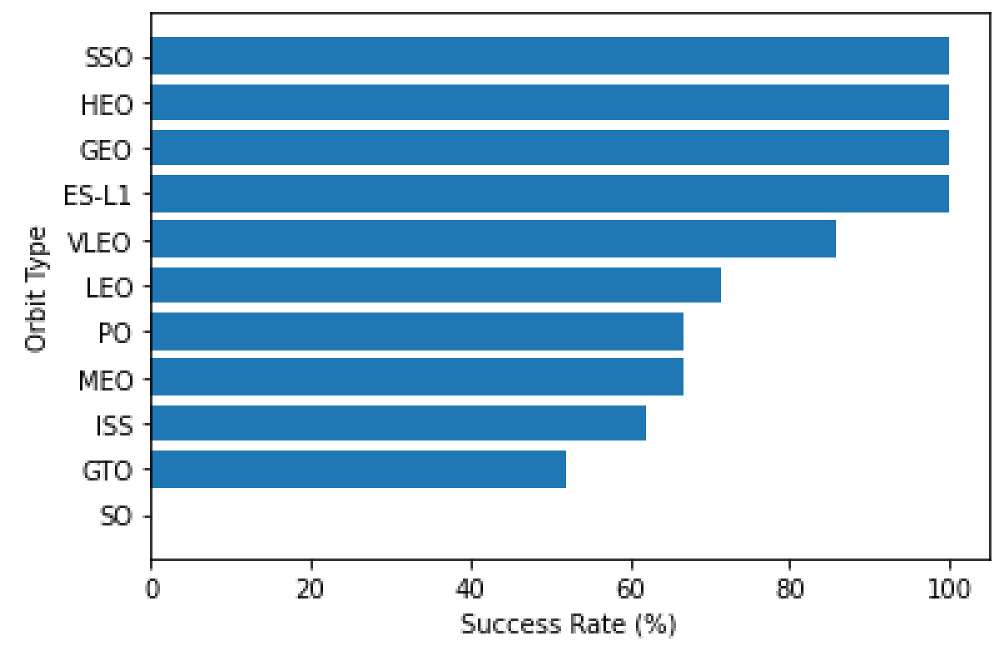
SKILLS NETWORK

PAYLOAD MASS VS LAUNCH SITE



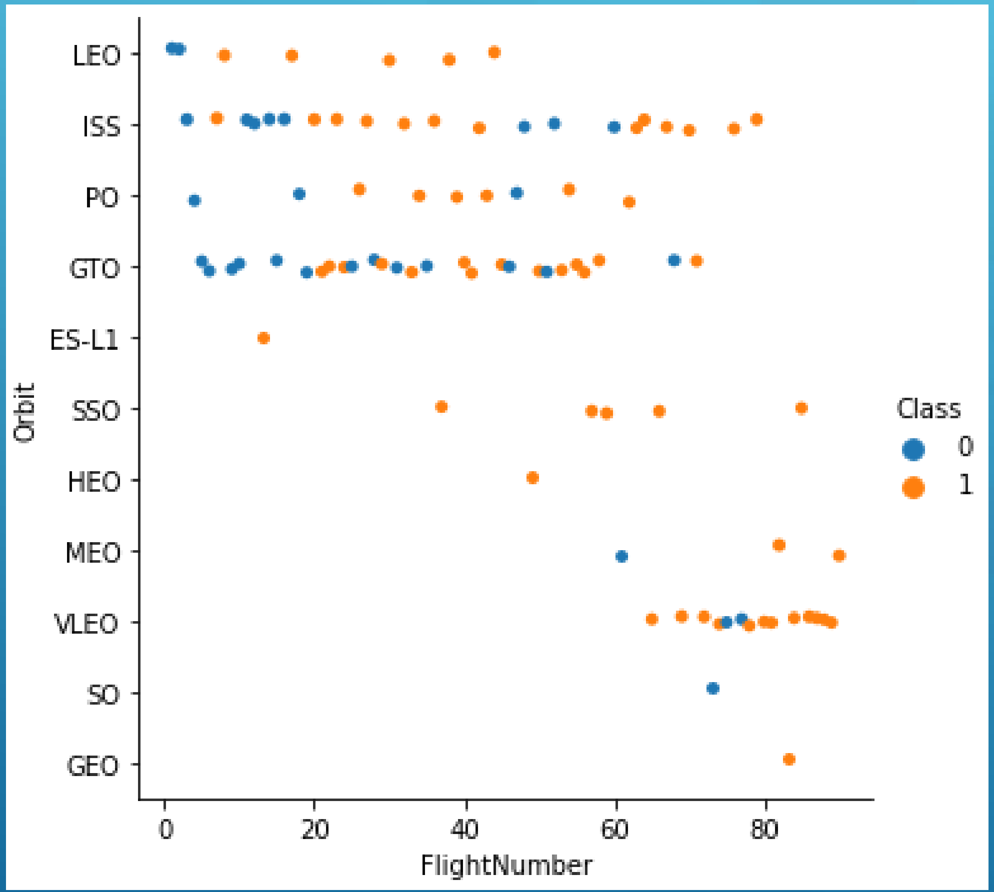
- The Blue (0) represents failed launches and the Orange (1) represents successful launches
- The graph describes an **increase in successful launches for the 'VAFB SLC 4E' launch site as the payload mass increases**
- More information is needed for the other launch sites to determine whether there is correlation between payload and launch site

SUCCESS RATE VS ORBIT TYPE



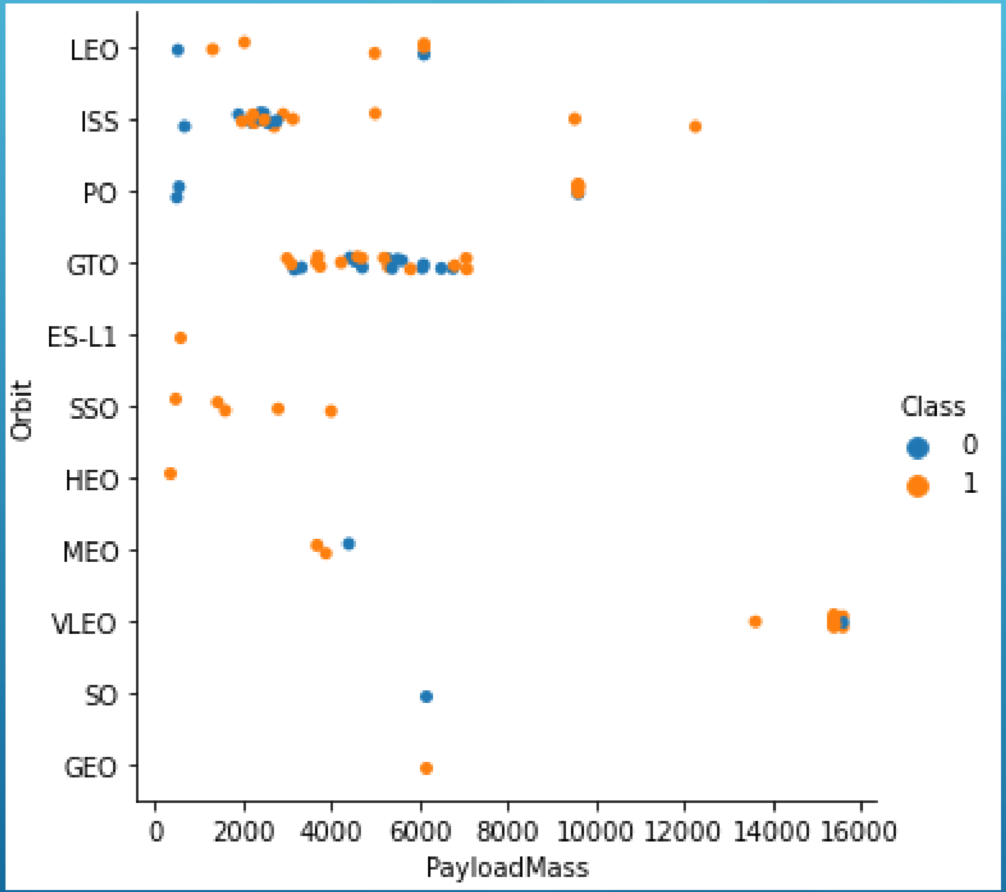
- Orbit types **SSO, HEO, GEO, and ES-L1 have the highest success rates (100%)**
- Orbit types VLEO, LEO, PO, MEO, and ISS have a success rate >= 50%
- Orbit types GTO, and SO have a success rate of <= 50%

FLIGHT NUMBER VS ORBIT TYPE



- The Blue (0) represents failed launches and the Orange (1) represents successful launches
- The orbit type **LEO is shown to have successful launches as the flight number increases**
- There is no correlation between flight number and success rate for the orbit type GTO
- In most cases, there seems to be positive correlation between orbit type and flight number

PAYLOAD MASS VS ORBIT TYPE

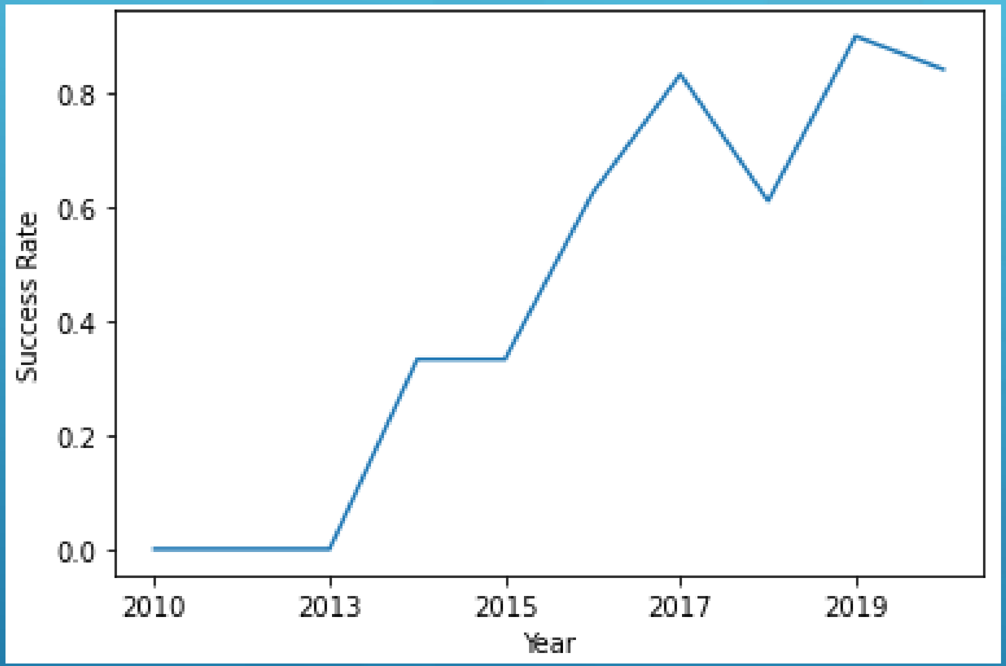


- The Blue (0) represents failed launches and the Orange (1) represents successful launches
- With heavier payloads the **success rate tends to increase for the orbit types LEO, ISS, and PO**
- There is no correlation between payload mass and success rate for the orbit type GTO

IBM Developer

SKILLS NETWORK

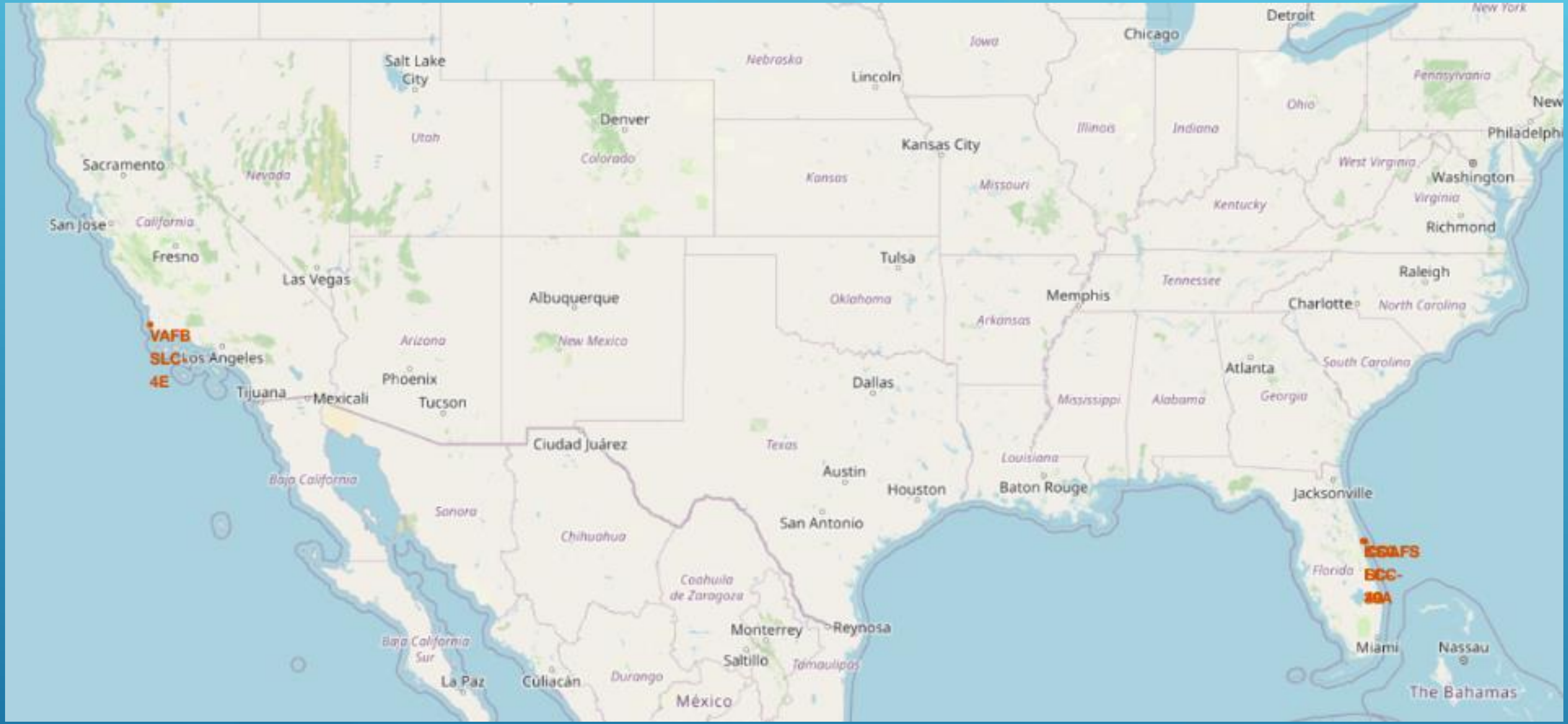
YEARLY LAUNCH SUCCESS RATE



- The success rate is shown to increase since 2013, and has kept increasing since
- The success rate dropped by 20% in 2018
- The most recent success rate is around 80%

INTERACTIVE MAP WITH FOLIUM

LAUNCH SITE LOCATIONS

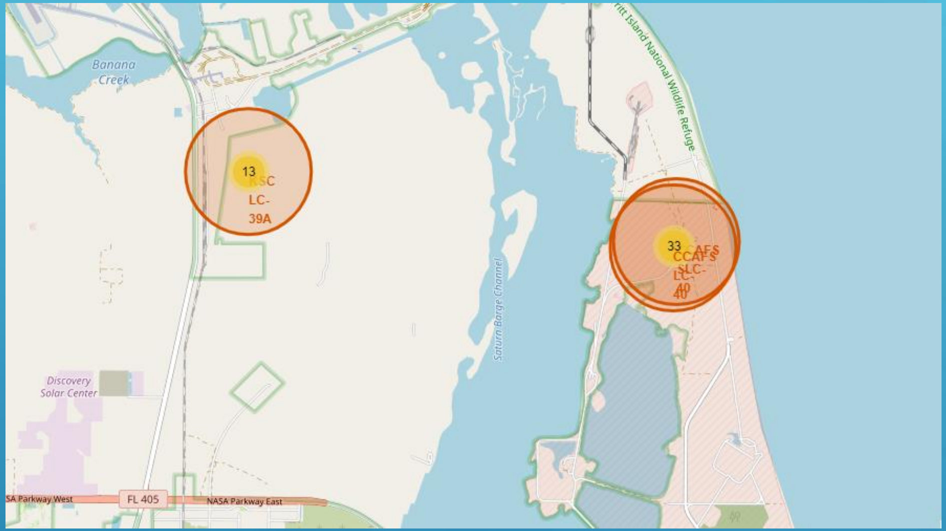


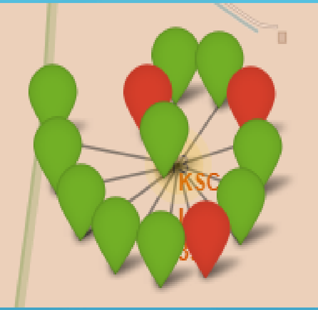
- All the launch sites have been marked on the map

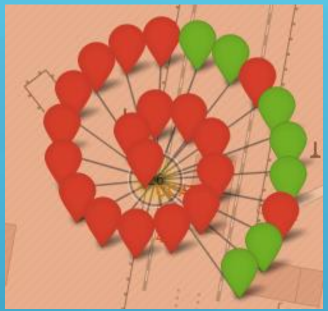
IBM Developer

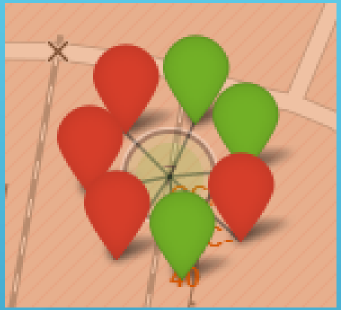
SKILLS NETWORK

SUCCESS/FAILED LABELED LAUNCHES







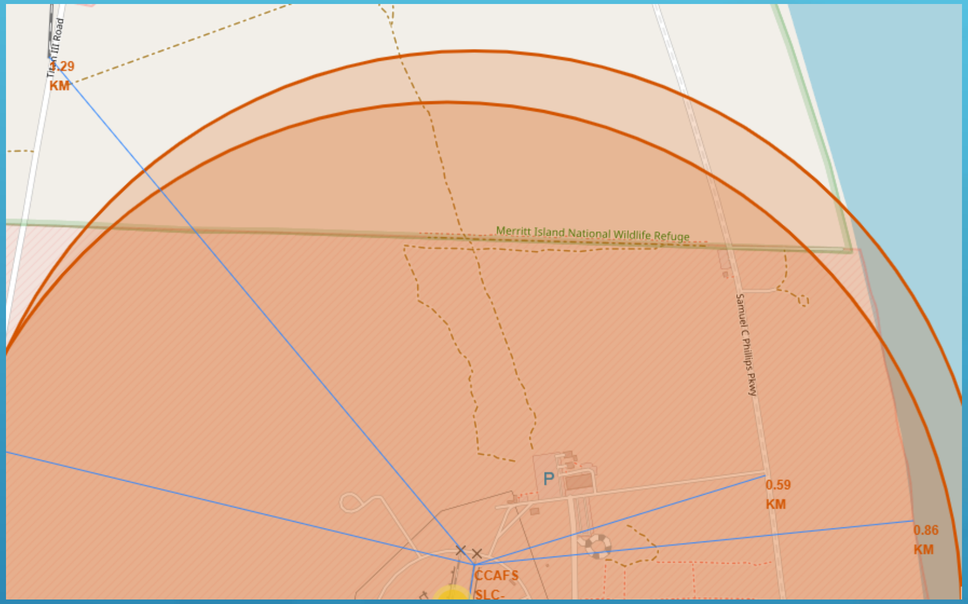


Launch sites in Florida

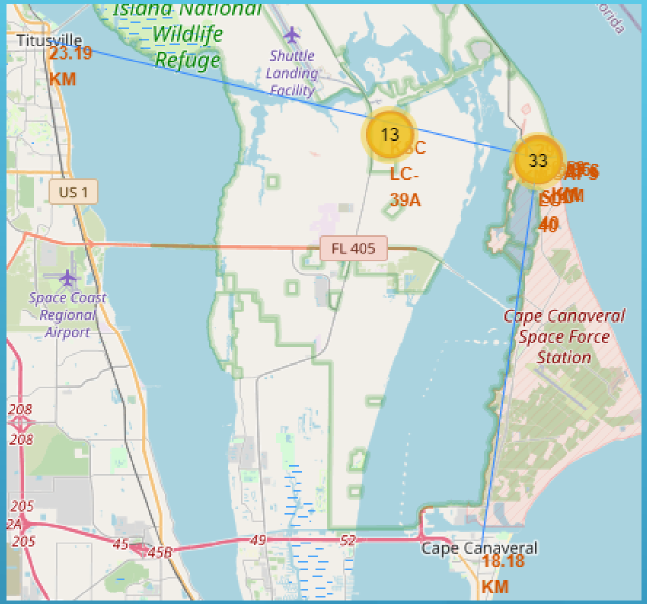
Launch site in California

All the successful and failed launches have been marked on the map as green or red respectively

LAUNCH SITE PROXIMITIES



- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to railways? Yes



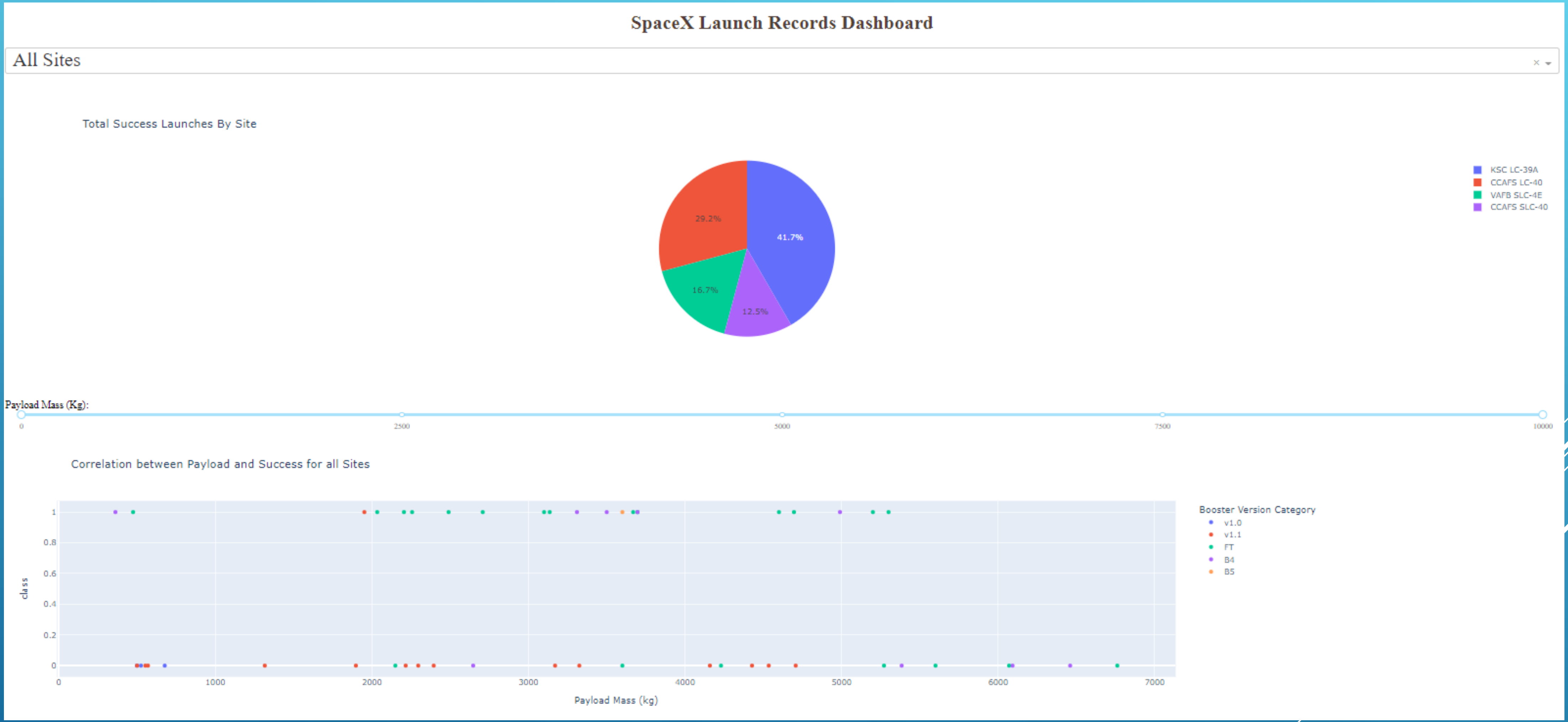
- Do launch sites keep a certain distance away from cities? Yes

IBM Developer

SKILLS NETWORK

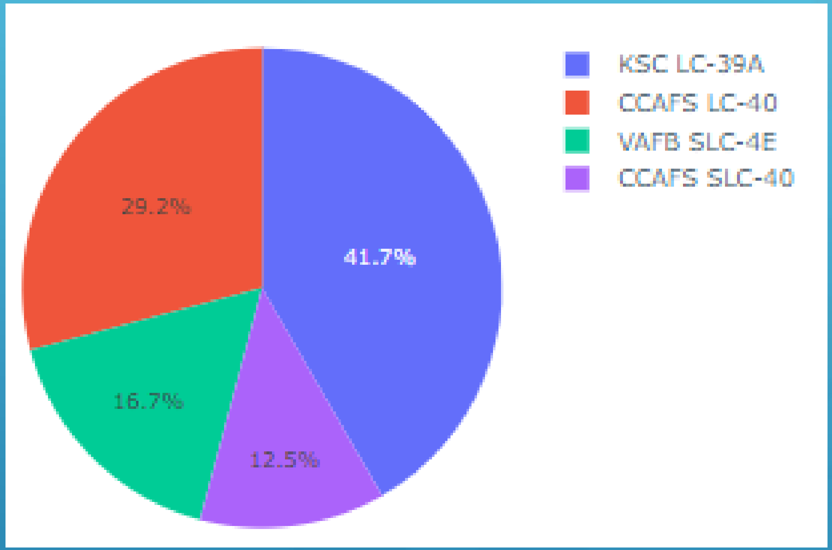
INTERACTIVE DASHBOARD WITH PLOTLY DASH

DASHBOARD



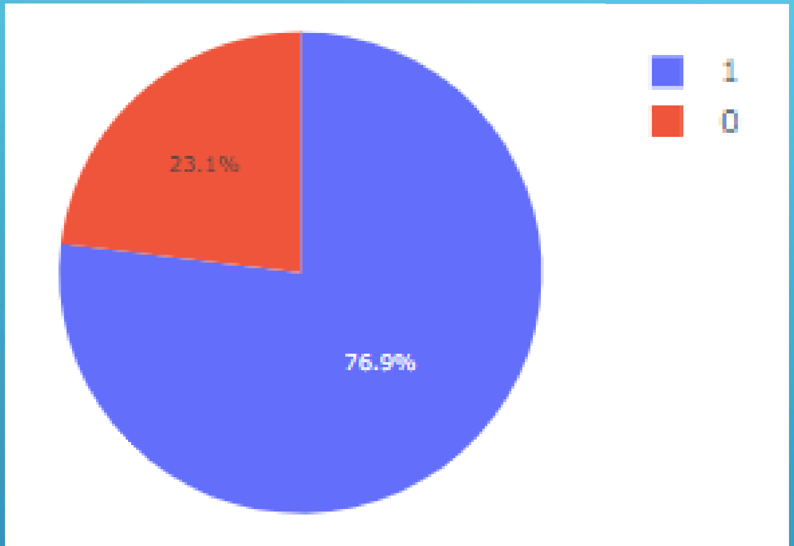
SpaceX Launch Records Dashboard
All Sites
Total Success Launches By Site
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload Mass (Kg):
0
2500
5000
7500
10000
Correlation between Payload and Success for all Sites
class
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

SITE WITH THE LARGEST SUCCESSFUL LAUNCHES



- KSC LC-39A has the most successful launches

SITE WITH THE HIGHEST LAUNCH SUCCESS RATE



- KSC LC-39A has the highest launch success rate with a 76.9% success rate
- The other launch sites CCAFS LC-40, VAFB SLC-4E, and CCAFS SLC-40 have 73.1%, 60%, and 57.1% respectively

PAYLOAD RANGE WITH THE HIGHEST LAUNCH SUCCESS RATE

- The payload range 0 – 5000 has the highest launch success rate

PAYLOAD RANGE WITH THE LOWEST LAUNCH SUCCESS RATE

- The payload range 5000 – 10000 has the lowest launch success rate

IBM Developer

SKILLS NETWORK

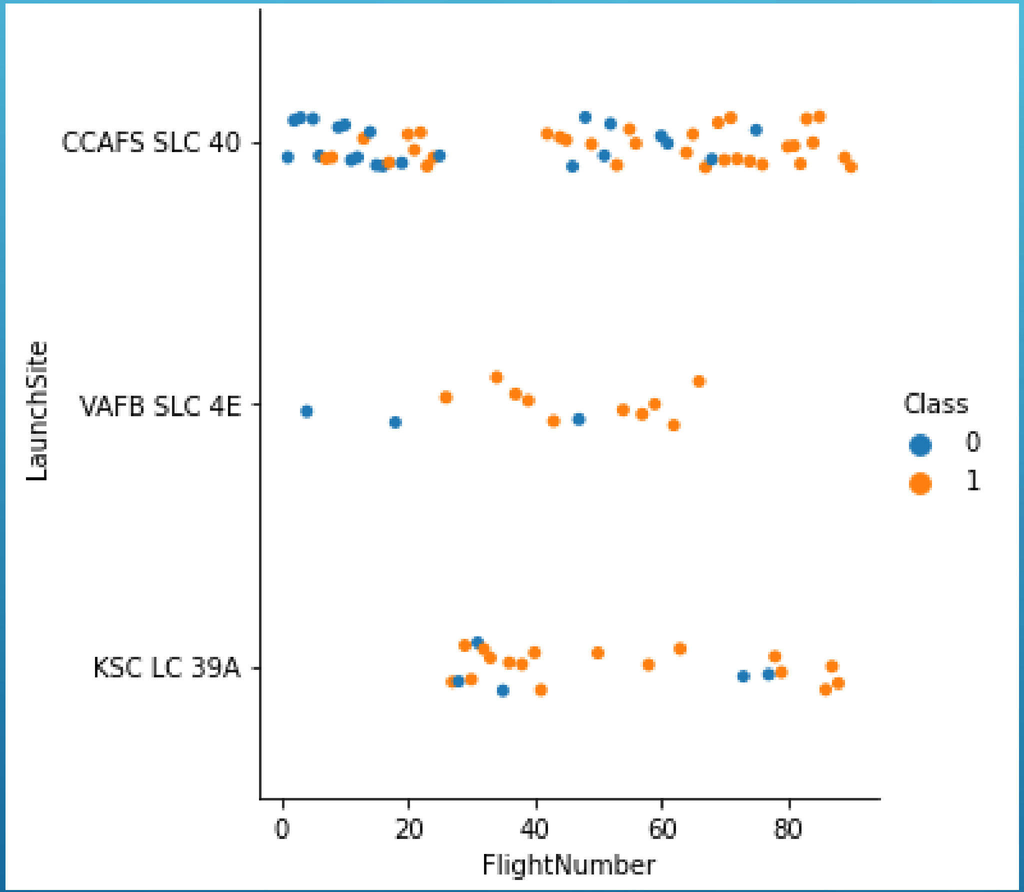
THE BOOSTER VERSION WITH THE HIGHEST SUCCESS RATE

- The booster version FT has the highest success rate
- Given that the dataset is small, the reliability of this outcome is quite low

IBM Developer

SKILLS NETWORK

FLIGHT NUMBER VS LAUNCH SITE



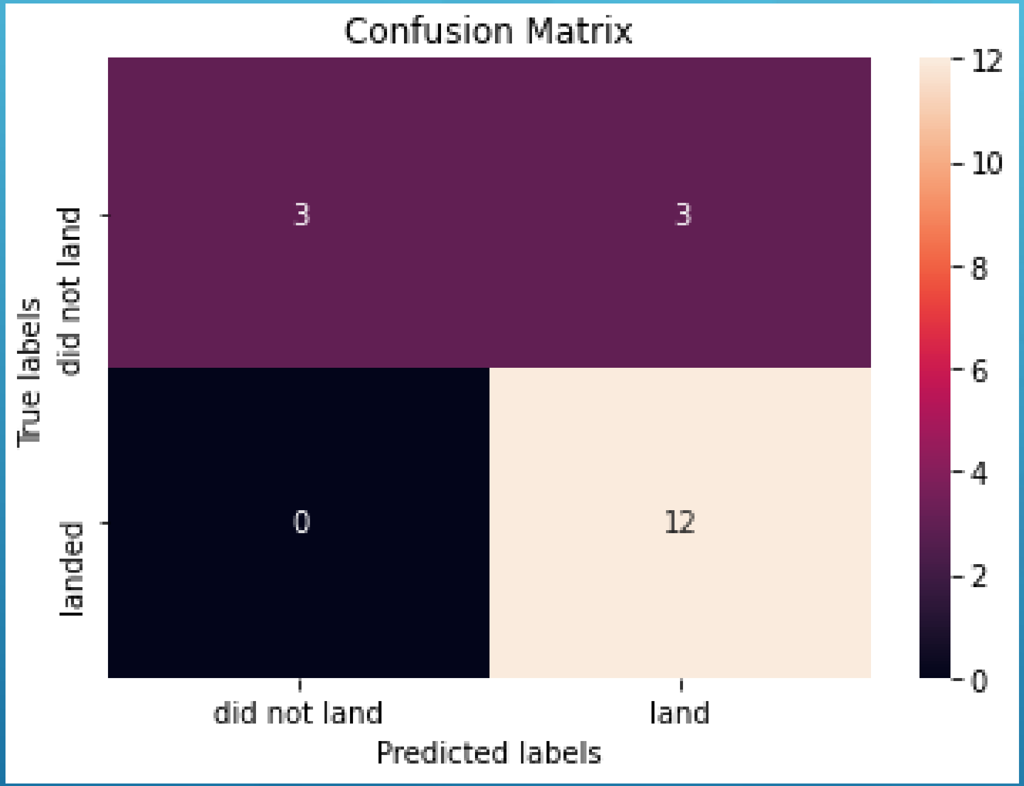
- The Blue (0) represents failed launches and the Orange (1) represents successful launches
- The graph describes an **increase in successful launches as the number of flights increases**
- The number of successful launches is shown to increase for all launch sites after the 30th flight

IBM Developer

SKILLS NETWORK

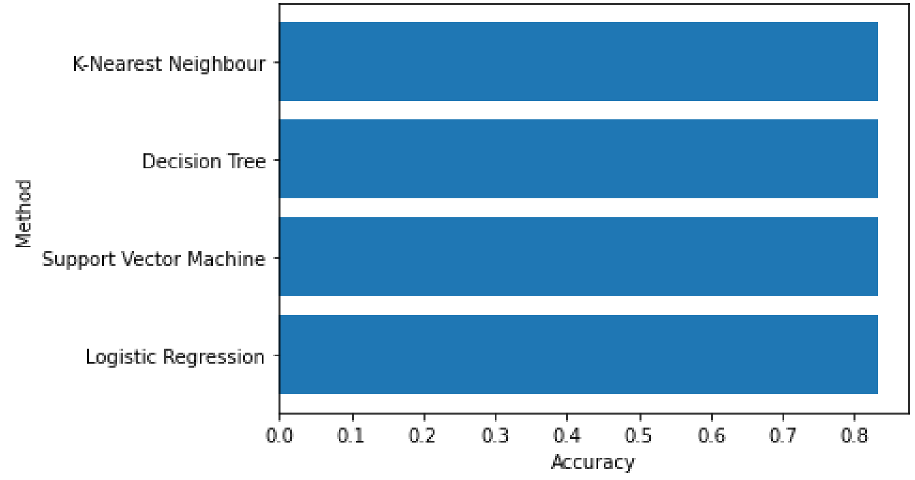
PREDICTIVE ANALYSIS (CLASSIFICATION)

CONFUSION MATRIX



- All models created displayed the same confusion matrix
- The main problem is the false positives as 3 landings were predicted to land but in actuality, they didn't
- Overall, the models are pretty good at predicting which landings will be successful (80% correct prediction) but they aren't good at predicting which landings will fail (50% correct prediction)

MODEL ACCURACY



	Accuracy
Logistic Regression	0.833333
Support Vector Machine	0.833333
Decision Tree	0.833333
K-Nearest Neighbour	0.833333

- All models had an accuracy of 83.33% when tested with the test set
- More data is needed in order to differentiate the models from each other as the sample size is quite small

CONCLUSION

- Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%)
- KSC LC-39A has the most successful launches and the highest launch success rate
- There is usually positive correlation between the number of flights and the launch success rate
- The sample size of the data must increase in order to differentiate the models from each other to provide a reliable answer to which model performs best

APPENDIX

- GitHub URL
- EdX Course Link

IBM Developer

SKILLS NETWORK

THANK YOU