Space Race,
A Data Science Approach

An analysis by: Harold Martinez

Location: Mexico

Date: July-22

Outline

1. Executive Summary
2. Introduction
3. Methodology
    a) Data Collection (API – Web Scraping)
    b) Exploratory Data Analysis
    c) Interactive Visual Analytics
    d) Predictive Analytics
4. Results
    a) Visualization – Charts
    b) Map / Dashboard
    c) Predictive Analytics
5. Conclusion
6. Appendix

1. Executive Summary

Space X data from 2010 to date was analyzed on regards the different launches that have been executed for the company at different locations. This information includes all the booster types, and do not discriminate on the Mission Status. After gathering the data and handling it to be Machine Learning ready, **4 algorithms** (KNN, Decision Tree, Support Vector Machine (SVM), Logistic Regression) **were trained, fit, and evaluated.** The Accuracy and Confusion Matrix for each was calculated and the best algorithm was selected. There was a tie between KNN, Decision Tree, and SVM. However, it is considered that **Decision Tree is the best algorithm for the task in hand with an accuracy of 95% upon the test dataset.**

2. Introduction

Space X competitive advantage lays in the fact that rockets can be re-utilized and launch costs are cut from 165M USD (competitors) to 62M USD. For such reason, the success of Stage 1 is critical. **The goal of this analysis is to create a Machine Learning pipeline** using the KNN, Decision Tree, Support Vector Machine (SVM), and Logistic Regression algorithms to predict the Stage 1 success. The following questions are proposed:

- **What factors determine landing success?**
- **What are feature interactions on landing success cases?**

IBM Developer SKILLS NETWORK

3. Methodology

The study was conducted using the **5-Step Data Science Methodology.** Starting with the **Business Needs definition,** and then going all the way from **Data Collection** (mainly from SpaceX repositories), **Data Wrangling** (clean-up, binary categorization, data quality analysis, etc.), **Analysis and Model Building** (KNN, Decision Tree, Support Vector Machine (SVM), and Logistic Regression), up to **Results Interpretation** (Model Evaluation and Selection)

Data Analysis Process

5. INTERPRET RESULTS

4. DETERMINE ANALYSIS

3. DATA WRANGLING

2. DATA COLLECTION

1. DEFINE QUESTIONS AND GOALS

During the following slides, the information required to answer the Research Questions will be presented, as well as the results obtained and its interpretation.

Data Sources & General Information

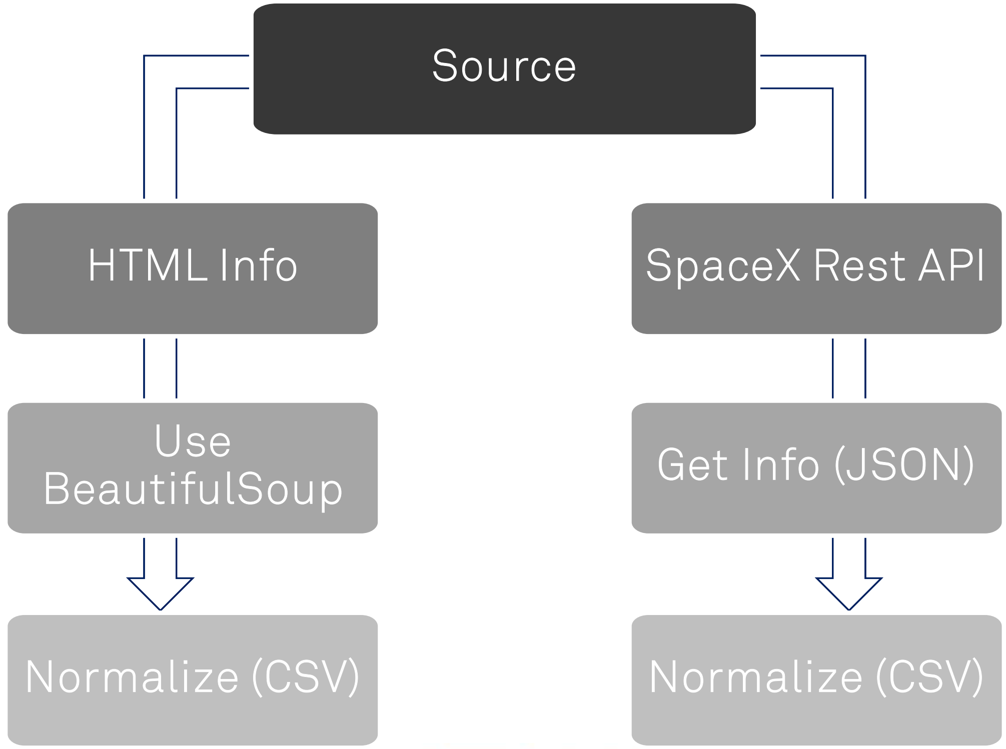
DATA SOURCE

SUBSETS

- SpaceX.csv
- spacex_launch_geo.csv
- spacex_launch_dash.csv

IBM Developer

DATA COLLECTION PROCESS



SKILLS NETWORK

Data Collection: SpaceX API (1 of 3)

01. Get API Response

```
response = requests.get(static_json_url)
response.status_code
```

02. Convert Response to .json

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

03. Custom API Calls

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

Data Collection: SpaceX API (2 of 3)

04. Prepare Dataframe

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

Data Collection: SpaceX API (3 of 3)

05. Filter Data

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data.loc[data['BoosterVersion']!='Falcon 1']
```

06. Cleanup

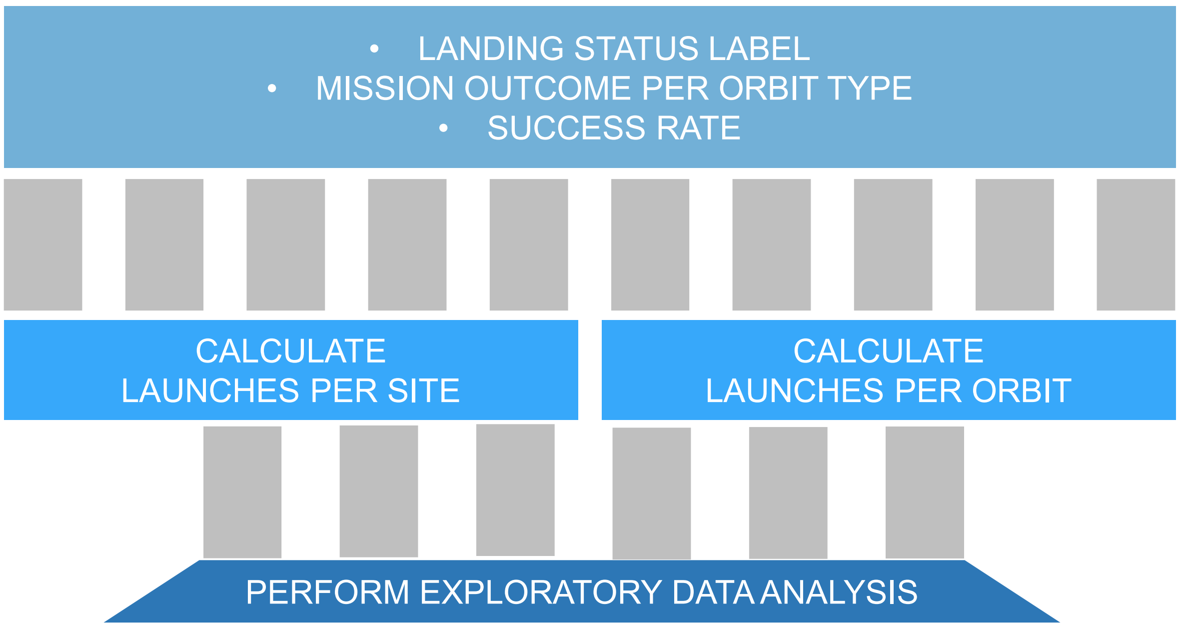
```
# Calculate the mean value of PayloadMass column
PayLoadMass_mean = data_falcon9["PayloadMass"].mean()

# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].replace(np.nan, PayLoadMass_mean, inplace=True)

data_falcon9.isnull().sum()
```

IBM Developer

SKILLS NETWORK

Data Wrangling

OBJECTIVES



ABOUT THE DATA

- Data contains **information for different mission outcomes** for all the available launch sites.
- **Missions are orbit dependent.** This information is also included in the dataset.
- The **success rate is not explicitly accounted for. This** is the parameter of interest and **must be calculated.**

EDA with Data Visualization

SCATTER CHART

- Flight Number Vs.
  - Launch Site
- Payload Mass (kg) Vs.
  - Launch Site
- Orbit Type Vs.
  - Flight Number
  - Payload Mass

BAR GRAPH

- Success Rate Vs.
  - Orbit Type

LINE PLOT

- Success Rate Vs.
  - Launch Year

EDA with SQL

QUESTIONS TO ANSWER

- **Finding unique** launch sites
- Displaying **5 records (only)** where launch site begins with "CCA"
- **Calculate the total payload mass** carried by boosters launched by NASA (CRS)
- **Calculate average payload mass** carried by booster version F9 v1.1
- Listing the boosters that **complied with specific mission parameters**
- **Listing records by** Mission Outcome Status.
- Finding the boosters which **carried the maximum** payload mass amount
- Finding **specific record details for a specific year**
- **Rank mission outcomes** between 2 dates in descending order.

Interactive Analytics: Map

QUESTIONS TO ANSWER

- **Are launch sites near railways?**

→ They are at least 1 km away.

- **Are launch sites near highways?**

→ No, they are not. There are access roads but no major highways

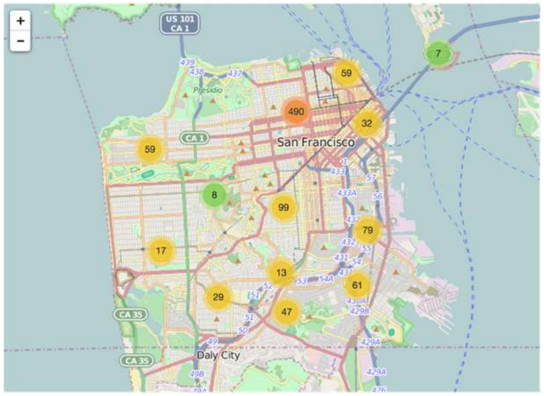
- **Are launch sites near coastlines?**

→ Yes, they are.

- **Do launch sites keep certain distance away from cities?**

→ Yes, they do.

ILLUSTRATIVE ONLY



Interactive Analytics: Dashboard

TECHNOLOGY

- Using Flask and Dash libraries.
- Web based

PIE CHART

- Success Rate
  - For all launch sites
- Success & Failure Rate
  - For each individual site

SCATTER CHART

- Mission Outcome Vs. Payload Mass (kg)
  - For all launch sites
  - For each individual site
  - Filtered by Payload Mass (kg) range

Predictive Analytics

1

- Load datasets to Numpy and Pandas
- Transform
- Split into training / test
- Select machine learning algorithms
- Set parameters per algorithm
- Fit and Train models

2

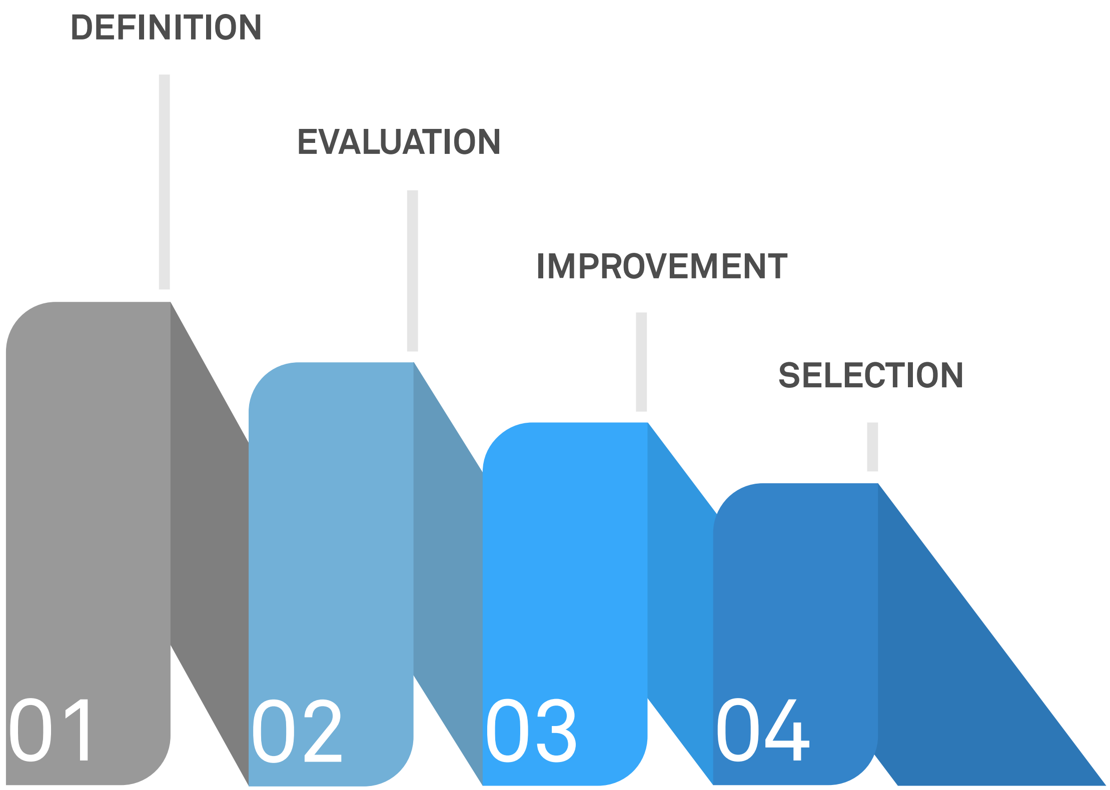
- Check accuracy and tune hyperparameters
- Plot Confusion Matrix

3

- Feature Engineering & Algorithm Tuning

4

- Select best algorithm



IBM Developer

SKILLS NETWORK

4. Results

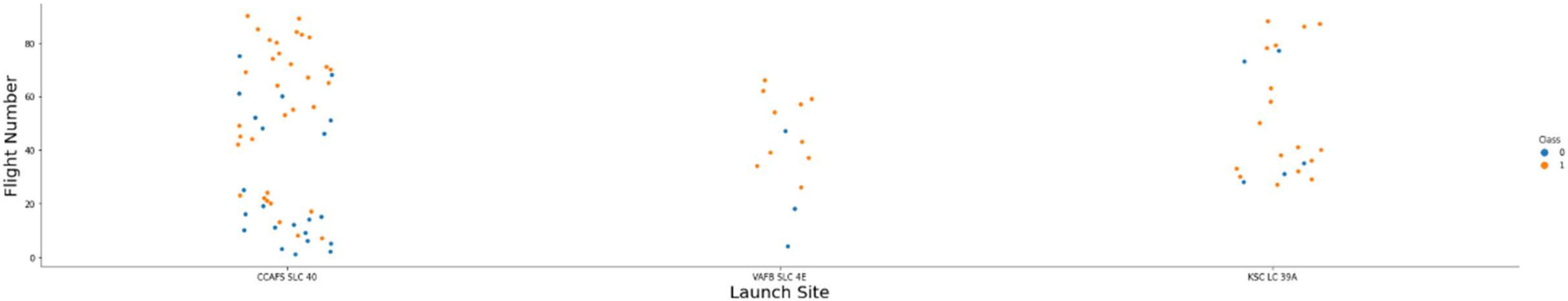
- From the 4 algorithms, there was a triple tie in first place with 95% accuracy between KNN, SVM, and Decision Tree.

- Through the exploratory analysis some additionally hypothesis regarding the impact of the learning curve and non-acquired data arose that might be suitable for increasing the model robustness.

- Some of the variables analyzed don’t seem fit for the decision-making process. (i.e., Orbit Type)

Flight Number vs Launch Site

- Even though, overall success rate is lower than the other sites, net total successes are almost 2x the 2nd best site.



- **Success: (33, 60%)**
- Failure: (22, 40%)
- **Total: 55 Launches**

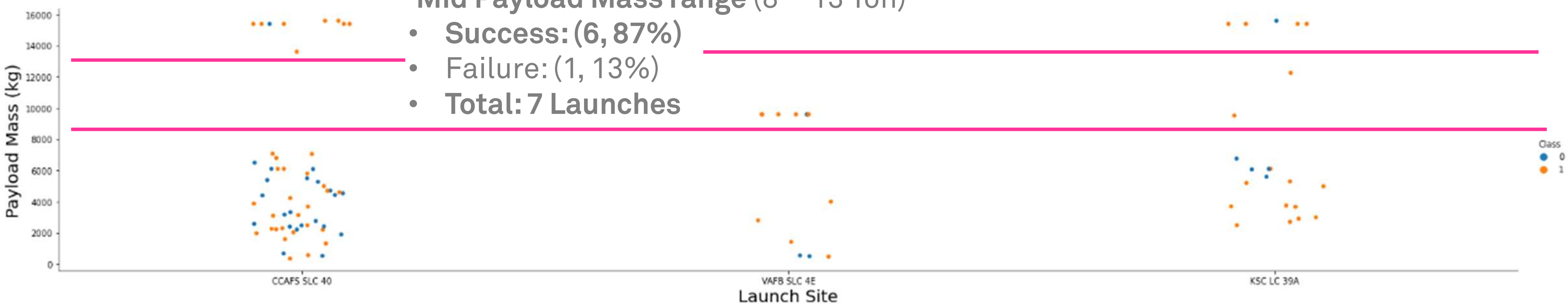
- **Success: (10, 77%)**
- Failure: (3, 23%)
- **Total: 13 Launches**

- **Success: (17, 77%)**
- Failure: (5, 23%)
- **Total: 22 Launches**

Payload Mass (kg) vs Launch Site

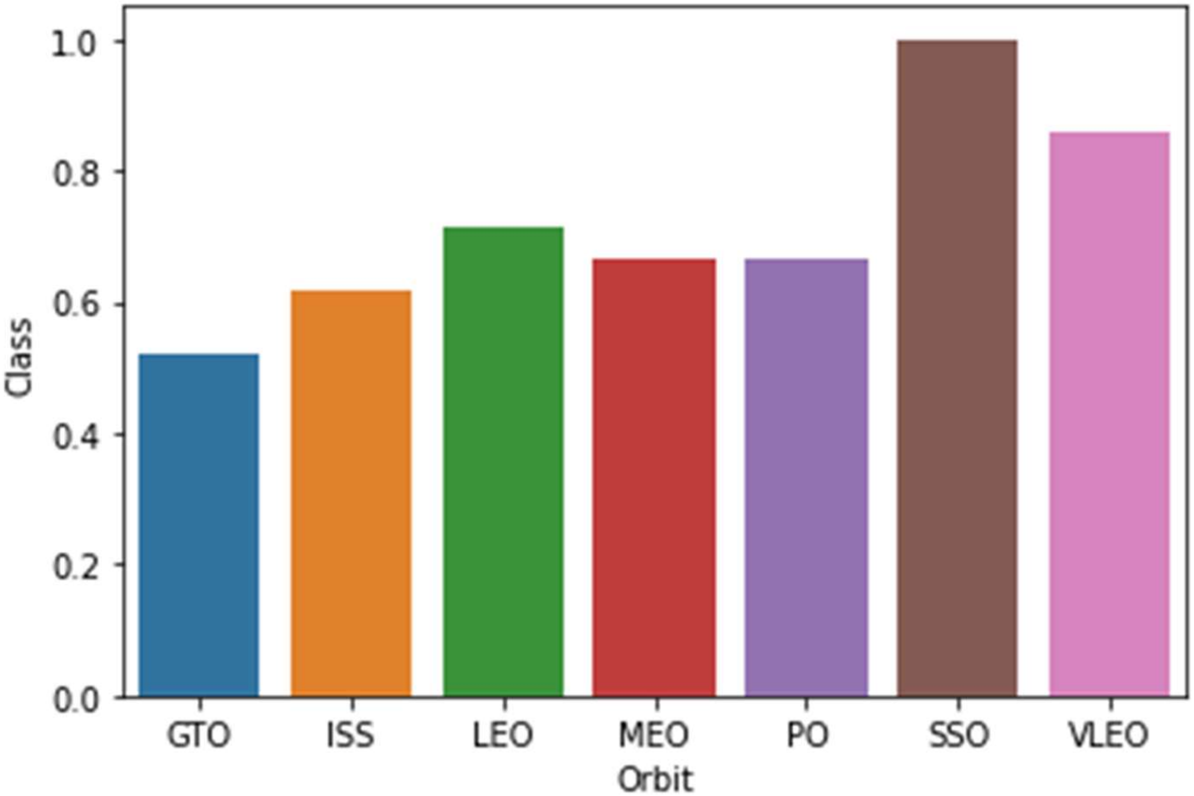
Heavy Payload Mass range (13 Ton and up)
- **Success: (11, 85%)**
- Failure: (2, 15%)
- **Total: 13 Launches**

Mid Payload Mass range (8 – 13 Ton)
- **Success: (6, 87%)**
- Failure: (1, 13%)
- **Total: 7 Launches**



Light Payload Mass range (0 – 8 Ton)
- **Success: (41, 60%)**
- Failure: (27, 40%)
- **Total: 68 Launches**

Success Rate vs Orbit



!

- **ES-L1, GEO, HEO, and SO orbits** had only **1 launch each** with a **success rate of 100%**
- SSO Orbit has a **100% success rate in 5 launches.**
- VLO Orbit has an **86% success rate in 14 launches.**

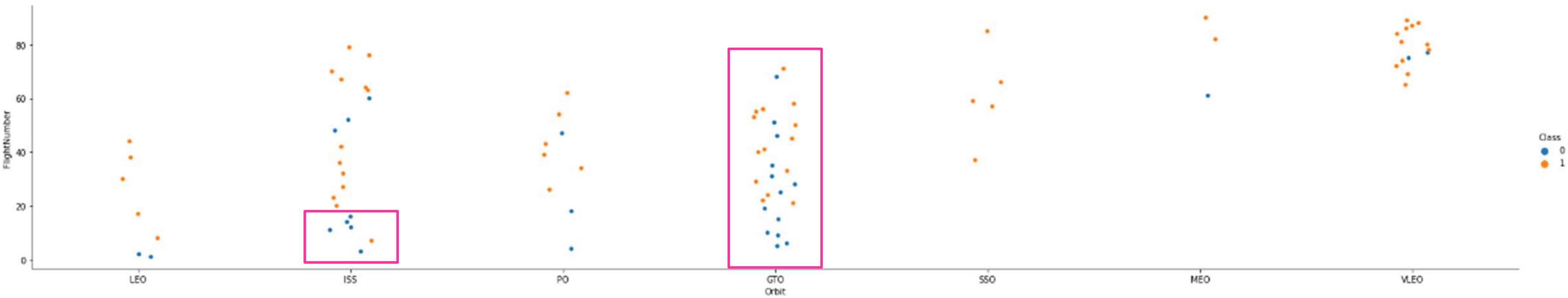
IBM Developer

SKILLS NETWORK

Flight Number vs Orbit

! **ES-L1, GEO, HEO, and SO orbits** had only **1 launch.**

!
- **Unpredictable behavior** indicates an out-of-control launching process



!
- **Initial failures** could indicate configuration issues **(Learning Curve)**

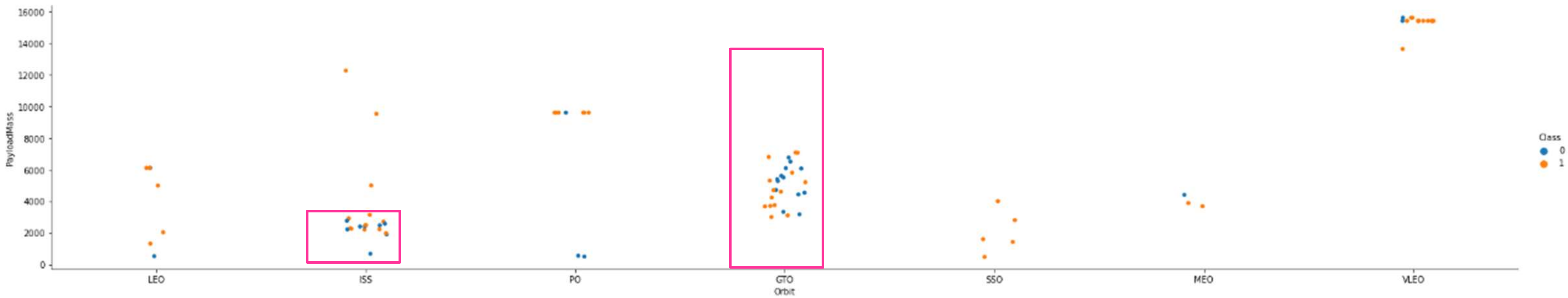
- **After the 20 continuous flight attempts mark,** Mission Outcome seems to stabilize towards success.

Payload Mass (kg) vs Orbit

! **ES-L1, GEO, HEO, and SO orbits** had only **1 launch.**

!

- **Unpredictable behavior** indicates an out-of-control launching process



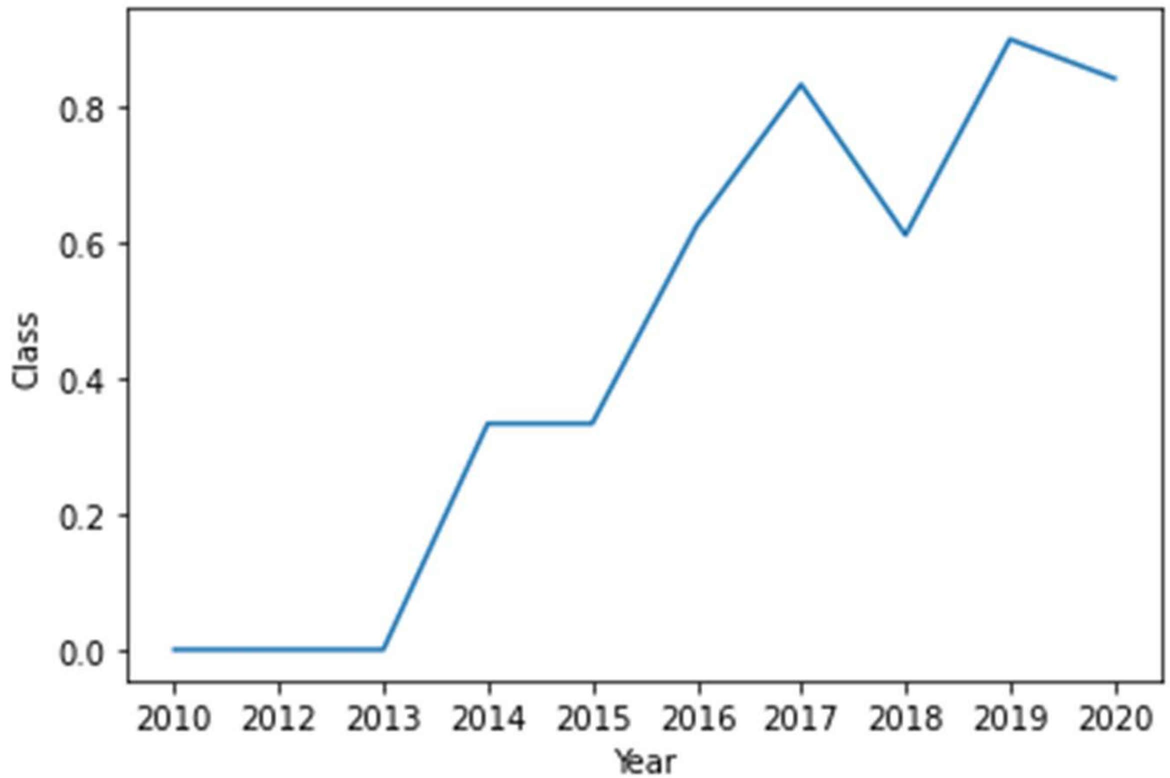
!

- **Unpredictable behavior** indicates external factors affecting Mission Outcome

- Same exact behavior as previous analysis. **There's no evidence of a clear impact due to the orbit type.**

Payload Mass (kg) vs Orbit

- As the number of launches has continuously and steadily been increasing since 2013, the success rate has increased as well.
- The 2018 dive is the result of data not being assigned to a landing path



EDA with SQL: Results

QUESTIONS TO ANSWER

- **Finding unique** launch sites

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

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

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

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

EDA with SQL: Results

QUESTIONS TO ANSWER

- Displaying **5 records (only)** where launch site begins with “CCA”

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
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

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- **Calculate the total payload mass** carried by boosters launched by NASA (CRS)

```
%%sql
SELECT Customer, SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
GROUP BY Customer
HAVING Customer = "NASA (CRS)"
```

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

Customer	SUM(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- **Calculate average payload mass** carried by booster version F9 v1.1

```
%%sql

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

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

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- Listing the boosters that **complied with specific mission parameters**

```
%%sql
SELECT DISTINCT "Landing _Outcome", Customer, Date FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (ground pad)"
ORDER BY Date ASC LIMIT 1

--SELECT DISTINCT MIN(Date) FROM SPACEXTBL
--WHERE "Landing _Outcome" = "Success (ground pad)"
```

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

Landing _Outcome	Customer	Date
Success (ground pad)	NRO	01-05-2017

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- Listing the boosters that **complied with specific mission parameters**

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
SELECT Booster_Version
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (drone ship)"
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

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

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

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- **Listing records by** Mission Outcome Status.

```
%%sql
SELECT TRIM(Mission_Outcome), COUNT(Mission_Outcome) AS "Total"
FROM SPACEXTBL
GROUP BY TRIM(Mission_Outcome)
```

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

TRIM(Mission_Outcome)	Total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

IBM Developer SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- Finding the boosters which **carried the maximum** payload mass amount

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

```
%%sql
SELECT DISTINCT Booster_Version
FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

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

EDA with SQL: Results

QUESTIONS TO ANSWER

- Finding **specific record details for a specific year**

```
%%sql
SELECT
(
    CASE substr(Date,4,2)
        WHEN "01" THEN "January"
        WHEN "02" THEN "February"
        WHEN "03" THEN "March"
        WHEN "04" THEN "April"
        WHEN "05" THEN "May"
        WHEN "06" THEN "June"
        WHEN "07" THEN "July"
        WHEN "08" THEN "August"
        WHEN "09" THEN "September"
        WHEN "10" THEN "October"
        WHEN "11" THEN "November"
        WHEN "12" THEN "December"
    END
) AS Month, Booster_Version, Launch_Site, "Landing _Outcome"
FROM SPACEXTBL
WHERE
"Landing _Outcome" = "Failure (drone ship)"
AND
substr(Date,7,4)='2015'
```

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

Month	Booster_Version	Launch_Site	Landing _Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

IBM Developer

SKILLS NETWORK

EDA with SQL: Results

QUESTIONS TO ANSWER

- **Rank mission outcomes** between 2 dates in descending order.

```
%%sql
SELECT Year, COUNT(*) AS Successes
FROM
(SELECT Date, substr(Date,7,4)*1 AS Year, substr(Date,7,4)*10000+substr(Date,4,2)*100+substr(Date,1,2) AS Datecode
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "%Succ%"
AND Datecode BETWEEN 20100604 AND 20170320)
GROUP BY Year
ORDER BY Successes DESC
```

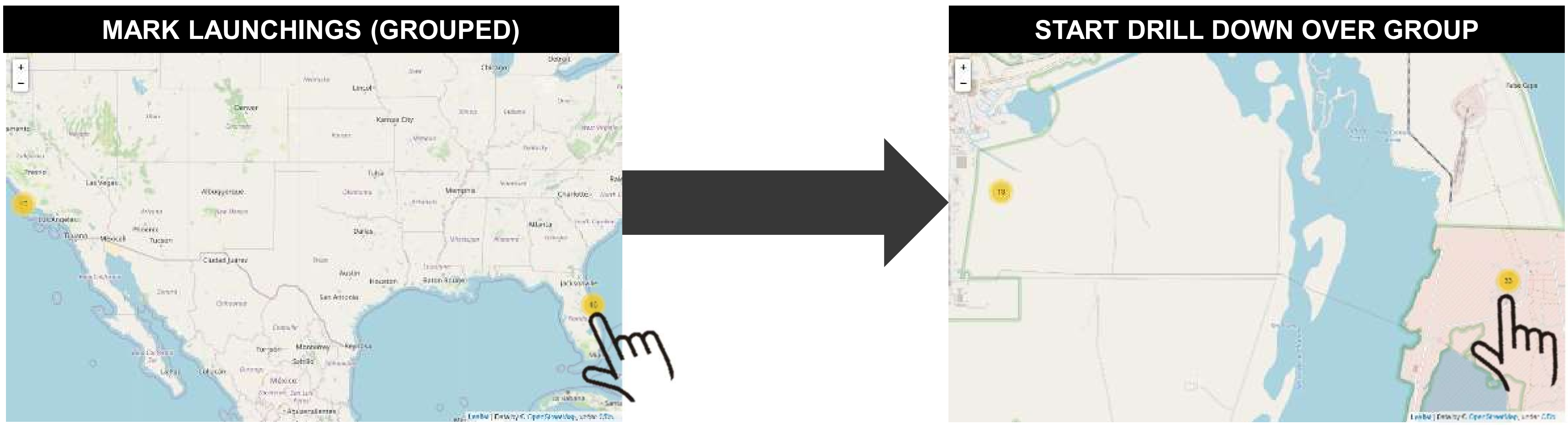
```
 * sqlite:///my_data1.db
Done.
```

Year	Successes
2016	5
2017	2
2015	1

IBM Developer SKILLS NETWORK

Interactive Map

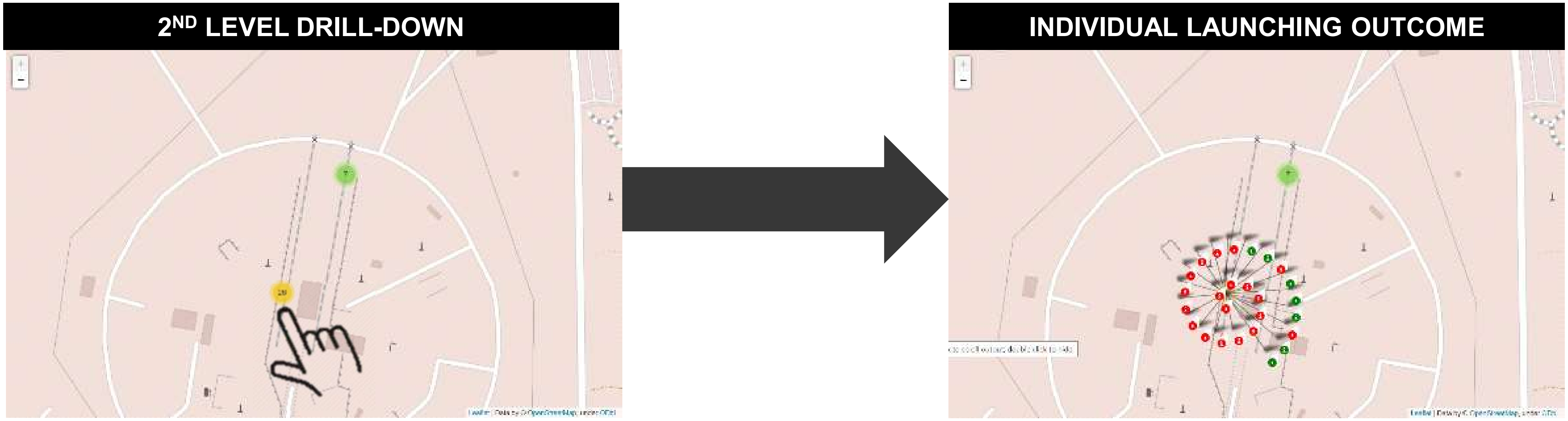
- **All records for all launch sites are presented on the map.** Records are grouped based on proximity so that the map looks neat as user zooms out. **Zooming in** (clicking on the circles) **will drill down and present the "un-grouped" data.**



Interactive Map

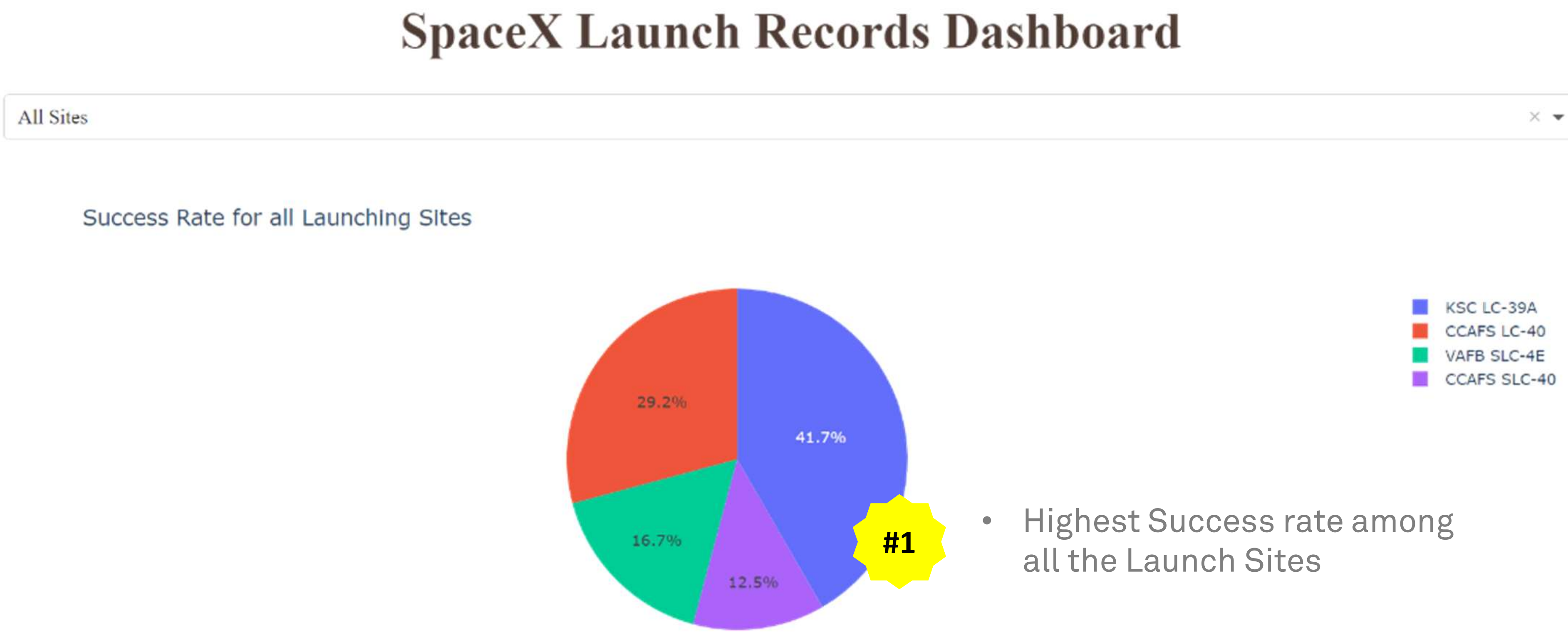
Success Failure

- At the maximum zoom level **data is broken down to the individual records.** Before reaching the deepest level, “Yellow” circles just indicate that additional zoom is available. At the final level they indicate mixed results predominantly failures. **“Green”** circles indicate that **around 50% of the individual records are favorable.**



IBM Developer SKILLS NETWORK

Success Rate for all Launch Sites



- Highest Success rate among all the Launch Sites

IBM Developer

SKILLS NETWORK

Success Rate Individual Site: KSC LC-39A

SpaceX Launch Records Dashboard

KSC LC-39A

Success Rate for KSC LC-39A

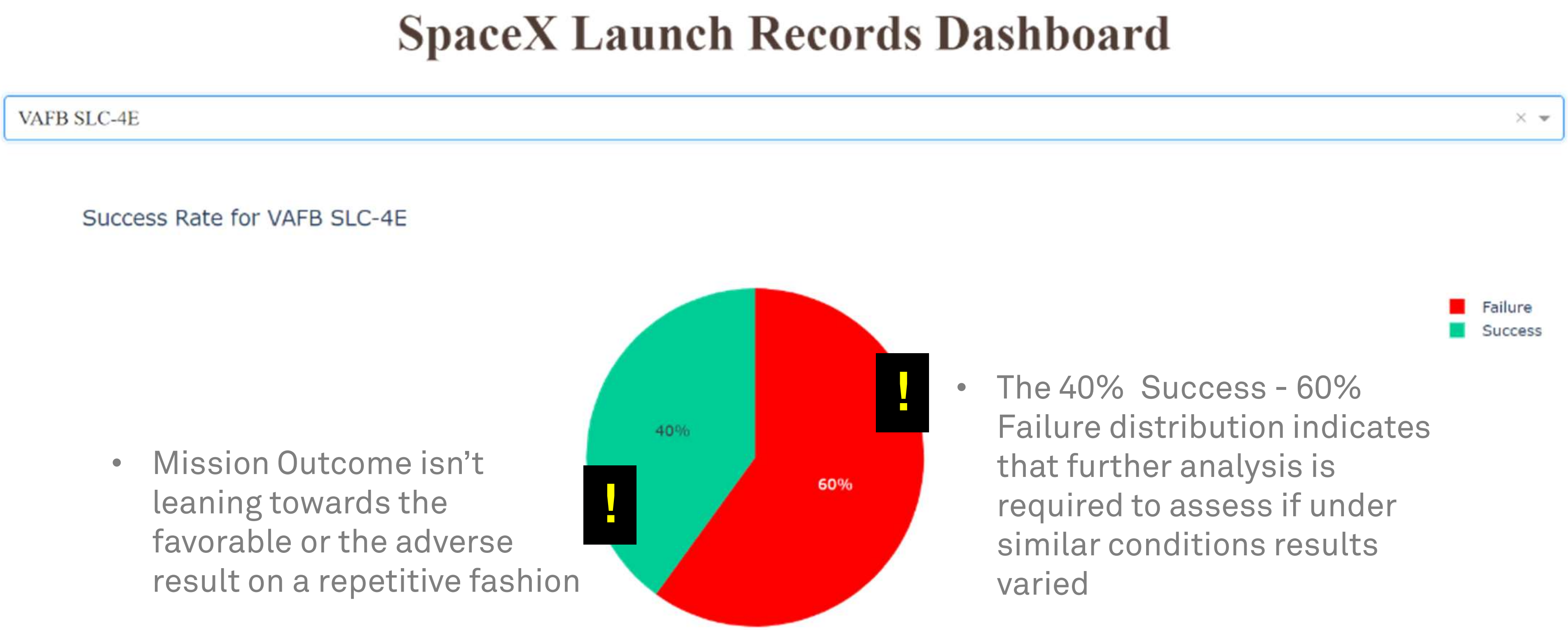
#1

- Highest Individual Success rate **(77 out of every 100 attempts)**

IBM Developer

SKILLS NETWORK

Success Rate Individual Site: VAFB SLC-4E



- Mission Outcome isn't leaning towards the favorable or the adverse result on a repetitive fashion

- The 40% Success - 60% Failure distribution indicates that further analysis is required to assess if under similar conditions results varied

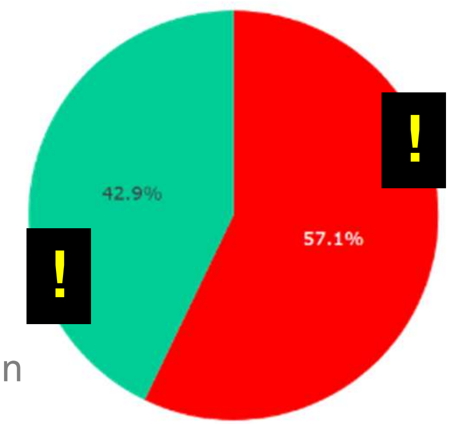
Success Rate Individual Site: CCAFS SLC-40

SpaceX Launch Records Dashboard

CCAFS SLC-40

Success Rate for CCAFS SLC-40

- Same behavior as VAFB SLC-4E



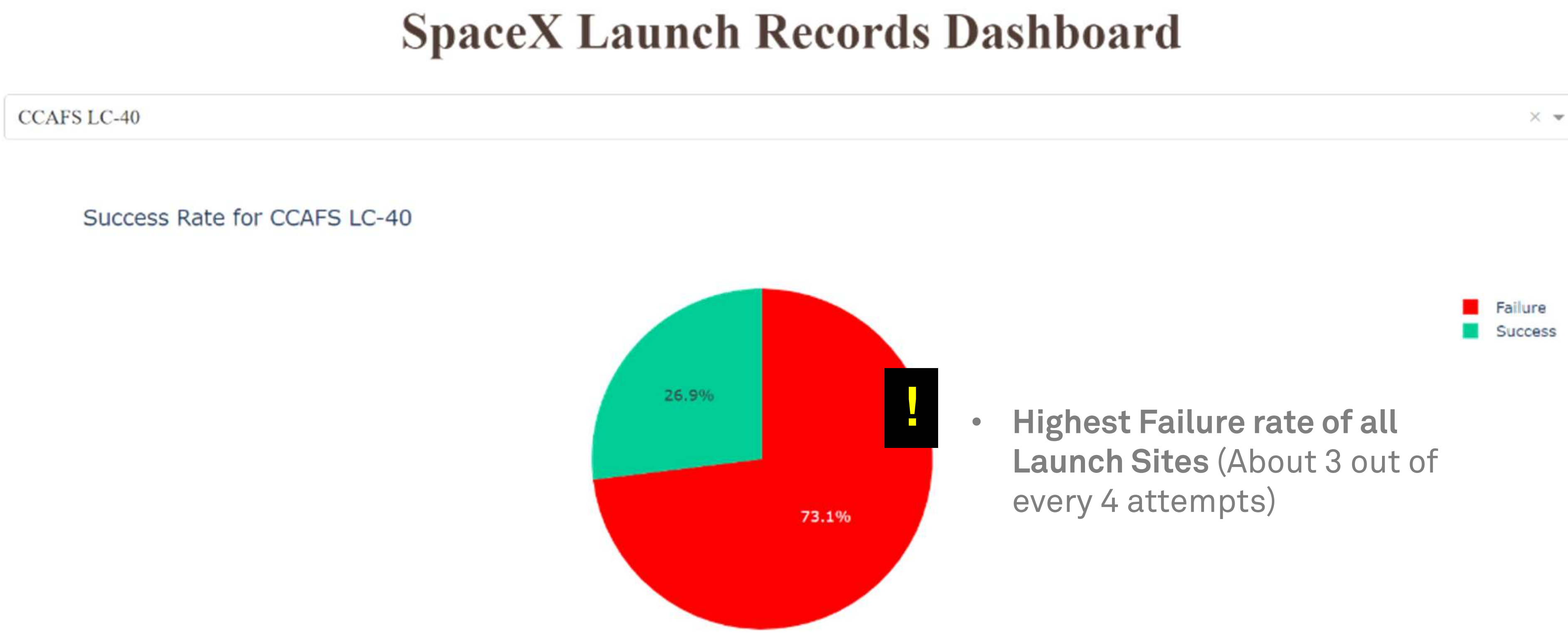
- Mission Outcome isn’t leaning towards the favorable or the adverse result on a repetitive fashion

- The 40% Success - 60% Failure distribution indicates that further analysis is required to assess if under similar conditions results varied

IBM Developer

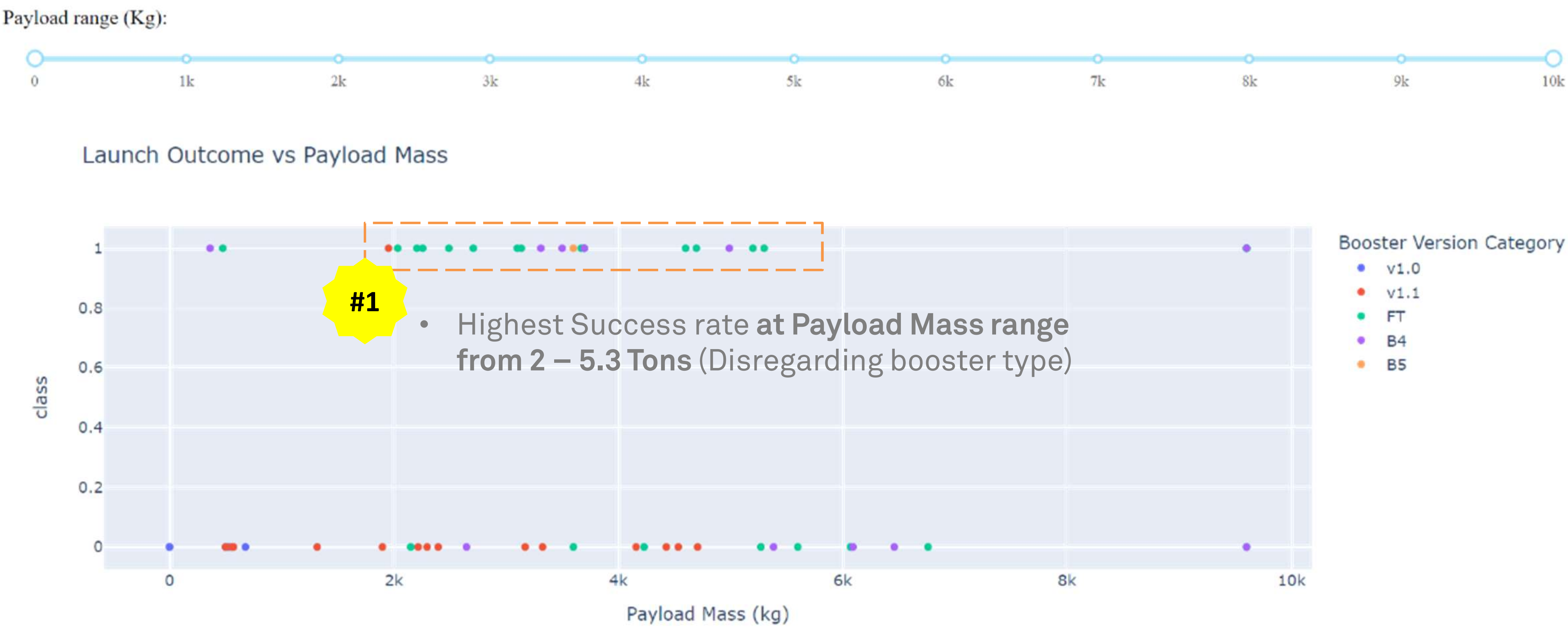
SKILLS NETWORK

Success Rate Individual Site: CCAFS LC-40



- **Highest Failure rate of all Launch Sites** (About 3 out of every 4 attempts)

Mission Outcome vs Payload Mass (kg)



IBM Developer

SKILLS NETWORK

Predictive Analytics: Summary

QUESTIONS TO ANSWER

- What is the model that performs the best?

TASK 12

Find the method performs best:

```
models = [logreg_cv, svm_cv, tree_cv, knn_cv]
results = pd.DataFrame(index=range(len(models)), columns=["Model","Score"])

for rix, model in enumerate(models):
    results["Model"].at[rix] = model.best_estimator_
    results["Score"].at[rix] = model.best_score_

results
```

	Model	Score
0	LogisticRegression(C=1)	0.85
1	SVC(gamma=0.03162277660168379, kernel='sigmoid')	0.95
2	DecisionTreeClassifier(max_depth=2, max_featur...	0.95
3	KNeighborsClassifier(n_neighbors=1, p=1)	0.95

SOLUTION

- From the modelling, **3 algorithms perform up to the same level of accuracy:** Decision Tree, KNN, and Support Vector Machine **with 95%.**

- The 3 algorithms serve well the classification purpose of this analysis. However, it is considered that the **Decision Tree** algorithm offers advantages over both KNN and Support Vector Machine algorithms.

Predictive Analytics: Decision Tree

MODEL FITTING

TASK 8

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(tree, parameters)
tree_cv.fit(X_test, Y_test)
```

```
GridSearchCV(estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

TUNED HYPERPARAMETERS

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 2, 'max_features': 'auto', 'min_samples_leaf': 2, '
min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.95
```

TASK 9

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

```
print("accuracy :",tree_cv.best_score_)
```

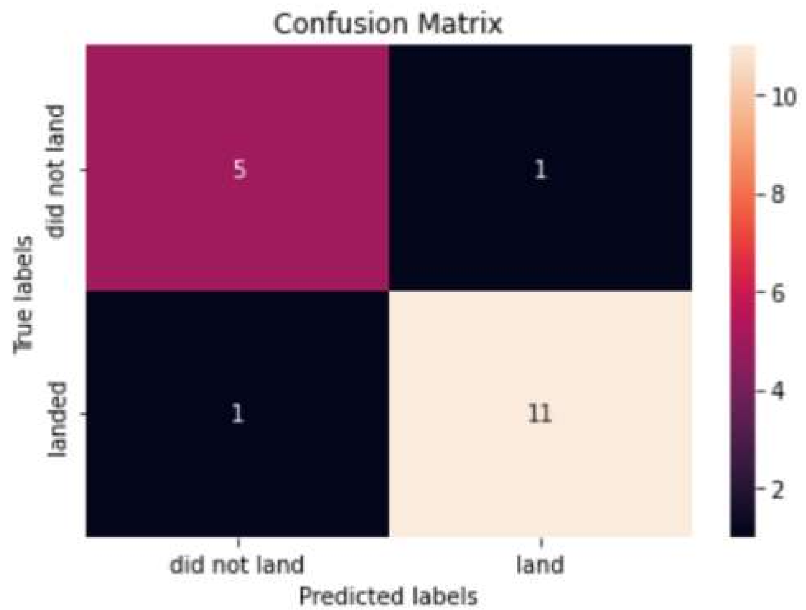
```
accuracy : 0.95
```

IBM Developer

SKILLS NETWORK

Predictive Analytics: Decision Tree

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



5. Conclusion

After reviewing the data available, **the Space Race seems to be dominated by those who attempt the most to keep flying.** Data indicates that **there is a learning curve impacting different metrics** and that after a certain threshold is surpassed performance improves. Additionally, Success Rate has been increasing since 2013 just like efforts are.

Regarding the analysis, for the classification purpose **there are at least 3 methods that are fit for the task.** Each of them has its Pros and Cons, but it is fair to say that at the current complexity required**, the Decision Tree algorithm beats both Support Vector Machine and KNN algorithms.** Deeper analysis is required to analyze some of the hypothesis that arose during the study.

6. Appendix

Access to the Datasets, Tools, Jupyter Notebooks, and other files are available upon request.

https://github.com/hmartinez89/IBB_DataScience_Certification/tree/master