



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Chukwuemeka Okoli  
2<sup>nd</sup> December 2021



# Outline

---

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

# Executive Summary

---

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

# Introduction

---

- Project background and context

Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- 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.
- What operating conditions needs to be in place to ensure a successful landing program.



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection using SpaceX API.
- Perform data wrangling: Applied one-hot encoding to categorical variables
- Perform EDA using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models

# Data Collection

---

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using `.json()` function call and turn it into a pandas dataframe using `.json_normalize()`.
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.

# Data Collection – SpaceX API

---

```
In [10]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_

We should see that the request was successfull with the 200 status response code

In [11]: response.status_code

Out[11]: 200

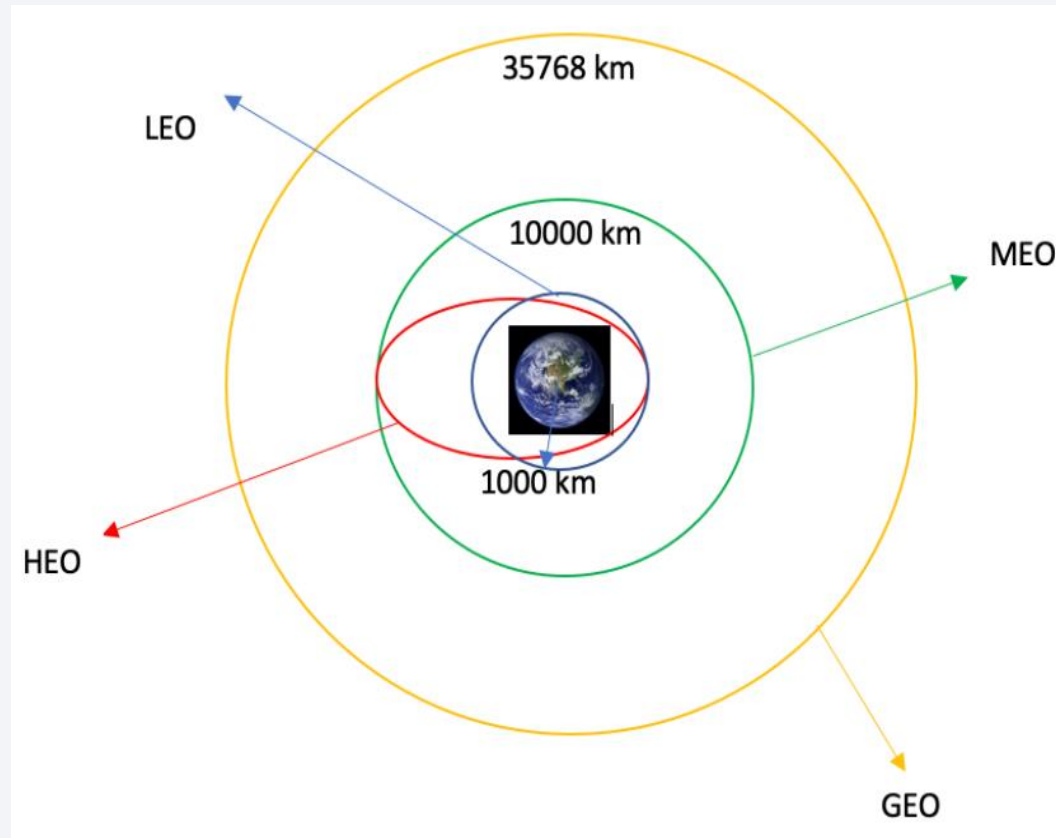
Now we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

In [12]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is <https://github.com/minhvu01122001/IBM-Data-Science-Project/blob/main/1.%20Data%20Collection.ipynb>



# Data Wrangling



- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is <https://github.com/minhvu01122001/IBM-Data-Science-Project/blob/main/2.%20Data%20Wrangling.ipynb>

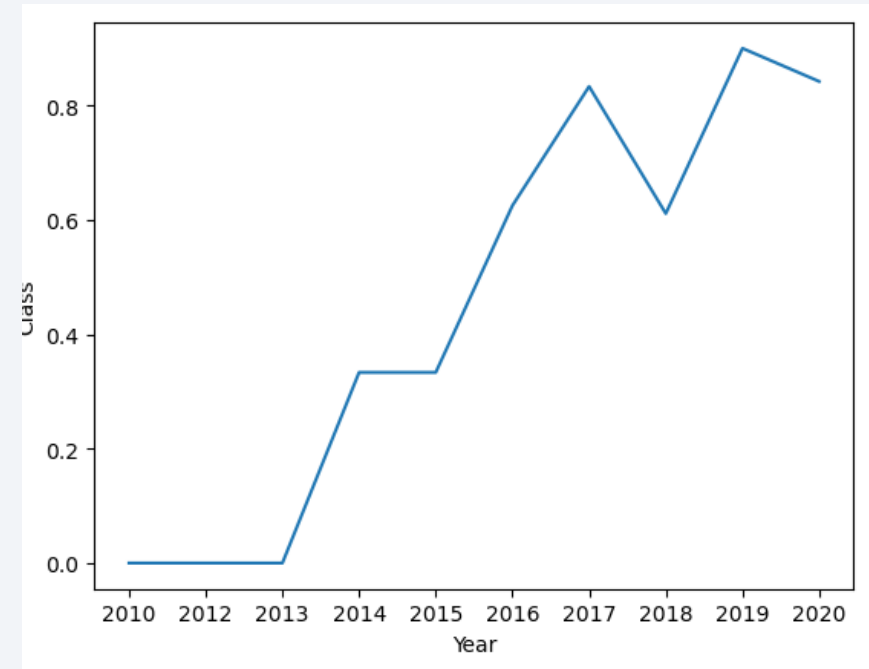
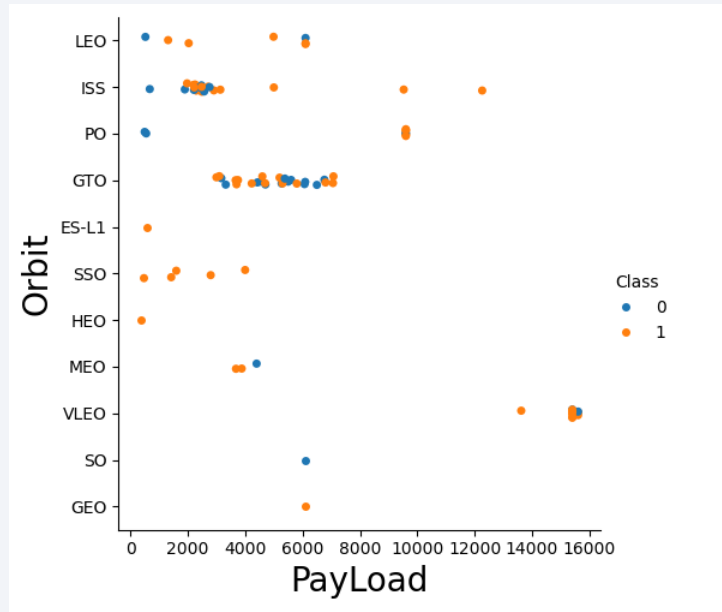
# EDA with SQL

---

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is

# EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.



- The link to the notebook is <https://github.com/minhvu01122001/1BM-Data-Science-Project/blob/main/4.%20EDA%20Data%20Viz.ipynb>

# Build an Interactive Map with Folium

---

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

# Predictive Analysis (Classification)

---

- 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.
- The link to the notebook is <https://github.com/chuksoo/IBM-Data-Science-Capstone-SpaceX/blob/main/Machine%20Learning%20Prediction.ipynb>



# Results

---

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



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks are layered and have a textured, almost woven appearance. A faint, light blue grid pattern is visible across the entire background, particularly prominent in the blue section on the left.

Section 2

# Insights drawn from EDA



# All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

In [9]:

```
%%sql  
SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
```

\* sqlite:///my\_data1.db

Done.

Out[9]:

**Launch\_Site**

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

# Launch Site Names Begin with 'CCA'

---

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

```
%%sql
SELECT * FROM SPACEXTABLE
where "Launch_Site" LIKE 'CCA%'
LIMIT 5
```

- We used the query above to display 5 records where launch sites begin with 'CCA'

# Total Payload Mass

---

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

In [12]: task_3 = '''
          SELECT SUM(PayloadMassKG) AS Total_PayloadMass
          FROM SpaceX
          WHERE Customer LIKE 'NASA (CRS)'
          '''
          create_pandas_df(task_3, database=conn)

Out[12]:
```

	total_payloadmass
0	45596



# Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

```
In [13]: task_4 = '''
          SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
          FROM SpaceX
          WHERE BoosterVersion = 'F9 v1.1'
          '''
          create_pandas_df(task_4, database=conn)
```

```
Out[13]:
```

	avg_payloadmass
0	2928.4

# First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad

In [32]:

```
%%sql  
SELECT Min("Date")  
FROM SPACEXTABLE  
WHERE "Landing_Outcome" = 'Success (drone ship)'
```

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

Out[32]: **Min("Date")**

2016-05-27

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [15]: task_6 = '''
          SELECT BoosterVersion
          FROM SpaceX
          WHERE LandingOutcome = 'Success (drone ship)'
             AND PayloadMassKG > 4000
             AND PayloadMassKG < 6000
          ...
          create_pandas_df(task_6, database=conn)
```

```
Out[15]:
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

# Total Number of Successful and Failure Mission Outcomes

In [37]:

```
%%sql
SELECT COUNT("Mission_Outcome") AS COUNT_VALUE, "Mission_Outcome"
FROM SPACEXTABLE
GROUP BY "Mission_Outcome"
```

\* sqlite:///my\_data1.db

Done.

Out[37]:

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

# Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
In [39]: %%sql
SELECT "Booster_Version", "PAYLOAD_MASS_KG_"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS_KG_" = (SELECT MAX("PAYLOAD_MASS_KG_") FROM SPACEXTABLE)
```

\* sqlite:///my\_data1.db  
Done.

```
Out[39]:
```

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



# 2015 Launch Records

---

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
In [42]: %%sql
SELECT
SUBSTR(Date, 6, 2) AS Month,
SUBSTR(Date, 0, 5) AS Year,
"Booster_Version",
"Launch_Site",
"Landing_Outcome"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)' AND SUBSTR(Date, 0, 5) = '2015'
```

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

```
Out[42]:
```

Month	Year	Booster_Version	Launch_Site	Landing_Outcome
10	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

# Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

In [60]:

```
%%sql
SELECT COUNT("Landing_Outcome") AS COUNT_VALUE, "Landing_Outcome"
FROM
(SELECT * FROM SPACEXTABLE
WHERE Date>'2010-06-04' AND Date<'2017-03-20')
GROUP BY "Landing_Outcome"
ORDER BY COUNT_VALUE DESC
```

\* sqlite:///my\_data1.db

Done.

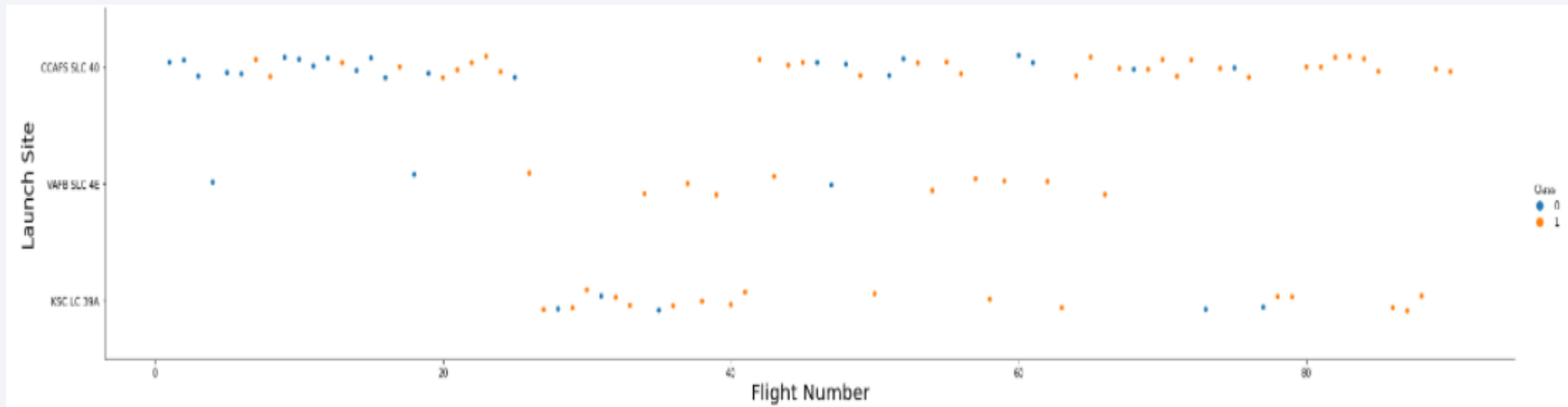
Out[60]:

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

# Flight Number vs. Launch Site

---

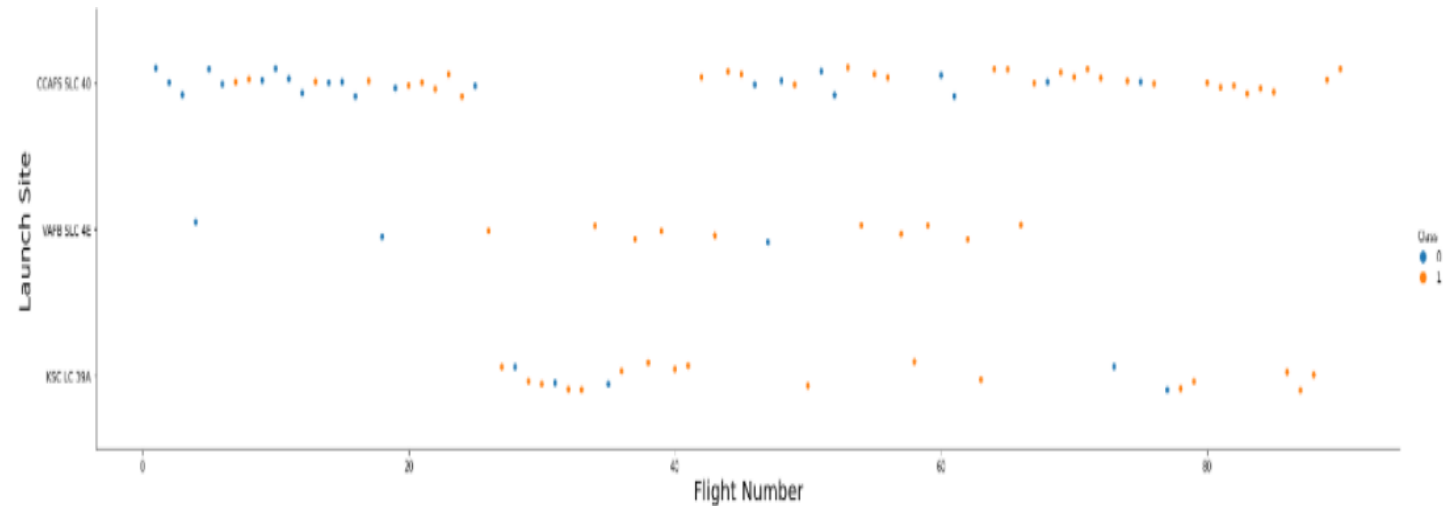
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



## Payload vs. Launch Site

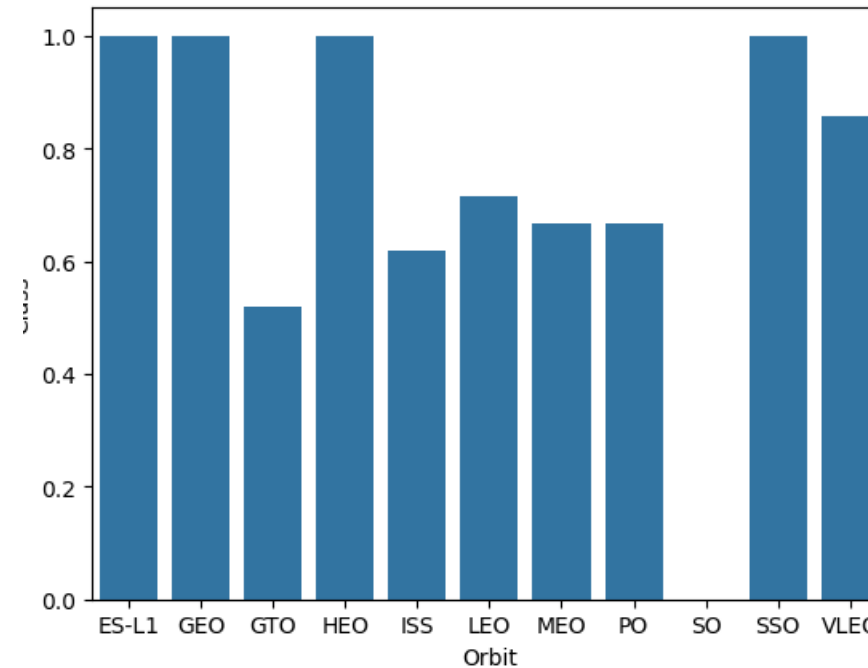


The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



# Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

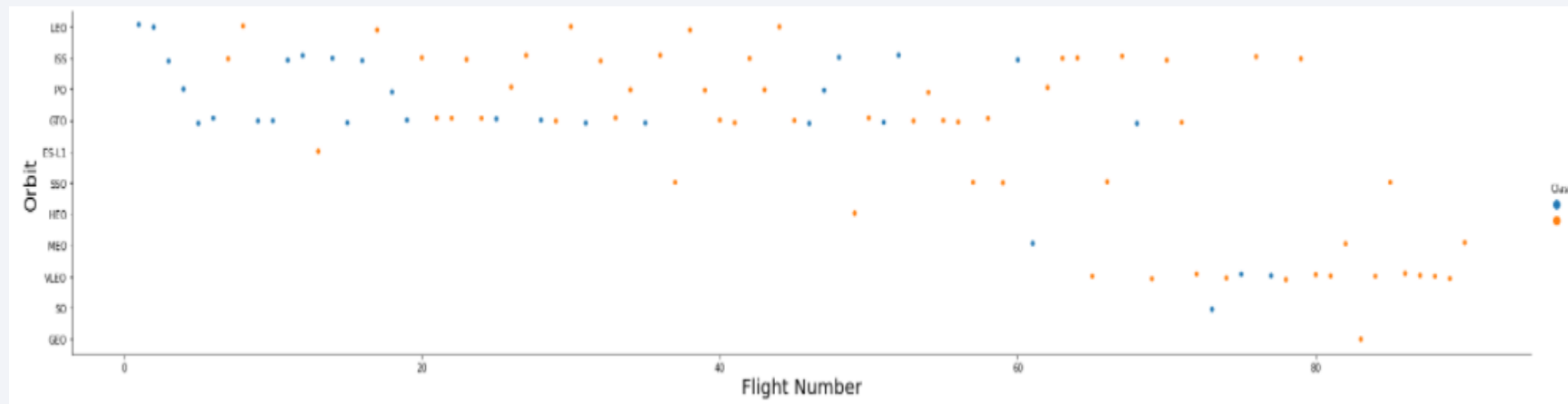




# Flight Number vs. Orbit Type

---

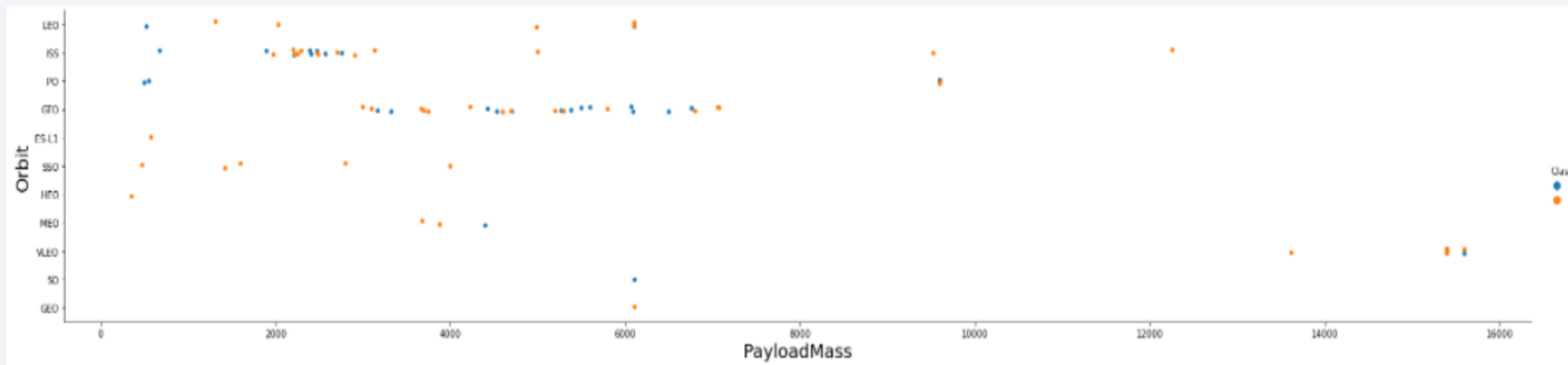
- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



# Payload vs. Orbit Type

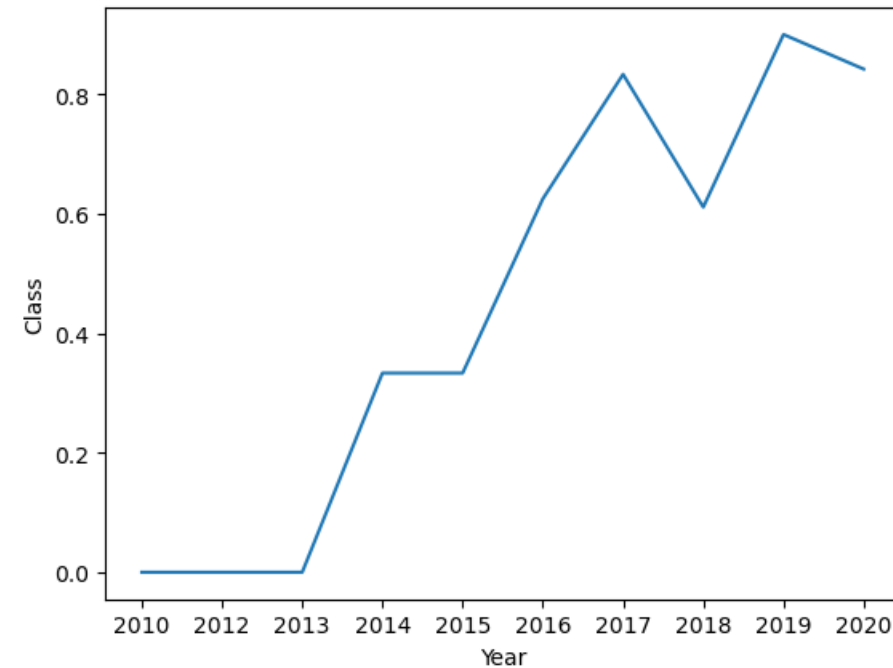
---

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



# Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.

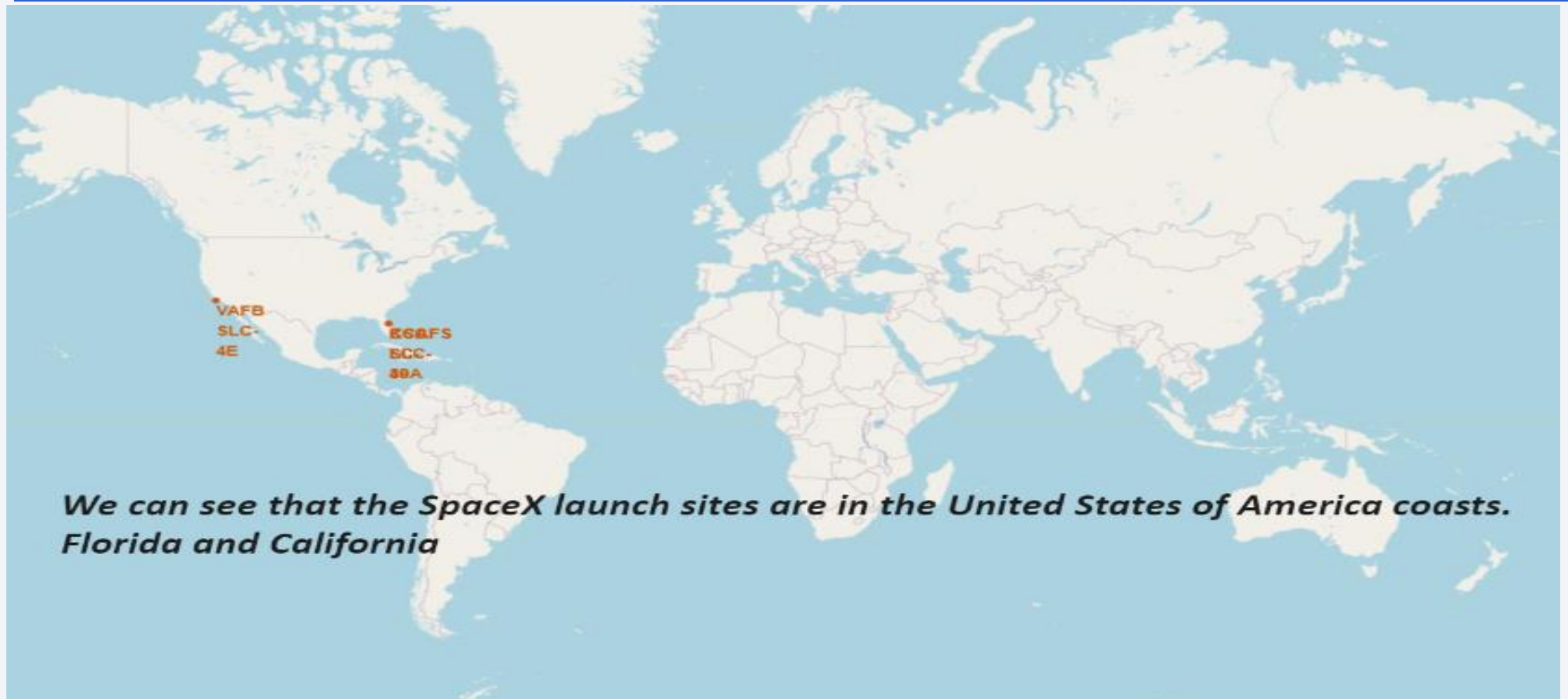


Section 4

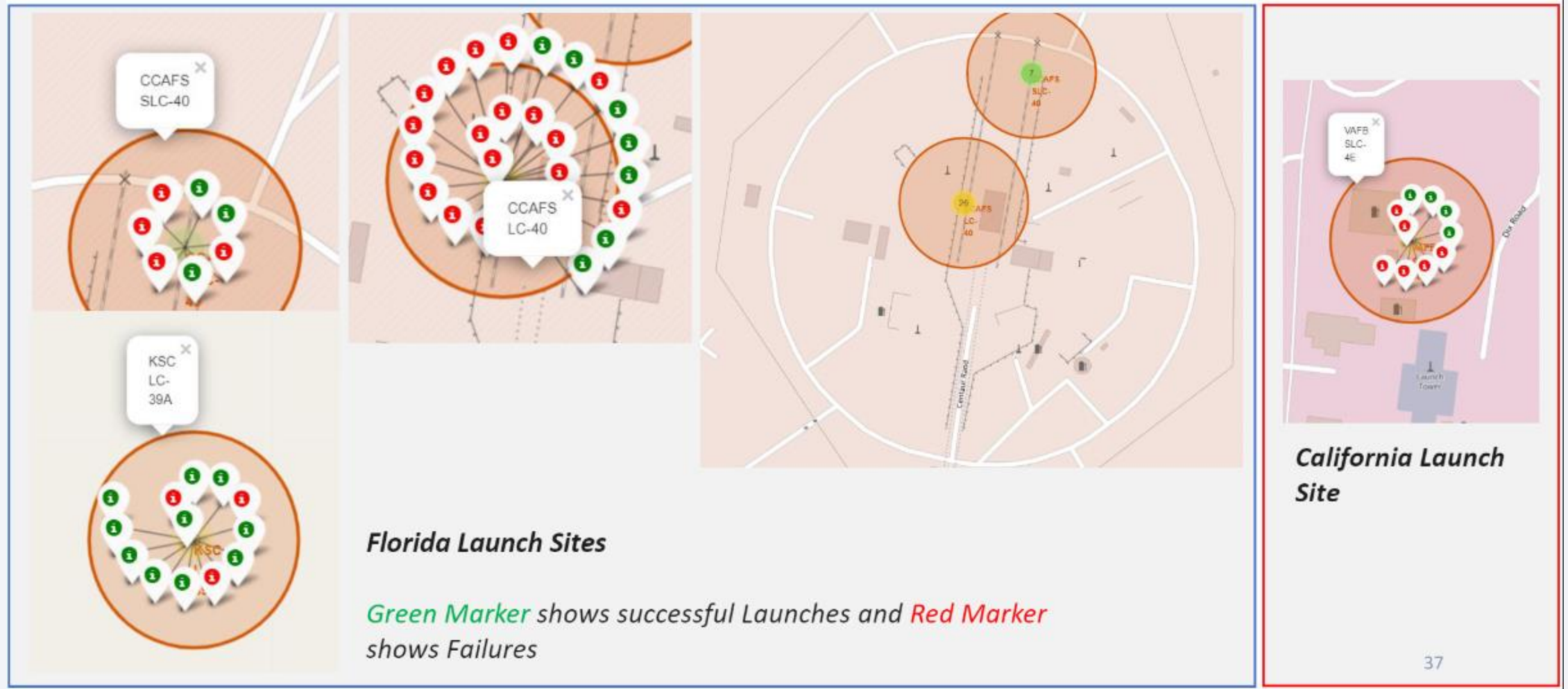
# Launch Sites Proximities Analysis



# All launch sites global map markers



# Markers showing launch sites with color labels





# Launch Site distance to landmarks



- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes





Section 6

# Predictive Analysis (Classification)

# Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

```
models = {'KNeighbors': knn_cv.best_score_,
          'DecisionTree': tree_cv.best_score_,
          'LogisticRegression': logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm, 'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

Best model is DecisionTree with a score of 0.8732142857142856

Best params is : {'criterion': 'gini', 'max\_depth': 6, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'splitter': 'random'}

# Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



# Conclusions

---

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.



Thank you!

