



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Harshini Saraswathi  
08/02/2023



# Outline

---

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

# Executive Summary

---

- **Summary of methodologies**

- Data Collection via API, Web Scraping
- Exploratory Data Analysis (EDA) with Data Visualization
- EDA with SQL
- Interactive Map with Folium
- Dashboards with Plotly Dash
- Predictive Analysis

- **Summary of all results**

- Exploratory Data Analysis Results
- Interactive maps and Dashboards
- Predictive 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.
- Problems you want to find answers
  - 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 ?



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

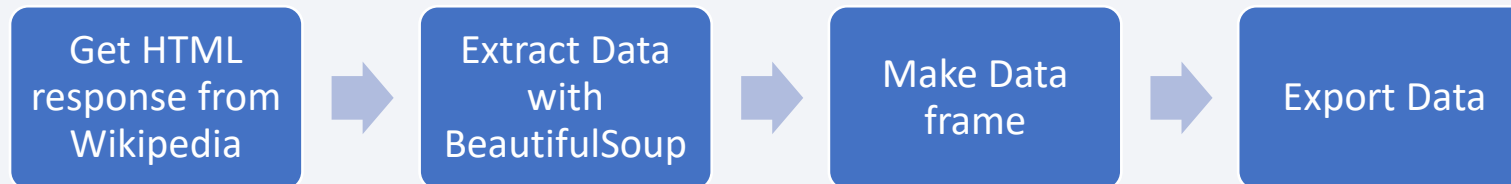
---

- **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/](https://api.spacexdata.com/v4/)



- **The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.**



# 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 to JSON file

```
data = response.json()
data = pd.json_normalize(data)
```

## 3. Transform Data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

## 4. Create dictionary with data

```
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

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

## 6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

## 7. Export to file

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

[Link to code](#)



# Data Collection - Scraping

## 1. Getting response from HTML

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

## 2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

## 3. Find all tables

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

## 4. Get column names

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

## 5. Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelevant 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']=[]
```

## 7. Create dataframe from dictionary

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

## 8. Export to file

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

## 6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

# Data Wrangling

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

## 1. Calculates launches for each site

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

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

## 2. Calculate 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
SO        1
ES-L1    1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

## 3. Calculate the number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

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

## 4. Create landing outcome label from Outcome column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

## 5. Export to file

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

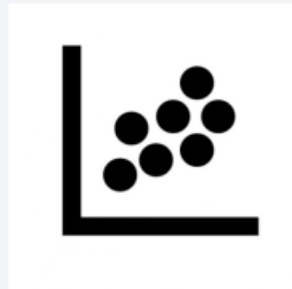
[Link to code](#)

# EDA with Data Visualization

---

- **Scatter Graphs**

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass



*Scatter plots show relationship between variables.  
This relationship is called the correlation.*

- **Bar Graph**

- Success rate vs. Orbit

*Bar graphs show the relationship between numeric and categoric variables.*



- **Line Graph**

- Success rate vs. Year

*Line graphs show data variables and their trends.  
Line graphs can help to show global behavior and  
make prediction for unseen data*



[Link to code](#)

# 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
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
    - List the total number of successful and failure mission outcomes.
  - List the names of the booster\_versions which have carried the maximum payload mass.
  - List the records which will display the month names, failure landing\_outcomes in drone ship, booster versions, launch\_site for the months in year 2015.
  - Rank the count of successful landing\_outcomes 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's coordinate with label showing its name (folium.Circle, folium.map.Marker).
  - Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (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 payload mass in a fixed range (*dash\_core\_components.RangeSlider*).
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (*plotly.express.scatter*).

[Link to code](#)

# Predictive Analysis (Classification)

---

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models 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



The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is dynamic and technological.

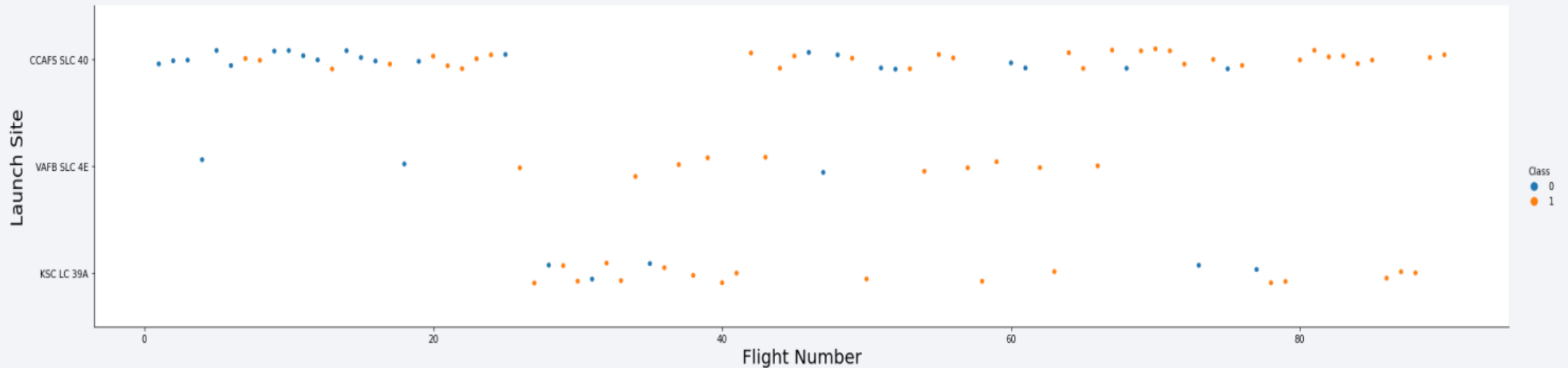
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

---

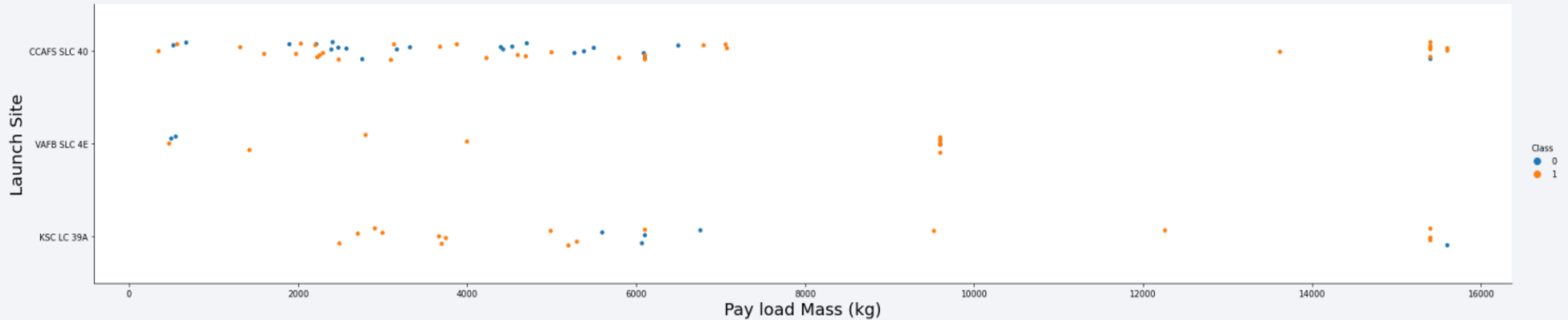


**We observe that, for each site, the success rate is increasing.**



# Payload vs. Launch Site

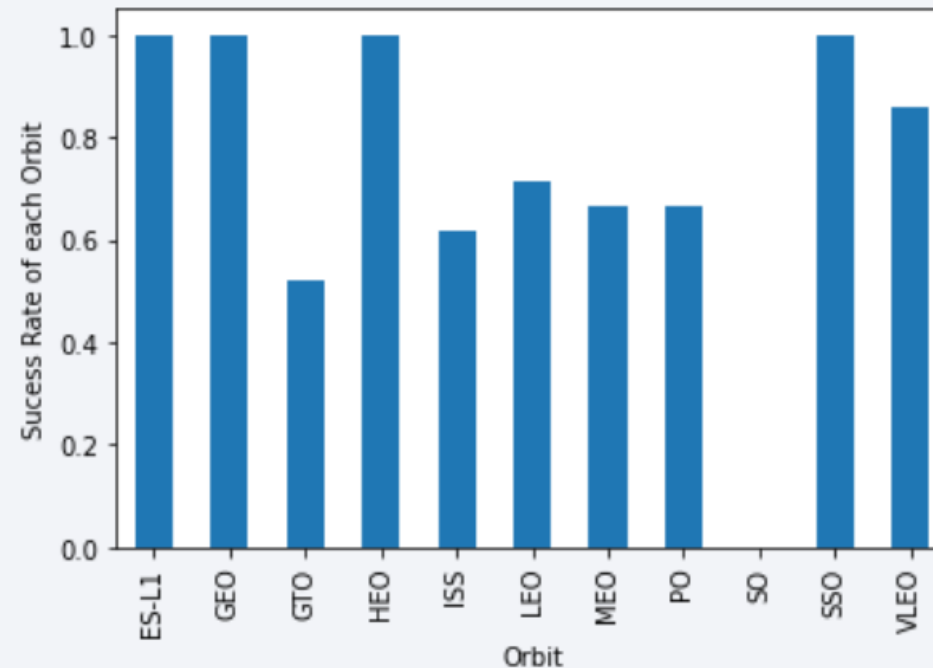
---



**Depending on the launch site, a heavier payload may be a consideration for a successful landing. On the other hand, a too heavy payload can make a landing fail.**

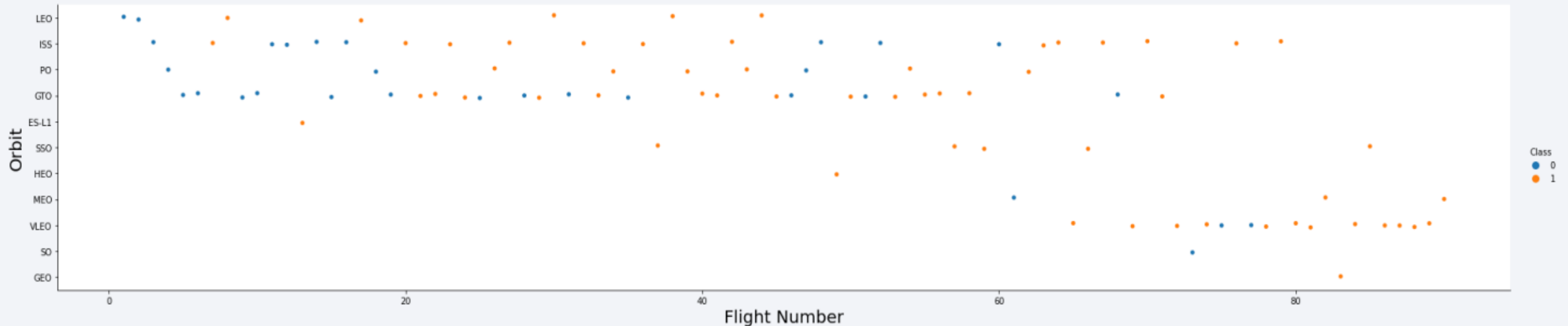
# Success Rate vs. Orbit Type

---



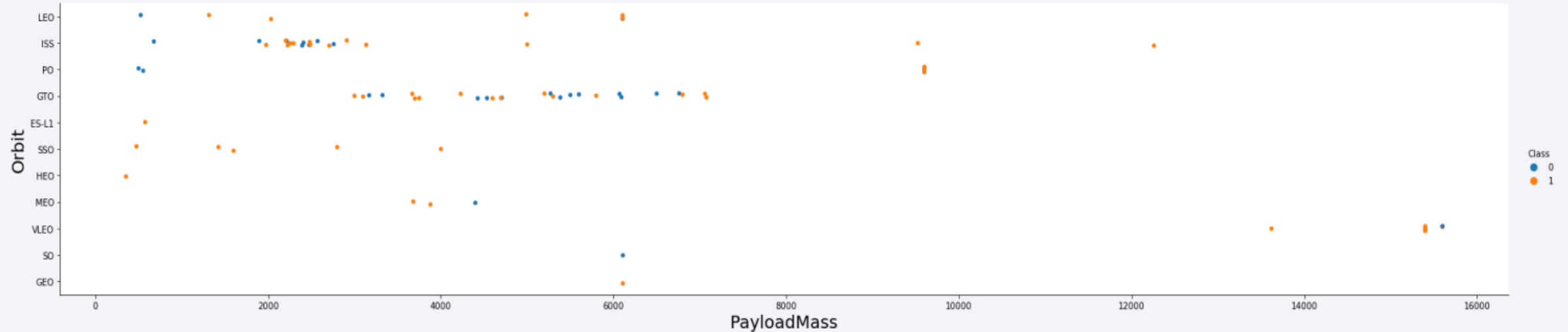
**With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.**

# 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 success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.

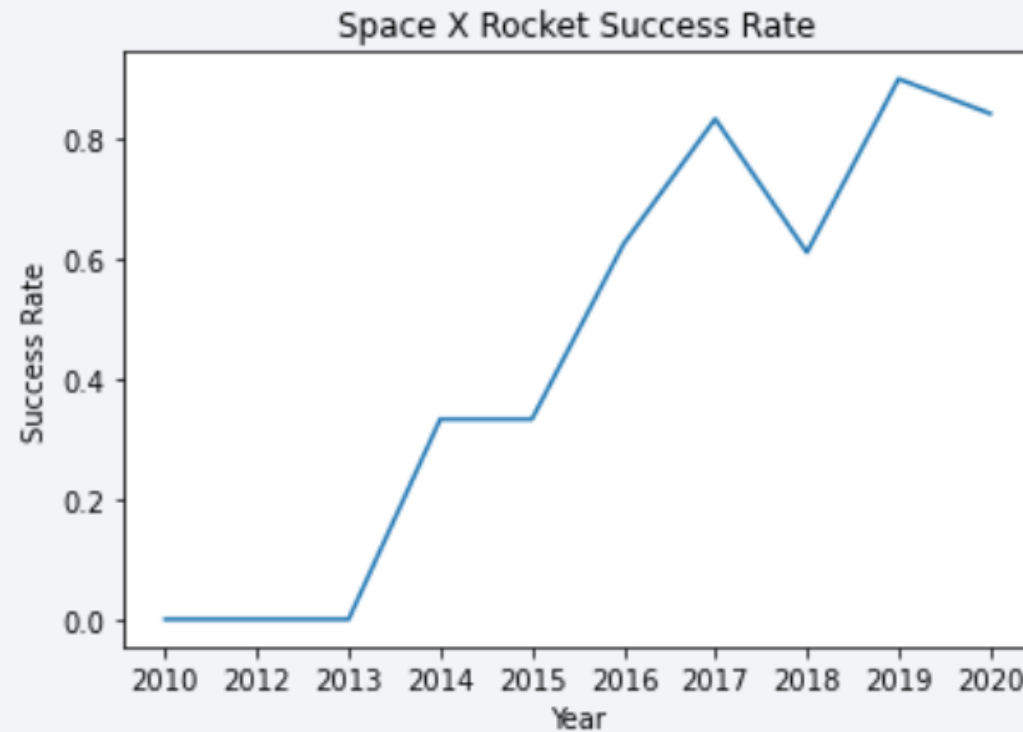
# Payload vs. Orbit Type



**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.



# All Launch Site Names

---

## SQL QUERY

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

## Explanation

The use of DISTINCT in the query allows to remove duplicate LAUNCH\_SITE.

## Results

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

# Launch Site Names Begin with 'CCA'

---

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

# Total Payload Mass

---

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

# Average Payload Mass by F9 v1.1

---

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

# First Successful Ground Landing Date

---

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here



## Successful Drone Ship Landing with Payload between 4000 and 6000

---

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

# Total Number of Successful and Failure Mission Outcomes

---

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

# Boosters Carried Maximum Payload

---

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

# 2015 Launch Records

---

- List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

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

---

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Present your query result with a short explanation here

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

# <Folium Map Screenshot 1>

---

- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

## <Folium Map Screenshot 2>

---

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot



# <Folium Map Screenshot 3>

---

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot



Section 4

# Build a Dashboard with Plotly Dash

# <Dashboard Screenshot 1>

---

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

## <Dashboard Screenshot 2>

---

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

## <Dashboard Screenshot 3>

---

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

# Predictive Analysis (Classification)



# Classification Accuracy

---

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy

# Confusion Matrix

---

# Conclusions

---

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure 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 best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as 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.

# Appendix

---

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!

