IBM Developer
SKILLS NETWORK

by KERKAD Amira
Jully 2022

SpaceX Falcon 9 Landing Prediction

Agenda

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

Agenda

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

Executive Summary

- Summary of methodologies
  1. Data Collection through API
  2. Data Collection with Web Scraping
  3. Data Wrangling
  4. Exploratory Data Analysis with SQL
  5. Exploratory Data Analysis with Data Visualization
  6. Interactive Visual Analytics with Folium
  7. Machine Learning Prediction

- Summary of all results
  1. Exploratory Data Analysis result
  2. Interactive analytics in screenshots
  3. Predictive Analytics result

Agenda

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

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 needed solving ?

  - What factors determine a successful landing for a rocket?
  - What determines the success rate of a successful landing.
  - What conditions need to be in place to maximize successful landing rate.

Agenda

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

Methodology

- Step 1: Data collection
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Step 2: Data wrangling (Transforming data for ML)
  - Encoding and Pruning data
  - Exploratory Data Analysis (EDA) using visualization and SQL
- Step 3: Visual Analytics using Folium and Plotly Dash
- Step 4: Machine Learning Prediction

Methodology

Step 1 : Data Collection

SpaceX API :

- Request to the SpaceX API
- Clean the requested data

WebScrapping :

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/jupyter-labs-webscraping.ipynb

Methodology
Data Collection with SPACEX API

Data Collection performed following these steps :

1. **Retrieve data with "get request" to the SpaceX API.**
2. **Decode the response with "Json"**
3. **Normalize the pandas dataframe using .json_normalize().**
4. **Clean data types, check and fill missing values.**
5. Webscraping from Wikipedia for Falcon 9 launch records with BeautifulSoup into HTML table
6. Parse and convert the table into a pandas dataframe.

```
Entrée [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

Entrée [7]:  response = requests.get(spacex_url)
```

```
Entrée [11]:  # Use json_normalize meethod to convert the json result into a dataframe
              data= pd.json_normalize(response.json())
```

```
Entrée [22]:  # Create a data from launch_dict
              df = pd.DataFrame(launch_dict)
```

```
Entrée [42]:  # Calculate the mean value of PayloadMass column
              meanPayloadMass = data_falcon9['PayloadMass'].mean()
              # Replace the np.nan values with its mean value
              data_falcon9['PayloadMass']=data_falcon9['PayloadMass'].replace(np.nan,meanPayloadMass)
              data_falcon9.isnull().sum()
```

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Methodology

Data Collection with Web Scraping

Data Collection performed following these steps :

1. Retrieve data with “get request” to the SpaceX API.
2. Decode the response with “Json”
3. Normalize the pandas dataframe using .json_normalize().
4. Clean data types, check and fill missing values.
5. **Webscraping from Wikipedia for Falcon 9 launch records with BeautifulSoup into HTML table**
6. **Parse and convert the table into a pandas dataframe.**

```
Entrée [4]:  static_url = "https://en.wikipedia.org/w/index.php?title=List_of
```

```
Entrée [17]:  # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
              soup = BeautifulSoup(res, "html.parser")
```

```
Entrée [24]:  launch_dict= dict.fromkeys(column_names)

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

              # Let's initial the launch_dict with eac
              launch_dict['Flight No.'] = []
              launch_dict['Launch site'] = []
              launch_dict['Payload'] = []
```

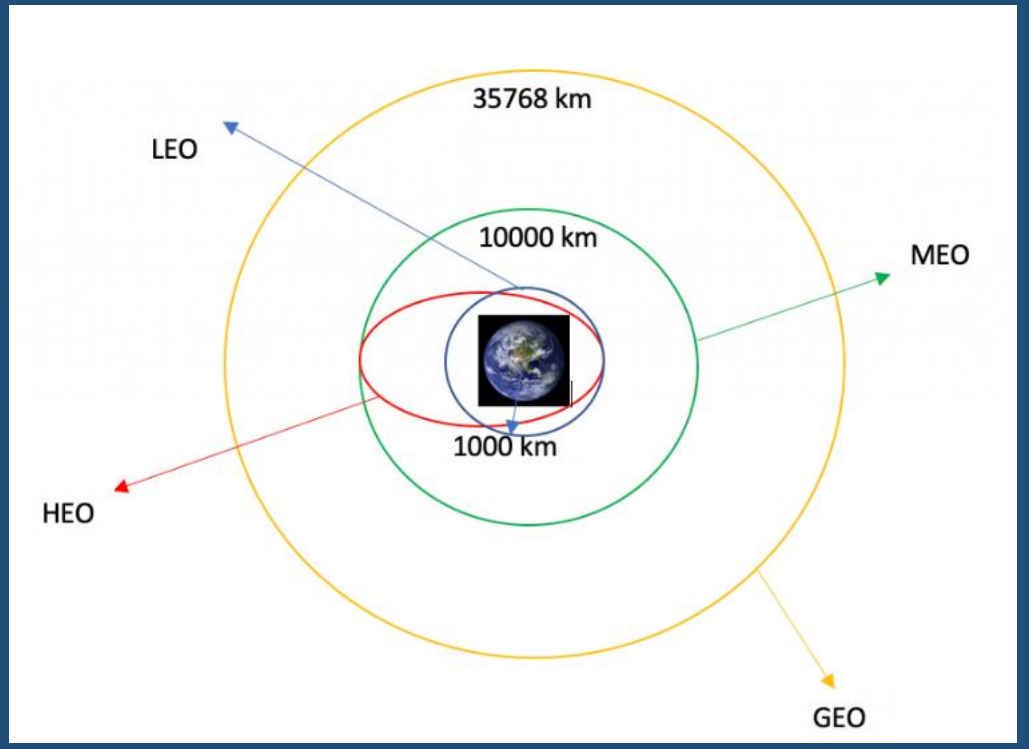
```
Entrée [26]:  df=pd.DataFrame(launch_dict)
```

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/jupyter-labs-webscraping.ipynb

Methodology

Step 2 : Data Wrangling

Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



Several different cases where the booster did not land successfully :

- Sometimes a landing was attempted but failed due to an accident;
- True Ocean : the mission outcome was successfully landed to a specific region of the ocean
- RTLS : the mission outcome was unsuccessfully landed to a ground pad.
- False ASDS : the mission outcome was unsuccessfully landed on a drone ship.
- ...

Convert those outcomes into Training Labels with 1 = the booster successfully landed & 0 = it was unsuccessful.

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Methodology

Step 2 ; Data Wrangling

1

```
Entrée [2]: df=pd.read_csv("https://cf-courses-data.s3.us.cloud-obje
            df.head(10)
```

Out[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	Launch
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCA SLC
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCA SLC

2

```
Entrée [3]: df.isnull().sum()/df.count()*100
```

```
Out[3]: FlightNumber      0.000
        Date              0.000
        BoosterVersion    0.000
```

3

```
Entrée [4]: df.dtypes
```

```
Out[4]: FlightNumber      int64
        Date             object
        BoosterVersion   object
        PayloadMass     float64
        Orbit            object
```

4

```
e [5]: # Apply value_counts() on column LaunchSit
       df['LaunchSite'].value_counts()
```

```
Out[5]: CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        Name: LaunchSite, dtype: int64
```

5

```
Entrée [6]: # Apply value_counts on Orbit c
            df['Orbit'].value_counts()
```

```
Out[6]: 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
```

6

```
Entrée [7]: # landing_outcomes = values on Outcome column
            landing_outcomes = df['Outcome'].value_counts()
            landing_outcomes
```

```
Out[7]: 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
```

7

We create a set of outcomes where the second stage did not land successfully:

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

```
Out[9]: {'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

8

```
Entrée [13]: # landing_class = 0 if bad_outcome
             # landing_class = 1 otherwise
             landing_class = []
             for key,value in df["Outcome"].items():
                 if value in bad_outcomes:
                     landing_class.append(0)
                 else:
                     landing_class.append(1)
```

This variable will represent the classification variable that re
one means the first stage landed Successfully

```
Entrée [14]: df['Class']=landing_class
             df[['Class']].head(8)
```

9

```
Entrée [15]: df.head(5)
```

Out[15]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN

1. Get dataframe from csv
2. Identify missing data %
3. Get Numerical & Categorical columns
4. Get the Nbr of launches per site
5. Nbr & Occurance of each orbit
6. Nbr & Occurance of missing outcomes per orbit
7. Get the set of bad outcommes
8. Create a landing outcome label from Outcome column
9. And here is our final dataframe

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Methodology

EDA with Visualization

Objectives :

A. Exploratory Data Analysis using Pandas and Matplotlib

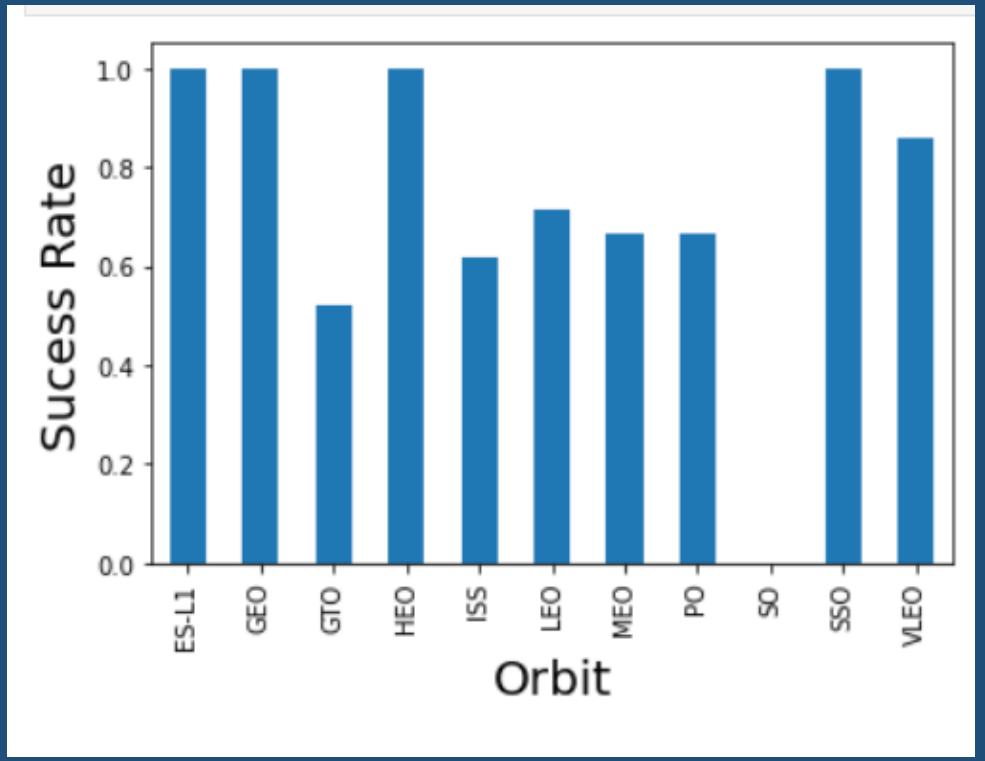
B. Preparing Data Feature Engineering

A. **Visualize Relationship between variables :**

1. Flight Number and Launch Site
2. Payload and Launch Site
3. Success rate of each orbit type
4. FlightNumber and Orbit type
5. Payload and Orbit type
6. Success yearly trend

B. Features Engineering :

1. Create dummy variables to categorical columns
2. Cast all numerical variables to Float64
3. We can now export it to a CSV



Notebook Link:

https://github.com/AmiraKerkad/IBMCapstone/blob/main/jupyter-labs-eda-dataviz.ipynb

Methodology

EDA with SQL

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

In [31]:
```
# %sql select LANDING__OUTCOME, count(LANDING__OUTCOME) as count from SPACEXTBL group by LANDING__OUTCOME having date between '04-06-2010' and '20-03-
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

Done.

Out[31]:

landing__outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt

Use SQL language to request relevant data from the database :

→ Create and insert data in the database

→ Set credentials to access database

→ Use %sql to make queries

Examples of retrieved information :

- Names of the unique launch sites in the space mission
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Total number of successful and failure mission outcomes
- Names of the booster_versions which have carried the maximum payload mass.
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Methodology

Step 3: Visual Analytics

Visual Analysis has been done using :

Follium : Use Maps to be able to find some geographical patterns about launch sites.

Plotly Dash : Use Dashboards to visualize relationship between factors and plot charts

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Methodology

Visual Analytics with Follium

The launch success rate may depend also on the location and proximities of a launch site, i.e., the initial position of rocket trajectories.

Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

- **TASK 1:** Mark all launch sites on a map
- **TASK 2:** Mark the success/failed launches for each site on the map
- **TASK 3:** Calculate the distances between a launch site to its proximities

In [3]:

```
import folium
import wget
import pandas as pd
```

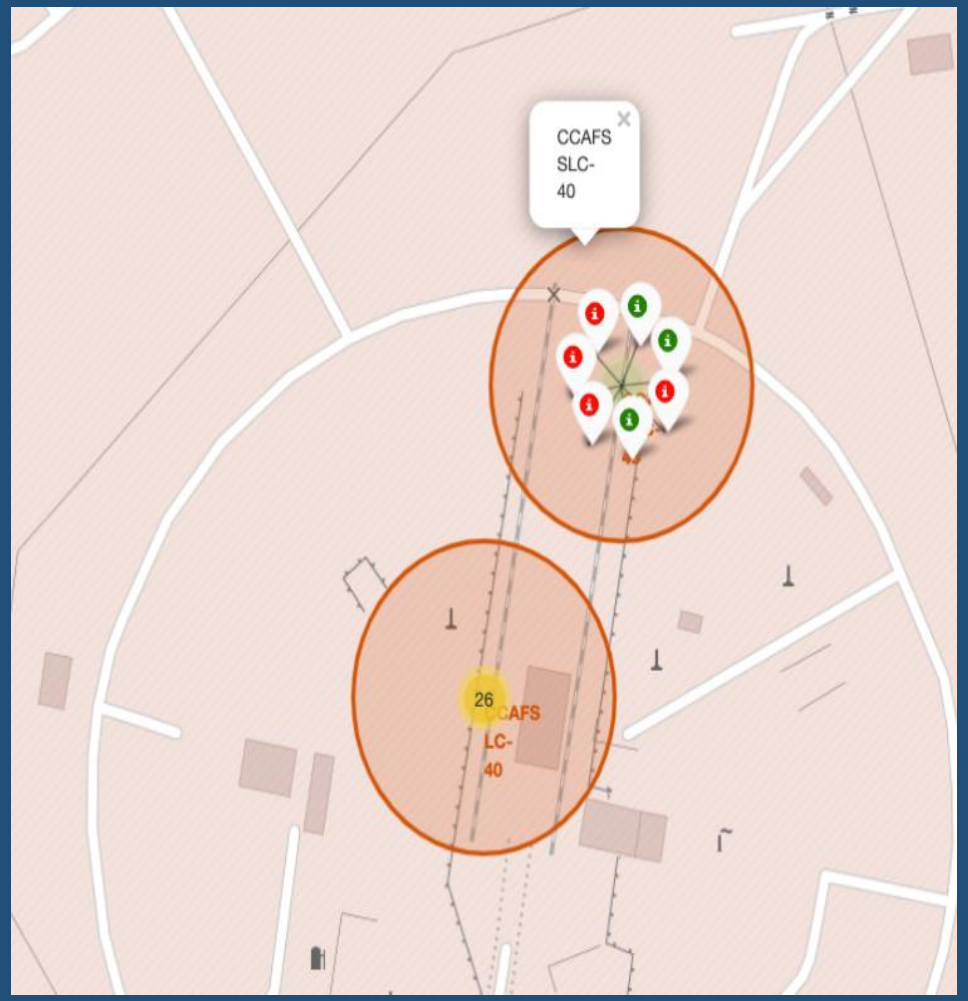
In [4]:

```
# Import folium MarkerCluster plugin
from folium.plugins import MarkerCluster
# Import folium MousePosition plugin
from folium.plugins import MousePosition
# Import folium DivIcon plugin
from folium.features import DivIcon
```

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Methodology

Visual Analytics with Follium



On Folium Map : Mark launch sites with markers, circles and lines to mark the success or failure of launches for each site.

Assign the feature launch outcomes (failure = 0, success = 1) to respective classes.

Calculate the distances between a launch site to its proximities considering their distance from :

- Railways
- Highways
- Coastlines
- Cities…

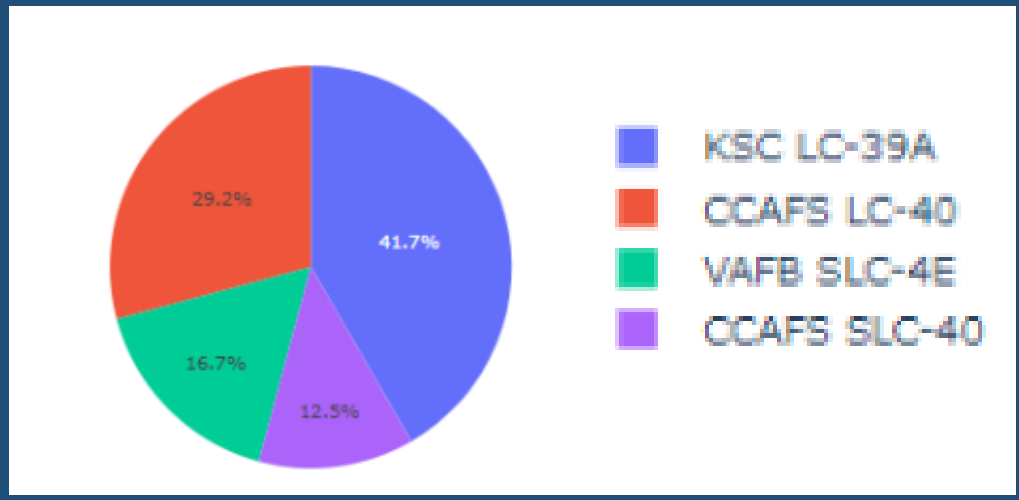
Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Methodology

Visual Analytics with Plotly Dash

Plotly Dash allows as to create interactive dashboards

With the dashboard completed, you are able to analyze SpaceX launch data in real-time,



… and answer the following questions:

1. Which site has the largest successful launches?
2. Which site has the highest launch success rate?
3. Which payload range(s) has the highest launch success rate?
4. Which payload range(s) has the lowest launch success rate?
5. Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate ?

Methodology

Step 4 : Machine Learning Prediction

Perform exploratory Data Analysis and determine Training Labels

- Create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

Notebook Link:
https://github.com/AmiraKerkad/IBMCapstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

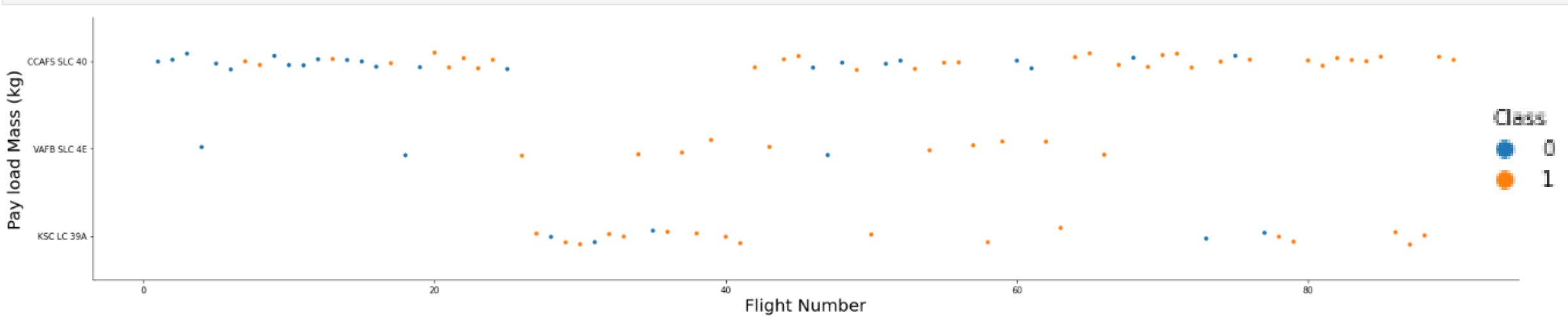
Agenda

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

Results

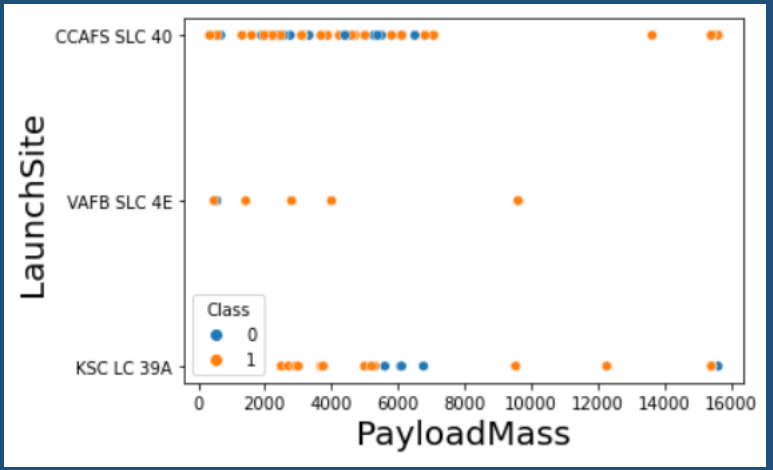
- Summary of all results
  1. Exploratory Data Analysis result
     - Comparisons & criterion relationships
  2. Interactive analytics in screenshots
     - Dashboards & Maps
  3. Predictive Analytics result

Results
EDA with Visualization

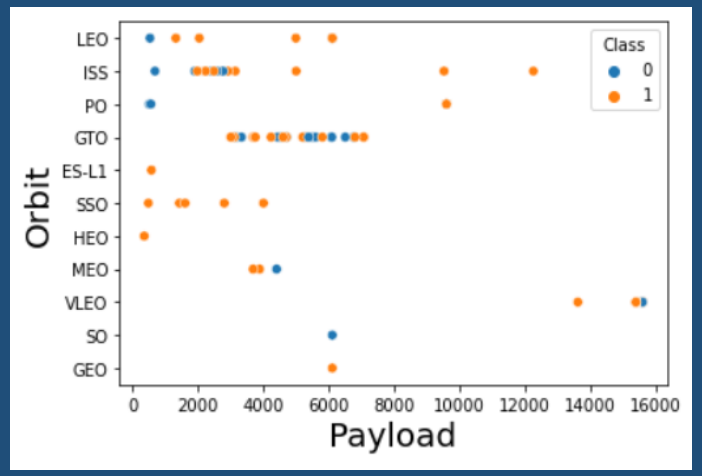


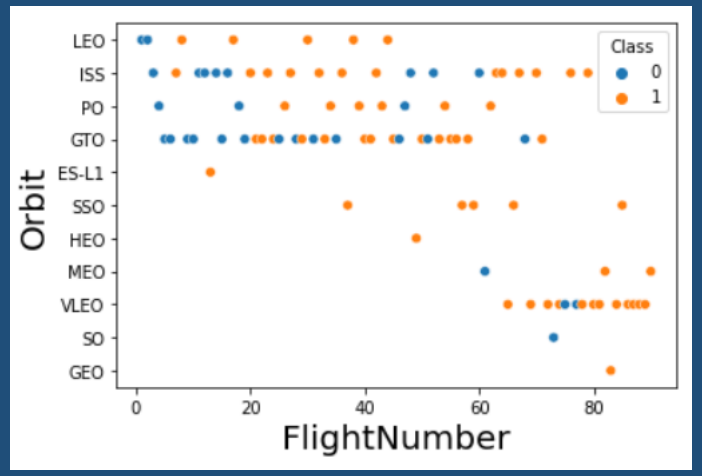
From the upper plot : the more there are flights at a launch site, the greater the success rate at that launch site.

From the right plot : Higher Payload Mass is related to more success in the 3 Launch Sites



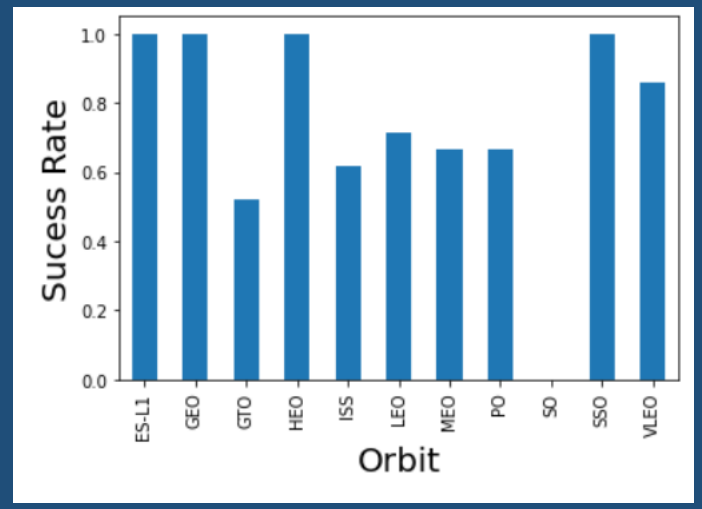
Results
EDA with Visualization

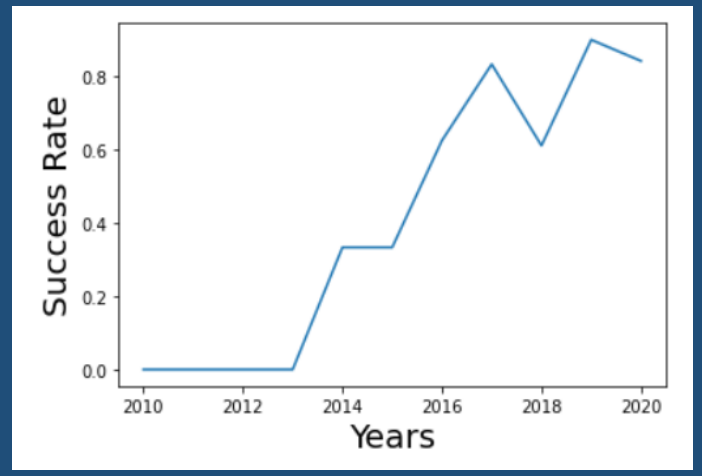




Relationship between :

1. Orbit vs. Payload
2. Orbit vs. Flight Number
3. Orbit vs. Success Rate
4. Years vs. Success Rate





Results
EDA with SQL

total_payloadmass
45596

avg_payloadmass
2928.4

firstsuccessfull_landing_date
2015-12-22

The total number of successful mission outcome is:

successoutcome
100

The total number of failed mission outcome is:

failureoutcome
1

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

Main information retrieved by aggregations and filtering using SQL :

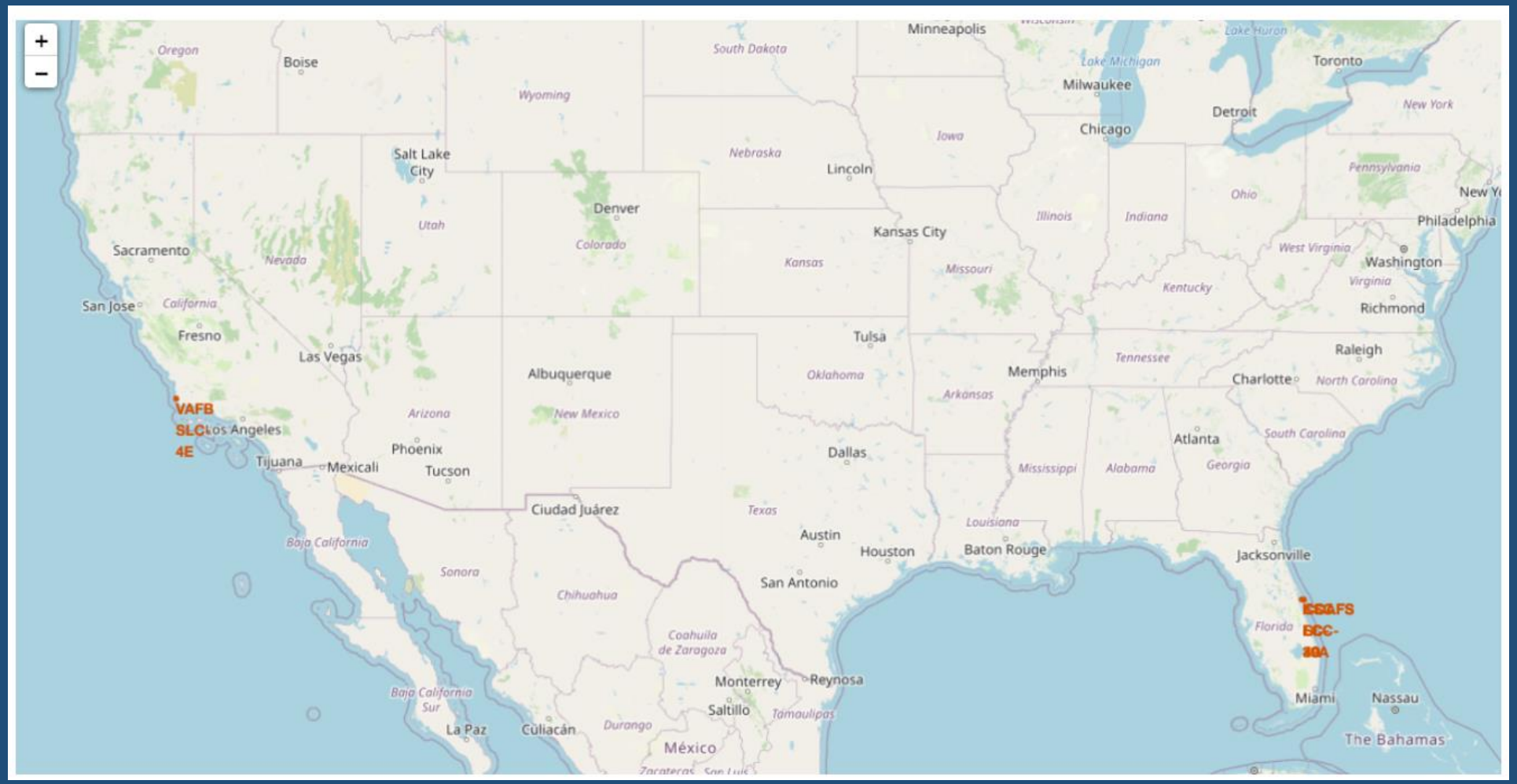
1. Total payload mass carried by boosters launched by NASA (CRS)
2. Average payload mass
3. First successful landing date
4. Total num of successful/unsucc mission outcome
5. Distinct Booster versions
6. Failed landing outcomes in drone ship on 2015
7. Landing outcomes number between 2010 & 2017

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Results

Interactive Analytics

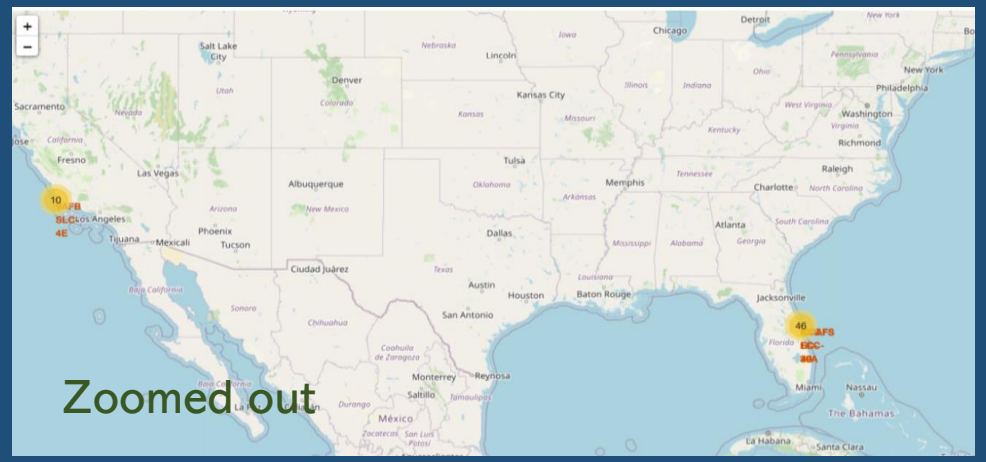


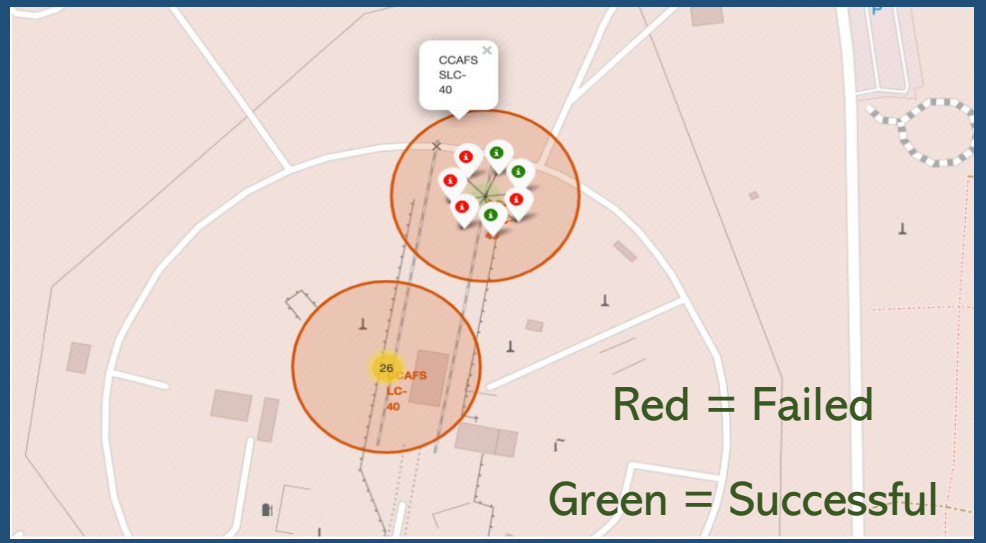
Here are the SpaceX launch sites situated in the USA

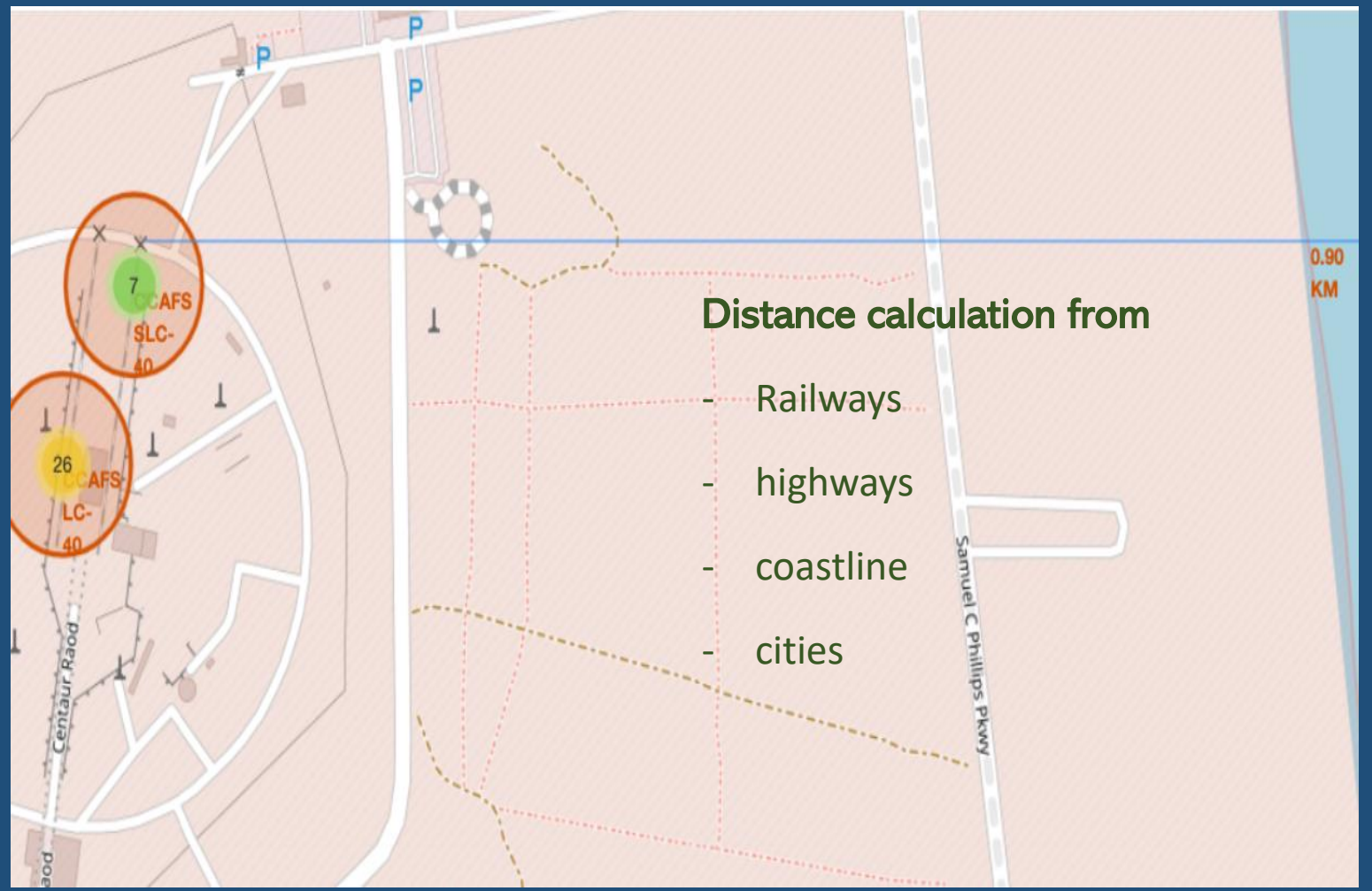
(California & Florida)

Results

Interactive Analytics

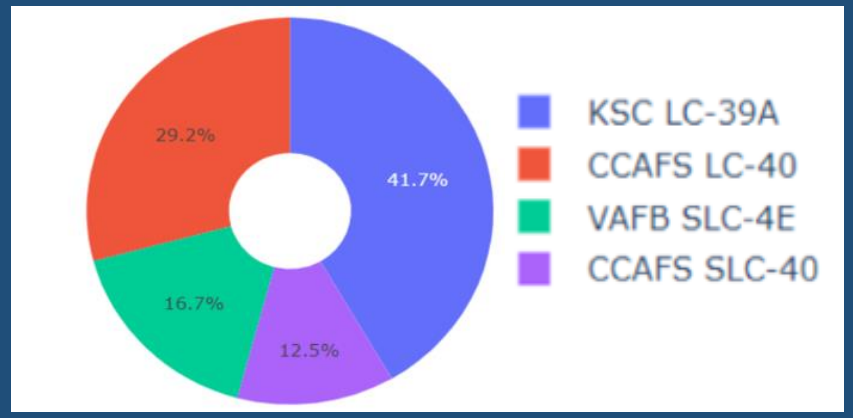




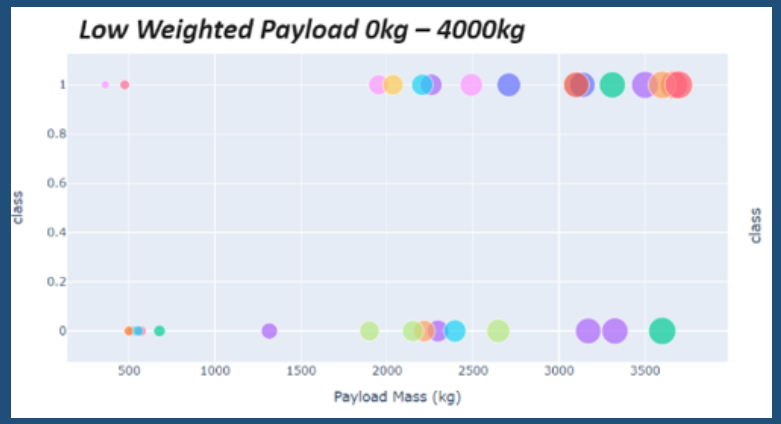


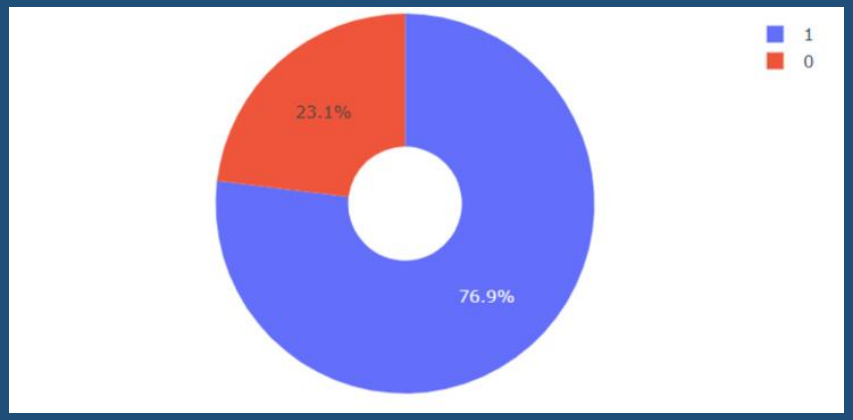
Results
Interactive Analytics

Different Dashboard are generated used Plotly Dash

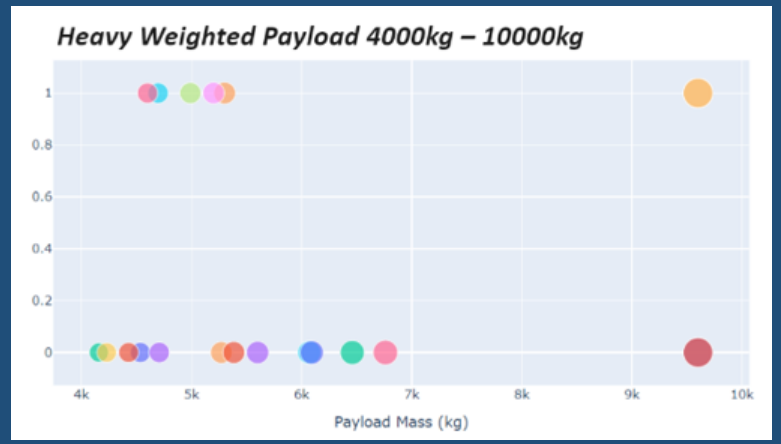


Successful rates per site : KSC LC-39A has the highest rate





Successful/Fail rate for KSC LC-39A launch site



Low Weighted Payload have more successful rate

Results

Predictive Analytics Results

```
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

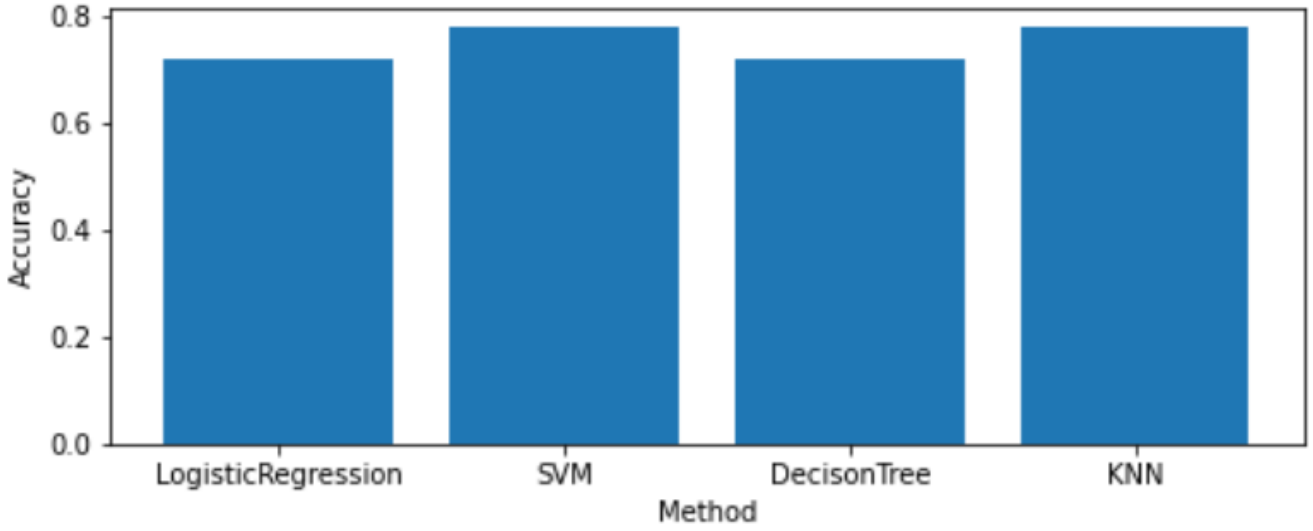
```
Train set: (72, 83) (72,)
Test set: (18, 83) (18,)
```

we can see we only have 18 test samples.

```
Y_test.shape
```

```
(18,)
```

Splitting data into Training & Test Sets



Comparing accuracy of the 4 prediction methods :

The best accuracy is given by K-NN & SVM methods.

Agenda

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

Conclusion

Main outcomes :

- The more the flight number, the greater the success rate at a launch site.
- Law weighted payloads have more successful rate.
- Highest successful landing concerns the following Orbits ES-L1, GEO, HEO, SSO, VLEO.
- KSC LC-39A had the most successful launches of any sites.
- Launch success rate increases continuously in last decade.
- KNN & SVM give the best prediction accuracy.

Agenda

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

Appendix

About Me ..

Dr. KERKAD Amira

PhD in Computer Science from ENSMA, Poitiers - France,

Lecturer at USTHB Algiers - Algeria

My GitHub : https://github.com/AmiraKerkad

My DBLP : https://dblp.org/pid/117/7923.html

My SemanticScholar : https://www.semanticscholar.org/author/Amira-Kerkad/2078850

My LinkedIn : https://dz.linkedin.com/in/kerkad-amira-b05b9764

Thank you