Data Science Capstone Project Report

# “Predicting the outcome of SpaceX Falcon9 first stage landing”

**Juan Francisco Velázquez Vadillo**
SPACE Y
Data Scientist
November 2022

**From the author**

This capstone project of the IBM Data Science Professional Certificate takes us to a crucial crossroads:

We are prepared to get and produce a huge amount of information… **now what?**

## From the author

**First things first**: Enjoy the ride!

**Second**: Do not be afraid of the number of slides necessary to display all the technical details required to complete the tasks in this course.

**Third**: Go get them! Prepare a *“**compelling** and **easy to understand story of all your data science journey** in this project”… **That´s it!**

***** From the **Submission Overview and Instructions** section of this capstone project

## From the author

To prepare this presentation I followed the instructions provided in the **Submission Overview and Instructions** section of this capstone project:

"…the **final task** of this capstone project is to create a **presentation based on the outcomes of all tasks** in previous modules and labs. Your presentation will develop into a **story of all your data science journey** in this project, and it should be **compelling** and **easy to understand**."

**From the author**

"…this **presentation** will be prepared **for your peer-data-scientists** whom are **eager to understand every technical detail** of this project."

"…this **presentation will be much more detailed and technical** than regular high-level and abstracted presentation for your executive team."

"…it should be straightforward for you to **abstract it into a high-level deck for your executive team and/or stakeholders**."

**From the author**

Thanks for your patience.

I hope you will find this information useful.

# EXECUTIVE SUMMARY

## Executive Summary

- **The race to make space travel affordable for every one is here**.
- Some providers advertise a cost upward of **165 million dollars**.
- **SpaceX** advertises **Falcon 9** rocket launches with a cost of **62 million dollars.**

- This **reduced cost** is mainly due to the fact that **SpaceX reuses the first stage** of Falcon 9 rocket.

- Through the use of **Data Science** and **Machine Learning** tools we developed **models that predict the outcome (success vs failure) of Falcon9 first stage landing.**

- This information allows us to **estimate the cost** of a Falcon9 launch.

Knowing the price of each Falcon9 launch will

**guide our strategies to bid against SpaceX**

# Table of Contents


| Table of Contents | Slide |
|---|---|
| **1.- Cover page** | 1 |
| **2.- From the author** | 2 |
| **3.- Executive summary** | 7 |
| **4.- Table of contents** | 11 |
| **5.- Introduction** | |
| Scenario | 15 |
| Goal | 16 |
| Strategy | 17 |
| **6.- Capstone project overview** | 18 |

# Table of Contents Slide

## 7.- Methodology


| | |
|---|---|
| Jupyter Notebooks generated in this work | 20 |
| Data collection SpaceX REST API | 24 |
| Data collection Web scraping | 25 |
| Data cleaning an wrangling | 26 |
| Exploratory Data Analysis with SQL | 27 |
| Exploratory Data Analysis with visualization | 28 |
| Feature engineering | 29 |
| Interactive Visual Analytics, Folium map | 30 |
| Interactive Visual Analytics, Dashboard | 31 |
| Predictive Analysis | 32 |

| Table of Contents | Slide |
|---|---|

## 8.- Results


| | |
|---|---|
| Data wrangling | 36 |
| Data Collection from SpaceX REST API | 39 |
| Data collection, Web scraping | 40 |
| Exploratory Data Analysis using SQL | 41 |
| Interactive Visual Analysis using Folium | 53 |
| Interactive Visual Analysis using Plotly Dash | 69 |
| Predictive Analysis | 81 |

# Table of Contents Slide


| | |
|---|---|
| **9.- Discussion** | 88 |
| **10.- Conclusion** | 110 |
| **11.- Appendix** (Methodology in more detail) | 112 |
| **12.- Acknowledgments** | 124 |

# Introduction

## Scenario

- Our company, **SpaceY**, wants to **lead the race to make space travel affordable for every one** .
- **Virgin Galactic**, **Rocket Lab**, **Blue Origin** and **SpaceX** are heavily investing in making space travel affordable for every one.
- By reusing the “first stage” of the Falcon9 rocket **SpaceX** can offer the **“cheapest” ticket to space**.

# Introduction

## Goal

Accurately predict SpaceX Falcon9

first stage landing outcome

# Introduction

## Strategy

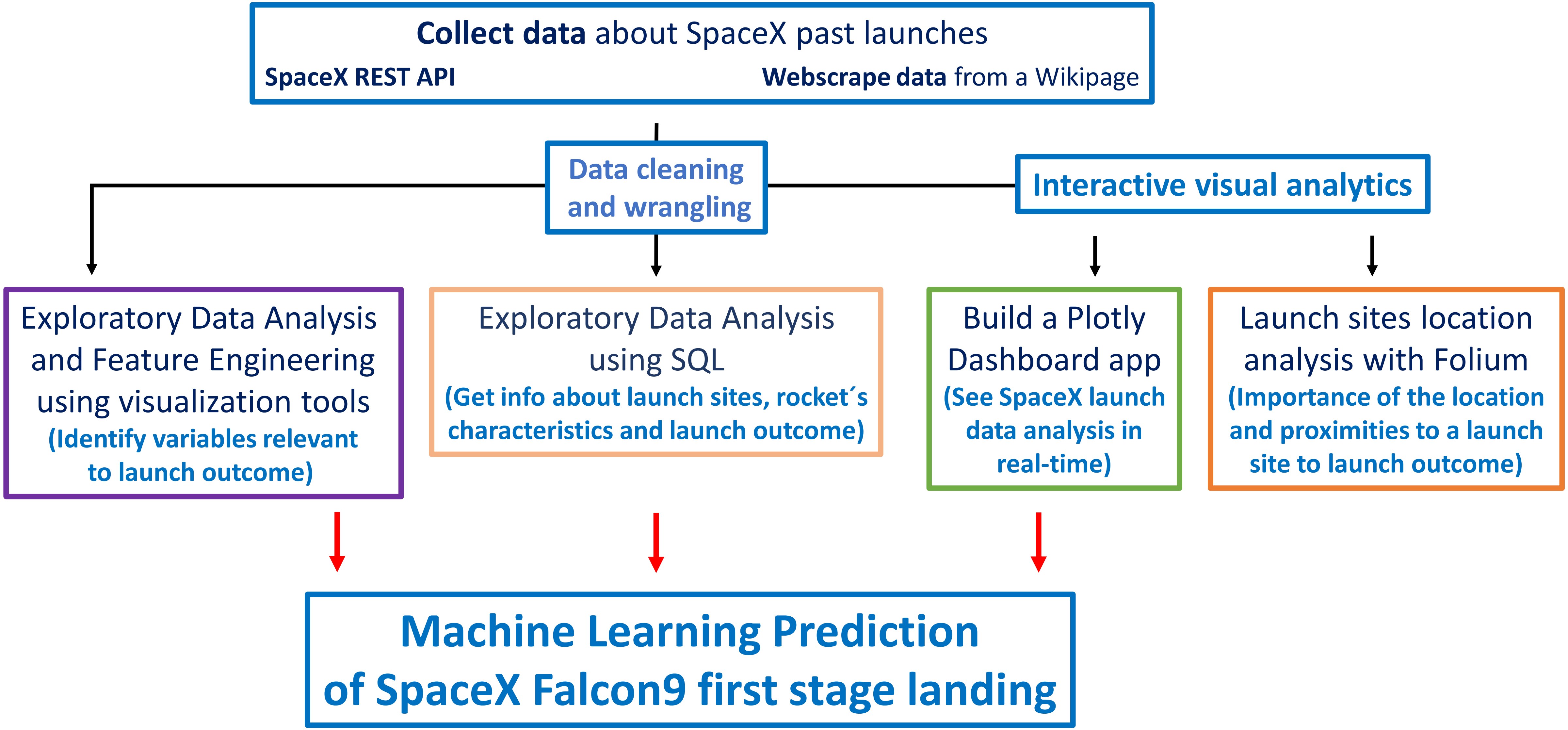
Using **Data Science** and **Machine Learning tools**

we developed models that accurately predict

the outcome (success vs failure) of the

Falcon9 first stage landing

# Capstone Project overview



# Methodology

# Jupyter Notebooks generated in this work

For detailed description of the work done in **Data collection** and **Data Wrangling** please view the following notebooks:

*When uploaded to GitHub some figures may not be displayed in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Complete the Data Collection API Lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

**Complete the Data Collection with Web Scraping lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/jupyter-labs-webscraping.ipynb

**Data Wrangling Lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

For detailed description of the work done in **Exploratory Data Analysis Using SQL** and **Exploring and Preparing Data for EDA viz** please view the following notebooks:

*When uploaded to GitHub some figures may not be displayed in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Exploratory Data Analysis using SQL**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

**Exploring and Preparing Data for EDA viz**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

For detailed description of the work done in **Interactive Visual Analytics Using Folium**
and **Make a Dashboard with PlotlyDash** please view the following notebooks:

*When uploaded to GitHub some figures may not be displayed in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Launch Sites Locations Analysis with Folium**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite%20%281%29.ipynb

**Make a Dashboard with PlotlyDash**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/spacex_dash.py

For detailed description of the work done in **Machine Learning Prediction lab** please view the following notebooks:

*When uploaded to GitHub some figures may not be displayed in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Machine Learning Prediction lab**

Assignment: Machine Learning Prediction

https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

# Data collection SpaceX REST API

Information from past launch data was obtained from

**Open Source SpaceX REST API**

(https://api.spacexdata.com/v4/launches/past)

Dataframes built: **df**, **data_falcon9**

Dataframe was filtered to only include Falcon 9 launches.

The LandingPad column retains None values.

Other missing np.nan values in the PayloaMass column

were replaced with the mean of this column.

# Data collection Web scraping

Data was collected using the package Beautiful Soap from the **Wikipedia** page titled **“List of Falcon 9 and Falcon Heavy launches”**

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

A **dataframe** was created by parsing the launch HTML tables.

## Data cleaning and wrangling

As described in the jupyter notebooks “Data Collection from SpaceX REST API”, “Data Collection from Wikipedia”, and “Data Wrangling”:

- We went through the process of fixing or eliminating incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within the datasets downloaded from SpaceX REST API and Wikipedia page “List of Falcon 9 and Falcon Heavy launches”.

- We found patterns in the data that helped determine what would be the label and relevant variables for training Machine Learning supervised models.

# Exploratory Data Analysis with SQL

**Goal**: to **identify attributes** that can be used to determine if the first stage can be reused.

We created the **SPACEXDATASET** in **DB2** database.

We wrote an executed **SQL queries** to gain information from this dataset.

# Exploratory Data Analysis with Visualization

**Goal**: to identify variables that can be used to determine if the first stage can be reused.

We used the **SpaceX** dataset provided by the IBM team and generated the corresponding **dataframe**.

Through **scatter point charts** we analyzed the effect of several variables on the launch outcome.

# Exploratory Data Analysis: Feature engineering

**Goal**: create a **clean dataframe** containing those **variables** that can be used **to determine if the first stage can be reused**.

We created the **features** dataframe with variables that affect the first stage **success rate landing**.

We applied the **get_dummies()** function on the categorical columns of **features** dataframe.

We cast all **numeric columns** to float64.

## Interactive Visual Analytics:

Launch Sites Locations Analysis with **Folium**

**Goal**: to create, using Folium, an **interactive map** that facilitates the identification of **relevant factors** involved in finding an **optimal location** for **building a launch site**.

# Interactive Visual Analytics:

## Build a Dashboard Application with Plotly Dash

**Goal**: to build a **Plotly Dash application** for users to perform **interactive visual analytics** on SpaceX launch data in real-time.

The dataset, provided by the IBM team, is the csv document

**"spacex_launch_dash.csv"**

## Predictive Analysis:

**Goal**: to build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully.

**Data** was **standardized** and **split** into **training and testing data**.

**Grid Search** was performed on the trained data to find the **hyperparameters** that allow a given Machine Learning model to perform best.

# Predictive Analysis:

**Machine Learning supervised learning techniques used:**

Logistic Regression

Support Vector Machine

Decision Trees

K nearest neighbors

## Predictive Analysis:

We **output the confusion matrix** and determined which model best predicts the outcome of each Falcon9 launch.

# RESULTS

# RESULTS

## Data wrangling

Dataset was imported from

URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv‘.

We identified and calculated the percentage of the missing values in each attribute.

| | |
|---|---|
| FlightNumber | 0.000 |
| Date | 0.000 |
| BoosterVersion | 0.000 |
| PayloadMass | 0.000 |
| Orbit | 0.000 |
| LaunchSite | 0.000 |
| Outcome | 0.000 |
| Flights | 0.000 |
| GridFins | 0.000 |
| Reused | 0.000 |
| Legs | 0.000 |
| LandingPad | 40.625 |
| Block | 0.000 |
| ReusedCount | 0.000 |
| Serial | 0.000 |
| Longitude | 0.000 |
| Latitude | 0.000 |

# Data wrangling

We calculated the number of launches on each site.

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

We calculated the number and occurrence of each orbit.

```
Number of each orbit:  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
Occurrence of each orbit:
 GTO      30.000000
ISS      23.333333
VLEO     15.555556
PO       10.000000
LEO       7.777778
SSO       5.555556
MEO       3.333333
ES-L1     1.111111
HEO       1.111111
SO        1.111111
GEO       1.111111
Name: Orbit, dtype: float64
```

# Data wrangling

We calculated the number and occurrence of mission outcome per orbit type.

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

We created a landing outcome label from Outcome column.

Successful landing: 0
Failed landing: 1

| | Class |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |

We calculated the successful launch rate: 0.666

# Data Collection

Rocket launch data requested from **SpaceX REST API** with the following URL:

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

**df** : clean dataframe used to predict the outcome of the Falcon9 first stage landing

**After data cleaning and wrangling** Dataframe used in Machine Learning Prediction

```
df.head(5)
```

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block | ReusedCount | Serial | Longitude | Latitude | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0003 | -80.577366 | 28.561857 | 0 |
| 1 | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0005 | -80.577366 | 28.561857 | 0 |
| 2 | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0007 | -80.577366 | 28.561857 | 0 |
| 3 | 4 | 2013-09-29 | Falcon 9 | 500.000000 | PO | VAFB SLC 4E | False Ocean | 1 | False | False | False | NaN | 1.0 | 0 | B1003 | -120.610829 | 34.632093 | 0 |
| 4 | 5 | 2013-12-03 | Falcon 9 | 3170.000000 | GTO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B1004 | -80.577366 | 28.561857 | 0 |

# Data Collection

Perform **web scraping** to collect Falcon 9 historical launch records from a Wikipedia page "List of Falcon 9 and Falcon Heavy launches"
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
Create a data frame by **parsing the launch HTML tables**

**First five rows of dataframe df** Dataframe used in Exploratory Data Analysis using SQL

```
In [20]: df.head()
Out[20]:
```

| | Flight No. | Launch site | Payload | Payload mass | Orbit | Customer | Launch outcome | Version Booster | Booster landing | Date | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | CCAFS | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success\n | F9 v1.0B0003.1 | Failure | 4 June 2010 | 18:45 |
| 1 | 2 | CCAFS | Dragon | 0 | LEO | NASA (COTS)\nNRO | Success | F9 v1.0B0004.1 | Failure | 8 December 2010 | 15:43 |
| 2 | 3 | CCAFS | Dragon | 525 kg | LEO | NASA (COTS) | Success | F9 v1.0B0005.1 | No attempt\n | 22 May 2012 | 07:44 |
| 3 | 4 | CCAFS | SpaceX CRS-1 | 4,700 kg | LEO | NASA (CRS) | Success\n | F9 v1.0B0006.1 | No attempt | 8 October 2012 | 00:35 |
| 4 | 5 | CCAFS | SpaceX CRS-2 | 4,877 kg | LEO | NASA (CRS) | Success\n | F9 v1.0B0007.1 | No attempt\n | 1 March 2013 | 15:10 |

# Exploratory Data Analysis using SQL

- These are the type of questions that can be asked, through **SQL querying** to get insight from the data.
- **SPACEXTBL** table data was provided by the IBM team

1.- **Display the names of the unique launch sites** in the space mission:

```
[10]: %sql select distinct "Launch_Site" from SPACEXTBL;

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

[10]:

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

# Exploratory Data Analysis using SQL

2.- Display the number of successful and failed missions.

**Number of successful and failed missions**

```
%sql select count("Mission_Outcome") as "Number_of_successful_missions" from SPACEXTBL where "Mission_Outcome" like "Succ%";
```

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

| Number_of_successful_missions |
|---|
| 100 |

```
%sql select count("Mission_Outcome") as "Number_of_failed_missions" from SPACEXTBL where "Mission_Outcome" like "Fail%";
```

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

| Number_of_failed_missions |
|---|
| 1 |

# Exploratory Data Analysis using SQL

3.- Display the date of the first successful landing outcome in ground pad.

**Date of the first successful landing outcome in ground pad**

```
%sql select min("Date") as "First_successful_landing_outcome", "Landing _Outcome" as "Ground-pad" from SPACEXTBL where "Landing
```

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

| First_successful_landing_outcome | Ground-pad |
|---|---|
| 01-05-2017 | Success (ground pad) |

## Exploratory Data Analysis using SQL

4.- Display 5 records where launch sites begin with the string 'CCA':

```
[11]: %sql select "Launch_Site" from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5;

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

[11]:

| Launch_Site |
|---|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

# Exploratory Data Analysis using SQL

5.- Display the total payload mass carried by boosters launched by NASA (CRS):

```
[12]: %sql select  "PAYLOAD_MASS__KG_", "Customer" from SPACEXTBL where "Customer" = "NASA (CRS)" limit 5;

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

[12]:

| PAYLOAD_MASS__KG_ | Customer |
|---|---|
| 500 | NASA (CRS) |
| 677 | NASA (CRS) |
| 2296 | NASA (CRS) |
| 2216 | NASA (CRS) |
| 2395 | NASA (CRS) |

# Exploratory Data Analysis using SQL

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

```
[13]: %sql select  AVG("PAYLOAD_MASS__KG_") as "AVERAGE_PAYLOADMASS_KG", "Booster_Version" from SPACEXTBL where "Booster_Version" = "

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

[13]:

| AVERAGE_PAYLOADMASS_KG | Booster_Version |
|---|---|
| 2928.4 | F9 v1.1 |

# Exploratory Data Analysis using SQL

7.- List the date when the first successful landing outcome in ground pad was achieved.

```
[15]: %sql select min("Date") as "First_successful_landing_outcome", "Landing _Outcome" as "Ground-pad" from SPACEXTBL where "Landing

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

[15]:

| First_successful_landing_outcome | Ground-pad |
|---|---|
| 01-05-2017 | Success (ground pad) |

# Exploratory Data Analysis using SQL

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

```
[16]: %sql select "Booster_Version", "PAYLOAD_MASS__KG_", "Landing _Outcome" from SPACEXTBL where ("PAYLOAD_MASS__KG_" BETWEEN 4000 A
```

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

[16]:

| Booster_Version | PAYLOAD_MASS__KG_ | Landing _Outcome |
|---|---|---|
| F9 FT B1022 | 4696 | Success (drone ship) |
| F9 FT B1026 | 4600 | Success (drone ship) |
| F9 FT B1021.2 | 5300 | Success (drone ship) |
| F9 FT B1031.2 | 5200 | Success (drone ship) |

# Exploratory Data Analysis using SQL

9.- List the total number of successful and failure mission outcomes.

```
[18]: %sql select count("Mission_Outcome") as "Number_of_successful_missions" from SPACEXTBL where "Mission_Outcome" like "Succ%";

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

[18]:

| Number_of_successful_missions |
|---|
| 100 |

```
[19]: %sql select count("Mission_Outcome") as "Number_of_failed_missions" from SPACEXTBL where "Mission_Outcome" like "Fail%";

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

[19]:

| Number_of_failed_missions |
|---|
| 1 |

# Exploratory Data Analysis using SQL

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

```
[20]: %sql select "Booster_Version", "PAYLOAD_MASS__KG_" as "Carrying_maximum_payload_mass_kg" from SPACEXTBL where "PAYLOAD_MASS__KG

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

[20]:

| Booster_Version | Carrying_maximum_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 |

# Exploratory Data Analysis using SQL

11.- 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.

```
[128]: %sql select  substr("Date", 4, 2) as "Month_of_2015", "Landing _Outcome", "Booster_Version", "Launch_Site"  from SPACEXTBL wher

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

[128]:

| Month_of_2015 | Landing _Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

# Exploratory Data Analysis using SQL

12.- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
[150]: %sql select  "Landing _Outcome", count(*) from SPACEXTBL where ("Date" between "04-06-2010" and "20-03-2017") AND "Landing _Out

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

[150]:

| Landing _Outcome | count(*) |
|---|---|
| Success | 20 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |

# Interactive Visual Analytics Using Folium

1.- Download and read the `spacex_launch_geo.csv` (Provided by the IBM team)
2.- Generate the corresponding spacex_df dataframe
3.- See the attributes

```
spacex_csv_file = wget.download('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/
spacex_df=pd.read_csv(spacex_csv_file)
spacex_df.head()
```

[4]:

| | Flight Number | Date | Time (UTC) | Booster Version | Launch Site | Payload | Payload Mass (kg) | Orbit | Customer | Landing Outcome | class | Lat | Long |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0.0 | LEO | SpaceX | Failure (parachute) | 0 | 28.562302 | -80.577356 |
| **1** | 2 | 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel o... | 0.0 | LEO (ISS) | NASA (COTS) NRO | Failure (parachute) | 0 | 28.562302 | -80.577356 |
| **2** | 3 | 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2+ | 525.0 | LEO (ISS) | NASA (COTS) | No attempt | 0 | 28.562302 | -80.577356 |
| **3** | 4 | 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500.0 | LEO (ISS) | NASA (CRS) | No attempt | 0 | 28.562302 | -80.577356 |
| **4** | 5 | 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677.0 | LEO (ISS) | NASA (CRS) | No attempt | 0 | 28.562302 | -80.577356 |

# Interactive Visual Analytics Using Folium

Get the coordinates of each launch site.

```
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

[5]:

| | Launch Site | Lat | Long |
|---|---|---|---|
| 0 | CCAFS LC-40 | 28.562302 | -80.577356 |
| 1 | CCAFS SLC-40 | 28.563197 | -80.576820 |
| 2 | KSC LC-39A | 28.573255 | -80.646895 |
| 3 | VAFB SLC-4E | 34.632834 | -120.610745 |

# Interactive Visual Analytics Using Folium

Data about the success/failed launches for each launch site on the Folium map.
Class **1** = sussessful launch
Class **0** = failed launch

```
[9]: spacex_df.tail(10)
```

[9]:

| | Launch Site | Lat | Long | class |
|---|---|---|---|---|
| 46 | KSC LC-39A | 28.573255 | -80.646895 | 1 |
| 47 | KSC LC-39A | 28.573255 | -80.646895 | 1 |
| 48 | KSC LC-39A | 28.573255 | -80.646895 | 1 |
| 49 | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 |
| 50 | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 |
| 51 | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 |
| 52 | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 |
| 53 | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 |
| 54 | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 |
| 55 | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 |

# Interactive Visual Analytics Using Folium

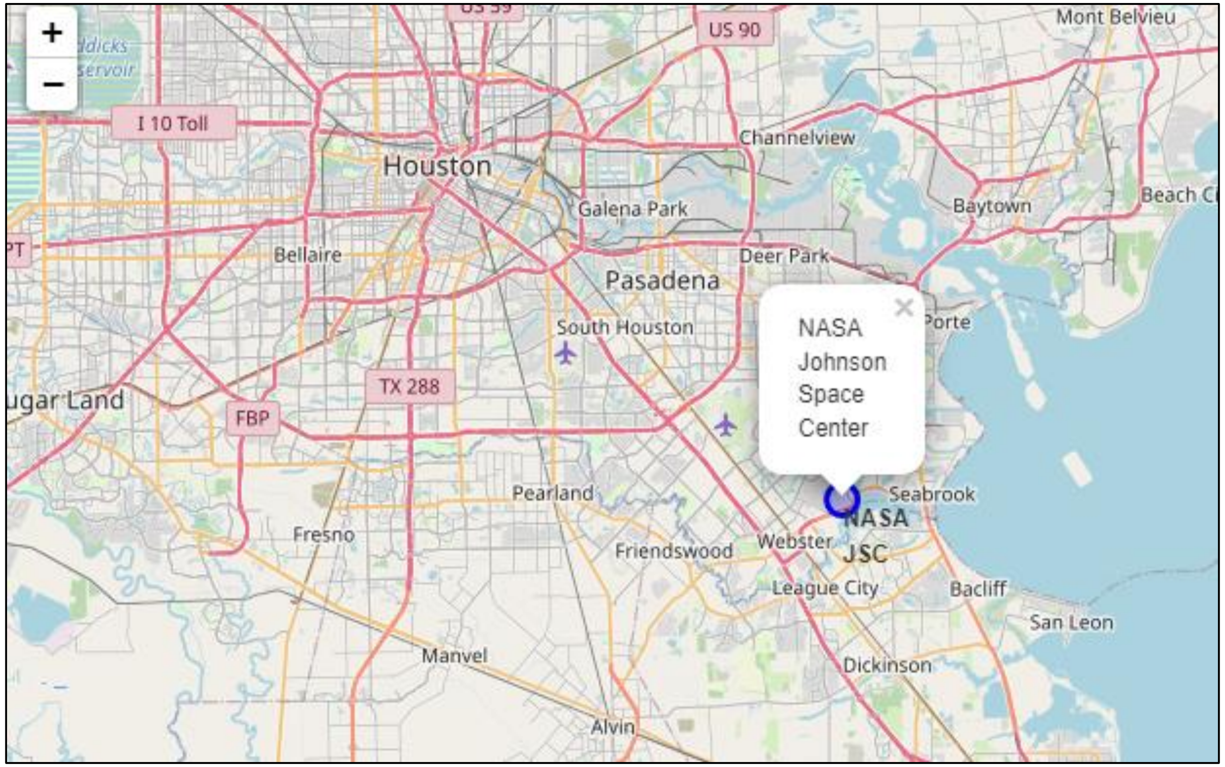
Preparing data to add a **marker color** to the Folium map
to identify **successful launch (class 1, Green)** or **failed launch (class 0, red)**

[11]:

| | Launch Site | Lat | Long | class | marker_color |
|---|---|---|---|---|---|
| **46** | KSC LC-39A | 28.573255 | -80.646895 | 1 | green |
| **47** | KSC LC-39A | 28.573255 | -80.646895 | 1 | green |
| **48** | KSC LC-39A | 28.573255 | -80.646895 | 1 | green |
| **49** | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 | green |
| **50** | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 | green |
| **51** | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 | red |
| **52** | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 | red |
| **53** | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 | red |
| **54** | CCAFS SLC-40 | 28.563197 | -80.576820 | 1 | green |
| **55** | CCAFS SLC-40 | 28.563197 | -80.576820 | 0 | red |

# Interactive Visual Analytics Using Folium

**How to locate a site in a map:** Create a red circle at **NASA Johnson Space Center's** coordinate with a popup label showing its name (NASA JSC).



# Interactive Visual Analytics Using Folium

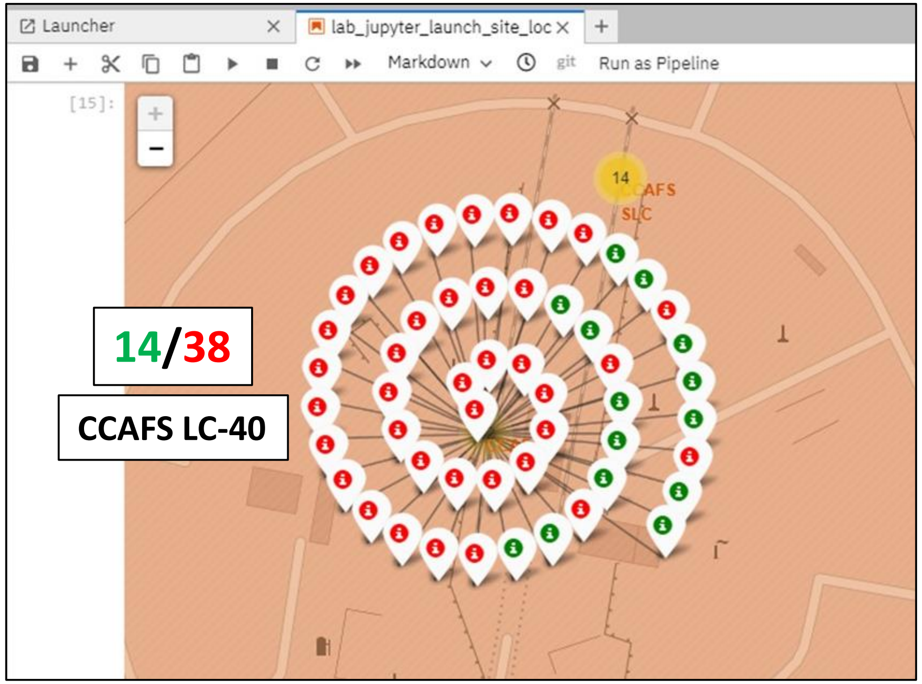
## Locating launch sites in a Folium map

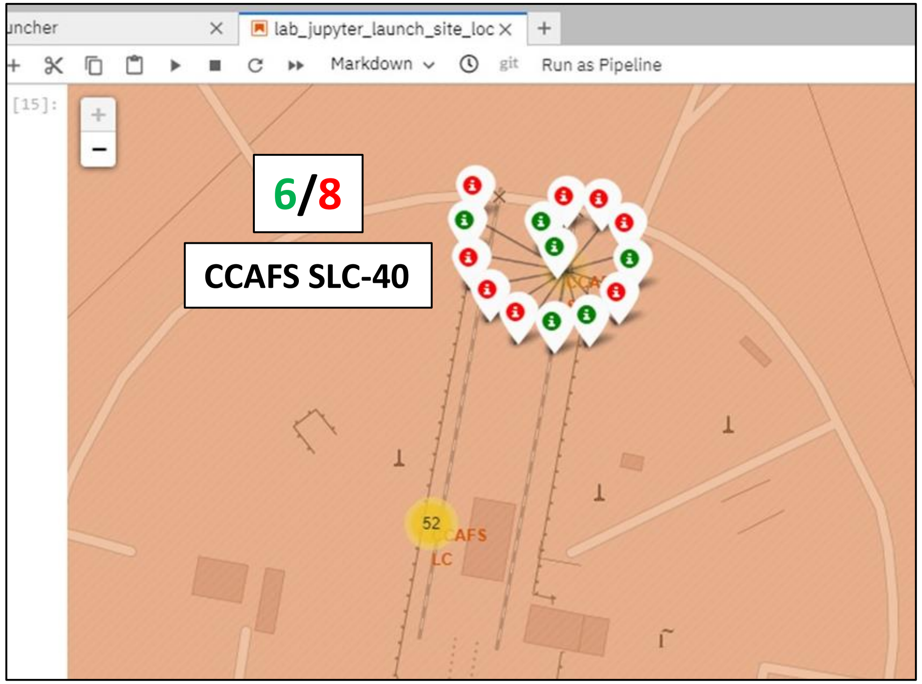
# Interactive Visual Analytics Using Folium

**Zoom In, visualizing launch sites**

GREEN: **SUCCESSFUL** Falcon9 first stage landing

RED: **FAILED** Falcon9 first stage landing



# Interactive Visual Analytics Using Folium

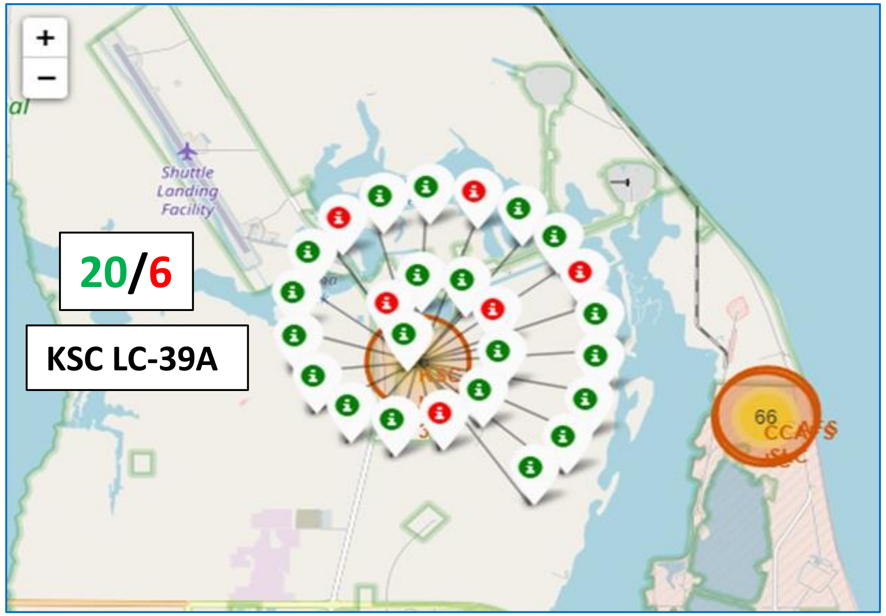
**Zoom In, visualizing launch sites**

GREEN: **SUCCESSFUL** Falcon9 first stage landing

RED: **FAILED LAUNCHES** Falcon9 first stage landing


8/12

VAFB SLC 4E




20/6

KSC LC-39A

# Interactive Visual Analytics Using Folium

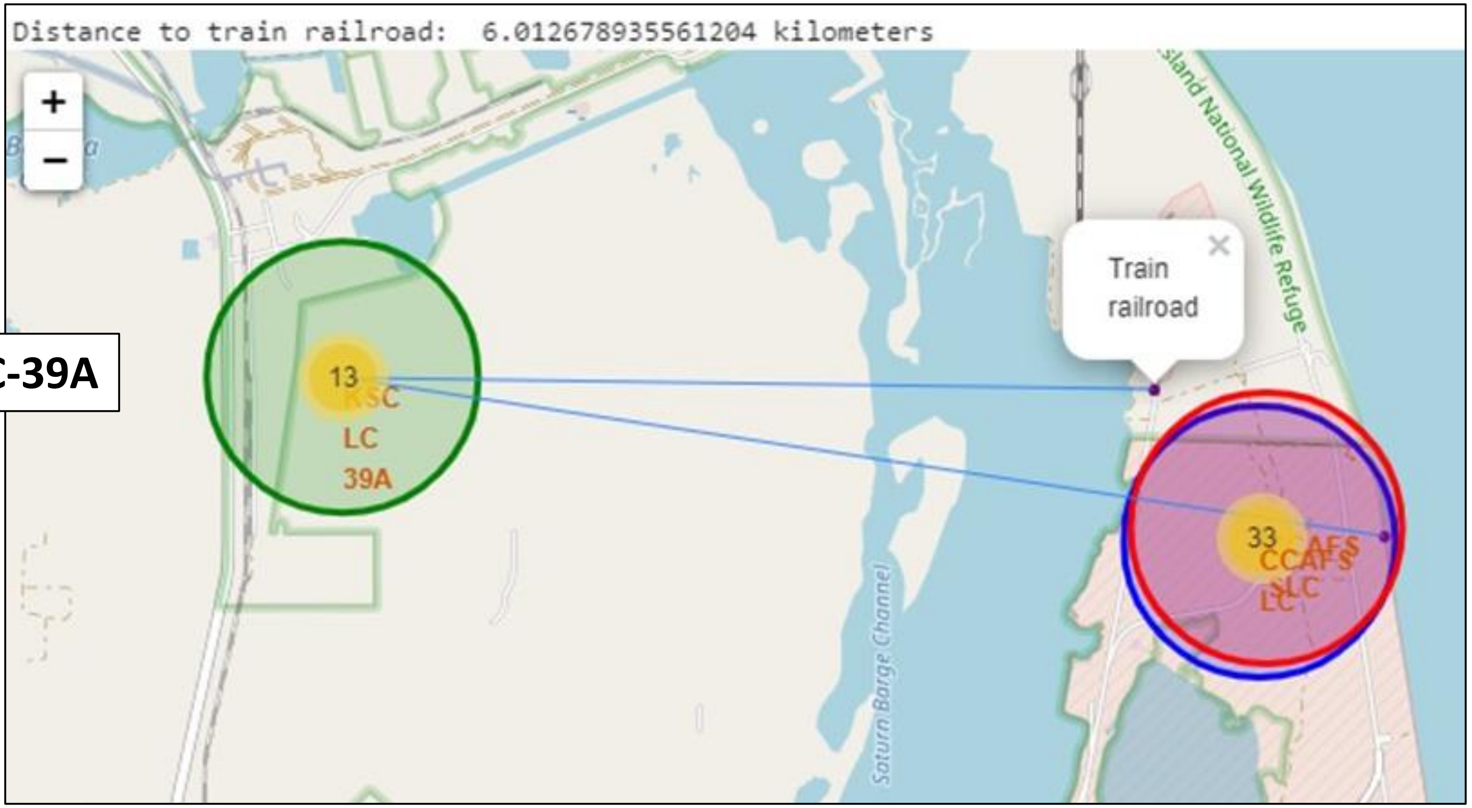
Calculating the distance between a launch site to its proximities.
Distance between **KSCLC-39A** launch site and **coast line**:
**13.762 kilometers**

KSCLC-39A

# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.
Distance between **KSCLC-39A** launch site and **train railroad**:
**6.012 kilometers**

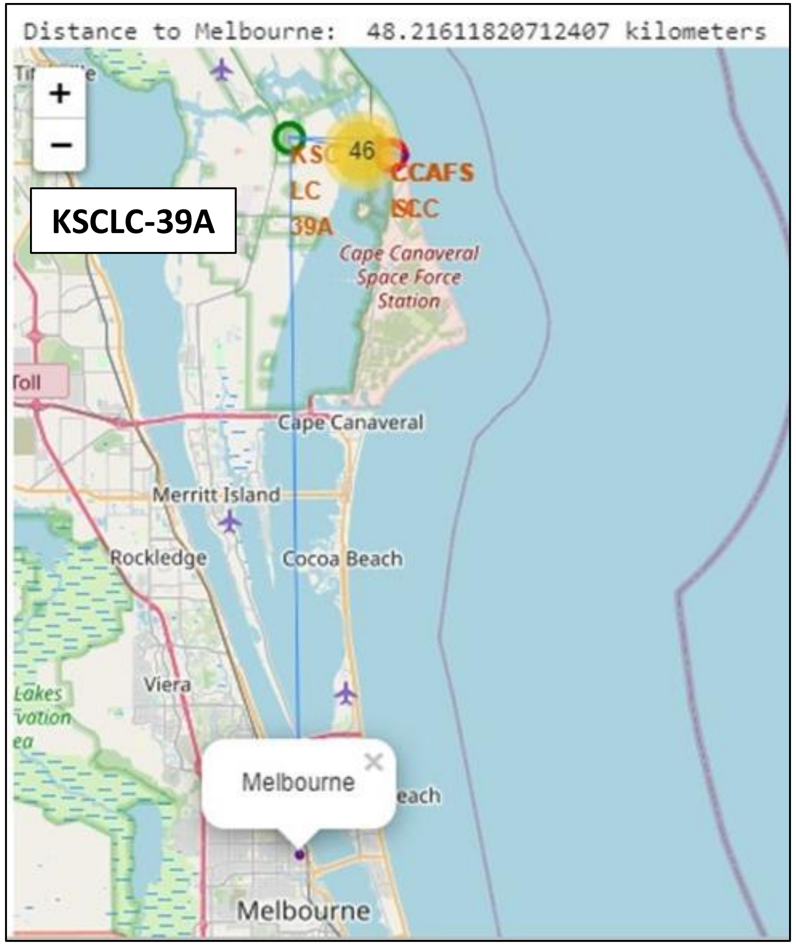
KSCLC-39A



# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

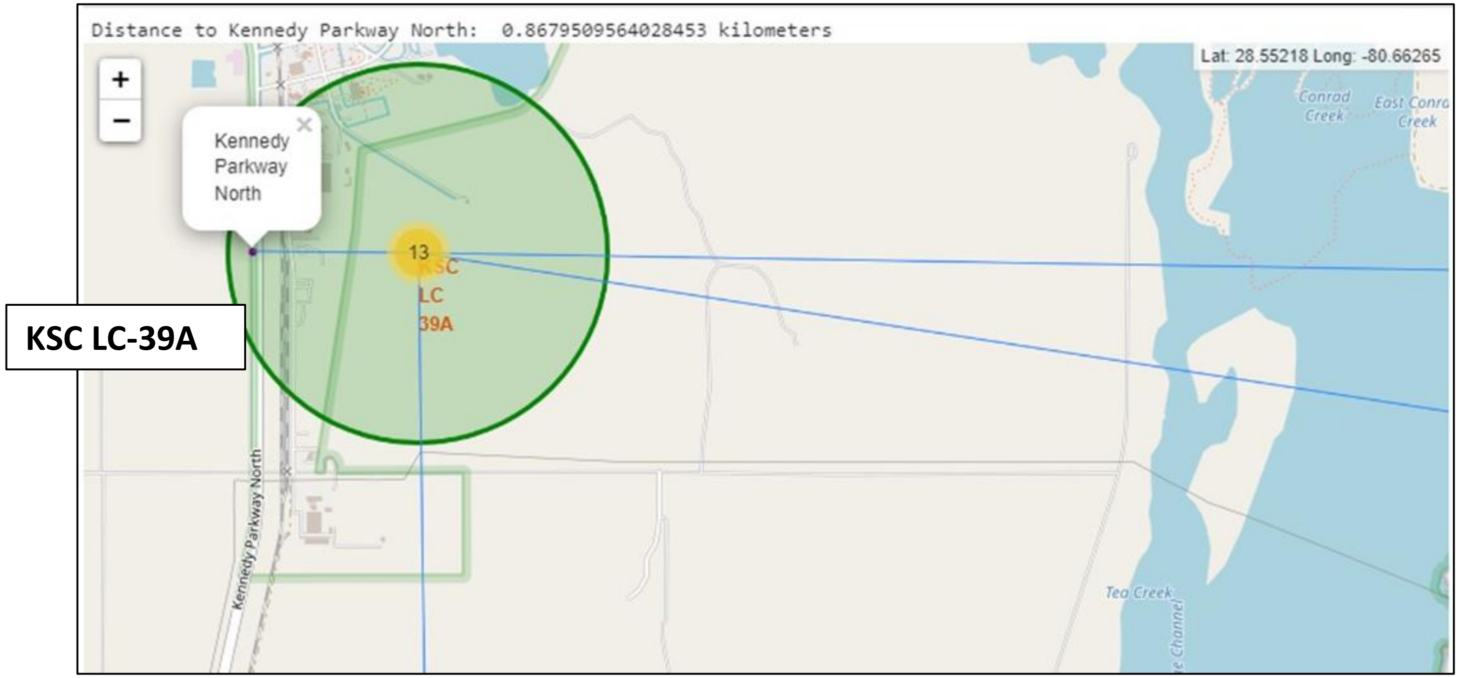
Distance between **KSCLC-39A** launch site and **Melbourne city**: **48.216 kilometers**



# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

Distance between **KSCLC-39A** launch site and **Kennedy Parkway North**: **0.8679 kilometers**

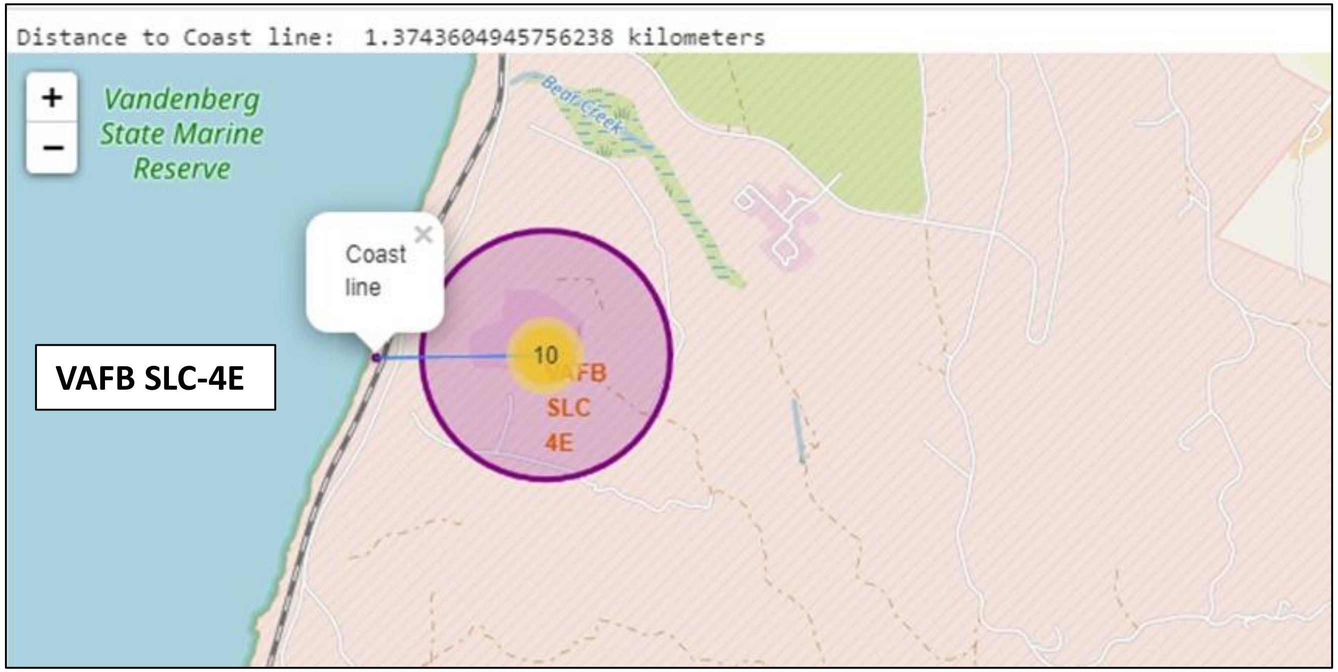


# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

Distance between **VAFB SLC-4E** launch site and **Coast Line**:

**1.3743 kilometers**

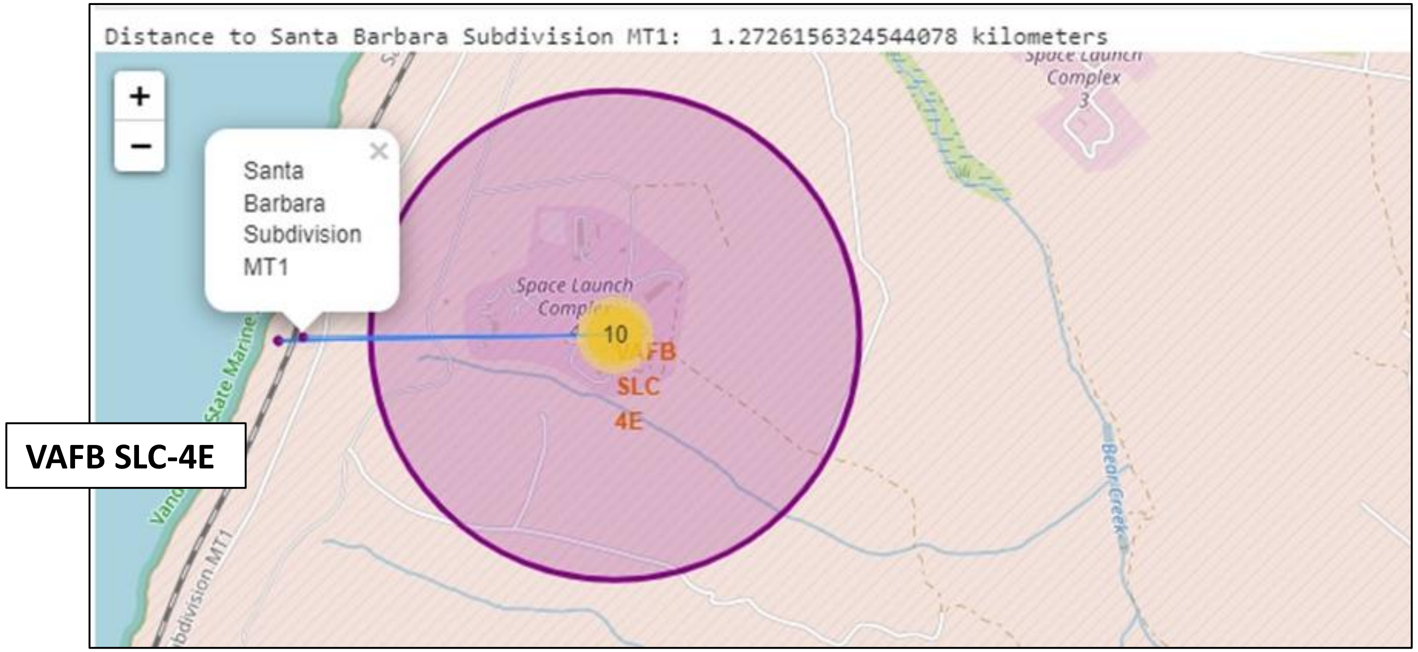


# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

Distance between **VAFB SLC-4E** launch site and **SBSD MT1** :

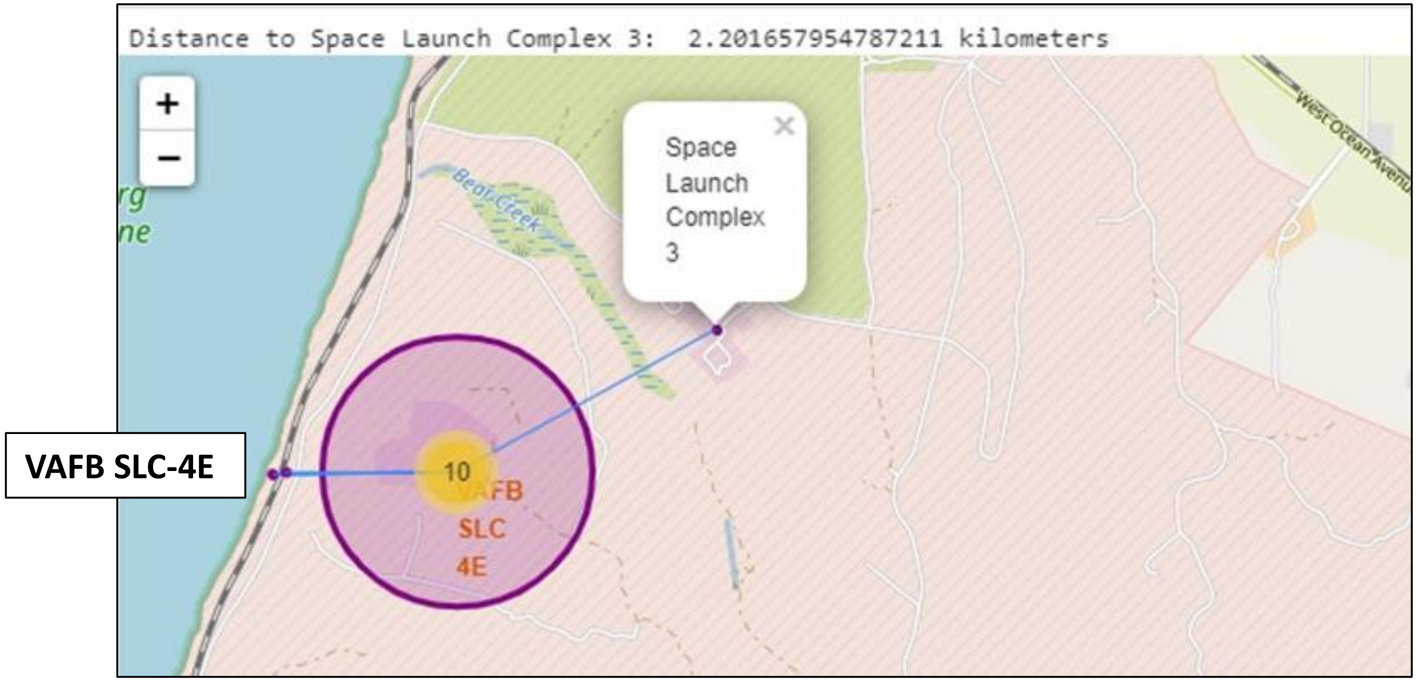
**1.2726 kilometers**



# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

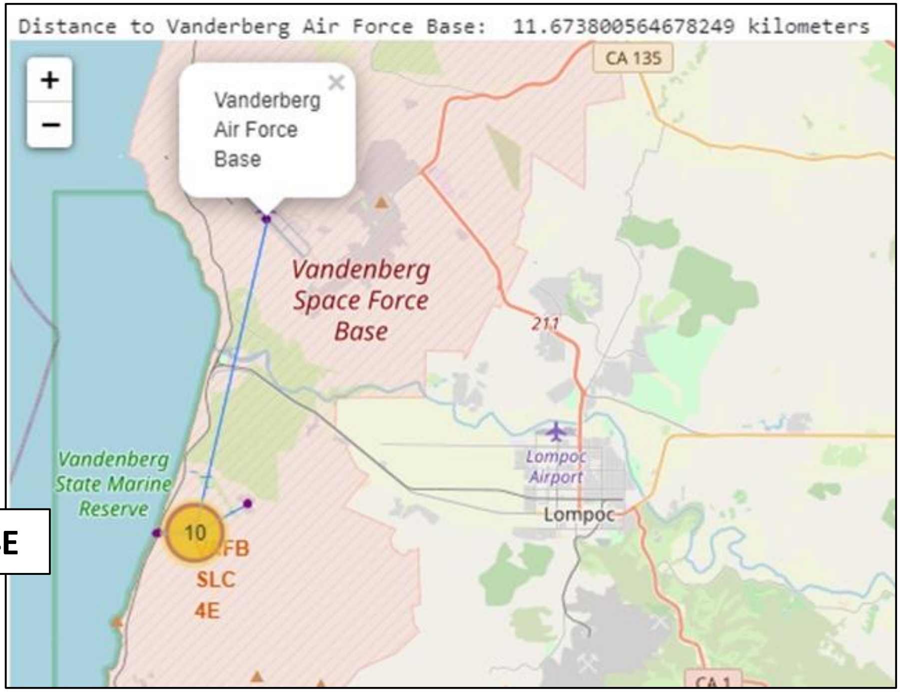
Distance between **VAFB SLC-4E** launch site and **Space Launch Complex 3**: **2.2016 kilometers**



# Interactive Visual Analytics Using Folium

Calculating the distance between a launch site to its proximities.

Distance between **VAFB SLC-4E** launch site and **Vanderberg Air Force Base**: **11.6738 kilometers**




VAFB SLC-4E

# Interactive Visual Analytics Using Plotly Dash

## Visualising information in an interactive Dashboard

Dataset provided by the IBM team: **spacex_csv_file**

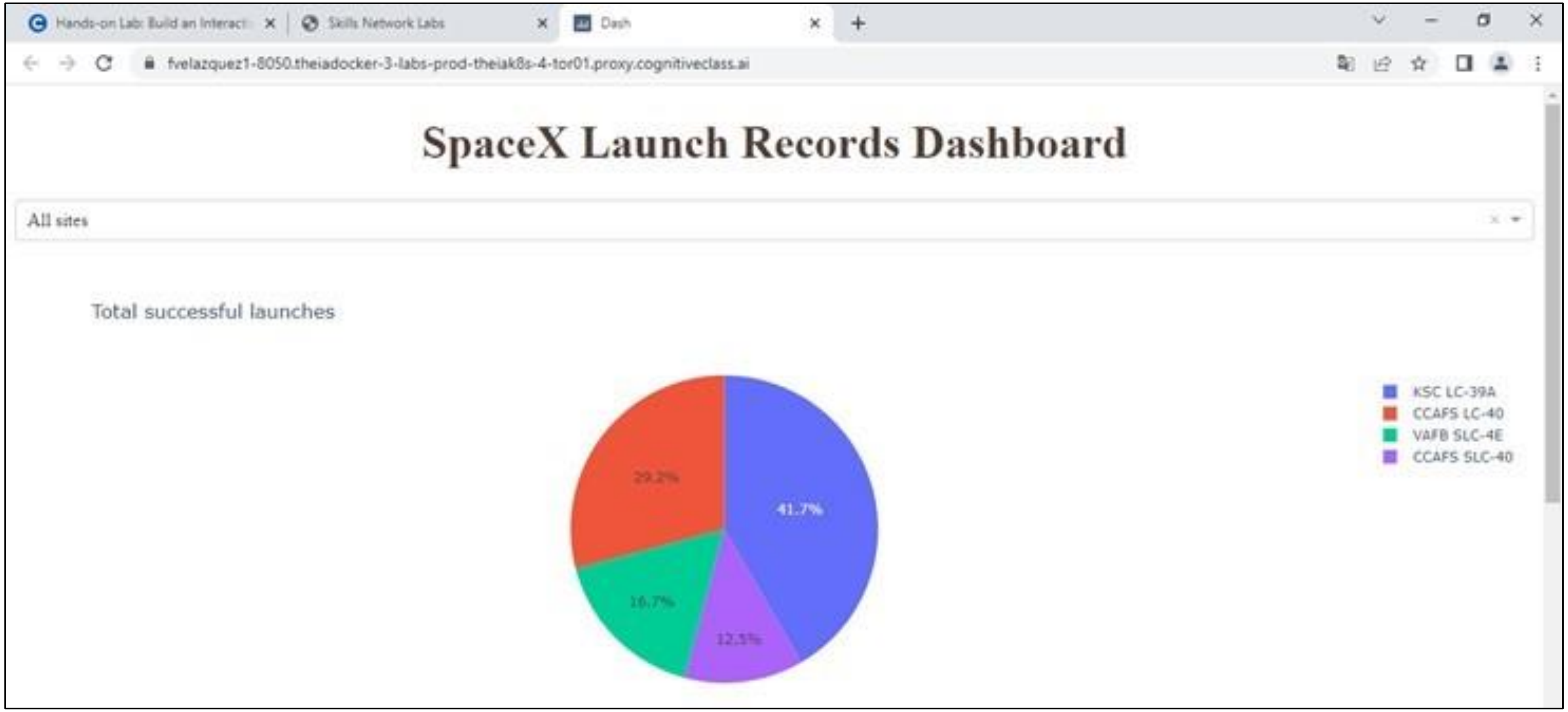
Dataframe used in the Dashboard application: **spacex_df**

## Variables used to create the Dashboard application:

| | | |
|---|---|---|
| Flight Number | Date | Time (UTC) |
| Booster Version | Launch Site | Payload Mass (kg) |
| Orbit | Customer | Landing Outcome |
| **class** | Lat | Long |

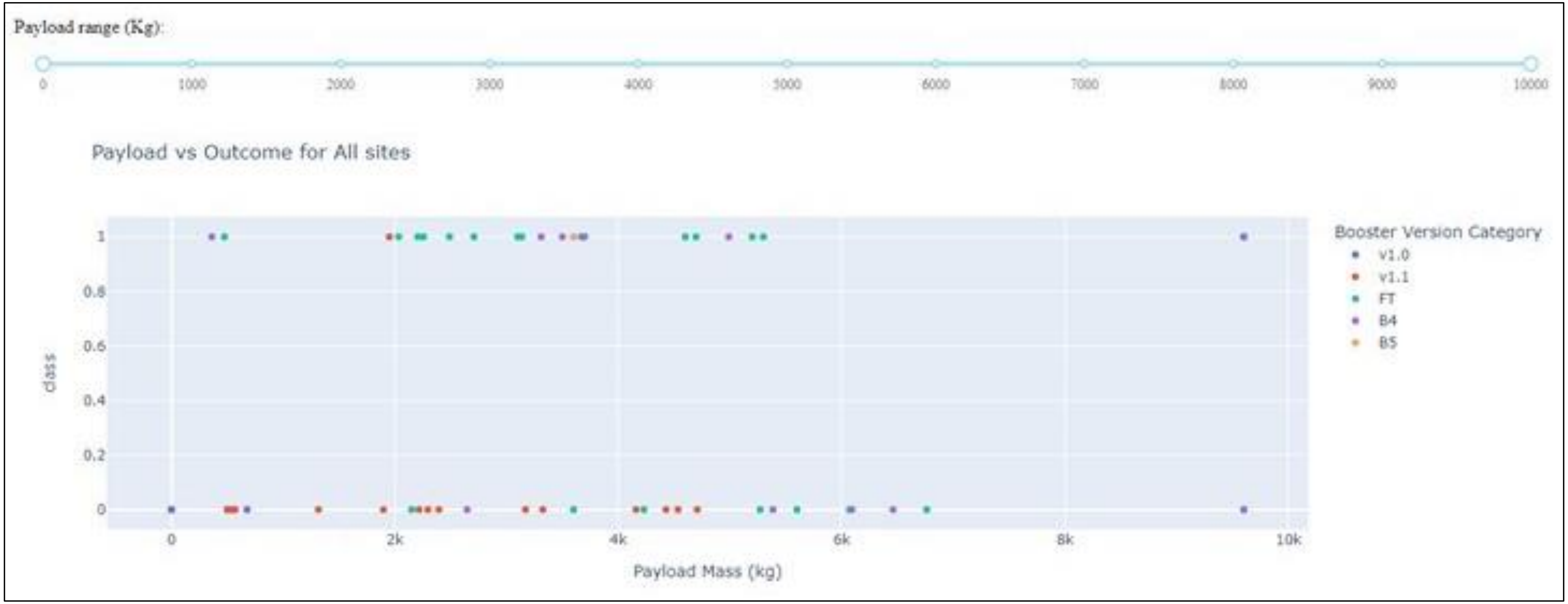
# Interactive Visual Analytics Using Plotly Dash

**KSC LC-39A** has the **highest number of successful** Falcon9 first stage landings among all four launch sites: **41.7%.**



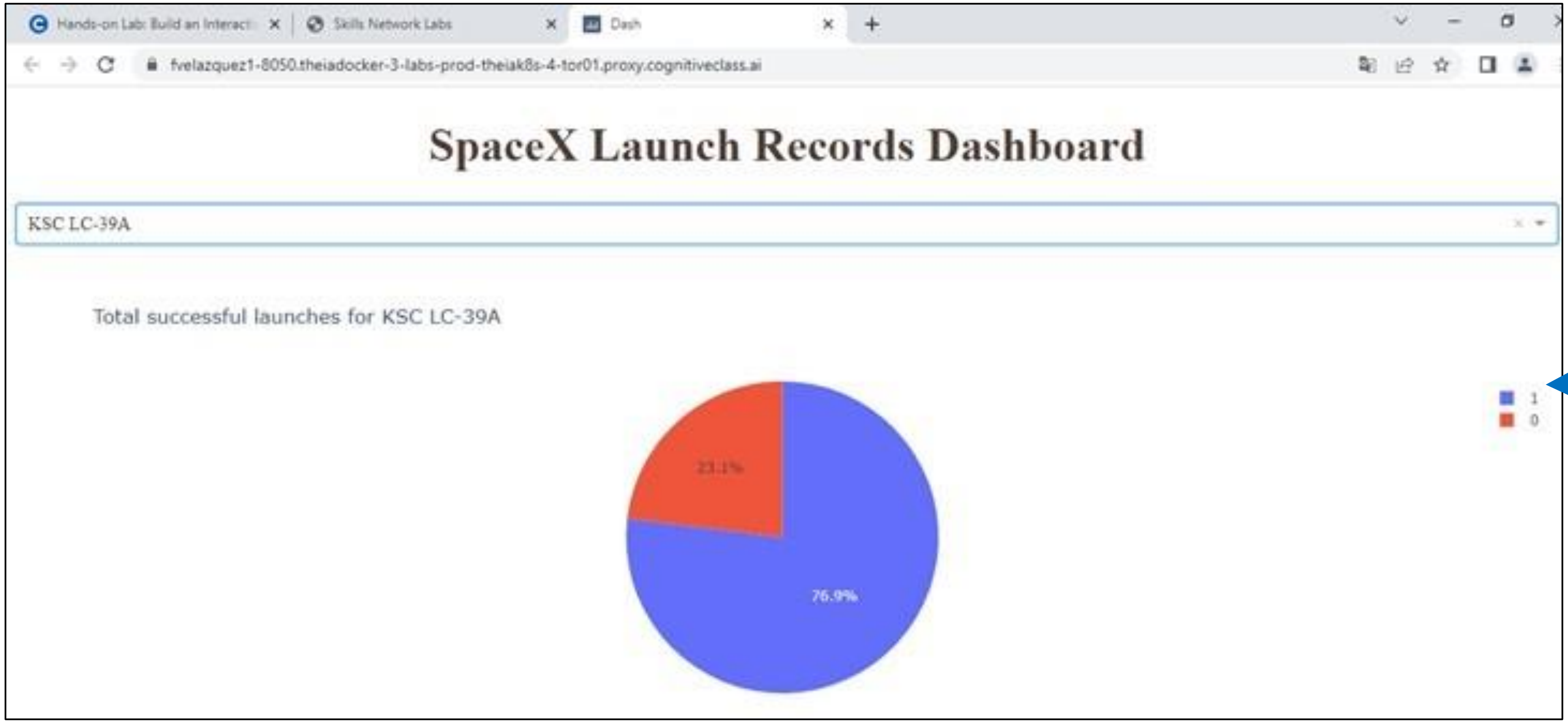
# Interactive Visual Analytics Using Plotly Dash

**FT Booster version** results in the **highest number of successful launches** (Falcon9 first stage successful landing outcome) .



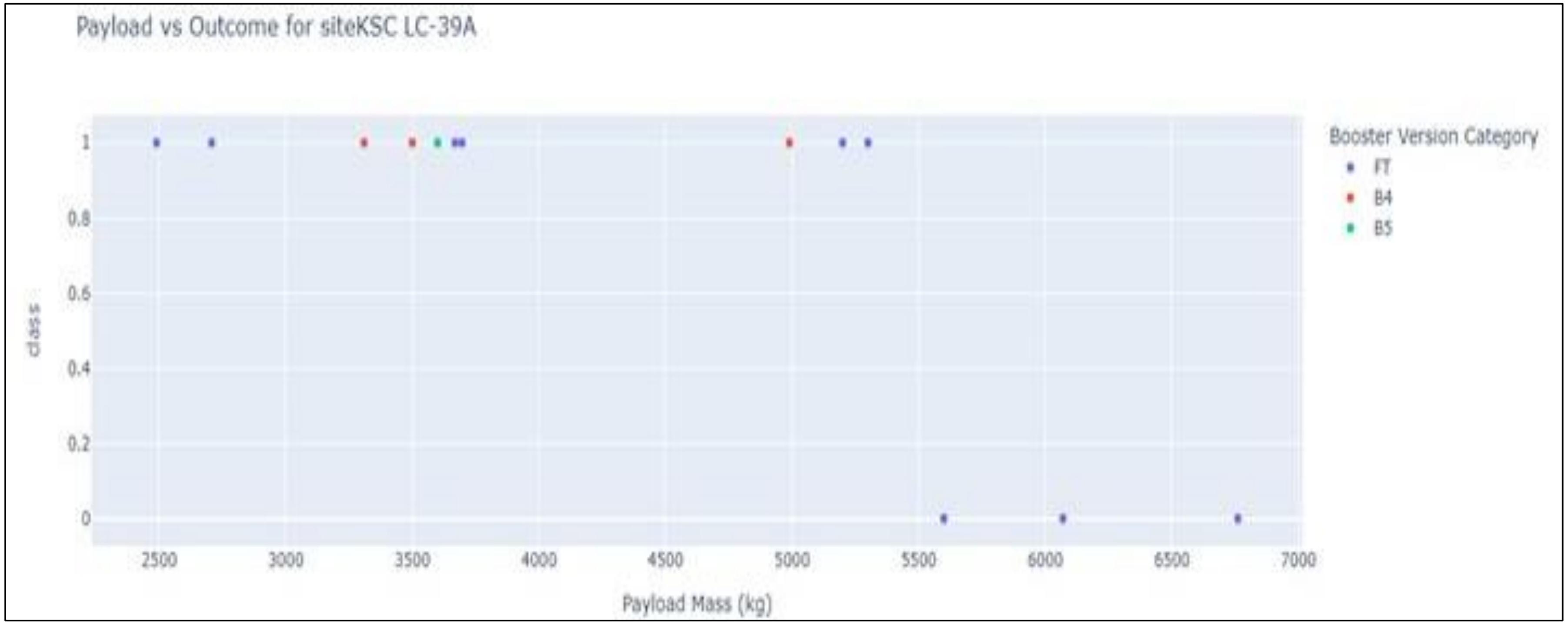
# Interactive Visual Analytics Using Plotly Dash

**KSC LC-39A** launch site: **76.9% rate of successful launches** (Falcon9 first stage successful landing).



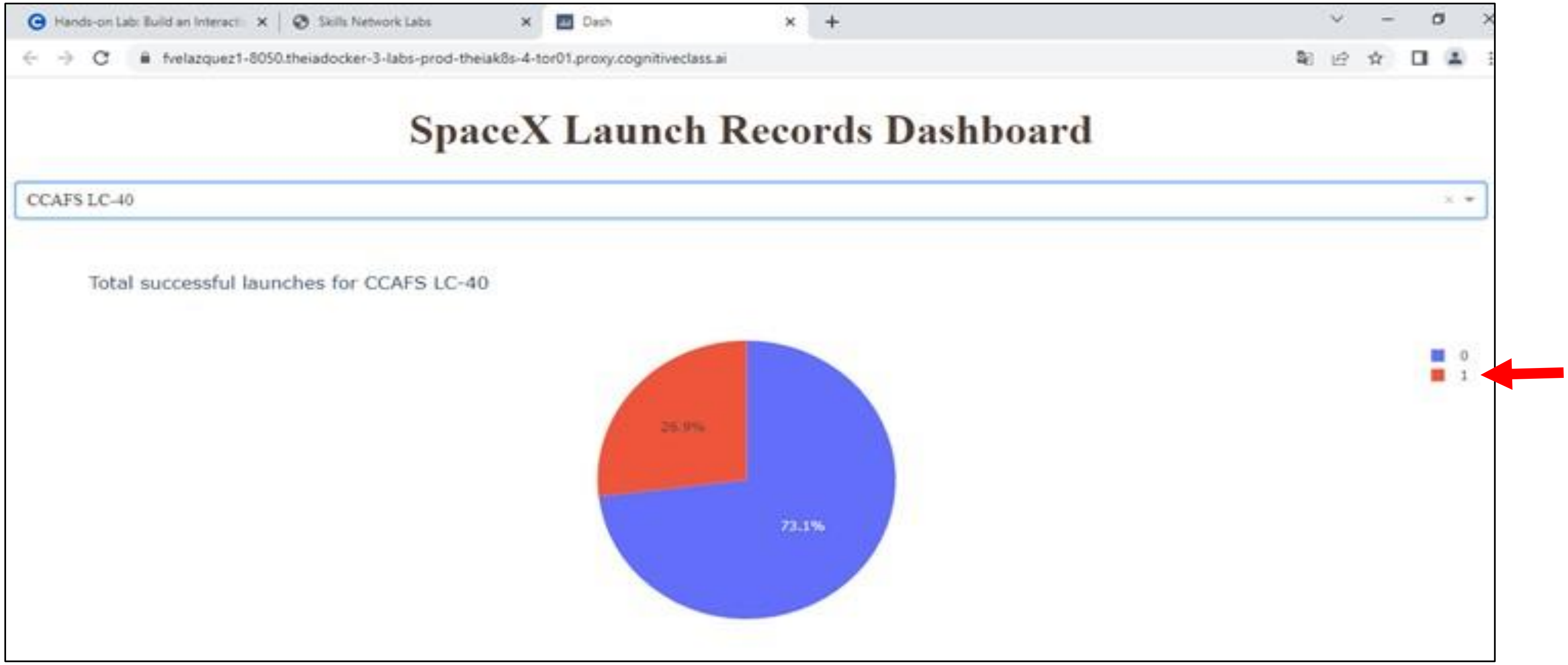
# Interactive Visual Analytics Using Plotly Dash

**KSC LC-39A launch site**: relationship between **Payload Mass** and **launch success/failure** (Falcon9 first stage landing outcome).



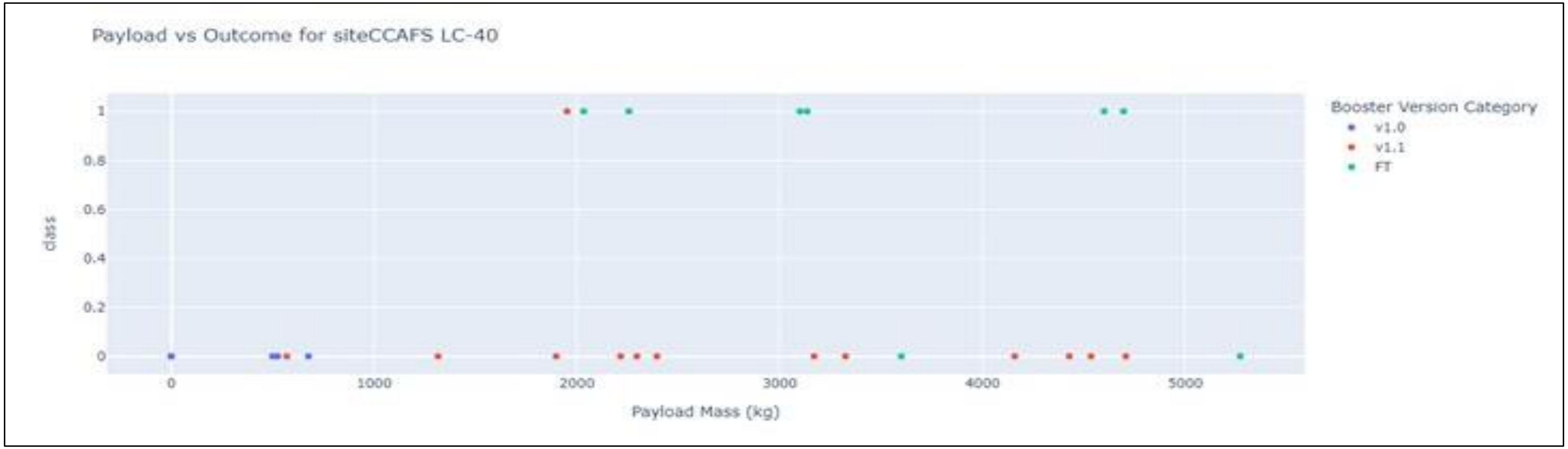
# Interactive Visual Analytics Using Plotly Dash

**CCAFS LC-40** launch site: **26.9%** **rate of successful launches** (Falcon9 first stage successful landing).



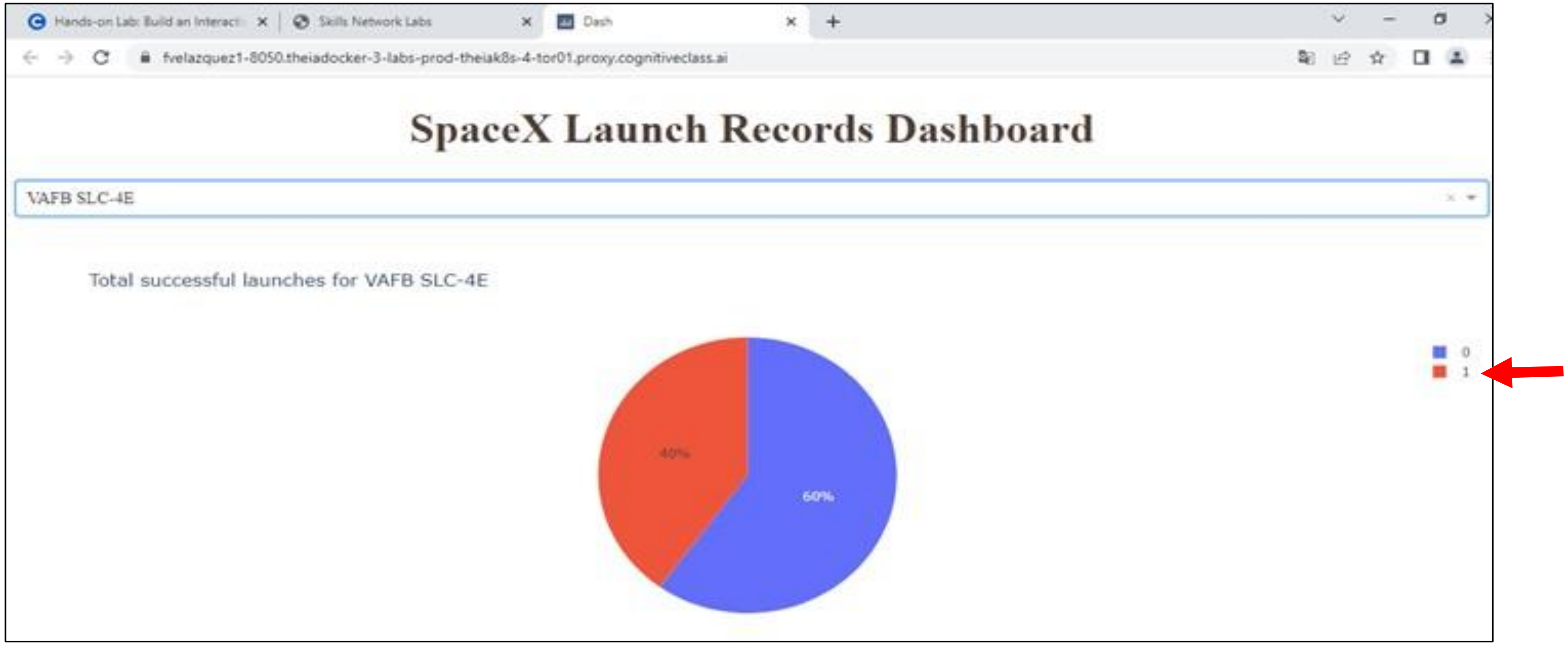
# Interactive Visual Analytics Using Plotly Dash

**CCAFS LC-40 launch site:** relationship between **Payload Mass** and **launch success/failure** (Falcon9 first stage landing outcome).



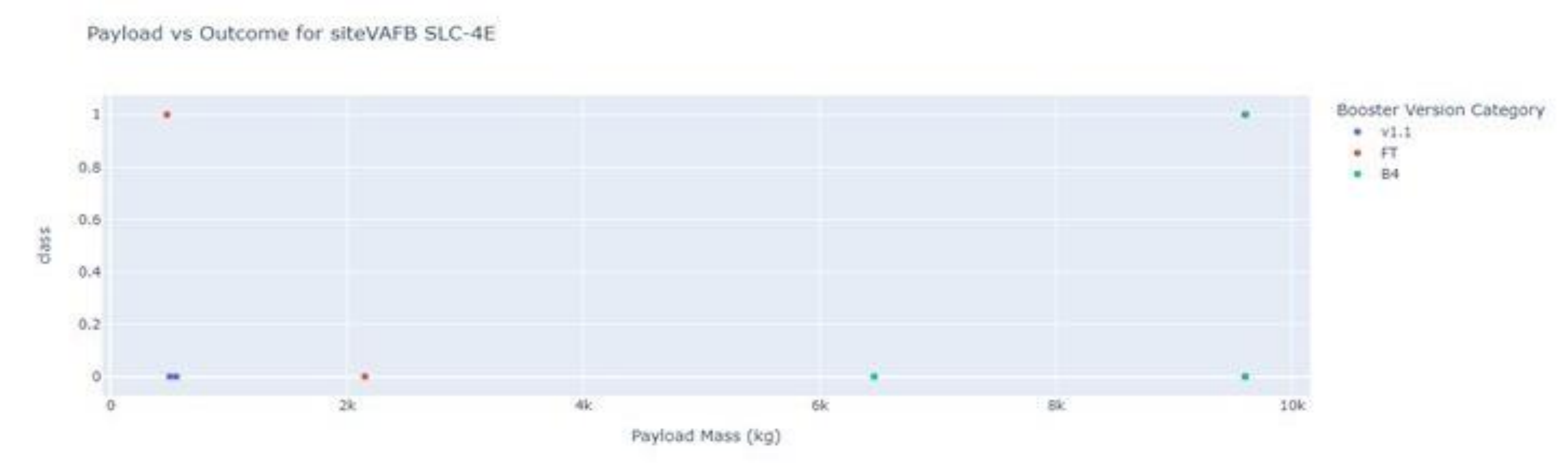
# Interactive Visual Analytics Using Plotly Dash

**VAFB SLC-4E** launch site: **40% rate of successful launches** (Falcon9 first stage successful landing).



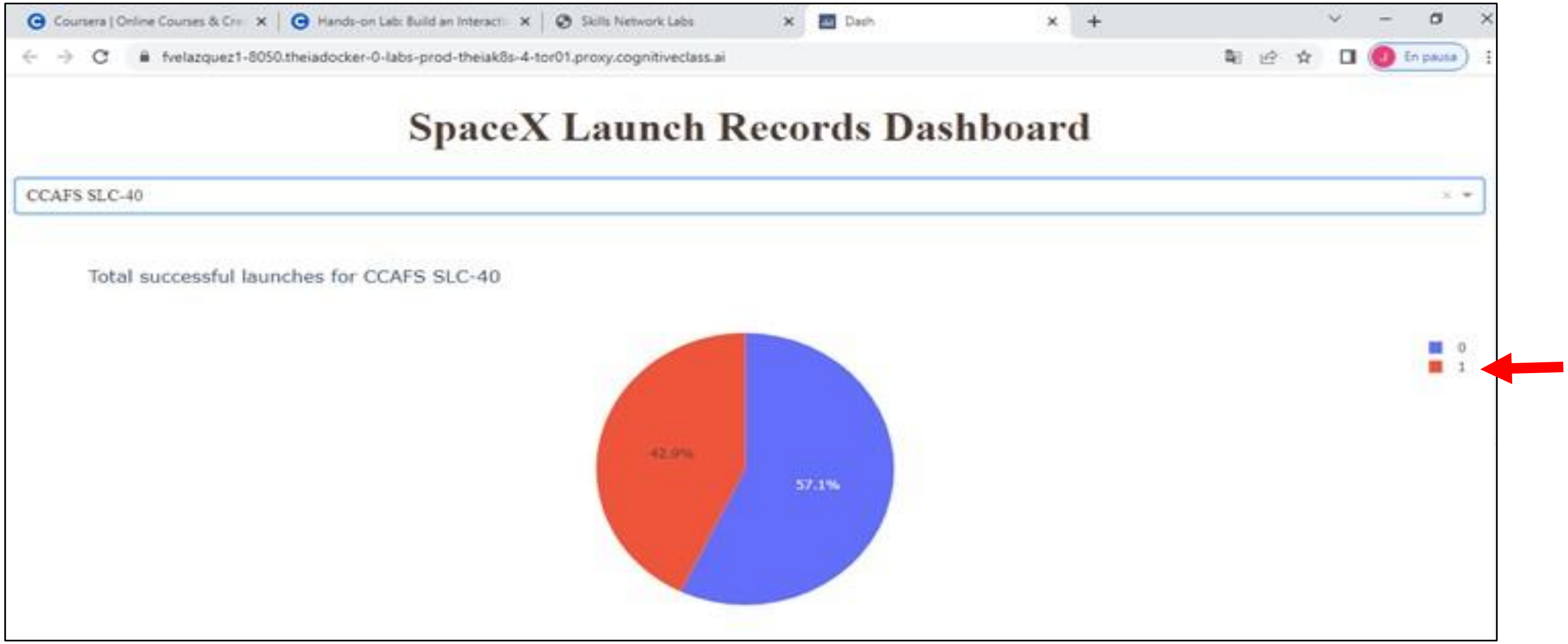
# Interactive Visual Analytics Using Plotly Dash

**VAFB SLC-4E launch site**: relationship between **Payload Mass** and **launch success/failure** (Falcon9 first stage landing outcome).



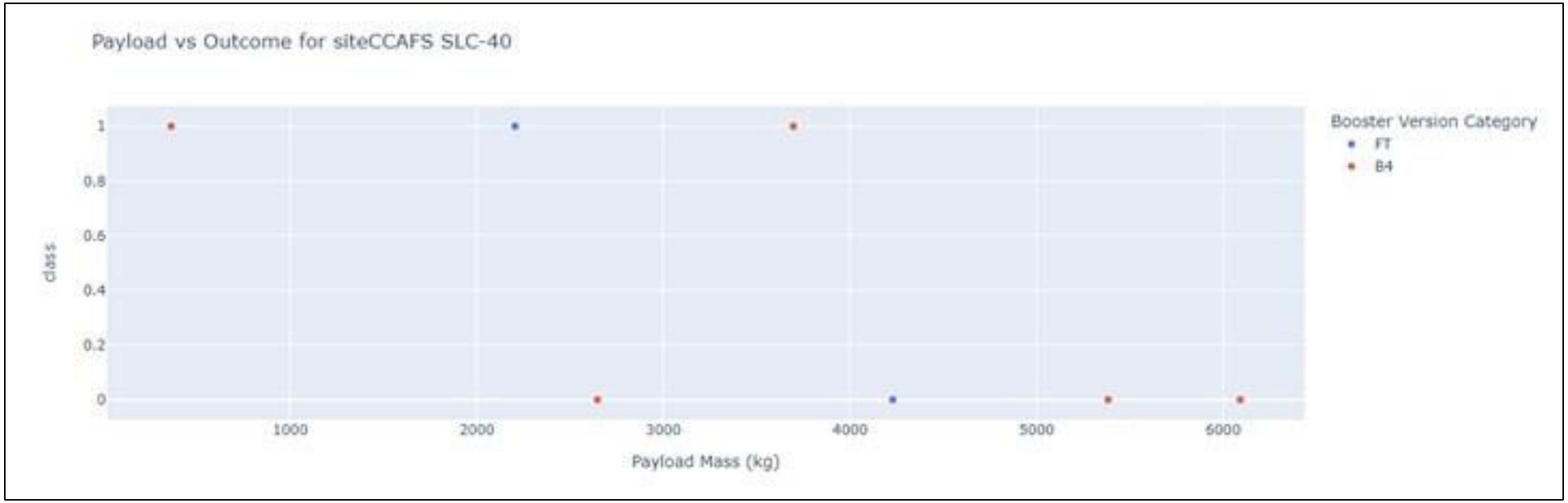
# Interactive Visual Analytics Using Plotly Dash

**CCAFS SLC-40** launch site: **42.9% rate of successful launches** (Falcon9 first stage successful landing).



# Interactive Visual Analytics Using Plotly Dash

**CCAFS SLC-40 launch site**: relationship between **Payload Mass** and **launch success/failure** (Falcon9 first stage landing outcome).



# Interactive Visual Analytics Using Plotly Dash

*** KSC LS-39A has the highest successful launch rate**

| Launch site | Launches | Successful launches | Successful launch rate |
|---|---|---|---|
| **KSC LC-39A** | **26** | **20** | **76.9%** |
| VAFB SLC-4E | 20 | 8 | 40.0% |
| CCAFS LC-40 | 52 | 14 | 26.9% |
| CCAFS SLC-40 | 14 | 6 | 42.9% |

# Predictive Analysis: Space X Falcon 9 First Stage Landing Prediction

**data**:

Dataframe provided by the IBM team.

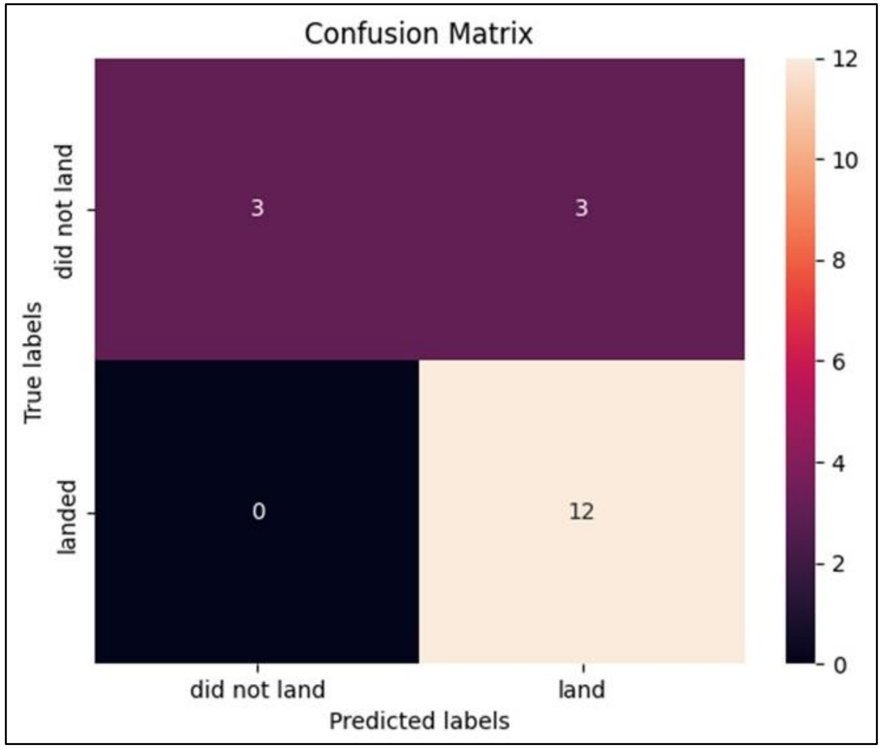
Dataframe used  to test the capacity of different **Machine Learning models** to accurately predict the outcome of each Falcon9 first stage landing outcome.

data.head()

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block | ReusedCount | Serial | Longitude | Latitude | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0003 | -80.577366 | 28.561857 | 0 |
| **1** | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0005 | -80.577366 | 28.561857 | 0 |
| **2** | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B0007 | -80.577366 | 28.561857 | 0 |
| **3** | 4 | 2013-09-29 | Falcon 9 | 500.000000 | PO | VAFB SLC 4E | False Ocean | 1 | False | False | False | NaN | 1.0 | 0 | B1003 | -120.610829 | 34.632093 | 0 |
| **4** | 5 | 2013-12-03 | Falcon 9 | 3170.000000 | GTO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 | 0 | B1004 | -80.577366 | 28.561857 | 0 |

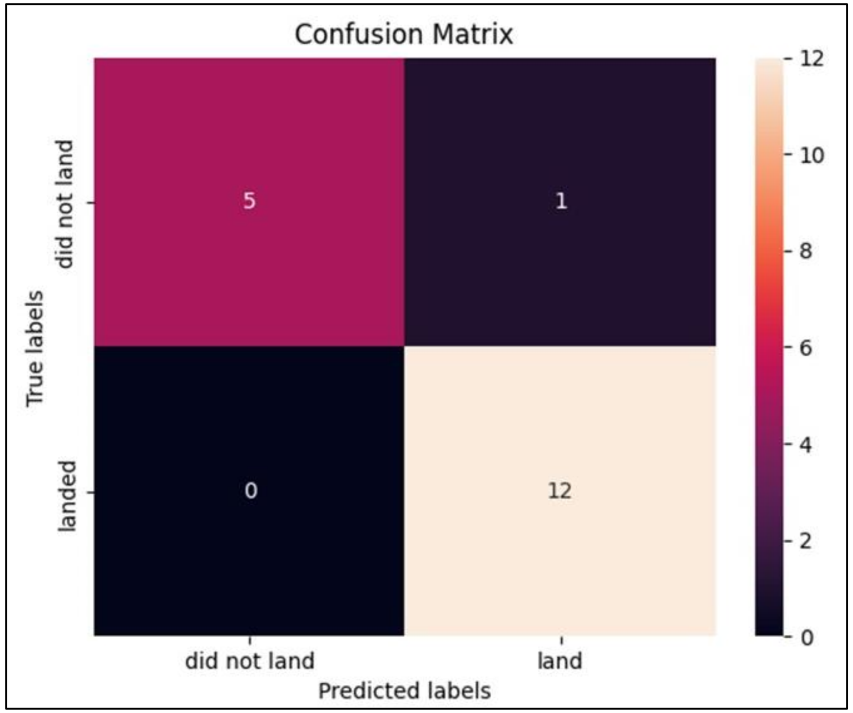
# Predictive Analysis:
# Space X Falcon 9 First Stage Landing Prediction

**Logistic Regression** model
**Accuracy:** 0.8333333333333334



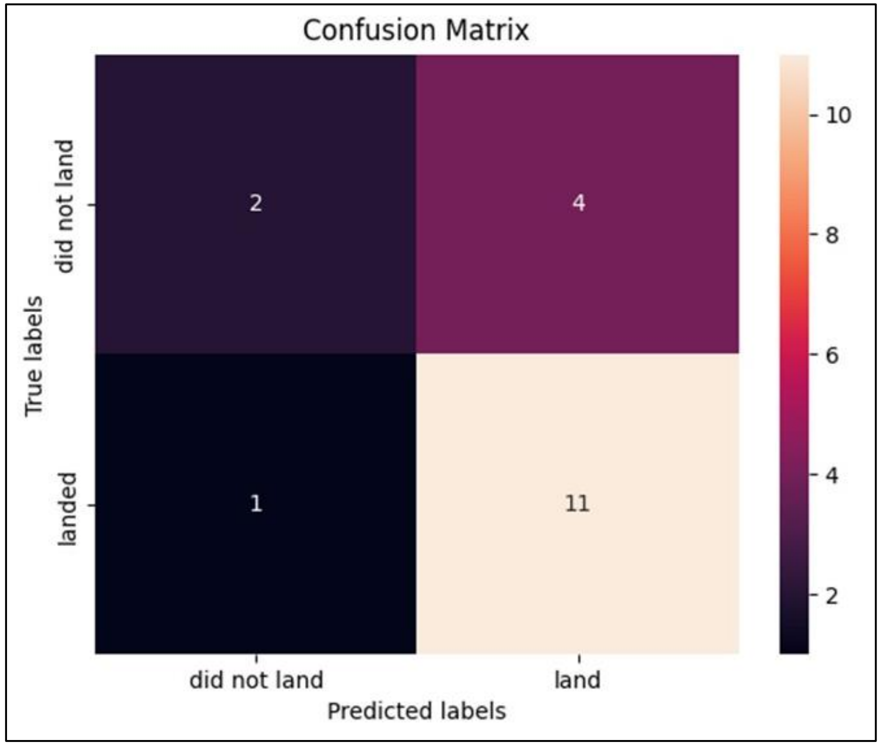
# Predictive Analysis:
# Space X Falcon 9 First Stage Landing Prediction

**Support Vector Machine** model
**Accuracy:** 0.9444444444444444



# Predictive Analysis:
# Space X Falcon 9 First Stage Landing Prediction

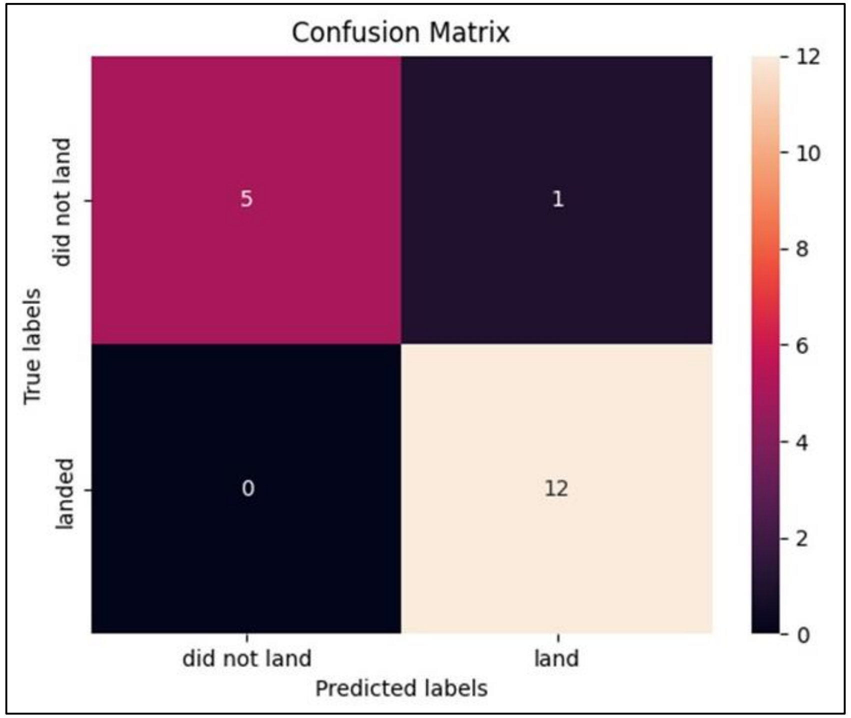
**Decision Tree** model
**Accuracy :** 0.7222222222222222



# Predictive Analysis:
# Space X Falcon 9 First Stage Landing Prediction

**K Nearest Neighbors** model

**Accuracy:** 0.9444444444444444



# Predictive Analysis:
# Space X Falcon 9 First Stage Landing Prediction

## Confusion Matrix: summary

| Machine Learning model | TP | TN | FP | FN | Total | Accuracy |
|---|---|---|---|---|---|---|
| Logistic Regression | 3 | 12 | 3 | 0 | 18 | 0.833333333 |
| **Support Vector Machine** | 5 | 12 | 1 | 0 | 18 | 0.944444444 |
| Decision Tree | 2 | 11 | 4 | 1 | 18 | 0.722222222 |
| **K Nearest Neighbors** | 5 | 12 | 1 | 0 | 18 | 0.944444444 |

**TP**: True positives
**FP**: False positives
**TN**: True negatives
**FN**: False negatives

# DISCUSSION

Data Science Capstone Project Report:

# "Predicting the outcome of SpaceX Falcon9 first stage landing"

## Scenario

- Our company SpaceY wants to compete with SpaceX in the race to make space travel affordable for every one.
- SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars.
- Low cost is due to the fact that SpaceX reuses the first stage of Falcon9 rocket.

## Goal

Our goal is to **accurately predict** SpaceX Falcon9 first stage landing outcome. Knowing the price of each Falcon9 launch will **guide our strategies to bid against SpaceX.**

## Strategy

We used **Data Science** and **Machine Learning tools** to develope **models** that **accurately predict the outcome** (success vs failure) of the Falcon9 first stage landing.

## Data collection

Data about past launches of SpaceX Falcon9 rocket was collected from the **SpaceX site**, **Wikipedia** and also provided by the **Data Science IBM team**.

Cleaned data was used to **identify factors relevant** to the successful landing of the first stage of Falcon9.

## Exploratory Data Analysis using SQL

We collected information about past launches:

- Names of the unique launch sites in the space mission.
- Records about specific launch sites.
- The number of launches in each SpaceX launch sites.
- The number and occurrence of each orbit.
- The number and occurrence of mission outcome per orbit type.

## Exploratory Data Analysis using SQL

- Total payload mass carried by specific boosters.
- Date when the first succesful landing outcome in ground pad .
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Total number of successful and failure mission outcomes.
- Names of the booster_versions which have carried the maximum payload mass.

## Exploratory Data Analysis using SQL

We collected the following information about:

- Records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
- The count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- We created a landing outcome label from Outcome column.

## Exploratory Data Analysis with visualization

We **identified attributes relevant to the Falcon9 first stage landing outcome**.

- We visualized the relationship between:

  Flight Number and Launch Site
  Payload and Launch Site
  Success rate and orbit type
  FlightNumber and Orbit type
  Payload and Orbit type
  Launch success and year trend

- We created dummy variables to categorical columns.
- We cast all numeric columns to float64.

## Interactive Visual Analytics with Folium map

We **identified some of the characteristics of proximities relevant for finding an optimal location to build a launch site**.

We used **Folium** to create an interactive map where:

We marked all launch sites.

We marked the success/failed launches for each site.

We calculated the distances between a launch site to its proximities.

# Build a Dashboard Application with Plotly Dash

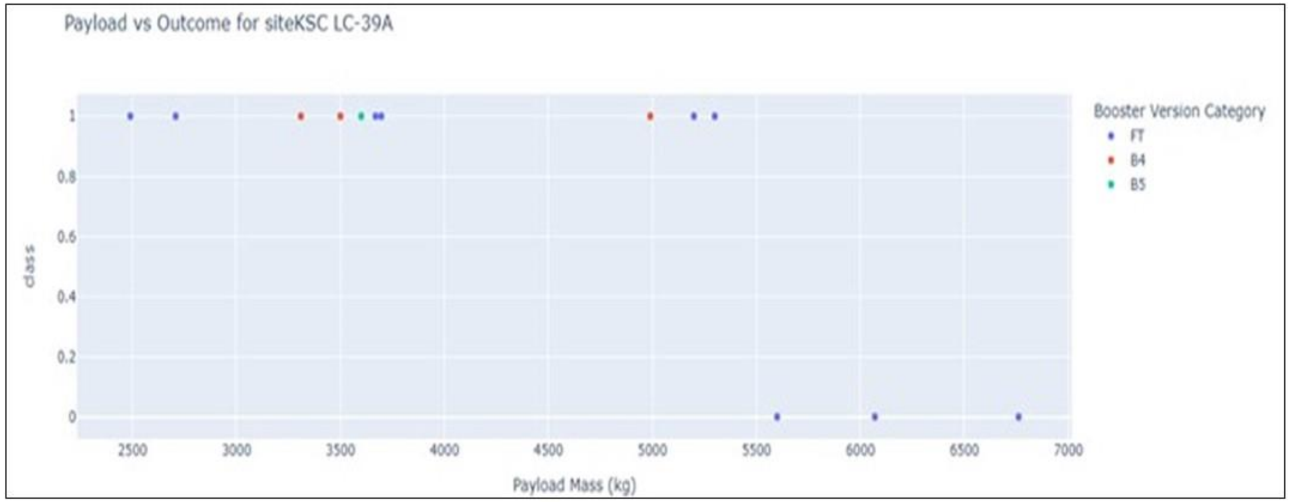
We successfully built a dashboard for users to perform **interactive visual analytics on SpaceX launch data in real-time**.

**Example**

**KSC LC-39A launch site**:

1.- The **FT Booster version** has the **highest number of successful launches** when **Payload Mass** is **below 5,500kg**.

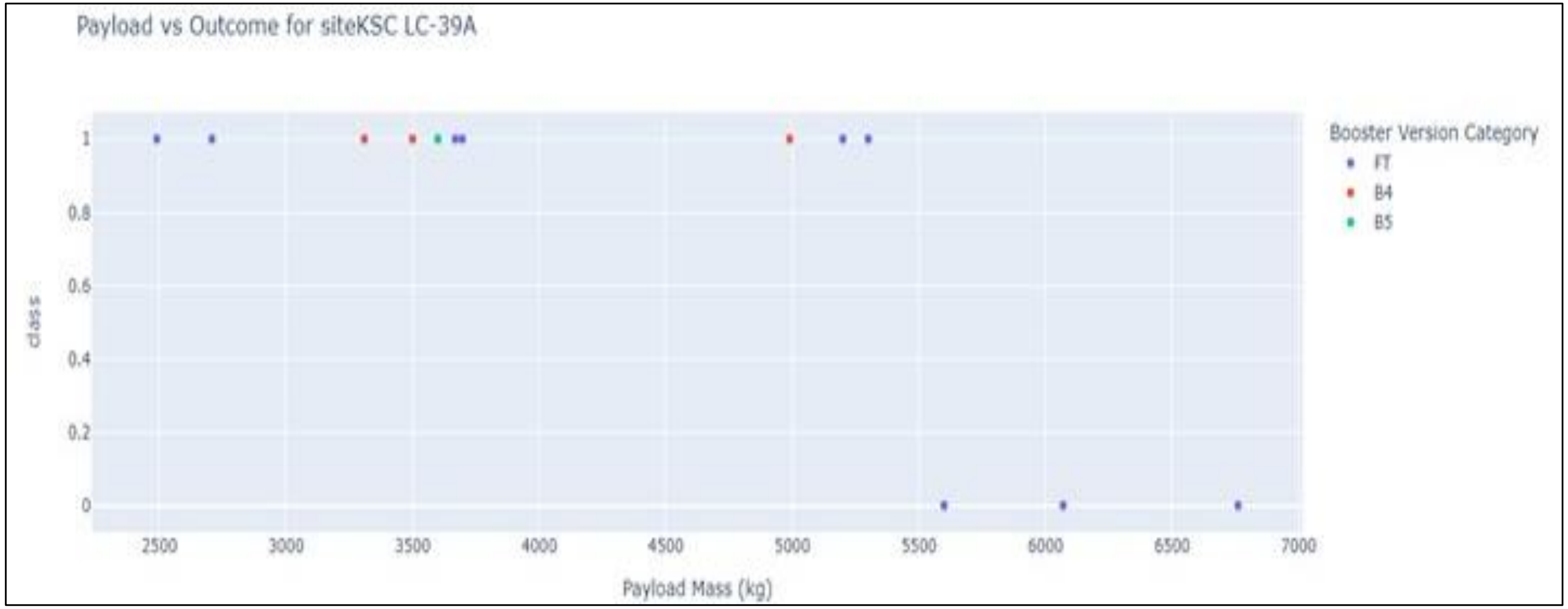
2.- **All launches** with **FT, B4** and **B5** are **successful** when **PayloadMass is below 5,5000 kg.**



## KSC LC-39A launch site:

1.- The FT Booster version has the **highest number of successful launches** when **Payload Mass** is **below 5,500kg**.

2.- **All launches** with FT, B4 and B5 are **successful** when **PayloadMass is below 5,5000 kg.**



## Predictive Analysis

We successfully **built a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully**.

**Preprocessing** permitted us to standardize the data, and **train_test_split** permitted us to split the data into training and testing data.

We **trained the model** and **performed Grid Search** to find the **hyperparameters** that allow a given algorithm to perform best.

With the **best hyperparameter values** we determined the **model** with the **best accuracy** using the training data.

We used different **machine learning supervised learning techniques** to **analyze data and predict the outcome of each Falcon 9 launch**:

Logistic Regression

Support Vector Machine

K nearest neighbors

Decision Tree

## Briefly:

**Logistic regression** is the right algorithm to start with classification algorithms; it has the **advantage** to be an easy, fast and simple classification method.

**Support Vector machine** is a type of Machine Learning technique that can be used for both classification and regression.

**Decision tree** is a tree based algorithm used to solve regression and classification problems; it can provide understandable explanation over the prediction.

**K-nearest neighbors** is a non-parametric method used for classification and regression; in **KNN**, we look for k neighbors and come up with the prediction.

Finally,

we output the **confusion matrix** and determined which **model best predicts the outcome** of each Falcon9 first stage landing.

# What is a Machine Learning model?

It is **a file that has been trained to recognize certain types of patterns**.

You **train a model** over a set of data, providing it an **algorithm that it can use to reason over and learn from those data**.

Once you have trained the model, you can use it to **reason over data that it hasn't seen before**, and **make predictions about those data**.

# Machine learning models at a glance

**Logistic Regression**: easy, fast and simple classification method.

**Support Vector Machine**: can be used for both classification and regression.

**Decision Trees**: provides understandable explanation over the prediction.

**K nearest neighbors**: non-parametric method used for classification and regression.

# What is a **Confusion Matrix**?

It is a **performance measurement for machine learning classification**.

Better the effectiveness, better the performance.

It is a table with 4 different combinations of predicted and actual values.

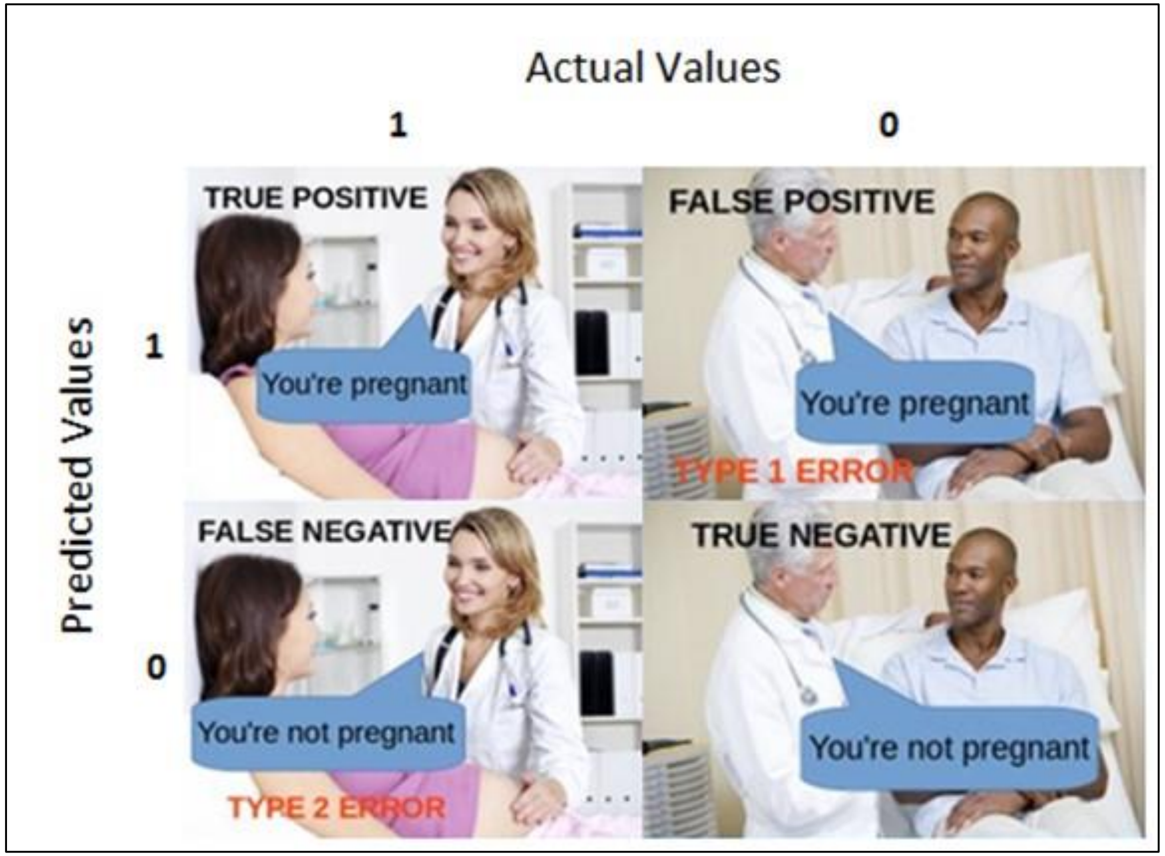
| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

**Sarang Narkhede**

https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

# Understanding **Confusion Matrix**

| Predicted Values | Actual Values: Positive (1) | Actual Values: Negative (0) |
|---|---|---|
| Positive (1) | TP | FP |
| Negative (0) | FN | TN |




**Sarang Narkhede**
https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

# Understanding **Confusion Matrix**

**True Positive**:

You predicted positive and it’s true.

You predicted that a woman is pregnant and she actually is.

**True Negative**:

You predicted negative and it’s true.

You predicted that a man is not pregnant and he actually is not.

**False Positive**: (Type 1 Error)

You predicted positive and it’s false.

You predicted that a man is pregnant but he actually is not.

**False Negative**: (Type 2 Error)

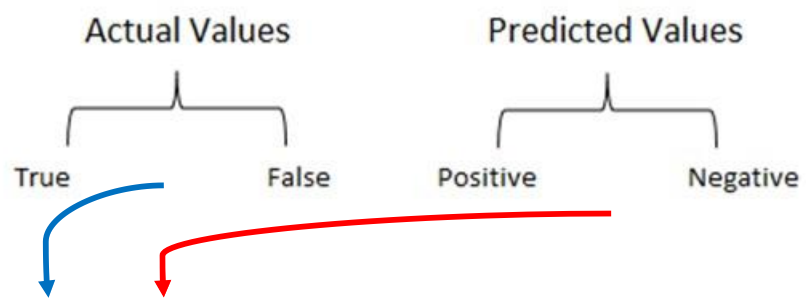
You predicted negative and it’s false.

You predicted that a woman is not pregnant but she actually is.

**Sarang Narkhede**

https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

# How to **Calculate a Confusion Matrix** for a 2-class classification problem?




| | y | y pred | output for threshold 0.6 | Recall | Precision | Accuracy |
|---|---|---|---|---|---|---|
| | 0 | 0.5 | 0 | | | |
| TP | 1 | 0.9 | 1 | | | |
| | 0 | 0.7 | 1 | | | |
| TP | 1 | 0.7 | 1 | 1/2 | 2/3 | 4/7 |
| TN | 1 | 0.3 | 0 | | | |
| | 0 | 0.4 | 0 | | | |
| TN | 1 | 0.5 | 0 | | | |

$$\textbf{Accuracy} = \frac{\text{Total correct predictions}}{\text{Total predictions made}} * 100$$

$$\textbf{Accuracy} = \frac{\textbf{2 TP + 2 TN}}{\textbf{7}\ \text{Total}} * 100$$

**Sarang Narkhede**

https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

# CONCLUSION

Discussion 20

# Predictive Analysis: Space X Falcon 9 First Stage Landing Prediction

## Summary of the confusion matrices

| Machine Learning model | TP | TN | FP | FN | Total | Accuracy |
|---|---|---|---|---|---|---|
| Logistic Regression | 3 | 12 | 3 | 0 | 18 | 0.833333333 |
| **Support Vector Machine** | 5 | 12 | 1 | 0 | 18 | 0.944444444 |
| Decision Tree | 2 | 11 | 4 | 1 | 18 | 0.722222222 |
| **K Nearest Neighbors** | 5 | 12 | 1 | 0 | 18 | 0.944444444 |

**TP**: True positives
**FP**: False positives

**TN**: True negatives
**FN**: False negatives

# Predictive Analysis: Space X Falcon 9 First Stage Landing Prediction

**Machine Learning models that performs best:**

Support Vector Machine and K Nearest Neighbor

Accuracy: **0.9444444444**

**Highest number of correct predictions**: True Positives (**5**) and True Negatives (**12**)

**Lowest number of wrong predictions**: False Positives (**1**)

**Do not detect False Negatives predictions (0)**

# APPENDIX

Methodology in
more detail

Appendix Methodology 1

# Data collection I

We followed the instructions given in

(Optional)Hands-on Lab: Complete the Data Collection API Lab (Capstone course Week 1)

**SpaceX Falcon 9 first stage Landing Prediction**

**Lab 1: Collecting the data**

Information from past launch data was obtained from

**Open Source SpaceX REST API**

(https://api.spacexdata.com/v4/launches/past)

**Appendix Methodology 2**

## Data collection I continued

Dataframe was filtered to only include Falcon 9 launches.
The LandingPad column retains None values.
Other missing np.nan values in the PayloaMass column were replaced with the mean of this column.

**Helper functions used to to call the API and append the data to the lists:**
From **rocket** column extract booster name
From **launchpad** column extract name of launch site, longitude, latitude
From **payload** column extract mass and orbit
From **cores** column extract outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core, the number of times this specific core has been reused, and the serial of the core.

Requested JSON results more consistent, we used the a **static response object** for this project.

We used **json_normalize** method to convert the json result into a dataframe named **data**

**Appendix Methodology 3**

## Data collection I continued

We kept only a subset of the features
We removed rows with multiple cores
We converted the date_utc to a datetime datatype and then extracted the date leaving the time
Using the date we restricted the dates of the launches
The data from these requests were stored in lists and used to create a new pandas dataframe named **df**

We removed the Falcon 1 launches keeping only the Falcon 9 launches.
We filtered the data dataframe using the BoosterVersion column to only keep the Falcon 9 launches.
We reseted the FlightNumber column
We saved the data to a new dataframe named **data_falcon9**.

**Appendix Methodology 4**

# Data collection I continued

**Data Wrangling**

Some of the rows are missing values in our dataset.

The **LandingPad** column will retain None values to represent when landing pads were not used.

We calculated the mean for the **PayloadMass**, and replaced `np.nan` values in the data with the mean.

The result is the final version of the dataframe **data_falcon9**

**Appendix Methodology 5**

# Data collection II

We followed the instructions given in
(Optional)Hands-on Lab: Complete the Data Collection with Web Scraping lab
(Capstone course Week 1)

**Space X Falcon 9 First Stage Landing Prediction**

**a) Web scraping Falcon 9** and **Falcon Heavy Launches Records** from **Wikipedia** with **BeautifulSoup** package.

**b) Parse the table** and convert it into a **Pandas data frame**

We performed **web scraping** on a Wiki page to collect a **list of Falcon 9 and Falcon Heavy launches Wikipage updated on 9th June 2021**

static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data frame **data_falcon9** was created by parsing the launch HTML tables.

**Appendix Methodology 6**

# Data collection II continued

- **Create a landing outcome label from Outcome column**
- Using the **Outcome**, we created a list where the element is **0** if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's **1**.
- Then we assigned it to the variable **landing_class**.
- This variable represents the **classification variable** that **indicates the outcome of each launch**.
- If the value is **0**, the first stage **did not land successfully**; **1** means the first stage **landed successfully**.

**Appendix Methodology 7**

For detailed description of the work done in **Exploratory Data Analysis with SQL** please view the following notebook:

**Complete the Exploratory Data Analysis with SQL Lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

For detailed description of the work done in **Exploratory Data Analysis with Visualization** and **Exploratory Data Analysis: Feature engineering** please view the following notebook:

**Exploratory Data Analysis with Visualization Lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

*When uploaded to GitHub some figures may not appear in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Appendix Methodology 8**

## Exploratory Data Analysis with SQL

- The goal is to identify those attributes that can be used to determine if the first stage can be reused.
- We created the **SPACEXDATASET** in **DB2** database.
- Ensured the Date Format isDD-MM-YYYY and timestamp isDD-MM-YYYY HH:MM:SS
- Changed the PAYLOAD_MASS__KG_ datatype to INTEGER.
- We wrote an executed **SQL queries** to gain information described in the first part of the  Exploratory Data Analysis within the Results section.

## Exploratory Data Analysis with Visualization

- The **goal** is to identify those attributes that can be used to determine if the first stage can be reused.
- We used the **SpaceX** dataset provided by the IBM team an generated the corresponding dtaframe.
- Through **scatter point charts** we analyzed the effect of several attributes on the launch outcome.

Appendix Methodology 9

# Exploratory Data Analysis: Feature engineering

- The **goal** is create a clean dataframe containing those attributes that can be used to determine if the first stage can be reused.
- We created the **features** dataframe with variables that affect the success rate.
- We applied the **get_dummies()** function on the categorical columns of **features** dataframe.
- We cast all numeric columns to float64.

**Appendix Methodology 10**

# Interactive Visual Analytics:

## Launch Sites Locations Analysis with Folium

- The **goal** is to create, using Folium, an interactive map that facilitates the identification of relevant factors involved in finding an optimal location for building a launch site.

For a detailed description please view the following notebook:

**Interactive Visual Analytics with Folium Lab**

https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite%20%281%29.ipynb

## Build a Dashboard Application with Plotly Dash

- The **goal** is to build a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- The dataset, provided by the IBM team, is the csv document "spacex_launch_dash.csv"

For a detailed description please view the following notebook:

**Build an Interactive Dashboard with Ploty Dash Lab**

https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/spacex_dash.py

*When uploaded to GitHub some figures may not appear in the notebook. Therefore, I uploaded the github link onto **nbviewer** to have all figures seen when accessing the link.

**Appendix Methodology 11**

For detailed description of the work done in **Predictive Analysis** please view the following notebook:

**Machine Learning Prediction lab**
https://nbviewer.org/github/VVJF/Coursera-IBM-Capstone-Project-2022/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

*When uploaded to GitHub some figures may not appear in the notebook. Therefore, I uploaded the githublink onto **nbviewer** to have all figures seen when accessing the link.

**Acknowledgments**

To the **IBM team** for creating this excellent Data Science Professional Certificate.

To the learners that participated in the Discussion Forums, their questions and suggestions were very helpful.

To the IBM staff for their support in the Discussion Forums.

To the “coding community” that made my life a lot, lot, lot happier!