IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<Name>
<Date>

# Outline

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

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

- Project background and context
- Problems you want to find answers

Section 1

# Methodology

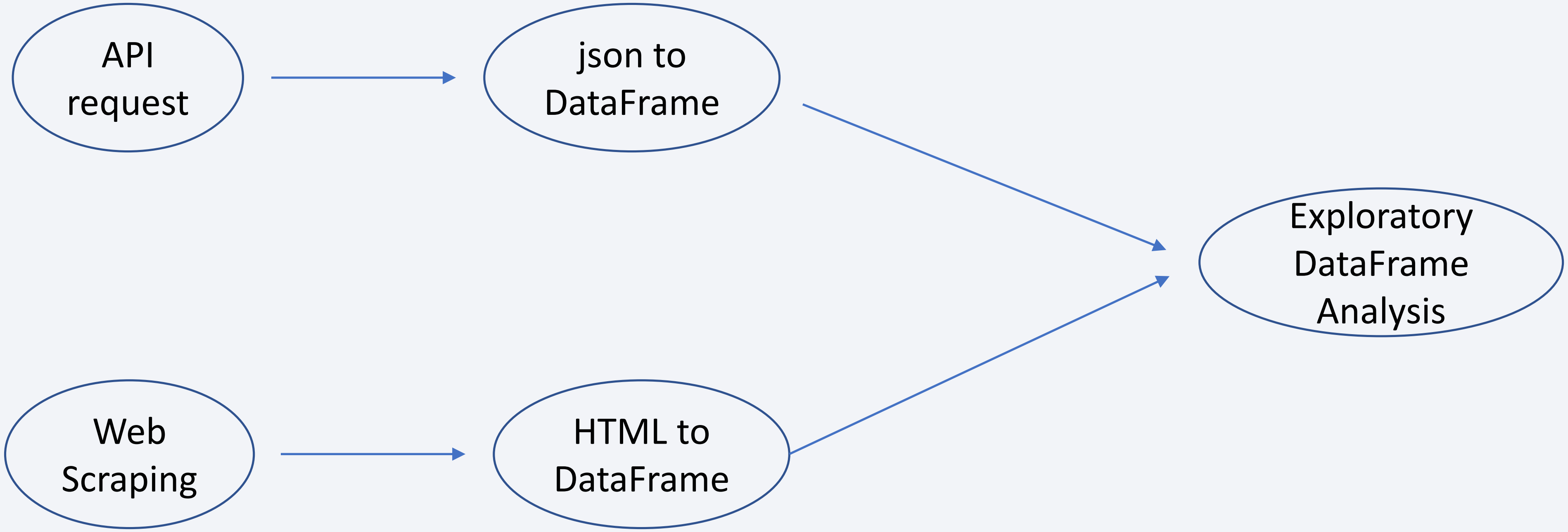
# Methodology

## Executive Summary – Jupyter Notebook

- Data collection methodology:
  - Data was Extracted from SpaceX API via Requets library and Postman. Postman is an API manager useful to visualize the json file.
- Perform data wrangling
  - Data was wrangled in Jupyter with Pandas. The preliminar version of SpaceX DataFrame was transformed to get the success rate of rocket landing.
- 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

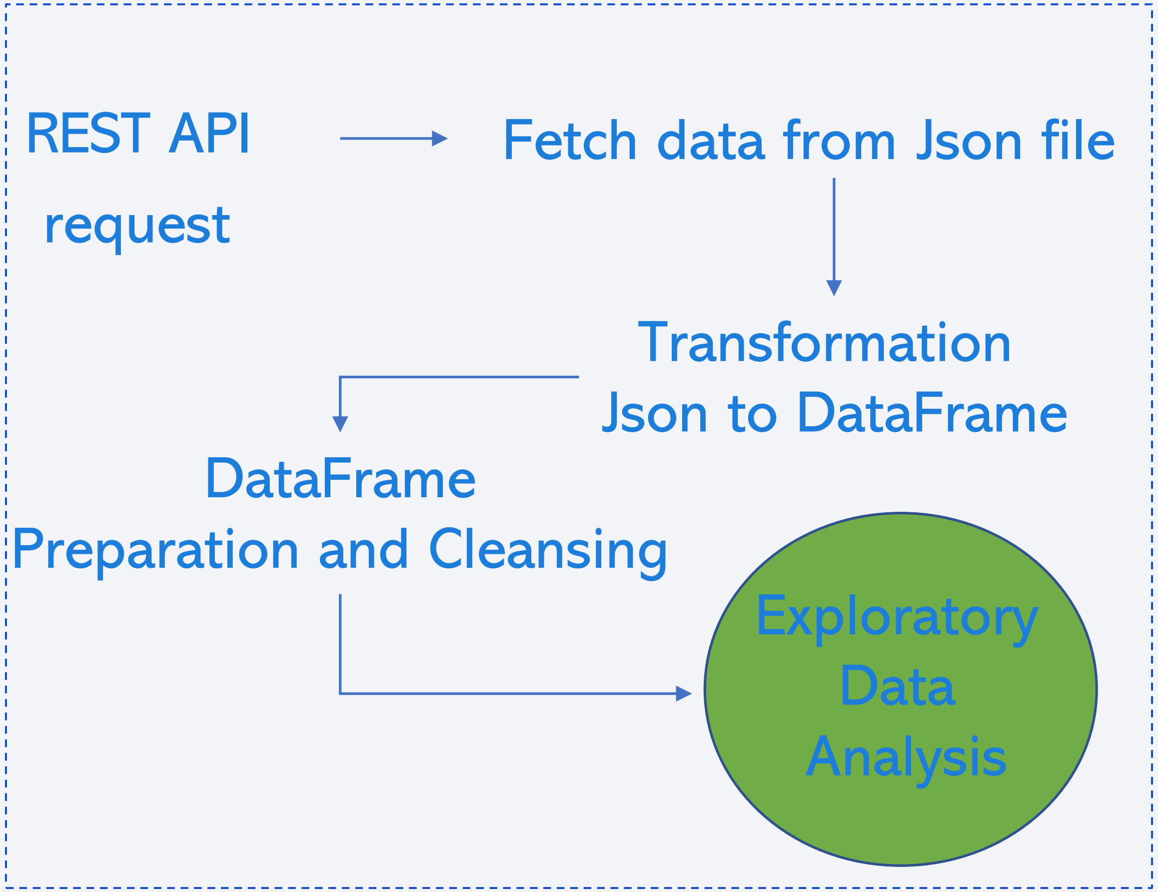
# Data Collection – SpaceX API & Web Scraping

- Data was collected using two methods: via API request and Web Scraping.



# Data Collection – SpaceX API

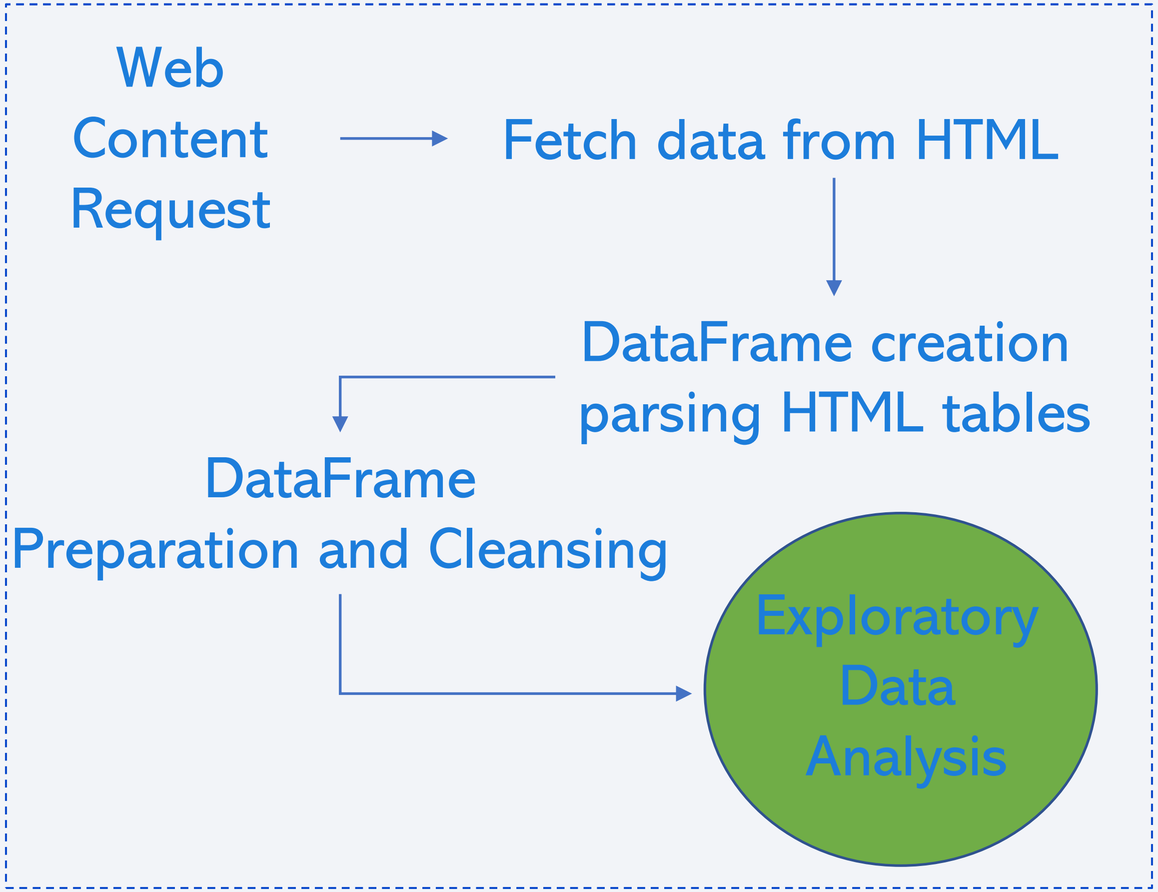
- On the right you see the schema for Data Collection thought SpaceX API
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



# Data Collection - Scraping

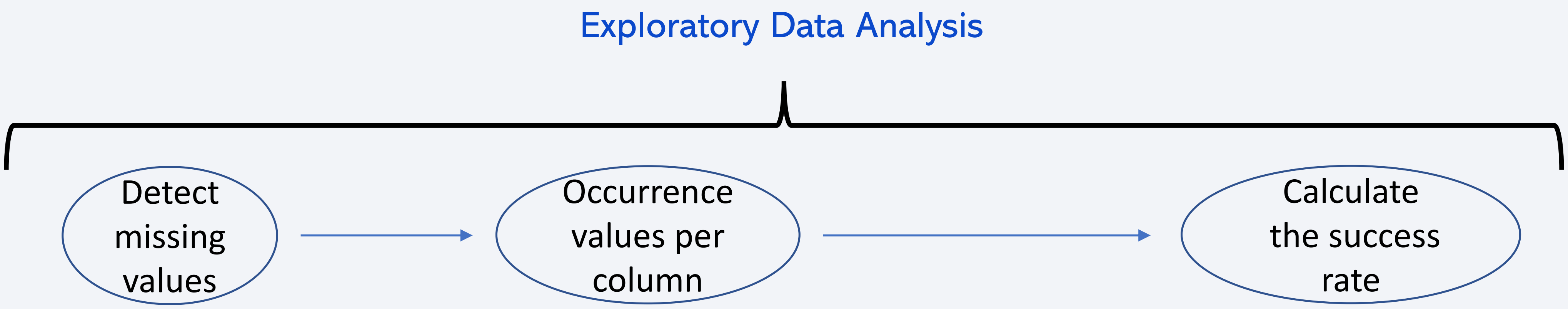
- On the right you see the schema for Data Collection thought Web Scraping

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



# Data Wrangling – Exploratory Data Analysis

- Let's do some Exploratory Data Analysis. First, check the percent of missing values. If possible, replace them to media/median/mode. Secondly, we want to see unique values per column and its occurrence. Thirdly, let's calculate the successful rocket land ratio!
- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone



# EDA with Data Visualization.

- I compared different features of the DataFrame using scatter plots and bars to identify correlations: FlightNumber (FN) Vs. PayLoadMass (PLM) (VAFB SLC 4E has a success rate of 77%), FN Vs. LaunchSite (LN) (in general, the success rate is increasing), PLM Vs. LN (VAFB-SLC launchsite there are no rockets launched for heavypayload mass), ES-L1, GEO, HEO, SSO have the best success rate. (bar chart, Orbit Vs. Success ratio), FN Vs. Orbit (LEO orbit the Success appears related to the number of flights, PLM Vs. Orbit (positive landing rate are more for Polar, LEO and ISS), and the success ratio is increasing with the time (>80% by 2020). Careful! Everything is based on correlations! Bad EDA! Further analysis is required!
- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone

# EDA with SQL

- SQL queries performed:

- %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
- %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)’
- %sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)’
- %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)’
- %sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%' OR MISSION_OUTCOME LIKE 'Failure%’
- %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- %sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, LANDING__OUTCOME AS LANDING__OUTCOME, BOOSTER_VERSION AS BOOSTER_VERSION, LAUNCH_SITE AS LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'

- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone

# Build an Interactive Map with Folium

- Map objects created (folium.): map, circle, map.marker, add_child, MarkerCluster, MousePosition, PolyLine
- These objects are necessary to spot places on folium maps, to make cluster of those places, and to measure the distance between hot spots.
- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone

# Build a Dashboard with Plotly Dash

- Written in HTML, the Dashboard displays a pie chart with the success ratio per launch locations and an interactive scatter plot with the booster used in each launch location and its weight.
- The pie chart was design to display the success ratio after launch selection (drop menu). Also, the scatter plot is updated accordingly to the selection displaying those boosters launched in that specific Air Base.
- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone

# Predictive Analysis (Classification)

- After standard scaling, extracting target variable and splitting the data, logistic regression, Supported Vector Machine, Decision tree, and K-NN algorithm were deployed to compared their performance.
- StandardScaler(), train_test_split(X, y, test_size=0.2, random_state=2), GridSearchCV(), gscv.fit(X_train,y_train), <model>_cv.best_params_, <model>_cv.best_score_,<model>_cv.score(X_test,y_test), yhat=<model>_cv.predict(X_test), plot_confusion_matrix(y_test,yhat)

- https://github.com/josemiguelregoterol/Applied-Data-Science-Capstone

# Results

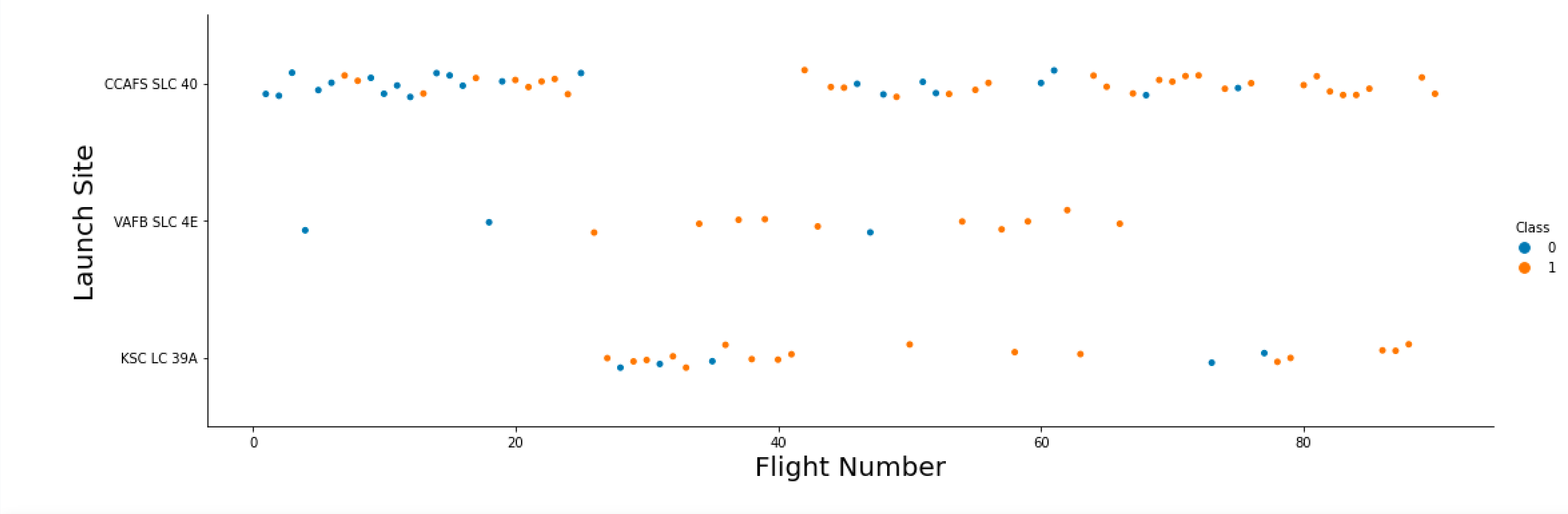
- SpaceX is improving the launches. The robust and constant rise of the success ratio from 2010 to 2020 is driving this company steady to the main goal: make spaceship's launches affordable.
- Some locations (KSC-LC) and boosters (FT) have an excellent success ratio (higher than 70%). It means that SpaceX employees are learning from mistakes (costly mistakes).
- This data is amazing for ML classification: the spaceship did not land or landed. Best Algorithm is Tree with a score of 0.875. Best Params is : {'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}

Section 2

# Insights drawn from EDA

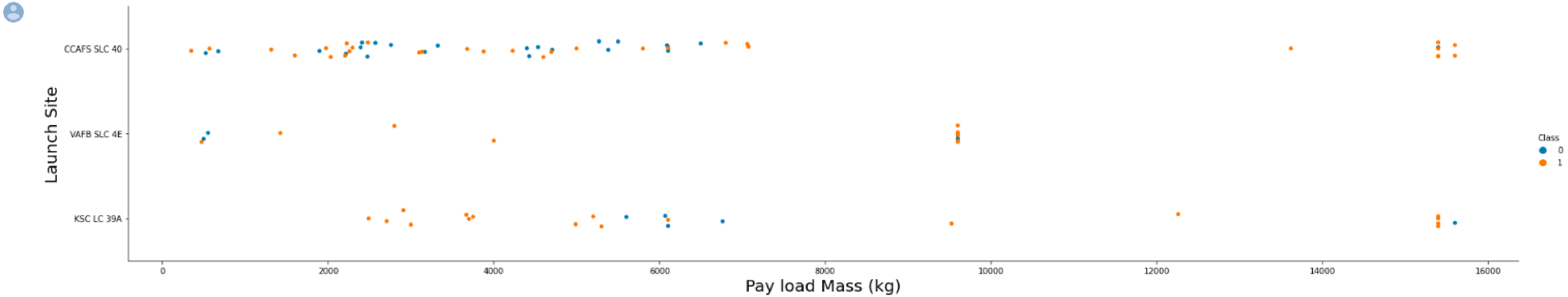
Careful! EDA based on plots are correlations!
Further analysis is required to draw any conclusion!
DO NOT JUMP INTO CONCLUSIONS FROM CORRELATIONS!

# Flight Number vs. Launch Site




For CCAFS there are two clusters (gap between clusters) of FN panning from the beginning to the last FN. For KSC there are three clusters (two gaps) starting from early mid FN to the latest FN

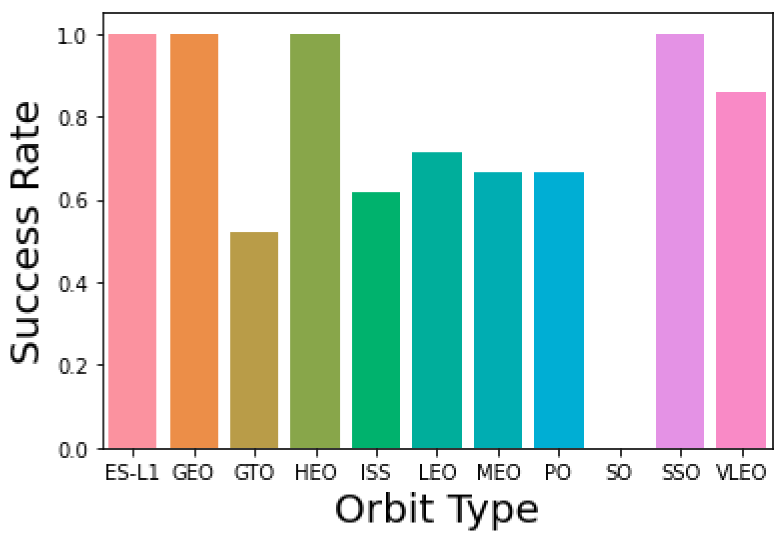
# Payload vs. Launch Site




Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

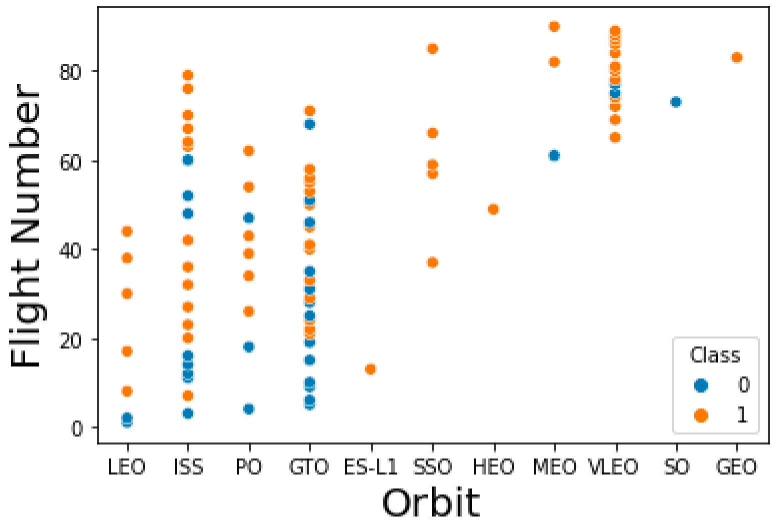
Maybe because it is too expensive to lunch those heavy rockets from VAFB-SLC? Transportation problem? Weather Issues? There was a strike at VAFB duringthe heavy-rocket-lunch week?

# Success Rate vs. Orbit Type



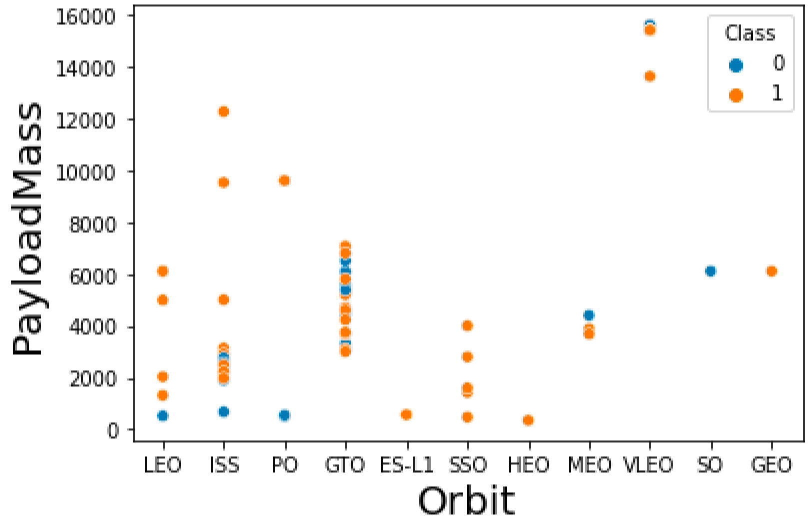

Analyze the ploted bar chart try to find which orbits have high sucess rate.

# Flight Number vs. Orbit Type




You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

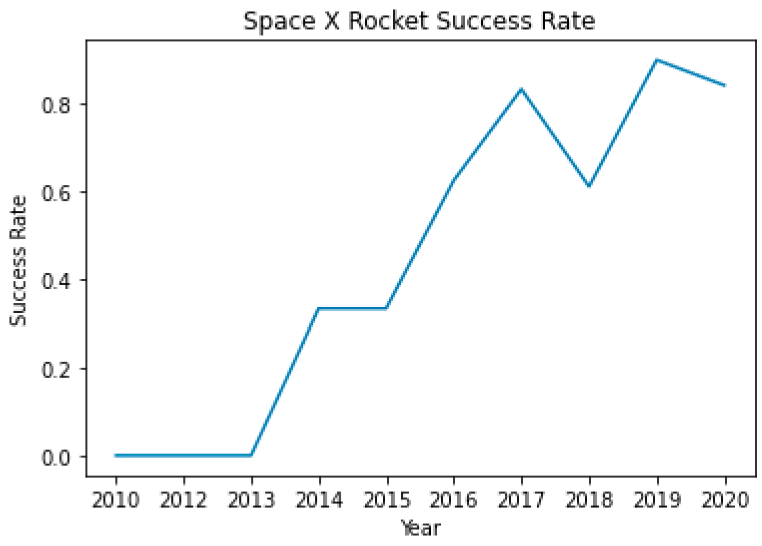
# Payload vs. Orbit Type




With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

# Launch Success Yearly Trend




you can observe that the sucess rate since 2013 kept increasing till 2020

# All Launch Site Names

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

[5]:
```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

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

[5]:

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

# Launch Site Names Begin with 'CCA'

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

```
[6]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

[6]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

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

```
[7]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

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

[7]:

| SUM(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

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

```
[8]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'

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

[8]:

| AVG(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

# First Successful Ground Landing Date

```
[11]: %sql SELECT * FROM SPACEXTBL WHERE 1=0
```

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

[11]:

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

```
[23]: %sql SELECT (SELECT min(DATE) FROM SPACEXTBL WHERE "Landing _Outcome"='Success (ground pad)') AS FIRST_SUCCESFUL_LANDING
```

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

[23]:

| FIRST_SUCCESFUL_LANDING |
|---|
| 01-05-2017 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

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

[18]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND "Landing _Outcome"='Success (drone ship)'
```

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

[18]:

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

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[27]:
```
%%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

[27]:

| SUCCESS | FAILURE |
|---|---|
| 100 | 1 |

# Boosters Carried Maximum Payload

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

```
[19]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

[19]:

| Booster_Version |
|---|
| F9 B5 B1048.4 |
| F9 B5 B1049.4 |
| F9 B5 B1051.3 |
| F9 B5 B1056.4 |
| F9 B5 B1048.5 |
| F9 B5 B1051.4 |
| F9 B5 B1049.5 |
| F9 B5 B1060.2 |
| F9 B5 B1058.3 |
| F9 B5 B1051.6 |
| F9 B5 B1060.3 |
| F9 B5 B1049.7 |

# 2015 Launch Records

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.

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

[26]:
```
%%sql
SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE"
FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

[26]:

| MONTH | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

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

## Task 10

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

[43]:
```
%%sql
SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'        '
GROUP BY "LANDING _OUTCOME" ORDER BY COUNT("LANDING _OUTCOME") DESC
```

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

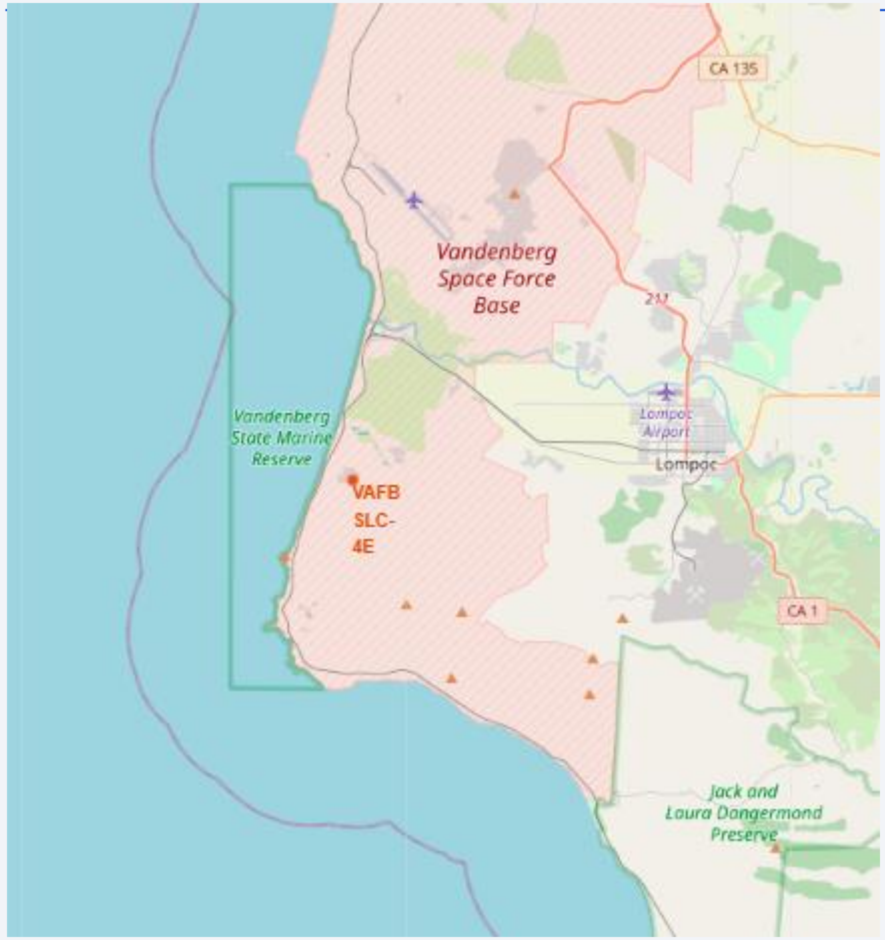
[43]:

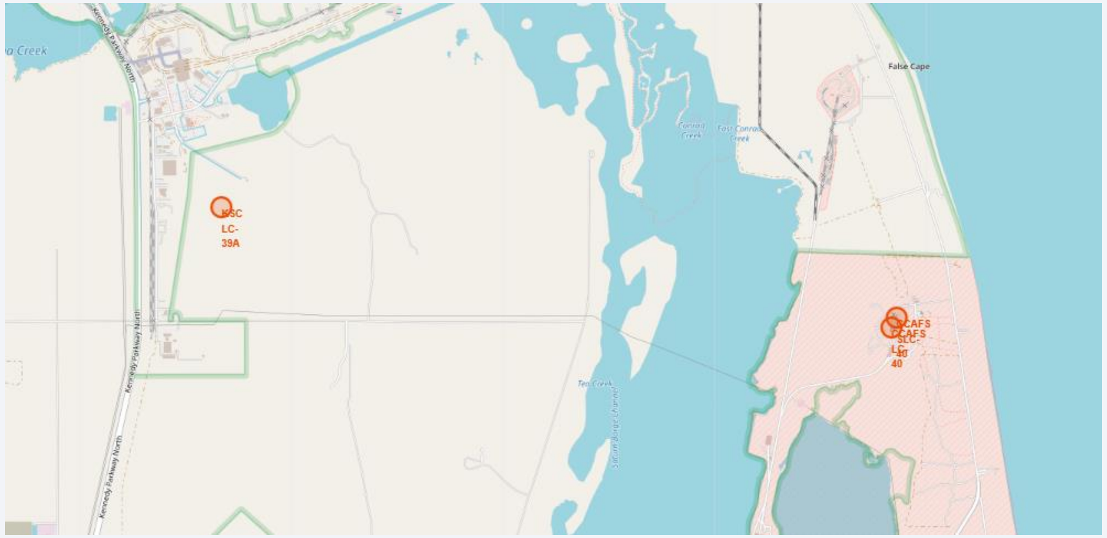
| Landing _Outcome | COUNT("LANDING _OUTCOME") |
|---|---|
| Success | 20 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |

Section 3

# Launch Sites Proximities Analysis

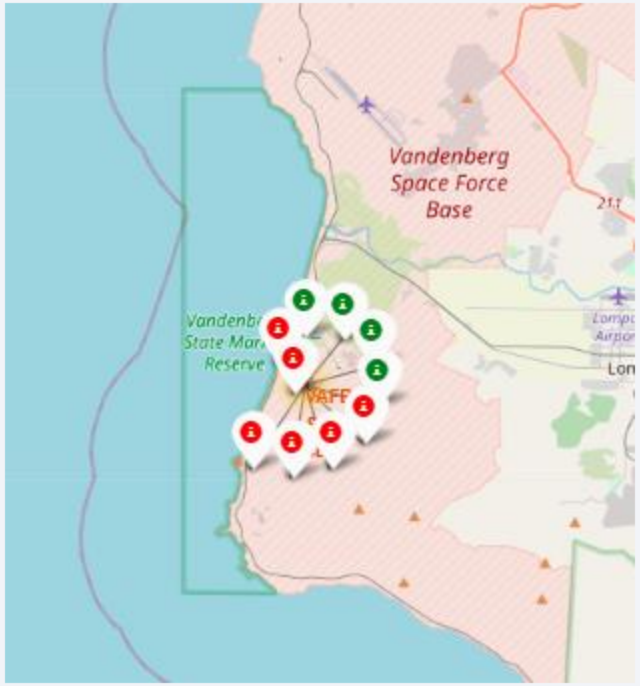
# Folium Map Screenshot: all launch sites' location markers (4) on the USA

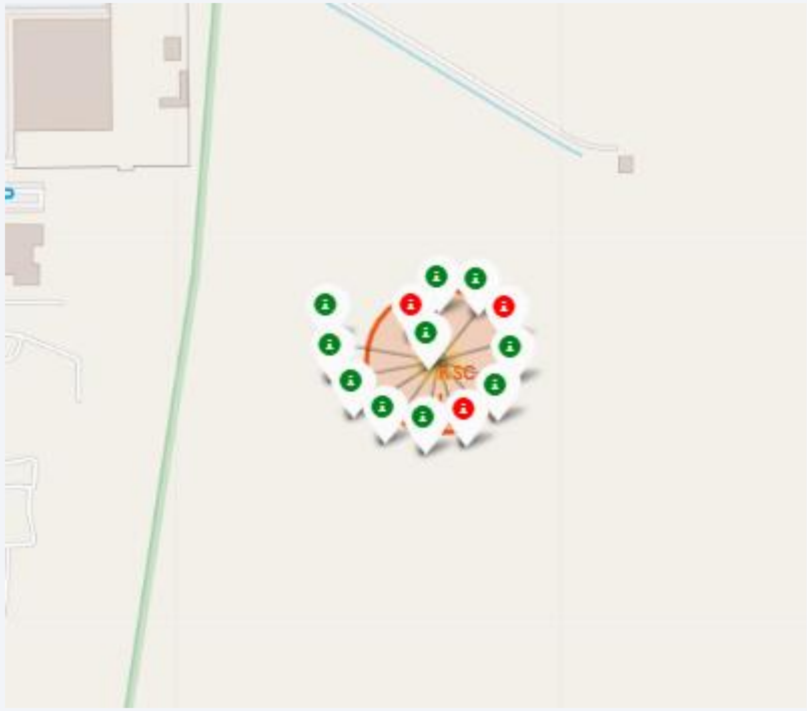


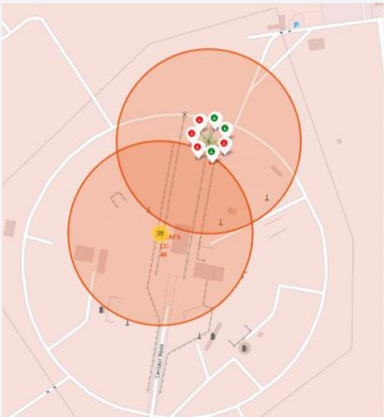



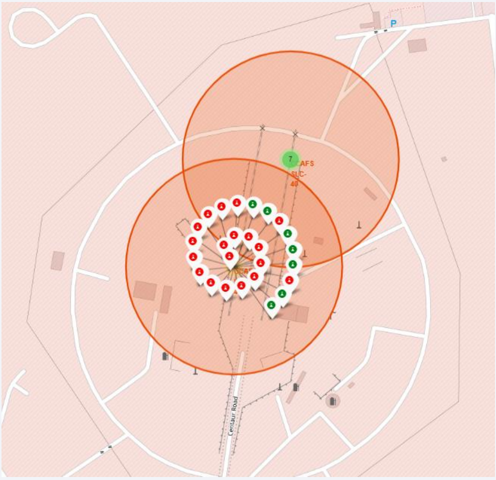

- The most important elements in this map are the lunch sites' location markers made from their coordinates using folium map.

# Folium Map Screenshot 2: Clustering fail and success launches



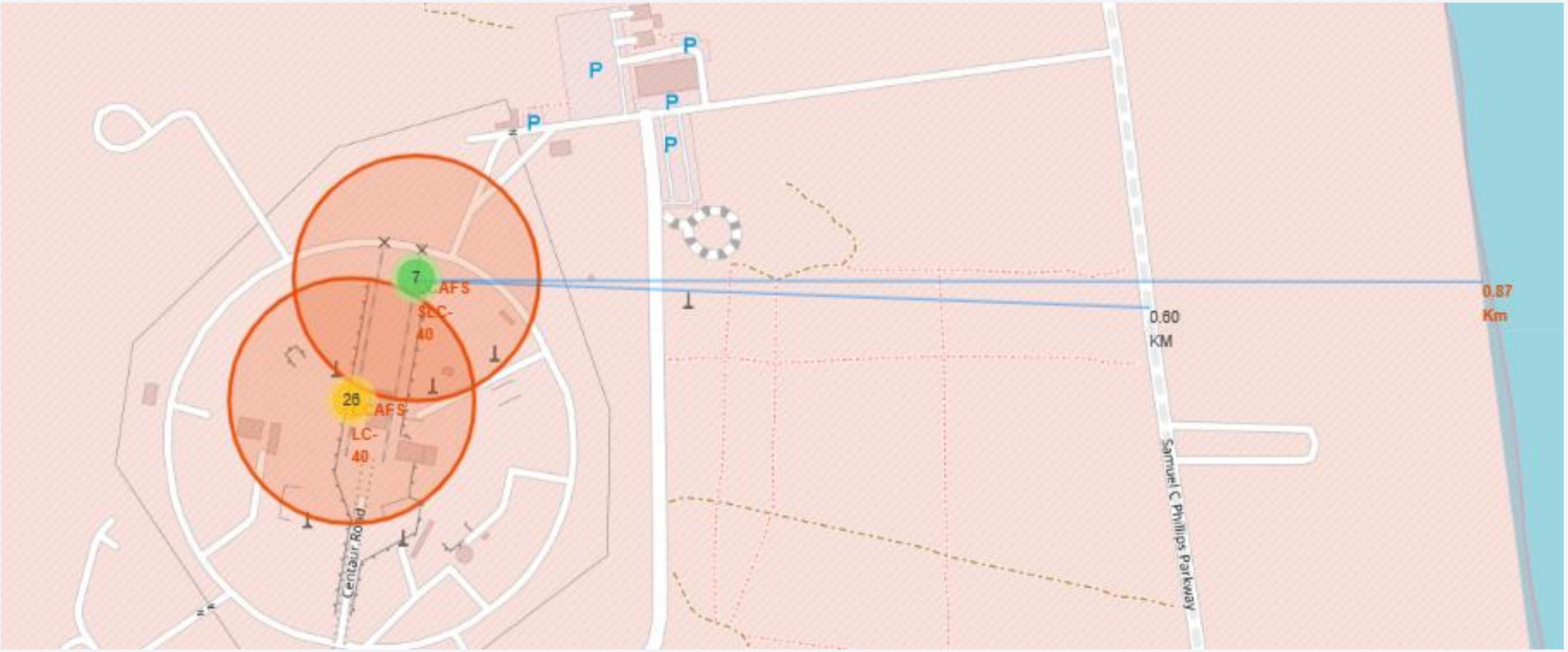







- By creating a column with binary values, red = fail and green = success, the launches are clustered into these 2 groups within the location marker.

# Folium Map Screenshot 3: Measuring distance with Folium

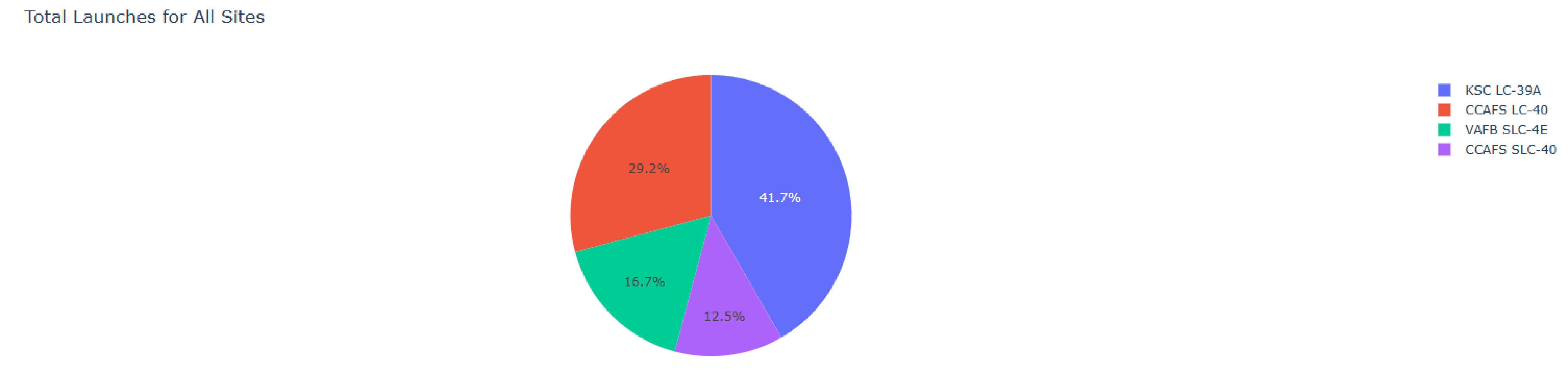



- Folium also gives the option to get the coordinates of the map from pointer of the mouse. This feature of folium is useful to measure distance between markers.

Section 4

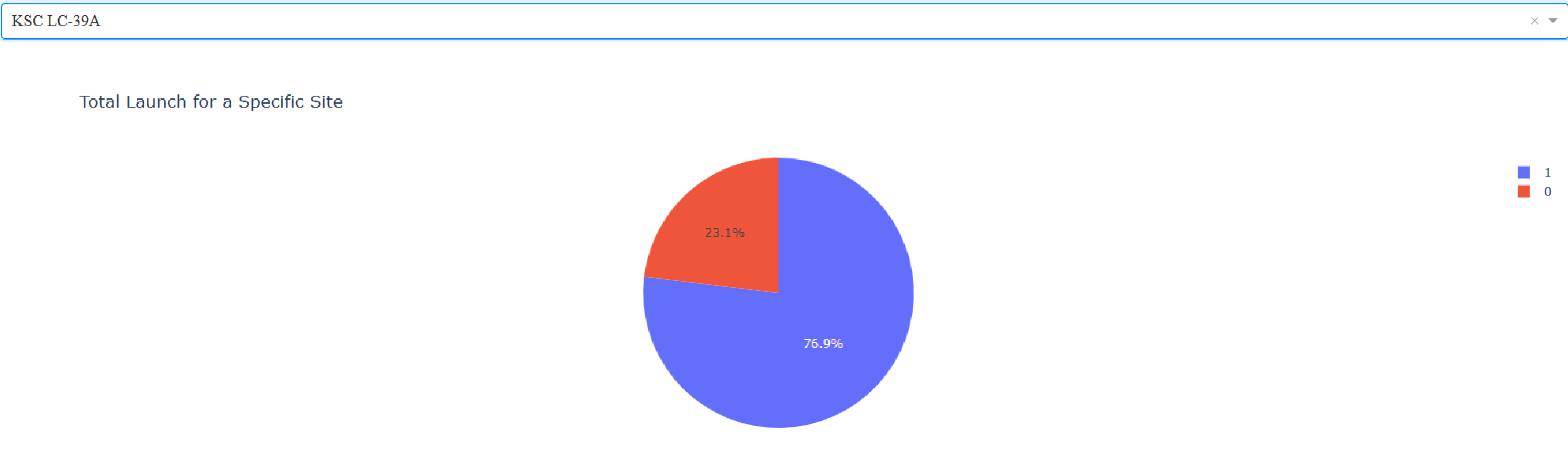
# Build a Dashboard with Plotly Dash

# SpaceX success rate per launch sites (2010 - 2020)




- Written in HTML, the plot is made as an output from a selection (input) embebbed in a callback. This pie chart plots the default selection (All sites)

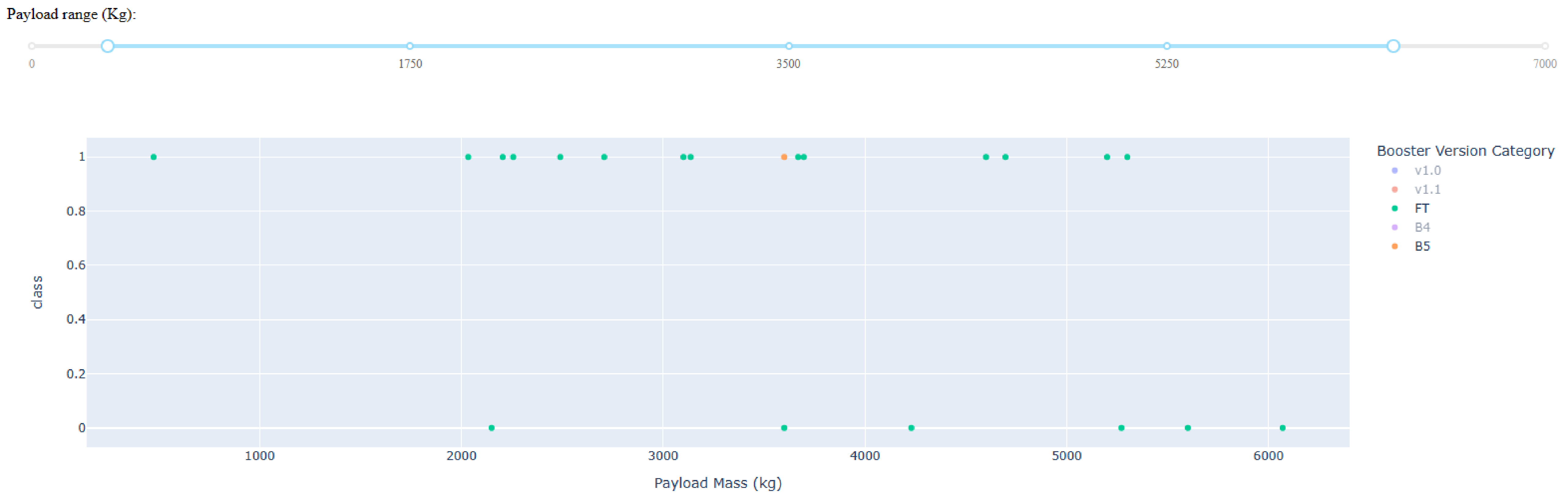
# Less than one forth is the lowest failure ratio for SpaceX




- In this case, the selection triggers the callback to display the values of a specific launch site.

# Comparative of two SpaceX booster with a higher success ratio than 70%

- FT booster has an excellent success ratio with more than 15 deployments. However, B5 booster has a perfect success ratio (1 out of 1).



Section 5

# Predictive Analysis (Classification)

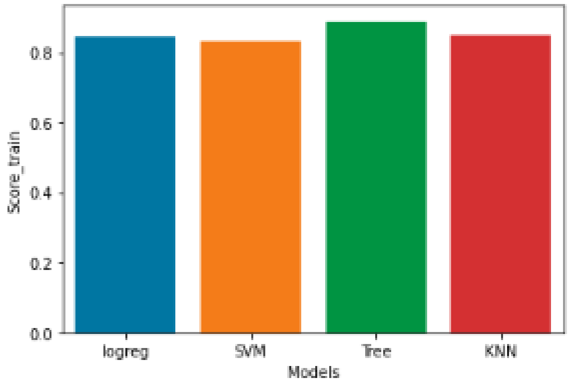
# Classification Accuracy

```
d = {'Score_train': pd.Series([0.8464285714285713,0.8333333333333334, 0.8892857142857142, 0.8482142857142858]), 'Models': ['logreg', 'SVM', 'Tree', 'KNN']}
d2=pd.DataFrame(data=d)
d2.head()
```

| | Score_train | Models |
|---|---|---|
| 0 | 0.846429 | logreg |
| 1 | 0.833333 | SVM |
| 2 | 0.889286 | Tree |
| 3 | 0.848214 | KNN |

```
[37] sns.barplot(data=d2, x="Models", y="Score_train")
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fdbb331c730>

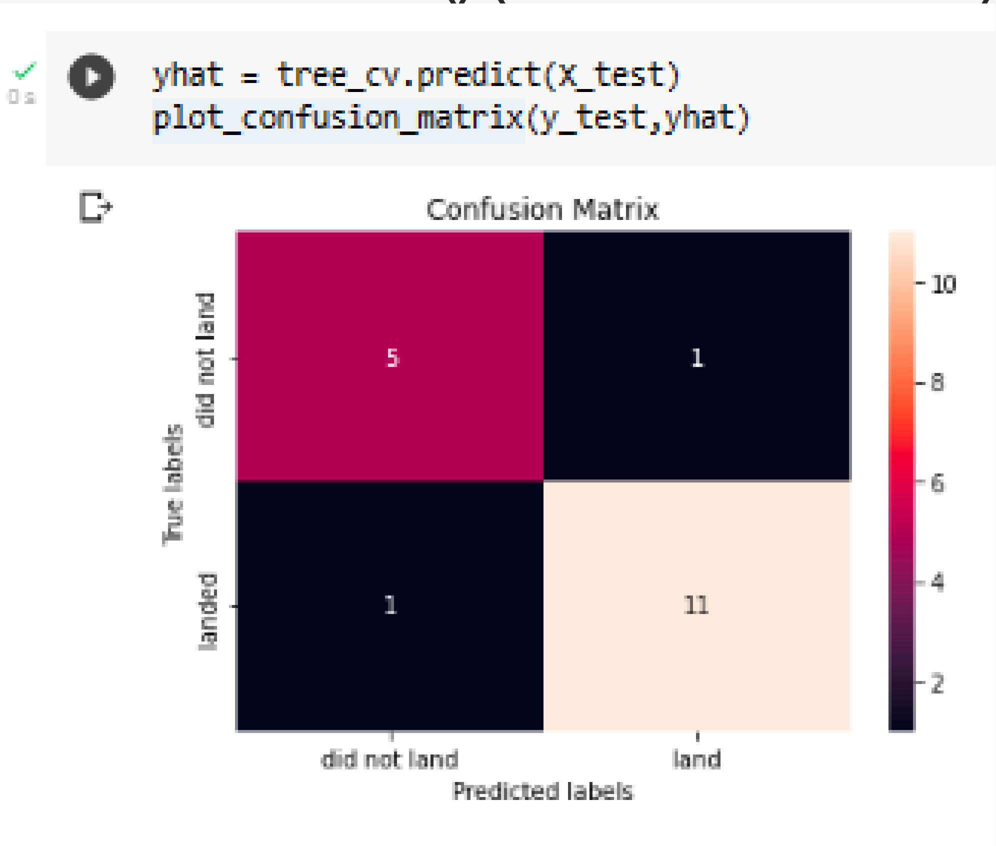



```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
```

# Confusion Matrix

- Confusion matrix of DecisionTreeClassifier() (Train Score = 0.88). Excellent TP and TN



# Conclusions

- Long story short: In ten years SpaceX is closer to its main goal: affordable spaceship travels.

# 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!