IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nikher Verma

30th October 2022

Table of Contents.

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

Problem Definition

- SpaceY is company who stepped recently in the space explorations and using the previous data ,of SpaceX a well renowned company that designs, manufactures and launches rockets and spacecrafts and dominate the market because of the reusability of its rockets, to find the cheapest and safest way to make space exploration affordable for everyone. However, not all rockets launched land successfully.

The problem is how we can leverage this unsuccessful landing and develop methodologies to predict if a rocket would land successfully or not because once done successfully a lot of cost can be saved and helps the company to predict many constraints.

The machine learning models would enable our company, SpaceY to determine the success of rocket landing. This is particularly significant as a failed landing could lead to huge financial losses. The prediction would help the cost analysis and make better offers than SpaceX.

Executive summary

- *Summary of methodologies*
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- *Summary of all results*
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- *Project background and context ?*

The aim of this project is to predict if the Falcon 9 first stage will successfully land. SpaceX can reuse the first stage and by doing so saves almost a 100 million. By determining if the 1st stage will land, we can determine the cost of a launch. This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.

- *Problems you want to find answers?*

• What are the main characteristics of a successful or failed landing ?

• What are the effects of each relationship of the rocket variables on the success or failure of a landing ?

• What are the conditions which will allow SpaceX to achieve the best landing success rate

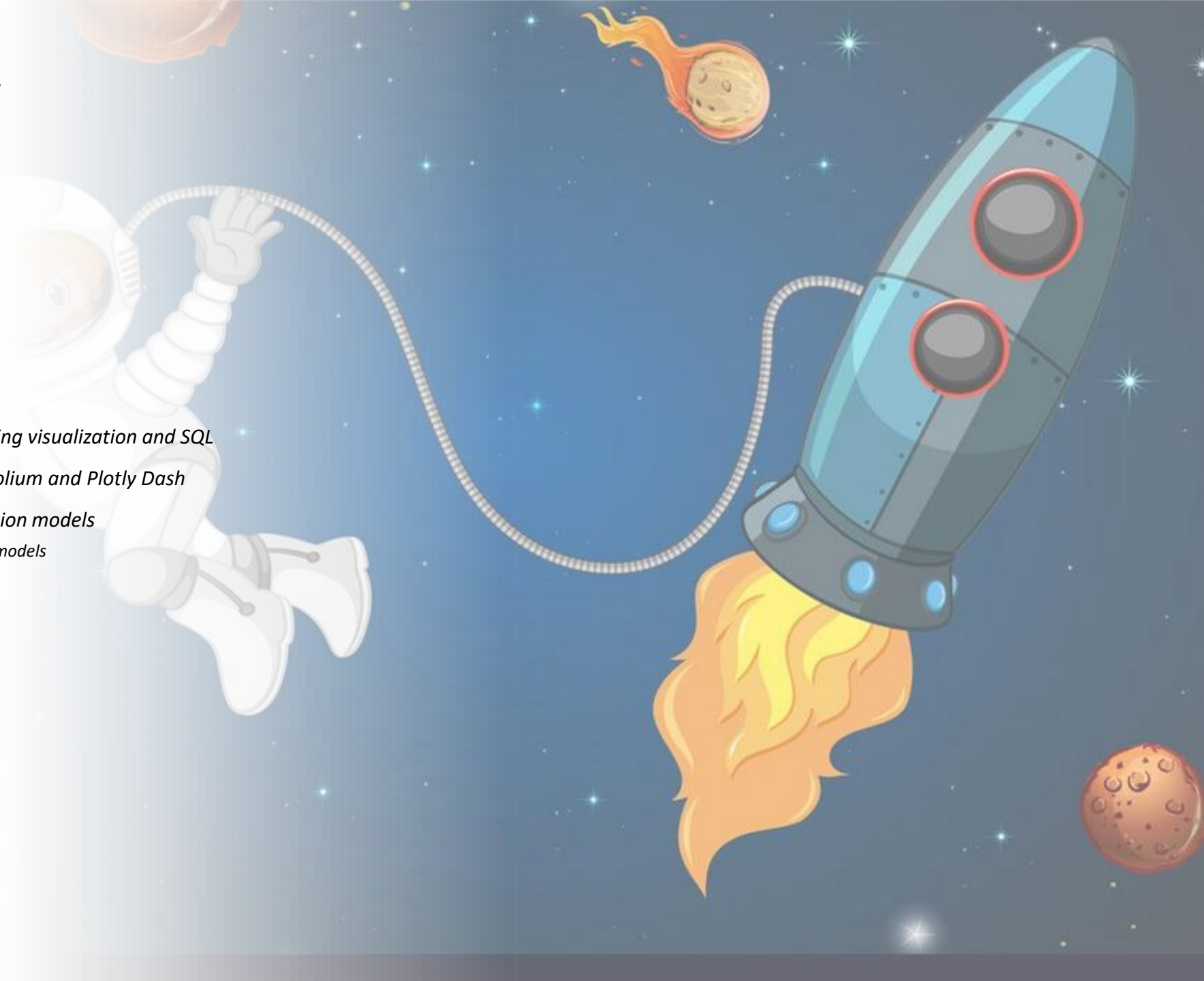
Section 1

Methodology

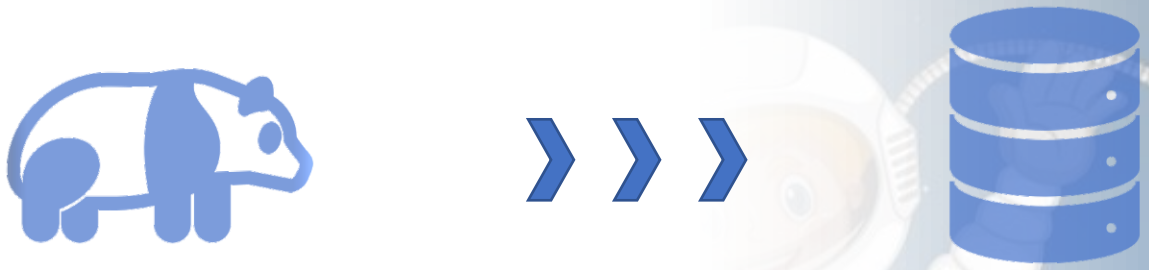
Methodology

Executive Summary

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



Data Collection - SpaceX API



Make Request to SpaceX API.

Decode the response content as a Json.

Turn Json into Pandas Data Frame.

API used to get info about Each launch using ID’s

Construct Data frame and filter for Falcon9

Replace NaN with calculated mean and Save into CSV file.

Github link: [Capstone-Project/1. Complete the Data Collection API Lab.ipynb at main · vermanikher/Capstone-Project (github.com)]

Data Collection – SpaceX API

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

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

3. Transform data

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

4. Create dictionary with data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe

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

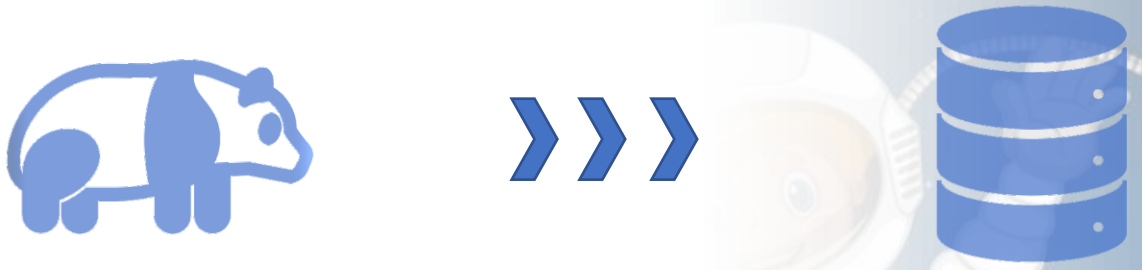
6. Filter dataframe

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

7. Export to file

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

Data Collection - SpaceX API



Request Falcon9 Launch Wiki page from its URL.

Create BeautifulSoup from HTML response.

Extract all attributes from HTML Header.

Create Empty Dictionary with keys.

Fill Dictionary with launch records extracted from table rows.

Convert dictionary into CSV Dataset.

Github link: [Capstone-Project/2. Data Collection with Web Scraping.ipynb at main · vermanikher/Capstone-Project (github.com)]

Data Collection - Scraping

1. Getting Response from HTML

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

2. Create BeautifulSoup Object

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

3. Find all tables

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

4. Get column names

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

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add data to keys

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

See notebook for the rest of code

7. Create dataframe from dictionary

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

8. Export to file

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

Data Wrangling

Github link:

Data Wrangling :

- Did booster land successfully ?
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- Transform string variables into categorical variables where-
  - 1 - Mission Successful.
  - 0 - Mission Failed.

Github link: [Capstone-Project/3. Complete the EDA lab.ipynb at main · vermanikher/Capstone-Project (github.com)]

Data Wrangling

1. Calculate launches number for each site

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

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

2. Calculate the number and occurence of each orbit

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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

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

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

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

4. Create landing outcome label from Outcome column

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

5. Export to file

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

EDA with Data Visualization

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

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

- ***Bar graphs-*** show the relationship between numeric and categoric variables.
  - Success rate vs. Orbit

- ***Line graphs -*** show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.
  - Success rate vs. Year

Github link: [Capstone-Project/3. Complete the EDA lab.ipynb at main · vermanikher/Capstone-Project (github.com)]

EDA with SQL

- Loading the dataset into the corresponding table in a Db2 database, and executing SQL queries to answer following questions:
  - Displaying the names of the unique launch sites in the space mission.
  - Displaying 5 records where launch sites begin with the string 'CCA’.
  - Displaying the total payload mass carried by boosters launched by NASA (CRS).
  - Displaying average payload mass carried by booster version F9 v1.1.
  - Listing the date when the first successful landing outcome in ground pad was achieved.
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - Listing the total number of successful and failure mission outcomes.
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
  - Ranking 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

Github link: [Capstone-Project/5. Complete the EDA with Visualization.ipynb at main · vermanikher/Capstone-Project (github.com)]

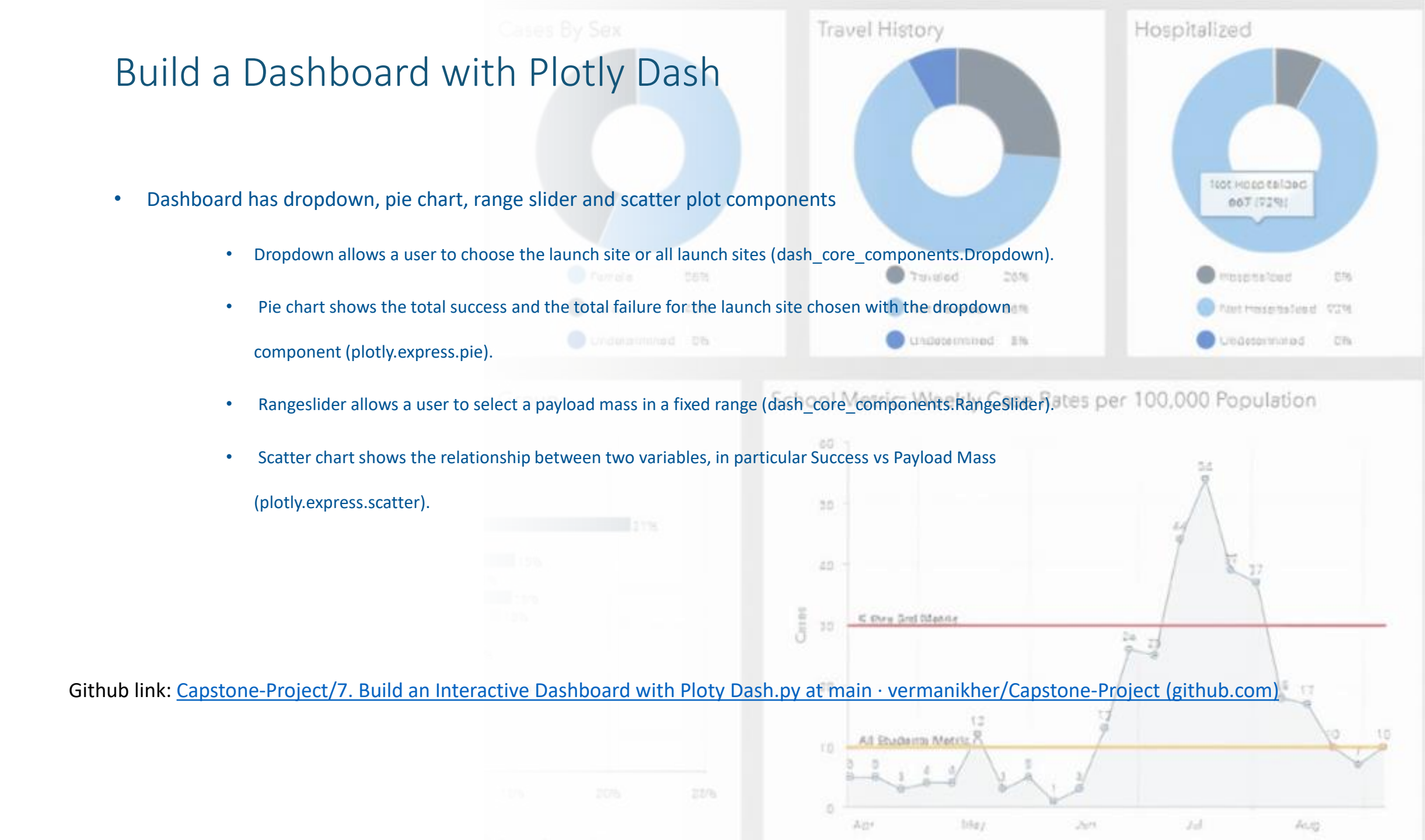
Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.Marker).
  - Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, folium.Icon).
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Github link: Capstone-Project/6. Complete the Interactive Visual Analytics.ipynb at main · vermanikher/Capstone-Project (github.com)

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, range slider and scatter plot components
    - Dropdown allows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
    - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (plotly.express.pie).
    - Rangeslider allows a user to select a payload mass in a fixed range (dash_core_components.RangeSlider).
    - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (plotly.express.scatter).



Github link: [Capstone-Project/7. Build an Interactive Dashboard with Ploty Dash.py at main · vermanikher/Capstone-Project (github.com)]

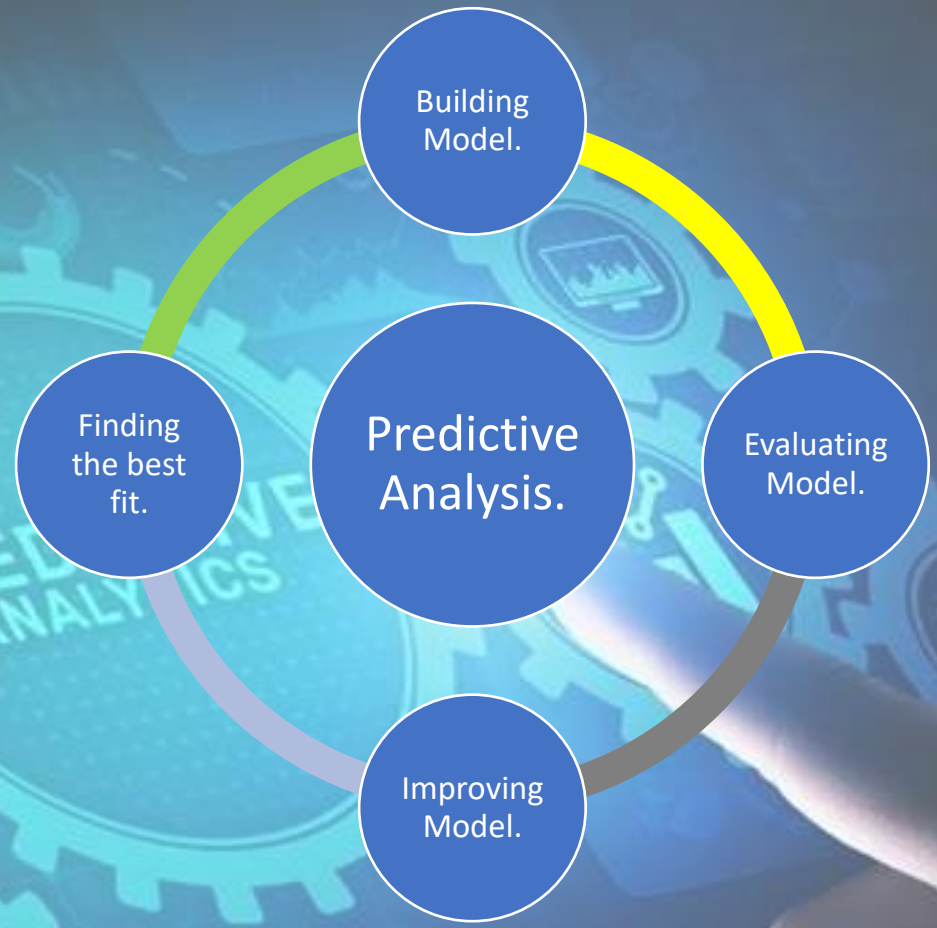
Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels.

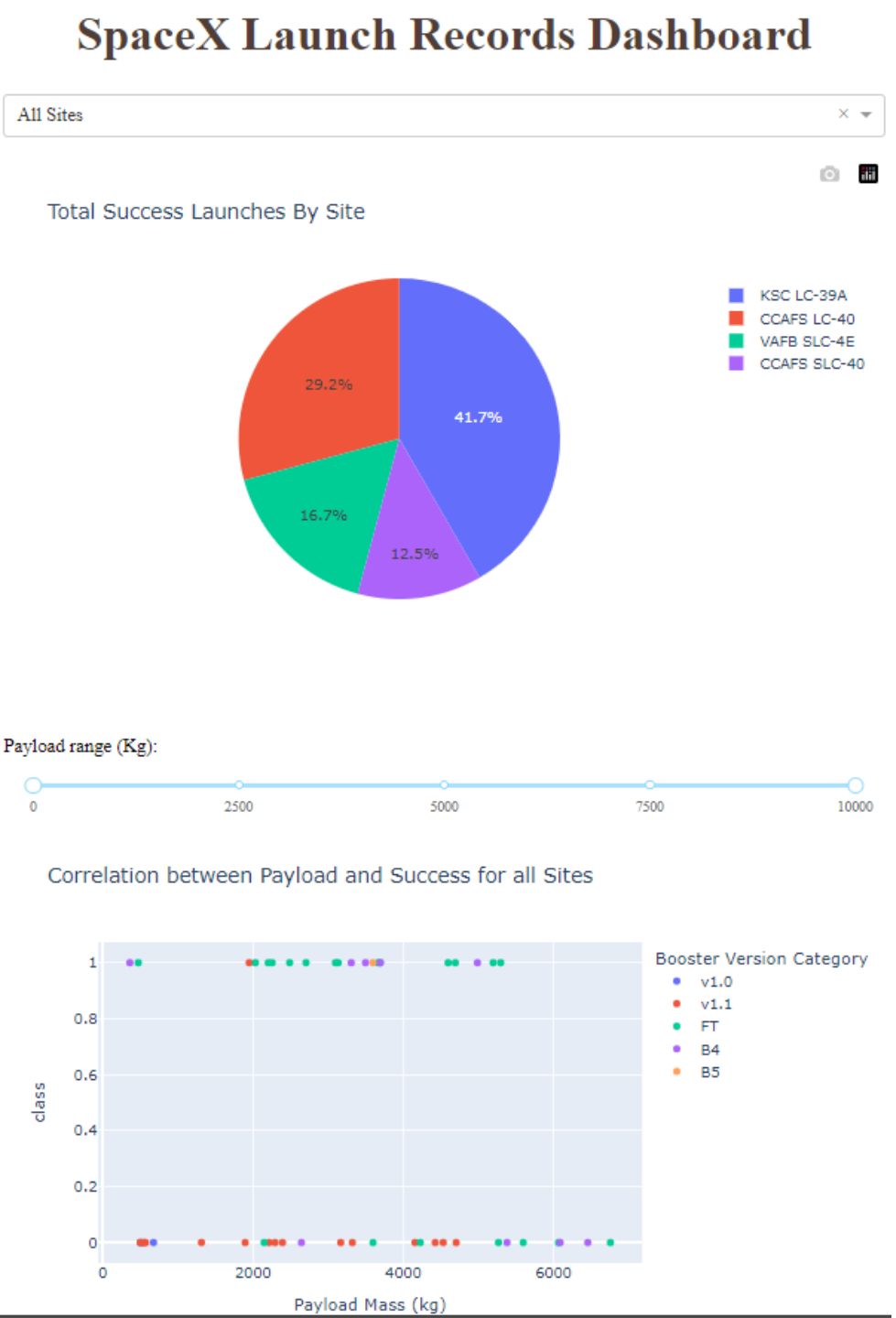
- Create a column for the class
- Standardize the data
- Split into training data and test data

Find best Hyperparameter for SVM, Classification Trees and Logistic Regression.

- Find the method performs best using test data

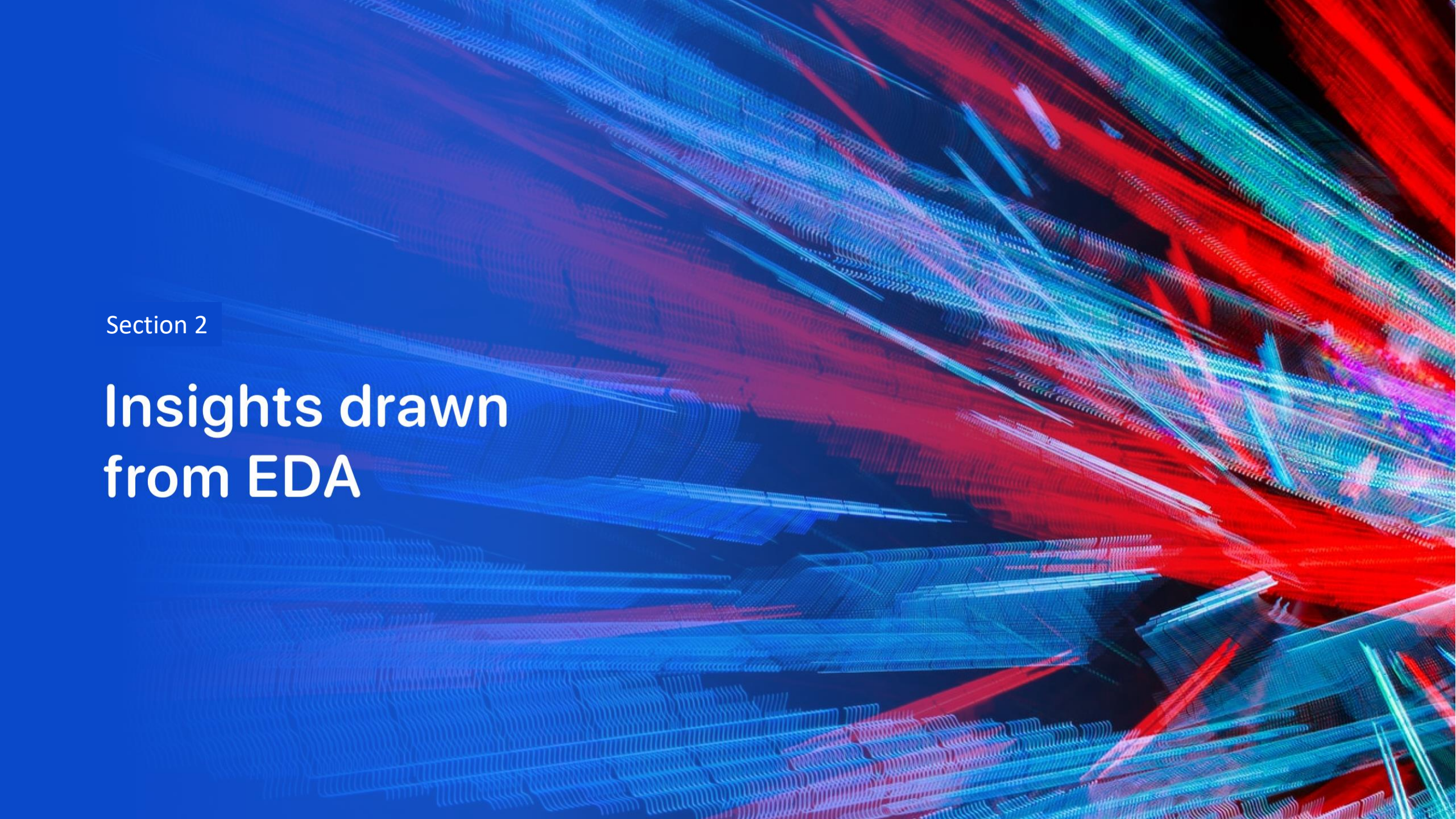


Github link:[Capstone-Project/8. Complete the Machine Learning Prediction lab.ipynb at main · vermanikher/Capstone-Project (github.com)]



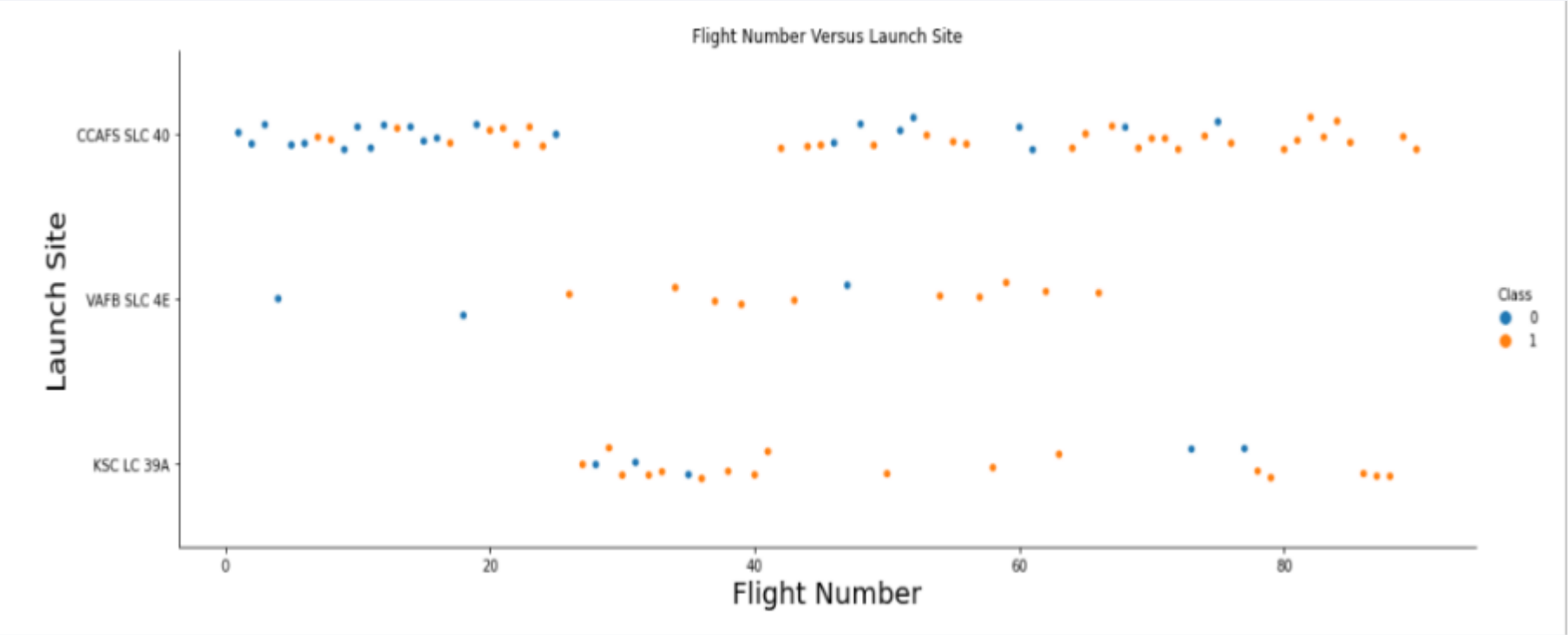
Results

- The left screenshot is a preview of the Dashboard with Plotly Dash.
- The results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and Interactive Dashboard will be shown in the next slides.
- Comparing the accuracy of the four methods, all return the same accuracy of about 83% for test data.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

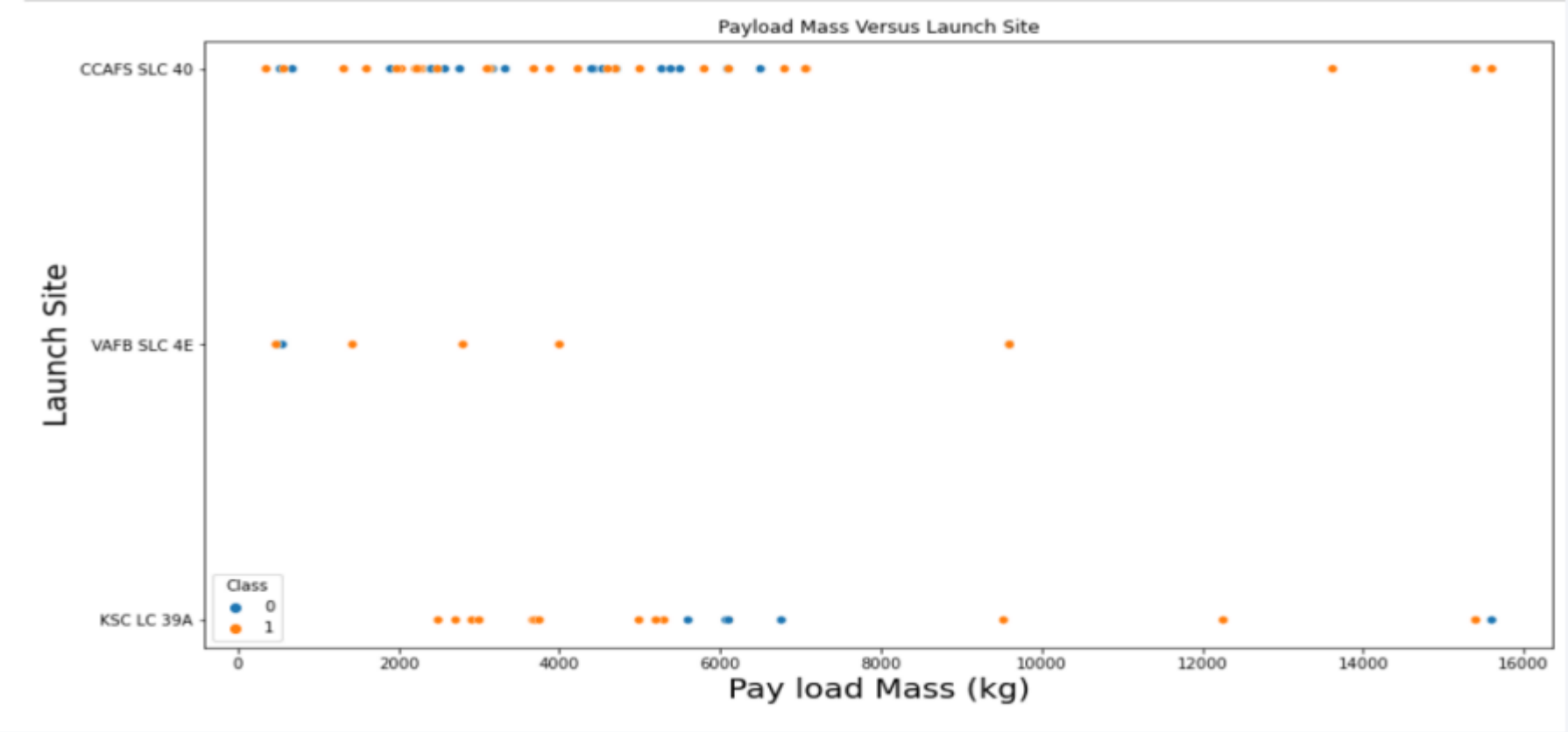
- The visualization depicts that the flight number is directly proportional to the number of successful landings.



- The launch site CCAFS SLC 40 had the greatest number of landing attempts while the site VAFB SLC 4E had the least number of attempts.

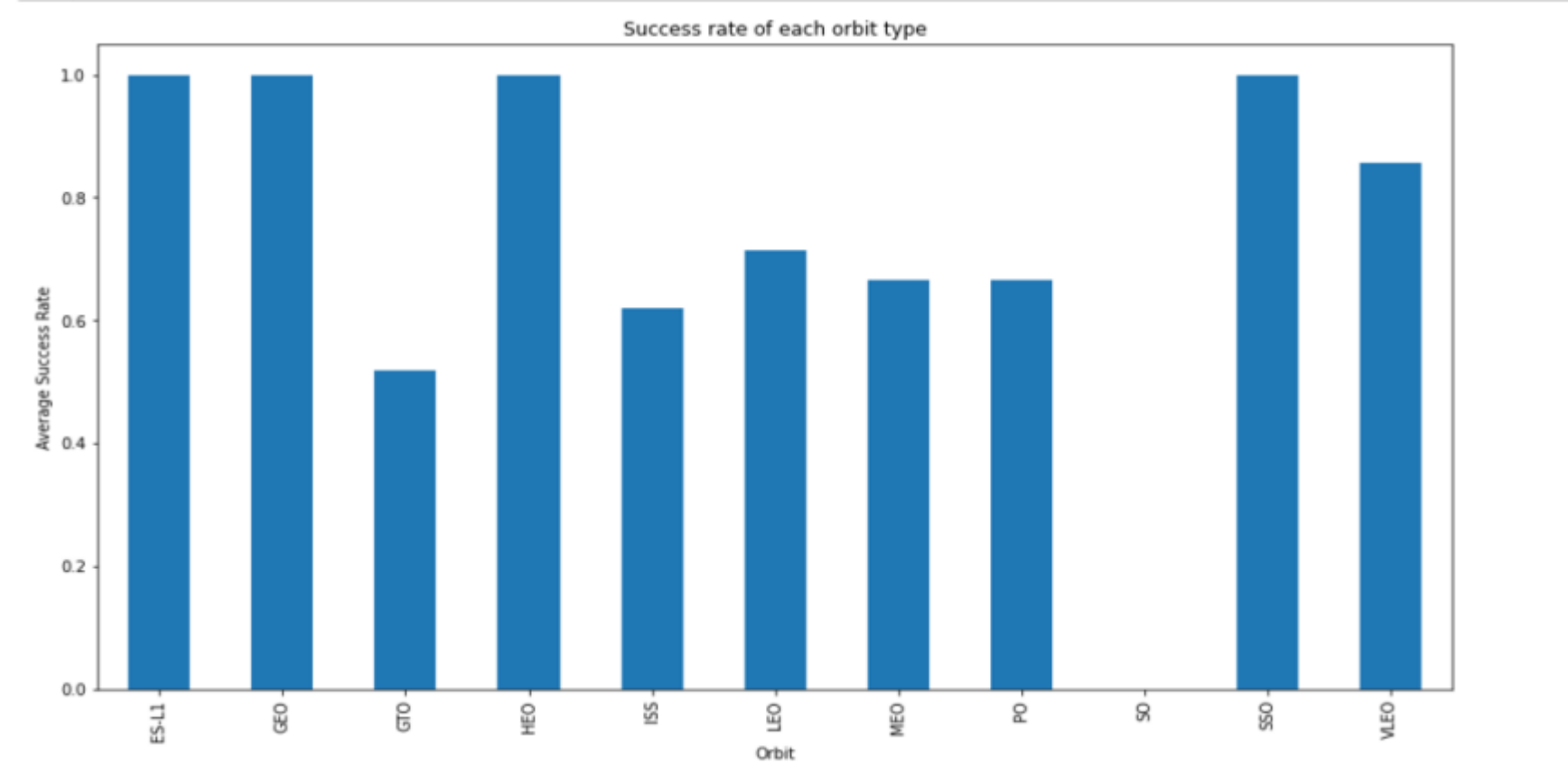
Payload vs. Launch Site

- The launch sites are sensitive to the mass of the payload in that rockets with payload mass greater than 10000kg are most unlikely to be launched at the sites.



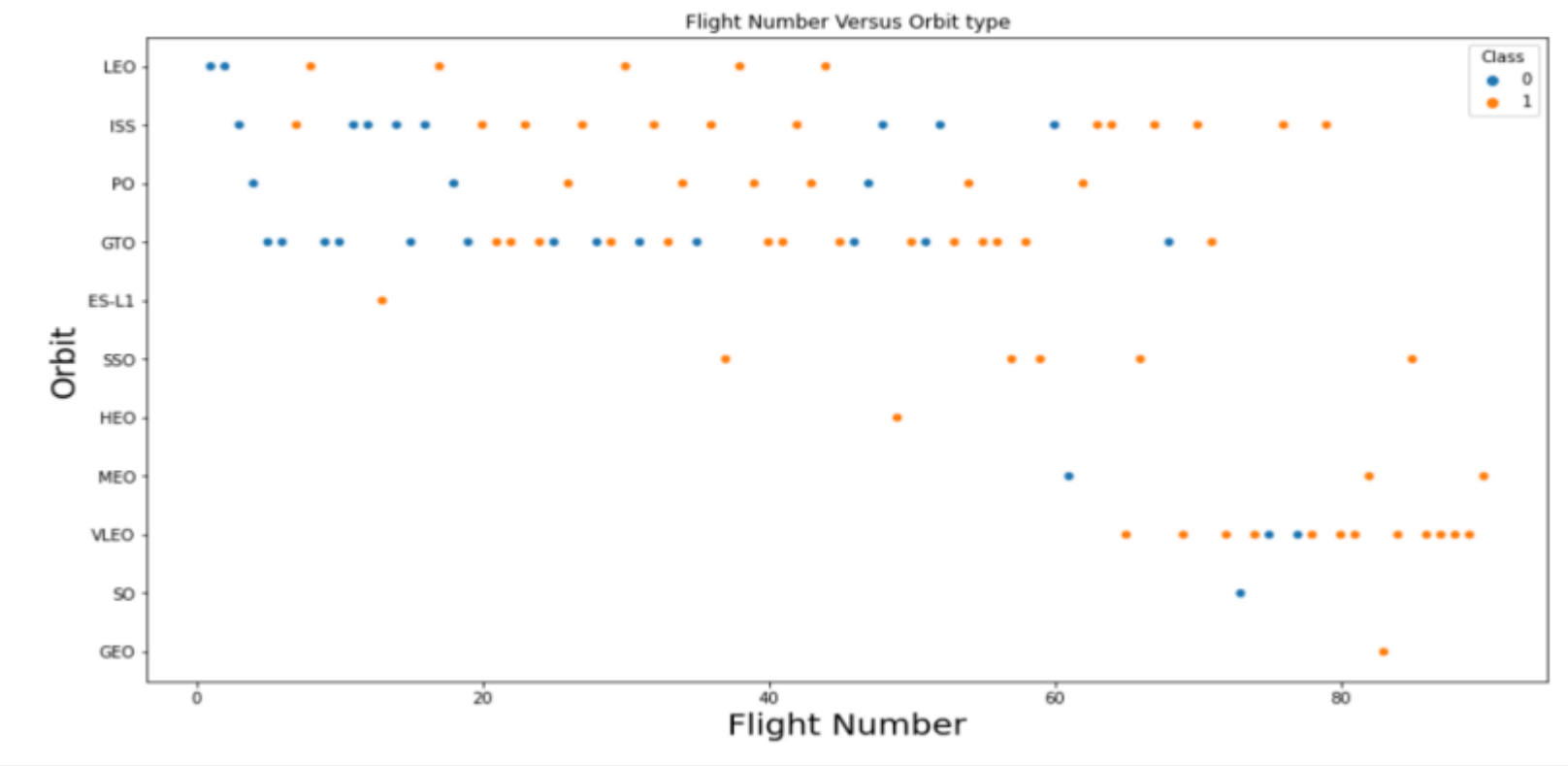
Success Rate vs. Orbit Type

- Four (4) of the Orbits (ESL1, GEO, HEO and SSO) records the most success rate on the average. This should be a crucial factor to consider when launching a rocket .



Flight Number vs. Orbit Type

- LEO, SSO and VLEO appears to have strong correlation with the flight number in recording landing successes.

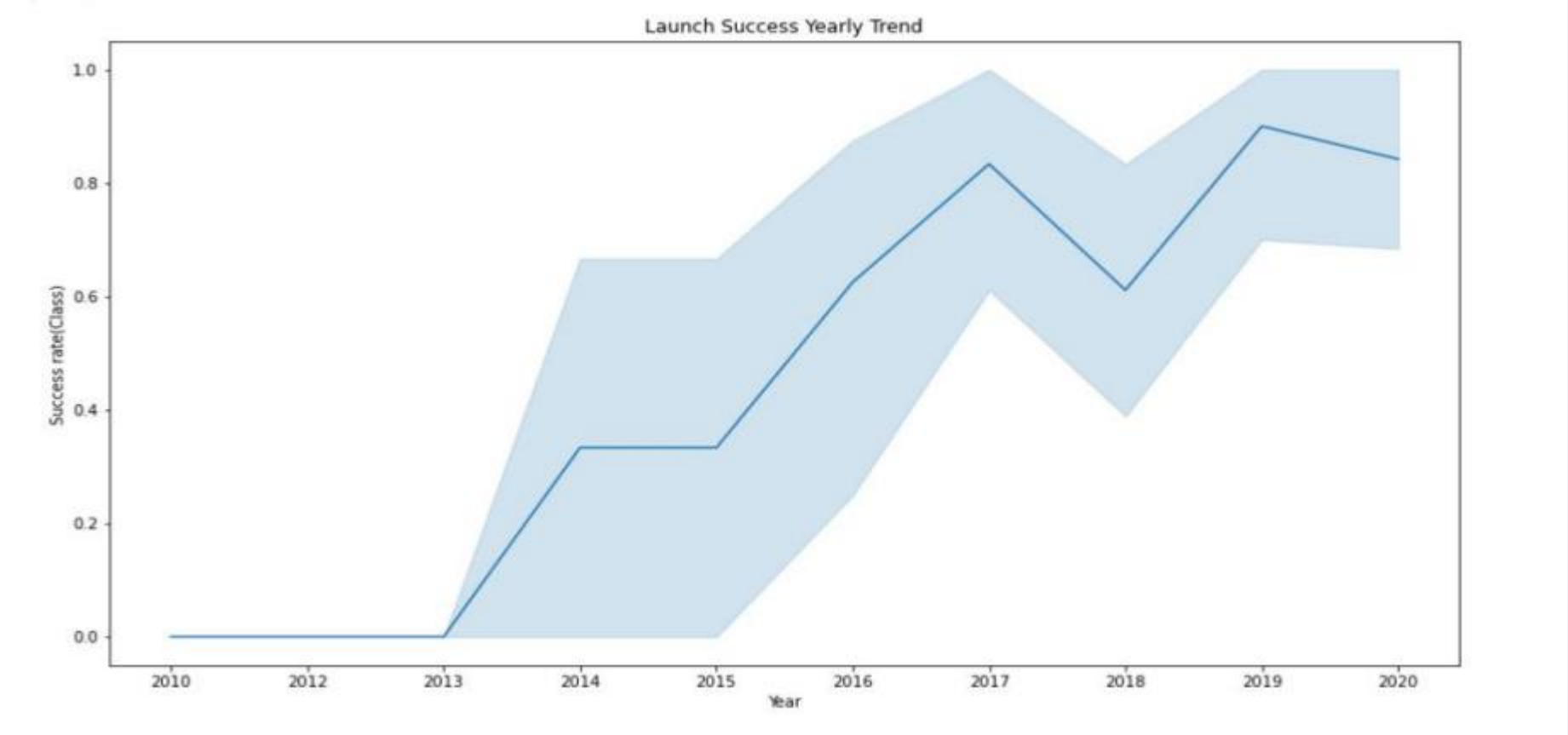


Payload vs. Orbit Type

- There are more successful launches for LEO and SSO Orbits when the payload mass is relatively small compared to GTO and ISS where the effect of the payload mass is not easily noticeable.
- GEO, ESL1 and GEO all had successful landing with low payload masses.

Launch Success Yearly Trend

- There has been a successful launch of the SpaceX Falcon 9 rocket from 2013. Albeit there was a slight decline in 2018 but the trend goes upward thereafter.



All Launch Site Names

- There are four (4) distinct launch sites gathered from the data viz:
- CCAFS LC-40
- CCAFS SLC-40
- KSC LC-39A
- VAFB SLC-4E

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- Query:

```
SELECT * FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Query:

```
SELECT SUM(PAYLOAD_MASS__KG_)
       AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Query:

```
SELECT AVG(PAYLOAD_MASS__KG_)
       AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Query:

```
SELECT MIN(DATE)
  AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME
        = 'Success (ground pad)'
```

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
    AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
SELECT MISSION_OUTCOME,
       COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
SELECT DISTINCT BOOSTER_VERSION,
       PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
       SELECT MAX(PAYLOAD_MASS__KG_)
       FROM SPACEXTBL)
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
SELECT LANDING__OUTCOME,
       BOOSTER_VERSION,
       LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME
          = 'Failure (drone ship)'
       AND YEAR(DATE) = '2015'
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
SELECT LANDING__OUTCOME,
       COUNT(LANDING__OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total_number DESC
```

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

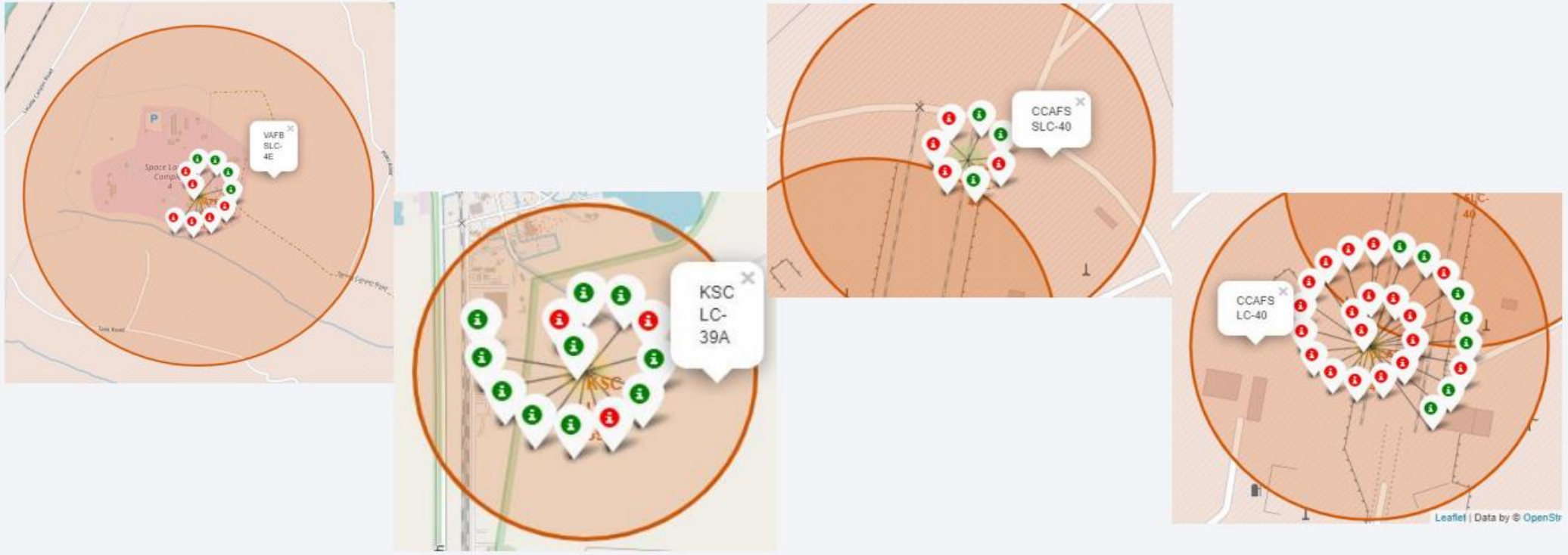
Section 3

Launch Sites Proximities Analysis

Folium map – Ground stations

Map1- Space X launch sites are located on the coast of the United States.

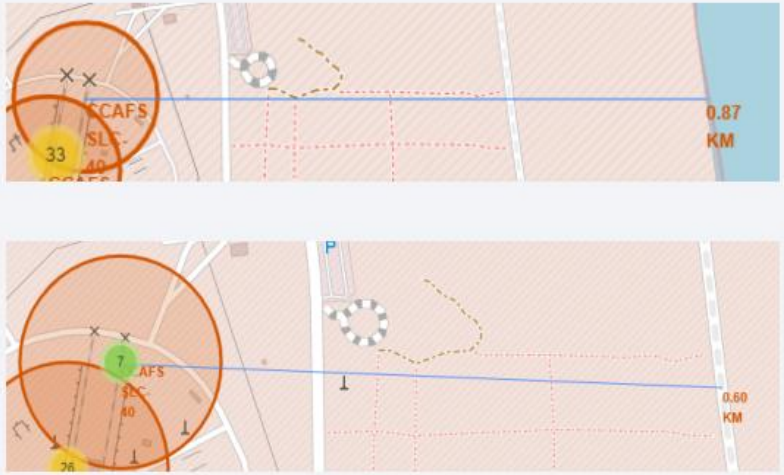
Folium map – Color Labeled Markers.

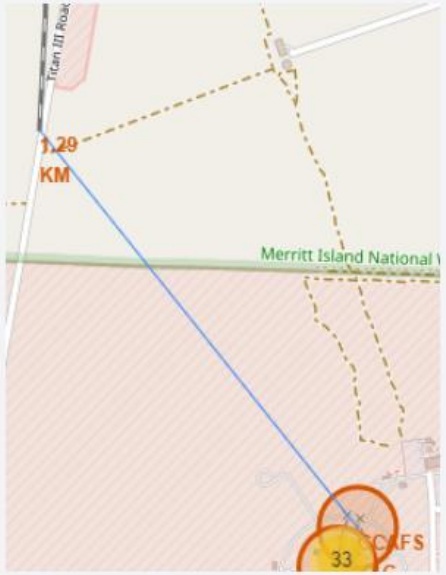


KSC LC-39A has a higher launch success rate.

- *Green marker* represents successful launches.
- *Red marker* represents unsuccessful launches.

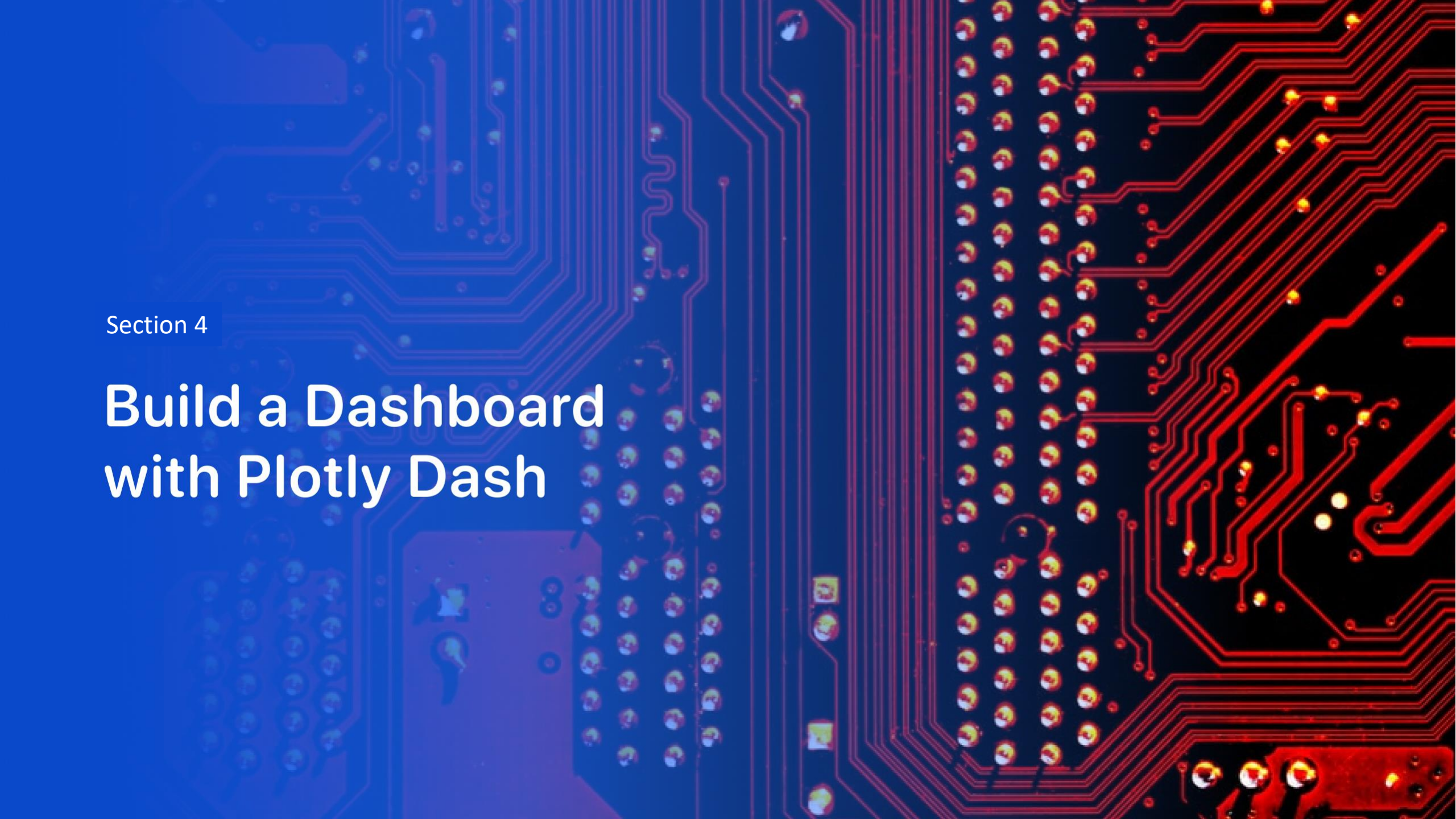
Folium Map – Distances between CCAFS SLC-40 and its proximities



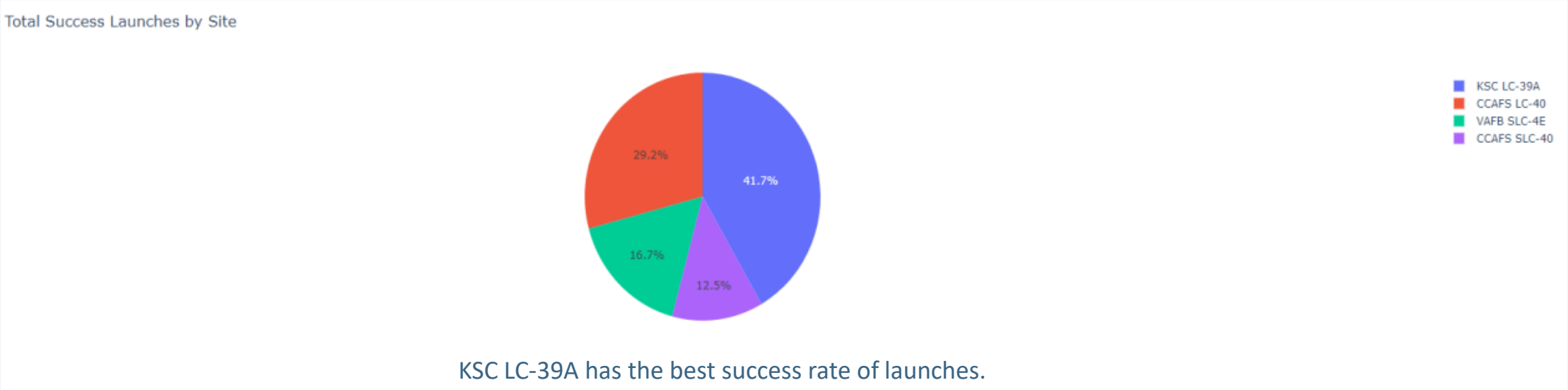


Proximity meter for CCAFS SLC – 40

- *The site is having all the means of transportation in proximity.*
- *Do CCAFS SLC-40 keeps certain distance away from cities ? No*

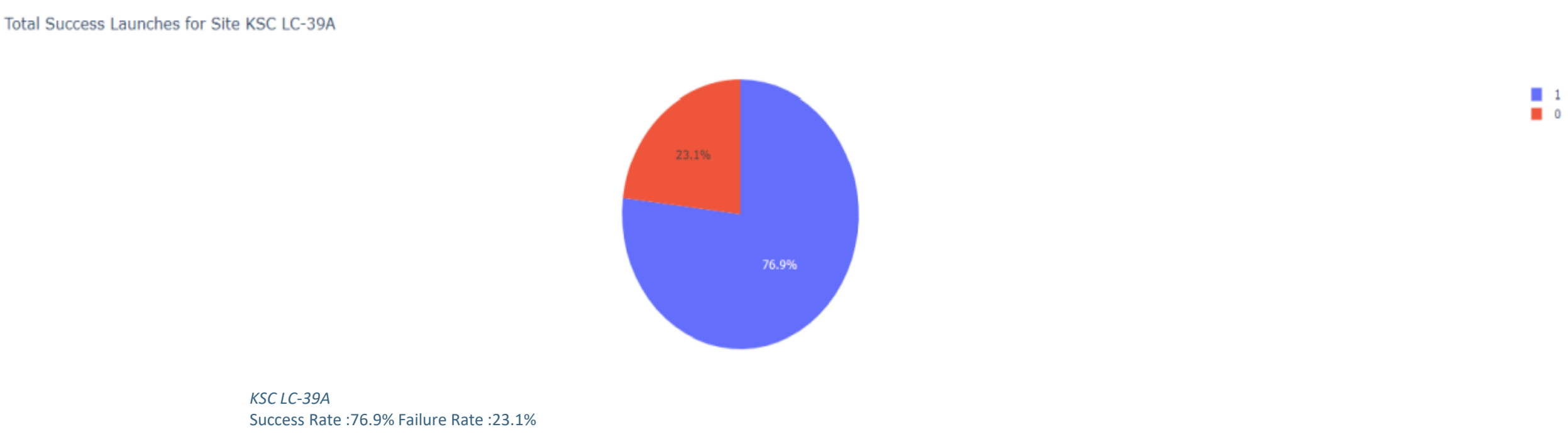

Section 4
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



KSC LC-39A has the best success rate of launches.

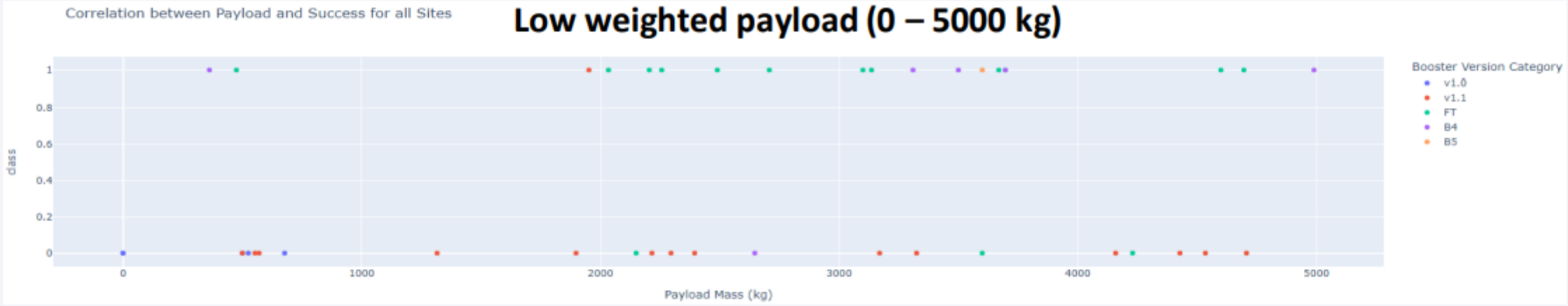
Dashboard – Total success launches for Site KSC LC-39A.

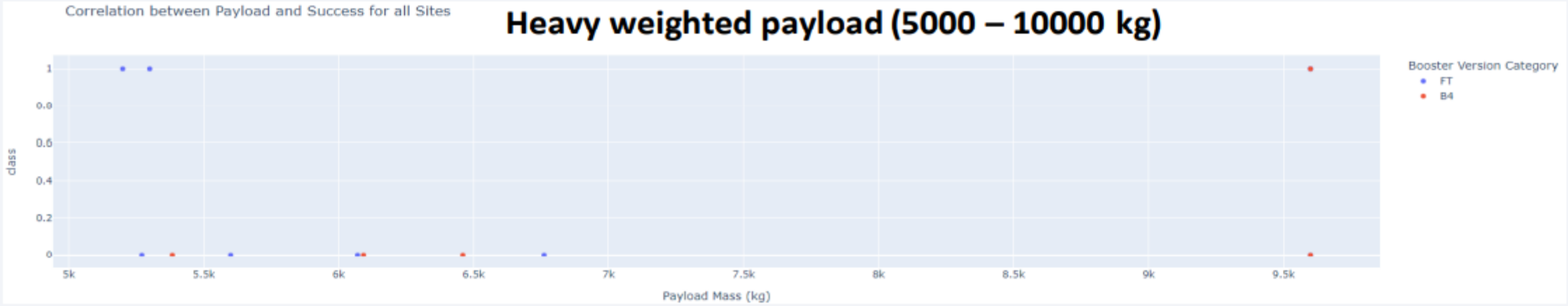


KSC LC-39A
Success Rate :76.9% Failure Rate :23.1%

Dashboard –

Low weighted payloads have a better success rate than the heavy weighted payloads





Payload mass vs Success for all sites

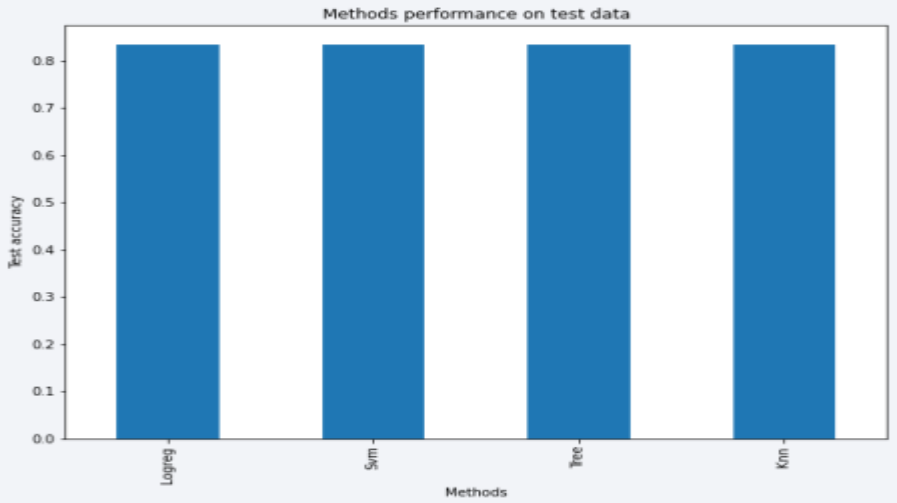
Section 5

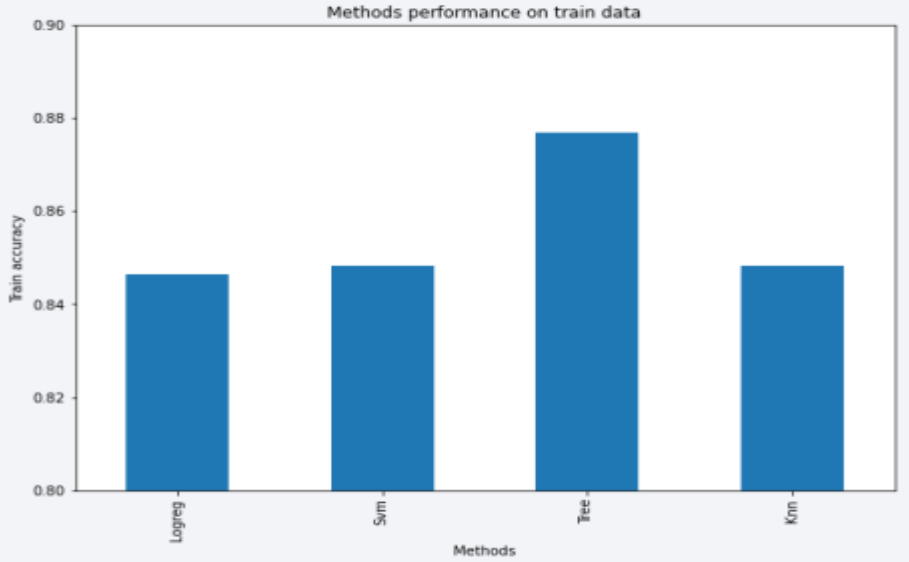
Predictive Analysis (Classification)

Classification Accuracy

Decision tree best parameters

```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```



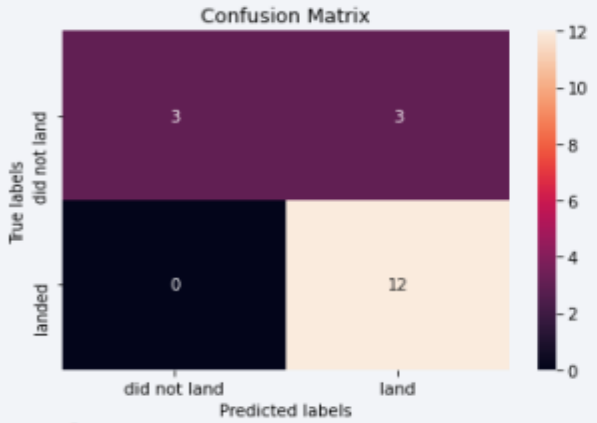


- For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we will take the decision tree.

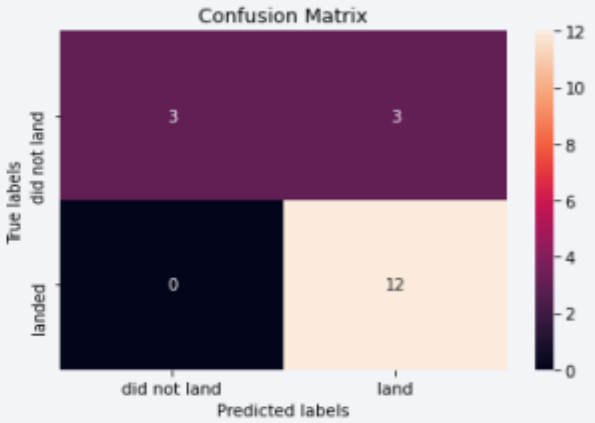
	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

Confusion Matrix

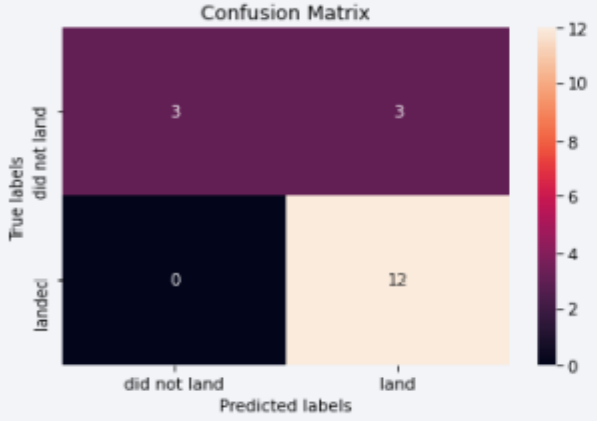
Logistic regression



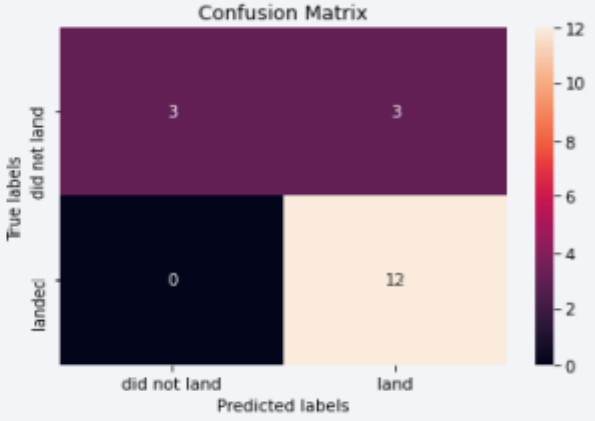
Decision Tree



kNN



SVM



Predicted values	Actual values: 1	Actual values: 0
1	TP	FP
0	FN	TN

As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Appendix

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

  Github Project link : vermanikher/Capstone-Project (github.com)

Thank you!