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

- Data collection using SpaceX REST API
- Data collection using web scraping
- Data wrangling
- Exploratory data analysis (EDA) with SQL
- EDA with data visualization
- Interactive map obtained with Folium Python library
- Dashboard built with Plotly Dash
- Building and training of the ML prediction model

Summary of methodologies

- Description of the findings obtained from the EDA
- Presentation of the interactive analytics results in screenshots
- Presentation of the prediction analysis results

Introduction

- **Project background and context**

SpaceX is the most successful company in rocket launches since it often can reuse the first stage of rockets. For this project we have to imagine to be a team of data scientists who work for a space launch company which wants to compete with SpaceX.

To do that, we have to gather information about SpaceX and to create dashboards in order to get useful insights from data for problems resolution.

- **Problems you want to find answers**

The problems to be solved in this project are:

- Finding data correlations and insights in order to individuate all factors that can influence the landing outcome
- Predicting if the first stage of rockets launched by the SpaceX Falcon 9 will land successfully or not through a Machine Learning (ML) approach

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology

Data were collected from SpaceX API through the “requests” library and with web scraping from Wikipedia.

- Perform data wrangling

Data were cleaned dealing with the missing values, categorical variables were identified and then transformed in binary ones by means of the one-hot encoding technique

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

All classification models were built with the Scikit-learn Python library, with which models were trained, their hyperparameters were tuned and finally their accuracy on test data was evaluated.

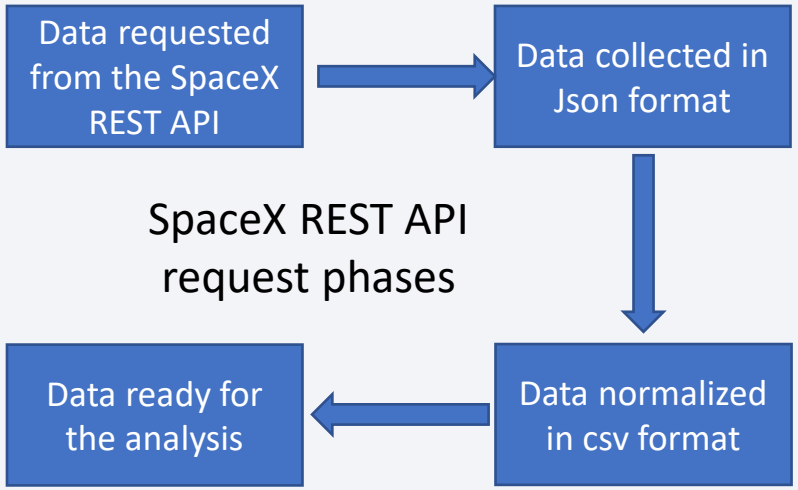
Data Collection

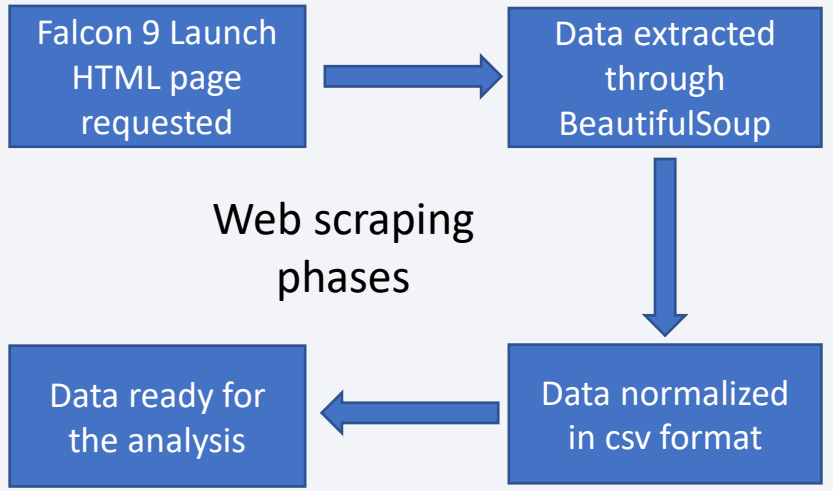
SpaceX launch data were taken from the SpaceX REST API through the «requests» library: these data contain information about rocket launches, like the model of the rocket, the payload, the launchpad the type of the landing and the landing outcome.

After being requested, data were decoded as a Json and then were turned into a Pandas dataframe for the analysis.

With BeautifulSoup method it is possible to perform a web scraping procedure that allows to collect other information about Falcon 9 Launch data.

Here below there are two schemes explaining the phases of data collection.





Data Collection – SpaceX API

On the right side, you can see a summary of data collection from SpaceX REST API. The main phases are:

1. Getting the SpaceX REST API response
2. Conversion of the response into a Pandas dataframe
3. Dropping of Falcon 1 data
4. Dealing with missing values

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

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

```
# Use json_normalize method to convert the json result into a dataframe
response = response.json()
data = pd.json_normalize(response)
```

```
# Hint data['BoosterVersion']!='Falcon 1'
df.drop(df.index[(df['BoosterVersion']=='Falcon 1')], axis=0, inplace=True)
data_falcon9 = df
data_falcon9.head()
```

```
# Calculate the mean value of PayloadMass column
data_falcon9['PayloadMass'].mean()
```

```
6123.547647058824
```

```
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, data_falcon9['PayloadMass'].mean())
```

```
data_falcon9['PayloadMass'].isnull().sum()
```

```
0
```

GITHUB LINK:

https://github.com/Ale-Paul92/Capstone/blob/main/data-collection-API.ipynb

Data Collection - Scraping

The main phases of web scraping are:

1. Requesting the HTML page
2. Creating a BeautifulSoup object
3. Finding all tables
4. Getting all column names
5. Creating a dictionary and adding tables data to it (to view the complete cell, see TASK 3 of the related notebook)
6. Converting the dictionary into a dataframe

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
data  = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number) #TODO-1
            #print(flight_number)
            datatimelist=date_time(row[0])
```

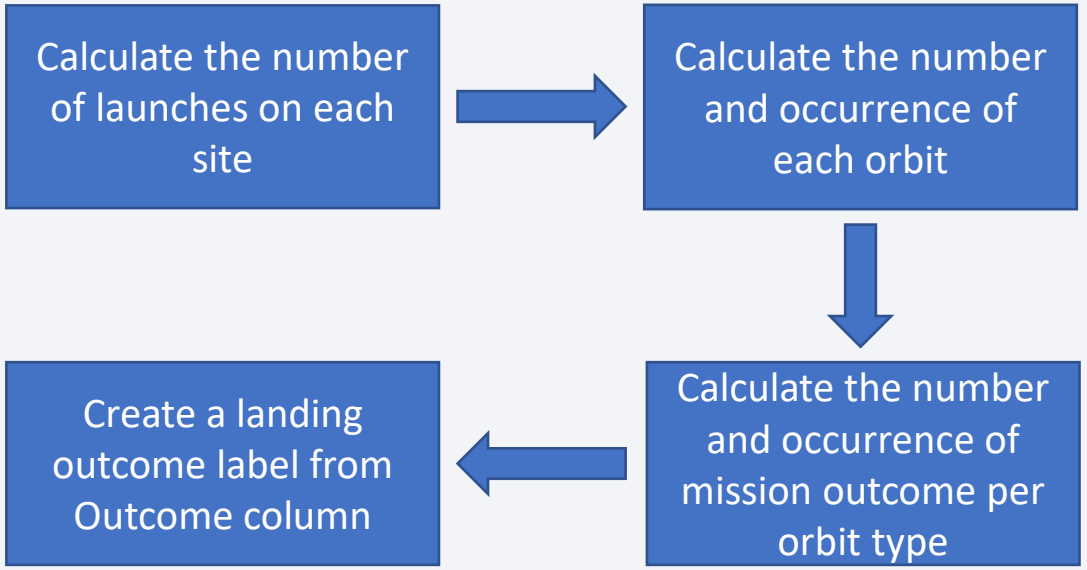
```
df=pd.DataFrame(launch_dict)
```

GITHUB LINK:

https://github.com/Ale-Paul92/Capstone/blob/main/Data_collection_with_web_scraping.ipynb

Data Wrangling

Data were processed in order to obtain a dataset that will then be used for the exploratory data analysis and for the unsupervised models training. At the end of data wrangling process, a label has been added to the dataframe to identify successful landings (the label will show "1") and failed ones (in this case the label will show "0").



GITHUB LINK:

https://github.com/Ale-Paul92/Capstone/blob/main/Data_wrangling.ipynb

EDA with Data Visualization

For exploratory data analysis, data visualization was performed by plotting the following charts:

- Scatter plot: Flight Number vs Payload Mass, Flight Number vs Launch Site, Payload Mass vs Launch Site, Flight Number vs Orbit type and Payload Mass vs Orbit type
- Bar chart: Success rate of each orbit
- Line plot: Success rate for each year since 2010 until 2020

We used these charts in order to verify the relationship among dataframe variables and try to understand if there is one of them that can influence landings outcome.

GITHUB LINK:

https://github.com/Ale-Paul92/Capstone/blob/main/Exploratory_data_analysis_with_visualization.ipynb

EDA with SQL

In this project we have used also SQL language to explore and analyze data. To do that, we used the following queries:

- 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 the 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 kg
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass using a subquery
- Listing the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015
- Ranking the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GITHUB LINK: https://github.com/Ale-Paul92/Capstone/blob/main/Exploratory_data_analysis_with_SQL.ipynb

Build an Interactive Map with Folium

We used the Python library Folium to create an interactive map in which we added the following objects:

- A circle marker for every launch site with the corresponding name as visible in the label
- A marker for every launch on the base of the landing outcome; it is green if the landing is successful, otherwise it is red.
- A polyline to track the distance between a launch site and another location, like the nearest coastline, highway, railway and city. Then distances have also been calculated.

We added these objects in order to point out on the map launch sites location, successful and failed rocket landings and the distance between a launch site and another landmark (coastline, highway etc.)

GITHUB LINK:

https://github.com/Ale-Paul92/Capstone/blob/main/Launch_site_location_Folium.ipynb

Build a Dashboard with Plotly Dash

By means of Plotly Dash we have built a dashboard that permits to choose which data visualize; for example, it is possible to analyze the percentage of successful landings for a specific launch site or to see how landing outcomes vary by selecting a specific Payload mass range.

Charts plotted are:

- A pie chart showing the success rate for a single launch site selected by the user of for all launch sites
- A horizontal line that allows to choose the Payload mass range
- A scatter graph showing the relationship between the payload mass in the range previously specified and the launch outcome for all booster versions.

These plots allowed to find the launch site with the highest launch success rate and the payload mass range which permits to achieve a higher success rate.

GITHUB LINK:

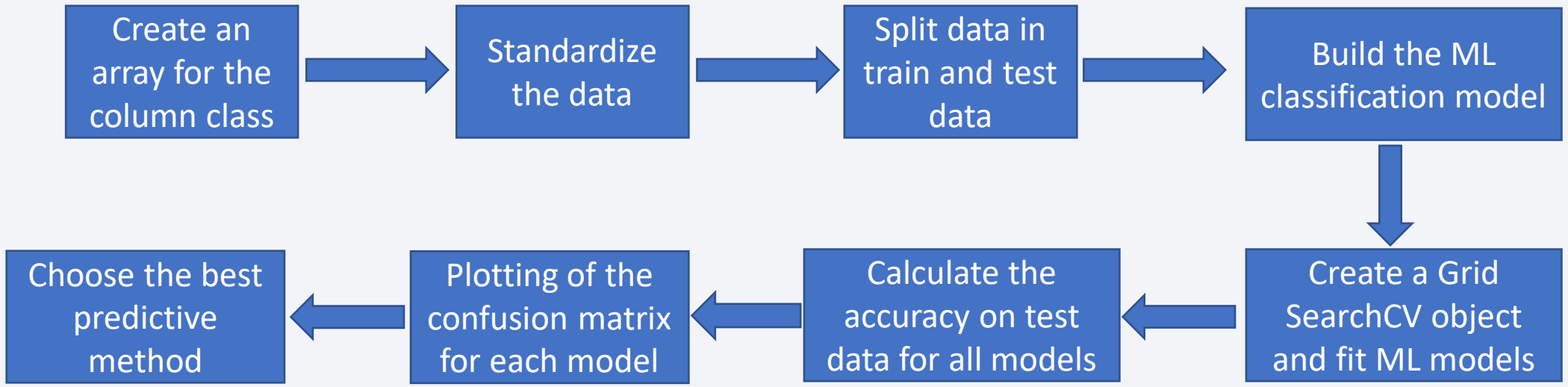
https://github.com/Ale-Paul92/Capstone/blob/main/SpaceX_plotly_dash.ipynb

Predictive Analysis (Classification)

For the predictive analysis, we developed the following machine learning models:

- Decision Tree
- K-nearest neighbors
- Logistic Regression
- Support Vector Machine

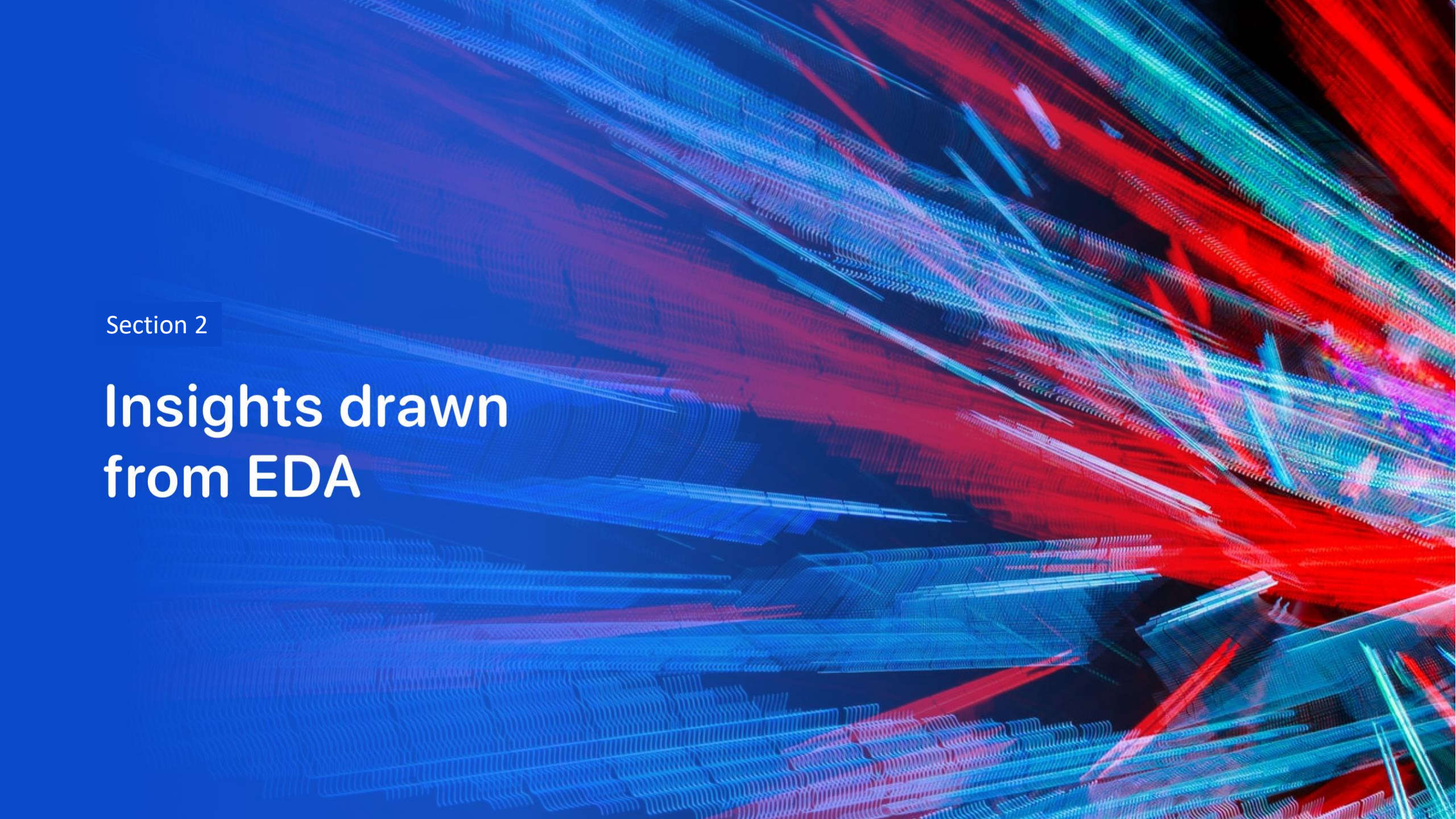
Here below you can see a scheme of the pipeline to find the best prediction model.



GITHUB LINK: https://github.com/Ale-Paul92/Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Models.ipynb

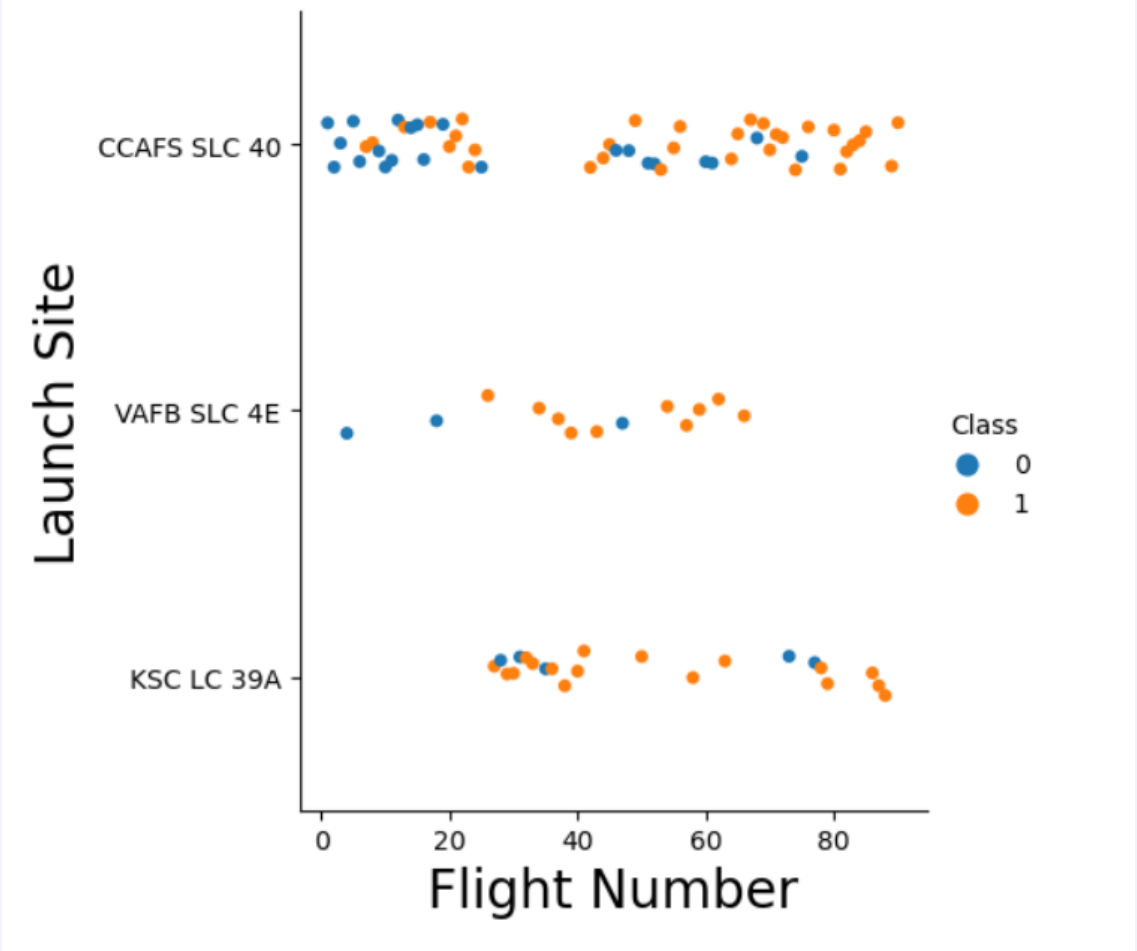
Results

- The SpaceX Falcon 9 launch success rate grows almost every year;
- ES L1, GEO, HEO AND SSO are the orbits with the highest success rate;
- KSC LC-39A is the the launch site with the highest success rate;
- Low weighted payloads ensure a higher success rate than high weighted ones;
- All machine learning models, which were developed in the present work, correctly classify test data with an accuracy of 83.33%.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

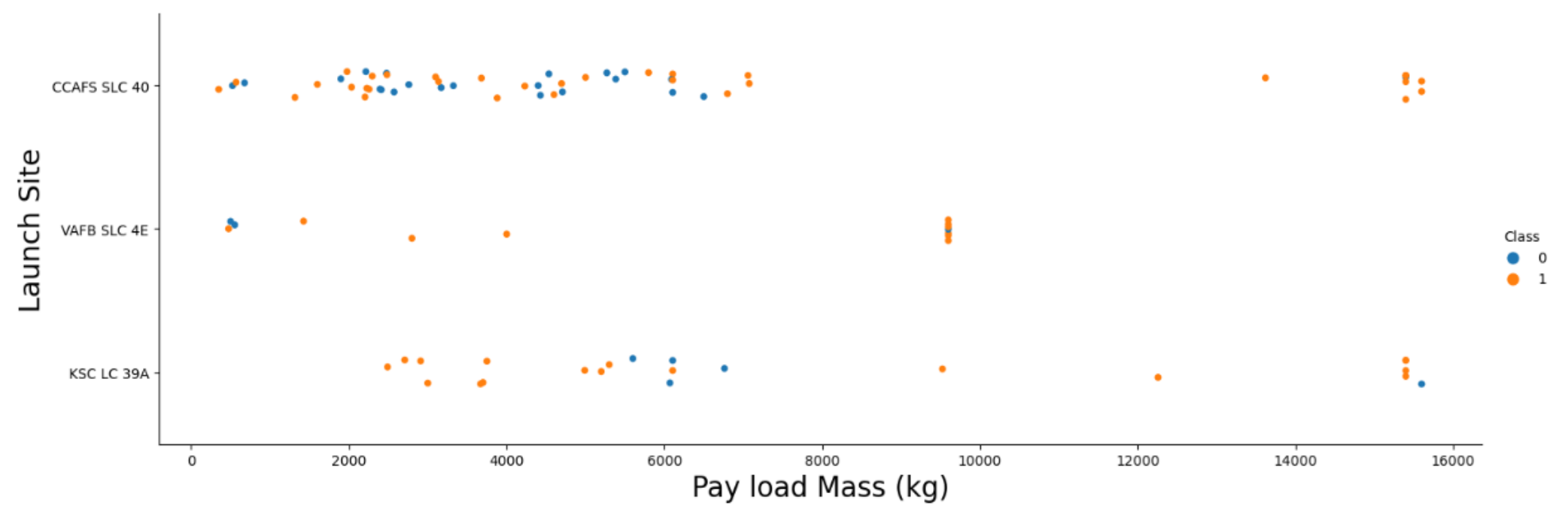
- In the figure on the right, you can see a graph which shows how the success rate varies for each launch site with the variation of the flight number. For every launch site, the success rate increases as the flight number grows up, especially for CCAFS SLC 40 launch site.



Payload vs. Launch Site

By observing the graph below, it is possible to notice that the higher the payload mass, the higher the success rate will be for CCAFS SLC 40 and KSC LC 39A launch sites.

For VAFB SLC 4E there are no launches with pay load mass heavier than 10000 kg, but generally it is not so clear if the success rate strictly depends on pay load mass.



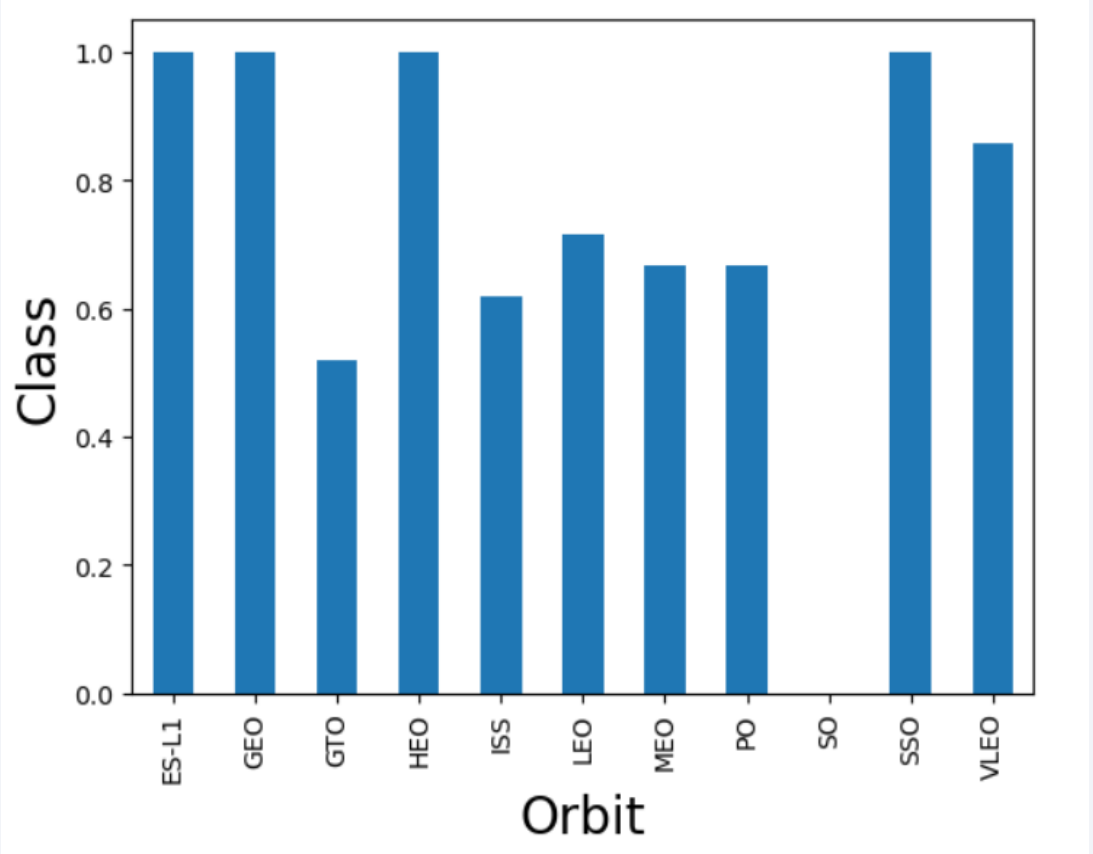
Success Rate vs. Orbit Type

The graph on the right shows the success rate (Class) for every orbit of rocket launches.

It is clear that it is very high for 5 orbits:

- ES-L1
- GEO
- HEO
- SSO
- VLEO

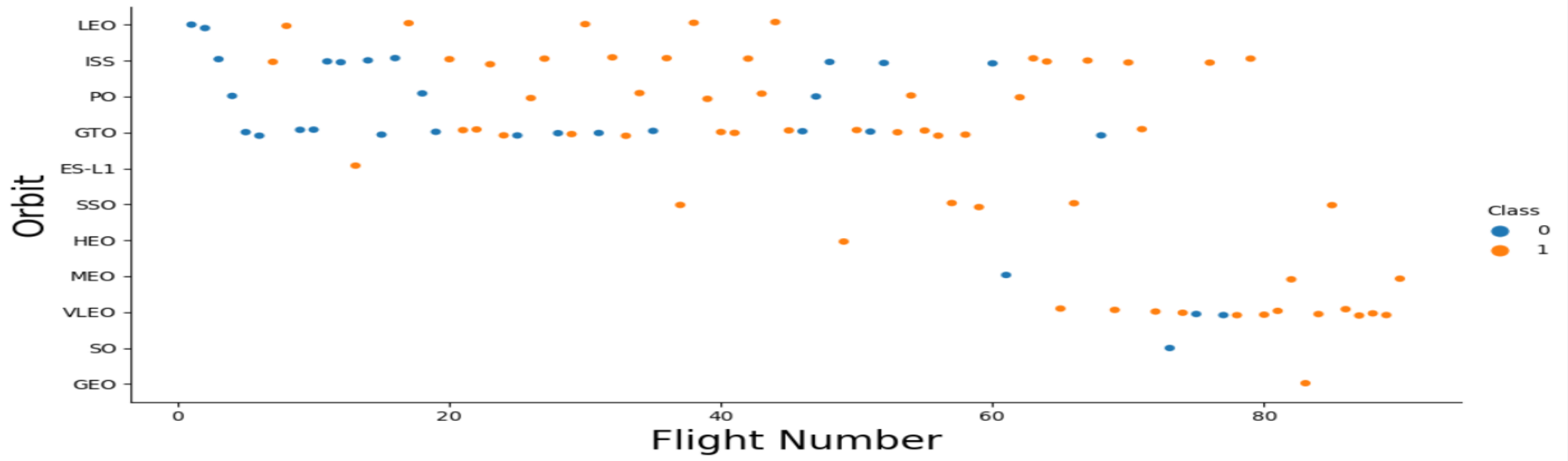
For the first 4 orbits the success rate is maximum.



Flight Number vs. Orbit Type

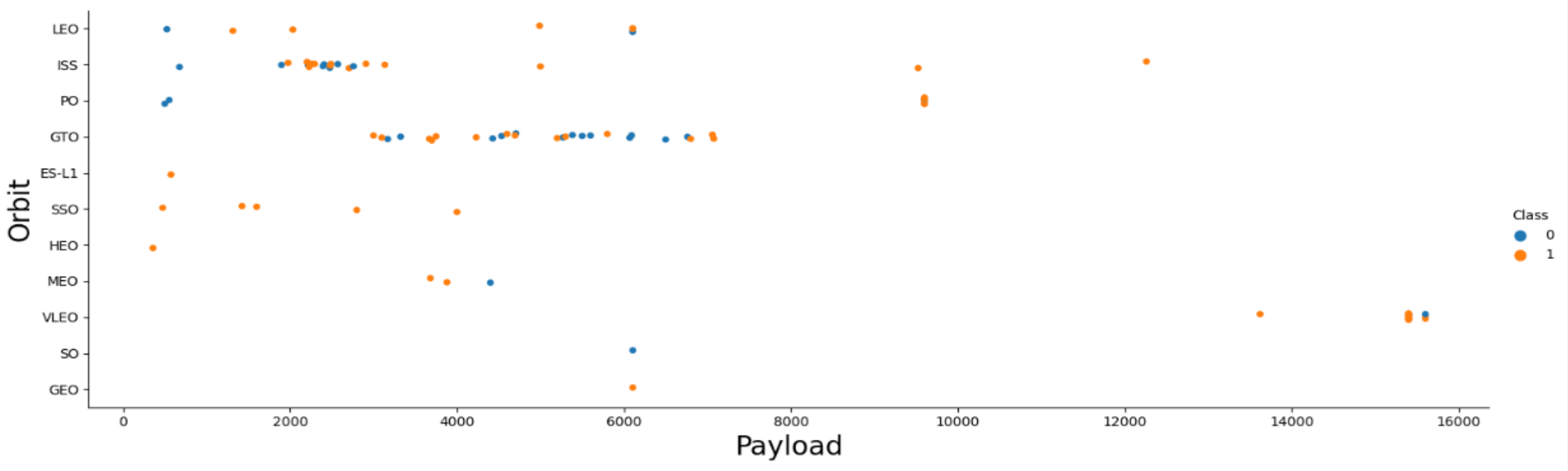
From the graph below, it is clear that for small flight numbers the success rate is very low; by increasing the flight number, the success rate generally grows for most orbits.

Since we have seen in the previous slides that the success rate of rocket landings for launch sites improves with the increasing of flight number, it could be a hypothesis that there is a correlation between success rate and the flight number.



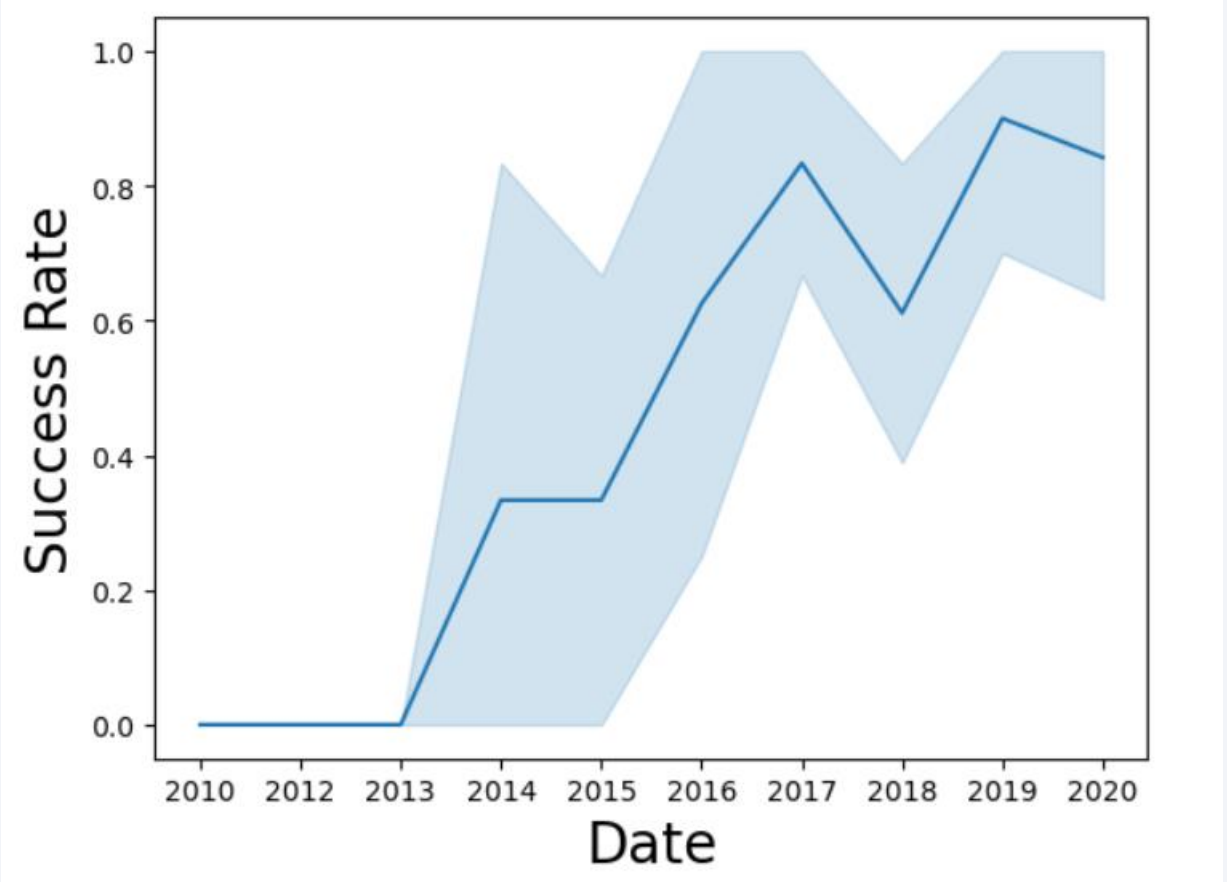
Payload vs. Orbit Type

From the graph below we can see immediately that for GTO success rate doesn't depend on the payload mass. But it is the same also for most of the other orbits, except LEO and ISS where the success rate increases with the growing of payload mass.



Launch Success Yearly Trend

In the figure on the right, it is possible to see that the success rate increases almost every year since 2013 until 2020.



All Launch Site Names

On the right you can see the selection of the names of all launch sites:

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

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

```
In [8]: %sql SELECT DISTINCT(Launch_Site) FROM SPACEX;
 * sqlite:///bludb.db
Done.
```

Out[8]:

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

Launch Site Names Begin with 'CCA'

Here below, you can see the first 5 rows of the database where “Launch_Site” column contains a name beginning with “CCA”.

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

In [9]:
```
%%sql
SELECT * FROM SPACEX
WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Out[9]:

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

By using a query composed by the aggregate function “SUM” and the WHERE clause, we got the total payload mass by boosters that were launched by NASA.

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

In [10]:
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) as 'Total payload mass of NASA (CRS)' FROM SPACEX
WHERE Customer = 'NASA (CRS)'
```

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

Out[10]:

Total payload mass of NASA (CRS)
45596

Average Payload Mass by F9 v1.1

By using a query composed by the aggregate function “AVG” and the WHERE clause, we got the average payload mass (expressed by kg) carried by boosters version F9 v1.1.

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

In [11]:
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) as 'Average payload mass of Booster_Version F9 V1.1' FROM SPACEX
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

Out[11]:

Average payload mass of Booster_Version F9 V1.1
2928.4

First Successful Ground Landing Date

By means of a query constituted by the aggregate function “MIN” and the WHERE clause, we obtained the date of the first successful landing outcome on ground pad.

In [13]:
```
%%sql
SELECT MIN(DATE) FROM SPACEX
WHERE "Landing _Outcome" = "Success (ground pad)"
```

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

Out[13]:

MIN(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

By selecting the column containing the name of all boosters through the “DISTINCT” clause and the double “WHERE” one, we got the name of boosters that have a successful landing in drone ship with a payload mass between 4000 and 6000 kg.

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

In [14]:
```
%%sql
SELECT DISTINCT(Booster_Version) FROM SPACEX
WHERE "Landing _Outcome" = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000);
```

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

Out[14]:

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

Total Number of Successful and Failure Mission Outcomes

By selecting the column containing the outcome of each mission and the total number of successful and failed landings together with the “COUNT” and “GROUP” clauses, we got the small table shown in the picture on the right.

List the total number of successful and failure mission outcomes

In [15]:
```
%%sql

SELECT MISSION_OUTCOME, COUNT(*) as "TOTAL NUMBER" FROM SPACEX
GROUP BY MISSION_OUTCOME;
```

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

Out[15]:

Mission_Outcome	TOTAL NUMBER
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

On the right, you can see a table that was obtained by selecting the “booster_version” and “payload mass kg” columns with the help of a subquery (second line of the code).

The table shows the boosters with the maximum payload mass expressed in kg.

In [17]:
```
%%sql
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEX
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

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

Out[17]:

Booster_Version	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

2015 Launch Records

With the substr keyword and the double “WHERE” clause (with another substr keyword) it was possible to obtain the table in the figure below where the failed landing outcomes are shown.

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

In [18]:
```
%%sql
SELECT substr(Date, 4, 2) as "Month", "Landing _Outcome", Booster_Version, Launch_Site from SPACEX
WHERE "Landing _Outcome" = 'Failure (drone ship)' AND substr(Date,7,4) = '2015';
```

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

Out[18]:

Month	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

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

By selecting the “landing _Outcome” column and using the “COUNT”, “WHERE”, “BETWEEN”, “GROUP BY”, “ORDER BY” and “DESC” clauses, the table in the figure below was obtained: in this you can see the count of landing outcomes that occurred between 04-06-2010 and 20-03-2017.

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

In [19]:
```
%%sql
SELECT "Landing _Outcome", COUNT(*) as "Count" from SPACEX
WHERE "Landing _Outcome" LIKE "Success%" AND "Date" BETWEEN "04-06-2010" AND "20-03-2017"
GROUP BY "Landing _Outcome" ORDER BY "Count" DESC;
```

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

Out[19]:

Landing _Outcome	Count
Success	20
Success (drone ship)	8
Success (ground pad)	6

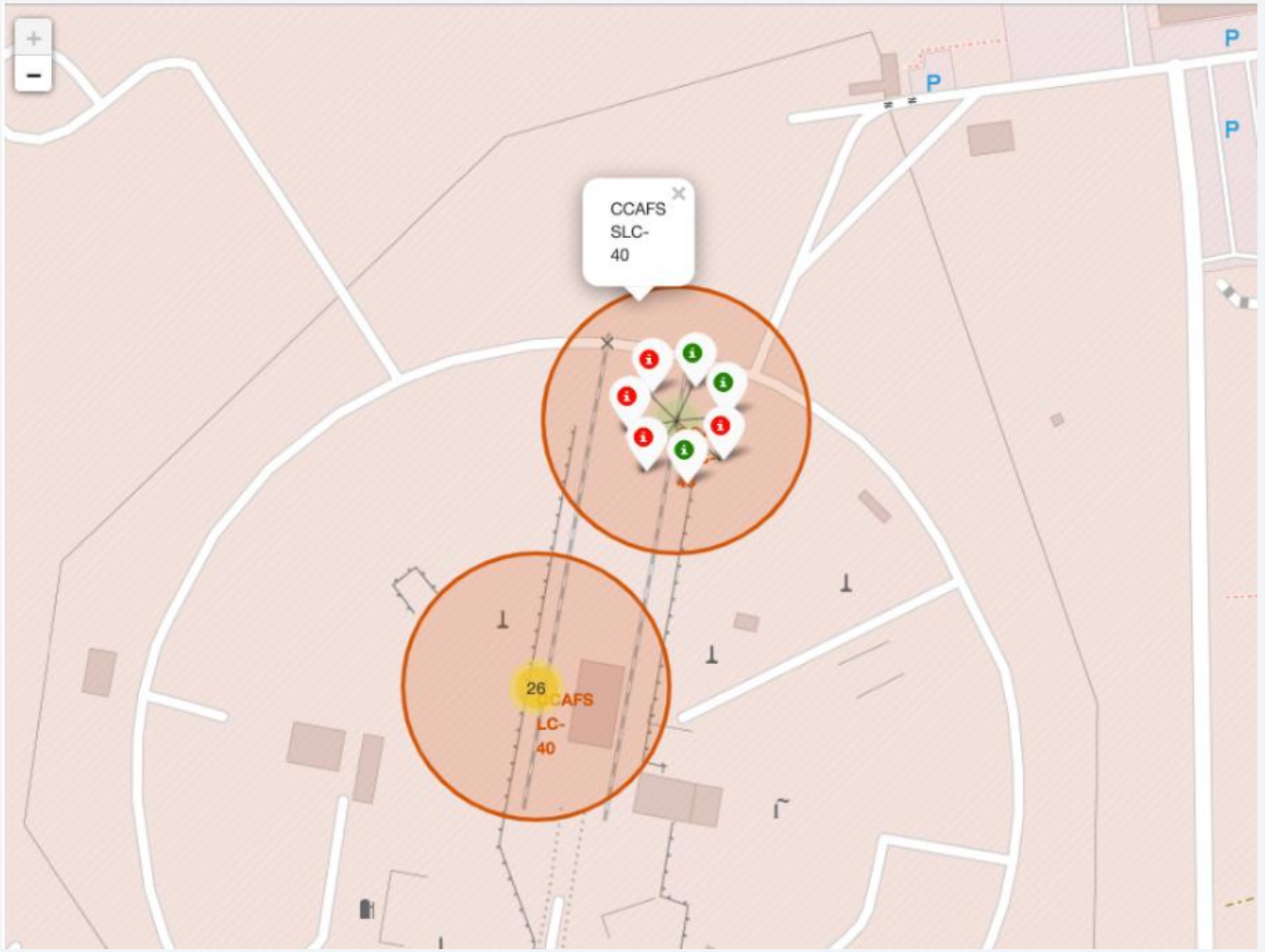

Section 3
Launch Sites
Proximities Analysis

Location of the launch sites on the map

In the picture below, you can see the localization of all launch sites, one on the left side and 3 (almost overlapped) on the right one.

Launches outcome reported on the map

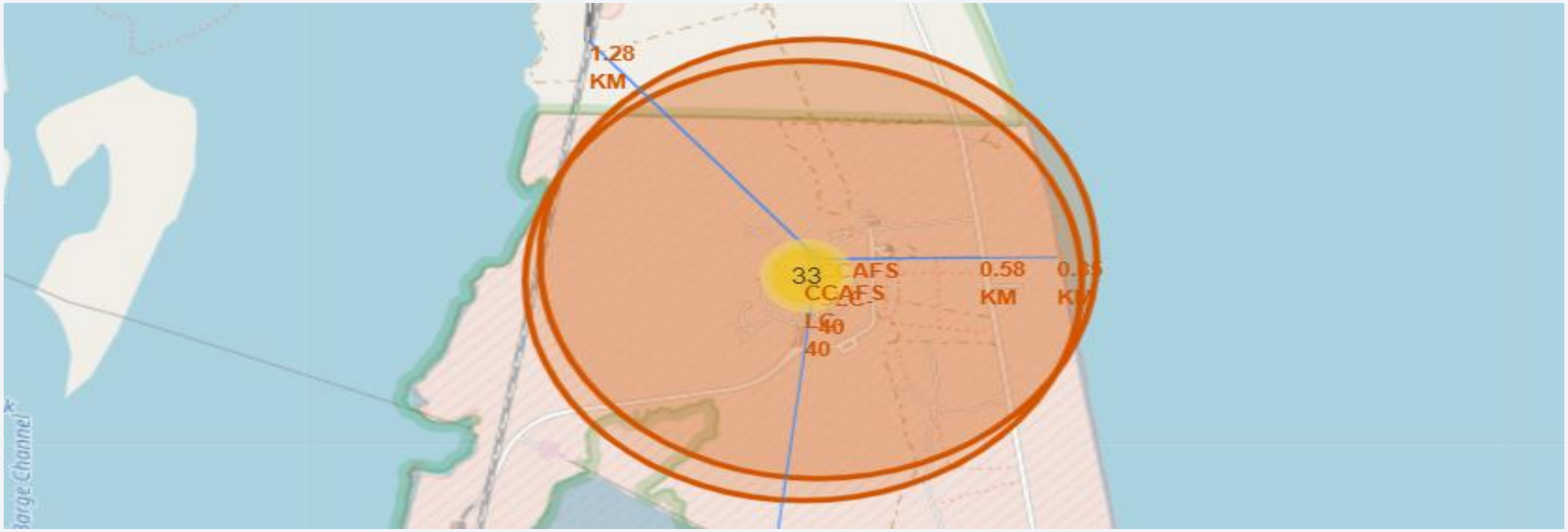
In the figure on the right you can see the zoomed Folium map showing launch outcomes which are labeled with red or green markers: the red ones represent the failed launches, while the green ones indicate successful launches.

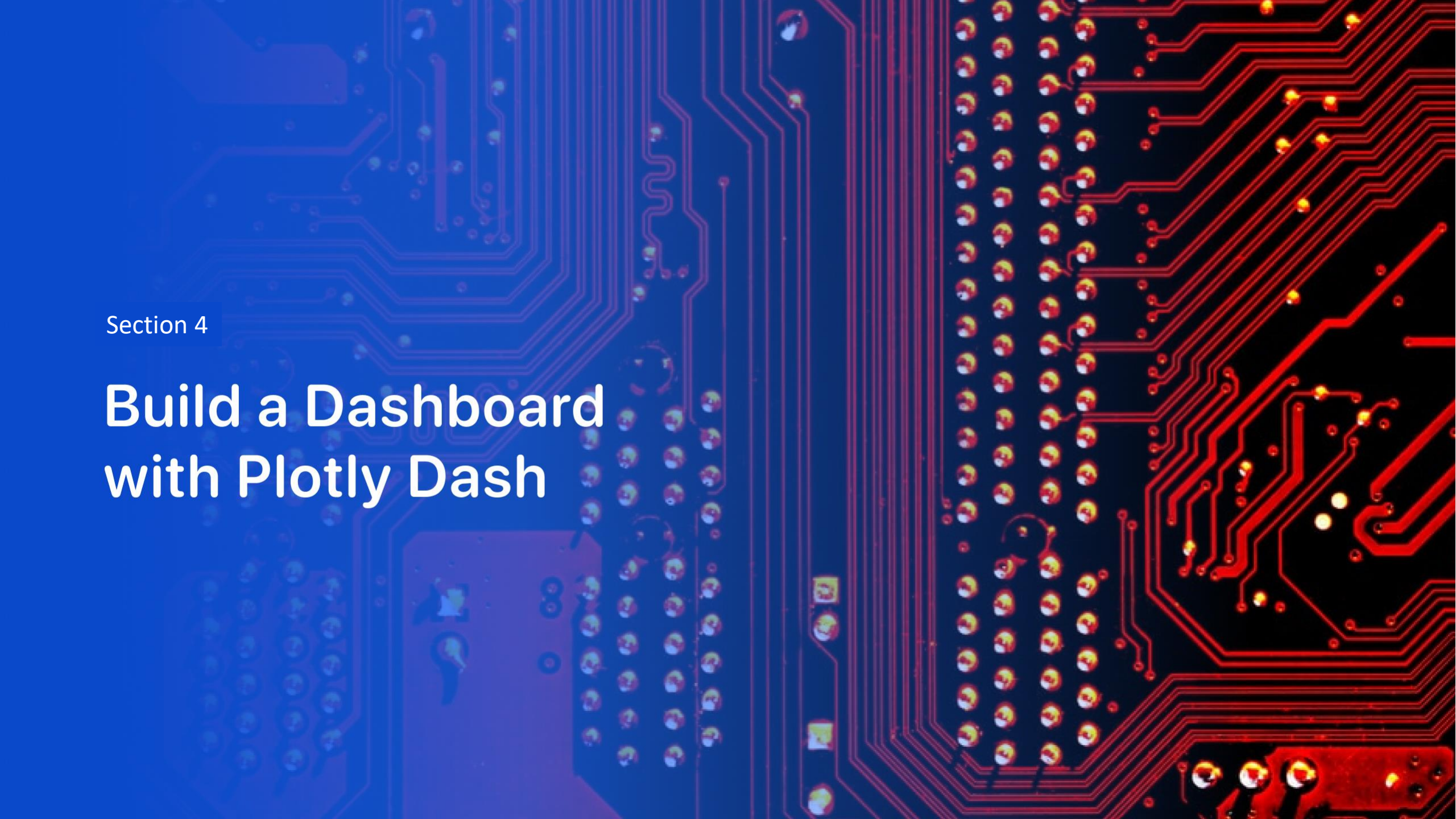


Launch site distance from the closest city, railway and highway

In the figure below, you can see the CCAFS SLC-40 launch site with lines connecting it to the closest railway (on the left side), highway (on the right side) and city (that is the down line).

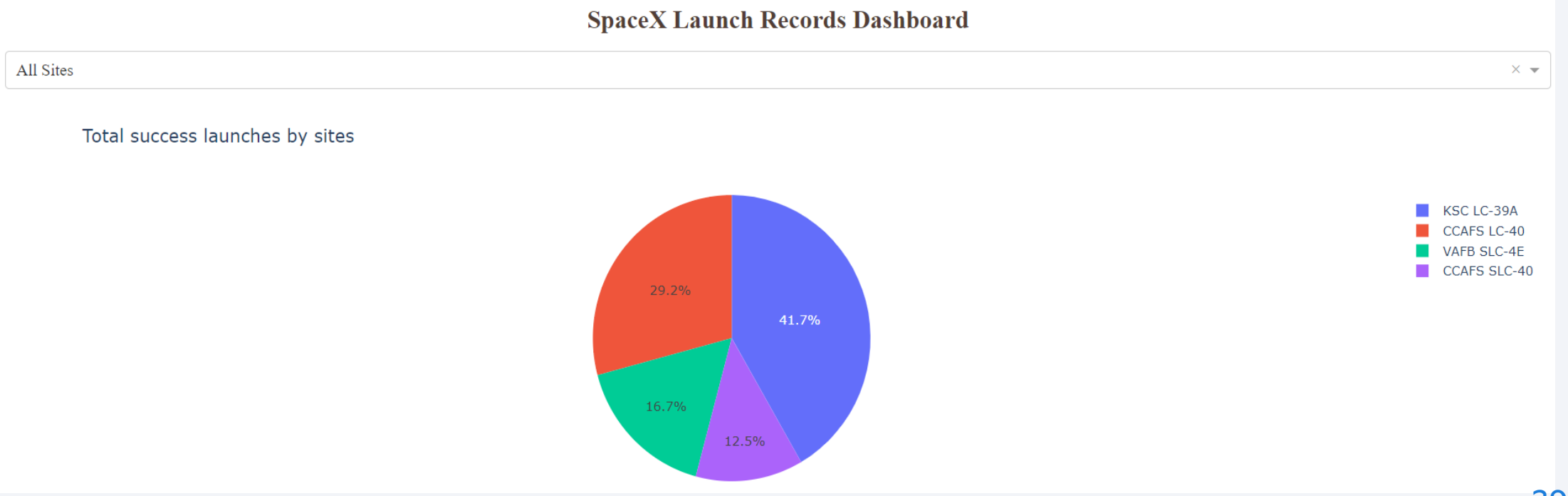
Please note that for what concerns the line on the right side, there are two distance values: 0,58 km which indicates the distance from the nearest highway (what I was looking for) and 0,85 km that is indicative of the distance of the launch site from the closest coastline.




Section 4
Build a Dashboard with Plotly Dash

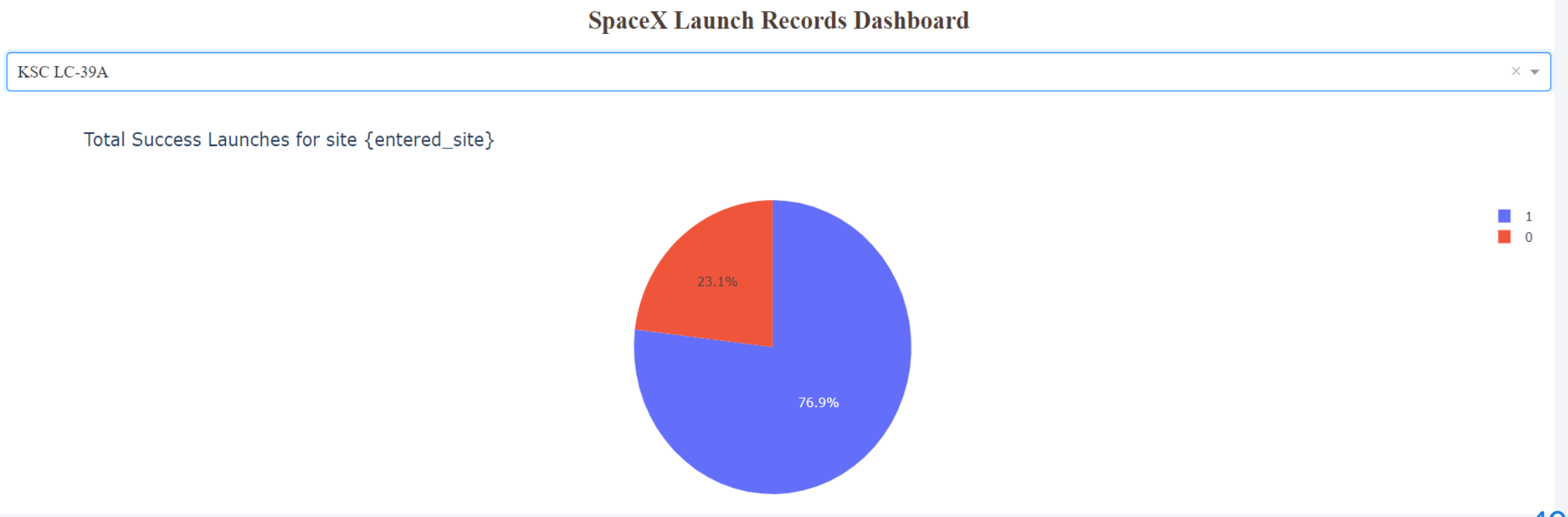
Graph with success launches for all sites

In the figure below a pie chart is reported: this graph shows the percentage values of successful landings for each launch site. It is possible to notice immediately that the launch site with the highest landing success percentage is KSC LC-39A.



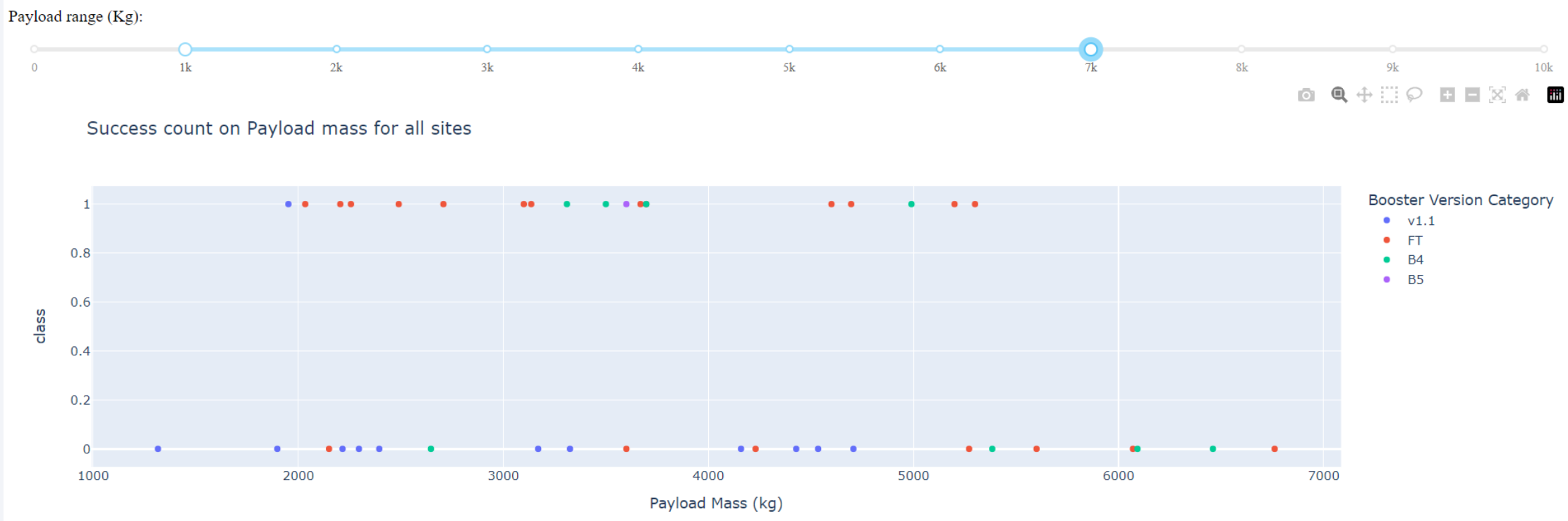
Launch site with the highest success rate

The pie chart below shows the percentage value of successful landings (in blue) and of failed ones (in red) of the launch site having the highest landing success rate, that is KSC LC-39A.



How success rate varies with payload for launch sites

In the figure below, you can see the Payload Mass vs Launch Outcome scatter plot. Through the horizontal line above the graph, it is possible to choose the Payload range (in this case between 1000 and 7000 kg) and analyze the landing outcomes. For example it is evident that FT booster version has a high success rate, above all with a payload comprised between 2000 and 4000 kg; conversely, the success rate is generally low for v1.1 booster version.



Section 5

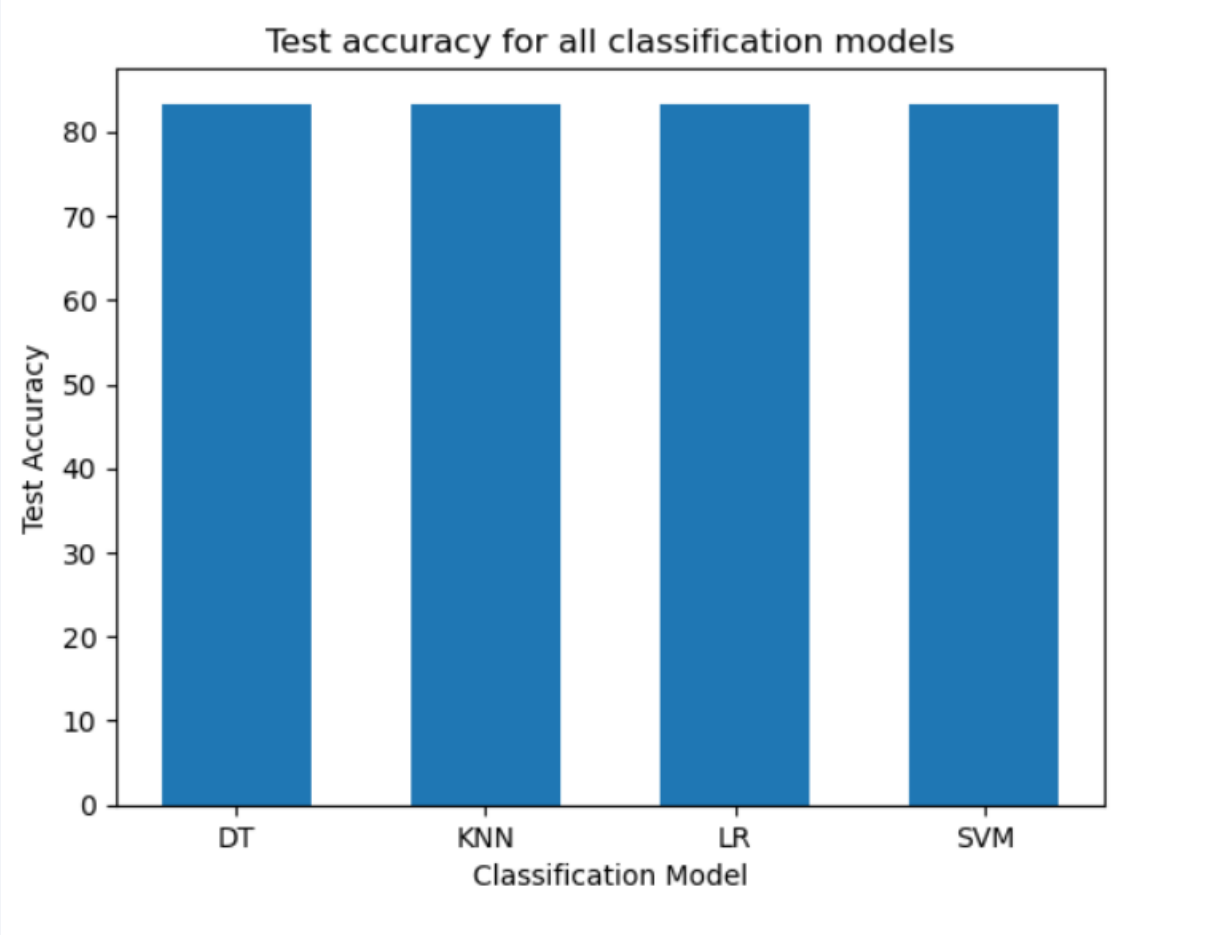
Predictive Analysis (Classification)

Classification Accuracy

We developed 4 machine learning models for the prediction of landing outcomes:

- Decision tree
- K-nearest neighbors
- Linear regression
- Support vector machine

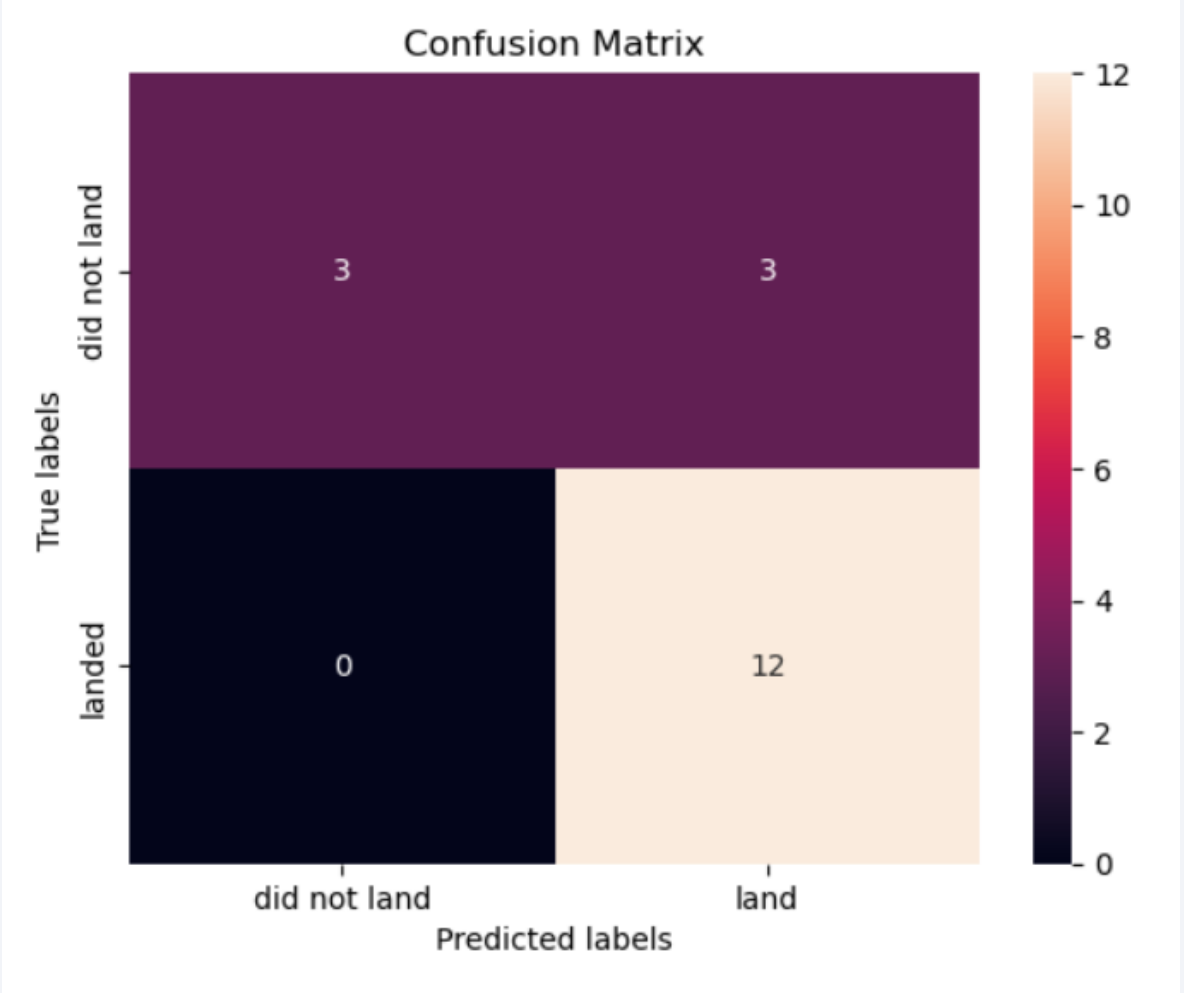
From the bar chart on the right, we can see that all models exactly have the same test accuracy, that is equal to 83.33%.



Confusion Matrix

Test accuracy is equal for all models, so the confusion matrix is the same for each one.

By observing the confusion matrix on the right part of the slide, we can see that all successful launches are correctly classified, while 3 of the 6 failed ones are wrongly labeled (i.e. as successful).



Conclusions

By considering the entire data analysis performed in this work, we can conclude that:

- The success rate, since 2013, is almost continuously growing;
- ES L1, GEO, HEO AND SSO are the orbits with the highest success rate;
- KSC LC-39A is the launch site with the highest success rate;
- Low weighted payloads (comprised between 2000 and 4000 kg) are associated to a higher success rate with respect to high weighted payloads;
- All machine learning models, which were developed in the present work, correctly classify test data with an accuracy of 83.33%.

Thank you!