Winning Space Race with Data Science

Mariano Salcedo
October 4th 2021

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

Summary of methodologies

- Data Collection
- EDA with Data Visualization
- EDA with SQL
- Interactive maps with Folium
- Dashboards with Plotly
- Predictive analysis

Summary of all results

- Preliminary analysis based on EDA
- Interactive maps and dashboards
- Predictive results

Introduction

Project background and context

- The aim is to predict if the Falcon 9 first stage will succesfully land on its ground base after being launched. The core of Falcon project is to reuse this first stage propulsion system, which is then traduced into a strong cost saving. By determining the chances of a certain mission to be succesfully (=.to land safely) we can estimate the cost of the missions.

Problems you want to find answers

- The main variables that participate into the Falcon 9 first stage success/fail.
- How does the main variables interact

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- SpaceX rest API
- Web scrapping from sites like Wikipedia

Perform data wrangling

- One hot encoding for Machine Learning algorithms

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

- The following datasets were collected from Rest SpaceX API:
  - Rockets - to learn the booster name
  - Launchpads - to know the name of the launch site being used, the logitude, and the latitude
  - Payloads - to learn the mass of the payload and the orbit that it is going to
  - Cores - to learn the outcome of the landing, the type of the landing, number of flights with that core, etc

Data Collection – SpaceX API

- Link to lab:
  [Lab 1 - Data Collection]

- Get and append data from API

```
response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
BoosterVersion.append(response['name'])
```

- Convert json into a dataframe

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

- Analyze/clean the data from different sources and create a dictionary to make a unique dataframe

```
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}
```

Data Wrangling

- Introduction

  In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

- Process

  1. Perform exploratory Data Analysis
  2. Calculate the number of launches on each site
  3. Calculate the number and occurrence of each orbit
  4. Calculate the number and occurence of mission outcome per orbit type
  5. Create a landing outcome label from Outcome column
  6. export it to a CSV for the next section

- Link to lab: Lab_2_data_Wrangling

EDA with Data Visualization

- Charts used:
  - Scatter plots to visualize relationship between variables
    - FlightNumber vs. PayloadMass
    - FlightNumber vs LaunchSite
    - Payload and Launch Site
    - FlightNumber and Orbit type
    - Payload and Orbit type
  - Bar plots to compare metrics from different variables
    - Success rate of each orbit type
- Link to lab: [Lab_3_EDA_Visualization]

EDA with SQL

- Performed SQL queries:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000¶
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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
- Link to lab: [Lab 4 - EDA with SQL]

Build an Interactive Map with Folium

- Objects created and added to Folium map:
  - A folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas
  - A blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name
  - A circle for each launch site in data frame launch_sites
  - For each launch site, added a Circle object based on its coordinate (Lat, Long) values
  - Markers for all launch records
  - Marker clusters to group points with the same coordinates but different information
  - Calculate the distances between a launch site to its proximities and plot distance and a line between points
- All the objects were added in order to improve the understanding of the problem, by localizing all ground stations and visualize the number of launches in each one and its outcome.
- Link to lab: [Lab 5 - Visual analytics]

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions added to a dashboard:
  - Plot charts:
    - Showing the total launches by a certain site or all sites
    - Displaying the relative proportions of launches
    - If a single site was selected, the pie chart represents the proportion between success and fails.
  - Scatter charts:
    - Relationship between variables to understand how is the relations between them
    - There was added a slider to filter by Payload mass, in order to filter by different ranges of weight.
- Link to lab: SpaceX dashboard

Predictive Analysis (Classification)

- Build the model:
  - Load dataset into python
  - Transform the data to normalize and therefore improve the performance of machine learning algorithms
  - Split into training and test sets
  - Select between several algorithms to be used
  - Set the ranges of variables used for the selected algorithms and load them into GridSearch for parameter optimization
- Evaluating the model:
  - Accuracy checks were performed to every model
  - Plot confusion matrices for every model
- Improving the model
  - Featuring engineering
- Select the best algorithm to be used for predicting the launch outcome
- Link to lab: Lab 6 - Machine Learning lab

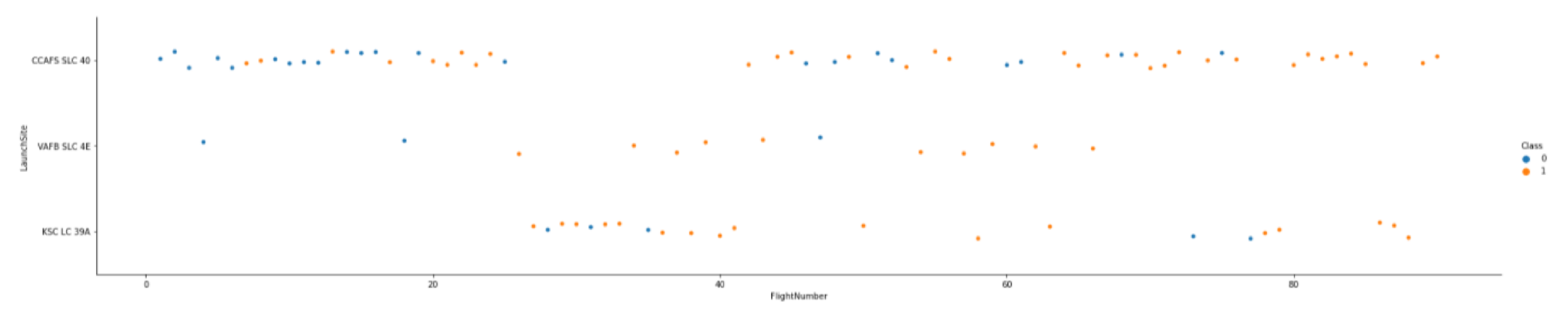
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

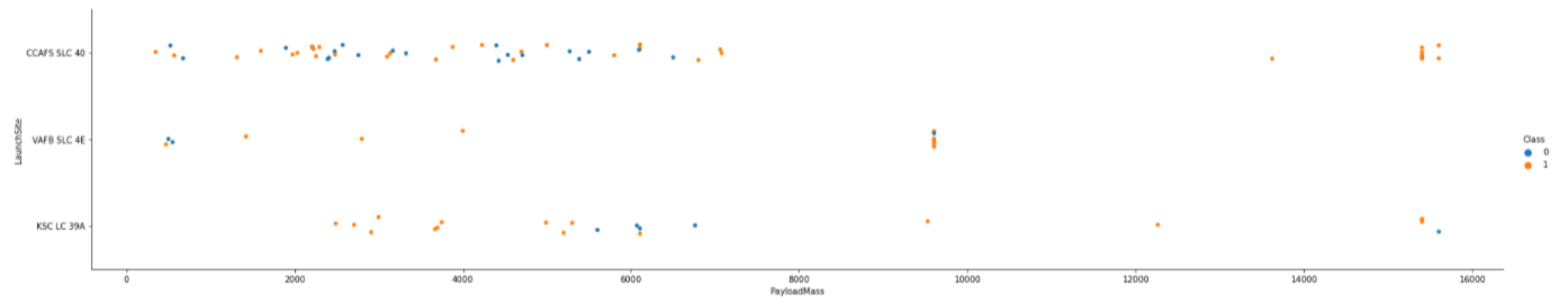
Insights drawn from EDA

Flight Number vs. Launch Site



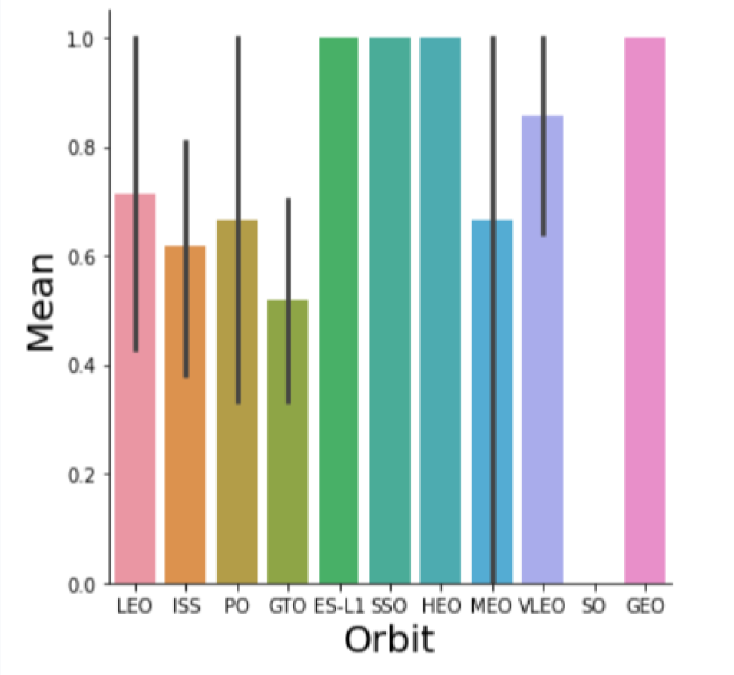
- We can see that for every site, the outcome improves with the amount of launches

Payload vs. Launch Site



- We see from this plot that, with the increasing of payload for site SLC 40, the success rate improves. This can be related to the test missions (lower payload) versus the “real” missions, with higher payload.

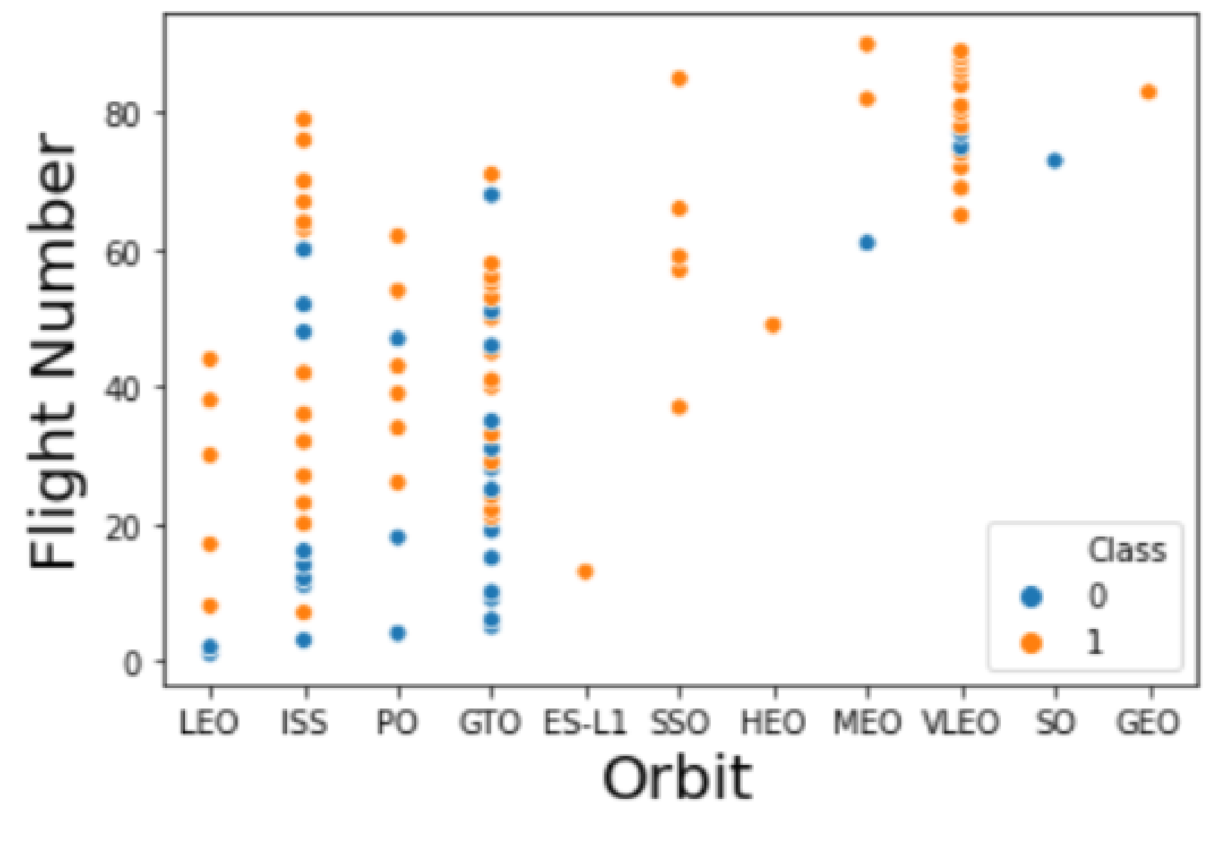
Success Rate vs. Orbit Type



- Clearly, we see that orbits ES-L1, SSO and HEO have the better success rates.

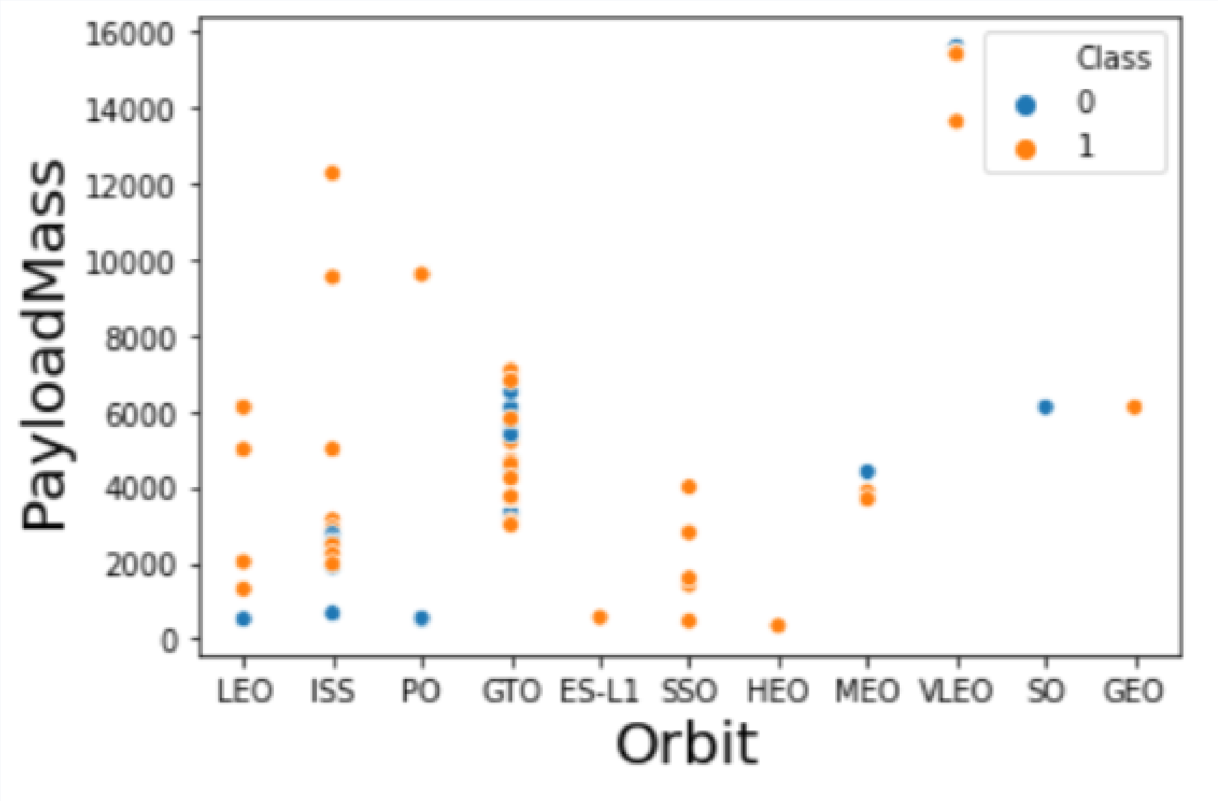
Flight Number vs. Orbit Type

- We see that for some orbits, the success rates improved with the missions, while others started latter and were benefit from learning from previous missions.



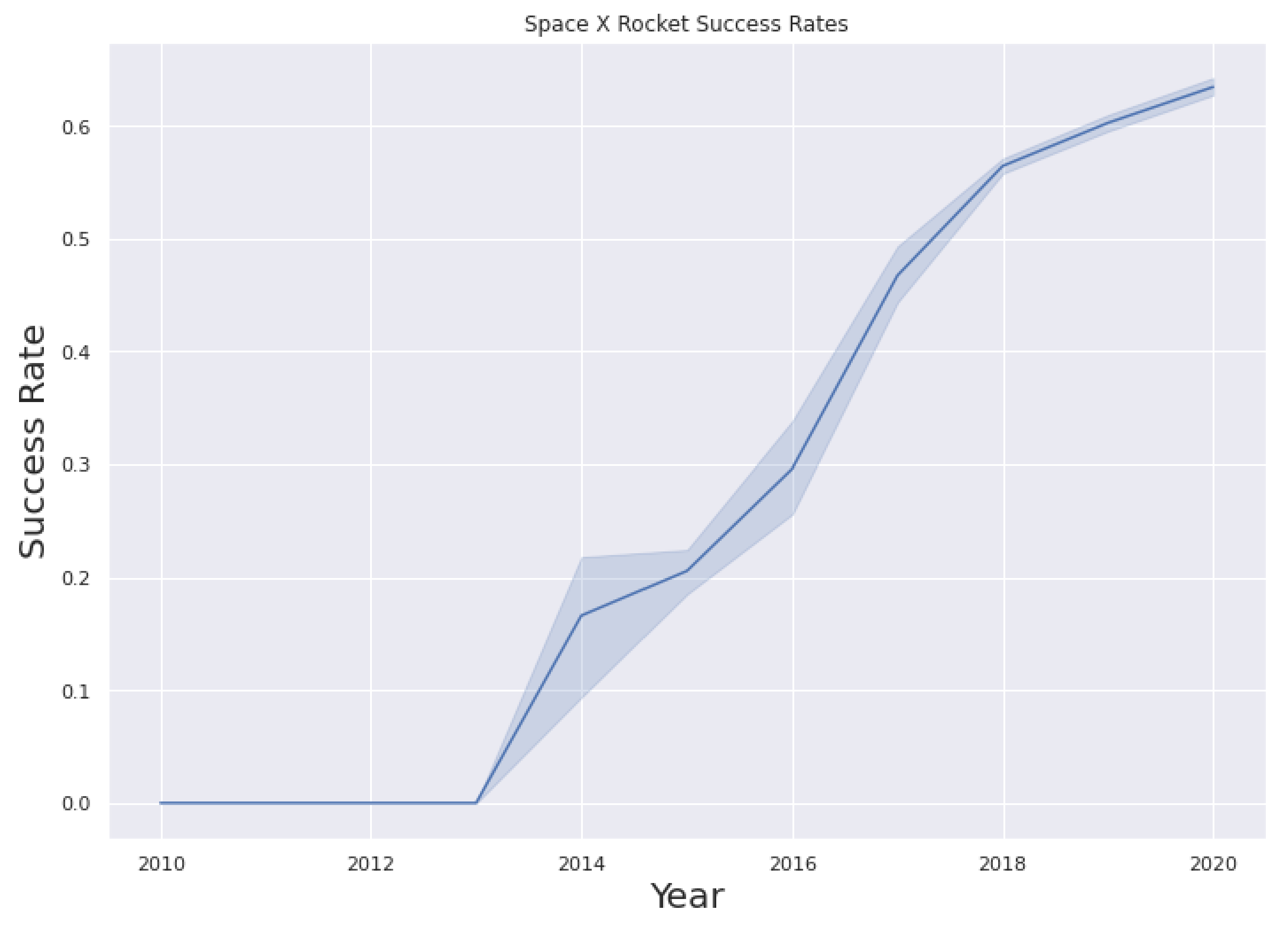
Payload vs. Orbit Type

- Some orbits were used for lower payloads (ie. SSO), others for higher ones (ie. VLEO), and some like ISS were used for a wide set of payloads.



Launch Success Yearly Trend

- We clearly see a tendency of improvement with the years (aka “learning curve”).



All Launch Site Names

- SELECT DISTINCT launch_site from SPACEXTBL
- By using DISTINCT we just keep the different objects of launch_site table

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

Launch Site Names Begin with 'CCA'

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

- SELECT * from SPACEXTBL WHERE launch_site like 'CCA%' LIMIT 5;
- By using the WHERE statement followed by like, we can filter the results to a specific group.

Total Payload Mass

- SELECT SUM(payload_mass__kg_) Payload_total_NASA FROM SPACEXTBL WHERE customer LIKE 'NASA (CRS)’;
- By using SUM within the SELECT we can add all records that matches with the WHERE condition.

payload_total_nasa
45596

Average Payload Mass by F9 v1.1

- SELECT AVG(payload_mass__kg_) Payload_avg_F9v1_1 FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1’;
- By using AVG within the SELECT we can get the average of all records that matches with the WHERE condition.

payload_avg_f9v1_1
2928

First Successful Ground Landing Date

- SELECT min(DATE) Date_Success_GroundPad FROM SPACEXTBL WHERE landing__outcome LIKE 'Success (ground pad)’;
- By using min we can select the oldest date that matches with the WHERE condition. We must select only the success records.

date_success_groundpad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- SELECT DISTINCT booster_version FROM SPACEXTBL WHERE landing__outcome LIKE 'Success (drone ship)' AND payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000;
- We filter the payload to a desired range by using WHERE statement. We also select DISTINCT in order to obtain just a single record for every booster version.

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

Total Number of Successful and Failure Mission Outcomes

- SELECT landing__outcome, COUNT(landing__outcome) as COUNT FROM SPACEXTBL WHERE landing__outcome LIKE 'Success%' OR landing__outcome LIKE 'Failure%' GROUP BY landing__outcome;
- We filter the results by WHERE condition and group them using GROUP BY statement. Then we can count the matches using COUNT.

landing__outcome	COUNT
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

- SELECT DISTINCT booster_version, payload_mass__kg_ FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
- We used a subquery in order to pre filter the dataset. Then we choose distinct values from the previous selection.

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

DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- SELECT DATE, booster_version, launch_site, landing__outcome FROM SPACEXTBL WHERE YEAR(DATE) = 2015 AND landing__outcome LIKE 'Failure%';
- We used YEAR function to get the year for all dates, and then select all that matches with 2015 and the outcome was a fail.

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

- SELECT landing__outcome, COUNT(landing__outcome) COUNT FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER by COUNT DESC;
- We used WHERE statement and group the results. Then by using COUNT we can get the matches for the required condition.

landing__outcome	COUNT
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

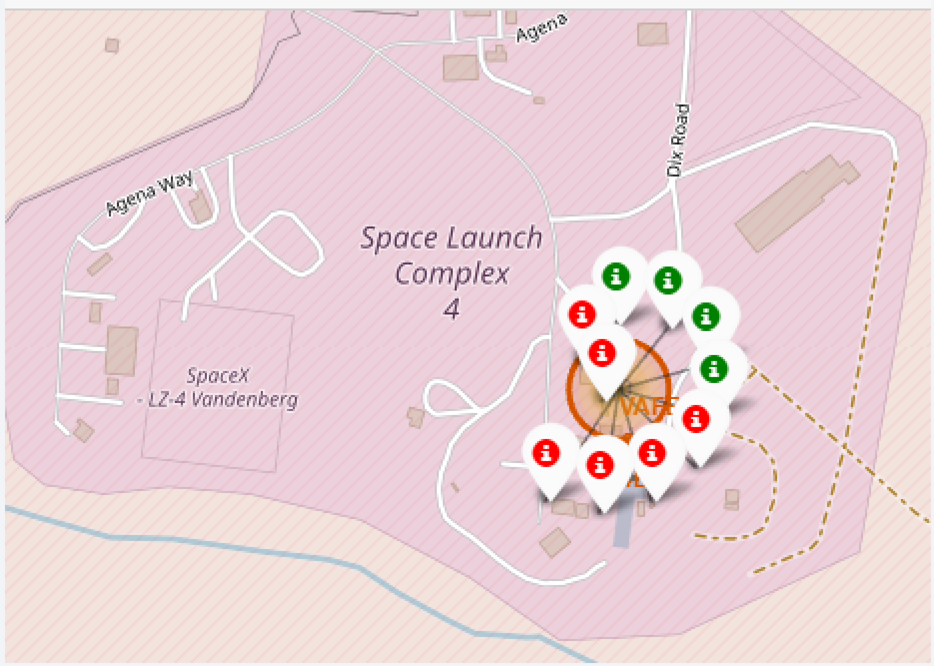
Section 4

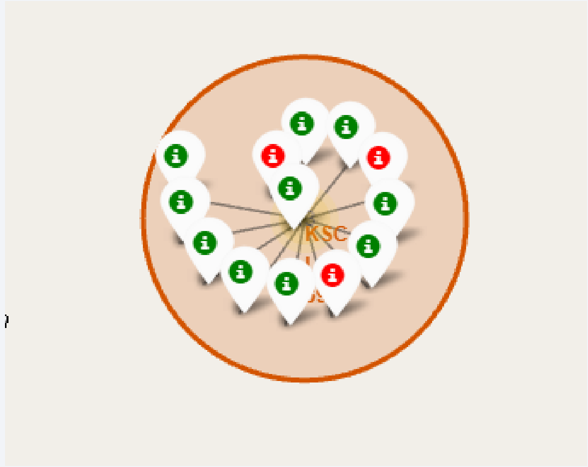
Launch Sites Proximities Analysis

Folium map – Ground Stations

- Here we plot the ground stations locations and the SpaceX missions

Folium map – Missions outcome

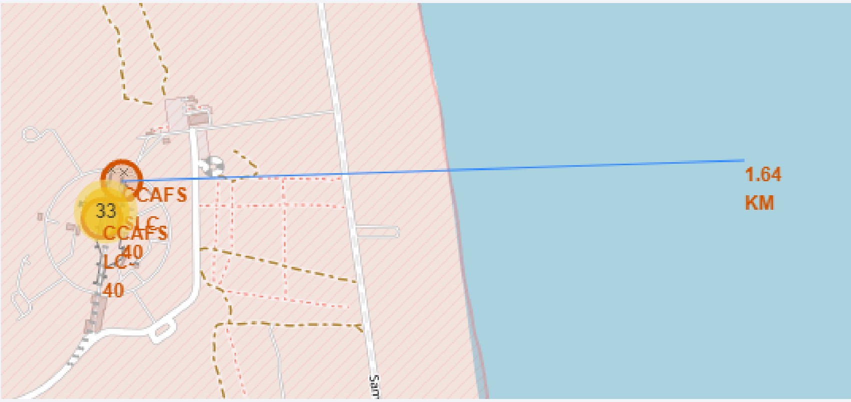


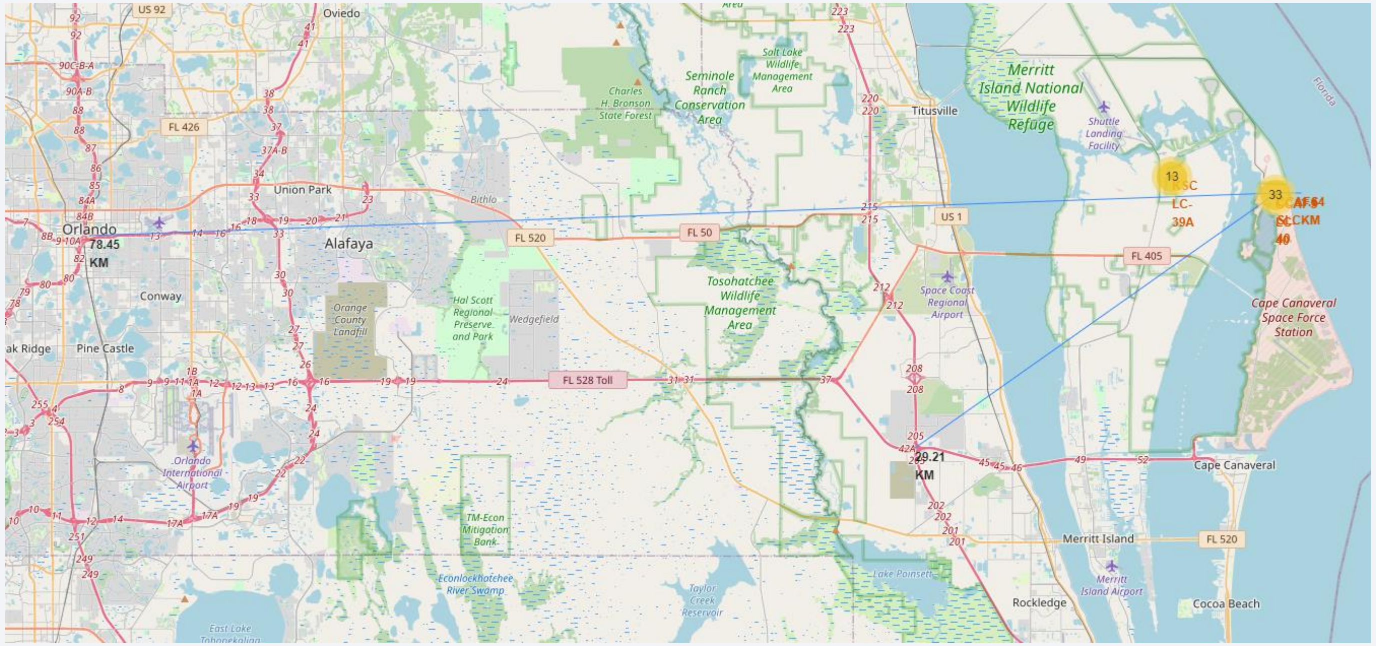


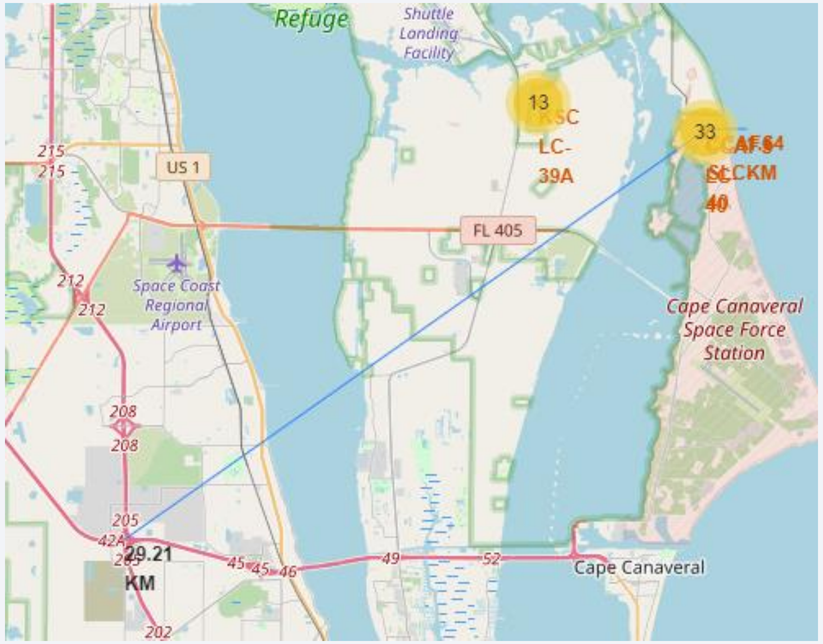
- Here we plot all the missions of every ground station and its outcome, being red if it was a fail mission, and green if it was a success one.

Folium map – Distances from ground stations to different POIs

- Here we plot the distance to different POIs, like the sea, the city of Orlando, a route intersection



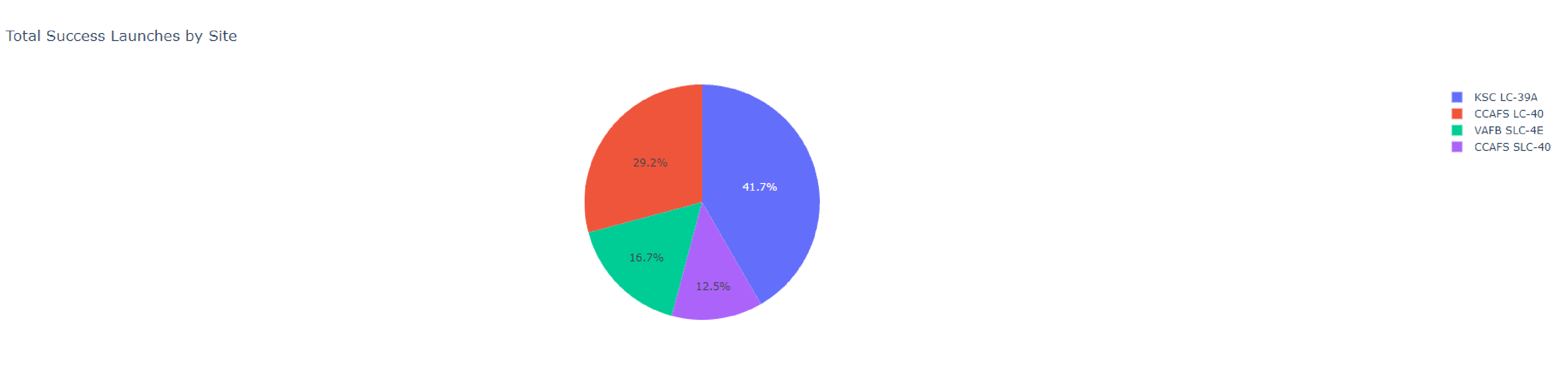




Section 5

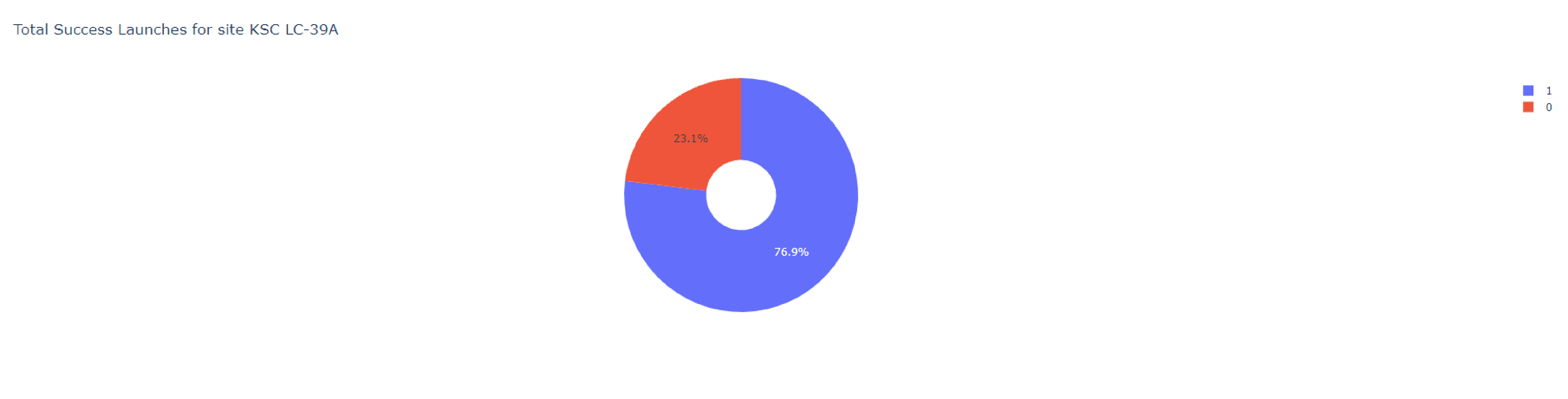
Build a Dashboard with Plotly Dash

Dashboard – Total success by site



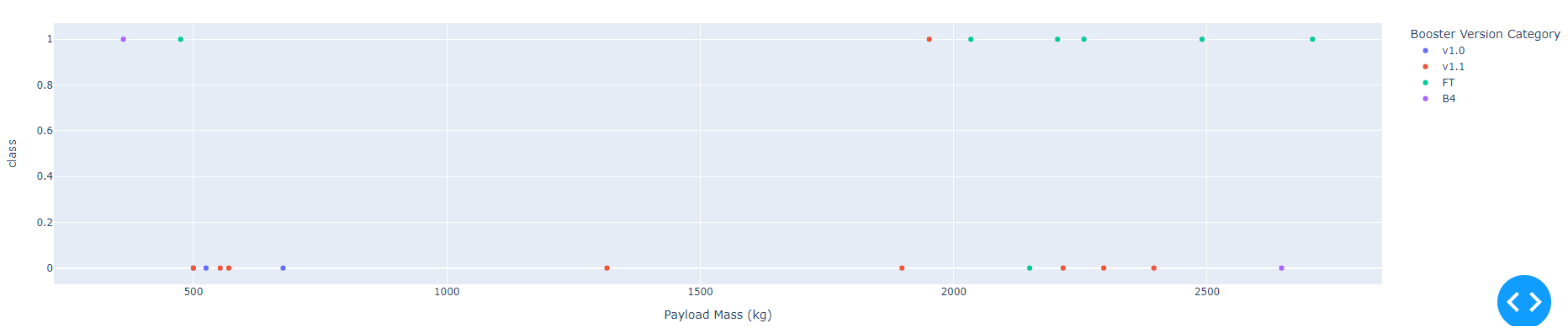
- We see from this chart how the success missions are distributed among all stations

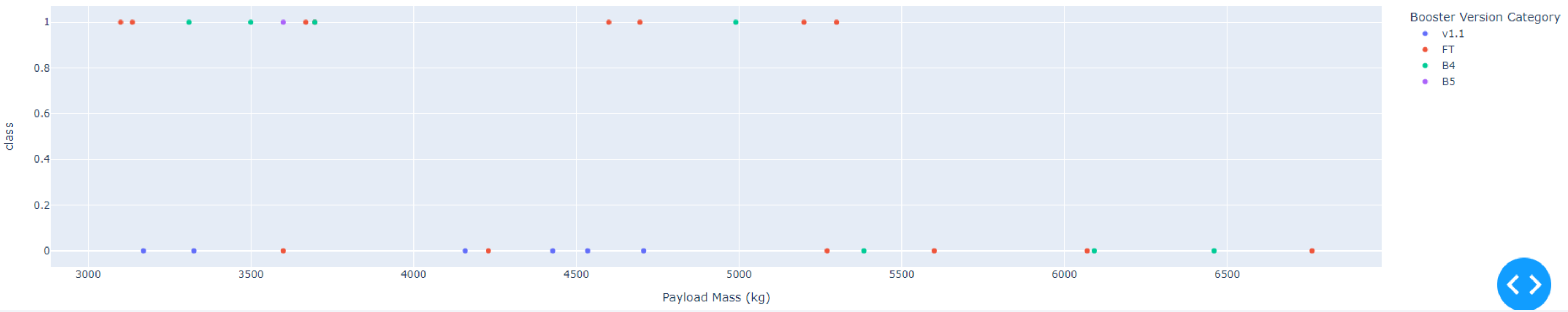
Dashboard – Outcome for station KSC LC-39A



- Mission outcomes for ground station KSC LC-39A, which is the one that has the greatest share in success missions among all stations.

<Dashboard Screenshot 3>





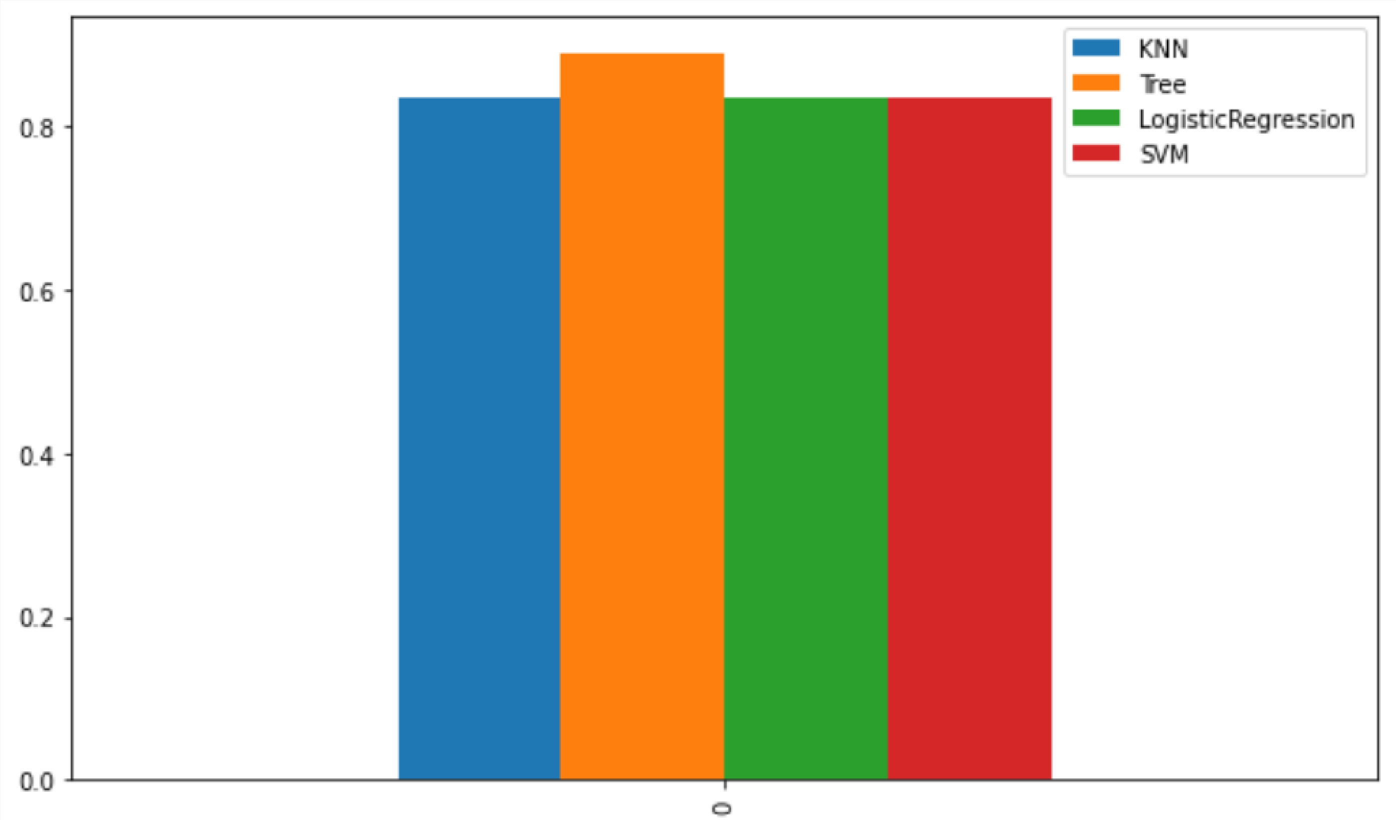
- 1st plot – Payload mass between 0 and 3000kg // 2nd plot – Payload mass between 3000 and 10.000kg. There's a "band" between 2000 and 3500kg in which we see that all boosters shows the best performance.

Section 6

Predictive Analysis (Classification)

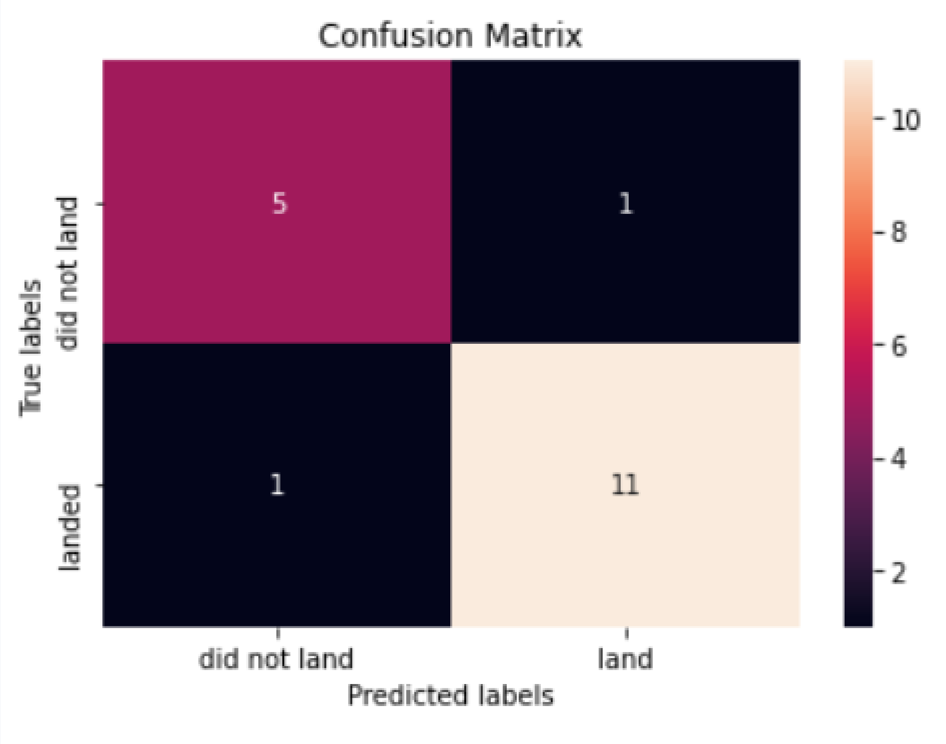
Classification Accuracy

- We see that all algorithms performed similar, being Decision Tree the one that showed the best accuracy.



Confusion Matrix – Decision Tree algorithm

- We see that **decision tree** showed the best performance for the given dataset and selected variables.
- It gives the best balance between matches and false positives / false negatives.



Conclusions

- Decision Tree was the algorithm that performed best among the tested ones. It showed the best balance between matches and false positives / false negatives.
- Mission success shows a clear correlation with number of previous launches, giving a clear indication of the SpaceX engineering team learning curve.
- Some orbits showed best success rate, which at first didn't seem to be some factor that correlates with mission success/failure. Maybe some better analysis could be made here in order to understand which other variables interact (physics, atmospheric, etc.).

Thank you!