



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Valentin Veintemilla
October 9th, 2024



Outline

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

Executive Summary

- Summary of methodologies
 - SpaceX Data Collection using SpaceX API
 - SpaceX Data Collection using Web Scraping
 - SpaceX Data Wrangling
 - SpaceX Exploratory Data Analysis using SQL
 - SpaceX EDA Data Visualization
 - SpaceX Launch Sites Analysis with Folium and Plotly Dash
 - SpaceX Machine Learning Landing Prediction
- Summary of all results
 - EDA Results
 - Interactive Visual Analytics and Dashboards
 - Predictive Analysis (Classification)

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
 - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - Data was split into train and test set. After that, we used different classification algorithms and chose the best one.

Data Collection

- Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia
- For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using `json_normalize()`. We then cleaned the data, checked for missing values and fill with whatever needed.
- For web scrapping, we Will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

- The information was extracted from a Public API where the data is stored
(<https://api.spacexdata.com/v4/launches/past>)
- <https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/1-jupyter-labs-spacex-data-collection-api.ipynb>



Data Collection - Scraping

- The information was extracted from a Wikipedia table
(https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)
- <https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/2-jupyter-labs-webscraping.ipynb>



Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outco column and exported the results to csv.
- <https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/3-labs-jupyter-spacex-Data%20wrangling.ipynb>



EDA with Data Visualization

- Scatter Graphs

- 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



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

- Bar Graph

- Success rate vs. Orbit

Bar graphs show the relationship between numeric and categoric variables.



- Line Graph

- Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.



- <https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/5-edadataviz.ipynb>

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
 - Displaying 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 achieved.
 - 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.
 - 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.
 - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/4-jupyter-labs-eda-sql-coursera_sqlite.ipynb

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas
 - Red circle at NASA Johnson Space Center's coordinate with label showing its name (fo/ium.Cirde, fo/ium.map.Marker).
 - Red circles at each launch site coordinates with label showing launch site name (fo/ium.Circ/e, fo/ium.map.Marker, fo/ium.features.Div/con).
 - The grouping of points in a cluster to display multiple and different information for the same coordinates (fo/ium.p/ugins.MarkeOuster).
 - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, fo/ium./con).
 - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, fo/iumPo/yLine, fo/ium.features.Div/con).
- The 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.
 - https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/6-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider 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`).
- https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

- Data preparation
 - Load dataset
 - Normalize data
 - Split data into training and test sets.
- Model preparation
 - Selection of machine learning algorithms
 - Set parameters for each algorithm to GridSearchCV
 - Training GridSearchModel models with training dataset
- Model evaluation
 - Get best hyperparameters for each type of model
 - Compute accuracy for each model with test dataset
 - Plot Confusion Matrix
- Model comparison
 - Comparison of models according to their accuracy
 - The model with the best accuracy will be chosen (see Notebook for result)
- https://github.com/vale20m/Applied-Data-Science-Capstone/blob/main/8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

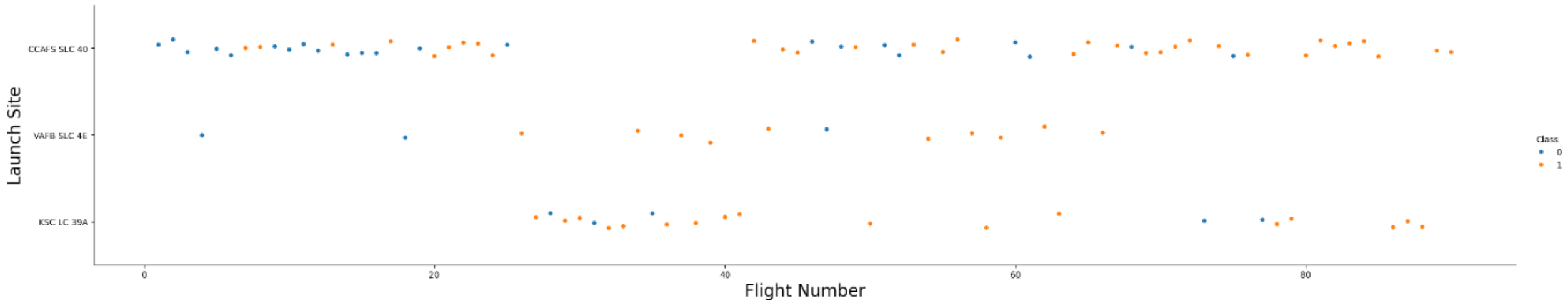
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

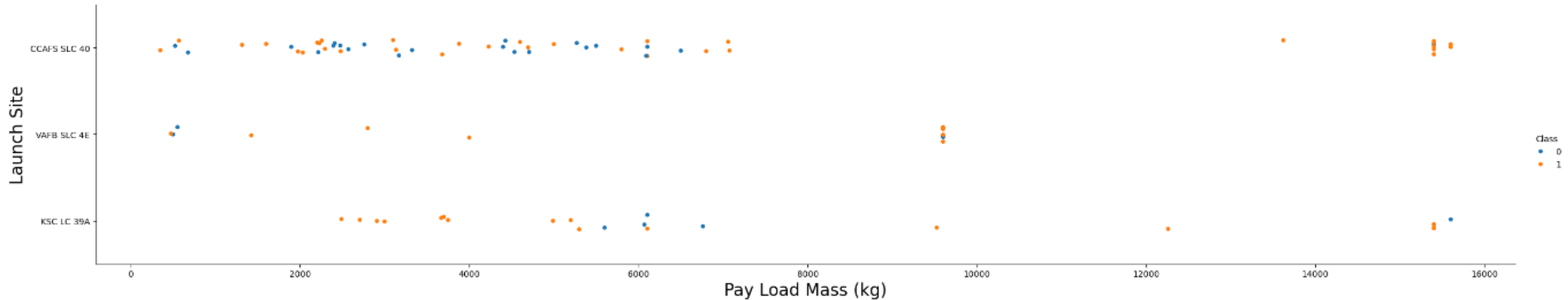
Insights drawn from EDA

Flight Number vs. Launch Site



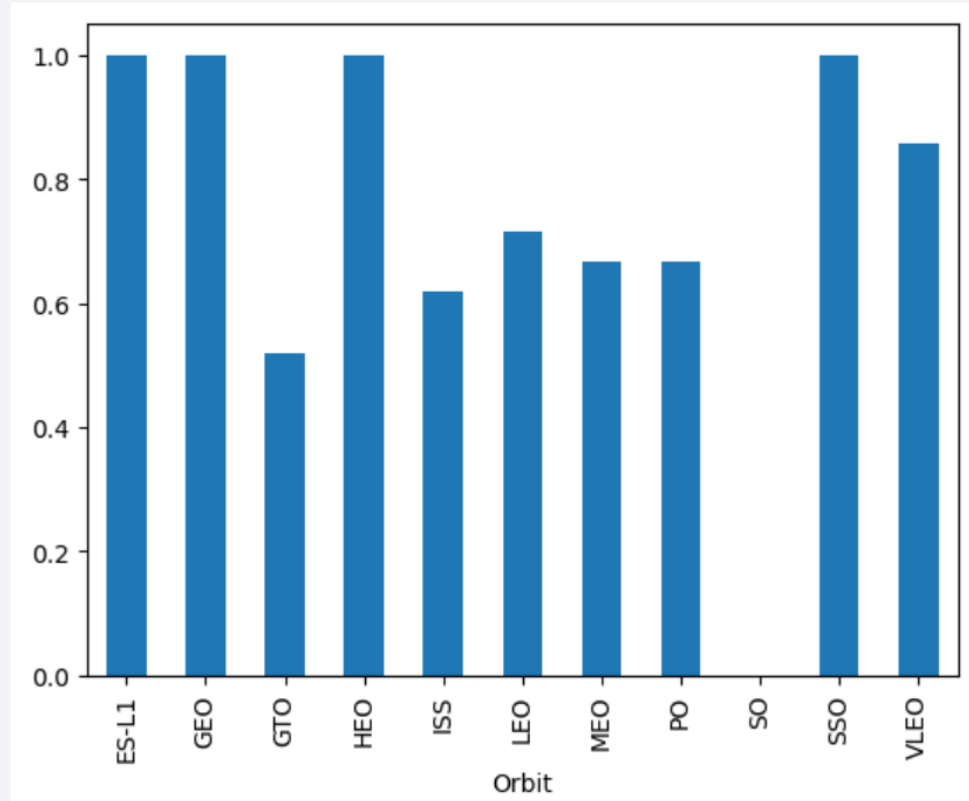
We observe that, for each site, the success rate is increasing.

Payload vs. Launch Site



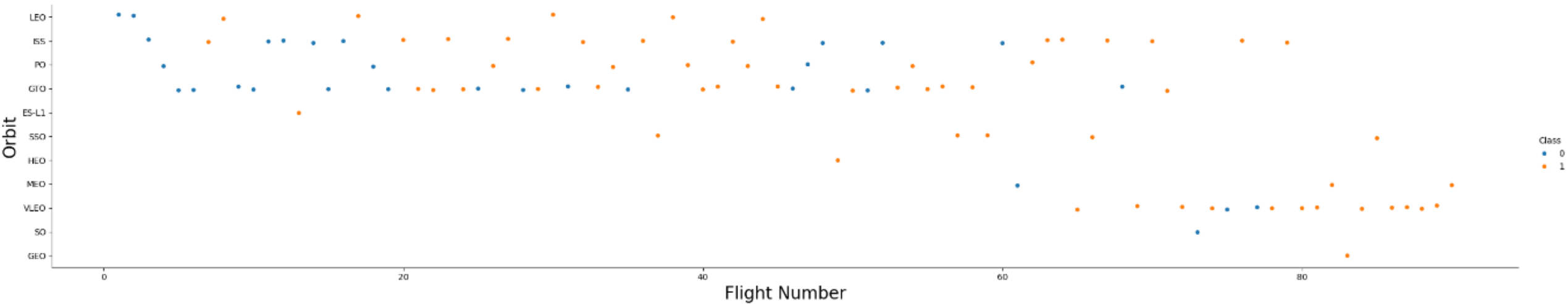
Depending on the launch site, a heavier payload may be a consideration for a successful landing. On the other hand, a too heavy payload can make a landing fail.

Success Rate vs. Orbit Type



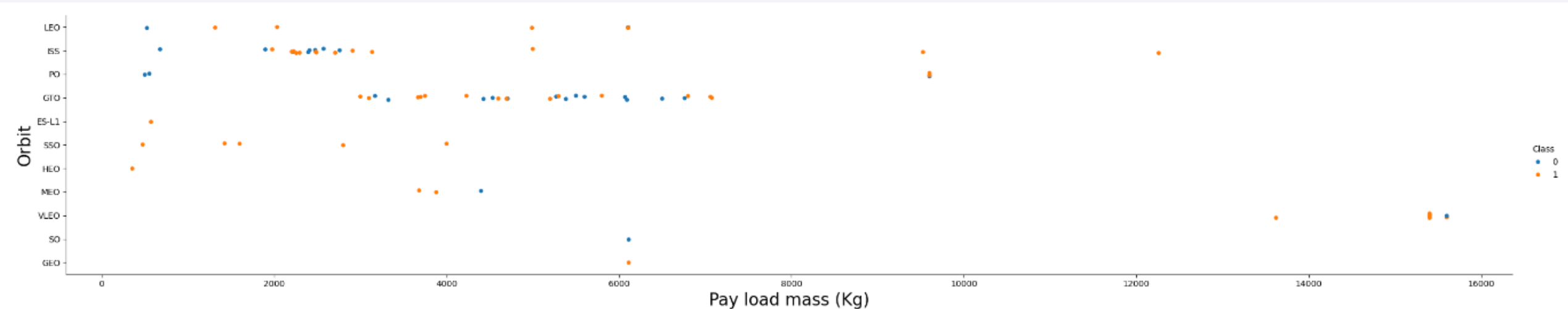
With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



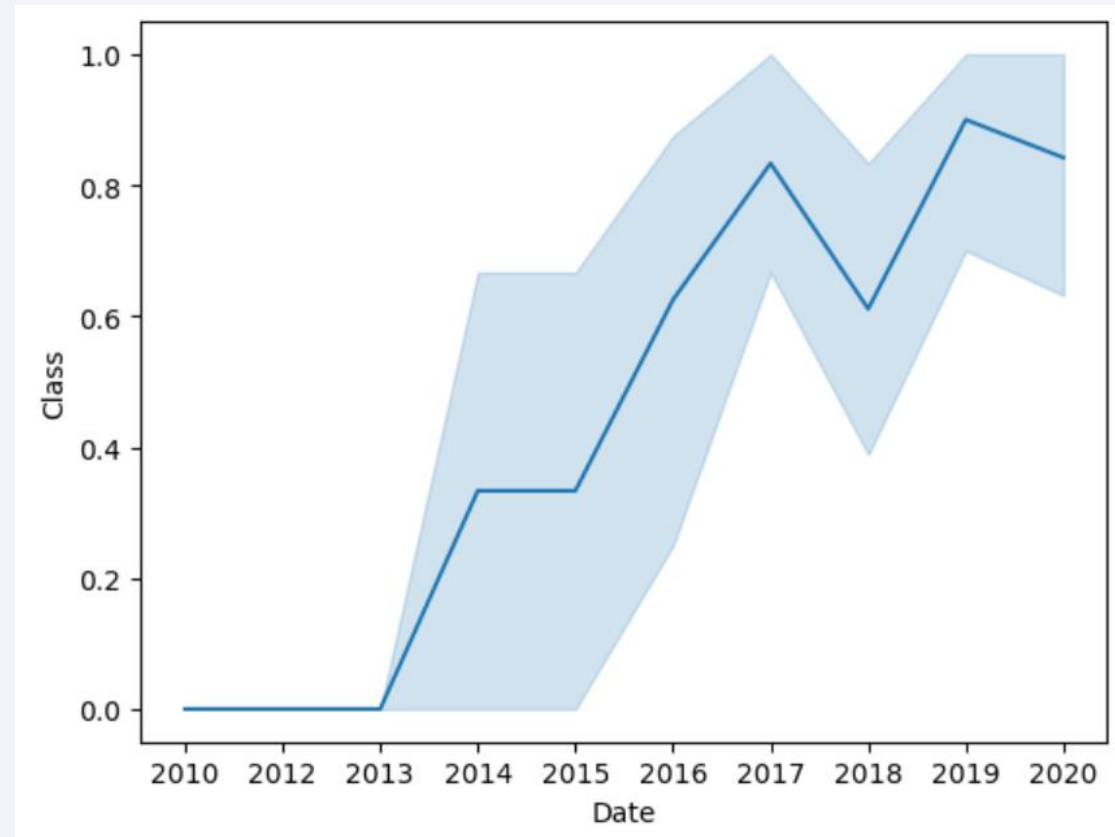
We notice that the success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights. But we can suppose that the high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.

Payload vs. Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for a GTO orbit improves the success of a launch.

Launch Success Yearly Trend



Since 2013, we can see an increase in the Space X Rocket success rate.

All Launch Site Names

SQL Query

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Results

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

Explanation

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SQL Query

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

Explanation

The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

Results

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

SQL Query

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

Results

SUM("PAYLOAD_MASS__KG_")
45596

Explanation

This query returns the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL Query

Results

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL Query

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success%'
```

Results

MIN("DATE")

01-05-2017

Explanation

With this query, we select the oldest successful landing.

The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS_KG_" > 4000 AND "PAYLOAD_MASS_KG_" < 6000;
```

Results

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

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%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
```

Results

SUCCESS	FAILURE
100	1

Explanation

With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS_KG_" = (SELECT max("PAYLOAD_MASS_KG_") FROM SPACEXTBL)
```

Explanation

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

Results

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

SQL Query

```
%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'
```

Results

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

Explanation

This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

SQL Query

```
%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 ;
```

Results

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

Explanation

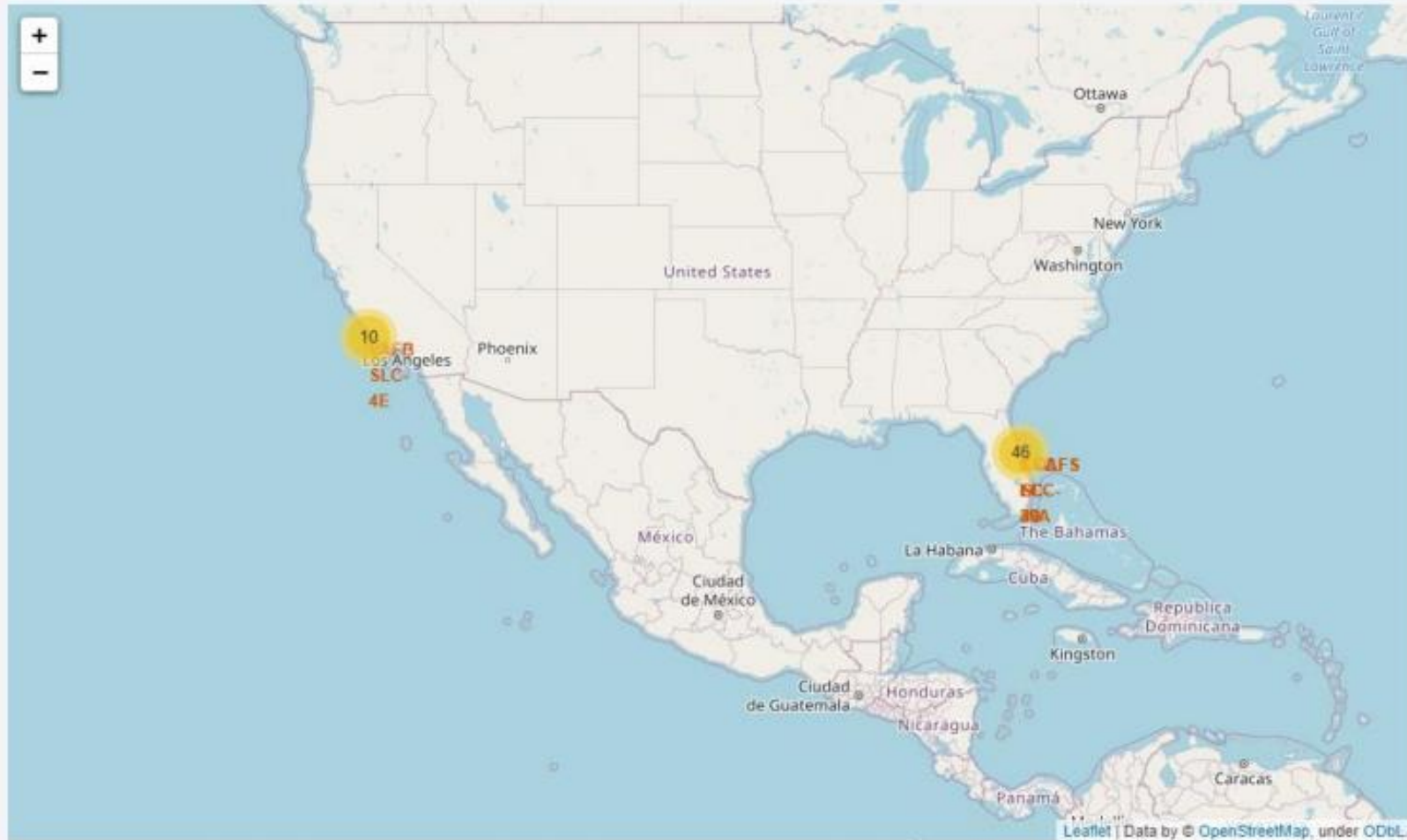
This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

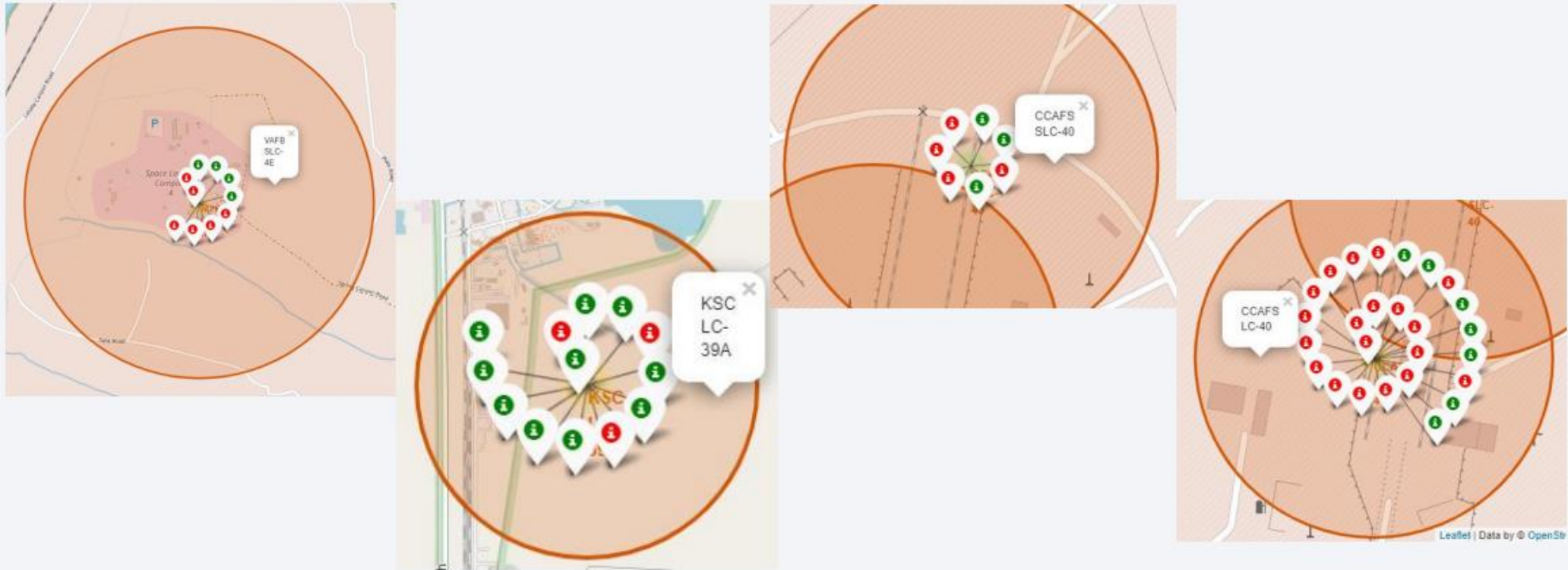
Launch Sites Proximities Analysis

Folium map – Ground stations



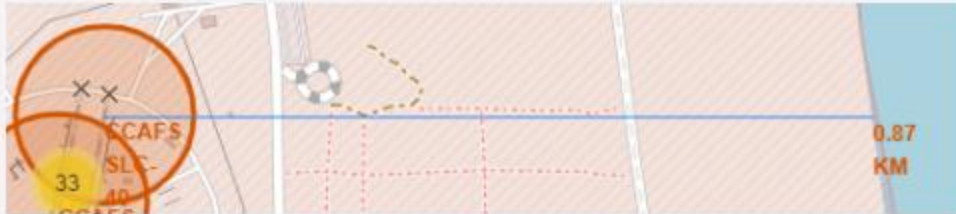
We see that Space X launch sites are located on the coast of the United States

Folium map – Color Labeled Markers



Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

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



Is CCAFS SLC-40 in close proximity to railways ? Yes

Is CCAFS SLC-40 in close proximity to highways ? Yes

Is CCAFS SLC-40 in close proximity to coastline ? Yes

Do CCAFS SLC-40 keeps certain distance away from cities ? No



Section 4

Build a Dashboard with Plotly Dash

Dashboard – Total success by Site

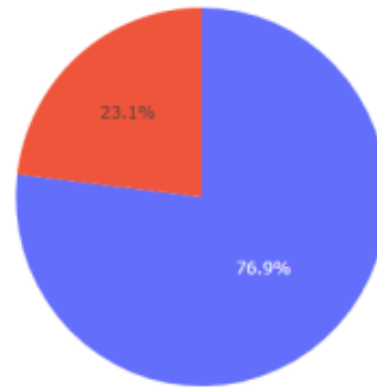
Total Success Launches by Site



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

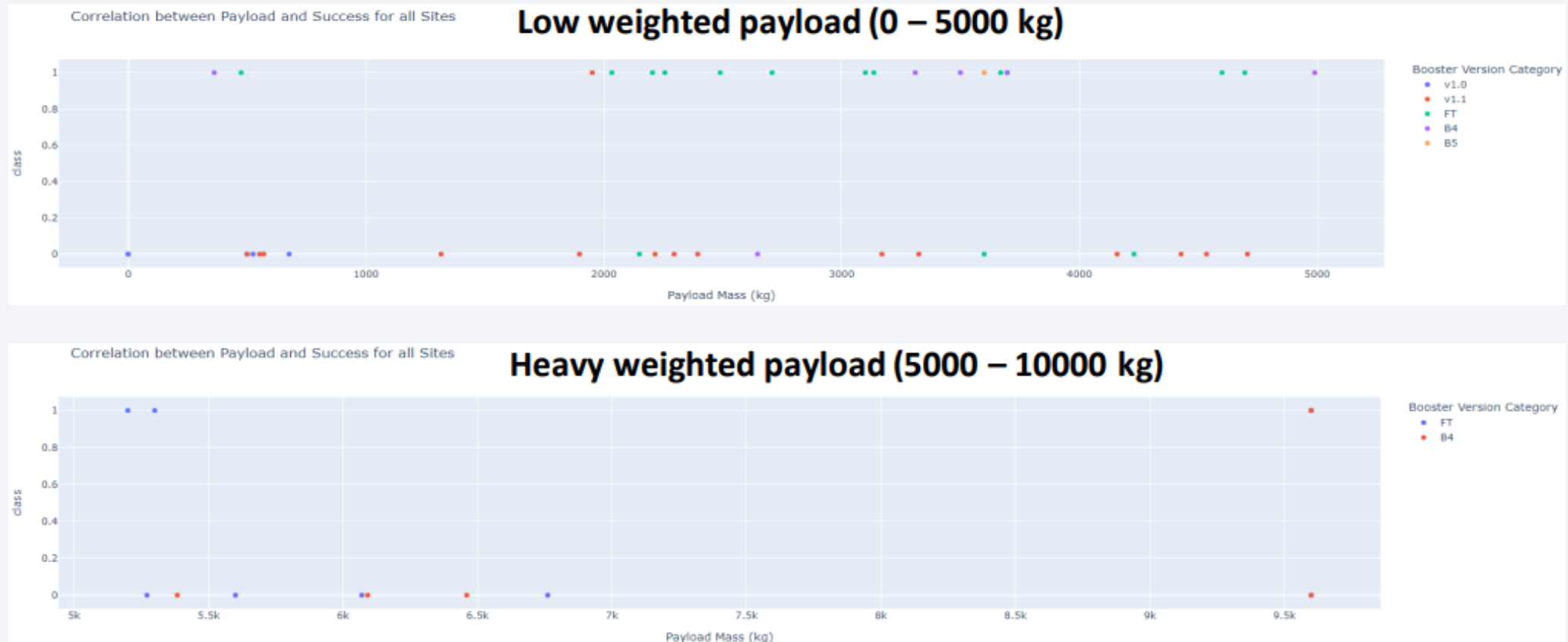
Dashboard – Total success launches for Site KSC LC-39A

Total Success Launches for Site KSC LC-39A



We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected



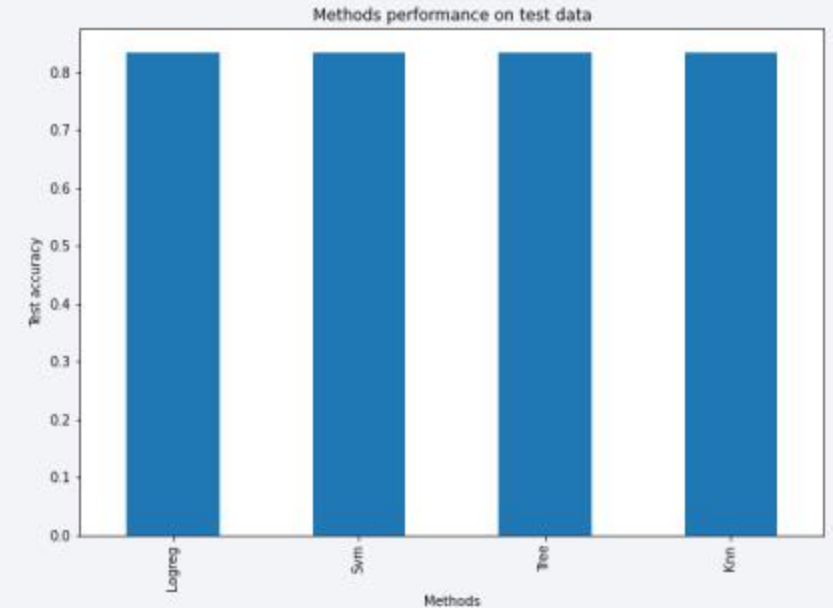
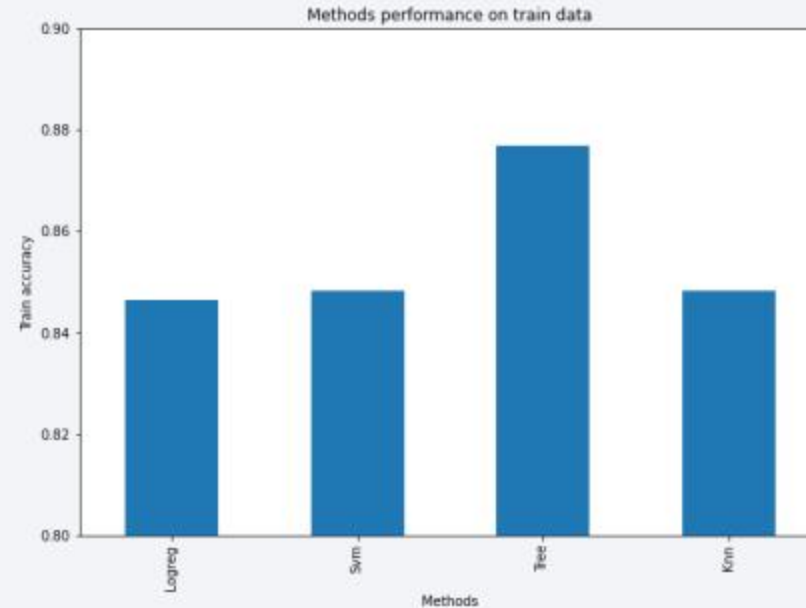
Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

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



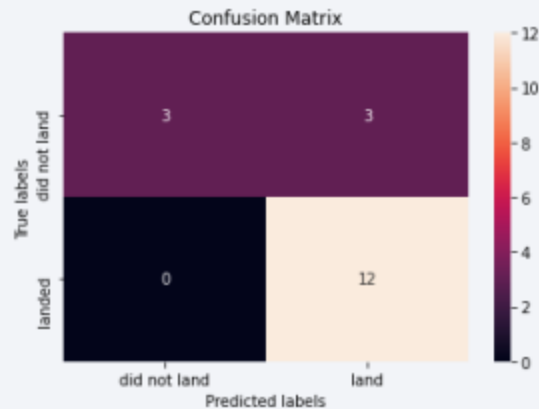
For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

Decision tree best parameters

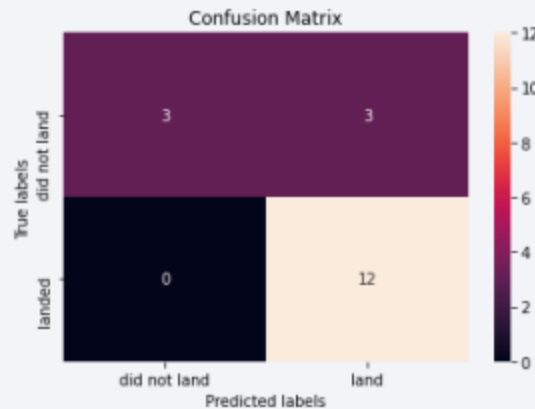
```
tuned hyperparameters :(best parameters) {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix

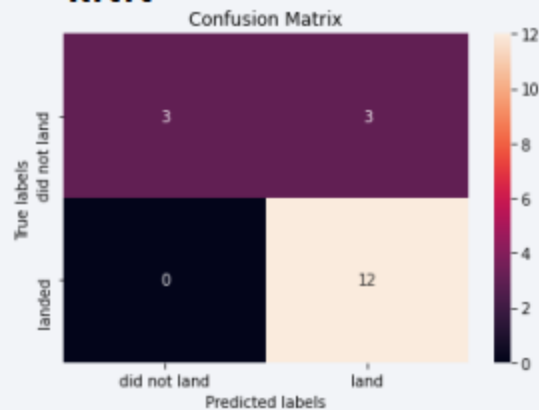
Logistic regression



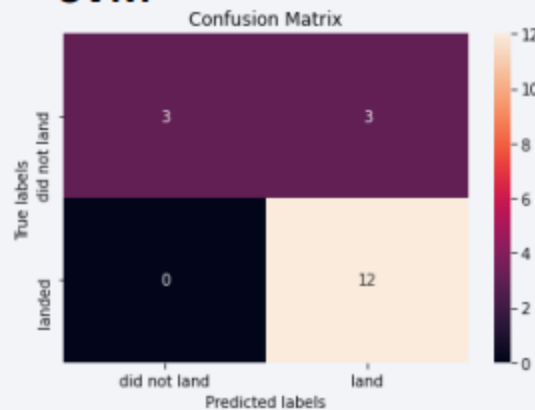
Decision Tree



kNN



SVM



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

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

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-LI.
- 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.

Thank you!

