



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Taha Saouri
24/01/2023



Outline

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

Executive Summary

- Different methodologies were used the problem we are working on, we used python as the programming language in addition to its diverse libraries such as Pandas and Numpy to process the data. SQL was also used for databases purposes.
- Using data analysis and predictive analysis, we were able to understand well the correlation and relations between different variables, thus we are able to predict an output with high precision.

Introduction

- The space race goes into the context of our project, especially the need of optimizing the procedures and evaluating the circumstances of the missions to be able to guarantee a certain success.
- The specific problem we are confronted with in this project is the Outcome of the landing of rockets stage 1, so we are able to reuse them in the future. Given the data we try to find patterns for the purpose of predicting the outcome of landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Data collection was performed using REST API and Web scraping
- Perform data wrangling
 - Data was processed using Pandas and Numpy
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - Sklearn was used to build a model to predict the outcome

Data Collection

- Data was collected from various sources, using different methods. First, we used REST API to collect data from a url in a json format, using requests library and method get. The retrieved Json data was then normalized to be converted to a data frame, so that it's ready to be processed and cleaned.
- The second way we used to collect data is Web scraping, we used the requests library and Beautifulsoup to retrieve data from a web page and parse the HTML. The find_all method from Beautifulsoup helped us to find all the tables present in the web page and store them in a list, then we used a function to go through all the tables and add them one by one to a dataframe we've created using the columns present in the webpage.

Data Collection – SpaceX API

- While collecting data with SpaceX API we had to create specific functions to deal with the API and retrieve the important data we need. In addition to requests library and pandas to retrieve the whole data and process it.
- [Notebook link\(github\)](#)

response = requests.get(url)

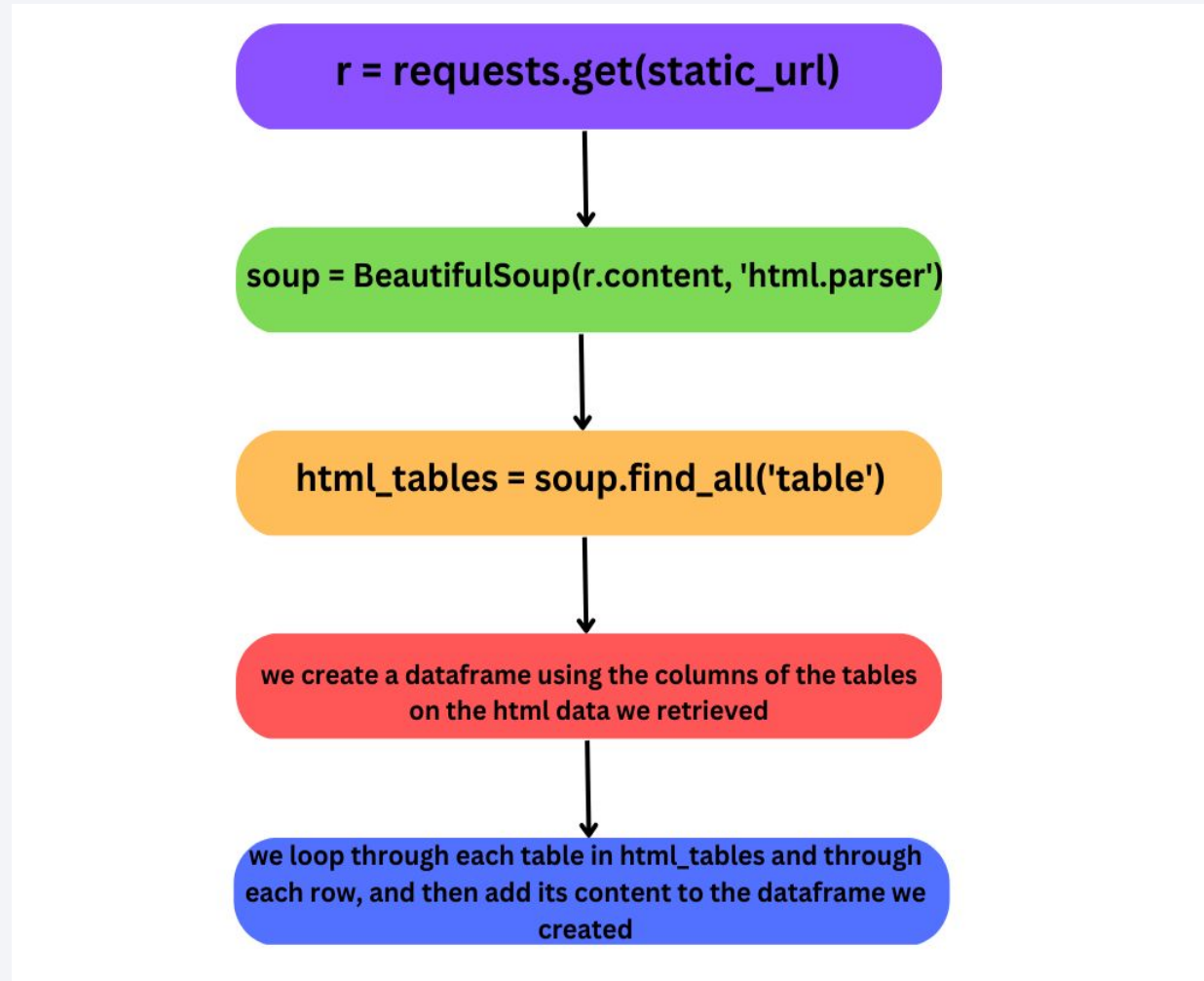
pd.json_normalize(response.json())

use functions to retrieve the data we want such as information concerning the rocket, the payload, the launchpad and the cores.

we create a new data frame and filter it from the data we don't want and deal with missing values

Data Collection - Scraping

- We start scraping the same way using a get call to retrieve the data, then we parse the html using BeautifulSoup, then we use find_all method with 'table' as parameter, we loop through the tables and append the rows data to a definitive newly created dataframe.
- [Notebook link\(github\)](#)



Data Wrangling

- We started data wrangling by calculating the number of launches on each site.
- We then calculated the number of occurrence of each orbit from the different orbits (such as LEO, VLEO, GTO, SSO, HEO...) and the number and occurrence of mission outcome per orbit type using method value_counts() from pandas.
- We finished data wrangling with creating a landing outcome label from outcome column, that we added to the data frame as 'Class'; 1 for success and 0 for failure.
- [Notebook link\(github\)](#)

EDA with Data Visualization

- We used multiple charts to represent the data, such as scatter plot to visualize the relationship between two variables and the impact on the outcome. Bar charts helped us visualize the relationship between a categorical variable and a continuous variable such.
- We also used line chart to visualize the development of a variable with time.
- [Notebook link\(github\)](#)

EDA with SQL

- 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 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.
- [Notebook link\(github\)](#)

Build an Interactive Map with Folium

- We created a map with folium, and added different objects to it such as markers, circles and icons to represent some coordinates which in our case represent launch sites locations.
- We used icons to mark the launch success or failure for each launch site, markers are used to mark a point and add some text or popup to it.
- Circles were used to customize our marked points
- [Notebook link\(github\)](#)

Build a Dashboard with Plotly Dash

- We added to our Plotly dashboard a Launch Site Drop-down Input Component, so that the user can select a launch site.
- We also added callback function to render a pie chart based on selected launch site dropdown in real-time.
- We finally added a Range Slider to Select Payload and callback function to render the scatter plot based on the range slider value selected.

Predictive Analysis (Classification)

- We started the preparation of our model by importing the data set and specify the features X and the label Y as numpy arrays.
- Then we normalized the data and split our data into training and testing.
- The next phase started which consisted in creating models with different algorithms and using *GridSearchCV* to find the best parameters.
- The different algorithms consisted in; Logistic regression, SVC, KNNs, and Decision tree.
- We evaluated each algorithms using the performance on the training set and the test set.
- [Notebook link\(github\)](#)

Results

- Exploratory data analysis results: we can conclude that there is a relationship between the different features of every launch and the outcome of the Landing, especially some specific features such as the Payload, the launch site and the orbit type.
- Predictive analysis results: we can conclude that the best algorithm to use for our model is Decision trees, which do very well on the training set as well as on the test set compared to the other algorithms, we can also remark that the precision is quite remarkable.

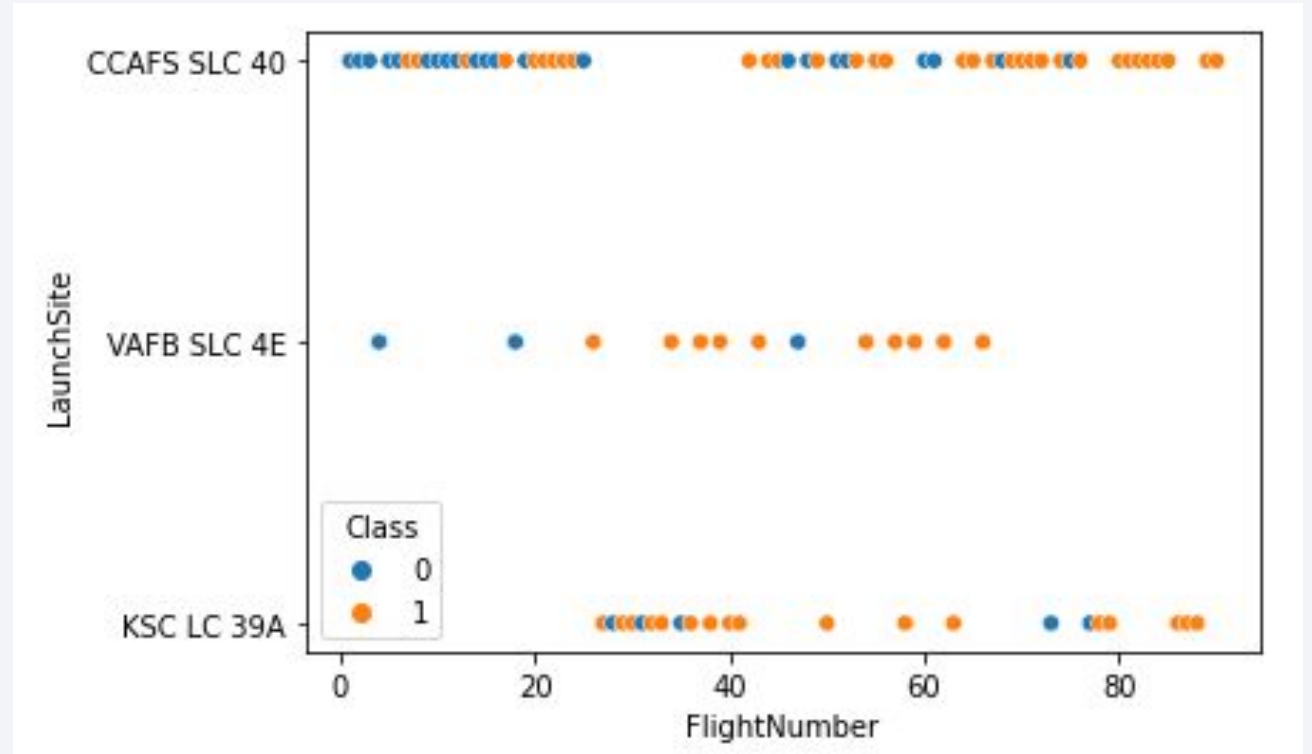
The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue and red on the right. These streaks are layered over a faint, light-blue grid pattern, creating a sense of depth and movement.

Section 2

Insights drawn from EDA

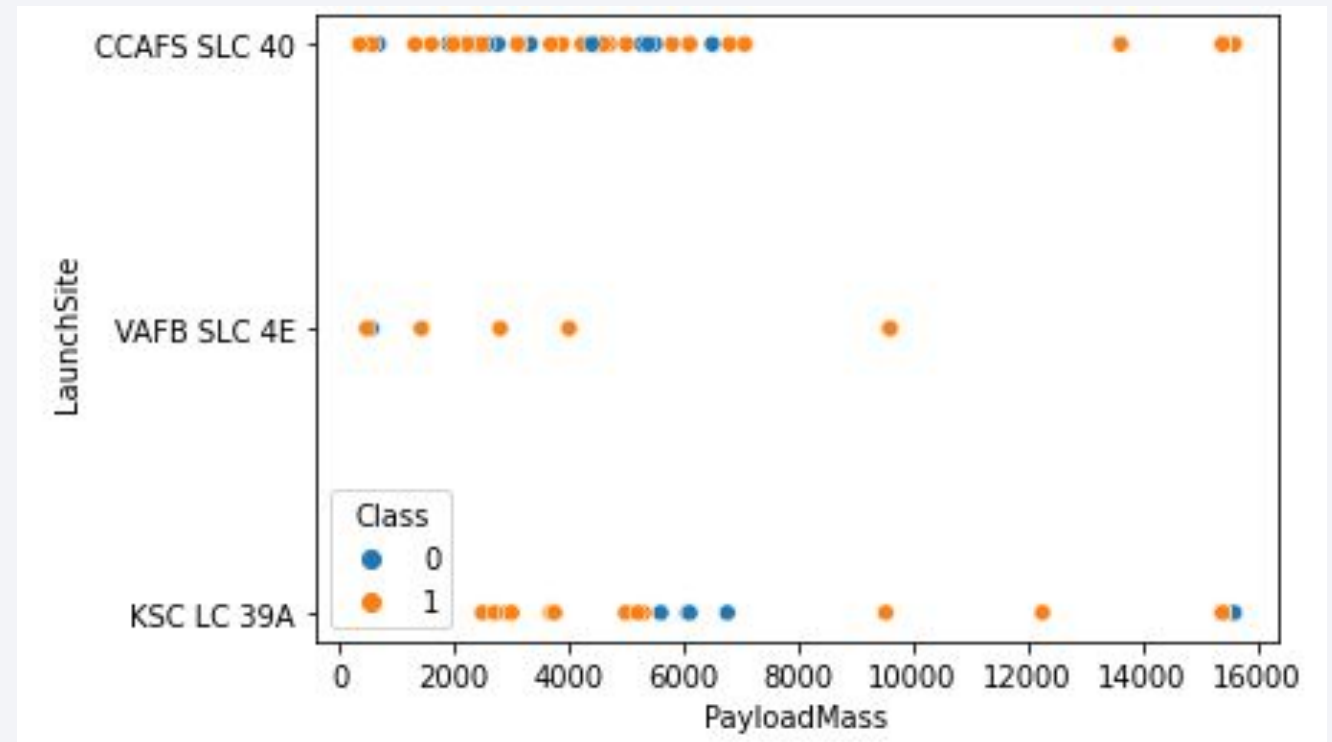
Flight Number vs. Launch Site

- From this plot, we can remark a lot of things such as the high frequency of failure in landing when the launch site is CCAFS SLC 40 and flight number below 40.
- The color yellow is for successful landings and blue for unsuccessful landings.



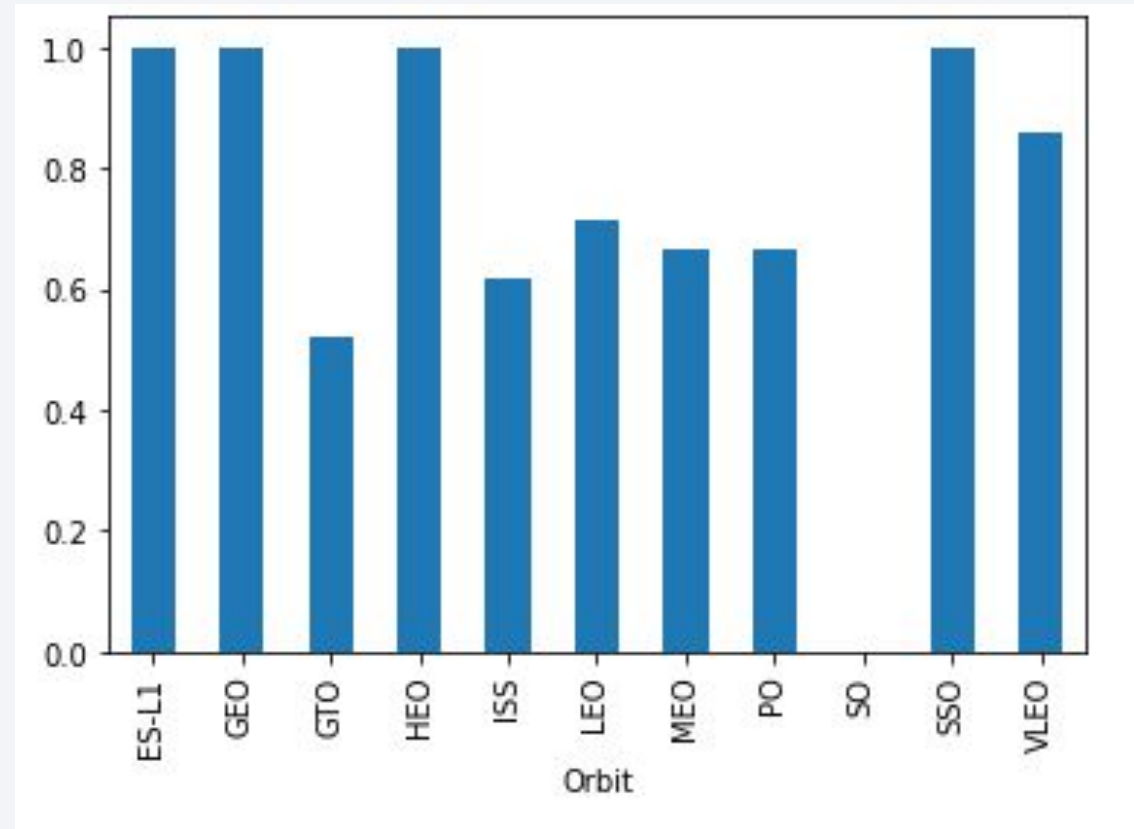
Payload vs. Launch Site

- We can remark that all the all landings with launchsite KSC LC 39A and a payload mass inferior to 5000 were successful.



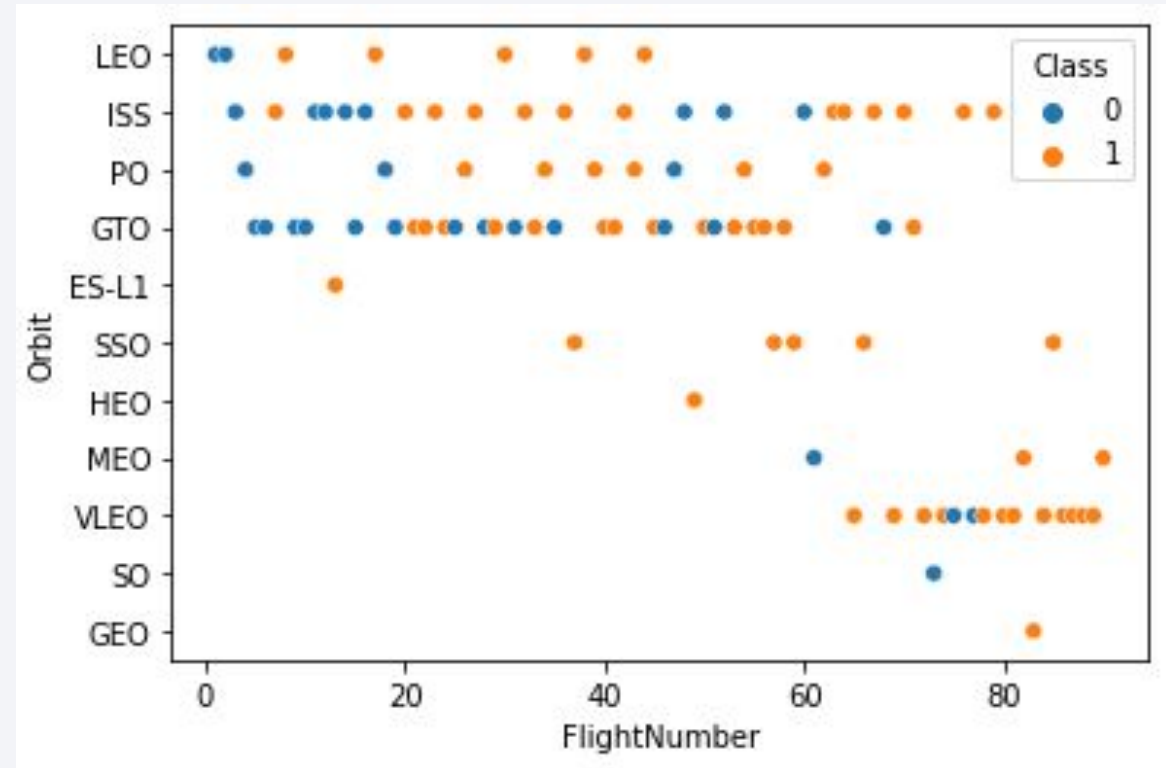
Success Rate vs. Orbit Type

- We can see that for some orbit types we have 100% success rate like HEO, and for some other 0% success rate like SO.



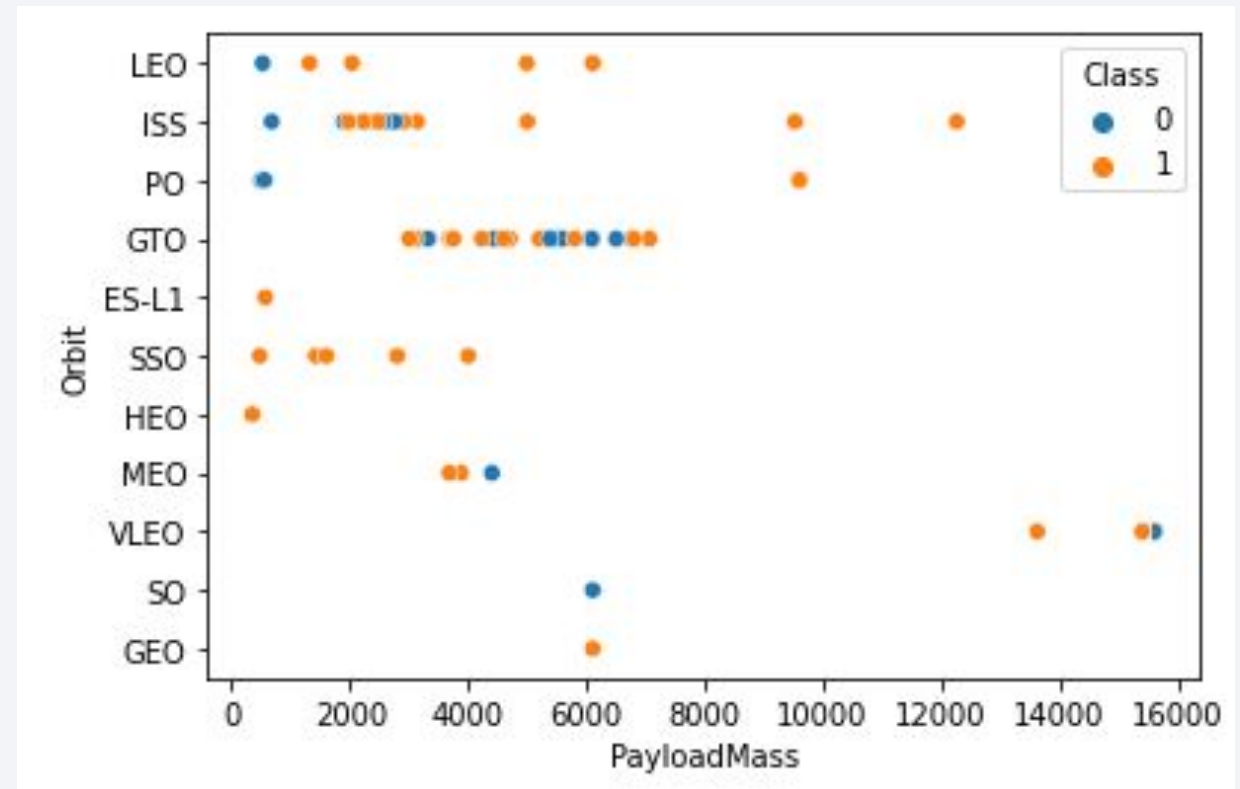
Flight Number vs. Orbit Type

- For some orbits we can see some patterns based on the flight number like for LEO and ISS, for some other orbits it's difficult.



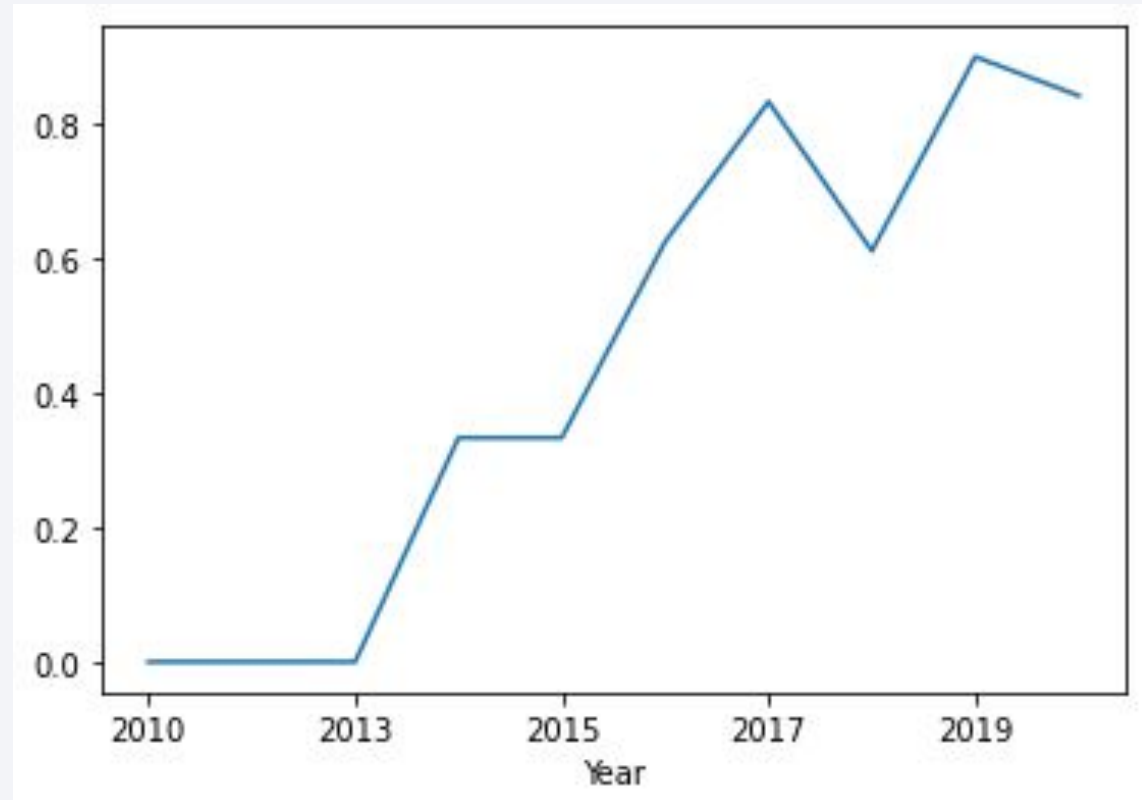
Payload vs. Orbit Type

- Again we can observe some patterns in the relationship between the orbit the payload mass that affects the outcome of the landing.



Launch Success Yearly Trend

- We can observe from the chart that there exists an upward trend as the years advance, starting with an average success rate of 0 to about 0.9 in 2020.



All Launch Site Names

- Using Sql we are able to execute a query to retrieve all launch sites names:
- We use the keyword distinct to only retrieve unique Launch site names.

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

Launch Site Names Begin with 'CCA'

- Using an sql statement we were able to retrieve all the launch site names that begin with 'CCA', we use the wildcard character % to precise the result of our query.

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

- We retrieve the total payload mass carried by boosters launched by NASA (CRS), we use the function sum and condition our statement using 'where'.

```
sum(PAYLOAD_MASS_KG_)
111268
```

Average Payload Mass by F9 v1.1

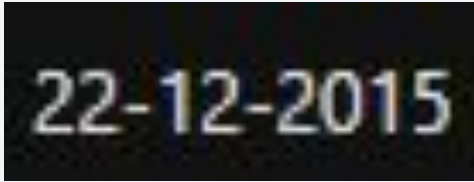
- we display the average payload mass carried by booster version F9 v1.1, we use the function avg and 'Where' for the condition:

```
avg(PAYLOAD_MASS_KG)
```

```
2928.4
```

First Successful Ground Landing Date

- We list the date when the first successful landing outcome in ground pad was achieved. We use the min function.



22-12-2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- We list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, using the keyword “between”.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06-05-2016	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
14-08-2016	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
11-10-2017	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- We calculate the total number of successful and failure mission outcomes, and group them by mission outcome using “group by”.

Mission_Outcome	count("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We list the names of the booster which have carried the maximum payload mass, we use a subquery.

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

- We list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, we use substr to retrieve month and year from date.

month	land	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

- We 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 :

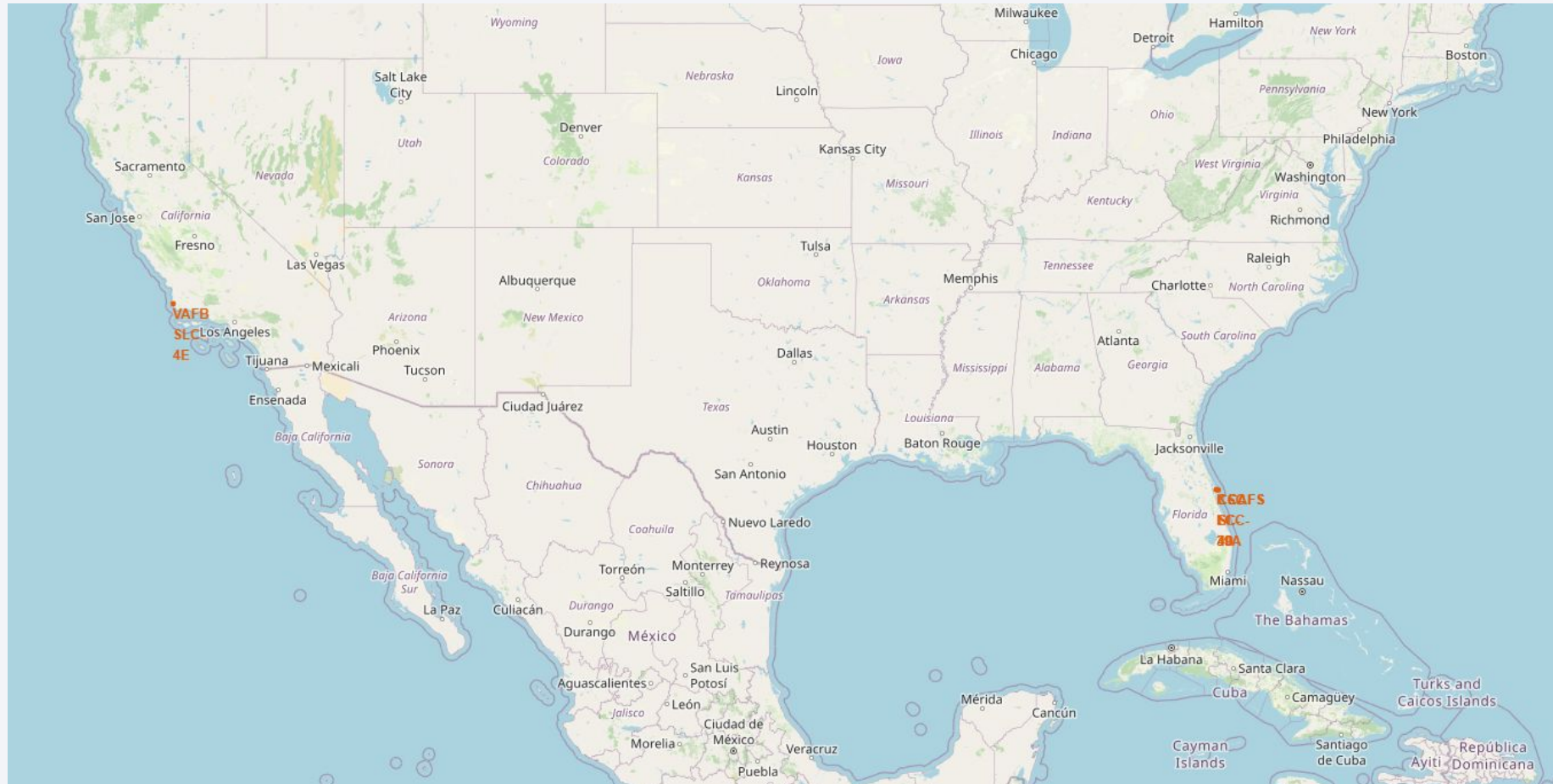
Landing_Outcome	ct
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky with stars and a view of the Earth's surface from space. The Earth's surface is mostly dark, with a dense network of yellow and orange lights representing city lights at night. The lights are concentrated in the lower right portion of the image, following the curve of the Earth. The upper portion of the image shows the dark blue sky with a few stars.

Section 3

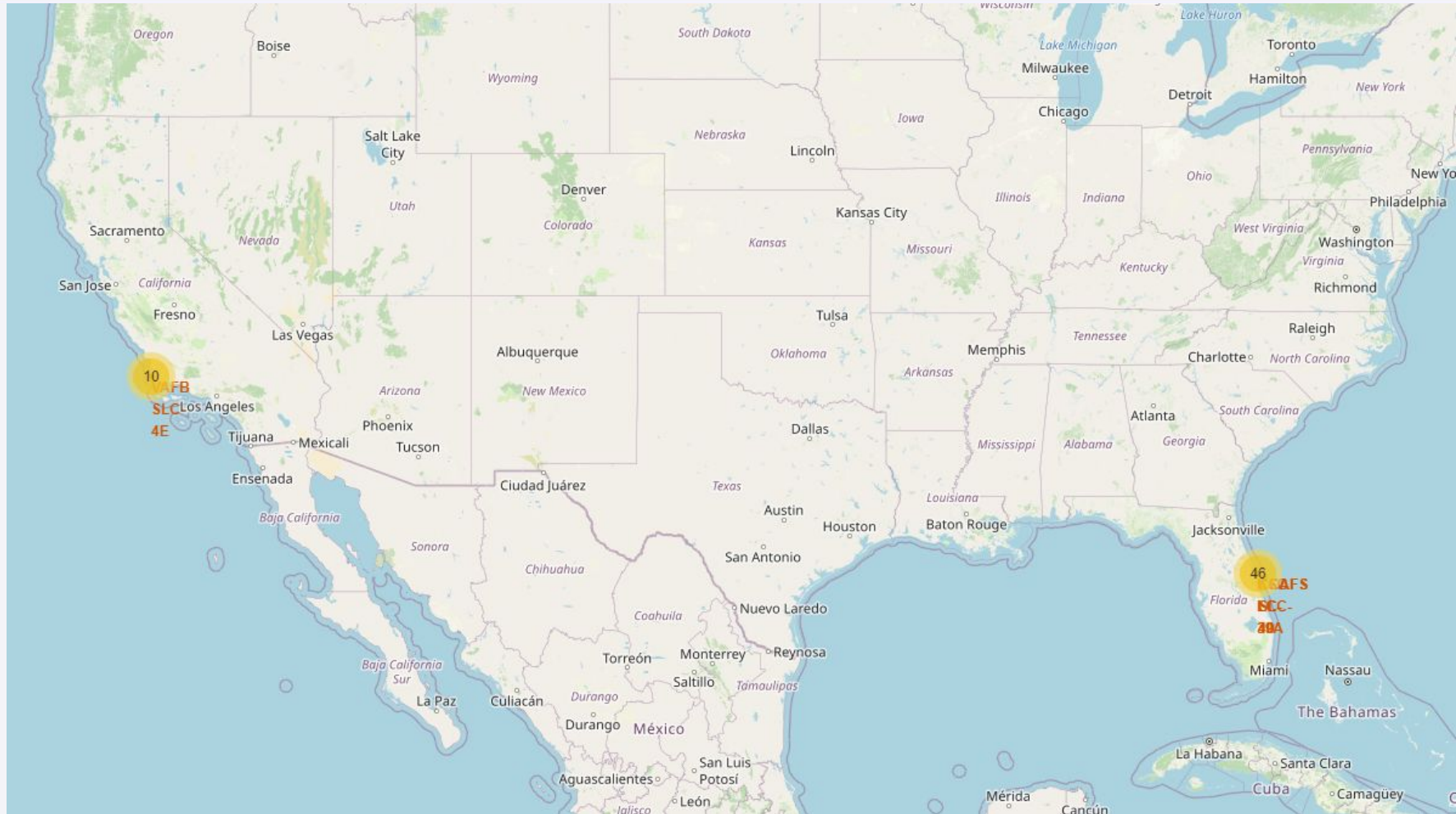
Launch Sites Proximities Analysis

The location of the all launch sites on a map



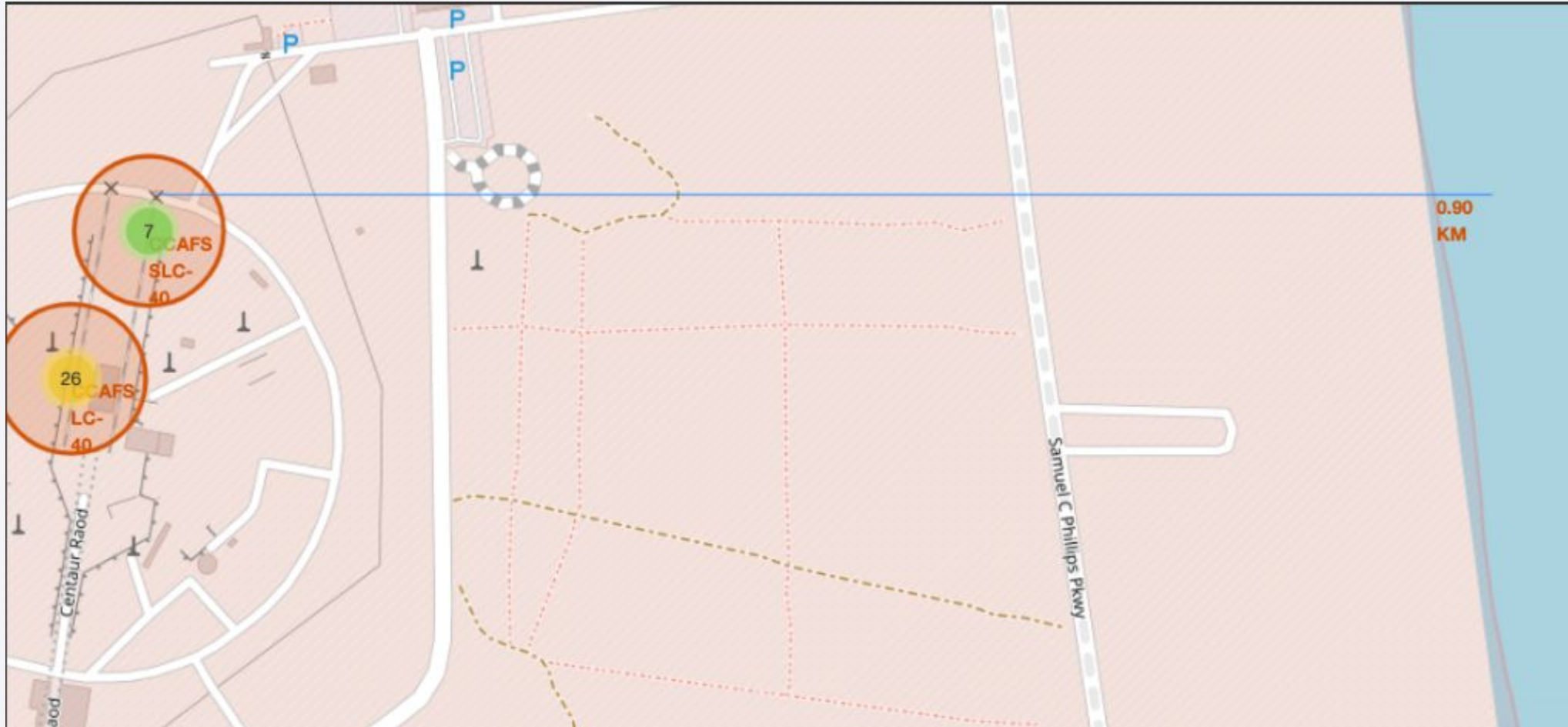
We can remark that two of the launch sites are close to each other and the third is on the opposite side of the country

success/failed launches for each site on the map



We can remark the number of launches on each site and if we zoom in we can distinguish the failed and successful launches

The distance between a launch site to a coastline



We can see that the distance between the launch site CCAFS SLC-40 and the closest coastline is 0.9 km



Section 4

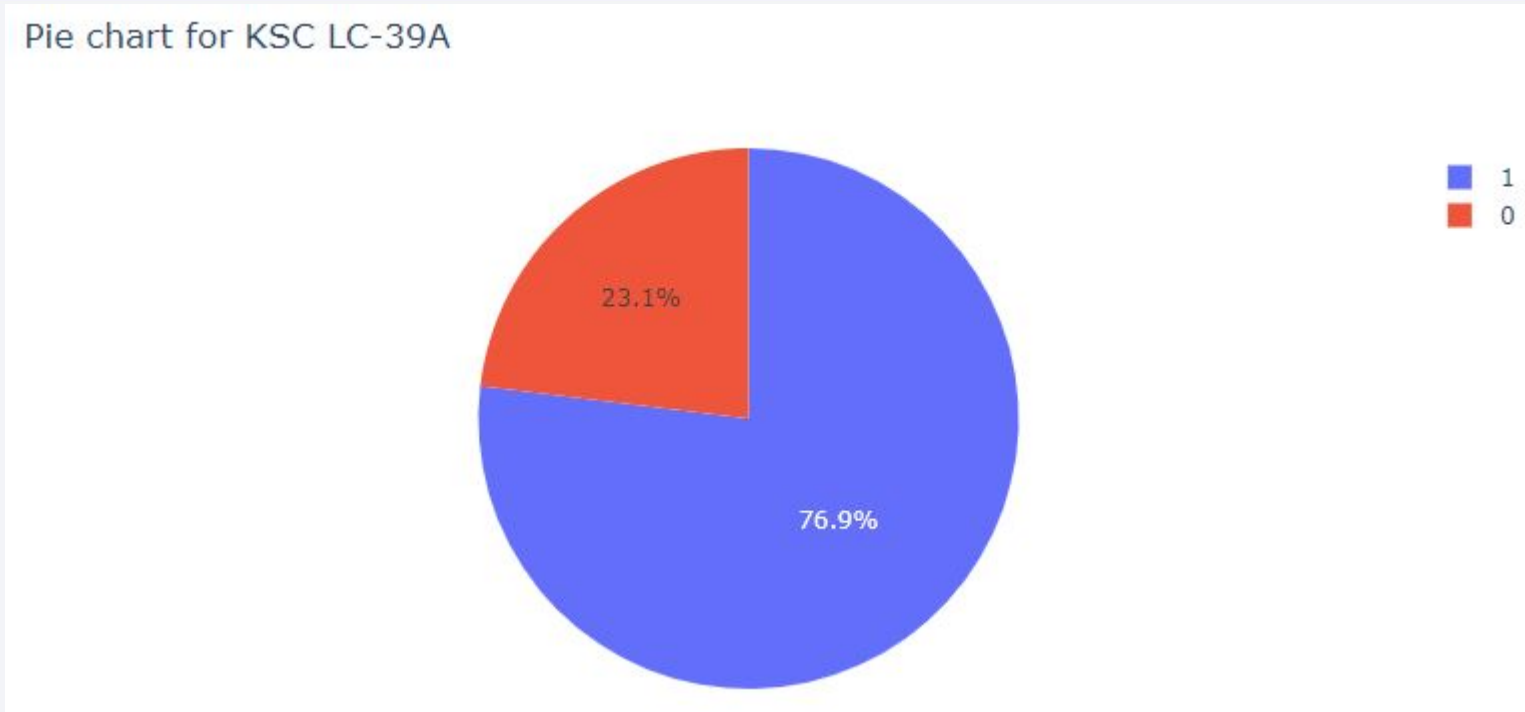
Build a Dashboard with Plotly Dash

Pie chart of launch success count for all sites



- From the chart above we can see the proportions of successful launch that represent each launch site.

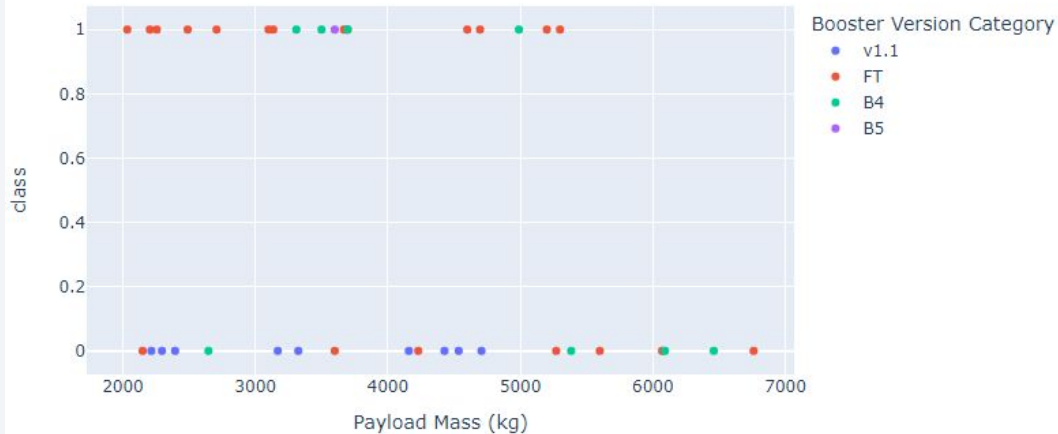
Pie chart for the launch site with highest launch success ratio



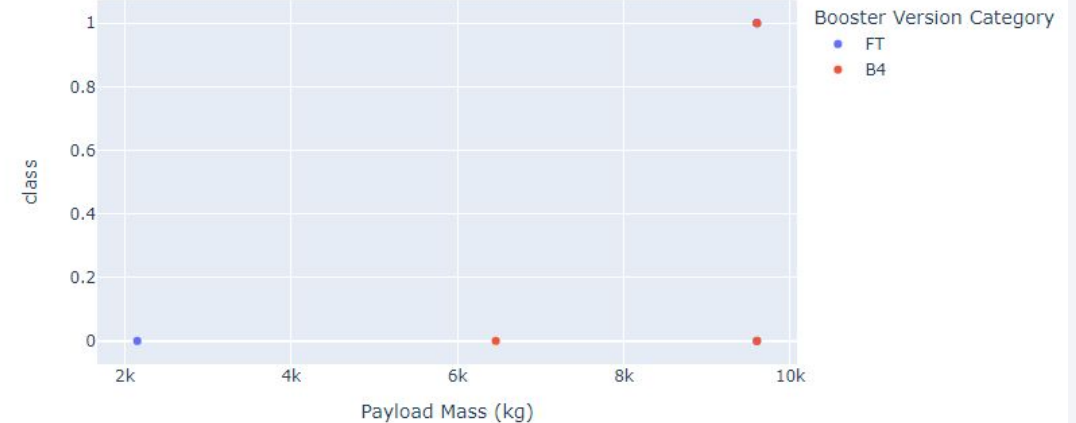
- We can see the the launch site with the highest launch success ratio is KSC LC-39A with 76.9% successful launches.

Scatter plot of Class vs the payload mass for all the sites

Scatter plot of all sites

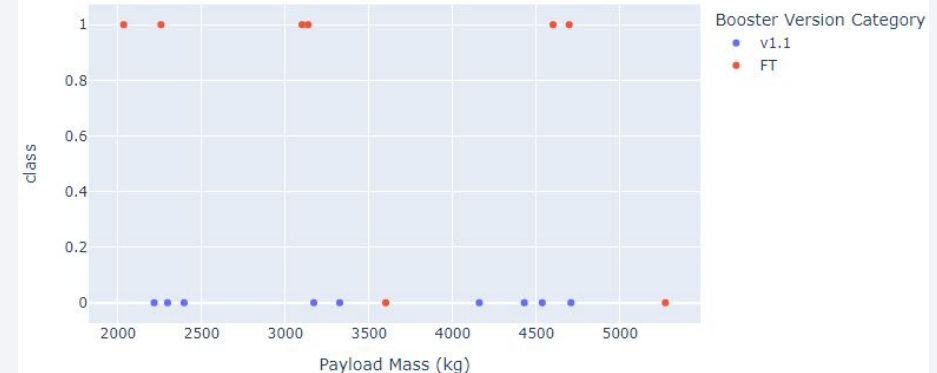


Scatter plot of VAFB SLC-4E



- We can see on the three chart, the disposition of the points based on class and Payload mass, the color represent the the booster version category.

Scatter plot of CCAFS LC-40

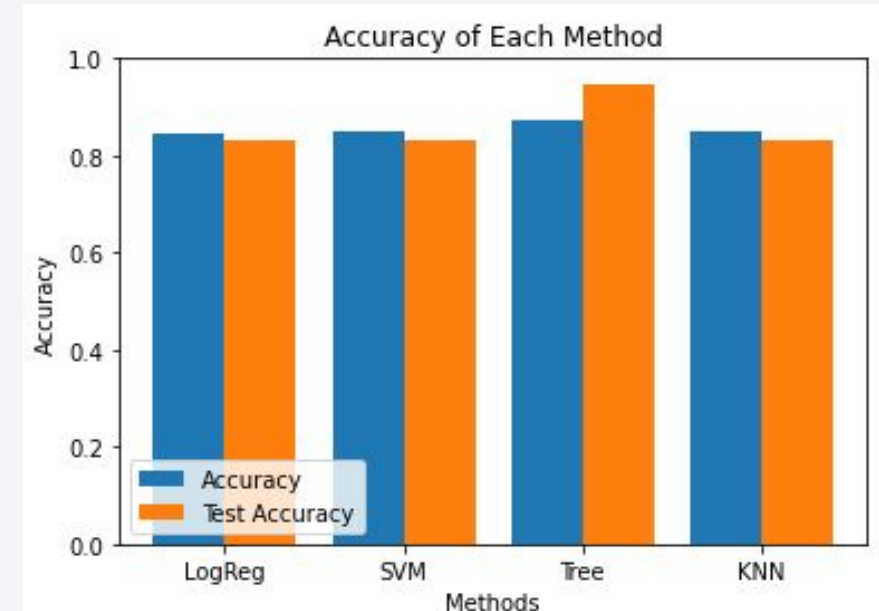


Section 5

Predictive Analysis (Classification)

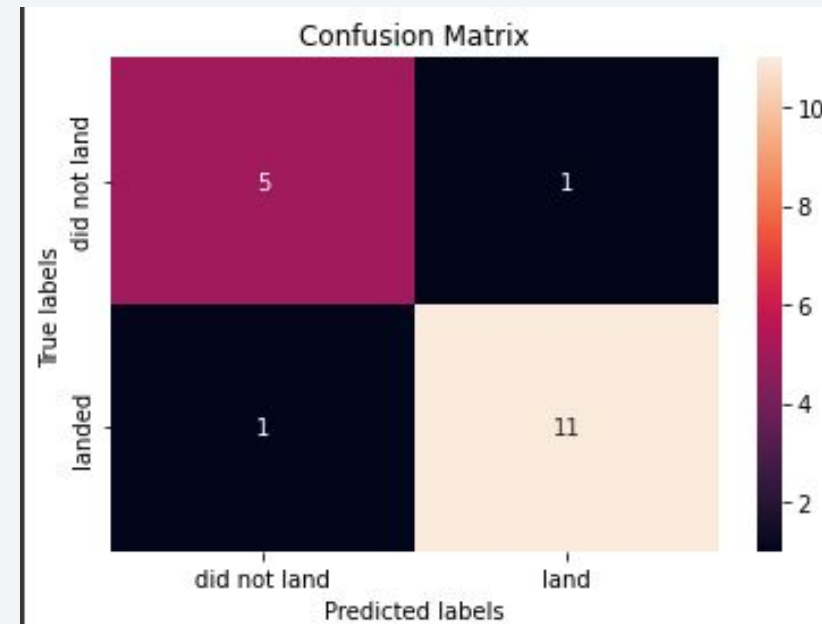
Classification Accuracy

- From the chart we can see that the tree model has the highest accuracy on both the training set and the test set, we can thus choose this model for predictive analysis.



Confusion Matrix

- From the confusion matrix we can remark that the tree model guessed all the flights that did land expect one, and the same with the flight that did not land, the model guessed them all except one. And thus the tree model misclassified only two cases.



Conclusions

- We can conclude from the exploratory data analysis that there exist patterns in the data, that can help us to predict the outcome of a launch.
- We can also say the best launch site is KSC LC-39A
- The yearly success rate is increasing over time
- Classification trees are the best solution to our model for predicting the outcome of a launch
- Payload mass is a very important feature, we can set a threshold to determine the outcome of a launch based on the payload mass.

Appendix

- [Github repo link](#)

Thank you!

