IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicolas MARAIS
27/01/2022

Outline

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

Executive Summary

- In order to provide a thorough analysis, we proceeded steps by steps: starting from Data Collection using SpaceX API and Web Scrapping to Data Wrangling to get a glimpse of the data. Then we achieved Exploratory Data Analysis (EDA) with Data Visualization (plots) and SQL. Next was building an interactive Map with Folium to highlights key information about launches and sites. Visualizing key data about launch sites and Payload was achieved thanks to building a Dashboard with Plotly Dash. Eventually, we compared and built a predictive classification model based on the datasets at our disposal.
- After computing all gathered data and insights we can consider a list of potential commercial orbits such as ISS, LEO, SSO, VLEO and GTO from an eastern launch site (KSC and/or CCAFS) and PO from a western launch site (VAFB). Along with these orbits, multiple booster versions can be considered depending on the payload: FT, B4, Heavy. We also recommend a further analysis about the Falcon 9 Block 5 that has shown promising performances on late SpaceX launches.

Introduction

Space Y is the newest sub-orbital and orbital reusable rockets manufacturer founded by Allon Mask.

Space Y wants to provide competitive launches and bid against the actual leader, Space X.

In order to achieve our goals, we plan on reusing Falcon 9 stage 1 rockets to minimize our overall launch costs using Data Science and Machine Learning models.

Therefore, we need to be able to assess :

- Which parameters are relevant to a successful landing.
- If a rocket will be land successfully or not.

Section 1

Methodology

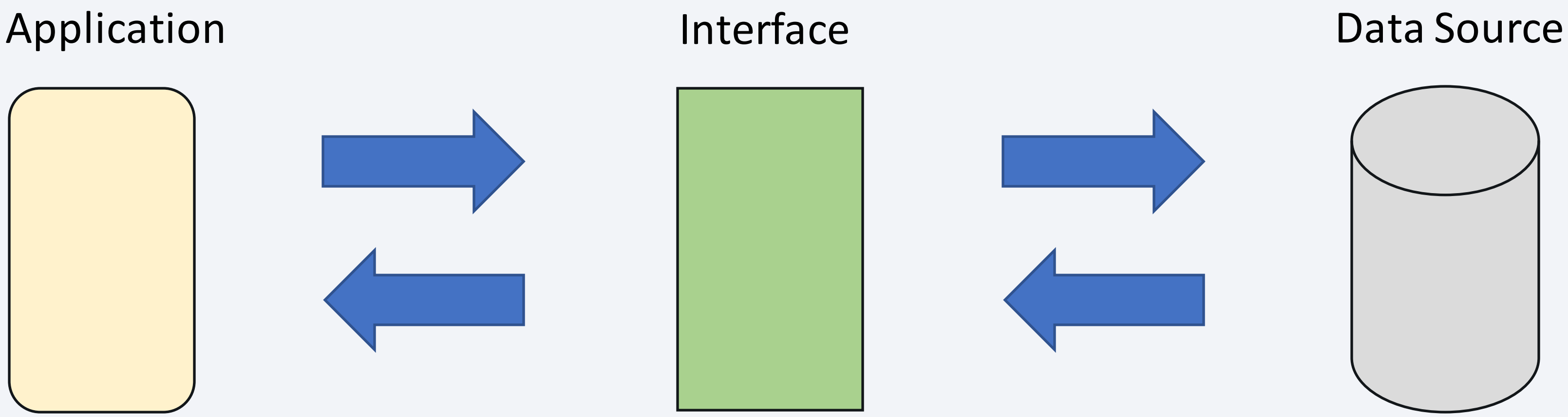
Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

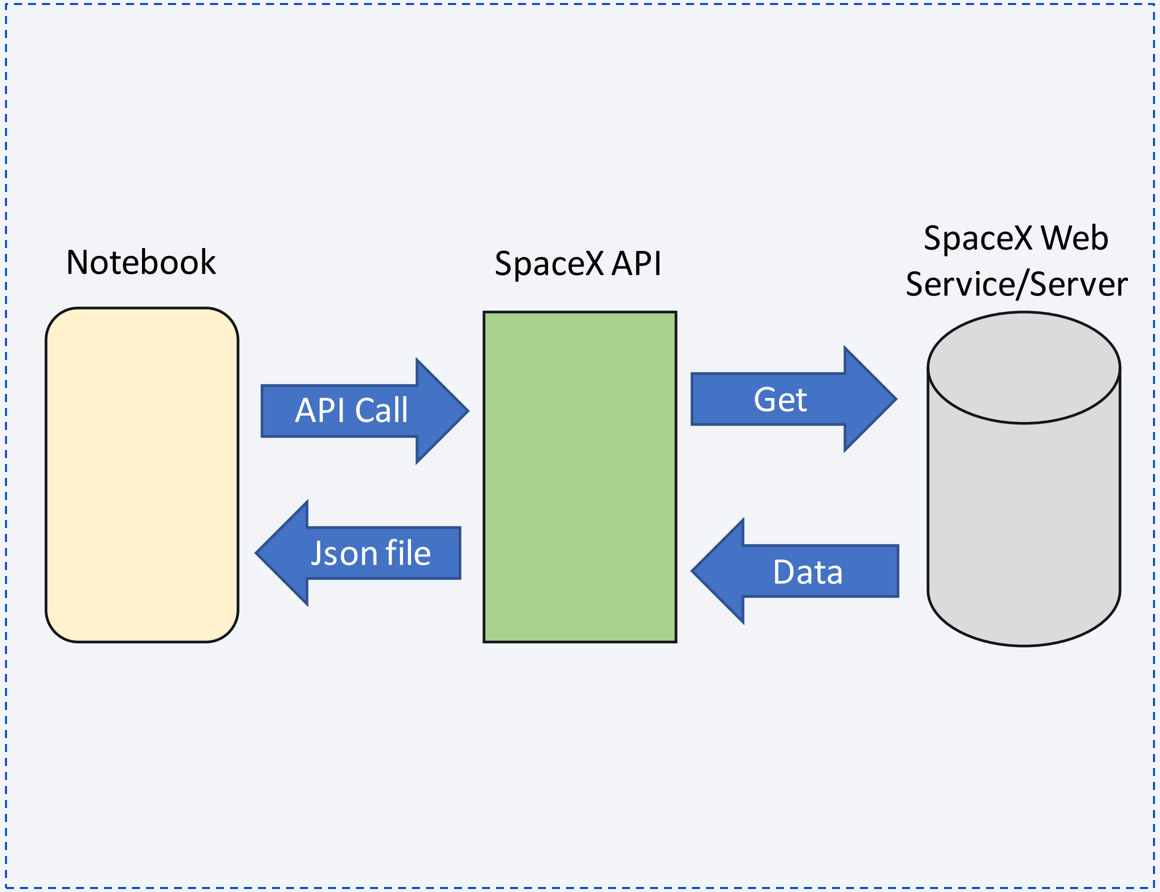
- Data sets were collected using 2 methods : SpaceX API and Web scrapping
- To collect data from a distant source, you need to call upon an interface (API or Library) that will interact with the data source



Data Collection – SpaceX API

- The notebook makes a get request to SpaceX API (= API Call)
- SpaceX API performs the get request to SpaceX Web Service/Server that sends back data
- SpaceX API formats the data into a Json file and forwards the result to the notebook

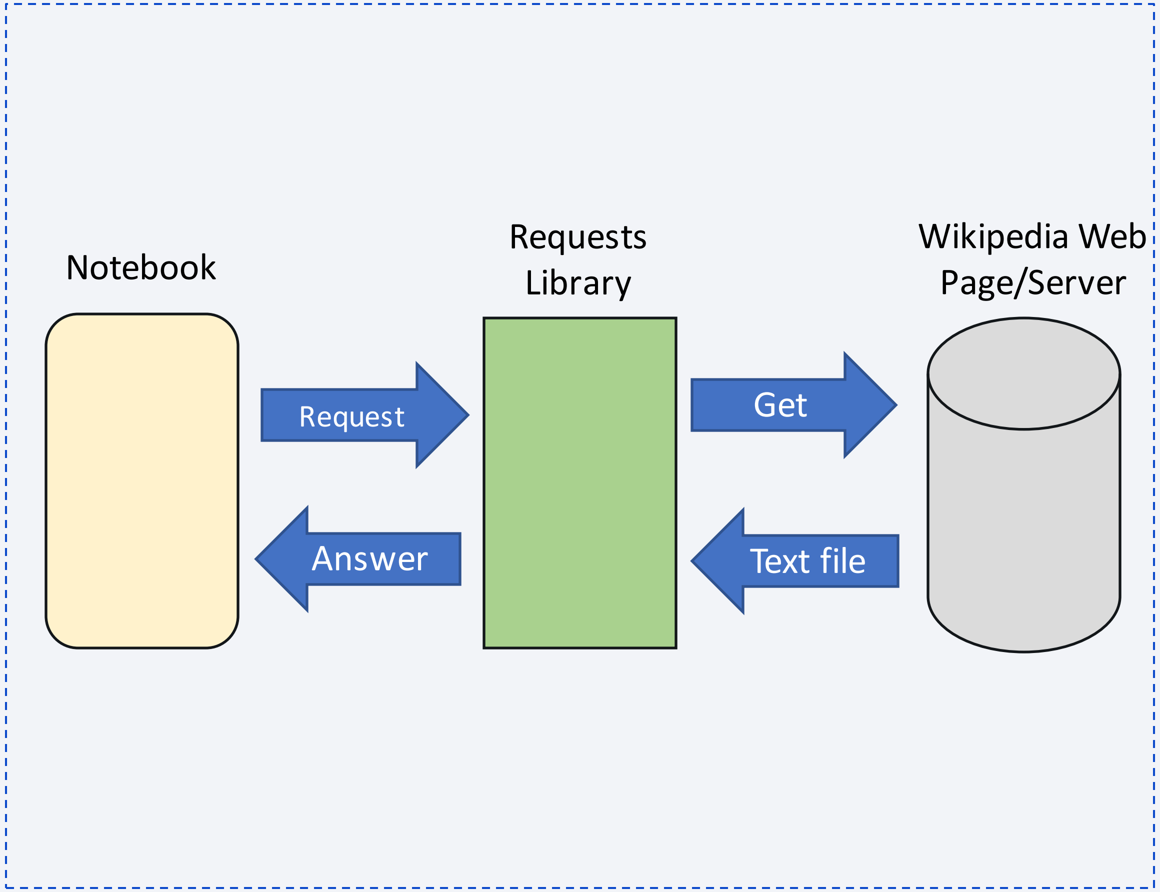
GitHub URL of the completed SpaceX API calls notebook



Data Collection - Scraping

- The notebook call upon the Requests library to execute a get method
- The Requests execute the get() toward the Wikipedia Server that send back the Webpage as text file
- The Requests library makes the text file available for the notebook to be used

GitHub URL of web scraping notebook



Data Wrangling

- In order to gain knowledge about the data set we ran some checks :
  - Null values : there was no null values to handle before going any further
  - Data Types (columns.dtypes) : which column is numerical or categorical
- Now that we have a rough idea about our data set, let's take a look at a few data:
  - Number of launches for each site
  - Number of launches for each type of orbit
  - Number of each type of outcome
  - Create a "Class" column that tells if the landing outcome is a success (1) or a failure (0)

GitHub URL of Data Wrangling related notebook

EDA with Data Visualization

- Scatter plot of Flight Number vs Launch site to figure out the repartition of all fight between all sites and the success rate's progression over time
- Scatter plot of Launch site vs Payload Mass to observe how the payload is distribute between sites
- Bar chart of Orbit vs Success Rate to compare every orbit success rate against each other
- Scatter plot of Flight Number vs Orbit to observe the advancement of success rate over time per orbit
- Scatter plot of Payload Mass vs Orbit to determine which payload correspond to which orbit
- Line chart of the success rate vs Years to evaluate the progression of the global success rate over the year

GitHub URL of EDA with data visualization notebook

EDA with SQL

List of SQL queries objectives :

- Select all launch site names
- Select 5 launch site names that begin with 'CCA'
- Count the total Payload Mass
- Calculate the Average Payload Mass for F9 v1.1
- Select the date of the 1st successful Ground Landing
- Select the booster version for successful Drone ship landing with a Payload Mass between 4000 and 6000
- Count the total number of Successful/Failure outcome
- Select the booster versions that carried the Max Payload Mass
- Select all 2015 failure records for Drone Ship landing
- Rank all landing outcomes from 2010-06-04 to 2017-03-02

GitHub URL of EDA with SQL notebook

Build an Interactive Map with Folium

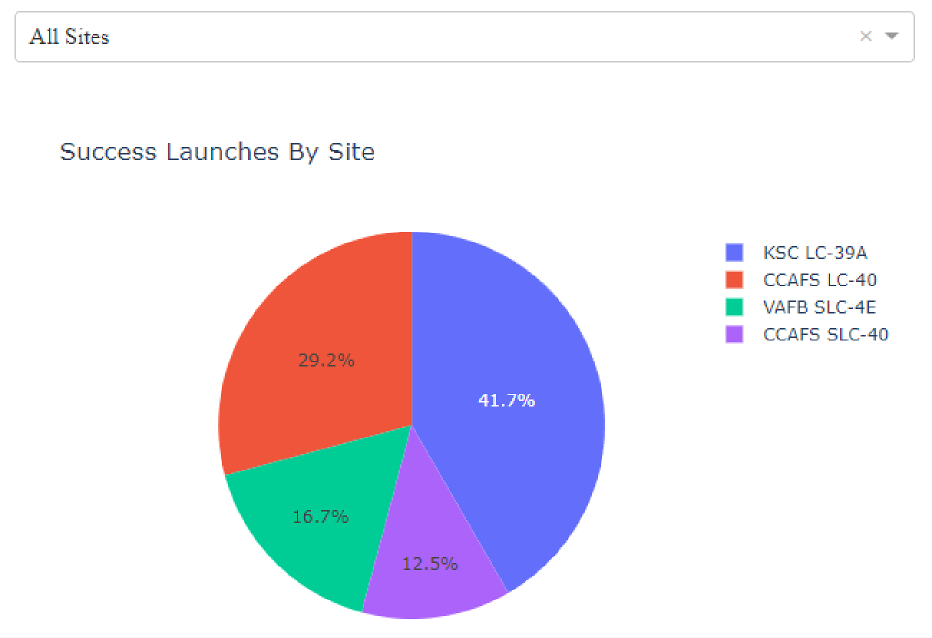
- A various number of map objects have been added to a Folium Map in order to highlight geographical characteristics:
  - Marker : used to visualize notorious location such as launch sites, cities, highways, trainrails, coastline, point of interest, etc
  - Circle : used to highlight a specific area (the area around each launch site in our case)
  - Cluster : used to group together outcome markers as there are really close to each other, we linked them to their relative launch site
  - Line : Highlights distance between a given site and neighboring cities, highways, coastline, touristic sites, etc.

GitHub URL of <u>Interactive Map with Folium map notebook</u>

Build a Dashboard with Plotly Dash

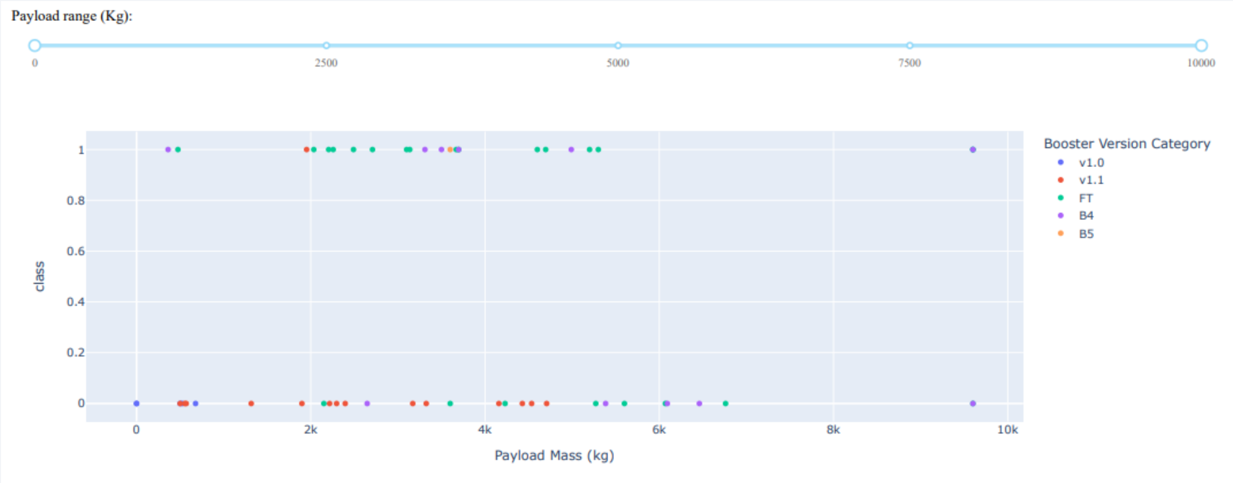
- We use a pie chart to visualize the success launches by site either :
  - All successful launches against all sites
  - Ratio success/failure by site
- We can select all or a specific site with the dropdown menu

GitHub URL of the interactive visual Plotly Dash file



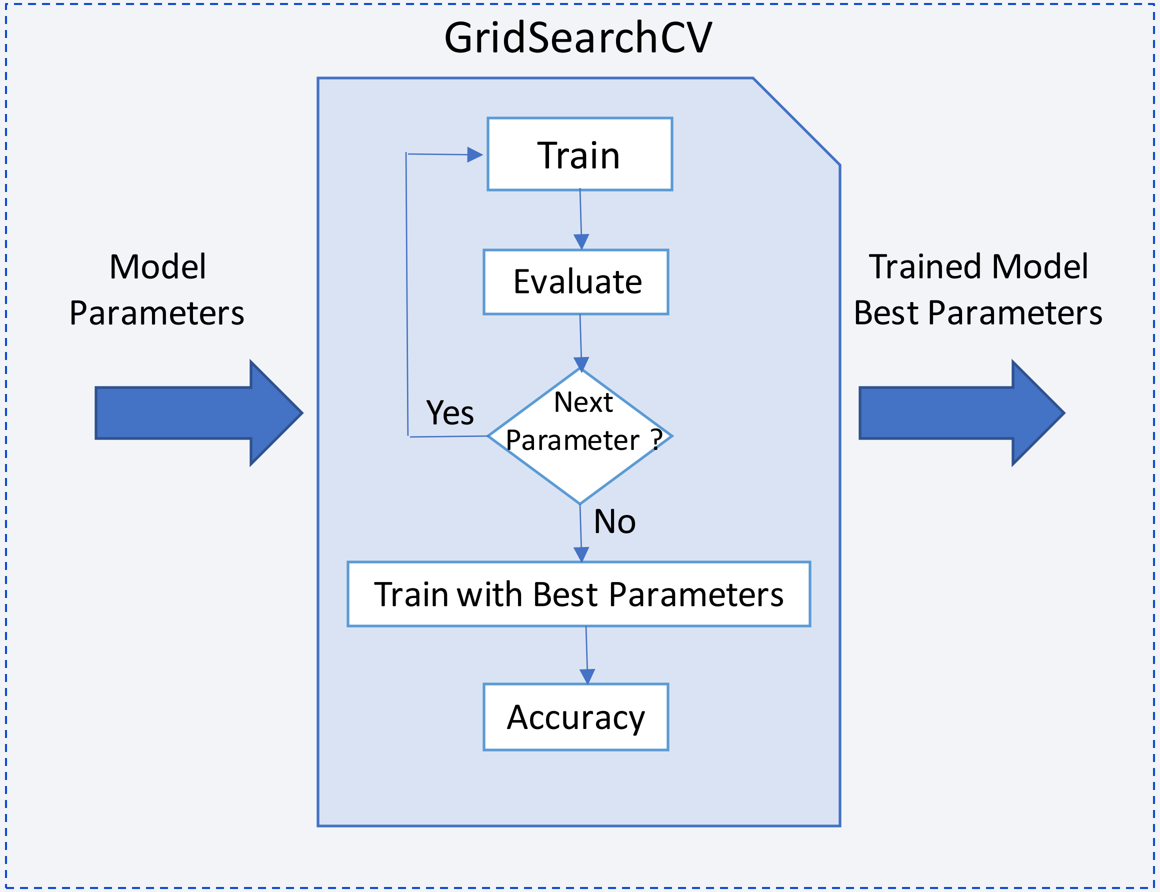
Build a Dashboard with Plotly Dash

- A scatter plot is used to observe the outcome of each launch depending on the payload mass and the booster version
- A slider is used to custom the range payload mass we want to visualize (from 0 to 10k KG)
- Source code file of the interactive visual Plotly Dash



Predictive Analysis (Classification)

- We used GridSearchCV from scikit-learn library to build, evaluate and improve multiple models and choose the best performing one out of them
- We input a model with parameters (like metrics and cv folds) and GridSearchCV returns a fitted model with the best parameters and score to us.
- GridSearchCV will train the given model by iterating the initial parameters n times (=cv folds) then select the best parameter by the given metrics and train the model with it before returning the result.
- GitHub URL of the predictive analysis lab



Results

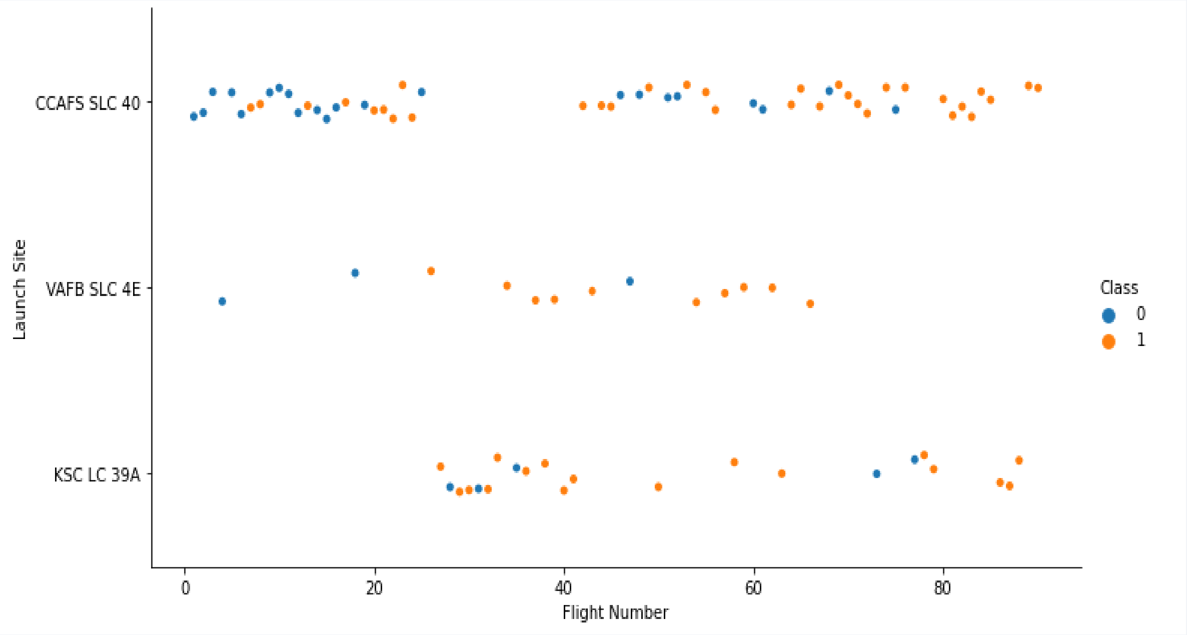
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

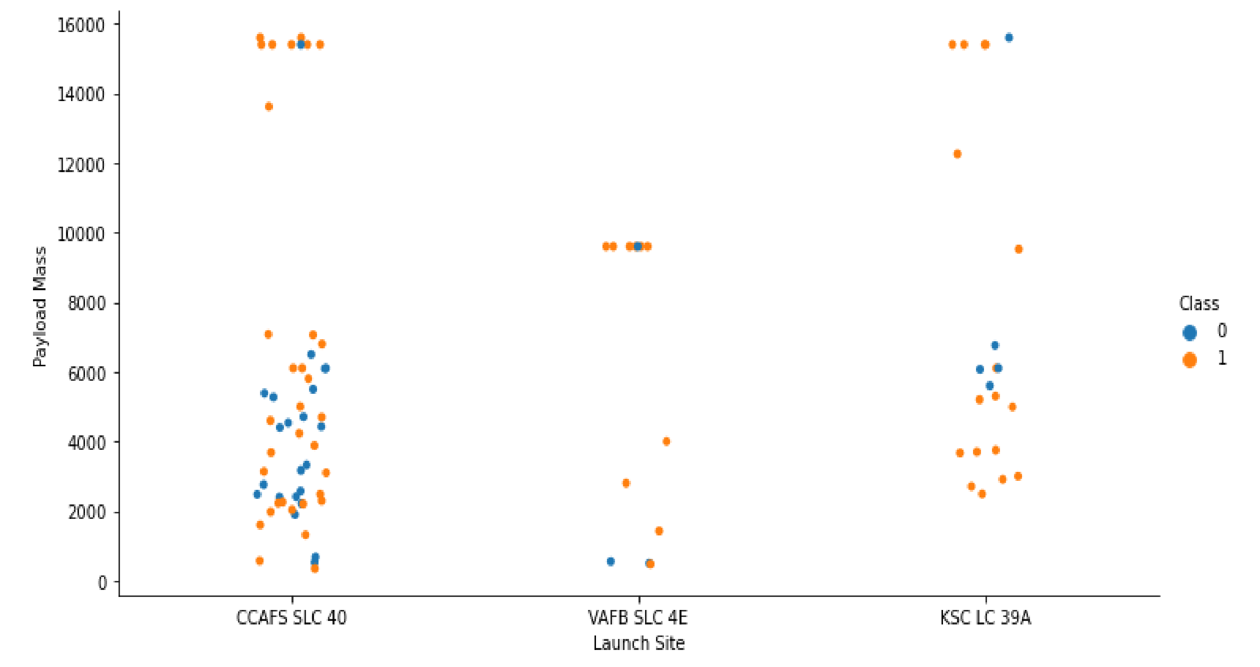
Flight Number vs. Launch Site

- The first flight occurred from CCAFS SLC-40, this site has higher launch count than the other 2 sites.
- VAFB SCL-4E has the lower number and frequency of flights then its circumstances might be special.
- KSC LC-39A was the last site used and we can observe that it seems to be used alternately with CCAFS SLC-40.
- The success rate increases with the number of launches therefore built-up experience increases the success rate



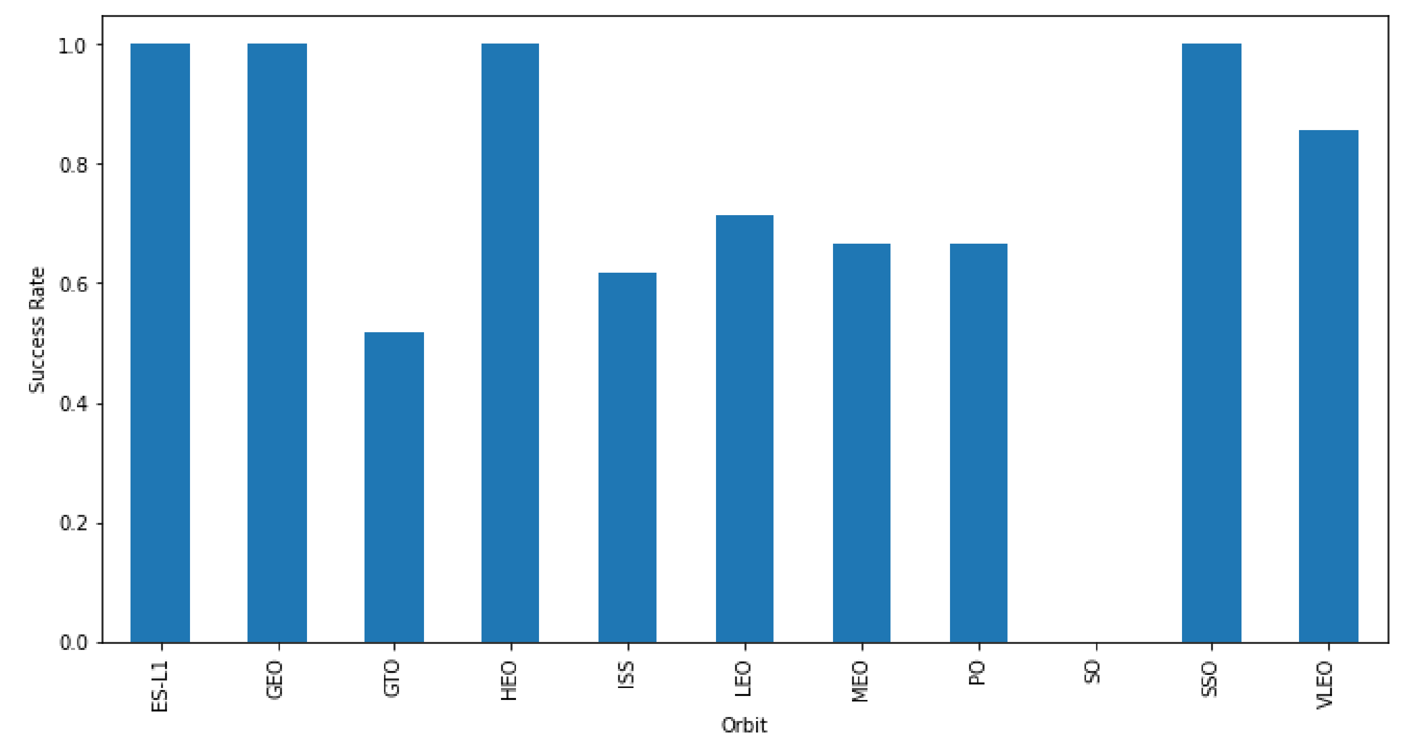
Payload vs. Launch Site

- There is no launches for heavy payload mass (> 10k KG) from VAFB SLC 4E site.
- We can note that heavy payload launches tends to have a better success rate that lighter payload launches.



Success Rate vs. Orbit Type

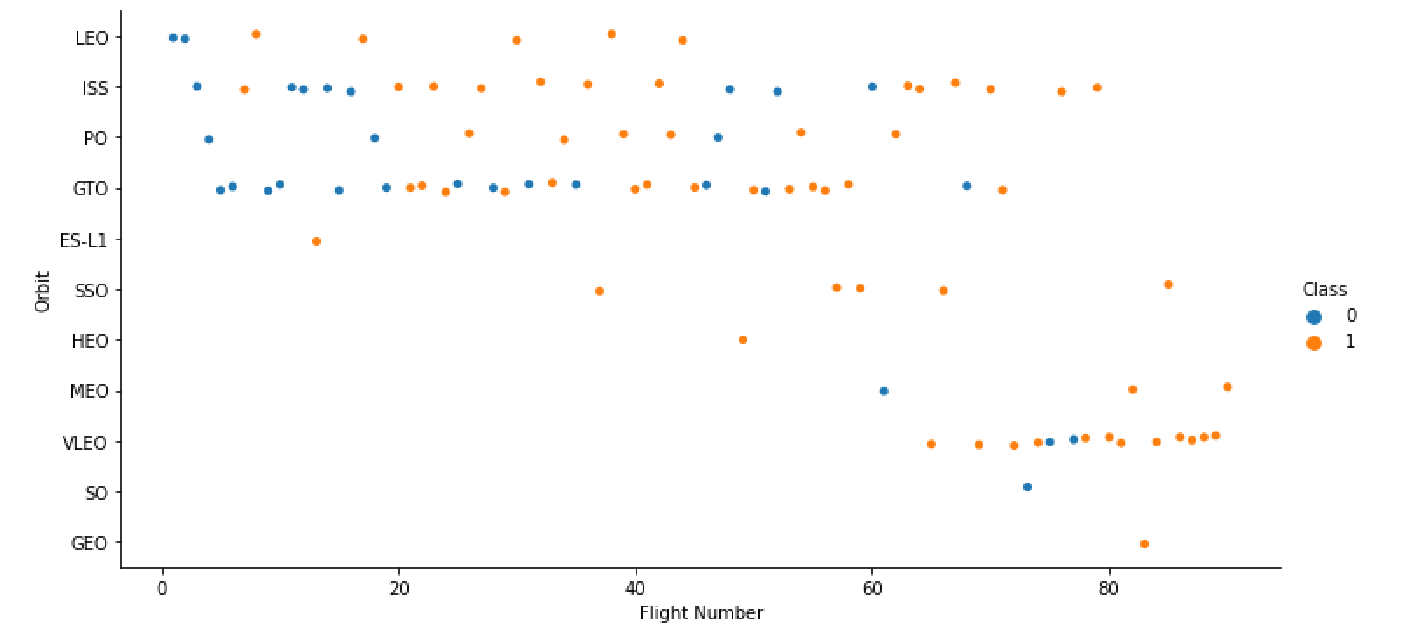
- During data wrangling, we noted that ES-L1, HED, SO and GEO have performed only one launch each, we would need more data to make sure the outcomes are relevant to the orbit
- The success rate is mostly over 60%, except for GTO



Flight Number vs. Orbit Type

- LEO orbit success appears related to the number of flights, as for PO and MEO orbits.

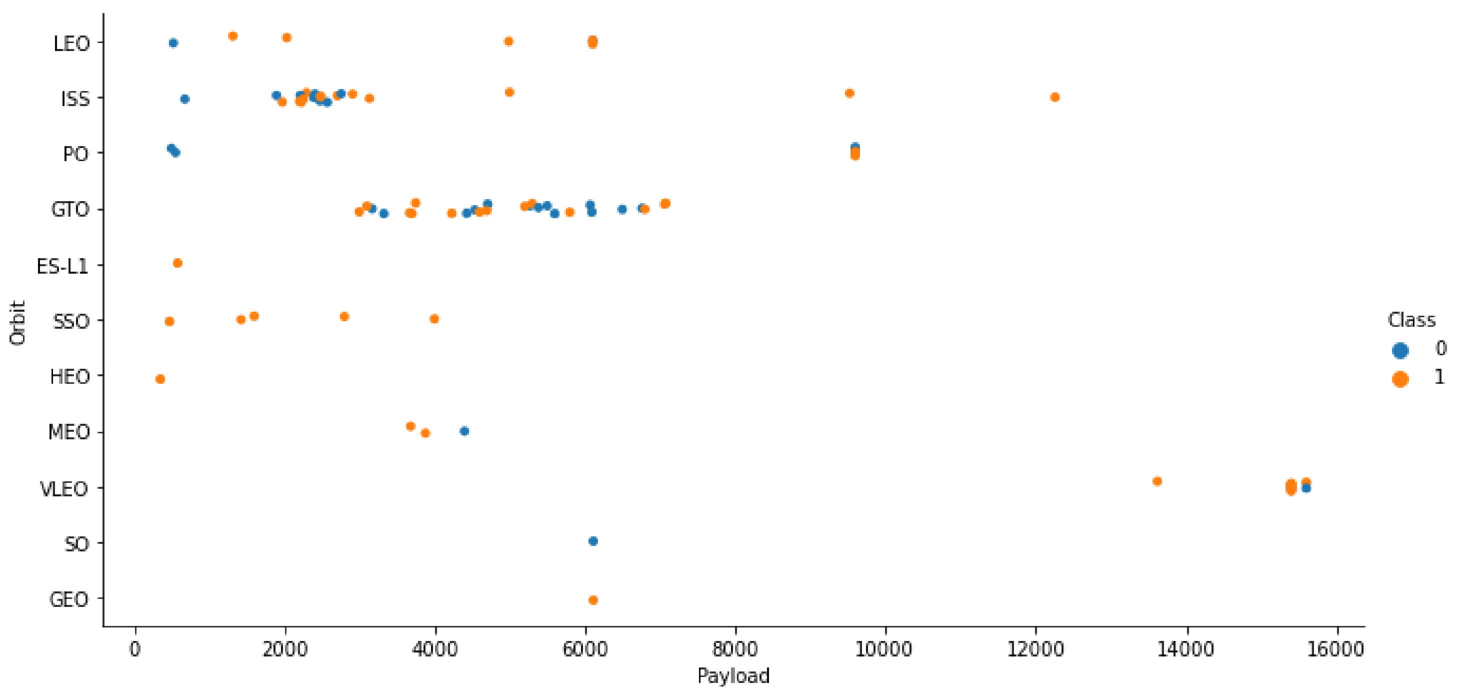
- There seems to be no relationship between flight number when it comes to GTO, ISS or VLEO orbits.



Payload vs. Orbit Type

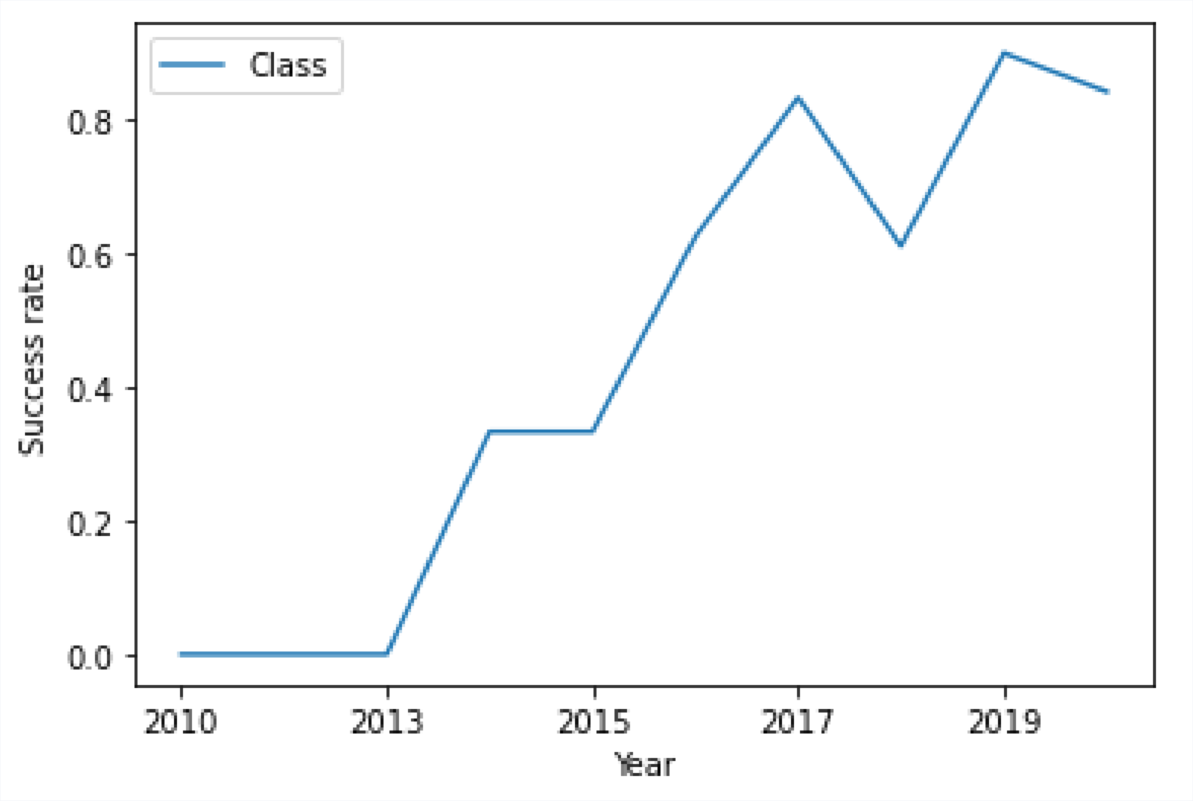
- ISS, PO and VLEO orbits appears to have a better success rate with heavy payloads

- GTO orbit success rate appears to have no relationship with the payload mass

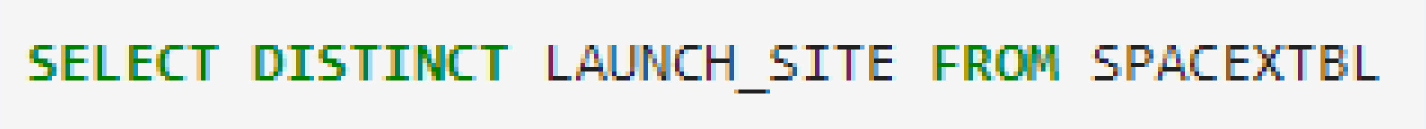


Launch Success Yearly Trend

- From 2013 to 2020, the overall success rate improved even if we can note 2 slight steps back (2018 and 2020)

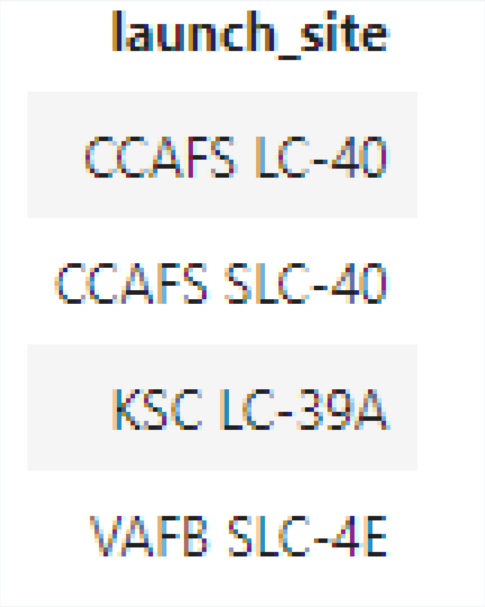


All Launch Site Names



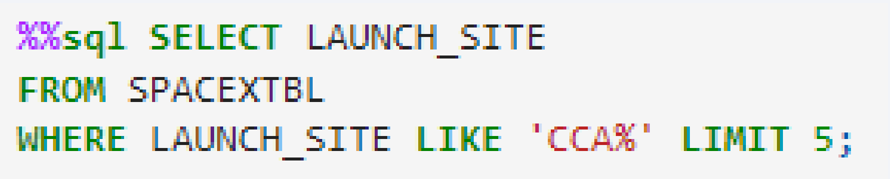
```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

- SELECT DISTINCT statement returns only different values (no duplicate)



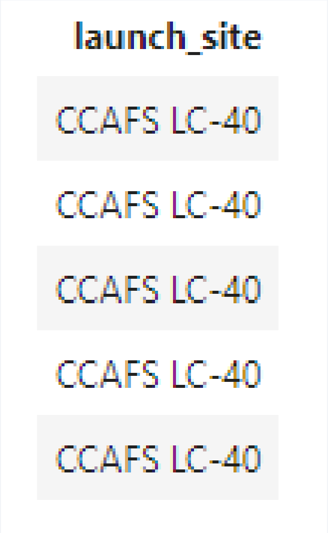
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'



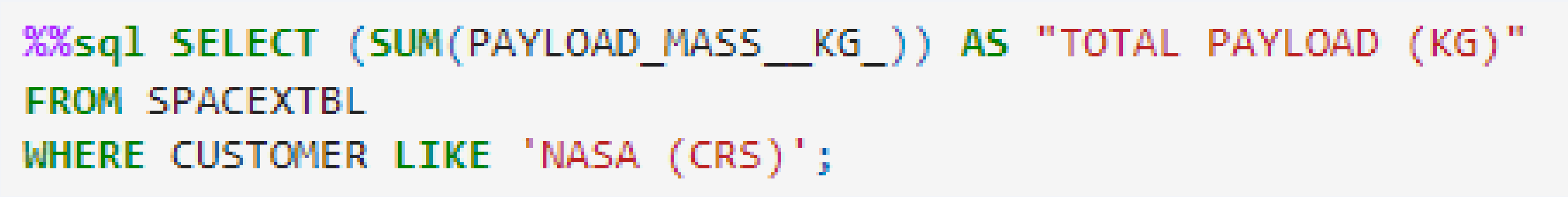
```
%%sql SELECT LAUNCH_SITE
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

- In a WHERE Clause, we use LIKE to compare strings and LIMIT 5 to fetch only 5 results



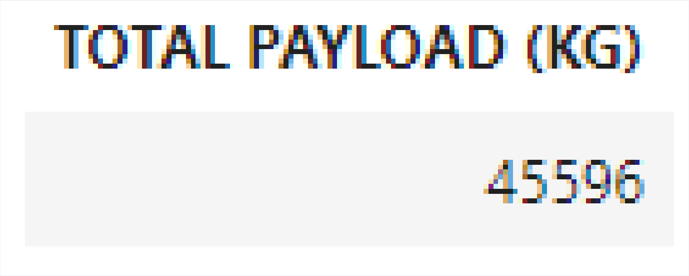
launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass



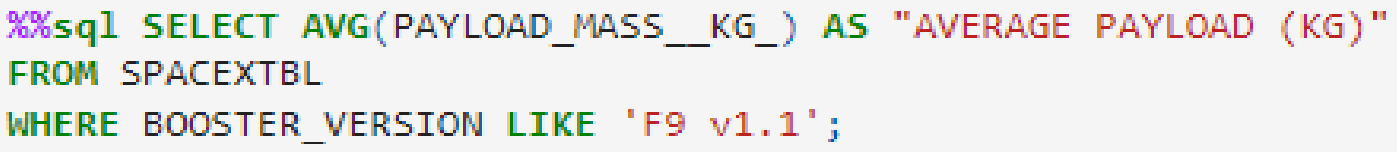
```
%%sql SELECT (SUM(PAYLOAD_MASS__KG_)) AS "TOTAL PAYLOAD (KG)"
FROM SPACEXTBL
WHERE CUSTOMER LIKE 'NASA (CRS)';
```

- The SUM function allows us to sum the values of a given column, the WHERE clause on the CUSTOMER column acts as a filter



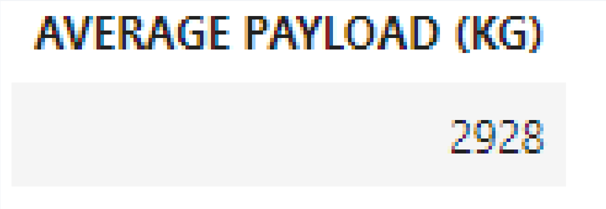
TOTAL PAYLOAD (KG)
45596

Average Payload Mass by F9 v1.1



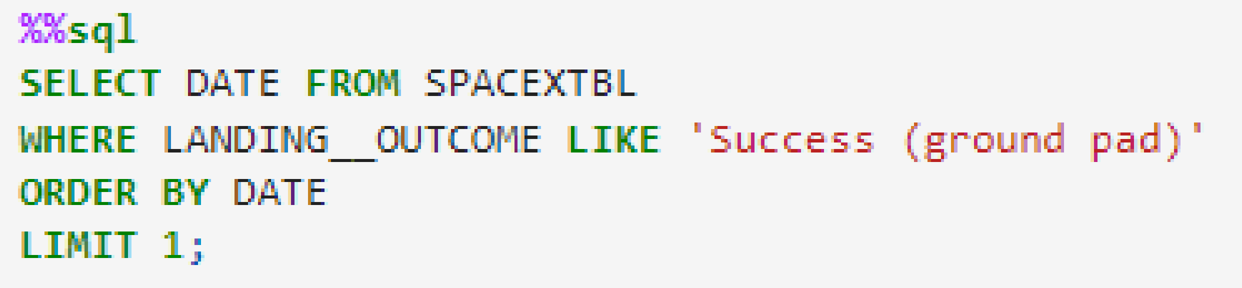
```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "AVERAGE PAYLOAD (KG)"
FROM SPACEXTBL
WHERE BOOSTER_VERSION LIKE 'F9 v1.1';
```

- The AVG function calculate the average value of a give column and the WHERE clause on the BOOSTER_VERSION column act as a filter for the given value



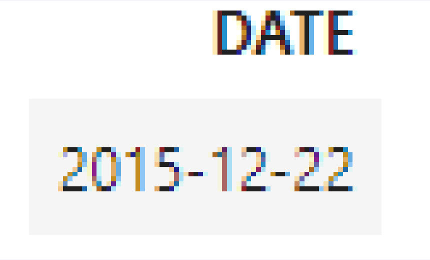
AVERAGE PAYLOAD (KG)
2928

First Successful Ground Landing Date



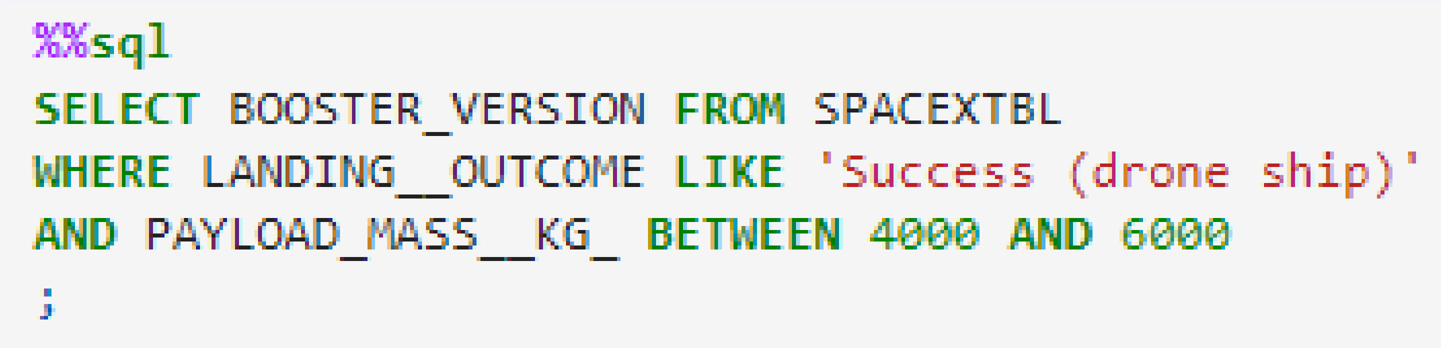
```
%%sql
SELECT DATE FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE 'Success (ground pad)'
ORDER BY DATE
LIMIT 1;
```

- The ORDER BY keyword is used to sort result-set (ascending order by default), the WHERE clause is used as a filter. LIMIT 1 is used to fetch only the 1st row.



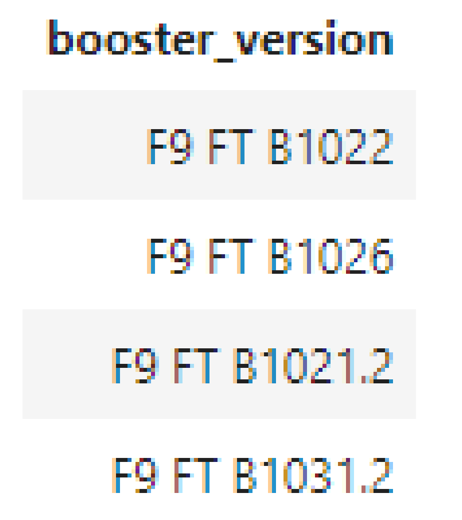
DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000



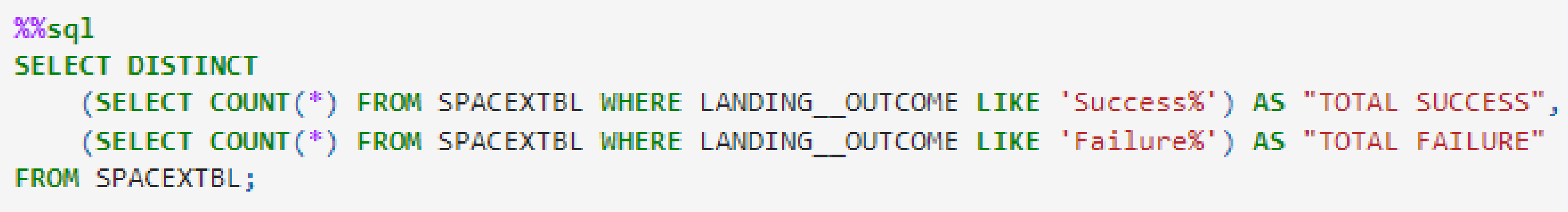
```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
;
```

- The WHERE clause is used as a filter for both LANDING__OUTCOME and PAYLOAD, with the keyword BETWEEN to specify the desired range.



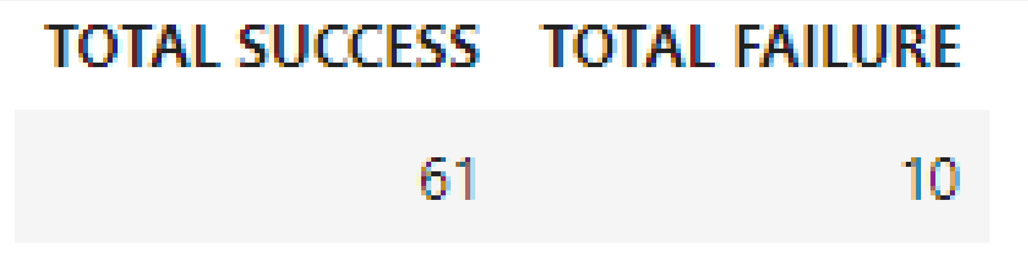
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes



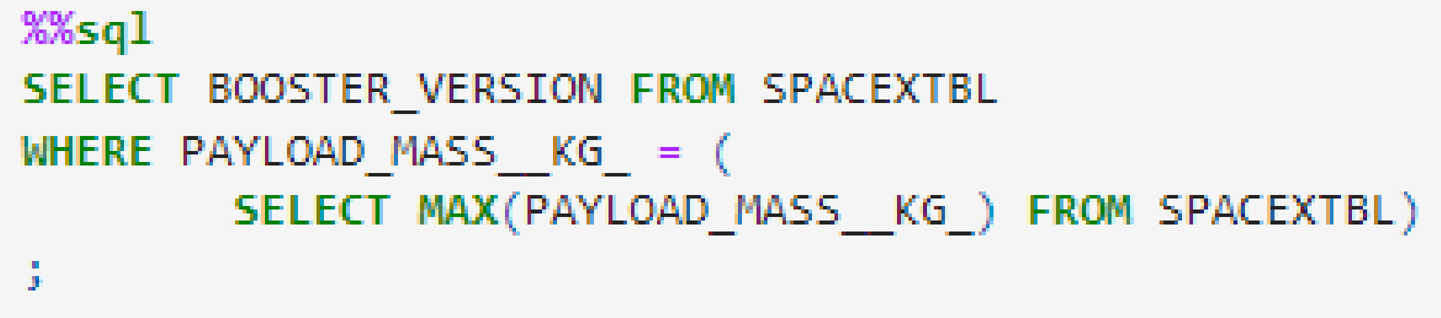
```
%%sql
SELECT DISTINCT
    (SELECT COUNT(*) FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success%') AS "TOTAL SUCCESS",
    (SELECT COUNT(*) FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Failure%') AS "TOTAL FAILURE"
FROM SPACEXTBL;
```

- To calculate each value, we need to do a subquery using the COUNT function and a WHERE clause to filter each LANDING__OUTCOME state. A SELECT DISTINCT statement is used to return only pairs of different values (= 1 row)

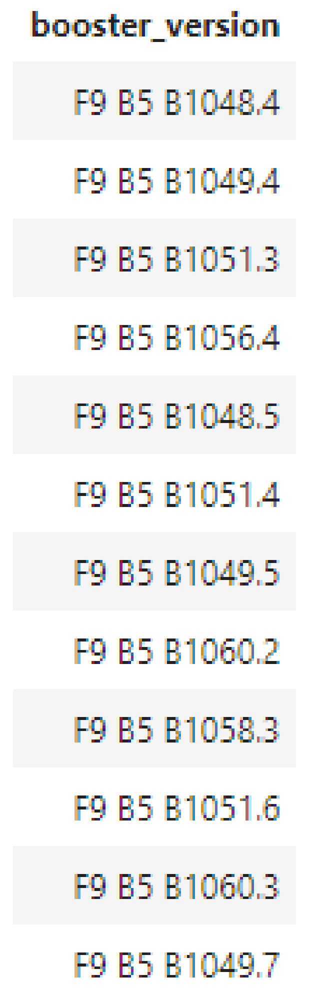


TOTAL SUCCESS	TOTAL FAILURE
61	10

Boosters Carried Maximum Payload



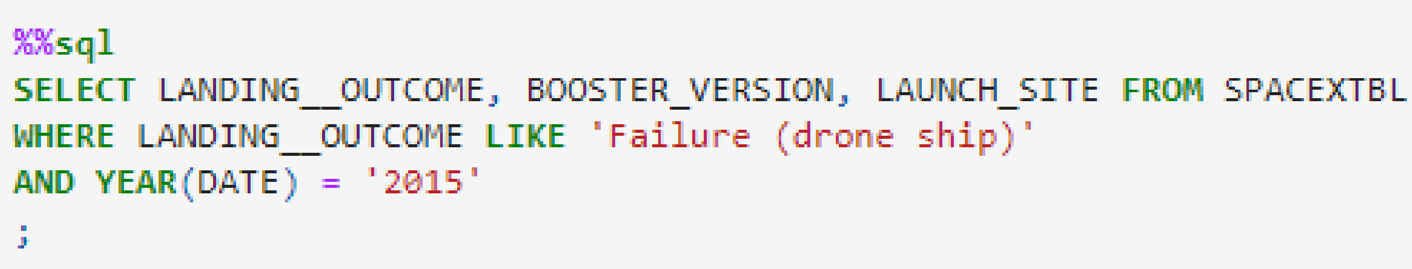
```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
        SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
;
```



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

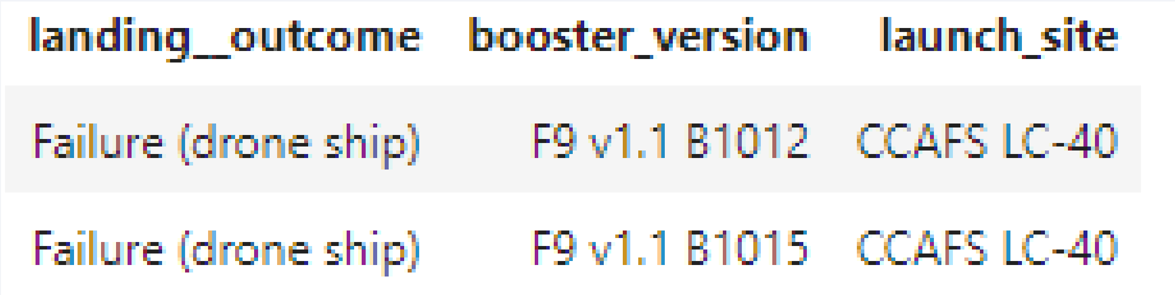
- In order to apply a filter using the WHERE clause, we need to use a subquery with the MAX function inside the WHERE clause to get the highest value for the PAYLOAD_MASS__KG_ column

2015 Launch Records



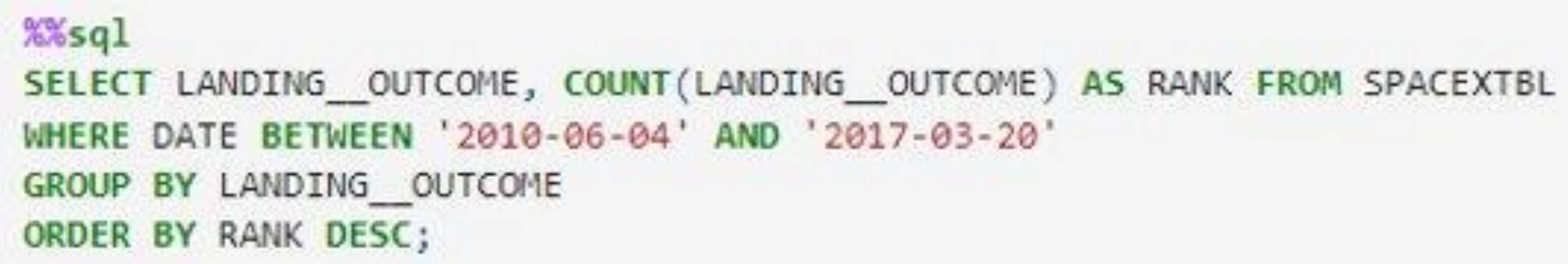
```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE 'Failure (drone ship)'
AND YEAR(DATE) = '2015'
;
```

- The WHERE clause is used to filter both LANDING__OUTCOME and DATE. We apply the YEAR function on the date column to use only the year as a filter parameter



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

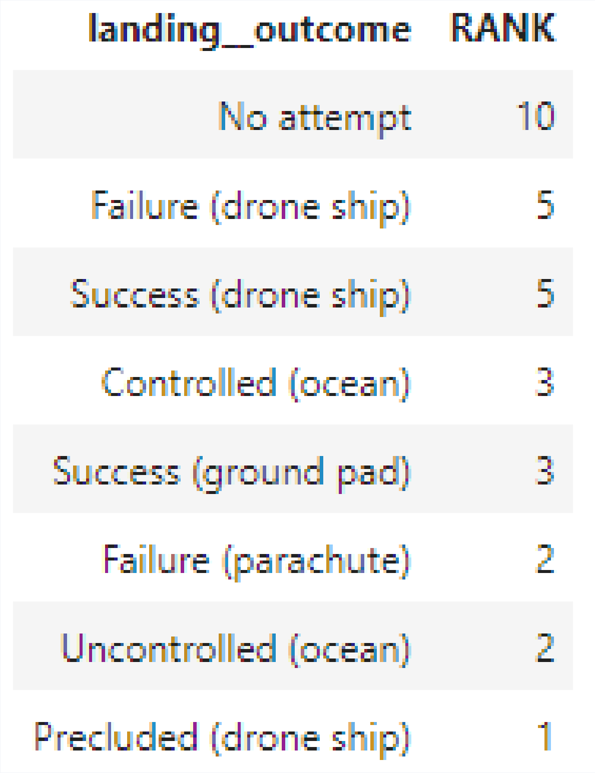
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20



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

- The COUNT function is used on LANDING__OUTCOME which is named RANK
- The WHERE clause with the BETWEEN keyword apply a filter on a range of dates
- The GROUP BY keyword aggregate the result set and the ORDER BY keyword is used with DESC

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



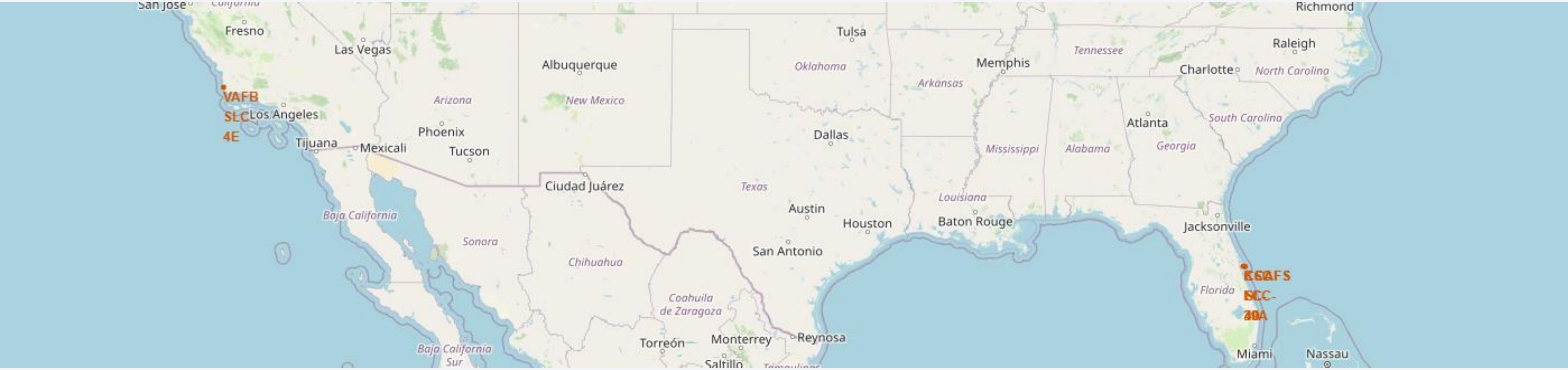
landing_outcome	RANK
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

SpaceX Launch Sites Location Map

- All launch sites are located near coastlines and in region with clear weather as it is a major parameter for a successful launch
- There are 2 sites on the eastern coast of Florida to use the Earth speed rotation at their advantage (easterly launches)
- There is 1 site on the coast of California that is used to launch rocket onto Polar orbits (southerly launches)

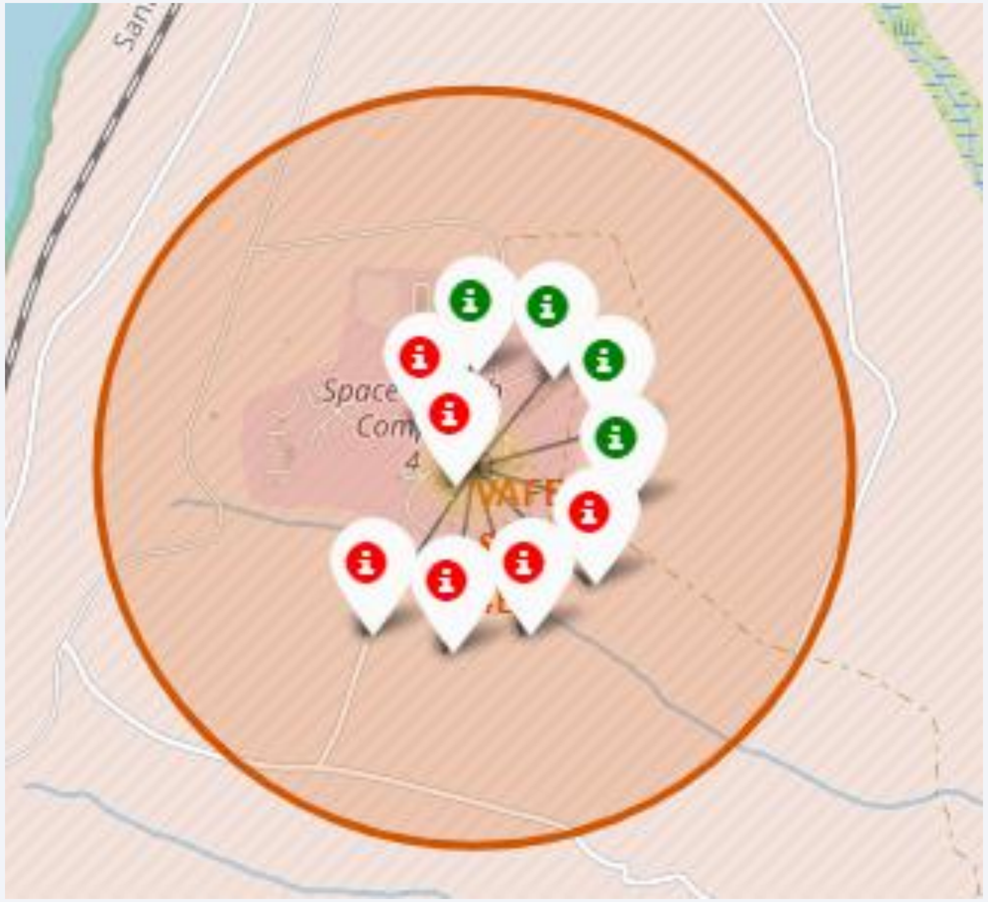


Please note that CCAFS SLC-40 is the same as CCAFS SLC-40 that has been renamed. Therefore I counted it as 1 site.

Launch Outcomes Map

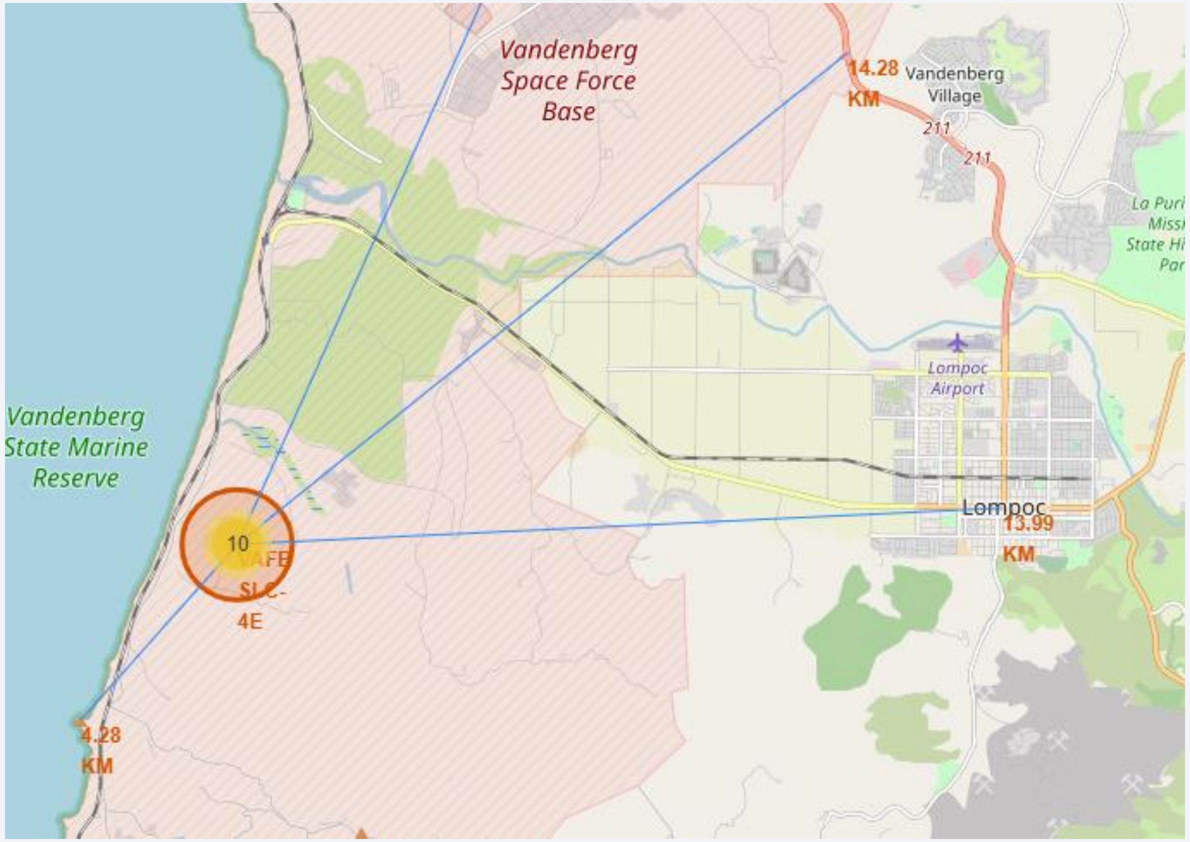
- The map highlights all launch outcomes for each site

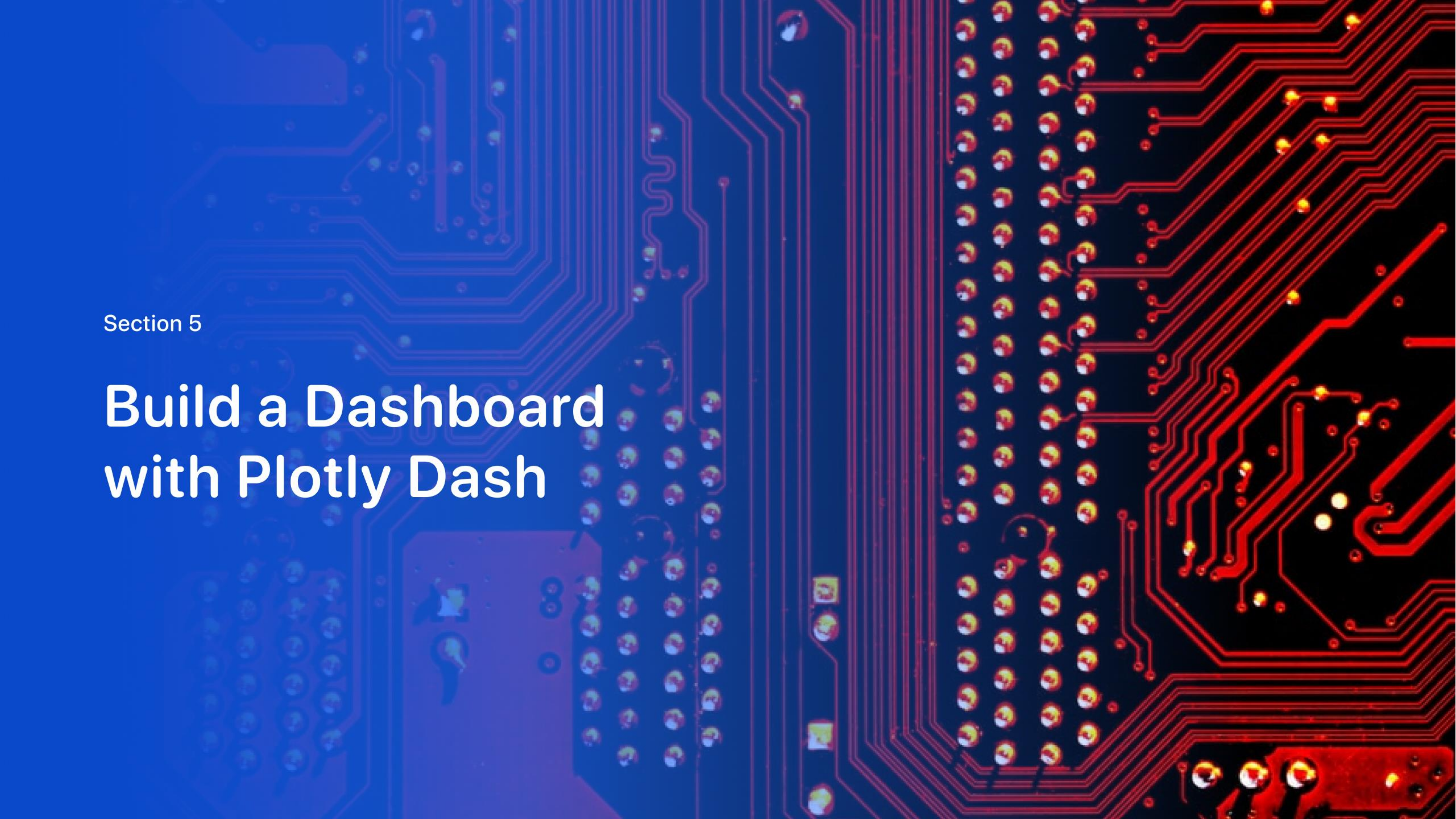
- Using the clustering feature from Folium, we placed green markers (success) and red markers (failure) for each launch to their respective launch site



Points of Interest proximity Map

- Launch sites are away from any non-essential launching facilities and buildings : Highways, Cities, Railways
- They are close to coastlines to limit any possible damage/danger to hinterlands due to failures.
- Launches are oriented toward water bodies.



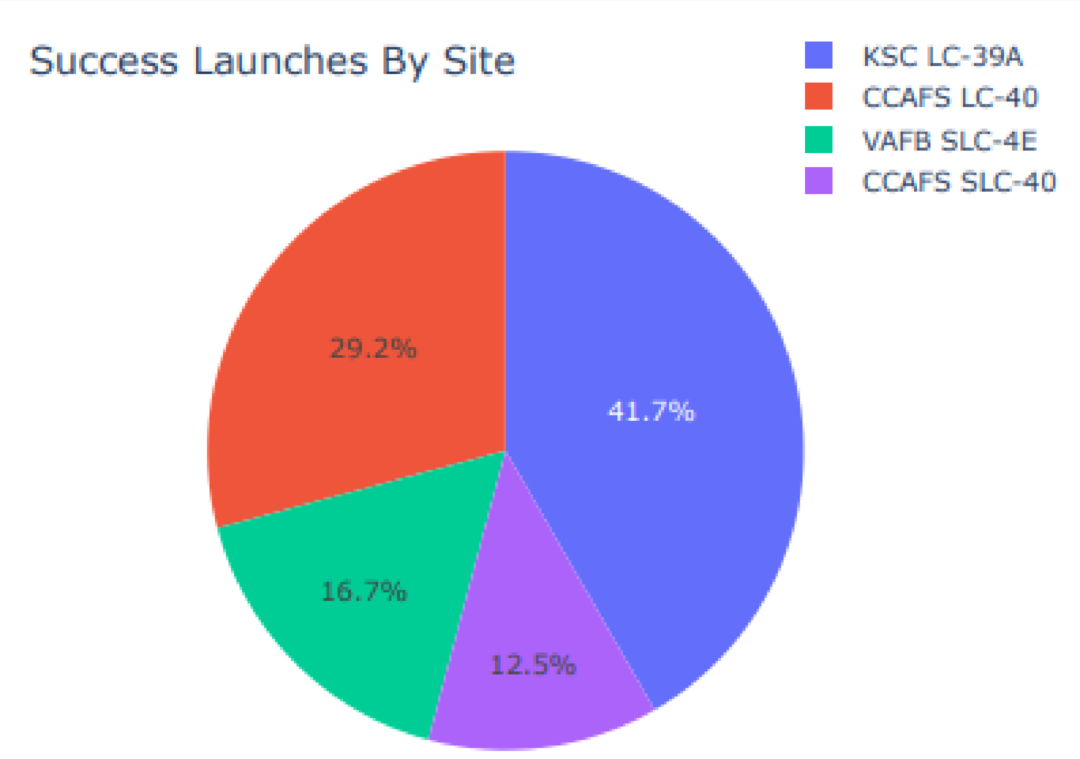


Section 5

Build a Dashboard with Plotly Dash

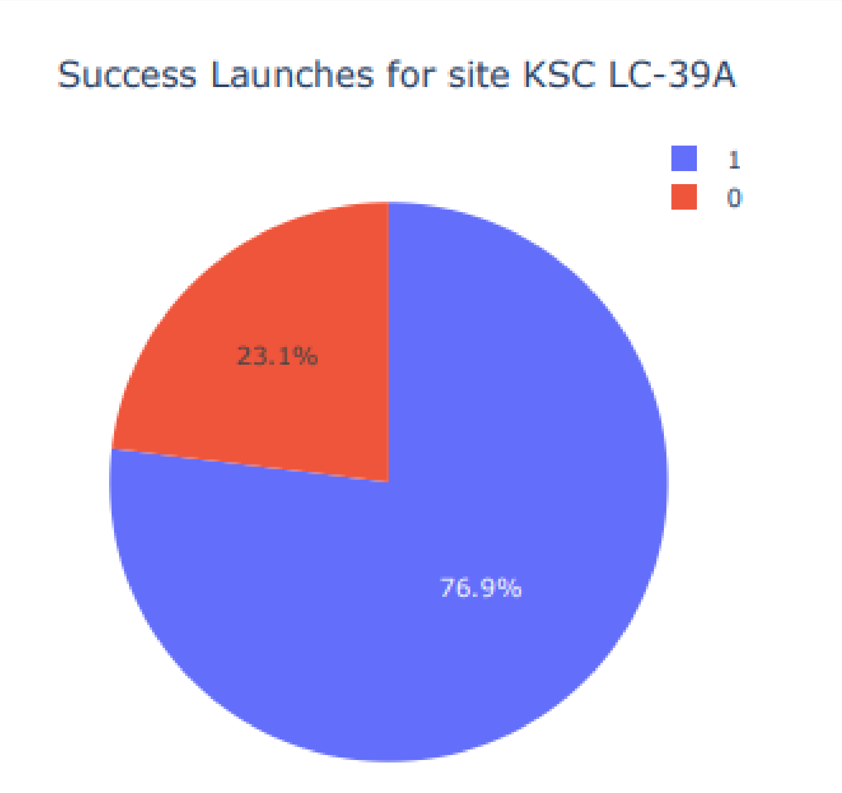
Launch Success by site

- We observe the repartition of the successful launches by site
- From this pie chart, KSC LC-39A appears to have the highest success rate
- If we take into consideration that the CCAFS LC-40 and SLC-40 are the same site that has been renamed, it has the same success rate as KSC LC-39A (41.7%)
- Furthermore, SLC-4E site that is located on the Californian coast has the lowest success rate (Polar Orbit launches)



Launch outcomes for KSC LC-39A

- KSC LC-39A has the best success ratio among all sites
- According to this pie chart, almost 77% of launches that happen at Kennedy Space Center are concluded by a successful landing of the rocket.



Relationship between Payload Mass and Booster Version

- As we can observe in Figures 1, 2 (slide 44) and 3 (slide 44):
  - V1.1 booster version has no record of a successful landing with the selected ranges
  - FT booster version has a high success rate with Payload < 5600 kg
  - Block 4 booster version has high success rate with Payload between 3000 and 5000 kg
  - Block 5 booster version has only 1 launch (success), that might be promising but we cannot conclude anything with only one record.

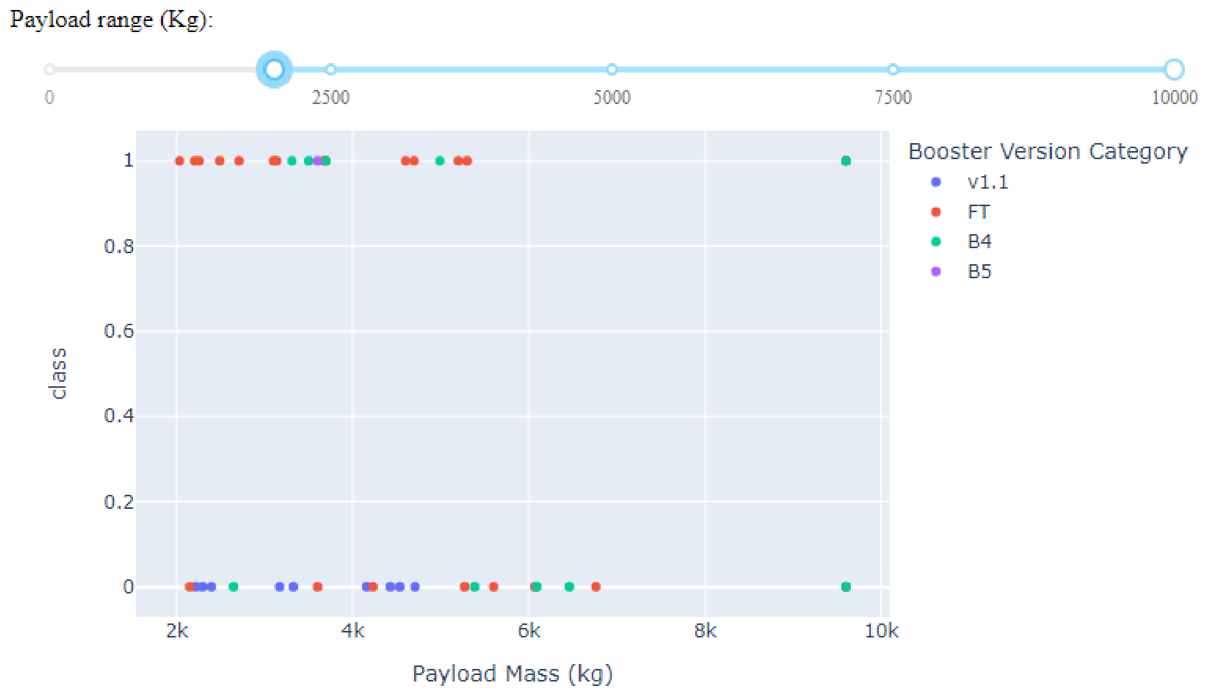


Figure 1

Relationship between Payload Mass and Booster Version

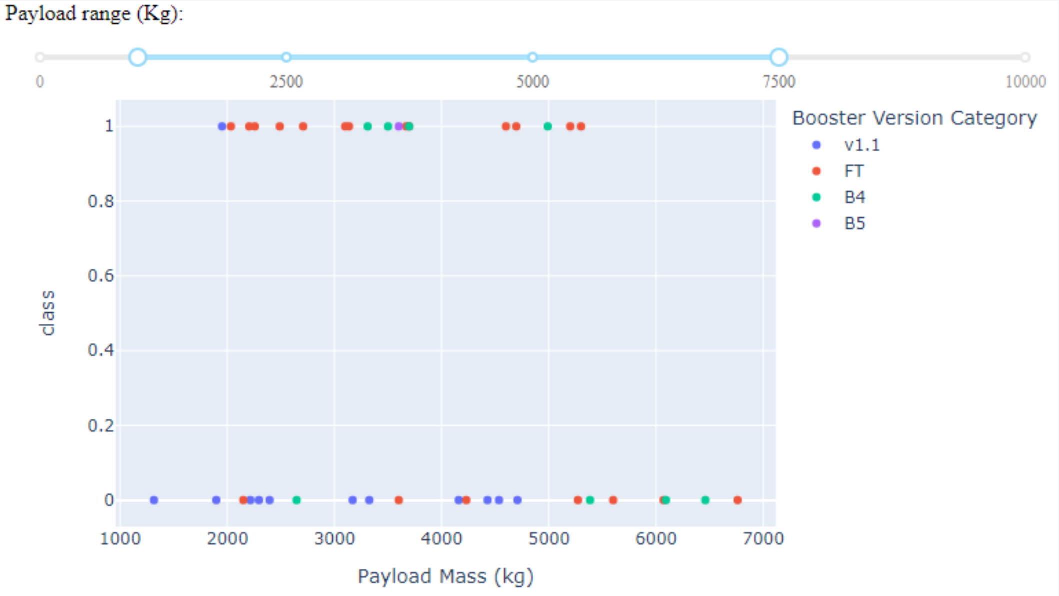


Figure 2

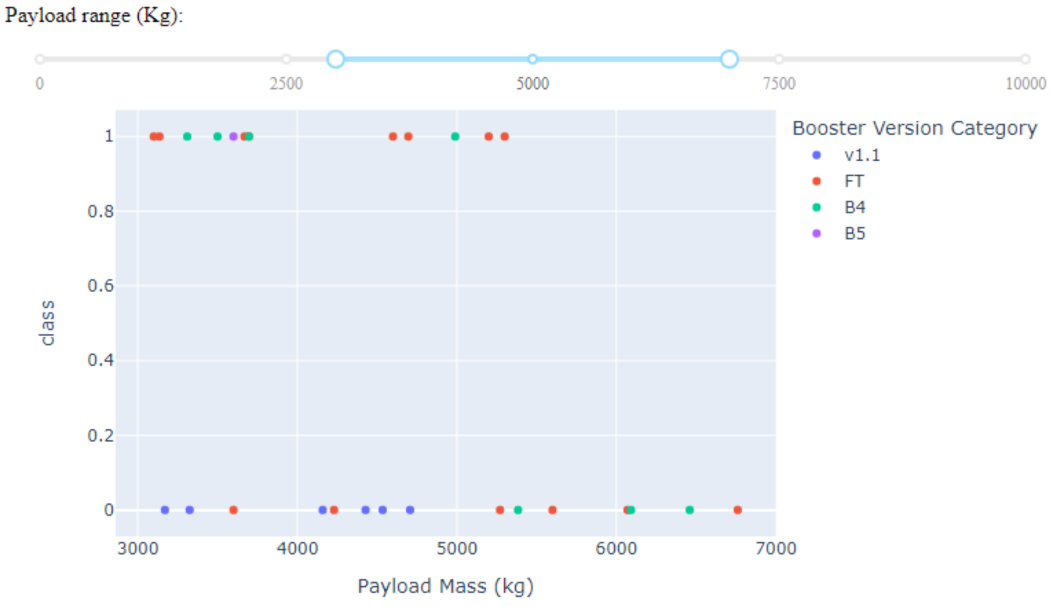


Figure 3

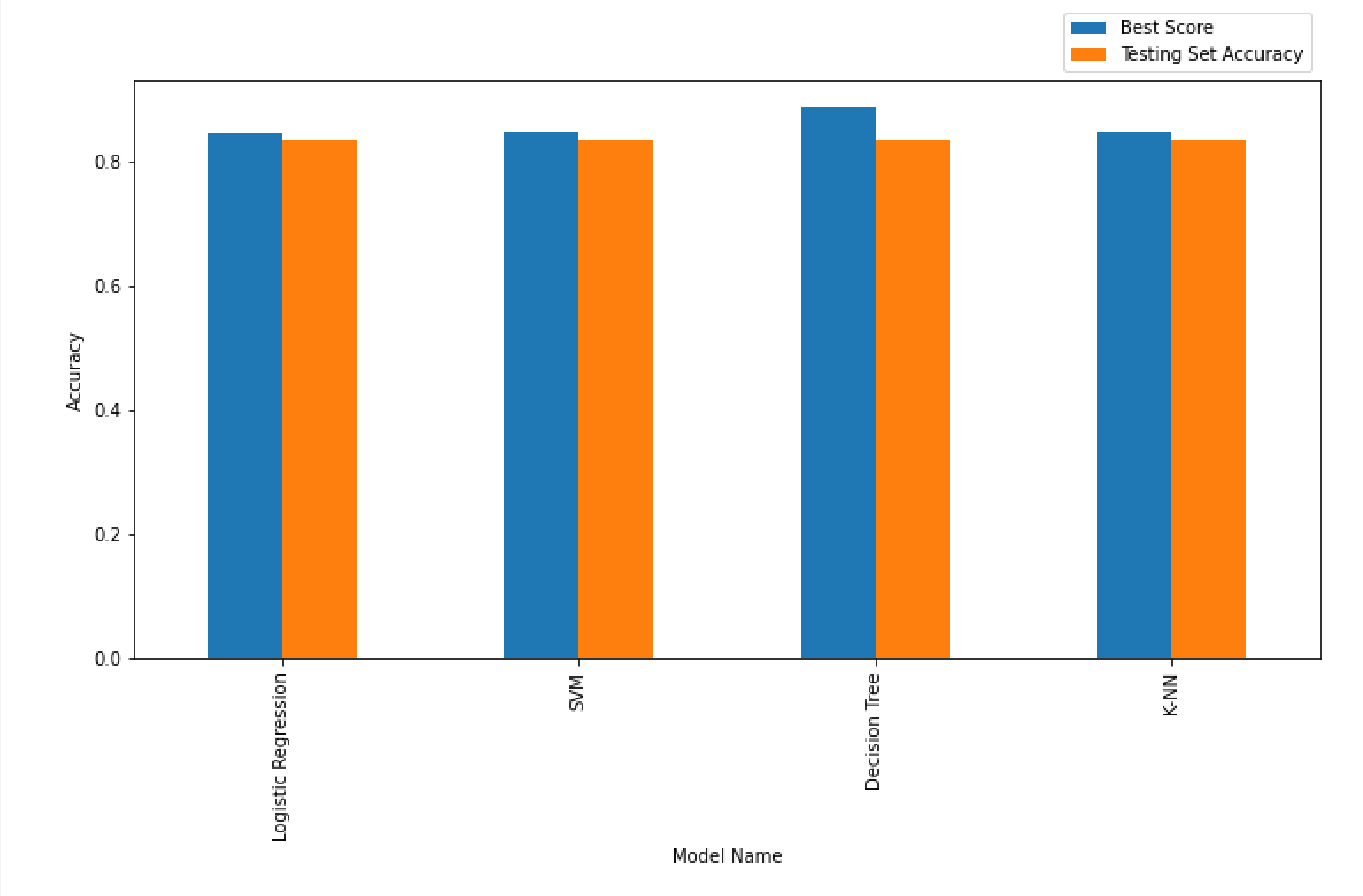
Section 6

Predictive Analysis (Classification)

Classification Accuracy

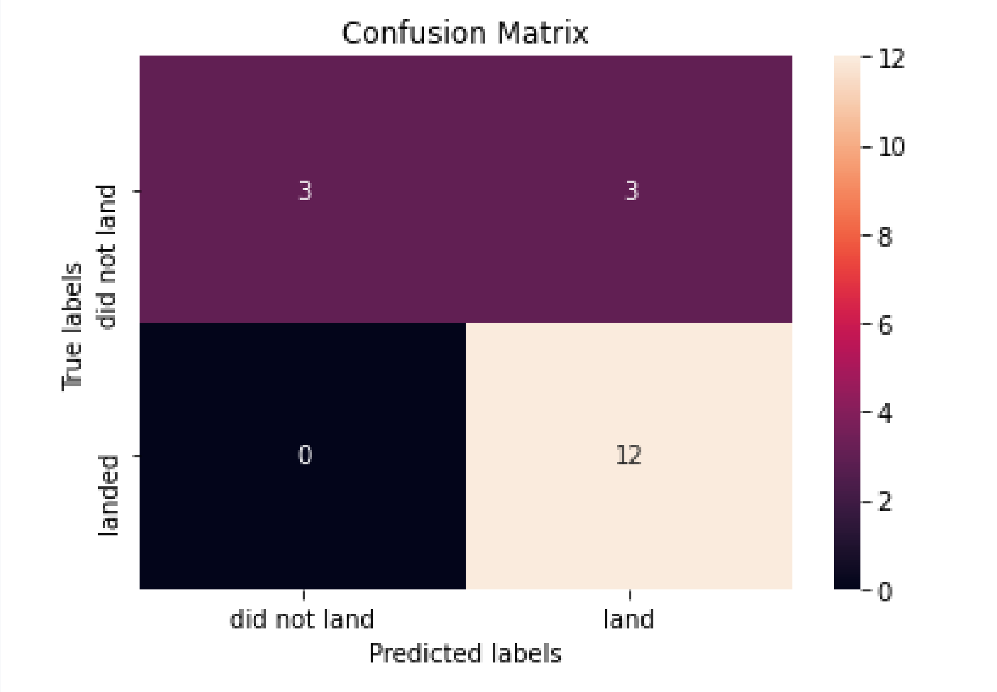
- We plotted the best score with the training set against the accuracy with the testing set, for each model

- Even if the Decision Tree model is the best with the training set, all models have a similar accuracy with the testing set.



Confusion Matrix

- Here is the confusion matrix for the Decision Tree model.
- We observe that there is no False Positive then we can assume that each landing predicted by the model will be a success.
- On the other hand, there are 3 False Negative results that might lead to waste rockets in situations where we could have safely landed them.



Conclusions

- Depending on the type of orbit we need to reach, we must have at least 2 launch sites, one on the east coast (KSC or CCAFS) and the other on the west coast (VAFB) for Polar orbit launches.
- In order to achieve a successful landing, we need to select the appropriate booster version according to the payload and the budget allowed to each launch :
  - FT for payload under 5,500Kg
  - B4 for payload between 3,000 and 5,000 kg
  - Heavy for payload over 10,000kg
  - B5 seems to have a really good reusable rate (up to 10 times) but we need to further investigate this specific version because it is the latest version used by SpaceX
- Some orbits have a really low commercial potential due to the low demand (1 launch) : ESL1, HEO, SO and GEO
- Orbits with the best commercial potential are : ISS, LEO, PO, SSO and VLEO due to their high success rate and a fair number of occurrences.
- We did not identify the features that affect GTO orbit launches success rate (further investigations required) but it success rate is still over 60% and has a fair number of occurrences too, that stills qualify it for commercial uses.
- In order to improve our decision making process, we need to collect more data about Flacon 9 Block 5 launches and to identify the estimated average reuse time for each booster version.

Appendix

All resources can be found online.

- SpaceX Falcon 9 Wikipedia webpage: https://en.wikipedia.org/wiki/Falcon_9
- All Hands-on Lab Notebooks GitHub repository: https://github.com/vahnorion/COURSEA_DS_CAPSTONE
  - Data Collection API Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/Data%20Collection%20API%20Lab.ipynb
  - Data Collection Web Scrapping Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/Web%20Scraping%20Lab.ipynb
  - Data Wrangling Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/EDA%20Lab%20(Wrangling%20Data).ipynb
  - EDA with Visualization Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/EDA%20with%20Visualization%20Lab.ipynb
  - EDA with SQL Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/EDA%20with%20SQL%20Lab.ipynb
  - Interactive Visual Analytics with Folium (Map) Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20Lab.ipynb
  - Python file created for Interactive Dashboard with Plotly Dash: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/spacex_dash_app.py
  - Dataset used for Interactive Dashboard with Plotly Dash (provided by Coursea): https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/spacex_launch_dash.csv
  - Repository of screenshots used in Interactive Dashboard with Plotly Dash : https://github.com/vahnorion/COURSEA_DS_CAPSTONE/tree/master/Dashboard%20states
  - Machine Learning Prediction Lab: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/Machine%20Learning%20Prediction%20Lab.ipynb
- This Assignment: https://github.com/vahnorion/COURSEA_DS_CAPSTONE/blob/master/ds-capstone-template-coursera.pdf

Thank you!