Winning Space Race with Data Science

Marcella Colombari

marcella.marci.colombari@gmail.com

2023-02-18

IBM Developer
SKILLS NETWORK

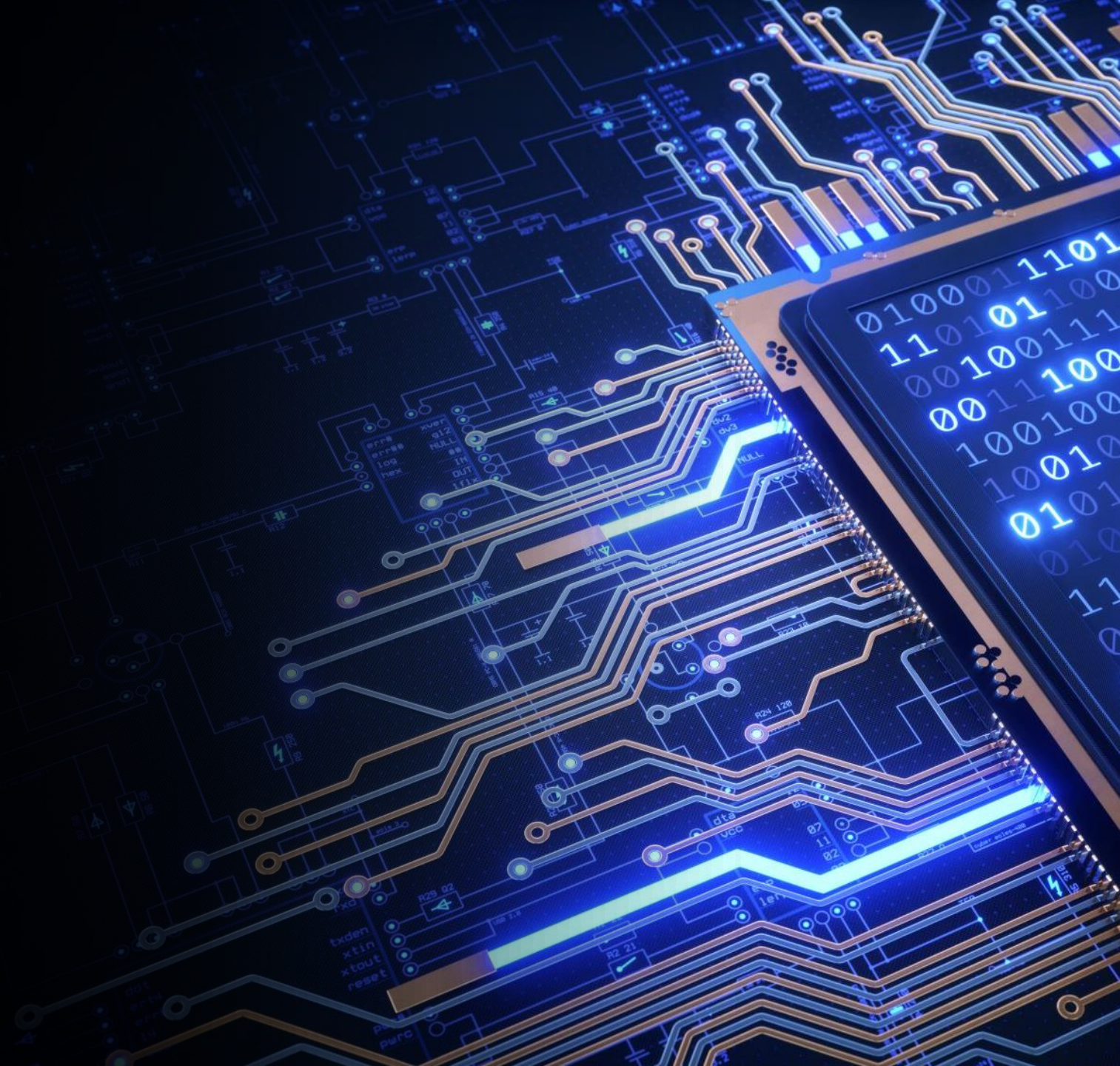
TABLE OF CONTENTS

INTRODUCTION

- This project revolves around ***SpaceX***, a rocket company that advertises launches for civilians around the Earth. The cost for a ticket to outer space is 62 million dollars, much less than the other provider's (up to 165 million dollars)
- Much of the savings is because *SpaceX* can reuse the first stage (i.e. the rocket that provides the primary propulsion from the Earth)
- In this study, we will determine if the first stage will land successfully, in order to determine the cost of a launch
- We will also find any useful correlations that can help us improve the future outcomes

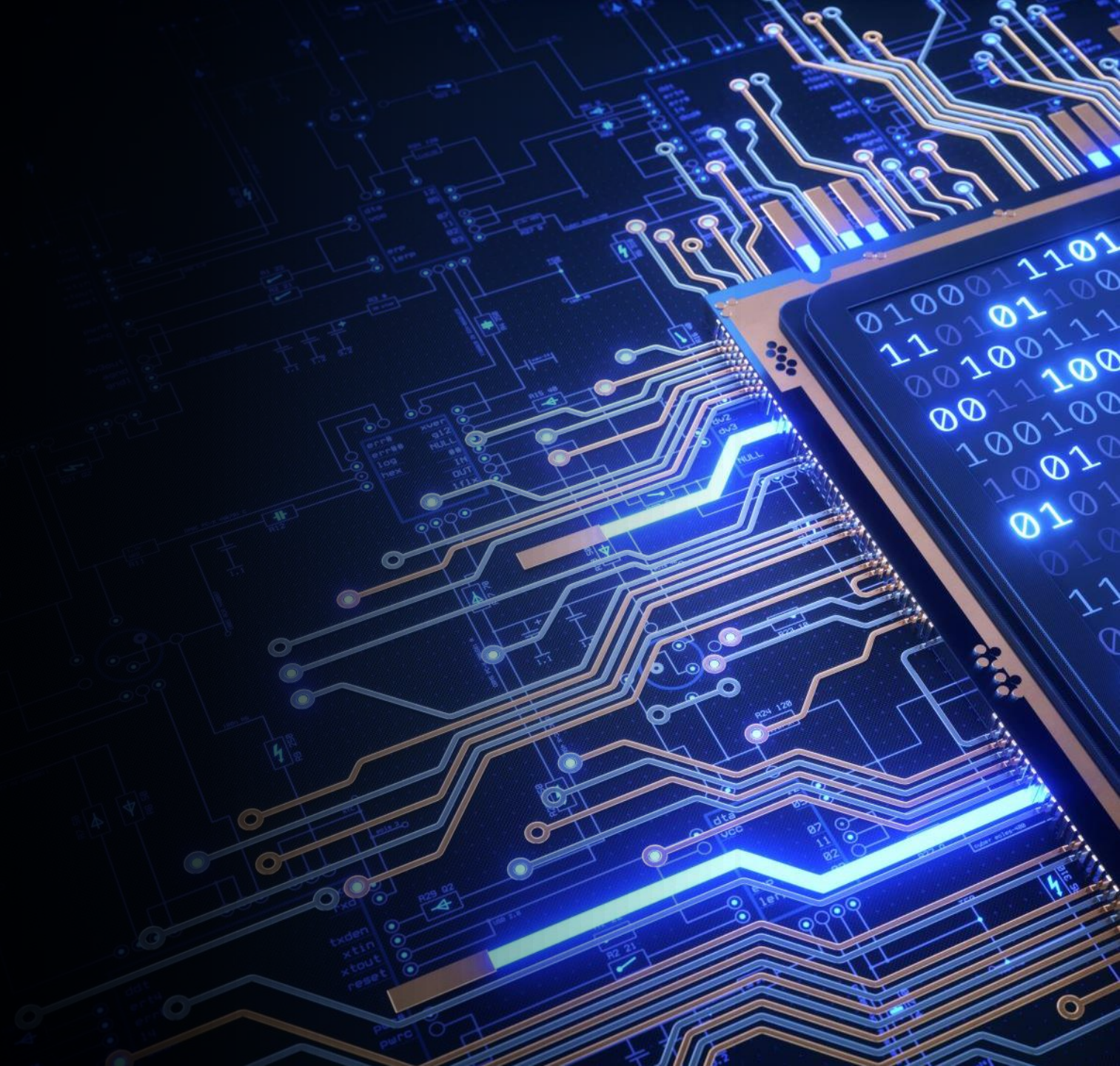
METHODOLOGY

- **Section 1**



Executive Summary

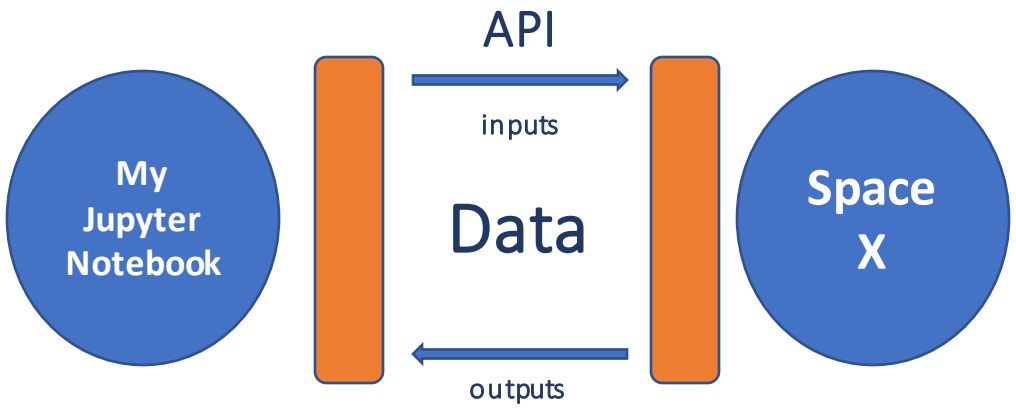
- **DATA COLLECTION** using two methods:
  - API request
  - Web Scraping
- **EDA** (Exploratory Data Analysis)
  - Data Wrangling
  - EDA with SQL
  - EDA with Data Visualization
- **INTERACTIVE VISUAL ANALYTICS** with Folium
- Building an **INTERACTIVE DASHBOARD** with Plotly Dash
- **MACHINE LEARNING PREDICTION**
- **INSIGHT** AND MORE **EDA**
- **CONCLUSIONS**



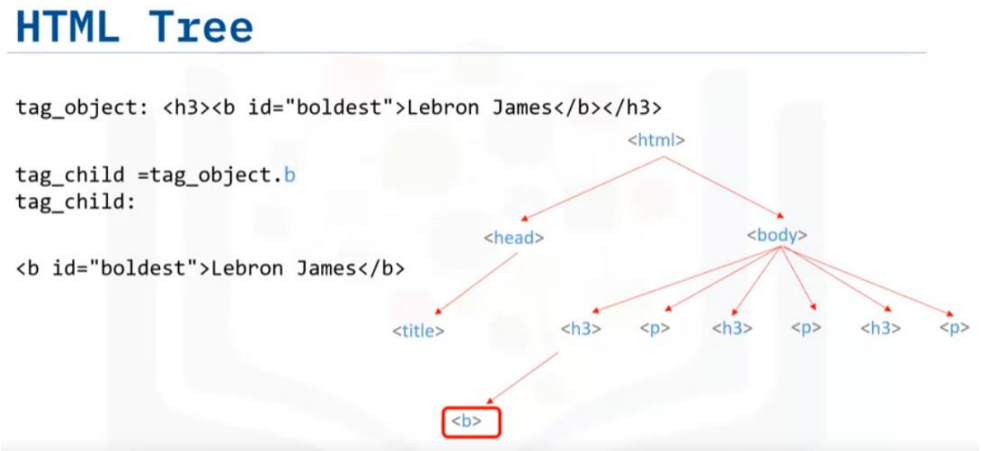
Data Collection

To retrieve the data we needed, we used two methods:

1) Data Collection via SpaceX API

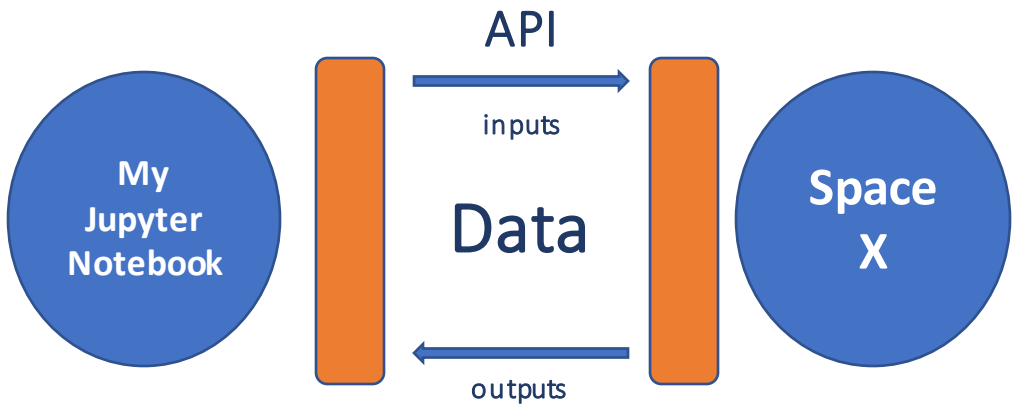


2) Data Collection via Web Scraping



Data Collection – SpaceX API

- I began the process using Requests to get the data
- Then I transformed the json data into a dataframe using Pandas
- The data seemed very messy, so I created a new list in order to build a new dataframe from those data
- I filtered the new dataframe to only include Falcon 9 launches
- Some data wrangling helped dealing with missing values



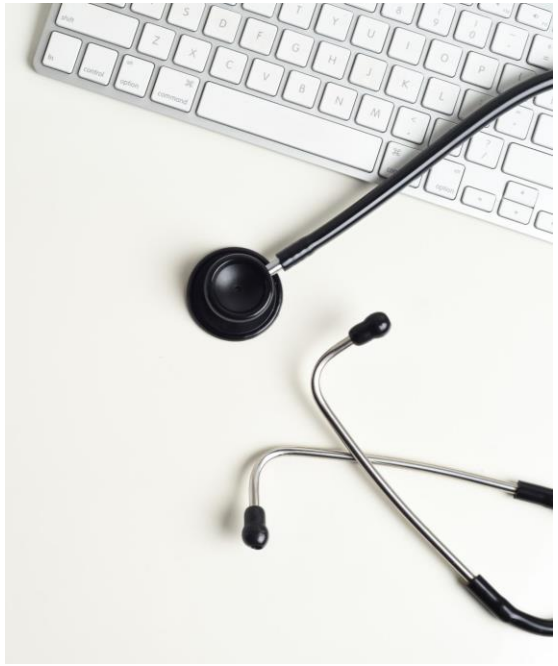
Here is a final view of the database I obtained:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitu
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561

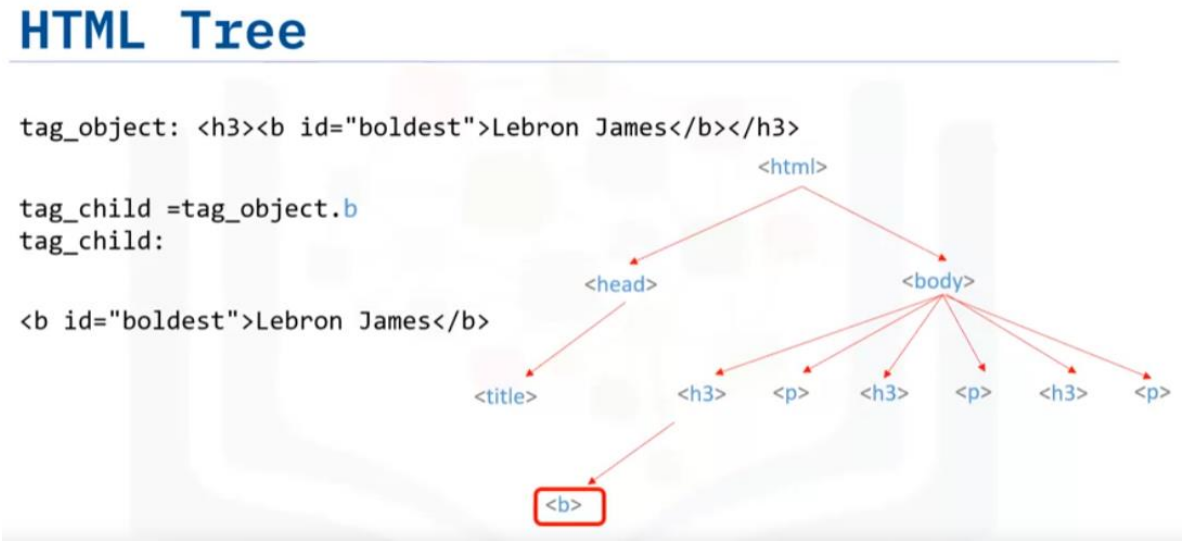
CTRL + Click HERE to visualize my Data Collection with API notebook on GitHub

Data Collection – Web Scraping

- Using Requests, I extracted a Falcon 9 launch record html table from Wikipedia
- Then I parsed the table with BeautifulSoup and converted it into a Pandas dataframe. I used custom functions in order to access the wanted data
- Finally, I exported the dataframe to a CSV file, in order to use it in the next steps



The logic used to use html in order to obtain a dataframe from Web Scraping is shown below:



CTRL + Click HERE to visualize my Data Collection with Web Scraping notebook on GitHub

Data Wrangling

I analyzed the data, found out missing data and identified both numerical and categorical columns

I then proceeded to calculate the number of launches on each site

Next, I calculated the number and occurrence of each orbit

Another step was to calculate the number and occurrence of mission outcome per orbit type

Finally, I added a categorical Class column to the dataframe and populated it with "1"s for positive outcomes (successful stage 1 landings) and "0"s for negative outcomes

This is the new dataframe with the categorical column "Class" which I will use in the next laboratories:

lumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

CTRL + Click HERE to visualize my Data Wrangling notebook on GitHub

EDA with SQL

Here are the queries I performed in this lab, in order to better understand the data:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch site 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. 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

CTRL + Click HERE to visualize my EDA with SQL notebook on GitHub

EDA with Data Visualization

In order to analyze the data and find correlations, I used three different charts.

SCATTER PLOTS were used for a variety of confrontations. In each case, the plots showed two different colors (for 1 and 0 class), making it visually easy to identify not only the correlation between variables, but the success of failure in those cases as well. Here are the parameters I used:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload Mass vs Launch Site
- Flight Number vs Orbit Type
- Payload Mass vs Orbit Type

A *BAR CHART* was used to visualize the relationship between success rate of each orbit type

A *LINE CHART* was used to visualize the launch success yearly trend

CTRL + Click HERE to visualize my EDA with Data Visualization notebook on GitHub

Build an Interactive Map with Folium

I used the Folium package to perform further visual analysis on the dataframe, this time using a world map.
I added several objects to our map, including:

- Circle objects and Labels showing the names of the sites

  Both these items helped to better comprehend what I saw on the map. The circle objects, along with the relative labels, highlighted the launch sites and made them immediately visible.
- Group of markers (of different colors) shown together on a single point, using Marker Cluster

  As I wanted to show the total of both successful (green) and unsuccessful (red) launches for each site, the marker cluster helped solve this problem. The result on the map was the total number of launches for the site. Then, zooming in clicking on the site, the various outcomes were shown in a cluster, each with the proper colors.
- Mouse position

  The mouse position was very useful in order to immediately identify the coordinates of a given point.
- Labels showing the distance between two coordinate points

  This was then used to determine the distance between two interesting sites (e.g. a launch site and the nearest railways) and numerically write the distance on the map.
- Lines that visually represented the distance between two coordinate points

  This way, the distance on the map was graphically even more intuitive

CTRL + Click HERE to visualize my Interactive Visualization with Folium notebook on GitHub

Build a Dashboard with Plotly Dash

If you can visually analyze a database, why not make it interactive with a dashboard?
Using Plotly Dash I built up our own app, with these peculiarities:

- A Dropdown menu

  This made possible to show the charts for all launch sites, or to select one of the four sites individually.
- A Callback function to render a Pie chart

  I linked this callback function to our dropdown menu, thus becoming able to show a pie chart with the percentage of successful/unsuccessful launches for each individual site, or to show all sites with the respective total of successful la unches.
- A Slider

  This was built to represent the scale from 0 to 10.000kg (with steps of 1000kg) and being able to manage the results depen ding on the Payload Mass.
- A Callback function to render a Scatter chart

  This chart showed the correlation between Payload Mass and the outcome Class. And not only that: I added distinct colors for the different Booster Versions, to make them immediately recognizable from one another. As always, our dropdown menu made possible to create a chart for all sites, or rather individual charts.
  Furthermore, the slider I previously built allowed me to restrict the results on the base of the Payload Mass.

CTRL + Click HERE to visualize my Dashboard Building code on GitHub

Predictive Analysis (Classification)

I took several steps in order to find the best classification model for our database. Here are the steps:

Preparation	Plot the confusion matrix, create a NumPy array for the column class and assign it to Y, standardize the data in X
Train/Test Split	Use the train_test_split function to split the data X and Y into training and test data, with test_size =0.2 and random_state = 2
Model: Logistic Regression	Fit the model to find the best parameters using GridSearch, then find the accuracy score and plot the confusion matrix
Model: SVM (Support Vector Machine)	Fit the model to find the best parameters using GridSearch, then find the accuracy score and plot the confusion matrix
Model: Decision Tree	Fit the model to find the best parameters using GridSearch, then find the accuracy score and plot the confusion matrix
Model: KNN (K-Nearest Neighbours)	Fit the model to find the best parameters using GridSearch, then find the accuracy score and plot the confusion matrix
Model Evaluation	Confront the accuracy of our four models and decide which better suited our case

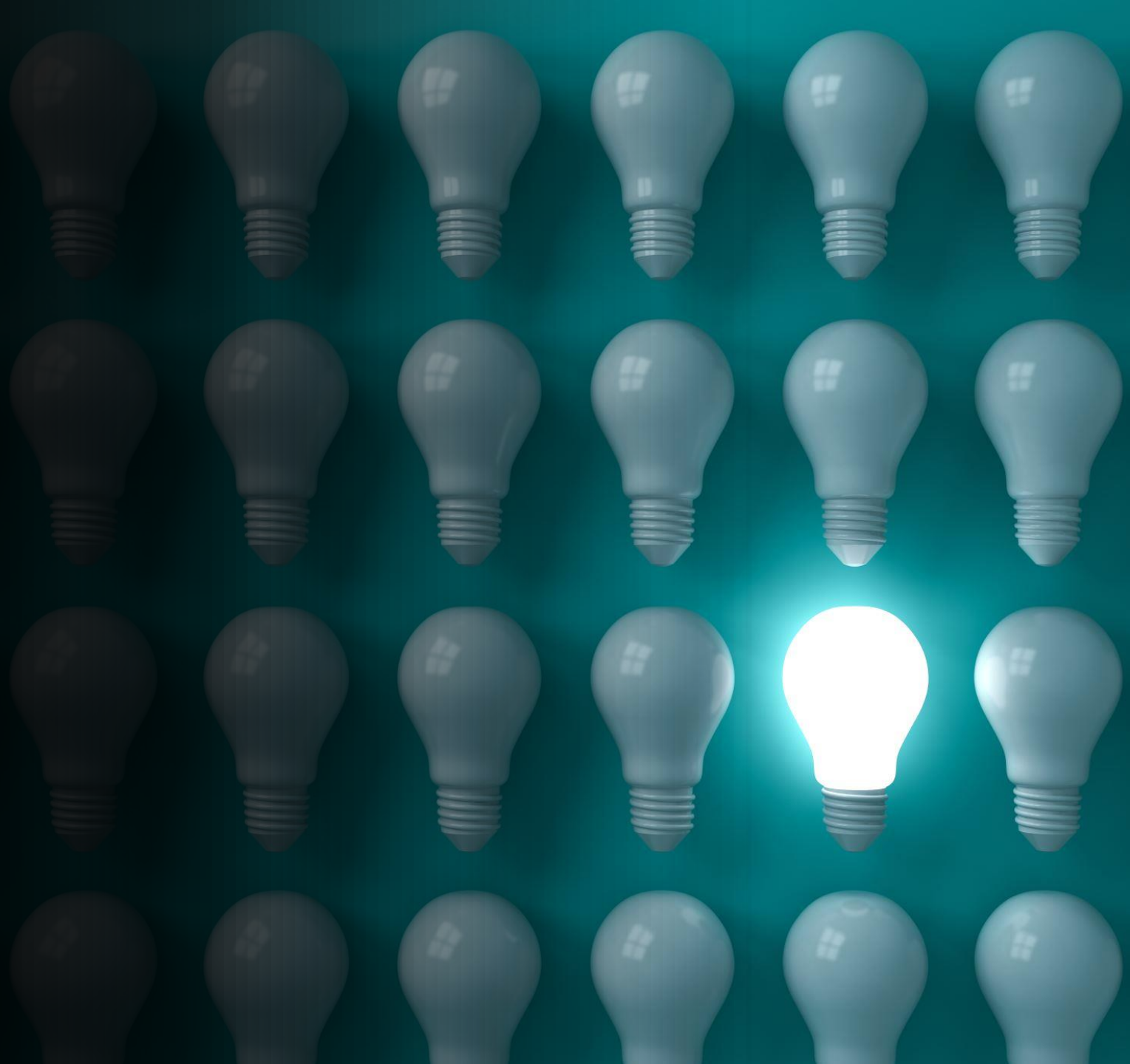
CTRL + Click HERE to visualize my Machine Learning Prediction notebook on GitHub

RESULTS TABLE

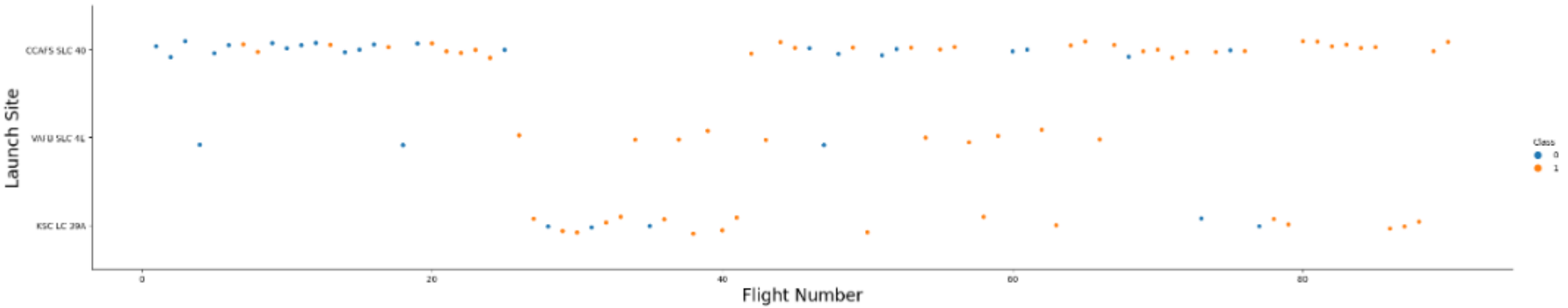
- Section 2: Exploratory Data Analysis results
- Data Visualization results
  - Section 3: Charts, SQL queries, Interactive Analytics maps
  - Section 4: Interactive Dashboard
- Section5: Predictive Analysis results

INSIGHT DRAWN FROM EDA

- **Section 2**



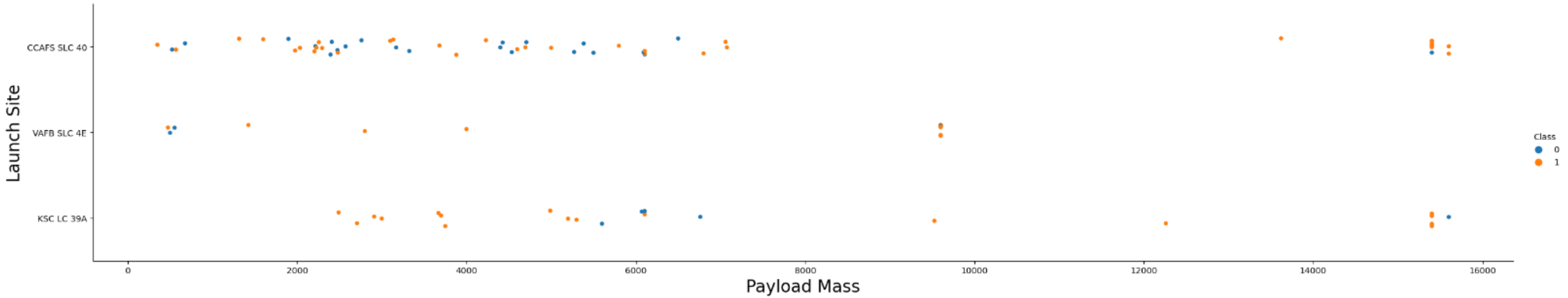
Flight Number vs. Launch Site



Here is the scatter chart plotting the number of the flight and the launch site, with the outcome classes shown in two colors.

- **CCAFS SLC-40** site shows the vast majority of the launches. Notably, the first ones took off almost exclusively from there
- **VAFB SLC-4E** it's the less used amongst the three sites, though regularly. No flies took off from there after number 66. Nonetheless, the success rate appears to be very good
- No fly took off from **KSC LC-39A** until n.27, after which this site took temporarily the place of CCAFS SLC-40, probably to see if the site was more promising. It shows a very good overall success rate
- **The success outcome** was poor at the beginning, but it kept on growing constantly over time, sign that there were tangible improvements in the procedure and/or the technology

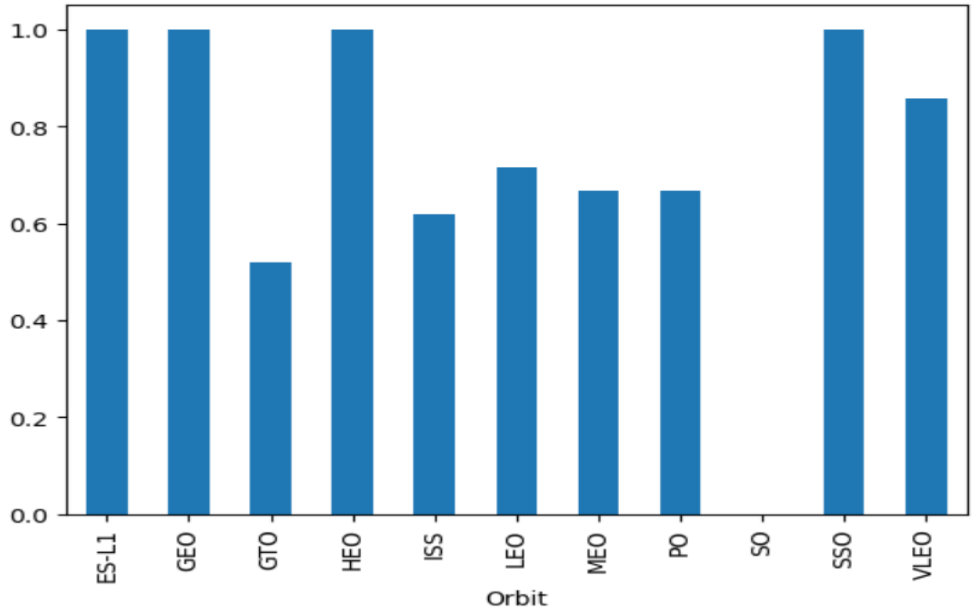
Payload Mass vs. Launch Site



Here is the scatter chart plotting the number of the flight and the payload mass, with the outcome classes shown in two colors.

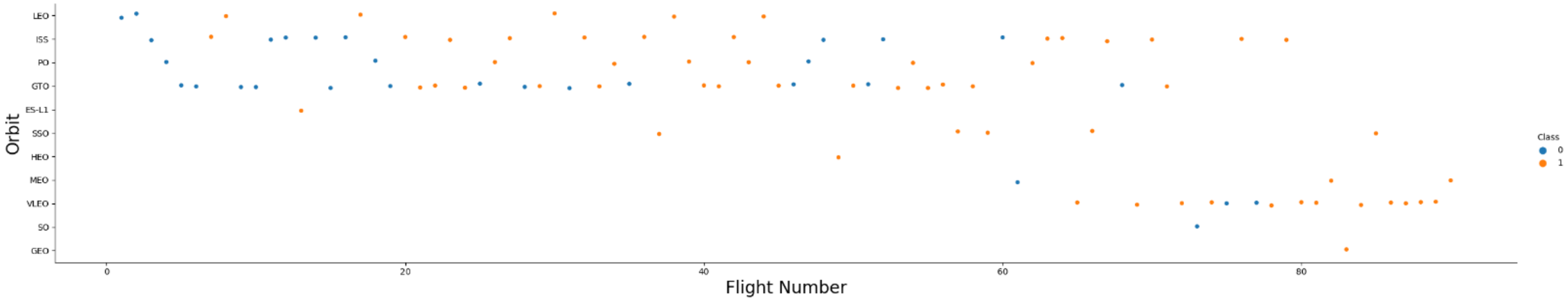
- **CCAFS SLC-40** site shows a large variety of payloads. It's not been utilized for payload masses between 7500 and 13500. Shows very good success rate for the heaviest payloads (around 15.000kg). It's been unpredictable from 500kg to 6500kg.
- **VAFB SLC-4E** shows a good success rate, except in case of extremely light payloads (around 500kg). It's been the most used site for payloads around 10.000kg. Furthermore, we can see no launches with 10k+ payloads from this site.
- **KSC LC-39A** has been utilized only from 2500kg+ payloads, which could possibly explain why the very first launches weren't made here. It showed impeccable success outcome from 2500 to 5500kg. Launches from 5500 to 7000kg showed a really bad outcome rate instead. Good success rate with the heaviest payloads (up to 15.000kg)
- **The heaviest the payloads, the more success rate** is the counter-intuitive thing that we can notice the most. The first thing that comes to mind is that the flights with the heaviest payloads could be among the last ones (where the success percentage was very high), but surely not among the first ones (where the landing of the first stage was quite unreliable)

Success Rate vs. Orbit Type bar chart



- Four orbits show a 100% success outcome (**ES-L1, GEO, HEO and SSO**)
- **VLEO** orbit is also very successful with over 80% positive outcomes
- Only **SO** orbit has not been traveled at all, though this is not very clear since SSO and SO were first introduced as synonyms.
- The less promising is **GTO** orbit with approximately 50% success
- The remaining orbits (**ISS, LEO, MEO and PO**) show a success percentage between 60% and 75%
- For the moment, there seem to be **no direct correlation** between distance from the Earth and success rate. For instance, ES-L1 is the farthest and completely successful, while GTO (though being very far from Earth) is the less successful.
- Of course we know the average of success but this alone **doesn't imply any correlation**. We don't yet know the number of flights per orbit, or the flight number vs orbit, so we must further investigate before coming to any conclusions

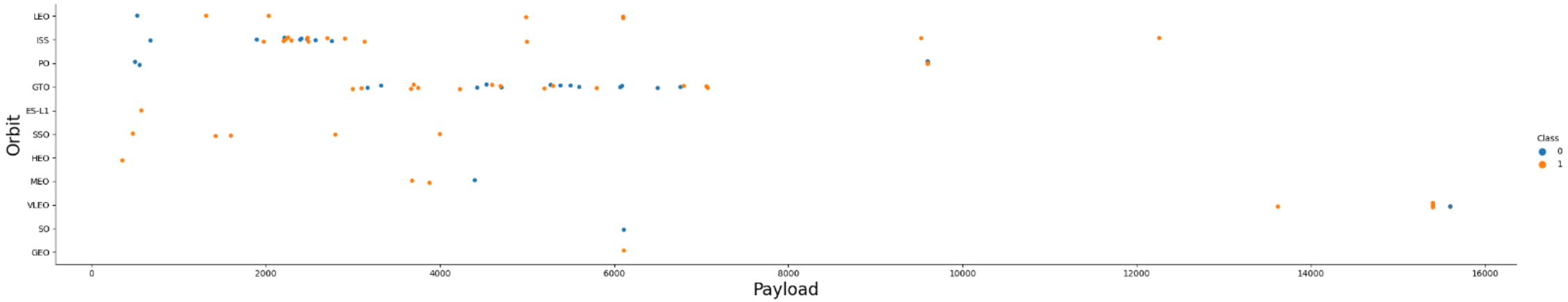
Flight Number vs. Orbit Type



Here is the scatter chart plotting the number of the flight and the orbit type, with the outcome classes shown in two colors.

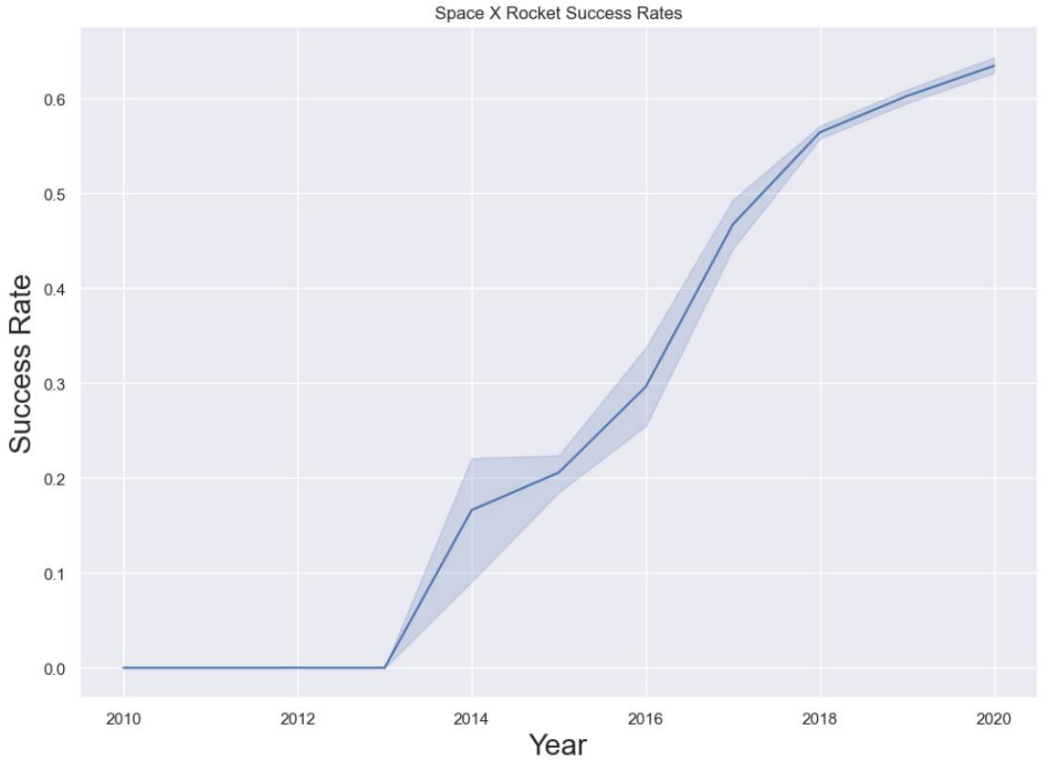
- **LEO (Low Earth Orbit)** has often been chosen the first flights, showing two unsuccessful landings at the beginning but then proving itself fully reliable. It's worth noticing though that after flight n.45 there have been no more flights on LEO orbit
- **ISS** orbit involves a modular space station, and has been chosen from the beginning and very often. After the first launches, the outcome was only positive for a while, but then there have been three negatives in a row and then only positives. The consecutive nature of these failures could be caused by similar conditions (payload? Booster category?)
- Only one flight on both **HEO** and **ES-L1** was made, thus showing that their 100% positive rate was statistically irrelevant
- A good number of flights has been made on **GTO** orbit (a medium stationary orbit, quite far away from the Earth). At first the outcomes were only negative, but the landing success improved over time
- **SSO** orbit has not been as popular as others, but notably showed only positive outcomes, even if only for 5 flights
- **VLEO (Very Low Earth Orbit)** has been traveled only after 65 flights, but has become the most popular as of late

Payload vs. Orbit Type



- **LEO (Low Earth Orbit)** shows reliability with almost all payload masses, except for the very light ones (which we now know, where the absolute first flights for this orbit)
- **ISS** orbit flights show a rate of success apparently unrelated to payload
- **GTO** orbit flights too show no correlation between payload and landing success (though as we just witnessed, the landing success steadily improved over time)
- **SSO** orbit flights only carried light payloads (up to 4.000kg) but were all successful
- **VLEO (Very Low Earth Orbit)** flights counterintuitively used only the heaviest payloads, showing only a single negative outcome

Launch Success Yearly Trend



- The chart shows irrefutably that **the successful outcomes steadily increased over time**, from 2013 to 2020
- The year with the the highest variability has been **2014**
- During the years, not only the success percentage increased greatly, but at the same time the variability decreased

All Launch Site Names

The following query retrieved the names of the unique launch sites:

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEX;
```

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- The **DISTINCT** function allowed us to retrieve only unique results from the column **LAUNCH_SITE** from our dataframe

This is a screenshot of the result:

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

Launch Site Names that Begin with 'CCA'

The following query retrieved the names of 5 launch sites that begin with 'CCA':

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

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- In the WHERE clause I used the function **LIKE 'CCA%'**, meaning that we are searching for results in the column LAUNCH_SITE that begin with 'CCA' and the the rest of the string doesn't matter
- We used **LIMIT 5** to restrict the results to just 5 rows

This is a screenshot of the result:

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

Total Payload Mass

The following query calculated the total payload carried by boosters from NASA:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- I used the **SUM** function with **PAYLOAD_MASS_KG_** as an argument, to calculate the sum of all the payloads
- In the WHERE clause I then restricted the results to only boosters from **customer 'NASA (CRS)'**, using the **operator =**

This is a screenshot of the result (1 result, total 45596kg):

1
45596

Average Payload Mass by F9 v1.1

The following query calculated the average payload mass carried by boosters version F9:

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

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- I used the **AVG** function with PAYLOAD_MASS_KG_ as an argument, to calculate the average of the payload mass
- In the WHERE clause I used the function **LIKE 'F9 v1.1%'**, meaning that we are searching for results in the column BOOSTER_VERSION that begin with 'F9 v1.1' and the rest of the string doesn't matter

This is a screenshot of the result (1 result, total 2534kg):

1
2534

First Successful Ground Landing Date

The following query found the date of the first successful landing outcome on ground pad:

```
%sql SELECT MIN(DATE) FROM SPACEX WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- I used the **MIN** function with **DATE** as an argument, to find the first day related to our WHERE clause
- In the WHERE clause I then restricted the results using **LANDING__OUTCOME = 'Success (ground pad)'** which restricted the research to only the successful outcomes on a ground pad. The result was the first available date with those parameters

This is a screenshot of the result (1 result, date 2015/15/22):

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The following query listed the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:

```
%%sql
SELECT DISTINCT(BOOSTER_VERSION), LANDING__OUTCOME, PAYLOAD_MASS__KG_ FROM SPACEX
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

- I used **%%sql**, a command from Python Magic for SQL to write exclusively in SQL language, inside a Python cell
- The **DISTINCT** function allowed us to retrieve only unique results from the column **BOOSTER_VERSION** of our dataframe
- In the WHERE clause I then restricted the results linking two clauses with the operators **=**, **AND** and **BETWEEN.**
  "LANDING__OUTCOME **=** 'Success (drone ship)' **AND** PAYLOAD_MASS_KG_ **BETWEEN** 4000 **AND** 6000" means that both conditions must be true in order to be included in the query results

This is a screenshot of the result:

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600

Total Number of Successful and Failure Mission Outcomes

The following query separately calculated the total number of successful and failed mission outcomes:

```
%sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEX GROUP BY MISSION_OUTCOME
```

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- I used the **COUNT** function with ***** as an argument, to find all the rows from the **MISSION_OUTCOME** column
- I then used the **GROUP BY** function, in order to group all the similar results from the **MISSION_OUTCOME** column

This is a screenshot of the result (2 results between Success and Failure, and the relative total):

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

The following query listed the names of the boosters which have carried the maximum payload mass:

```
%sql SELECT DISTINCT(BOOSTER_VERSION), PAYLOAD_MASS__KG_ FROM SPACEX WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)
```

This is a screenshot of the result:

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

- I used **%sql**, a command from Python Magic for SQL that allows us to write a line of code in SQL language, in a Python cell
- The **DISTINCT** function allowed us to retrieve only unique results from the column **BOOSTER_VERSION** of our dataframe
- In the WHERE clause I then restricted the results with the **MAX** function, writing "PAYLOAD_MASS__KG_ = (SELECT **MAX**(PAYLOAD_MASS__KG_) FROM SPACEX)" to restrict the results to only the maximum values of payload.
- I couldn't have used the MAX function directly after the initial SELECT, because that would have generated an error.

  The structure used here is called **subquerie**

2015 Launch Records

The following query listed the failed landing_outcomes in drone ship, their booster version, and launch site names for year 2015:

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

- I used **%%sql**, a command from Python Magic for SQL to write exclusively in SQL language, inside a Python cell
- In the WHERE clause I then restricted the results linking two clauses with the operators **=** , **AND.**
  "LANDING__OUTCOME **=** Failure (drone ship)' **AND** YEAR (DATE) = 2015**"** means that both conditions must be true in order to be included in the query results
- The way I used the **DATE** function this time, writing **YEAR(DATE)**, returned all the results containing the wanted year (all months and days indiscriminately)

This is a screenshot of the result:

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

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

The following query ranked 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:

```
%%sql
SELECT COUNT(*),LANDING__OUTCOME FROM SPACEX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY count(LANDING__OUTCOME) DESC
```

- I used **%%sql**, a command from Python Magic for SQL to write exclusively in SQL language, inside a Python cell
- I used the **COUNT** function with * as an argument, to find all the rows from the **LANDING__OUTCOME** column
- In the WHERE clause I then restricted the results temporally using **BETWEEN**
- I then used the **GROUP BY** function, in order to group all the similar results from the LANDING__OUTCOME column
- Eventually I used **ORDER BY** COUNT(LANDING_OUTCOME) using COUNT to find the total per each landing outcome and order the dataframe accordingly
- I used **DESC** to make sure that the dataframe was in descending order

This is a screenshot of the result:

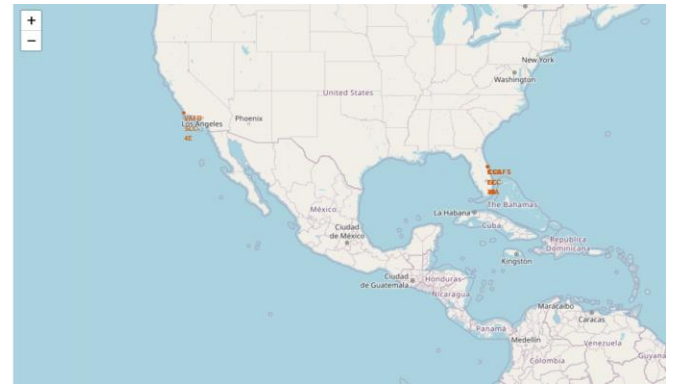
1	landing__outcome
10	No attempt
5	Failure (drone ship)
5	Success (drone ship)
3	Controlled (ocean)
3	Success (ground pad)
2	Failure (parachute)
2	Uncontrolled (ocean)
1	Precluded (drone ship)

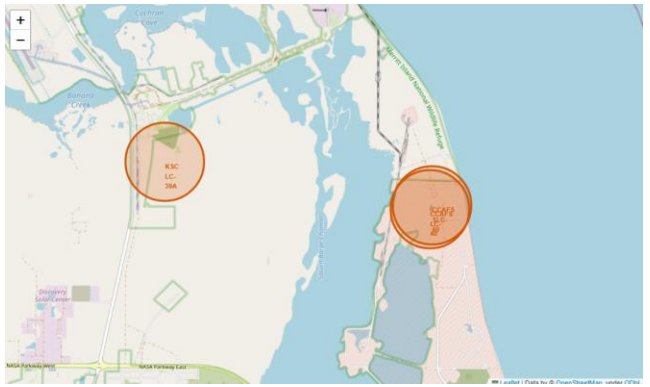
LAUNCH SITES PROXIMITIES ANALYSIS

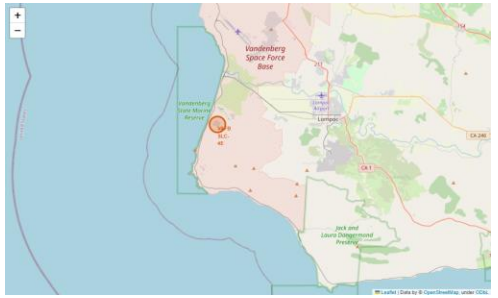
- **Section 3**

Localization of the Launch Sites

- The first chart shows that all the sites are built inside US territory.
- Another key factor is the proximity of all sites to the coast. This can be paramount for various reasons, one of which must be logistics. Secondarily, this facilitates the utilize of drone ships not far from the base, allowing stage 1 rockets to land in open waters, far from civilians and structures of any type
- The most utilized three sites (KSC LC-39A, CCAFS LC-40 and CCAFS SLC-40) are all in Florida, on the East Coast.
- The less utilized VAFB SLC-4E base is built in California





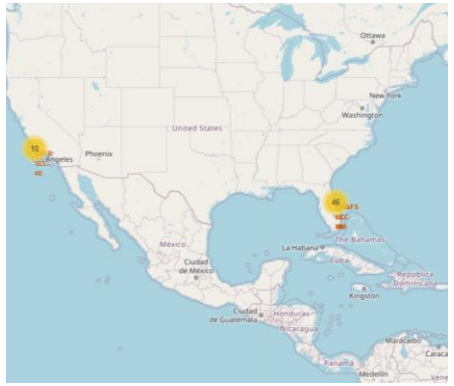


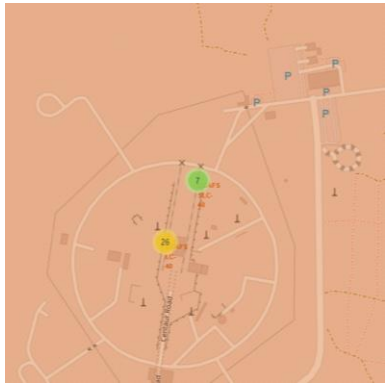
Visualization of the Launch Outcomes on the map

- The first chart shows the distribution of the launches between West and East Coast, with a large majority (46 vs 10) in the eastern sites

- The second chart shows the distribution between the single sites. We can see that the single base with the most launches (26) is CCAFS LC-40

- The third chart visually shows the single outcomes for the launches (green for success, red for failure). It becomes immediately clear even at first glance, that the base with the highest success rate is KSC LC-39A



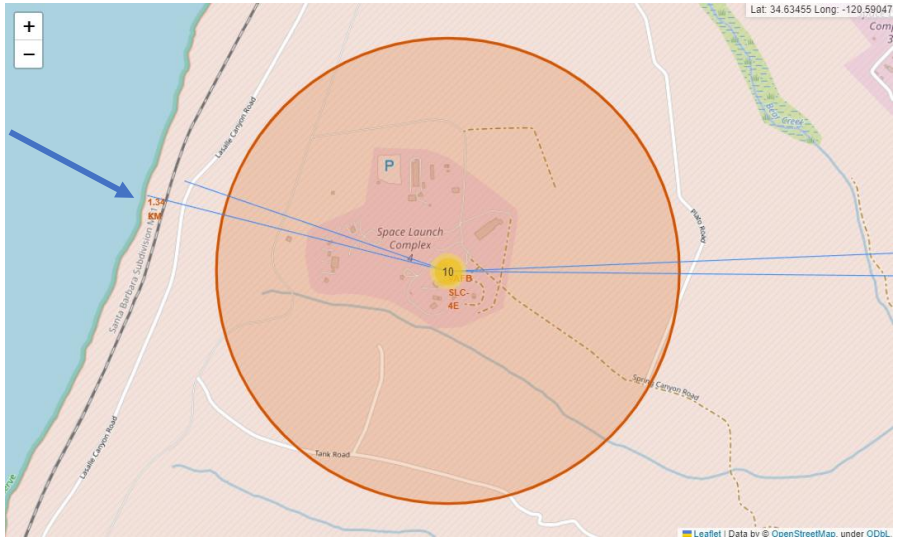


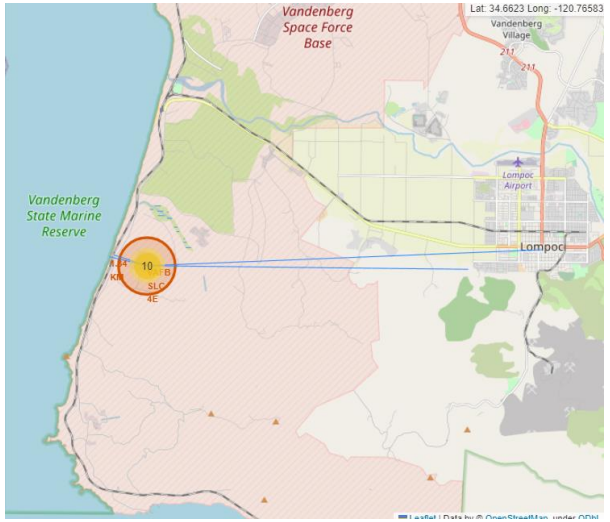
Graphic Representation of distances between Launch Sites and other sites of interest

- These charts depict the KSC LC-39A base. In the first chart we can immediately see that the base is very close to both the nearest railway and the coast (just 1,34km as shown by the blue arrow)

- Zooming out, in the second chart it becomes evident that the base is also very close to the nearest city (Lompoc) and the nearest highway (CA 1)

- The reasons for this strategical placement can be numerous: logistic of the spare parts, accessibility for the passengers of the flights, easier ocean landings and many more





BUILD A DASHBOARD WITH PLOTLY DASH

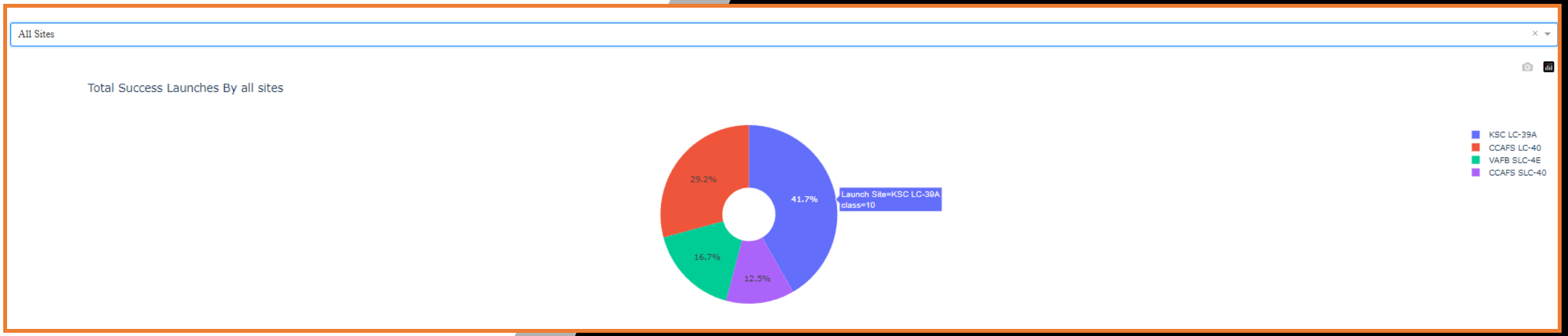
- **Section 4**

Overview on the Success Rate of all Launch Sites

- This pie chart shows the launch success count for all sites
- It is immediately clear that **KSC LC-39A** is the most successful launch site
- On the other hand, **CCAFS SLC-40** is the site with the lowest success count

In this dashboard you can see:

- The Dropdown Menu on the top (All Sites selected)
- The pie chart in the middle, shown the launch sites with different colors
- Also, the chart is mouse-responsive: hovering on the various parts lets us visualize additional data, such as the Launch Site's name and the total positive class (KSC LC-39A and 10 in this case)
- Finally, on the right you can find the legend with the various Launch Sites

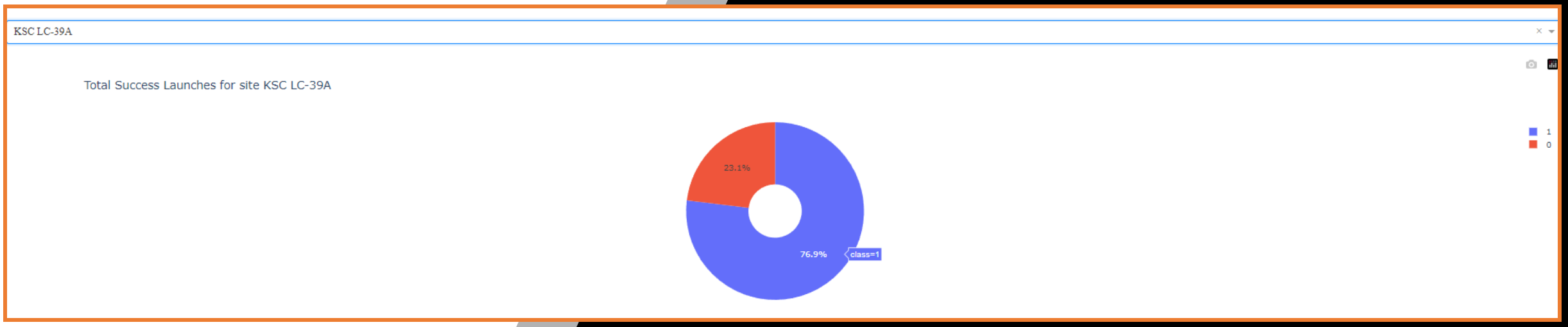


Most Successful Launch Site Chart

- This pie chart depicts the most successful among the launch sites: Florida based **KSC LC-39A**
- The success rate amounts to a reassuring **76.9%**

In this dashboard you can see:

- The Dropdown Menu on the top (**KSC LC-39A site** selected)
- The pie chart in the middle, shown the class outcomes with different colors
- Also, the chart is mouse-responsive: hovering on the various parts lets us visualize the relative class
- Finally, on the right you can find the legend with the two separate outcomes, 1 for positive and 0 for negative



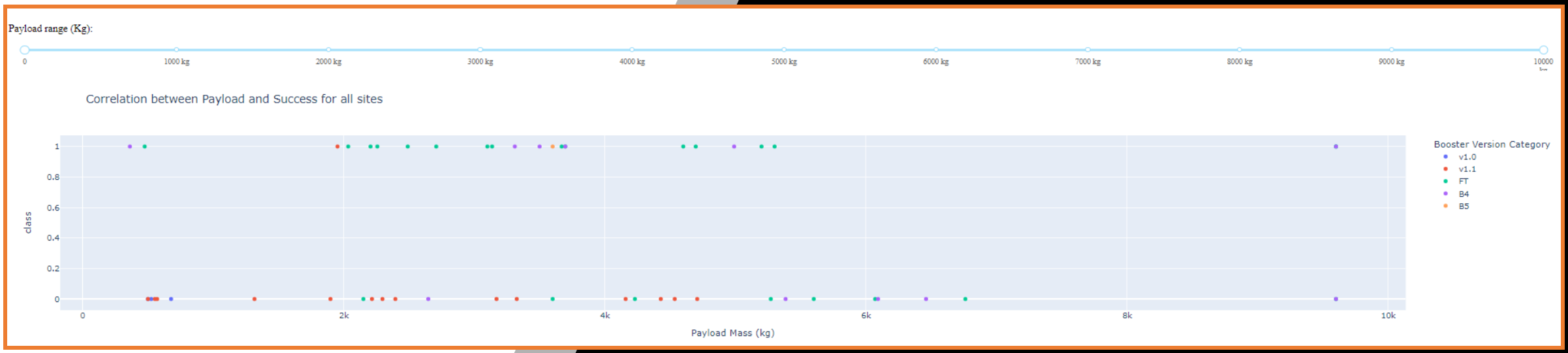
Interactive Payload range vs Success Class Dashboard

(including Booster Types – part 1)

- For the lighter payloads, a variety of boosters was used
- **V1 and V1.1** versions displayed a very high failure rate. No wonder, considering that they were the first versions
- **FT Boosters** performed very well with payloads under 5500kg
- Notably though, the payloads in this database were only under 10.000kg. This makes any insight that comes from this dashboard incomplete, since we know that there have been several launches with payload above 10.000kg, but I will investigate this aspect thoroughly in section 6.

In this dashboard you can see:

- The Payload Range Slider on the top (min 0kg selected)
- The scatter chart with X = Payload Mass and Y = Class
- Also, the chart is mouse-responsive: hovering on the various dots lets us visualize various info (Booster Category, Payload Mass, Class)
- Finally, on the right you can find the legend with the various booster types



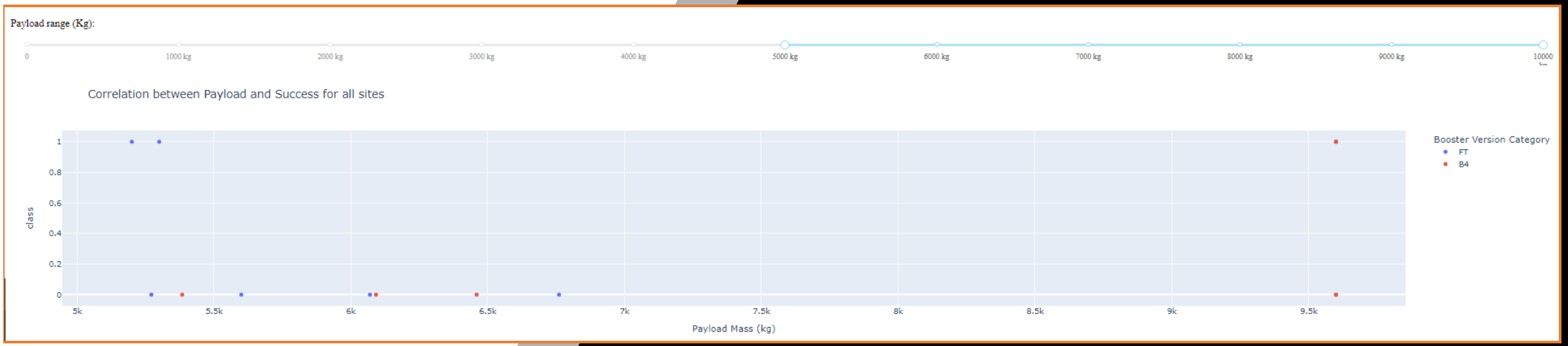
Interactive Payload range vs Success Class Dashboard

(including Booster Types – part 2)

- For payloads above 5000kg **only FT and B4** boosters were used
- For such **heavy payloads**, however, the general success rate is close to 0
- Notably, **FT boosters too** produce negative outcomes with 5000+ payloads, when their usual success rate is high.

In this dashboard you can see:

- The Payload Range Slider on the top (min 5000kg selected)
- The scatter chart with X = Payload Mass and Y = Class
- Also, the chart is mouse-responsive: hovering on the various dots lets us visualize various info (Booster Category, Payload Mass, Class)
- Finally, on the right you can find the legend with the various booster types



Further Analysis of the Launch Site with the highest success rates

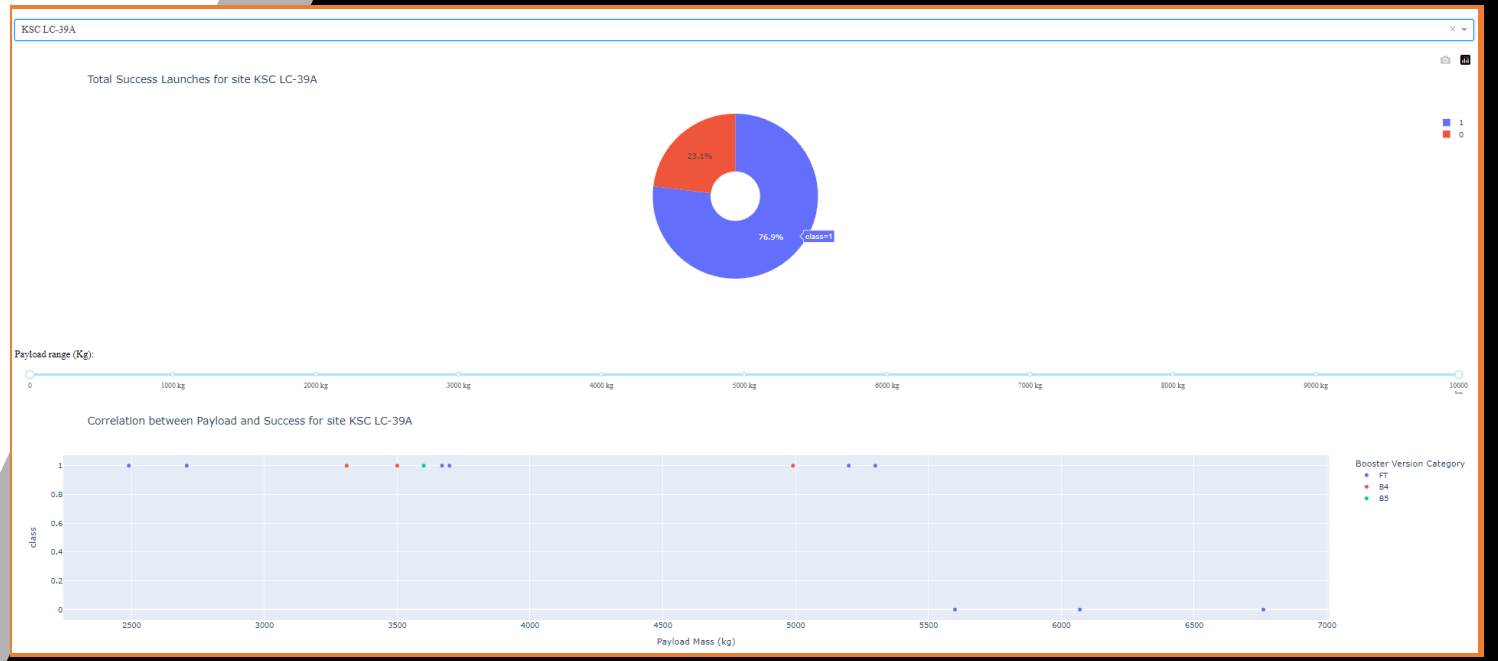
- We can see that the most successful launch site (KSC LC-39A) used almost exclusively **FT and B4 boosters**, the best versions

- Notably, the only three failures display a **mass heavier than 5000 and less than 10000kg**, confirming what we saw before

- The success of the launches from the KSC LC-39A station seems to depend from Booster Type (**more advanced technology**) and Payload (less than 5000kg)

In this dashboard you can see:

- The Dropdown Menu on the top (**KSC LC-39A** site selected)
- The pie chart in the middle, shown the class outcomes with different colors
- Also, the chart is mouse-responsive: hovering on the various parts lets us visualize the relative class
- On the right you can find the legend with the two separate outcomes, 1 for positive and 0 for negative
- In the middle we find the Payload Mass Slider (min 0kg)
- The scatter plot with X = Payload and Y = Class

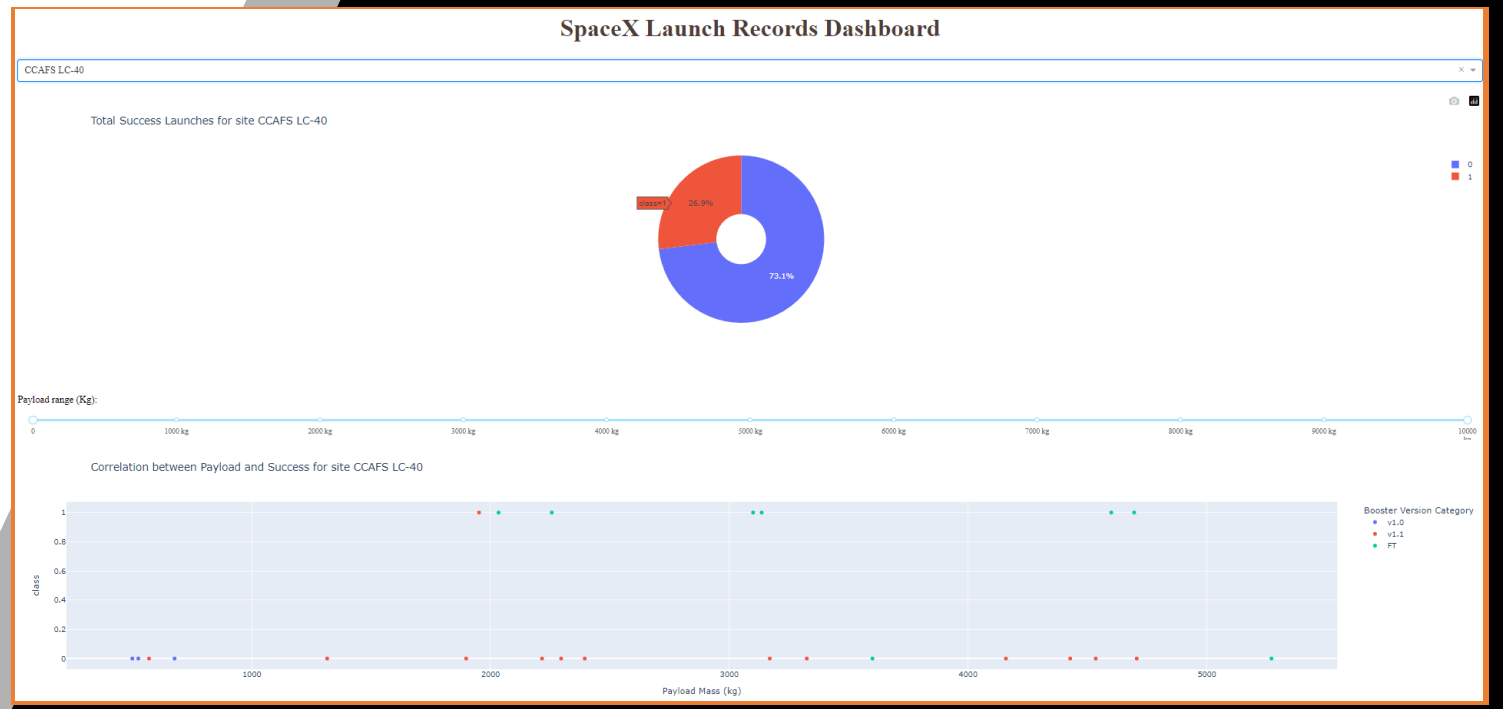


Further Analysis of the Launch Site with the lowest success rates

- The launch site with the lowest success rate (CCAFS LC-40) utilized a great majority of **V1 and V1.1 boosters** (the least successful ones) and this is clearly visible in the chart, heavily lowering the overall success rate for the launch site
- However, at CCAFS LC-40 the **FT boosters** confirmed their positive performances with light to medium payloads with a 75% success rate (6 out of 8)
- The low success rate for CCAFS LC-40 seems due to **less technologically advanced boosters**

In this dashboard you can see:

- The Dropdown Menu on the top (**CCAFS LC-40** site selected)
- The pie chart in the middle, shown the class outcomes with different colors
- Also, the chart is mouse-responsive: hovering on the various parts lets us visualize the relative class
- On the right you can find the legend with the two separate outcomes, 1 for positive and 0 for negative
- In the middle we find the Payload Mass Slider (min 0kg)
- The scatter plot with X = Payload and Y = Class



PREDICTIVE ANALYSIS (CLASSIFICATION)

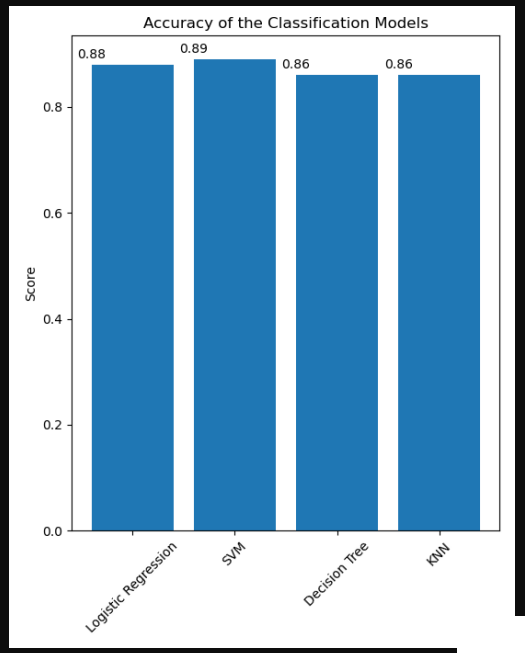
- **Section 5**

Classification Accuracy

- We tested four models on our dataframe:

1. **Logistic Regression**
2. **SVM** (Support Vector Machine)
3. **Decision Tree**
4. **KNN** (K-Nearest Neighbours)

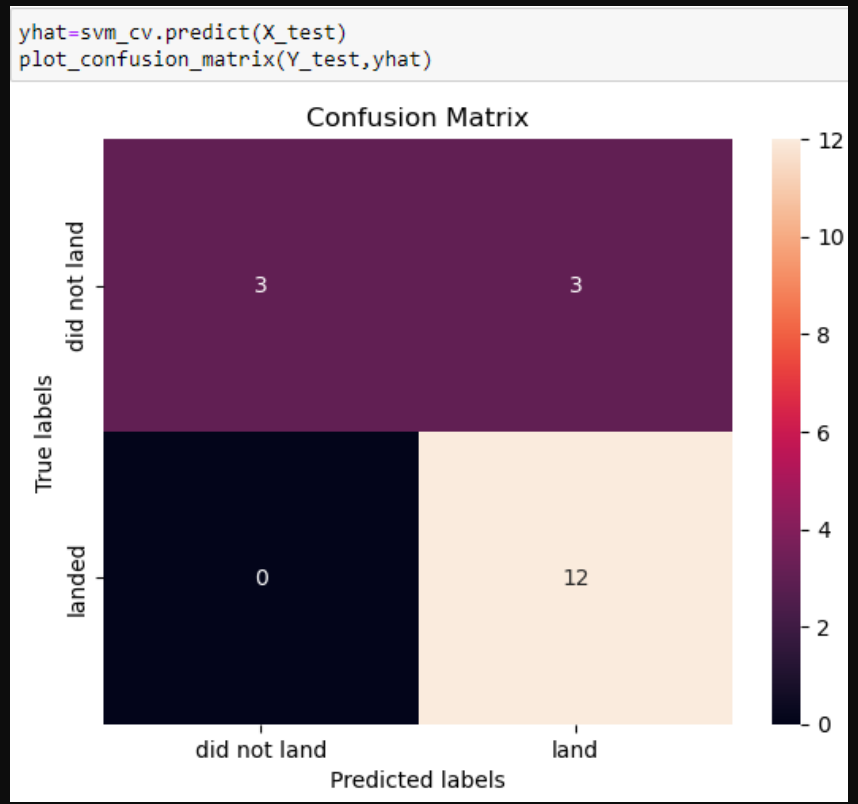
- The table on the right displays the Accuracy Score of all four models
- As we can see both in the table and in the bar chart, the most accurate model is **SVM**, though it was a very close call



	Score
Logistic Regression	0.88
SVM	0.89
Decision Tree	0.86
KNN	0.86

Confusion Matrix

- This is the Confusion Matrix we obtained from **Support Vector Machine Model**
- We can immediately notice a reassuring 100% of **true positives**
- The weak link of the chain are the **false positives** with their 50% percentage



INSIGHT AND MORE EDA

- **Section 6**

INSIGHT

1. **CCAFS LC-40 base** shows the lowest success rate, but it was also the first site to be utilized. The first booster versions V1 and V1.1 had a very high failure rate. Technology of the boosters evolved over time and the success rate raised steadily to a very promising level (only 1 negative in the last 10 flights, 3 in the last 20). There is no issue with the CCAFS LC-40 launch site itself.
2. **KSC LC-39A base** has the highest success rate, but as we have seen it utilized almost only the more technologically advanced FT and B4 boosters. Consequently, KSC LC-39A base seems to have no particular advantage over the other sites.
3. There has been a notable **problem with payloads between 5000 and 7000kg**. Even the FT boosters (usually very reliable) proved quite unsuccessful with those medium payloads
4. This could explain why the **higher orbits** (e.g. GTO) showed such an erratic success rate
5. We can also see, though, that the **success rate steadily raised during the years**. Is this anyhow related to Orbit or Payload, or just Booster Versions?
6. **Lower orbits** seem to have increased success rate and became the preferred choice over time
7. Since the first flights were made with the **unreliable V1 and V1.1 boosters**, I would like to consider them outliers and cut them out from my conclusions, since they will not be used anymore. But how can I be sure?

I need to make more Exploratory Data Analysis and confront the Booster Versions with the Flight Number

Further research of possible Correlations part 1

Unsatisfied with the insights I gathered, I needed one more answer: how many flights were taken using unreliable booster versions V1 and V1.1? Can I totally exclude those results from my future analysis? I then re-opened my EDA with Data Visualization notebook and tried to confront the Flight Number with the Booster Category. It was not possible though: the only version in the SPACEX database was a generic "Falcon9".

But I recall the dashboard I built used a different database, where the exact booster versions were available. I imported the spacex_launch_dash.csv and read it into a Pandas dataframe. I will refer to it as DASH.

I confronted the two databases and obtained the numbers in the screenshot on the right:

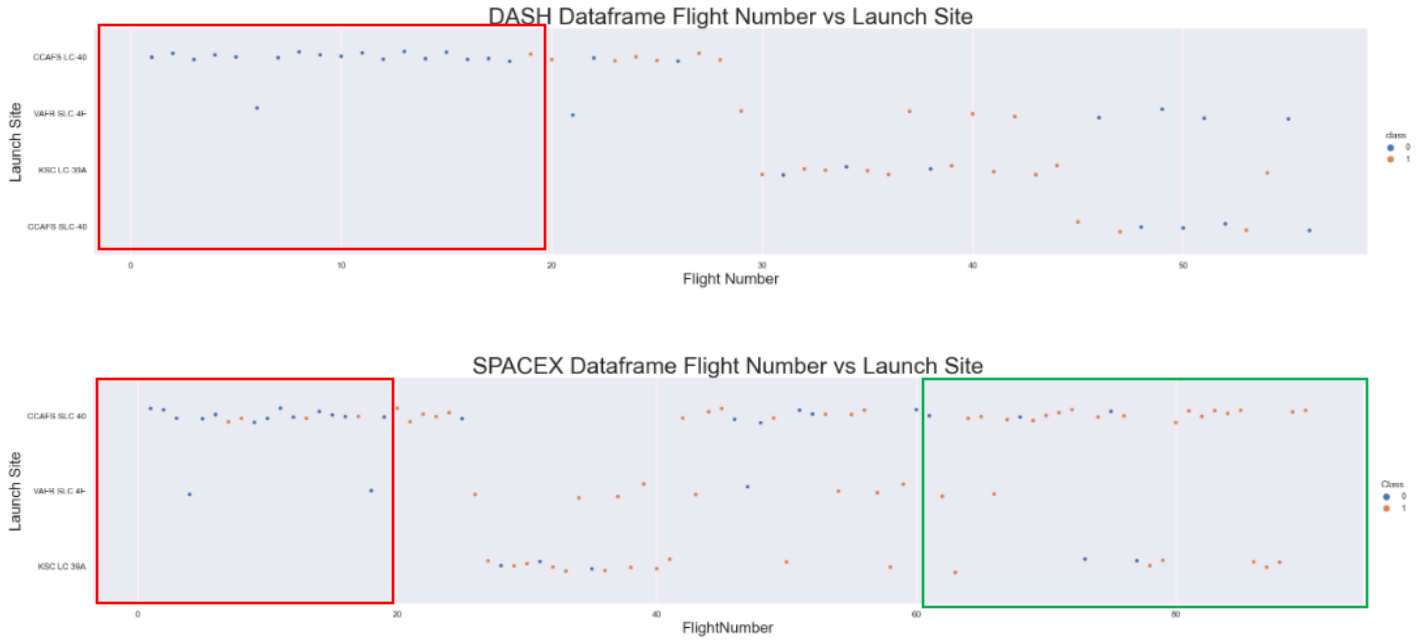
```
SPACEX Dataframes dimensions: (90, 18)
SPACEX_LAUNCH_DASH Dataframes dimensions: (56, 7)
```

As we can see, a lot of launches are missing in the DASH database (56 against 90). Can it still be used? DASH doesn't contain any date or launch id, and SPACEX doesn't have the boosters versions, so we can't use a SQL JOIN to relate the data. We can however confront them visually and make some considerations. Let's visualize a similar chart for both, using the attributes Flight Number and Launch Site that both have:

Confronting the two dataframes graphically, we notice two things:

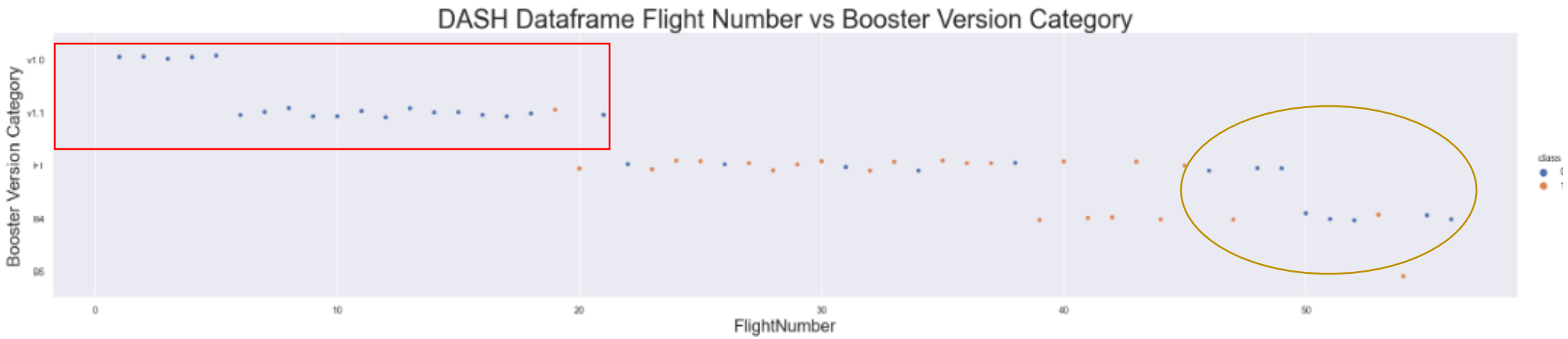
1. DASH dataframe misses the part from the 60th launch (green rectangle) which, as we have seen before, also represents the flights with payloads over 10000kg
2. The first 20 launches, though not identical, are really similar in both databases (red rectangle) and depicts almost all failed outcomes in both cases.

Since this two characteristics reassure me on the similarity of the two database (though not identical) I would like to conduct more researches.



Further research of possible Correlations part 2

I needed one more answer in particular: how many flights were taken using unreliable booster versions V1 and V1.1? Can I totally exclude those results from my analysis for the future launches? Let's use the DASH dataframe and build a scatter plot:



As I thought, the unreliable V1 and V1.1 boosters version were the **primary cause of failure** (as shown in the red rectangle).

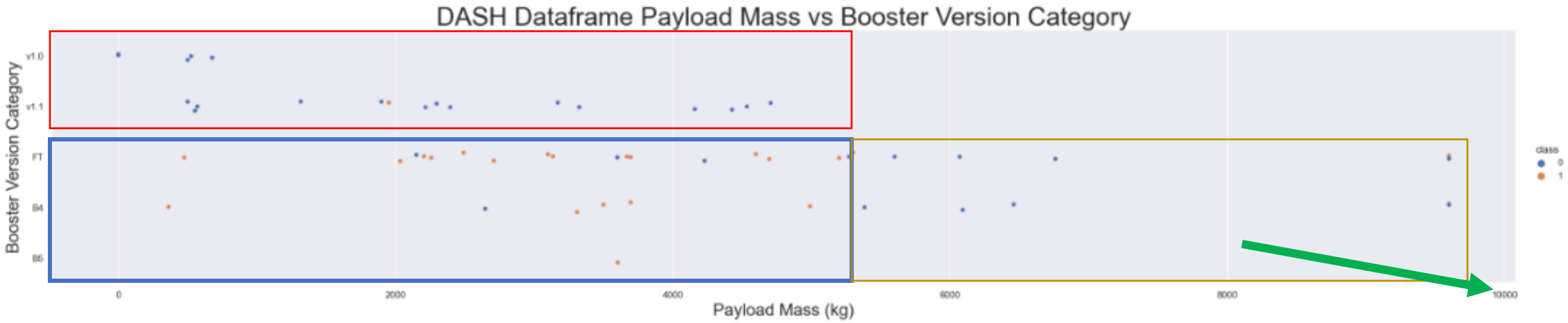
Since those versions are obsolete and have never been used anymore, we can safely assume that, for future launches, it is better to consider launches from FT boosters onward, because **previous booster problems have been already solved** and optimized with the newer versions.

One more question emerged though? How about all those failures in the **last launches**(golden circle)?

Since the booster unreliability was out of question at that time, it could have been a problem of **Payload Mass**. Or maybe the **Orbit** was involved? Another research was required.

Further research of possible Correlations part 3

Using the DASH dataframe once more, I confronted Payload Mass and Booster Version Category. Unfortunately, this dataframe only contains 10.000kg payloads or less. As we witnessed in the "Payload vs Launch Site" on page 18, flights with payloads heavier than 10.000kg showed almost pristine success rate.



1) As we said, all the older versions (V1 and V1.1) show a huge failure percentage (red rectangle)
2) The success rate increases dramatically with **FT or newer boosters** and payloads under 5500kg (blue rectangle)
3) Conversely, the **failures** for the newer booster versions were all caused by medium-heavy payload mass (golden rectangle). We know however that very heavy payloads (10000+) showed a very high success rate, so we are left with this range that apparently performed worse. Consulting page 21 "Payload vs Orbit Type" it appears that many launches with **payloads between 5.500 and 10.000kg** were directed for the **GTO orbit**, substantially higher than the average orbit. What were the booster versions for those launches?

Unfortunately, as I mentioned, this dataframe doesn't include payloads over 10.000kg (green arrow) nor Orbit types, and the original dataframe doesn't include the various Boosters Category, so I cannot make more precise correlations. I have however enough material for further research.

Further research of possible Correlations part 4

This time I wanted to confront the Orbit and the Booster Version. Since it was not possible to do it directly, I once again confronted the two dataframe using two identical parameters (Launch Site and Class) and alternating Orbit from SPACEX and Booster Version from DASH.

I then added a third chart showing Flight Number, Orbit and Launch Site to cross-test my hypothesis.

Interestingly, I immediately noticed a similar cluster in both of the first charts (red rectangles).

CCAFS LC-40 and CCAFS SLC-40 from DASH seem to be grouped in a single CCAFS SLC-40 in SPACEX. The two facilities are in fact adjacent.

The interesting fact is the correspondence between **V1 and V1.1** Booster Versions, and the **ISS** and especially **GTO orbits**.

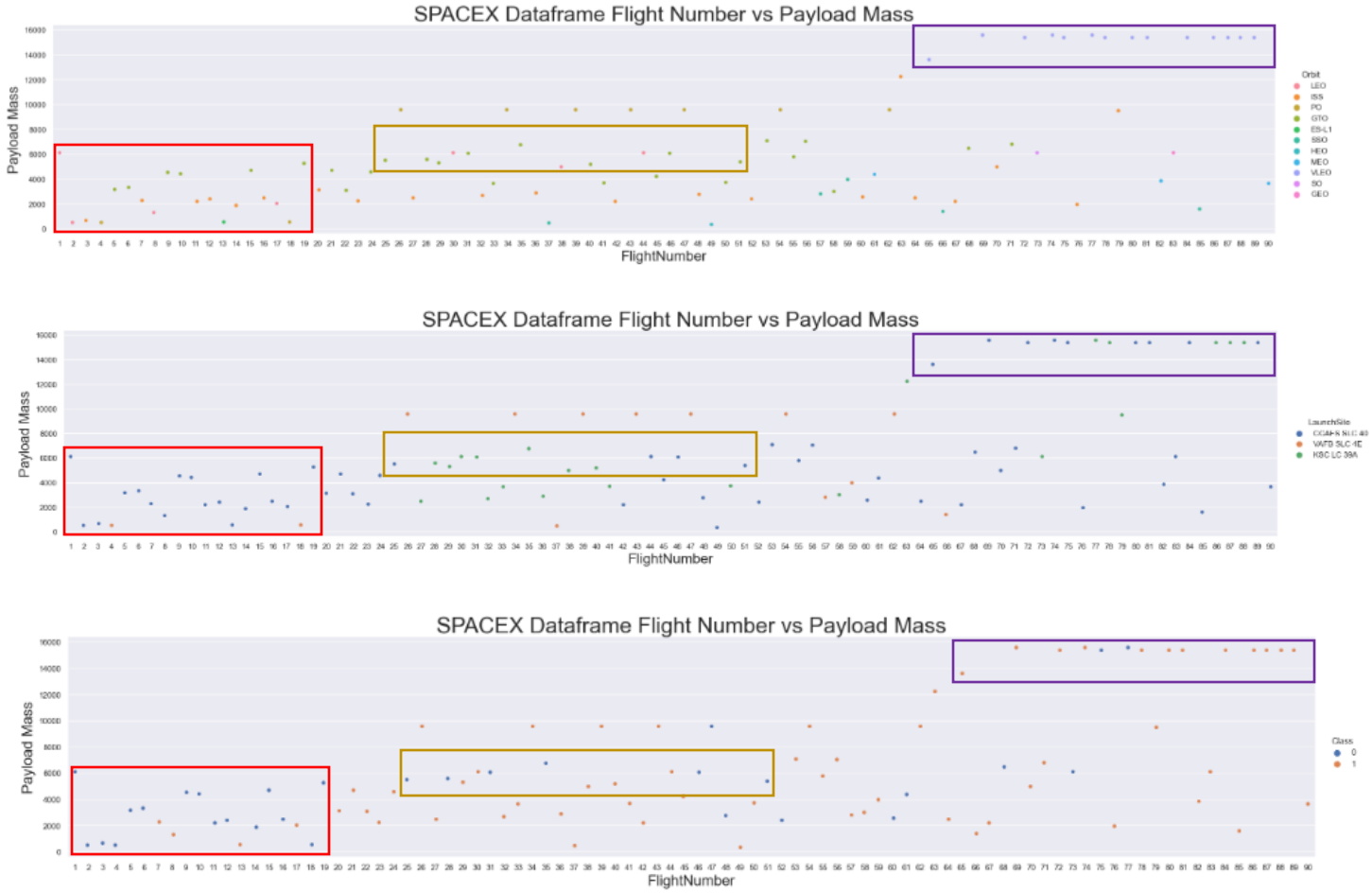
We know for sure from page 20 ("Flight Number vs Orbit type) that 5 GTO launches and 5 ISS launches were made inside the first 20 attempts (orange rectangles), made exclusively with V1 and V1.1 boosters.

Those two orbits are very different: ISS is a low orbit and GTO is quite high.

We can then conclude that a big part of **ISS and GTO orbits** launches depends on obsolete boosters, rather than the distance from Earth.

Further research of possible Correlations part 5

Once I clarified the origin of most failures, excluding variables that could have likely had a role (Orbit and Payload), I wanted to take a last look at the SPACEX dataframe, in order to zoom out and better understand the big picture. I made three charts with the same X and Y axis (Flight Number vs Payload Mass) and confronted them using three different hues (you can see them in the legends): Orbit, Launch Site and Class.



Three areas of interest were found:

1. The last launches (purple rectangle) show a sudden and constant **trend of launches to VLEO** (Very Low Earth Orbit). The interesting thing is that this launches so close to the Earth display the heaviest payloads, once again showing that payload and distance are not directly related. Also, these last flights show a very high success rate, demonstrating that payload and success rate are not correlated.
2. The red rectangle has been highly explored: the first versions of the boosters (**V1 and V1.1**) determined the very high failure rate here.
3. As for the golden rectangle, this remains less clear to me. Here we can find the **GTO** orbit with medium payload, and the average rate of failure is high. I have found no evident correlation to explain this, but I must point out that the success rate for GTO launches greatly raised over time.

I cannot directly confront Orbit and Booster Category, so my research comes to an end and it's time for conclusions.

You can find the additional researchs as "BONUS TASK 1-7" on my EDA with Data Visualization notebook on GitHub (Ctrl + click HERE)

Conclusions

- After excluding any other factor, I can conclude that the most important factor for increasing the success rate have been the **Boosters** (FT version or more recent).
- There seem to be no direct correlation between distance of the orbit (nor payload mass) and the success rate.
- An interesting trend has lately recently emerged in very low Orbit (VLEO), that have become very popular and have very high success rate.

SpaceX has found its competitive advantage with newer boosters and more low orbit flights

Appendix

Special Thanks to:

- **IBM and Coursera** for their financial aid program, without which I could not have attended their course.
- The person who made me discover Coursera in the first place: **thank you R!**

Thank you!

IBM Developer
SKILLS NETWORK