IBM Developer
SKILLS NETWORK

SPACEY

Winning Space Race with Data Science

Manu Cañete
Data Scientist Candidate

May 2025

Outline

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

Executive Summary

- **Summary of methodologies**
  - Web scraping and API access to collect raw data.
  - Data exploration and analysis using Python and SQL.
  - Plotting graphics and drawing maps to better understanding data.
  - Train, validate and test Machine Learning models to predict outcomes.
  - Show insights adapted to the audience.
- **Summary of results**
  - Data analysis can assist a brand-new space company to go to market.
  - Data can be collected from different sources and will need for cleaning, wrangling and analysis.
  - Machine learning models can predict outcomes using proper data and training.
  - Adapt messages and storytelling to audience is key to success.

Introduction

- **Project background and context.**
  - The new company takes profit from the existing and free data from the existing SpaceX company. Using data science techniques will be able to predict the success ratio of the launching module and therefore predict the cost of the mission. The available free data jointly with the modern analysis techniques and prediction models provide the perfect framework for the development and success of the brand-new company SpaceY.
- **Problems you want to find answers.**
  - The main problem that faces the new company is becoming competitive in a market leaded by the company SpaceX. Ability to reuse the first launching stage from the rocket is crucial for costs contention. Therefore, predicting the success ratio of the rocket launch (i.e. predicting is that first stage will properly land and ready to reuse) will be able to know costs in advance and provide the best competitive price to the potential customers.

Section 1

Methodology

Methodology: Summary

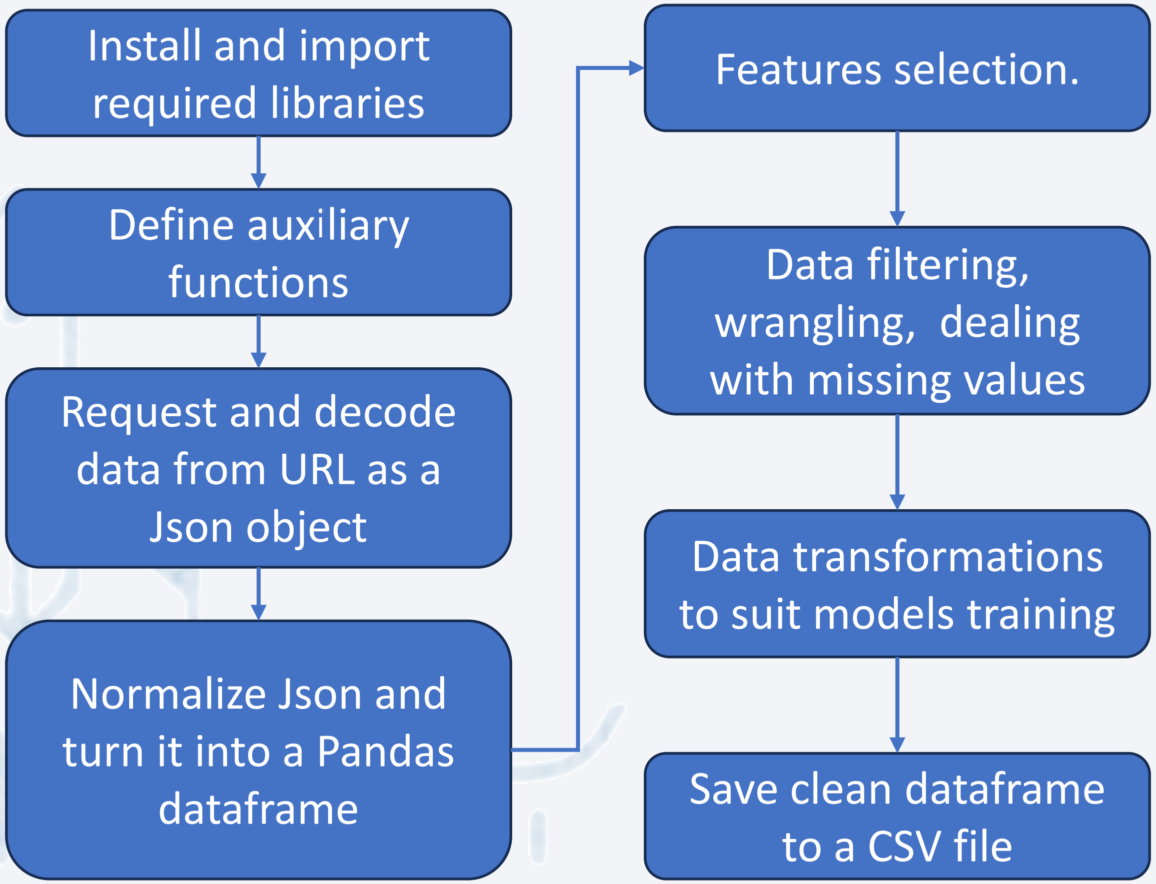
- Data collection methodology:
  - Collecting data from SpaceX sources through an **API**.
  - **Web-scraping** on the Wikipedia SpaceX tables.
- Perform data wrangling
  - Identification of relevant **features and target**, data type **homogenization and conversion** to train models to get the most accurate predictions.
- Perform exploratory data analysis (EDA) using **visualization and SQL**.
- Perform interactive visual analytics using **Folium** and **Plotly Dash**.
- Perform predictive analysis using classification models
  - **Feature selection** for efficient training, split into **train/test datasets** to validate models, tune **hyperparameters** to achieve the best metrics and avoid underfitting/overfitting, calculate **metrics** for each tuned model and **compare** models using metrics and confusion matrix.

Data Collection Methodology – SpaceX API

- Jupiter Notebook with Python code and results on GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/jupyter-labs-spacex-data-collection-api-v2-MCC.ipynb

API flow chart

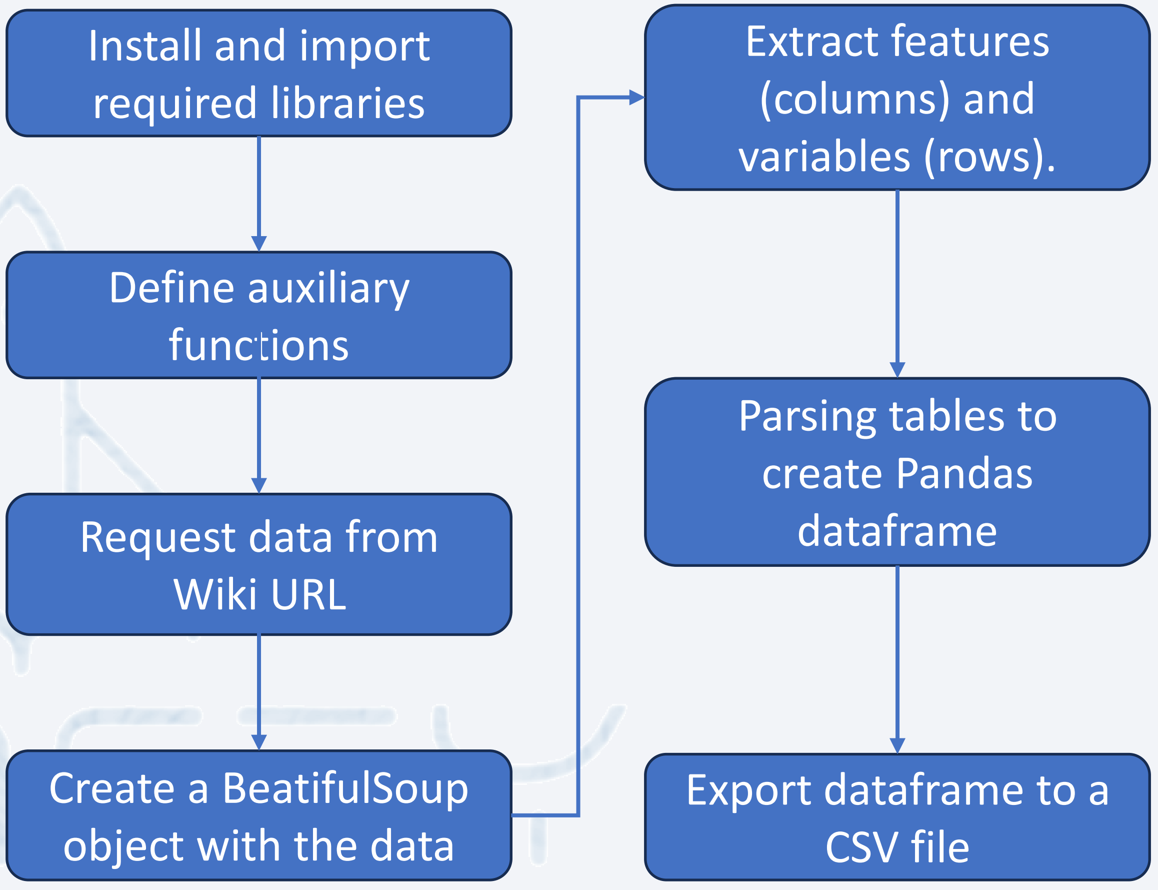


Data Collection Methodology – Web-Scraping

- Jupiter Notebook with Python code and results on GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/jupyter-labs-webscraping-MCC.ipynb

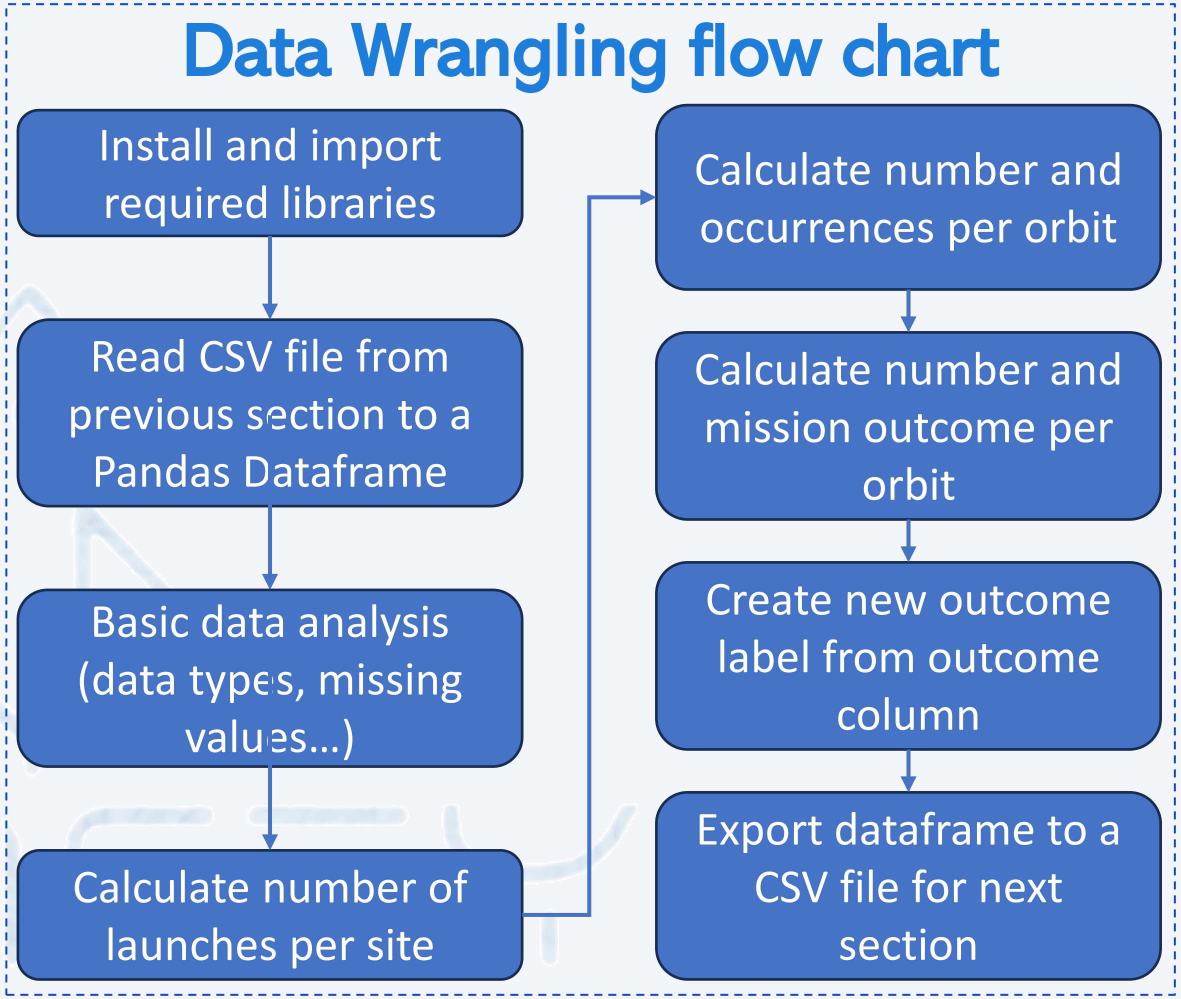
Web-Scraping flow chart



Data Wrangling Methodology

- Jupiter Notebook with Python code and results on GitHub:

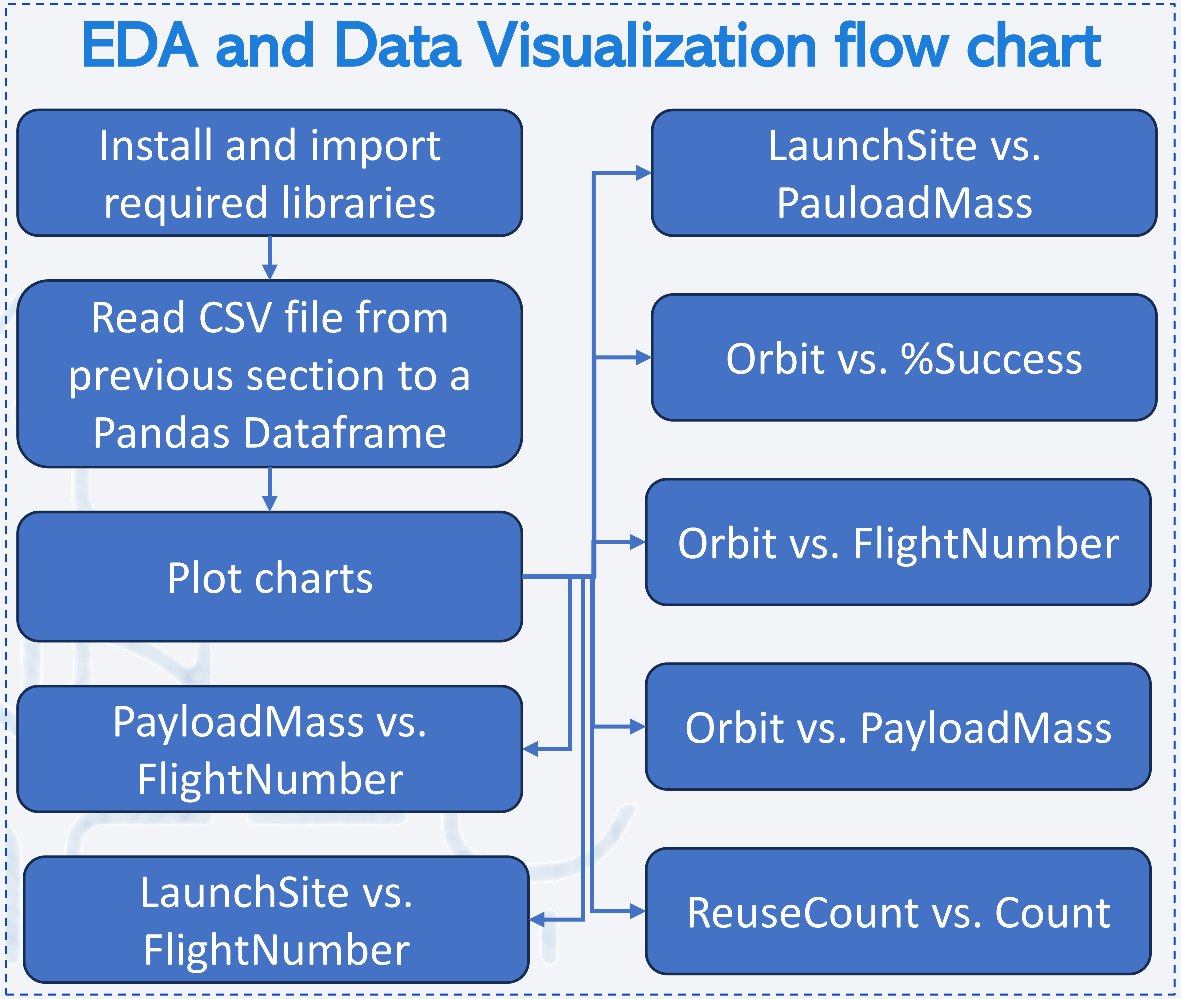
https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/labs-jupyter-spacex-Data%20wrangling-v2-MCC.ipynb



EDA and Data Visualization Methodology

- Jupiter Notebook with Python code and results on GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/jupyter-labs-eda-dataviz-v2-MCC.ipynb

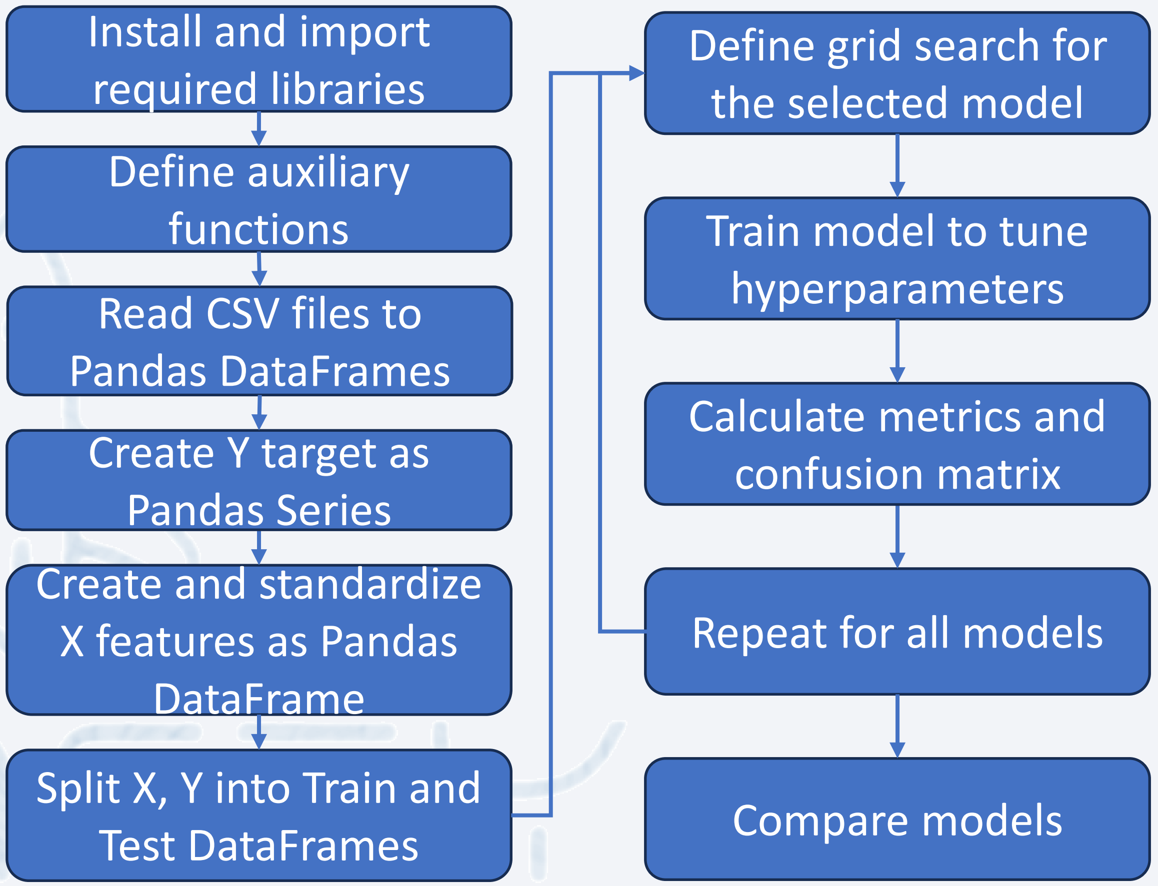


Predictive Analysis Methodology

- Jupiter Notebook with Python code and results on GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1-MCC.ipynb

Predictive Analysis flow chart



Section 2

Insights drawn from EDA

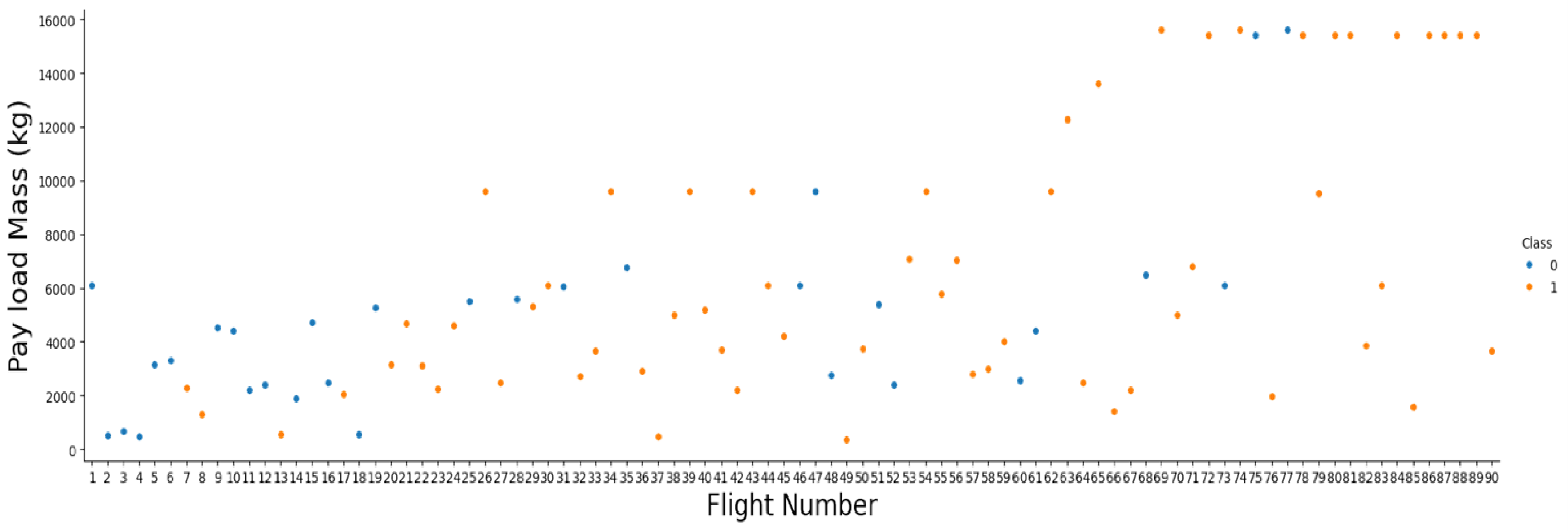
EDA with Data Visualization (1)

- Find EDA with visualization code and results in GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/jupyter-labs-eda-dataviz-v2-MCC.ipynb

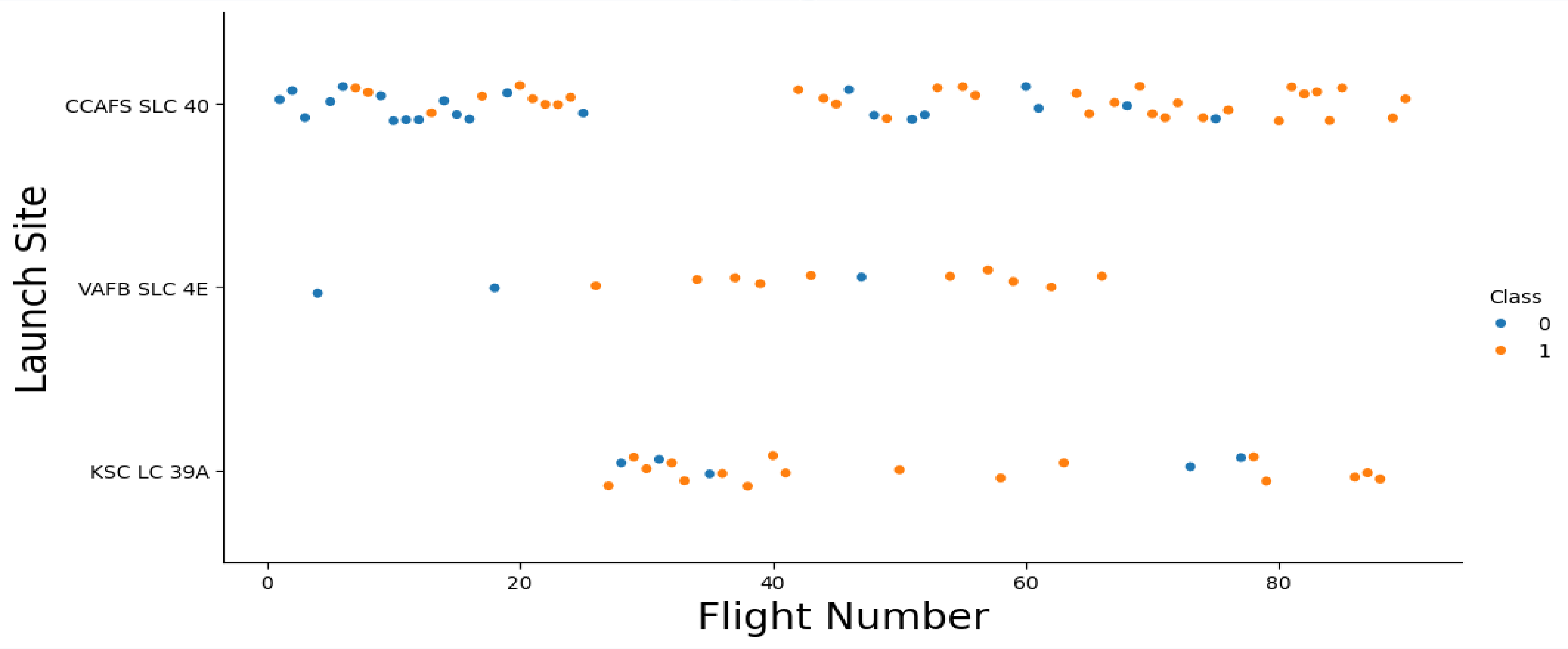
EDA with Data Visualization (2)

- **Chart**: PayloadMass vs. FlightNumber. Class 0 = fail (cyan). Class 1 = success (orange)
- **Insight 1**: as the flight number increases, the first stage is more likely to land successfully, showcasing the positive evolution of the technology and the solution.
- **Insight 2**: there are more orange dots below 8,000 kg payload than above, so it seems that the lower the payload the higher the probability of success.



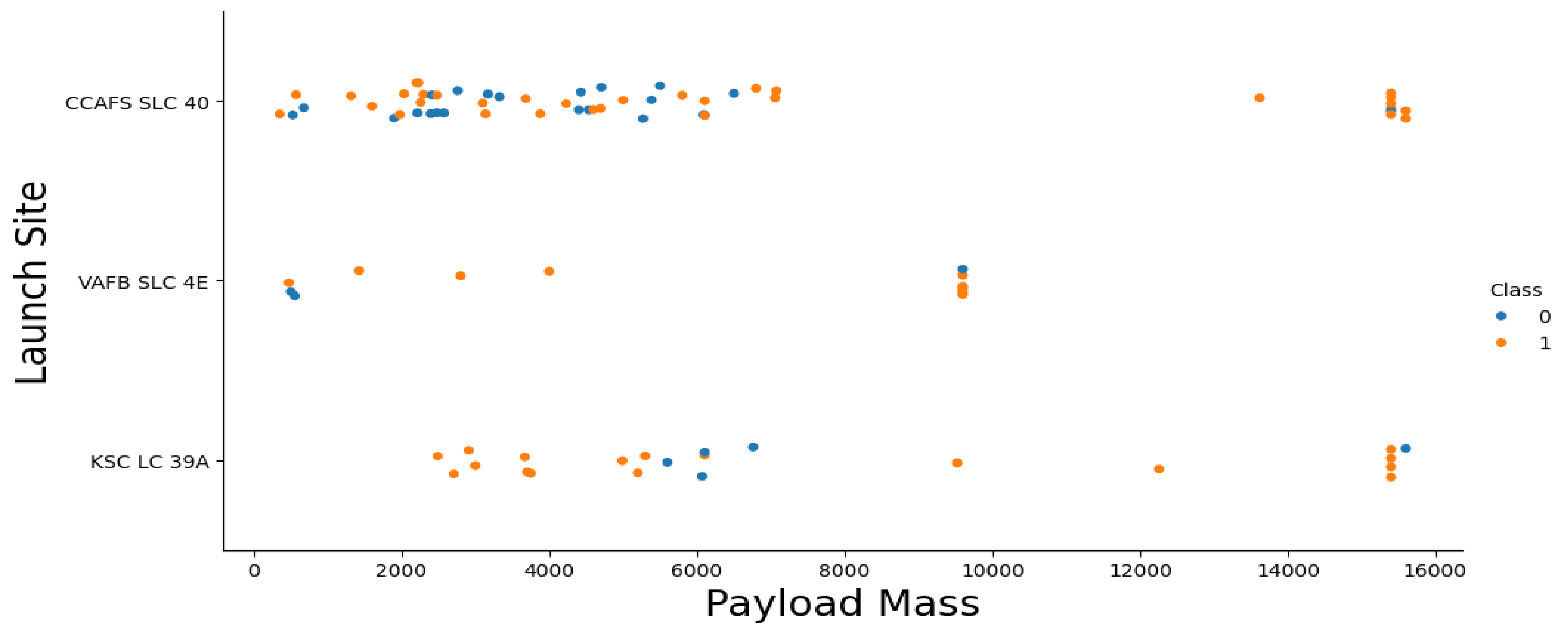
EDA with Data Visualization (3)

- **Chart**: FlightNumber vs. LauchSite. Class 0 = fail (cyan). Class 1 = success (orange)
- **Insight 1**: All three sites increase their success ratio as number of flights go ahead, so success appears to depend not on the site but on the evolution and fine tuning of the technology.
- **Insight 2**: Success ratios for VAFB SLC 4E and KSC LC 39A are almost the same, while CCAFS SLC 40 one is much lower. Need further analysis to find the reasons.



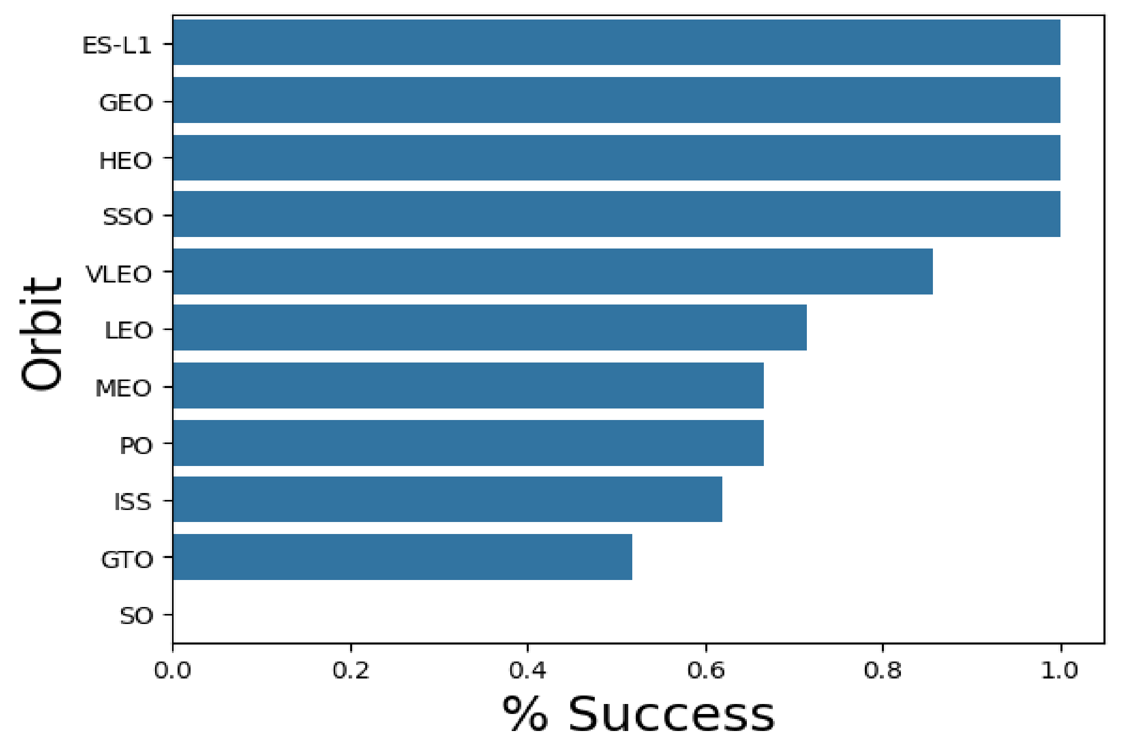
EDA with Data Visualization (4)

- **Chart**: Payload vs. LauchSite. Class 0 = fail (cyan). Class 1 = success (orange)
- **Insight 1**: For heavy payloads (over 8,000kg) the success ratio is much higher than for lighter payloads (under 8,000kg).
- **Insight 2**: VAFB SLC 4E does not have any heavy payload launch. Need further analysis to find our if it is a limitation or there are other reasons.



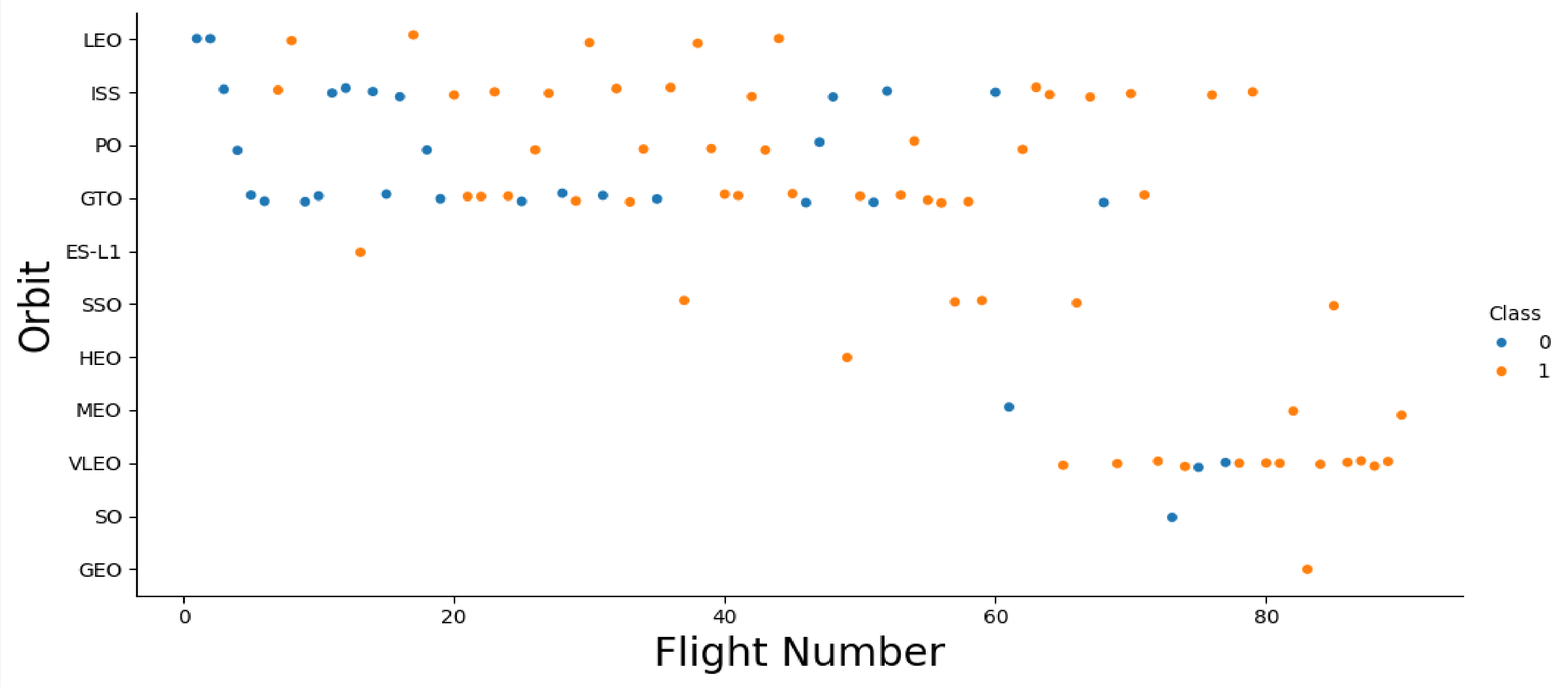
EDA with Data Visualization (5)

- **Chart**: % Success Rate vs. Orbit Type
- **Insight 1**: Four orbits (ES-L1, GEO, HEO, SSO) show complete success, but the number of launches for them is quite low (less than 10% of the total launches) so we cannot draw any conclusions.
- **Insight 2**: For the most popular orbits (GTO, ISS, VLEO), their success ratios are higher than 50%, being VLEO ratio the highest.



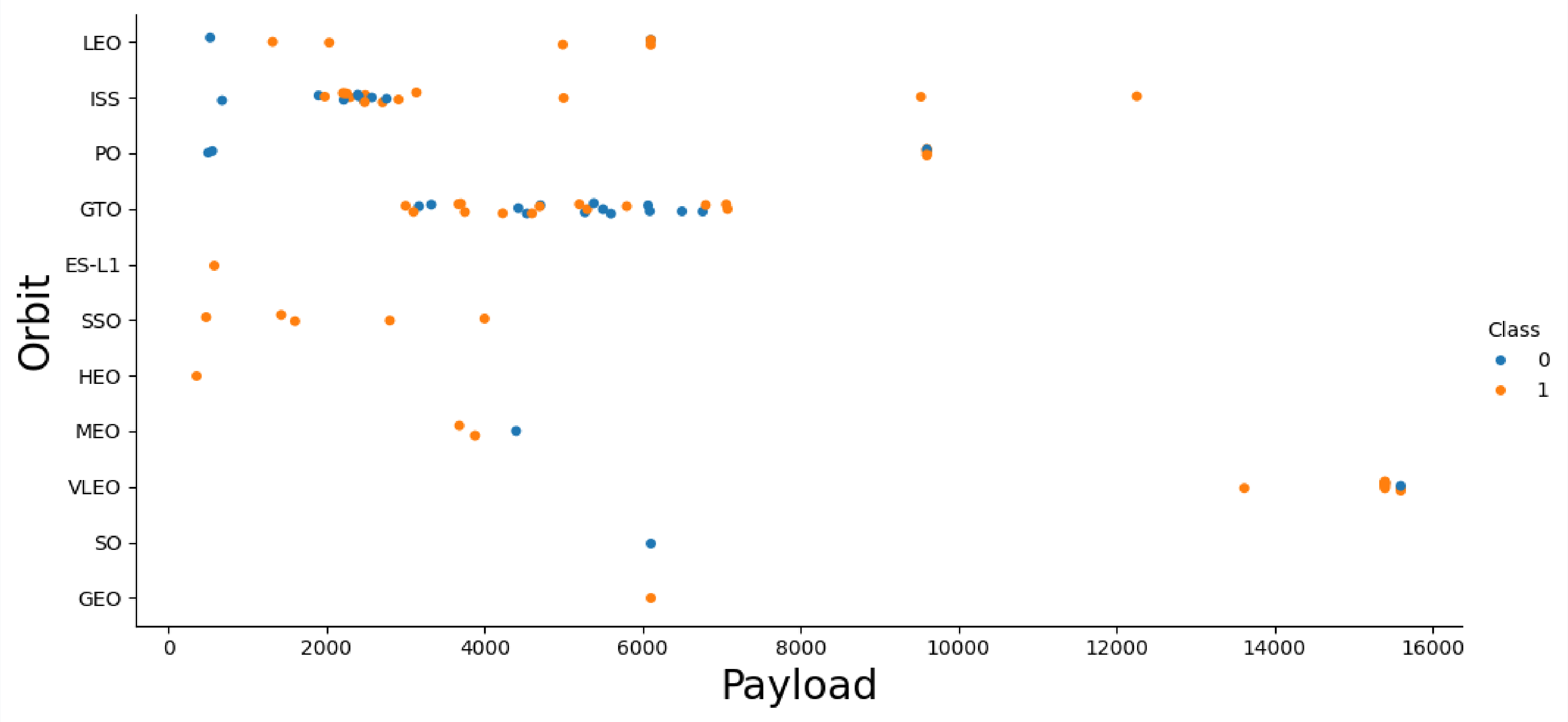
EDA with Data Visualization (6)

- **Chart**: Flight Number vs. Orbit Type. Class 0 = fail (cyan). Class 1 = success (orange)
- **Insight 1**: For the LEO, ISS and VLEO orbits, successful launches happen after a certain number of flights.
- **Insight 2**: For the GTO and PO orbits we have a mix of success and fail, even as the number of flights increases. All together suggests that each orbit requires a specific setup and these two orbit needs more flights to achieve the perfect system setup.



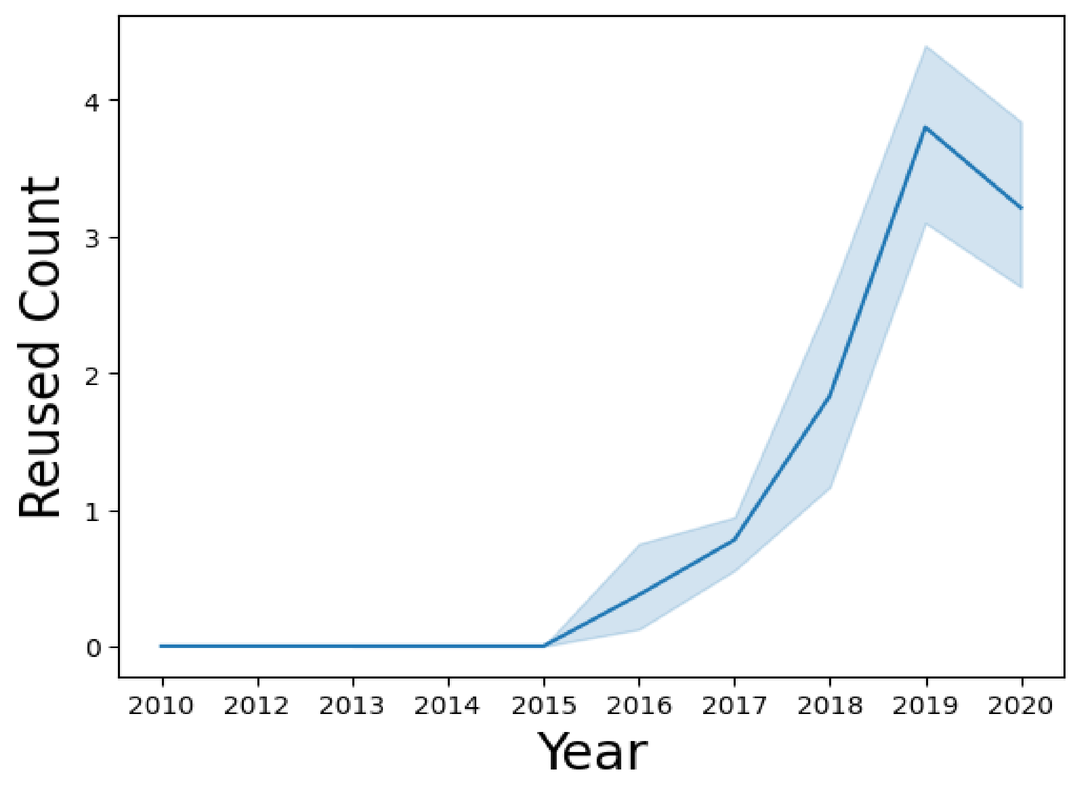
EDA with Data Visualization (7)

- **Chart**: Payload vs. Orbit Type. Class 0 = fail (cyan). Class 1 = success (orange)
- **Insight 1**: For the LEO and ISS orbits there are only success launches for their higher payloads.
- **Insight 2**: We don't see that trend for the GTO orbit. For the rest, we have few values to draw conclusions.



EDA with Data Visualization (8)

- **Chart**: Reused Count vs. Year.
- **Insight 1**: After five years of technological development [2010-2015], the company achieved its first successful launches.
- **Insight 2**: Since 2015 technology has matured, allowing the company to reuse the same first stage several times.



EDA with SQL (1)

- Find all the SQL requests and results in GitHub:

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/jupyter-labs-eda-sql-coursera_sqllite-MCC.ipynb

EDA with SQL (2)

- Display the names of the unique launch sites in the space mission.

```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

- Display 5 records where launch sites begin with the string 'CCA'

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

- Display the total payload mass carried by boosters launched by NASA (CRS)

```
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_MASS_NASA_CRS FROM SPACEXTABLE WHERE CUSTOMER
LIKE 'NASA (CRS)%';
```

- Display average payload mass carried by booster version F9 v1.1

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

- List the date when the first successful landing outcome in ground pad was achieved.

```
SELECT MIN(DATE) FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (ground pad)’;
```

EDA with SQL (3)

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

```
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTABLE WHERE LANDING_OUTCOME LIKE '%Success
(drone ship)%' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

- List the total number of successful and failure mission outcomes.

```
SELECT
    COUNT(*) FILTER (WHERE UPPER(Mission_Outcome) LIKE '%success%') AS Mission_Success,
    COUNT(*) FILTER (WHERE UPPER(Mission_Outcome) LIKE '%failure%') AS Mission_Failure
FROM SPACEXTABLE;
```

- List all the booster versions that have carried the maximum payload mass.

```
SELECT
    substr(Booster_Version,1,INSTR(Booster_Version||' B’,' B’)-1) AS Booster_Main_Version,
    MAX(PAYLOAD_MASS__KG_) AS Max_Payload
FROM SPACEXTABLE GROUP BY Booster_Main_Version ORDER BY Max_Payload DESC;
```

EDA with SQL (4)

- 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.

```
SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site FROM SPACEXTABLE
    WHERE substr(Date, 1, 4) = '2015' AND Landing_Outcome = 'Failure (drone ship)'
    ORDER BY Month;
```

- 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.

```
SELECT Landing_Outcome, COUNT(*) AS Attempts FROM SPACEXTABLE
    WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Attempts DESC;
```

EDA with SQL (5)

- Find the names of the unique launch sites
  - Using command DISTINCT to assure unique values.

```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

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

EDA with SQL (6)

- Find 5 records where launch sites begin with `CCA`
  - Using command WHERE to find in the LAUNCH_SITE columns, LIKE with argument 'CCA% for beginning with, then command LIMIT to show only 5 records.

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

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

EDA with SQL (7)

- Calculate the total payload carried by boosters from NASA
  - Using commands WHERE and LIKE to filter NASA boosters, then command SUM to create a new value with the total mass of the filtered records.

```
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_MASS_NASA_CRS FROM
SPACEXTABLE WHERE CUSTOMER LIKE 'NASA (CRS)%';
```

TOTAL_MASS_NASA_CRS
48213

EDA with SQL (8)

- Calculate the average payload mass carried by booster version F9 v1.1
  - Using commands WHERE and LIKE to filter F9 version boosters, then command AVG to create a new value with the average payload mass of the filtered records.

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

EDA with SQL (9)

- Find the dates of the first successful landing outcome on ground pad
  - Using command WHERE to strictly find the desired outcome, then command MIN(DATE) to get de first date.

```
SELECT MIN(DATE) FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (ground pad)’;
```

MIN(DATE)
2015-12-22

EDA with SQL (10)

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - Using command WHERE with a triple condition (landing outcome, minimum payload, maximum payload), then command DISTINCT to ensure unique values fulfilling the conditions.

```
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTABLE WHERE LANDING_OUTCOME
LIKE '%Success (drone ship)%' AND PAYLOAD_MASS__KG_ > 4000 AND
PAYLOAD_MASS__KG_ < 6000;
```

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

EDA with SQL (11)

- Calculate the total number of successful and failure mission outcomes
  - Using commands COUNT and FILTER twice (for success and failure) to get the required values.

```
SELECT COUNT(*) FILTER (WHERE UPPER(Mission_Outcome) LIKE '%success%') AS
Mission_Success, COUNT(*) FILTER (WHERE UPPER(Mission_Outcome) LIKE
'%failure%') AS Mission_Failure FROM SPACEXTABLE;
```

Mission_Success	Mission_Failure
100	1

EDA with SQL (12)

- List the names of the booster which have carried the maximum payload mass
  - Using command SUBSTR to have a subquery and create two new records, then command GROUP for grouping by booster and finally command ORDER (descending) by maximum payload.

```
SELECT substr(Booster_Version,1,INSTR(Booster_Version||' B’,' B’)-1) AS
Booster_Main_Version,MAX(PAYLOAD_MASS__KG_) AS Max_Payload FROM SPACEXTABLE
GROUP BY Booster_Main_Version ORDER BY Max_Payload DESC;
```

Booster_Main_Version	Max_Payload
F9	15600
F9 FT	9600
F9 FT	9600
F9 v1.1	4707
F9 v1.0	677
F9 v1.1	500

EDA with SQL (13)

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Using command SUBSTR to have a subquery and create three new records, then command WHERE for double filtering by required date and failure outcome, then finally command ORDER by month.

```
SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site
FROM SPACEXTABLE WHERE substr(Date, 1, 4) = '2015' AND
Landing_Outcome = 'Failure (drone ship)’ ORDER BY Month;
```

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

EDA with SQL (14)

- 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
  - Using command COUNT creating the new required column, then command WHERE filtering the date range, and finally grouping by landing outcome ordered by attempts beginning with the greatest.

```
SELECT Landing_Outcome, COUNT(*) AS Attempts FROM SPACEXTABLE WHERE Date >=
'2010-06-04' AND Date <= '2017-03-20' GROUP BY Landing_Outcome ORDER BY
Attempts DESC;
```

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

Section 3

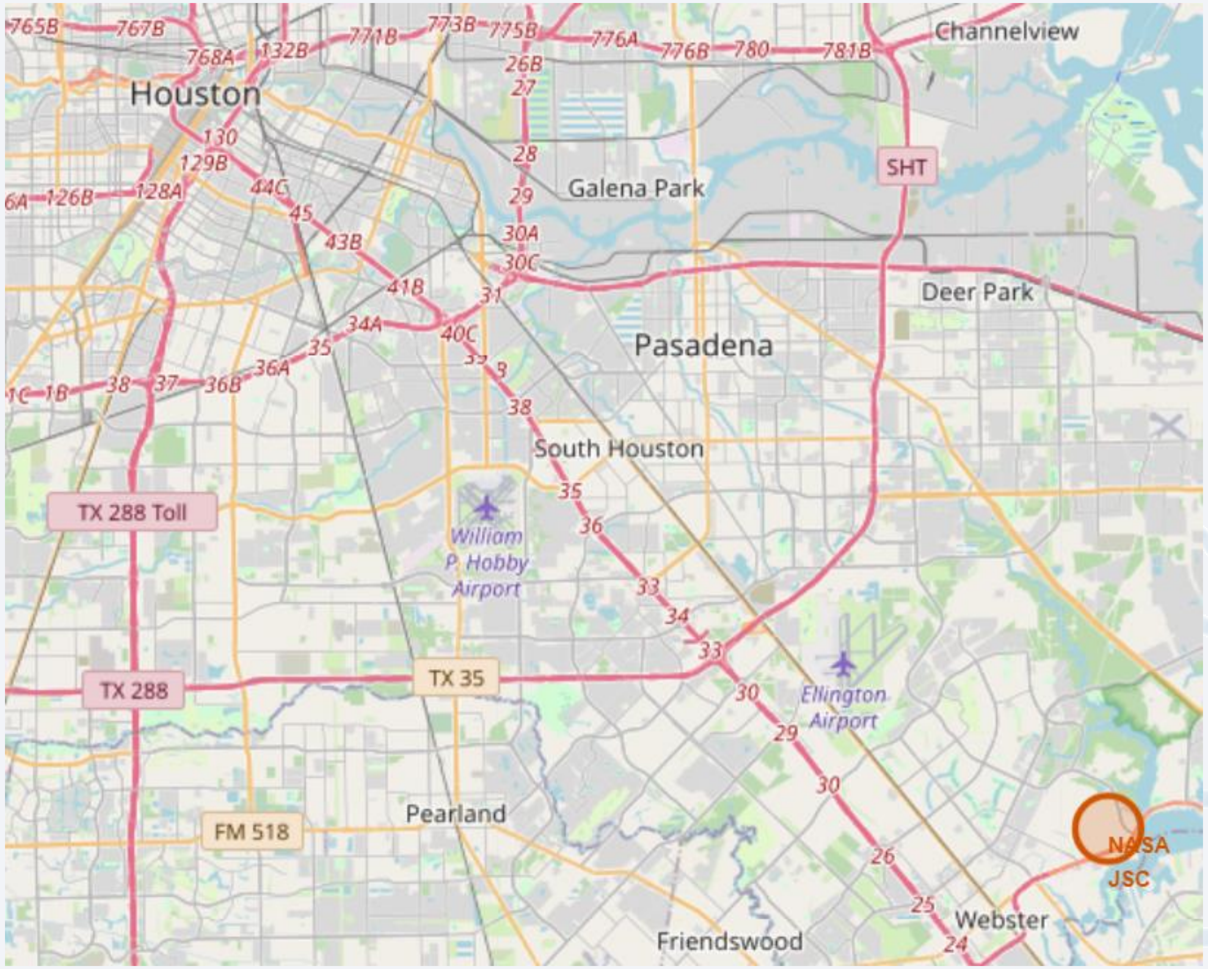
Launch Sites Proximities Analysis

Interactive Map with Folium (1)

- Creating a **Folium Map** centered at NASA Johnson Space Center.
- Adding a **Circle** to highlight NASA JSC, with a **Popup** label and a **Marker** with its name.
- Adding a **Circle** to each launching site, with a **Popup** label and a **Marker** with its name.
- Adding an **Icon** (location type) to each launching site, showing if the outcome was Successful (Green) or Failure (Red).
- Adding an **Icon** (specific type) to each proximity of interest (railway, coastline, airport, road,…).
- Adding a **Line** with distance **Label** from the launching site to each proximity of interest.
- The objects added to the map allows us to have a **visual presentation** of the launches (site location and launch result) as well as the proximities of interest and distances from launching site. The picture provides a **better understanding** of the locations, results and proximities involved, thus easing result interpretation and decision making.

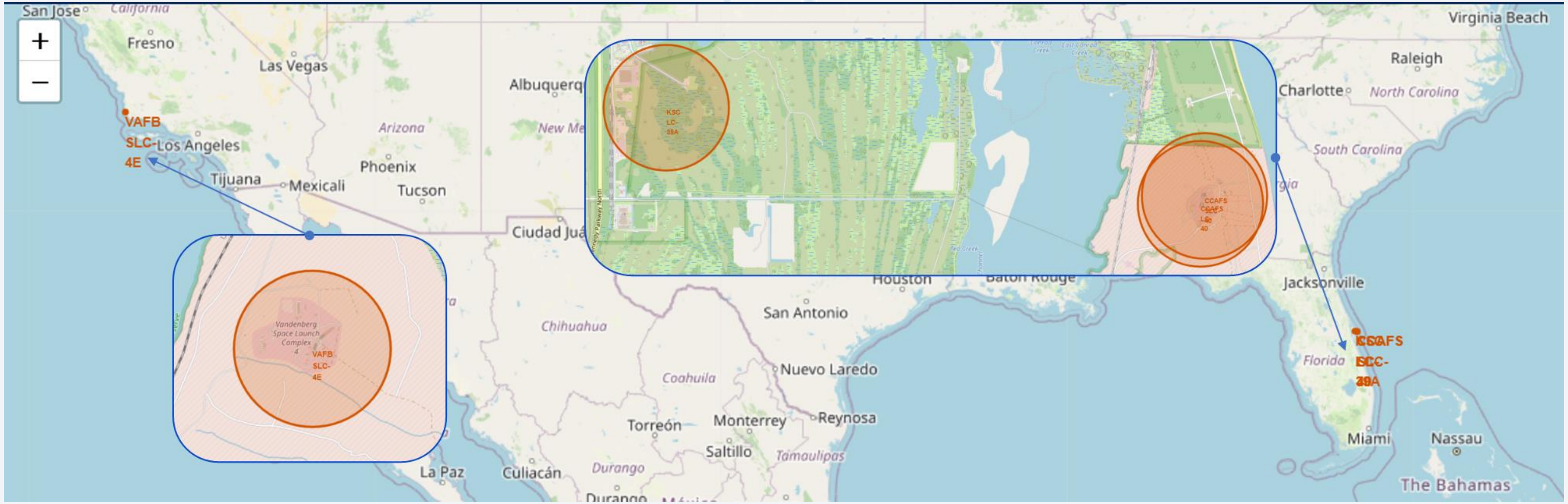
https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/lab-jupyter-launch-site-location-v2-MCC.ipynb

Interactive Map with Folium (2)



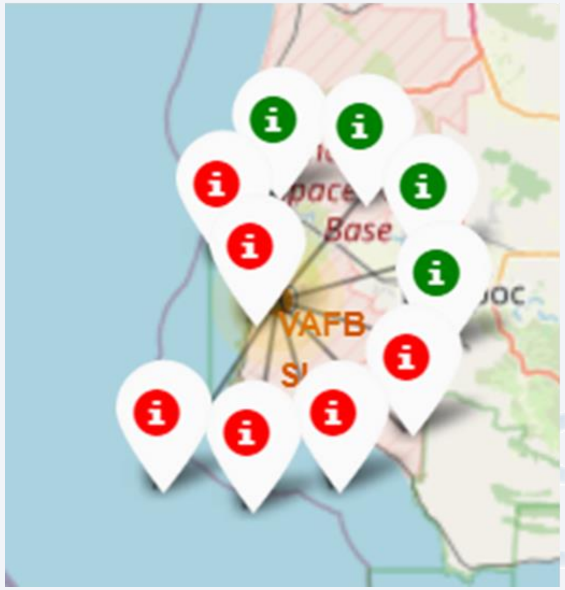
Map: All launch sites

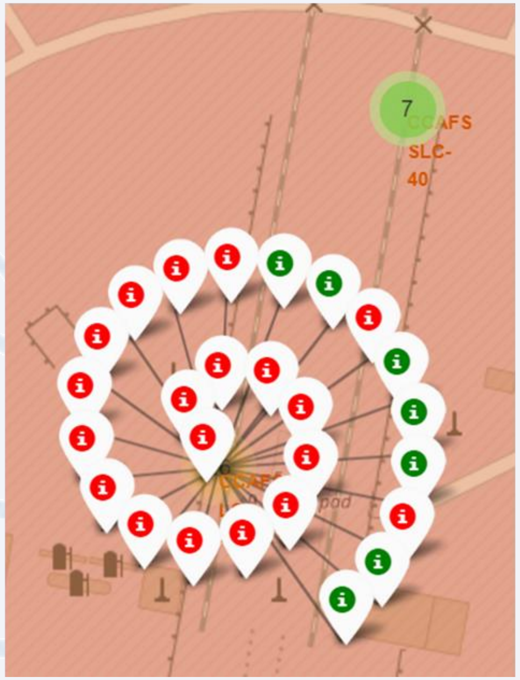
- The map shows the four launch sites. On the street open maps style map, we added a circle centered on the site coordinates and a marker with the name of site.
- Three of the sites are based in Florida (KSC LC, CCAFS LC, CCAFS SLC) and the fourth is based in California (VAFB SLC). Zooming in on the two areas to improve visualization. All four sites are close to the sea (safe areas in case of failure) and close to equator (earth rotation advantage)
- Site diversification allows date and time flexibility to increase number of launches.

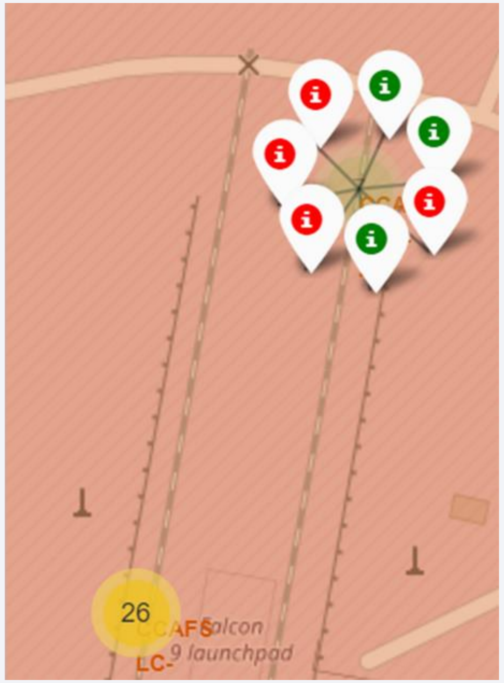


Map: Success/Failed launches for each site

- These maps (zoomed in over each area) show the number of launches from each site. Clicking over the site shows a list of success (green) and failed (red) outcomes.
- From the maps we can observe that KSC LC is the most successful site, while CCAFS LC is the one with less success launches.

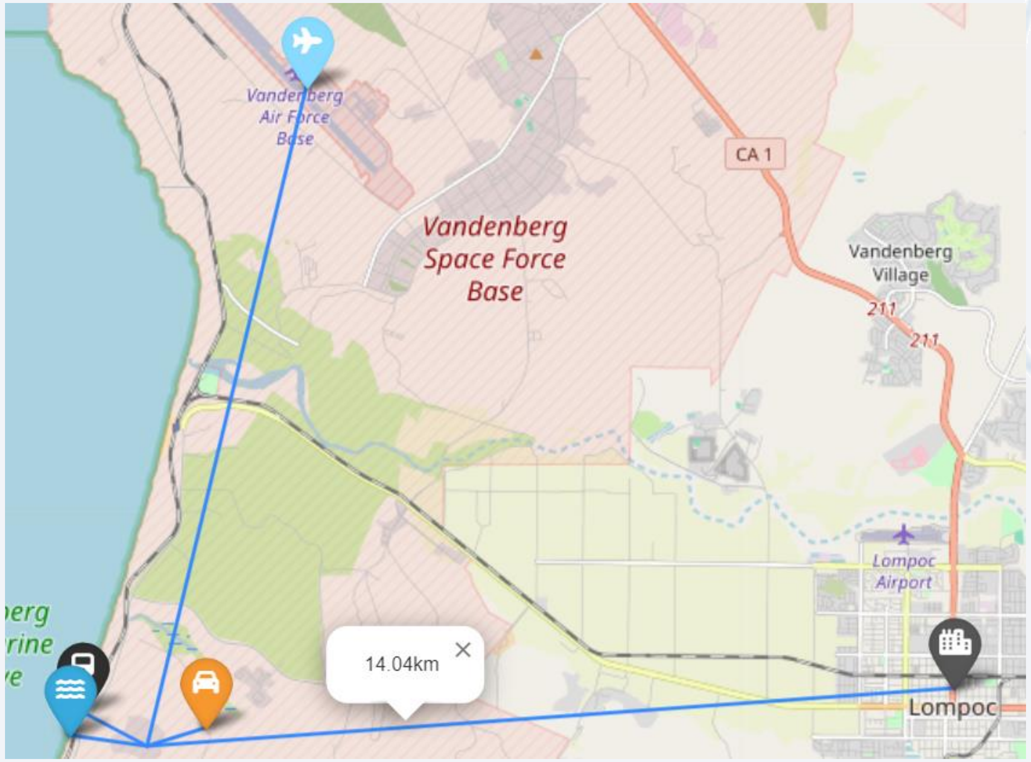






Map: Distances between site and its proximities

- Ground transports (roads, railway) are very close to the launch site (around 1km), and the nearest airport is only 11km away, thus easing the arrival of rocket parts and personnel.
- Coastline is quite close (1,35km) and nearest city is more than 14 km away, thus providing safe in case of incidents with launches and first stage recovery.



Section 4

Build a Dashboard with Plotly Dash

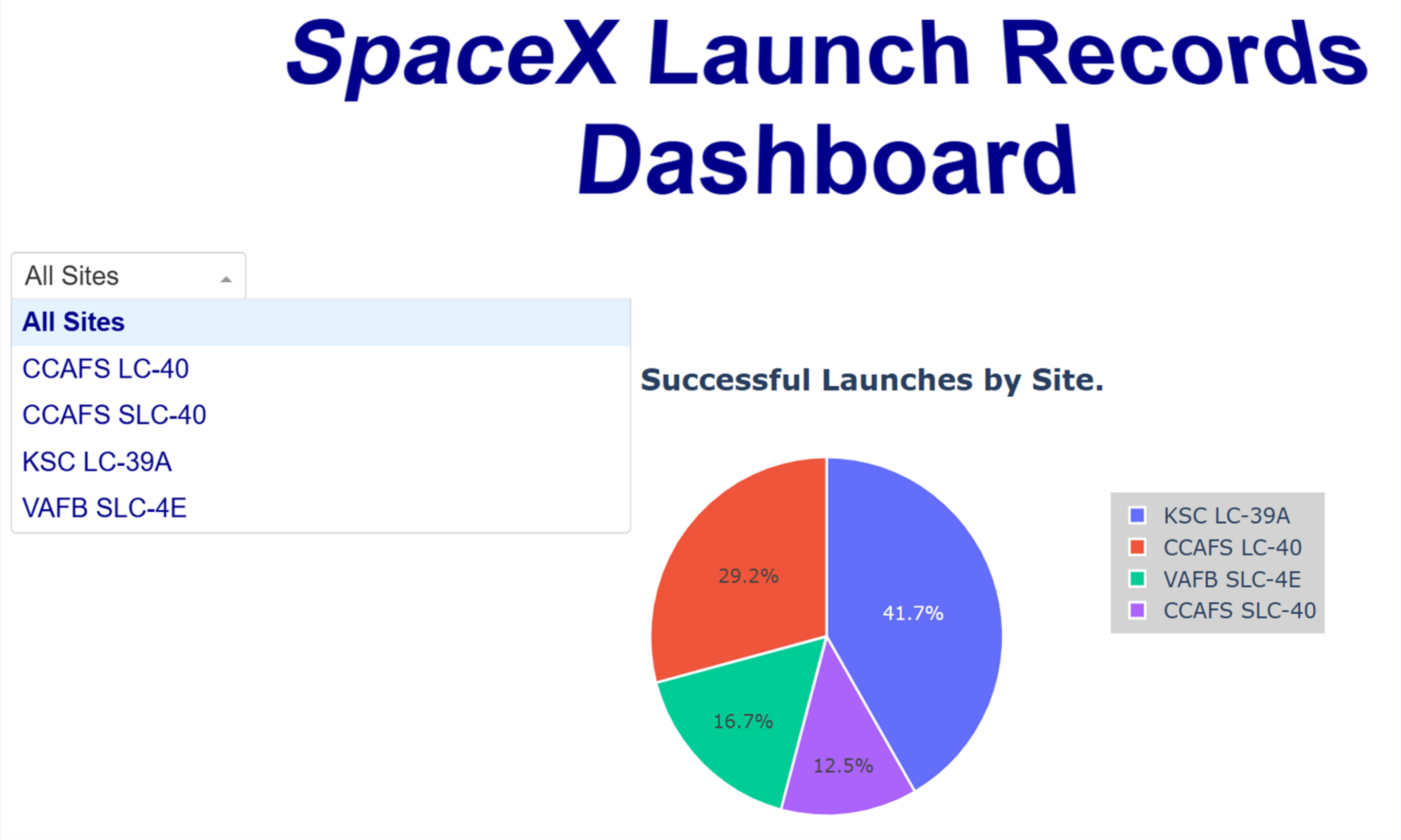
Dashboard with Plotly Dash

- **Successful Launches by Site**: Pie chart showing the success ratio for each site, allowing to find out the most successful site and its ratio.
- **Success ratio by Site:** Pie chart for one site showing the ratio success/failure for that site.
- **Success vs. Payload:** Scatter plot with success/failure outcome, with a different color for each site. We can filter all sites or a payload range using a slide control.
- The objective is gaining understanding about outcomes per site and per payload mass range, then figure out the reasons for such outcome.

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/spacex_dash_app-MCC.py

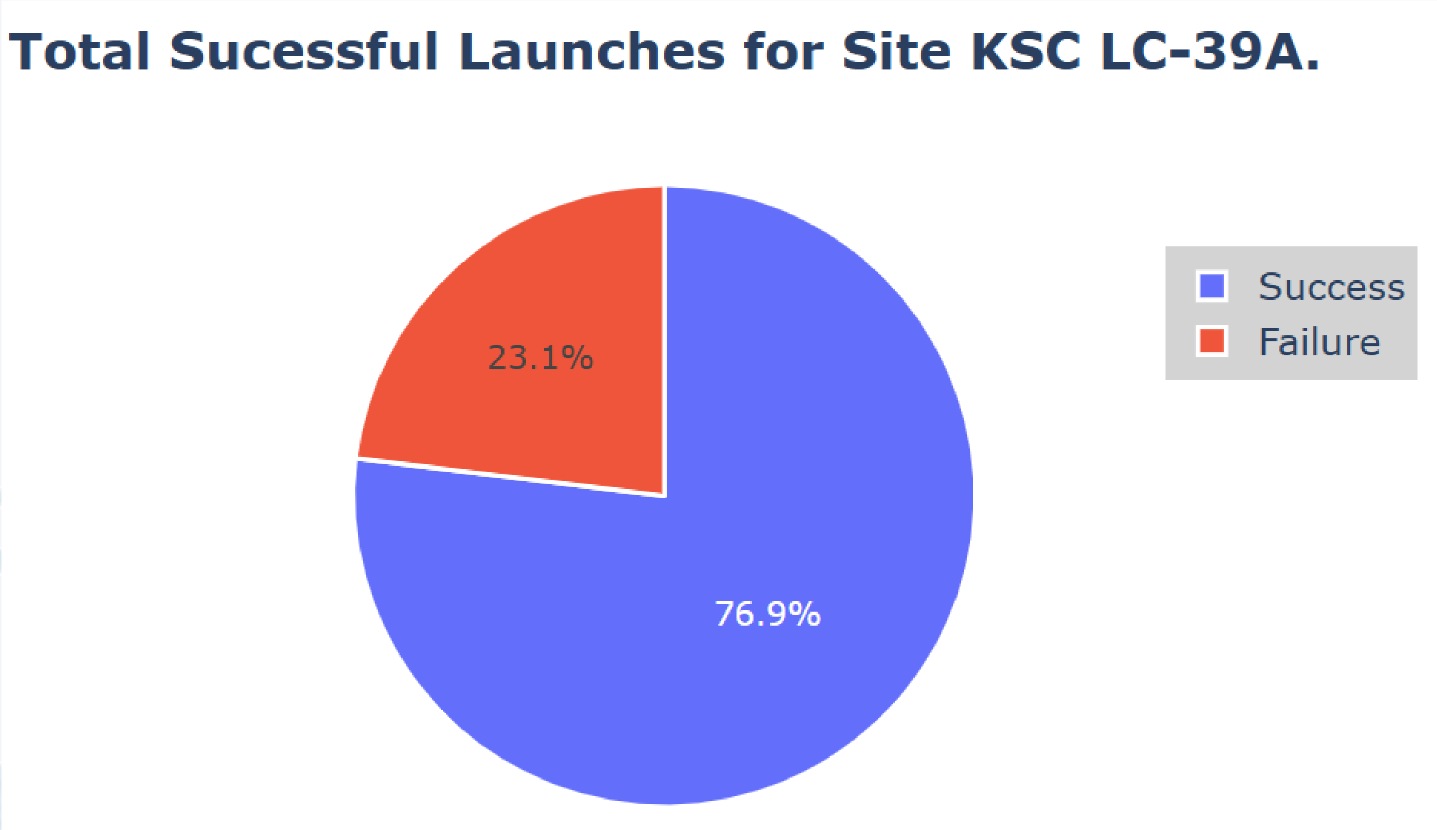
Dashboard 1: Successful Launches by Site

- **Successful Launches by Site**
- Most successful site is KSC LC, having c. 42% of the successes, followed by CCAFS (29%).
- Bear in mind that the picture represents how successful launches are distributed among sites, not a ratio success vs. failure.
- Need additional data (number of launches, payloads, date, booster…) to complement findings.



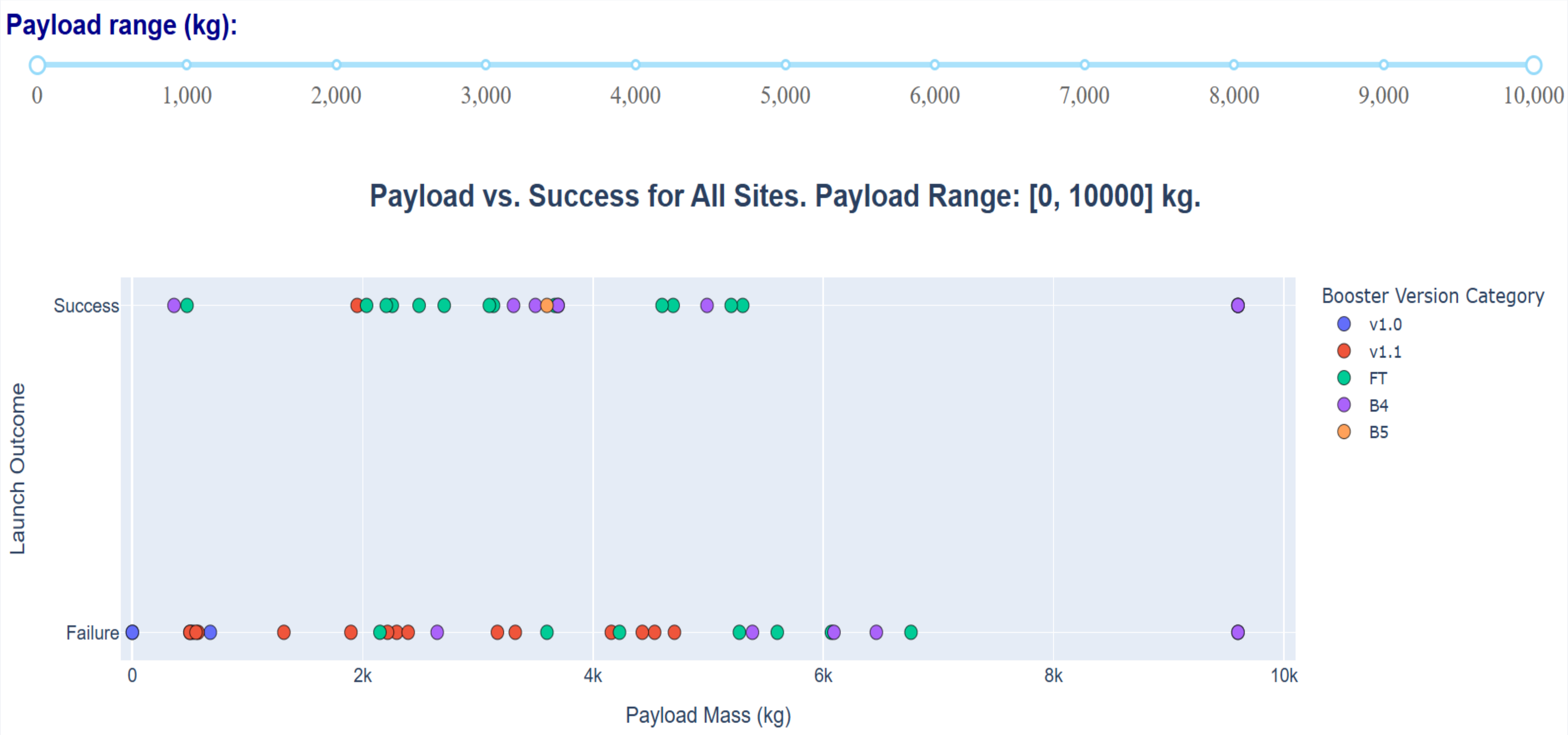
Dashboard 2: Site with highest success ratio

- **Site with highest success ratio**
- KSC LC has the highest launch success ratio, at c. 77%.
- CCAFS LC has slightly worse results (73%). The other sites have ratios above 50%.
- KSC LS is the best site, both in terms of successful launches and success ratio.
- Being located in Florida or California does not appear to be an indicator of success. Need for additional indicators.



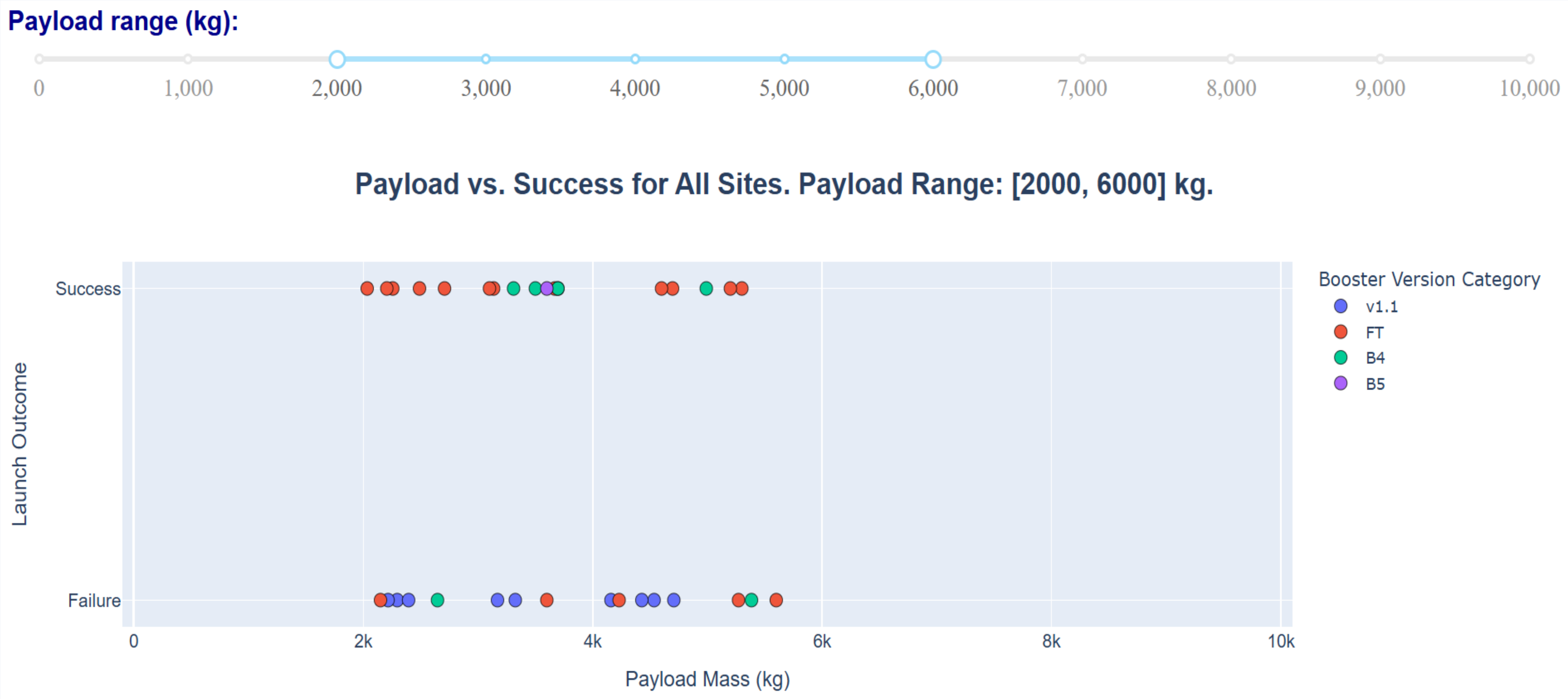
Dashboard 3. Success vs. Payload

- **Success vs. Payload:** All sites, full payload range
- Payload range between 2,000 and 6,000 kg accounts for the majority of successful launches.



Dashboard 4. Success vs. Payload

- **Success vs. Payload:** All sites, payload between 2,000 and 6,000 kg
- Booster FT is the one with the largest success rate.



Section 6

Predictive Analysis (Classification)

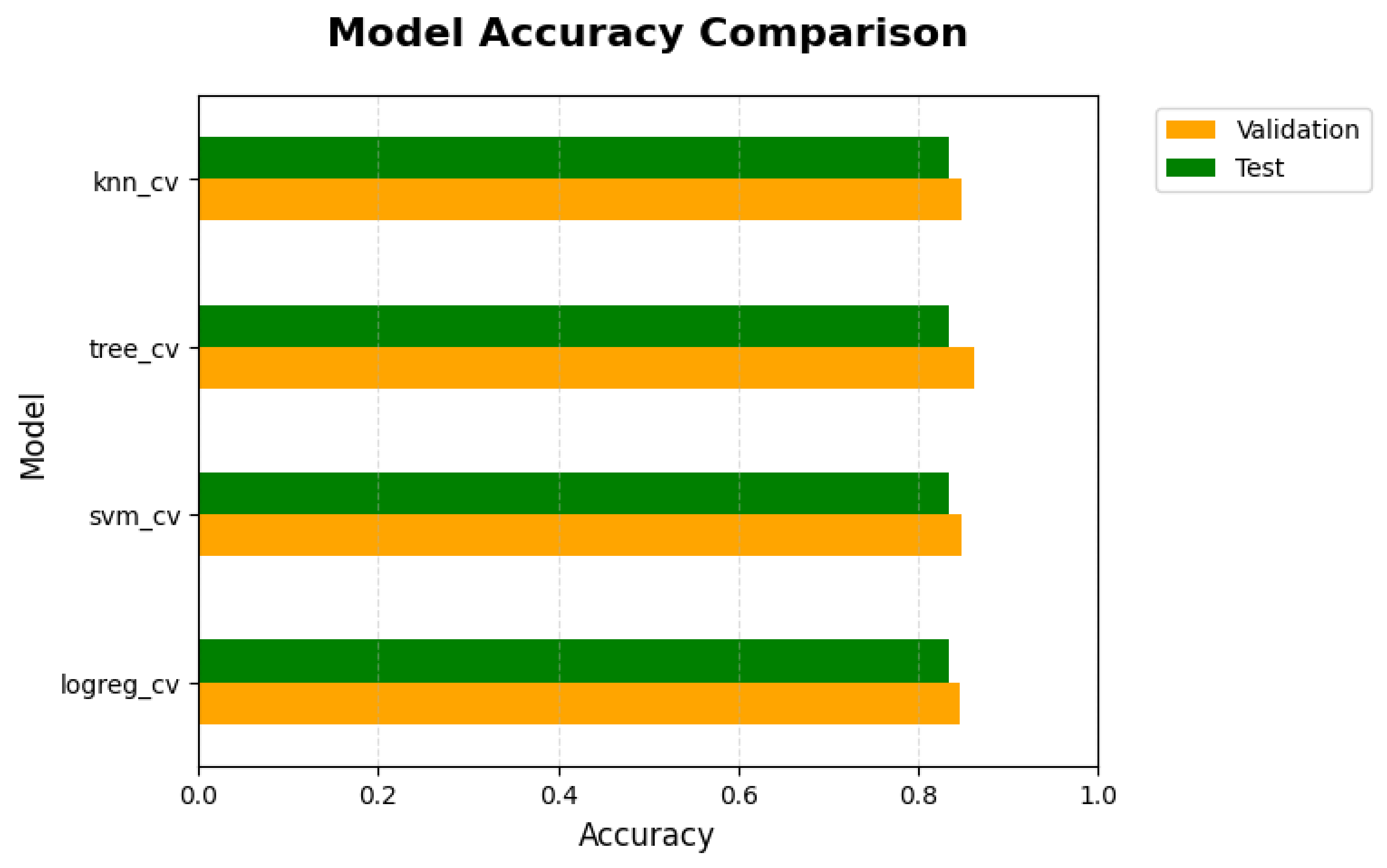
Predictive Analysis (Binary Classification)

- Features selection to ensure an efficient training and normalizing features dataset to ease model fitting.
- Split feature/target datasets into train/test datasets to asses model performance and prevent underfitting/overfitting.
- Select models that can solve a binary classification problem (success/failure).
- Define grid search parameters for each model to tune hyperparameters and apply cross-validation to prevent overfitting, thus to improve models' performance.
- Calculate proper metrics and confusion matrix to evaluate and compare models.

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1-MCC.ipynb

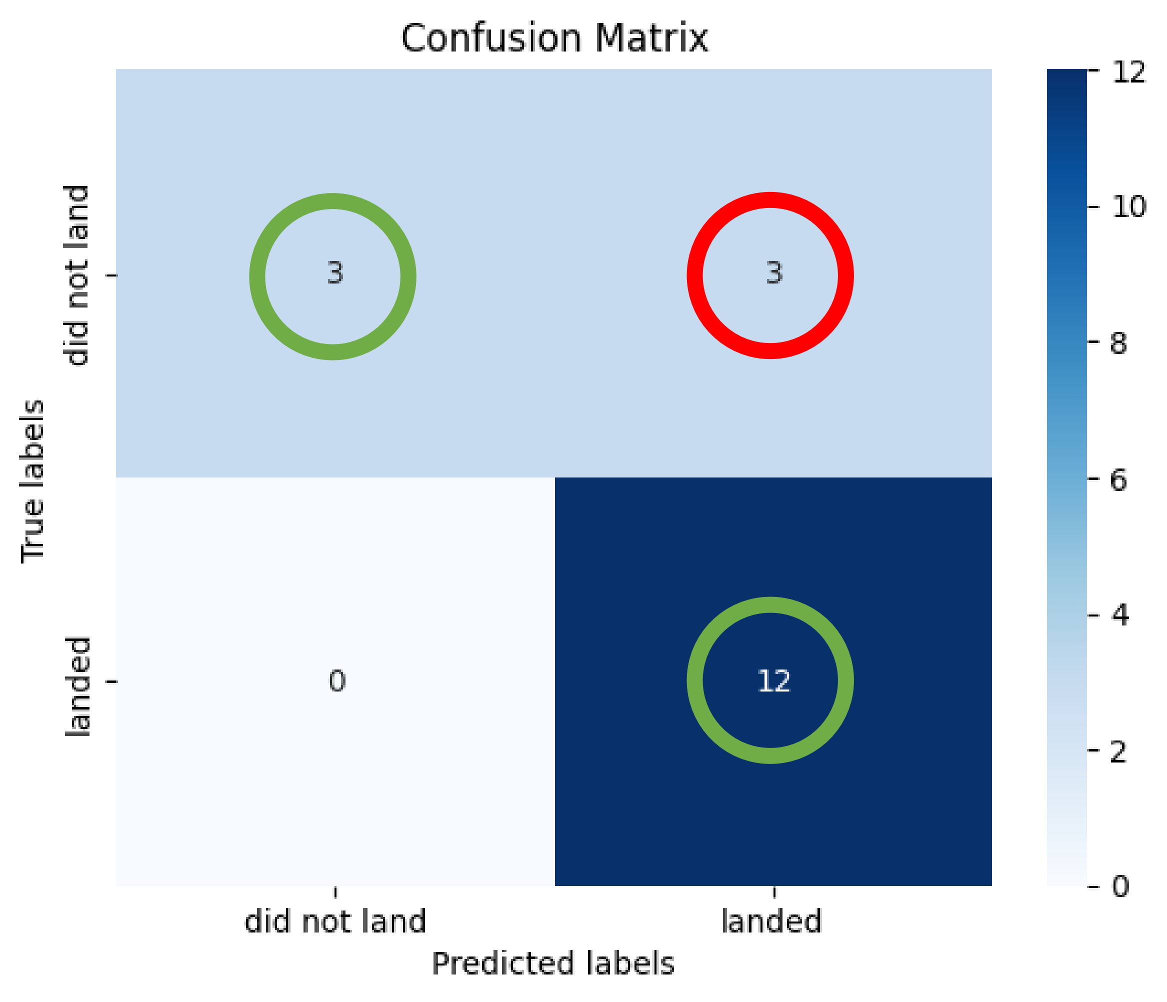
Classification Accuracy Comparison

- Same test data accuracy for all the models (0,833), due to the small size of the dataset.
- NO OVERFITTING: Test and validation accuracies are quite similar.
- NO UNDERFITTING: Both test and validation accuracies are high values (over 80%).



Confusion Matrix

- The four models gives the same accuracy value (0,833) and the same confusion matrix.
- Confusion matrix shows that the models can correctly classify True Positives and True Negatives.
- Models struggles with False Positives, i.e., prediction says **landed**, but actually it **did not land**.
- For further insights we need more test data and additional metrics like precision and recall.



Section 7

Conclusions, Insights, Recommendations and Next Steps

Conclusions and Insights

Data Scientists conclude that brand-new company SPACEY…

- ✓ Can use data analysis and machine learning models to predict future outcomes and achieve competitiveness into the space market.
- ✓ Should focus on mid range payloads to maximize booster reuse. Then, may complete capacity with smaller payloads (i.e. nanosatellites) to increase warehouse efficiency.
- ✓ Need for a launch site close to railway, airports and roads to ease goods provision and personnel commuting.
- ✓ Lauch site should be close to coastline and equator to reduce launching costs and improve safety.
- ✓ Keep on using data analysis to become a successful, reliable and profitable company.

SPACEY
Data Driven Space

Recommendations and Next steps

Data scientists recommends that SPACEY should…

- ✓ Compare data and results with other companies with significative presence in the space market (Rocket Lab, Blue Origin, Firefly, Ariane…).
- ✓ Increase dataset size for train/validation/test, maybe using cloud services to reduce processing time (Collab, Azure, AWS…)
- ✓ Find a launch site following proximity insights from previous slides.
- ✓ Design a PoC (Proof-of-Concept) to check and validate insights, assumptions and previsions. Then decide GO, NOT GO.
- ✓ Spread the "data driven" motto throughout all company departments (development, business, operation, resources…) turning it into the culture of the company.

Appendix

Find Jupiter Notebooks and Python code used for this project on GitHub:

Boosting Data Scientist Career

https://github.com/ManuC-023/SpaceY.-Data-Driven-Space

Thank you!