IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

RAGNO Tim

03/04/2024

Outline

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

Executive Summary

- **Summary of methodologies**

  The different methodologies I used during this project were : **Data Collection**, **Data Wrangling**, **Data Visualisation**, **EDA** and **Model Development** and **Evaluation**.

- **Summary of all results**

  1. **Clean, exploitable** Data from the SpaceX launches
  2. **Exploratory Data Analysis** on this Data
  3. Predictive analysis results

Introduction

- SpaceX has revolutionized **space exploration** with its innovations in cost-effective rocket technology. This **data science project** aims to analyze SpaceX launches to identify trends and insights related to launch successes, payload capacities, and the impact of reusable rocket technology, contributing to the understanding of SpaceX's influence on the future of space travel.

- Problems:
  - How can SpaceX **minimise** failures ?
  - What are the trends and correlations in Payload Size and other variables for example ?

Section 1

Methodology

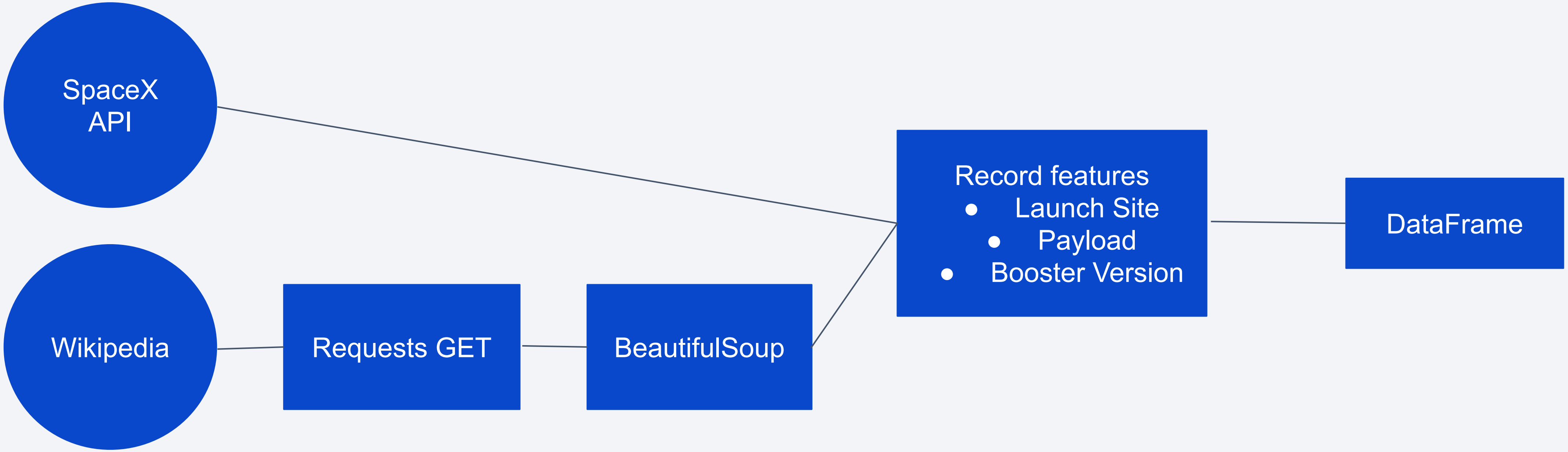
Methodology

Executive Summary

- Data collection methodology:
  - Two types of Data Sources : SpaceX API and Wikipedia
- Perform data wrangling
  - Missing values imputation, One-hot encoding, cleaning data…
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data Standardisation
  - Hyperparameter tuning and Model Evaluation

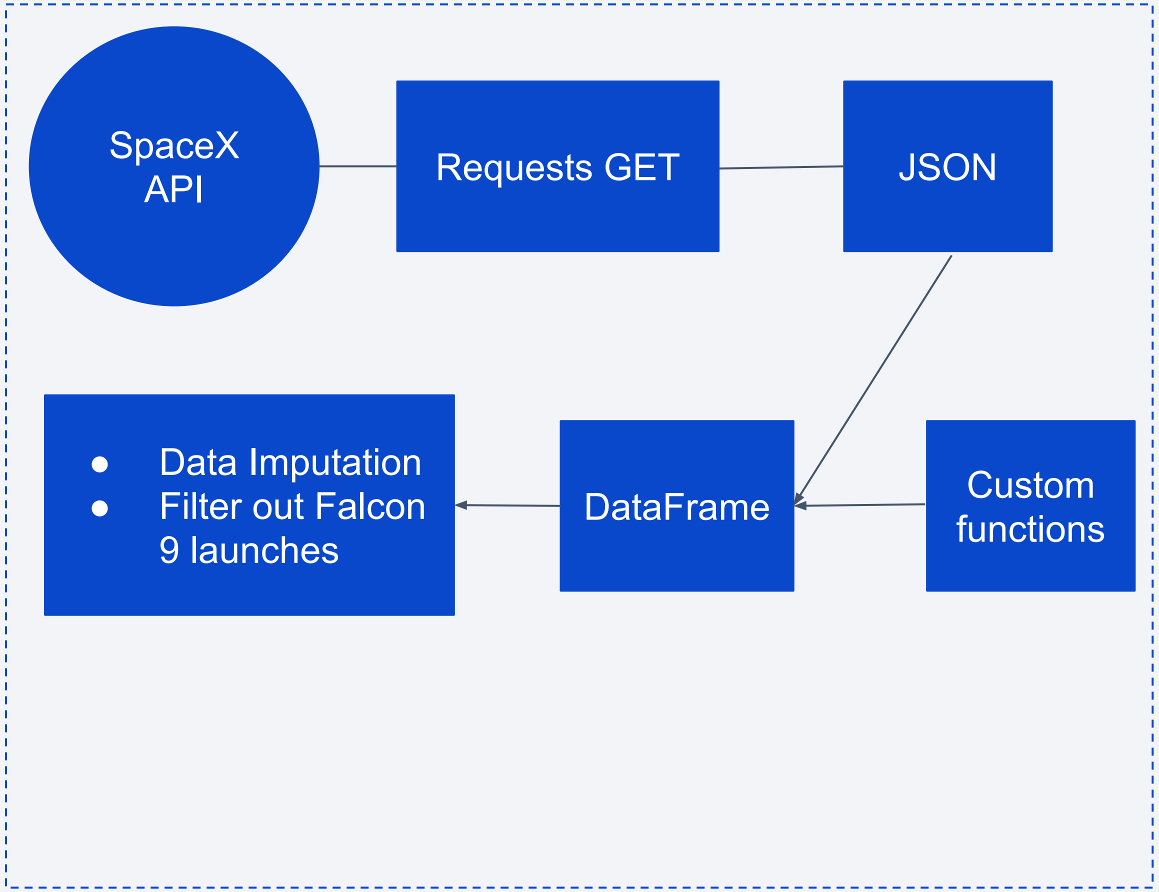
Data Collection

- The Data was collected in two ways : using the SpaceX **API** and scraping **Wikipedia**



Data Collection – SpaceX API

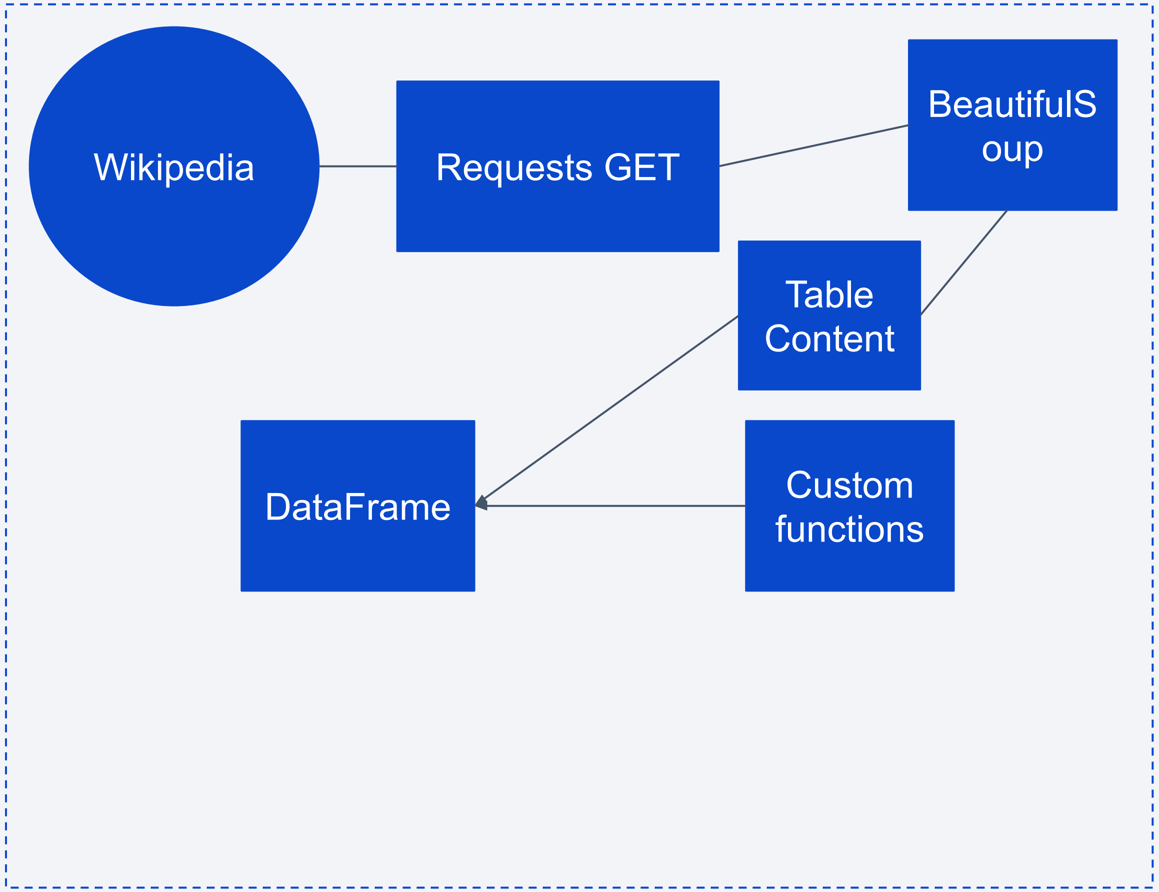
- Data collected through SpaceX's API
- JSON request -> DataFrame
- **Data Cleaning** after
- Custom functions are used to fill in the values for some features



[GitHub]

Data Collection - Scraping

- Used Requests and BeautifulSoup to extract table content
- Custom Python functions are used to extract the right info from the table



[GitHub]

Data Wrangling

- Data from the previous methods was imported into a DataFrame then modified using **Pandas** to suit different needs:
  - Creation of the **landing outcome** feature
  - Evaluation of the success rate

GitHub

EDA with Data Visualization

- Mainly using **Seaborn,** we visualised the relation between different variables:
    - Flight Number vs Payload Mass/Launch site using a **scatter plot** 
    - Success Rate and Orbit Type using a **barplot** 
    - Success Rate and Year using a **line plot** 

[Github]()

EDA with SQL

- Used Magic Functions to use SQL queries in the Jupyter Notebook
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - …

GitHub

Build an Interactive Map with Folium

- Create various Folium objects to add to a map of the SpaceX launches
  - **Circles** and **Markers** to identify the launch sites
  - Color on the Marker to identify if they were successful or not
  - A **Marker Cluster** to map all the the launches to a single point at lower zoom (avoid clutter)
  - **Polylines** to link the launch sites to the nearest landmarks (sea, trains…)

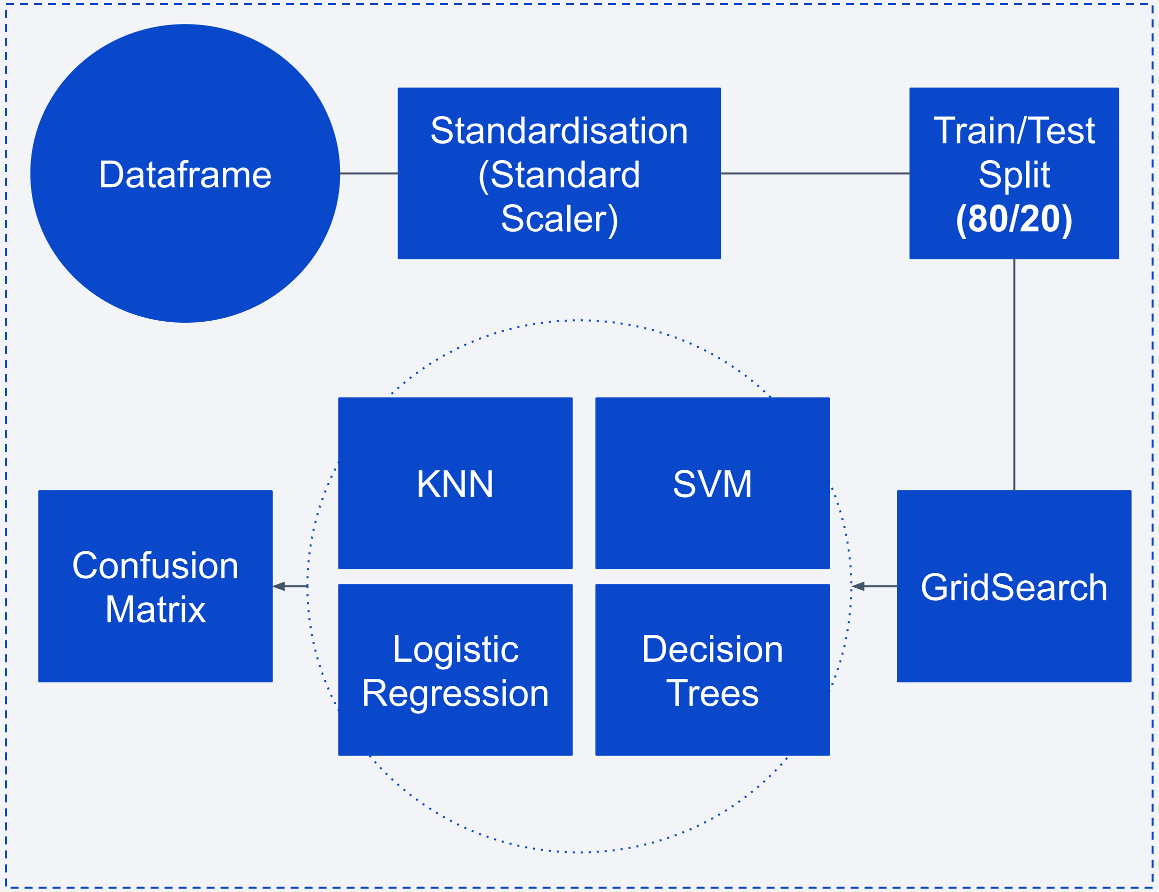
GitHub

Build a Dashboard with Plotly Dash

- Unfortunately, due to an issue with the lab link, I was unable to do this one :(. I will thus have no GitHub and no link to a notebook for the Dashboard creation. I looked forward to this one so I am a bit sad.

Predictive Analysis (Classification)

- **Preprocessed** the Data
- **Split** into Train/Test
- Fitted a **GridSearch** on 4 different models for classification
- Evaluated them using .score and **Confusion matrices**



GitHub

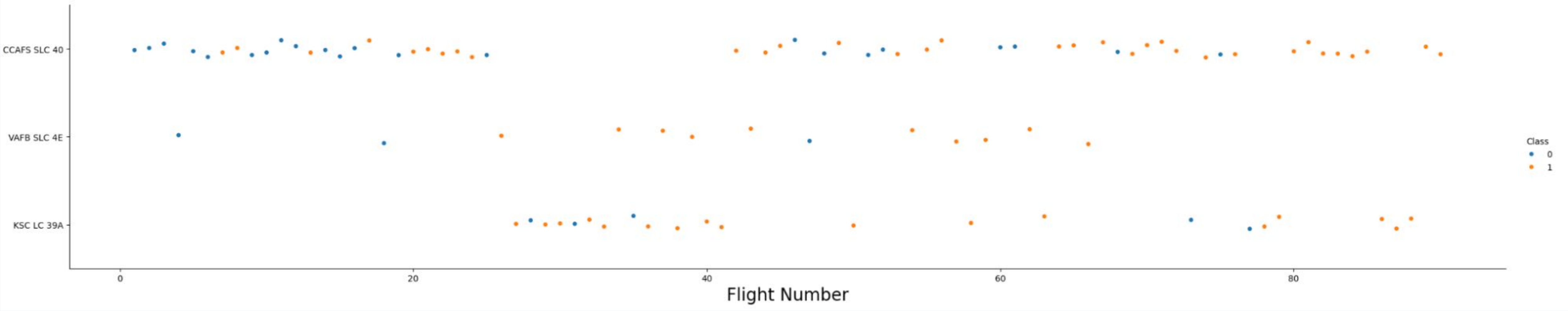
Results

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

Section 2

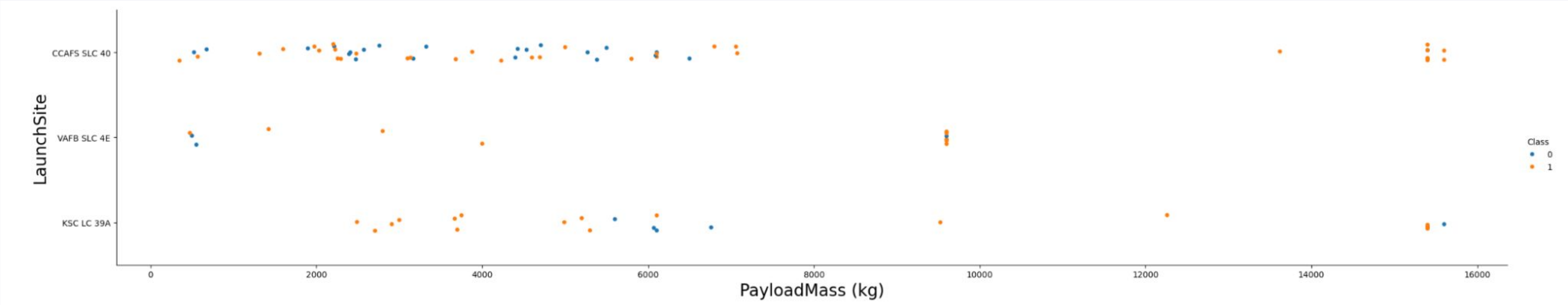
Insights drawn from EDA

Flight Number vs. Launch Site



- We can see that the success rate seems to **go up over time**
- VAFB SLC 4E sees very few launches but seems to get them right very often
- Recent launches (in this graph) have been stellar

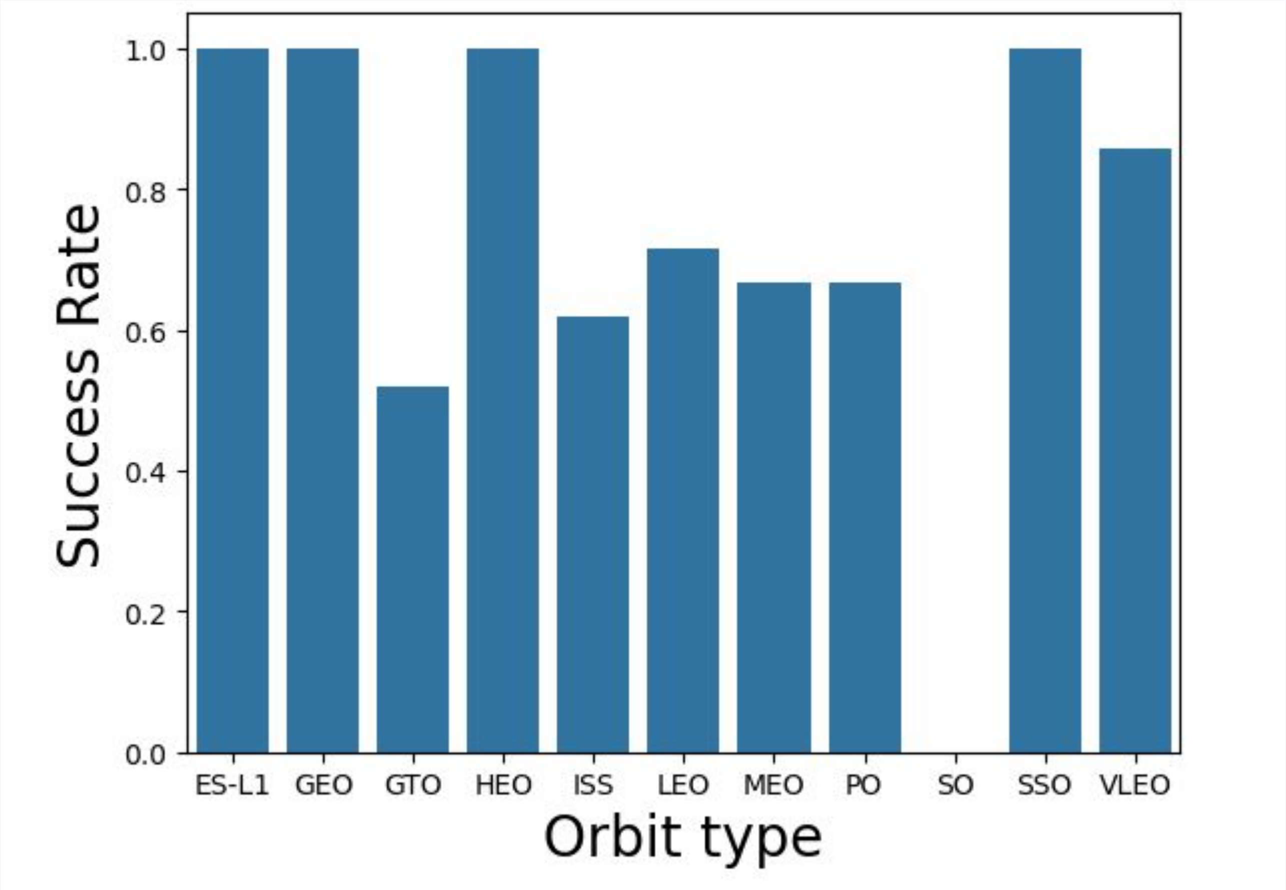
Payload vs. Launch Site



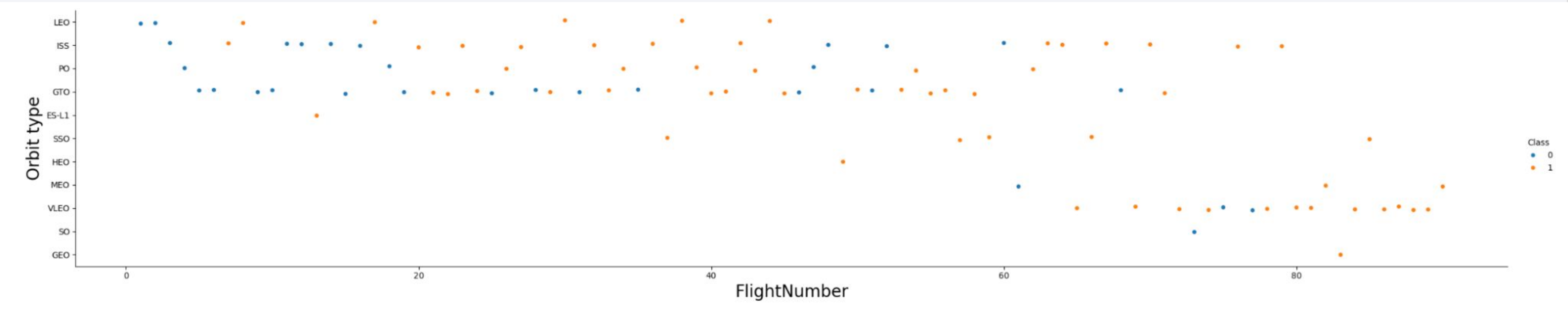
- **High** Payload mass is very rare but seems to get good results on all launch sites
- CCAFS SLC 40 seems to struggle the most all across the board (with **no relation** with Payload mass)

Success Rate vs. Orbit Type

- **GTO** has the lowest SR
- **Max** SR is split between 4 Orbit types (have to see the value counts to really compare the 4)

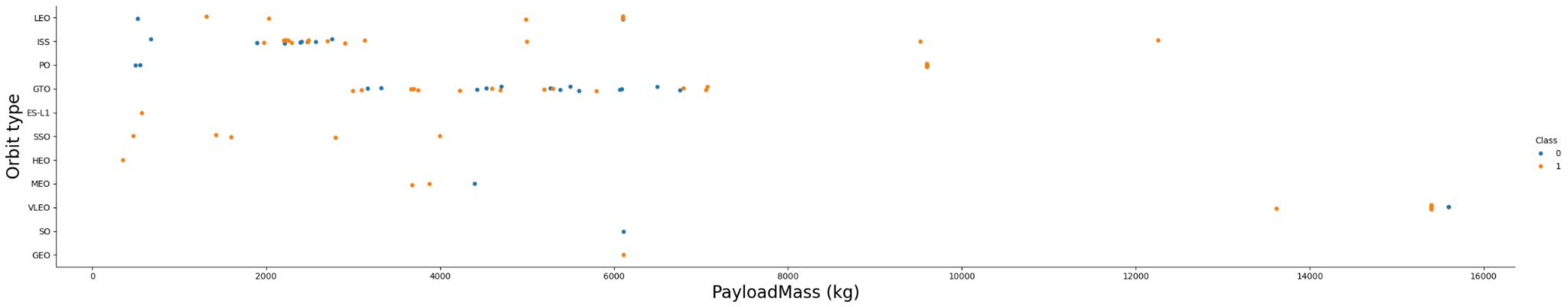


Flight Number vs. Orbit Type



- **VLEO** seems to have been a recent admission that eclipsed some of the earlier types
- **SSO** and **HEO** see very sporadic use but are always a success
- Hard to establish any global correlation

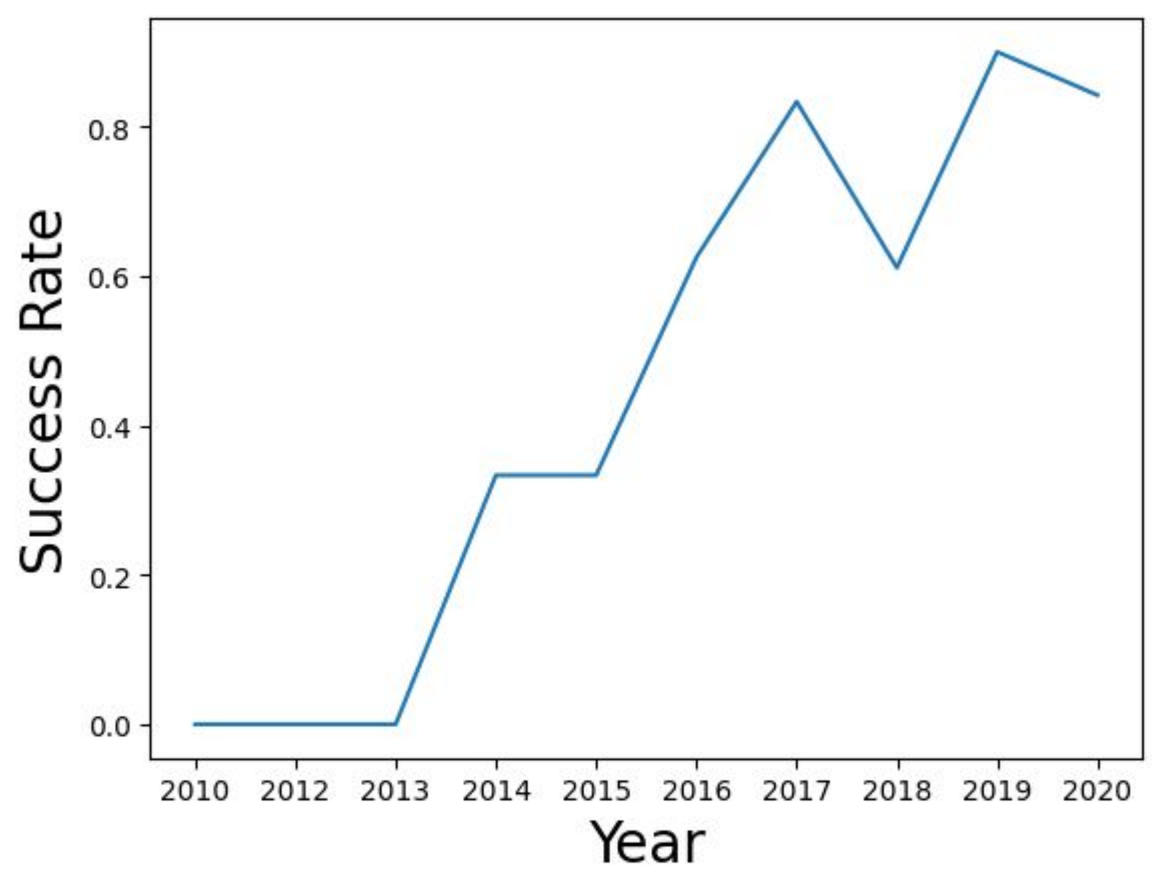
Payload vs. Orbit Type



- **GTO** seems to only carry missions with very specific payload size (between 3000 and 7000 approx)
- **VLEO** has only high payload size while **SSO** has only low one
- Lower payload doesn’t necessarily equate to better success for the statistically significant orbit types

Launch Success Yearly Trend

- Success Rate only goes up (except for 2018 and marginally for 2020)
- **2014** was a famous breakthrough with the first successes in SpaceX's history



All Launch Site Names

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

- Simple SELECT statement

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE \
WHERE Launch_Site LIKE 'CCA%' \
LIMIT 5
```

- The wildcard ‘%’ serves to select all launch sites beginning with something
- LIMIT 5 to only print out the first 5 results

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

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE\
WHERE Customer = 'NASA (CRS)'
```

- We select the aggregate SUM from the Payload Mass column
- Context on the number of NASA (CRS) launches would be needed for this number to make sense

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE \
WHERE Booster_Version = 'F9 v1.1'
```

- Same as before with **AVG**

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTABLE \
  WHERE Landing_Outcome = 'Success (ground pad)'
```

- Same with MIN (works on Date formats)
- This is the very famous ground pad success in 2015 that essentially re-put SpaceX on the map

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE \

WHERE Landing_Outcome = 'Success (drone ship)' AND
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

- BETWEEN is easier to write than writing out the two conditions
- AND to bind multiple conditions

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 Mission_Outcome, COUNT(*) FROM SPACEXTABLE \
GROUP BY Mission_Outcome
```

- GROUP BY to select by the different Mission Outcomes
- Success is very common in this dataset

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version FROM SPACEXTABLE \
        WHERE PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

- A Subquery is used since Max cannot be used outside the select statement for this task

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
%sql SELECT  substr(Date, 6,2) AS Month, * FROM SPACEXTABLE \
WHERE substr(Date,0,5) = '2015' AND Landing_Outcome = 'Failure (drone ship)' \
AND  substr(Date, 6,2) IS NOT NULL \
ORDER BY Date
```

- The functions used in the question are used to retrieve the months and year in MySQL
- ORDER BY is ascending by default

Month	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
01	2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
04	2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

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

```
%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) from SPACEXTBL \
WHERE DATE between '2010-06-04' and '2017-03-20' \
GROUP BY LANDING__OUTCOME \
ORDER BY LANDING_COUNT DESC;
```

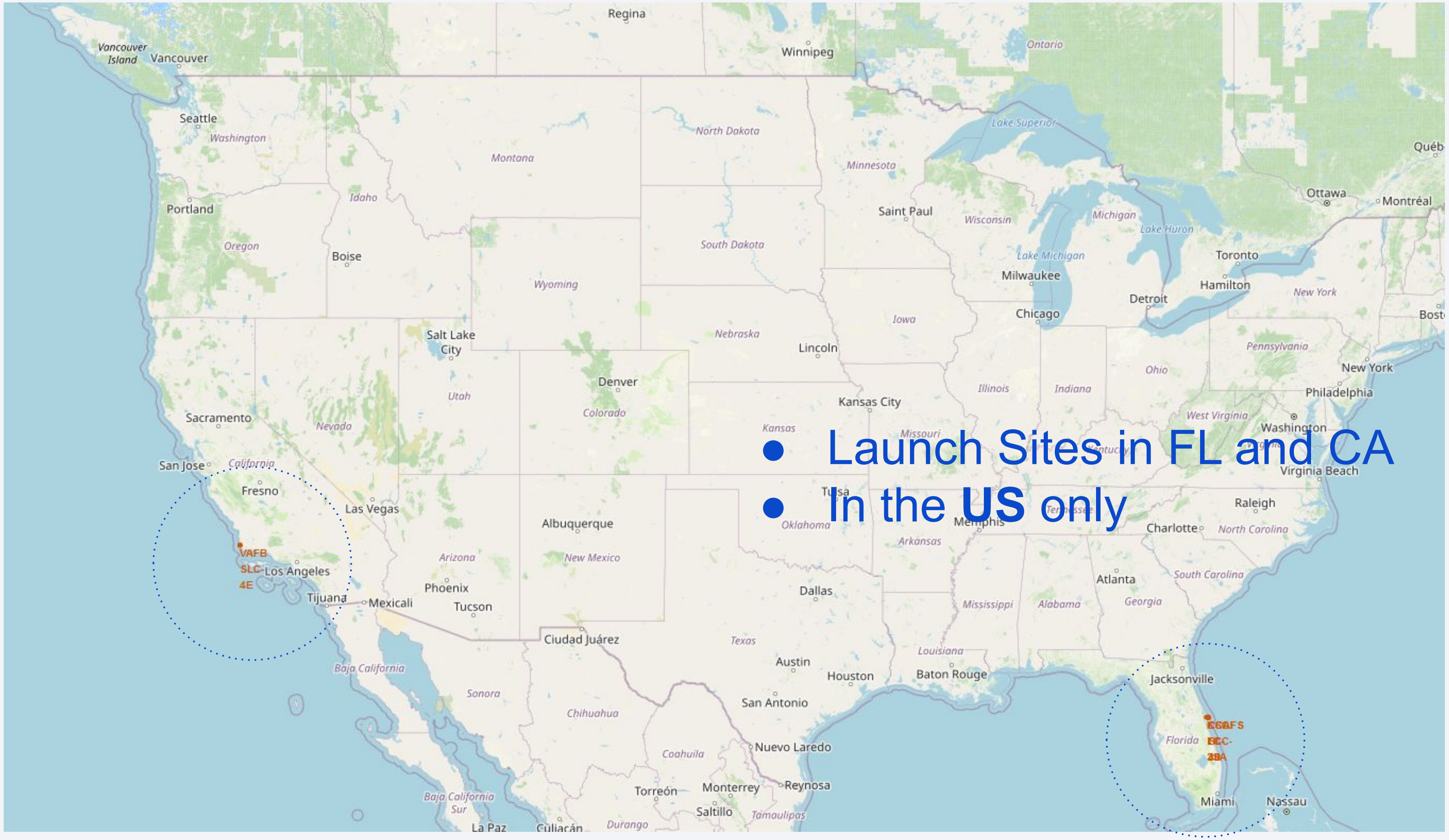
- BETWEEN can be used for dates

Landing_Outcome	Count
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

Launch Sites

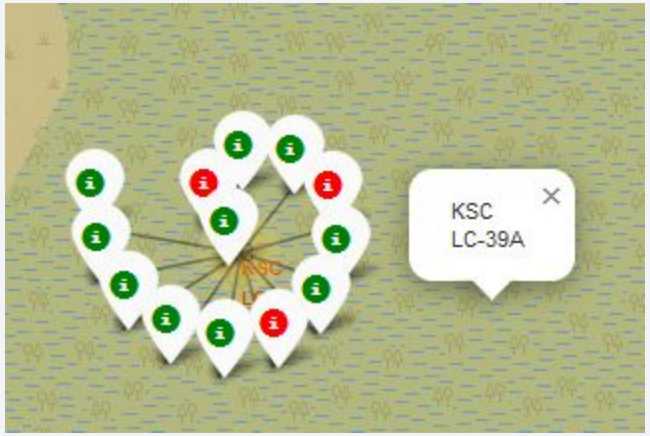


- Launch Sites in FL and CA
- In the **US** only

Successes and Failures

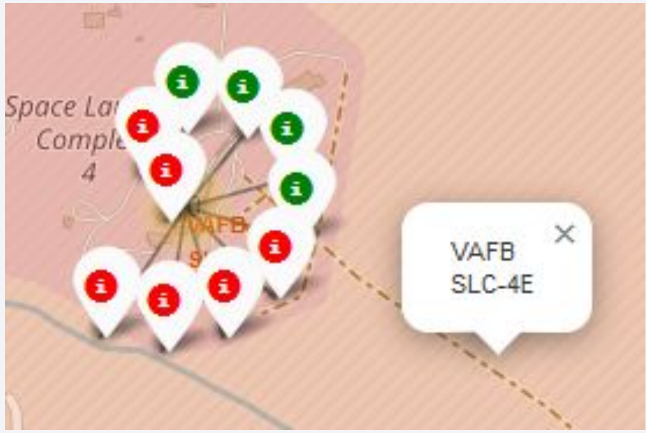
California launch site

Mostly **successes**



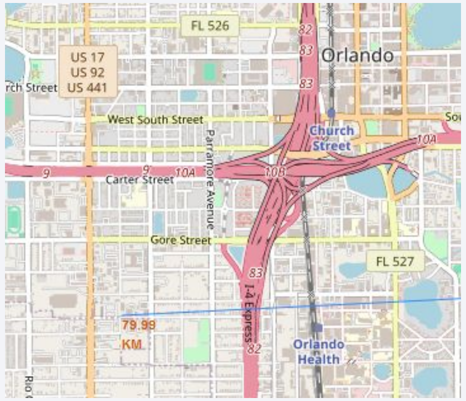
Florida launch site

Mostly **failures**



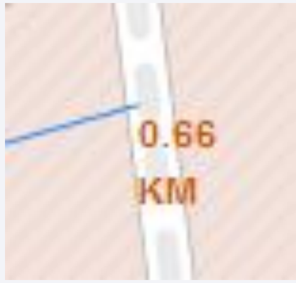
Proximity from Launch Sites

- This is all taken from the CCAFS - LC 40 in California
- In general, all launch sites are **very close to the sea** and to a means of transport (highway and/or railway)



City (Orlando)

Coastline



Highway

Railway

Section 4

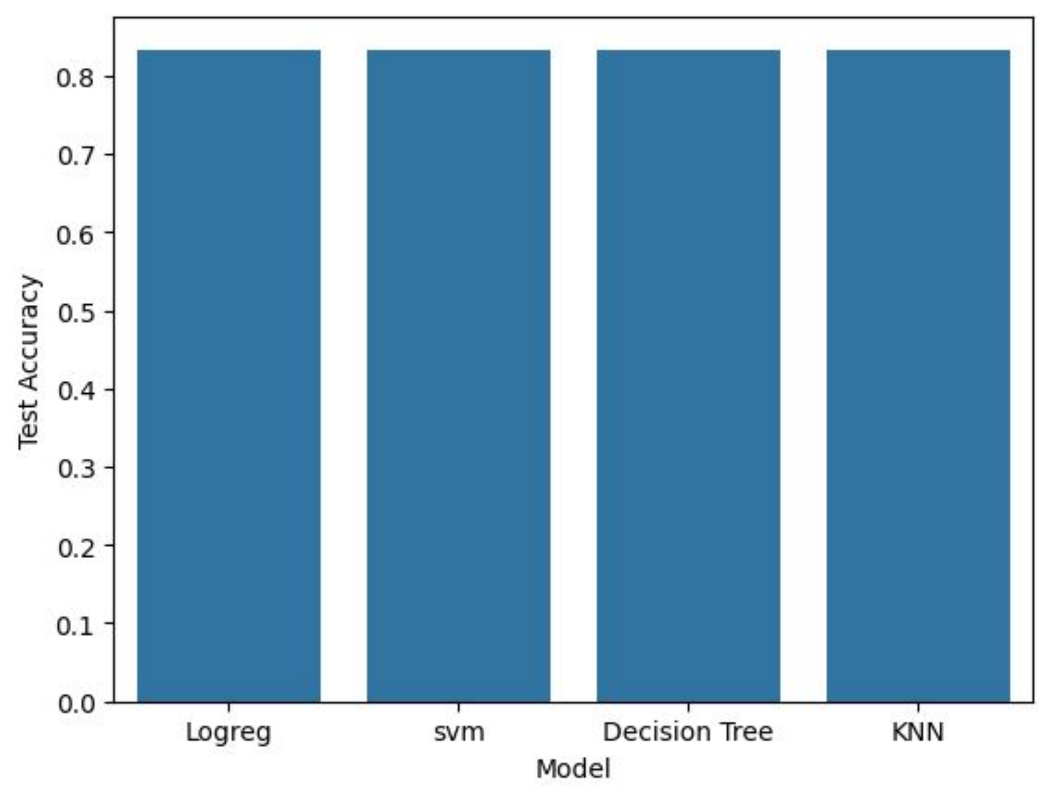
Build a Dashboard with Plotly Dash

Section 5

Predictive Analysis (Classification)

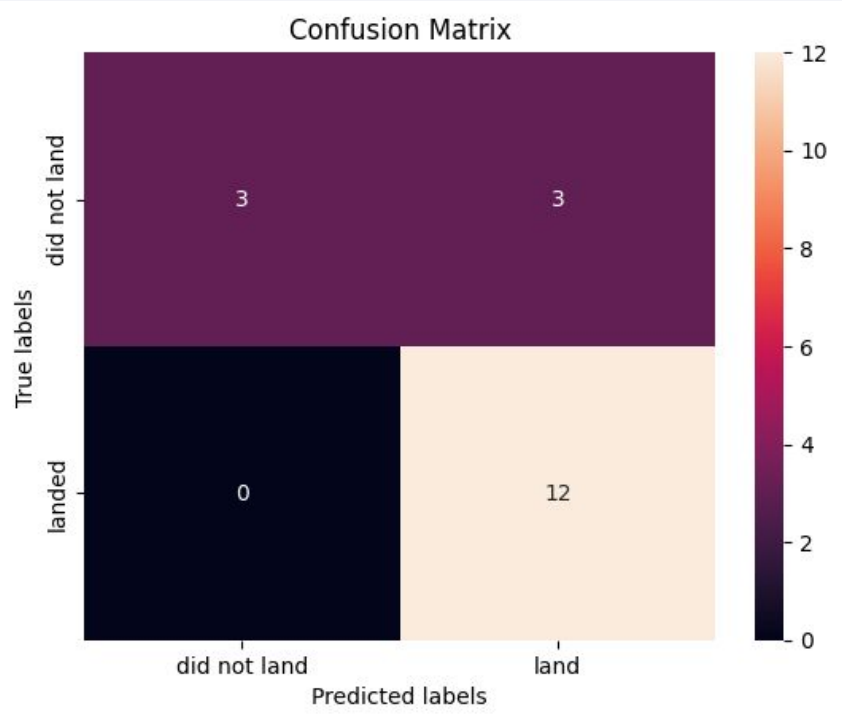
Classification Accuracy

- With the deprecation of the ‘auto’ parameter of Decision Tree, **all 4 models** now have the same Test Accuracy (and Confusion Matrix…) on the test set
- The Test set is **quite small** so this could’ve been expected



Confusion Matrix

- This is the confusion matrix of all 4 models we have examined. We can see that the models only got 3 predictions wrong (False Positives or false alarm, the model seem ‘overconfident’ in a way.



Conclusions

- From the EDA, we can see that important factors that could help the model in predicting success are the **Launch sites**, the **size of the Payload** and the **Orbit Type**. Time is also very much correlated with success, but this is not a real material, quantifiable factor.
    - A more thorough Explainable AI study would help in understanding the models' decision

- It's always necessary to take a step back when analysis ML results.
    - Here, **data** is somewhat lacking, 18 test samples is too little to make an earnest analysis on the correct model to choose. This is due to the complex nature of the problem.
    - In real life situations, we would most likely test other models (given a more ample test set). This would help us discern between the models base on confusion matrices, ROC curves, AUC…

Thank you!