IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

José Juan Martínez Tapia
28/mar/2024

Outline

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

* huge thanks to IBM for the help in Skills Networks labs!

Executive Summary

This project was divided into the following stages:

1. **Data Collection**
   - SpaceX REST API calls
   - BeautifulSoup Webscraping
   - SpaceX rocket launch raw dataset collecting
2. **Data Wrangling** & 3. **EDA**
   - pandas
   - NumPy
   - SQLite
   - Python and JupyterLab Data cleaning and analysis. Correlations, statistics and insights gathering.
4. **Visualization & Interactivity**
   - seaborn
   - matplotlib
   - Folium
   - Interactive & dynamic data visualizations
   - plotly | Dash
   - Web-based dashboard application
   - Geographic, statistical visualizations, and dinamic/interactive dashboard web application
5. **Machine Learning**
   - scikit learn
   - Grid Search, Logistic Regression, SVM, Decision Tree, KNN testing, tuning and validating
   - Model grid search, developing, validating and data prediction

Introduction

Allon Mask
Founder & CEO of SpaceY

At SpaceY, we're committed to revolutionizing the spatial industry by democratizing access to space through cost-effective solutions in rocket ship acquisitions and rentals.

Our core mission hinges on the innovative recovery and reuse of spatial infrastructure, allowing us to offer affordable space access globally.

This report, along with its additional materials and documentation, aims to showcase and present the results of the Machine-Learning-based prediction model and the data collection, cleaning, analysis, and preparing methodology, offering useful insights in the data with web-based dashboard applications, interactive and dynamic visualizations, and geographic data to better understand what our model is evaluating.

Section 1

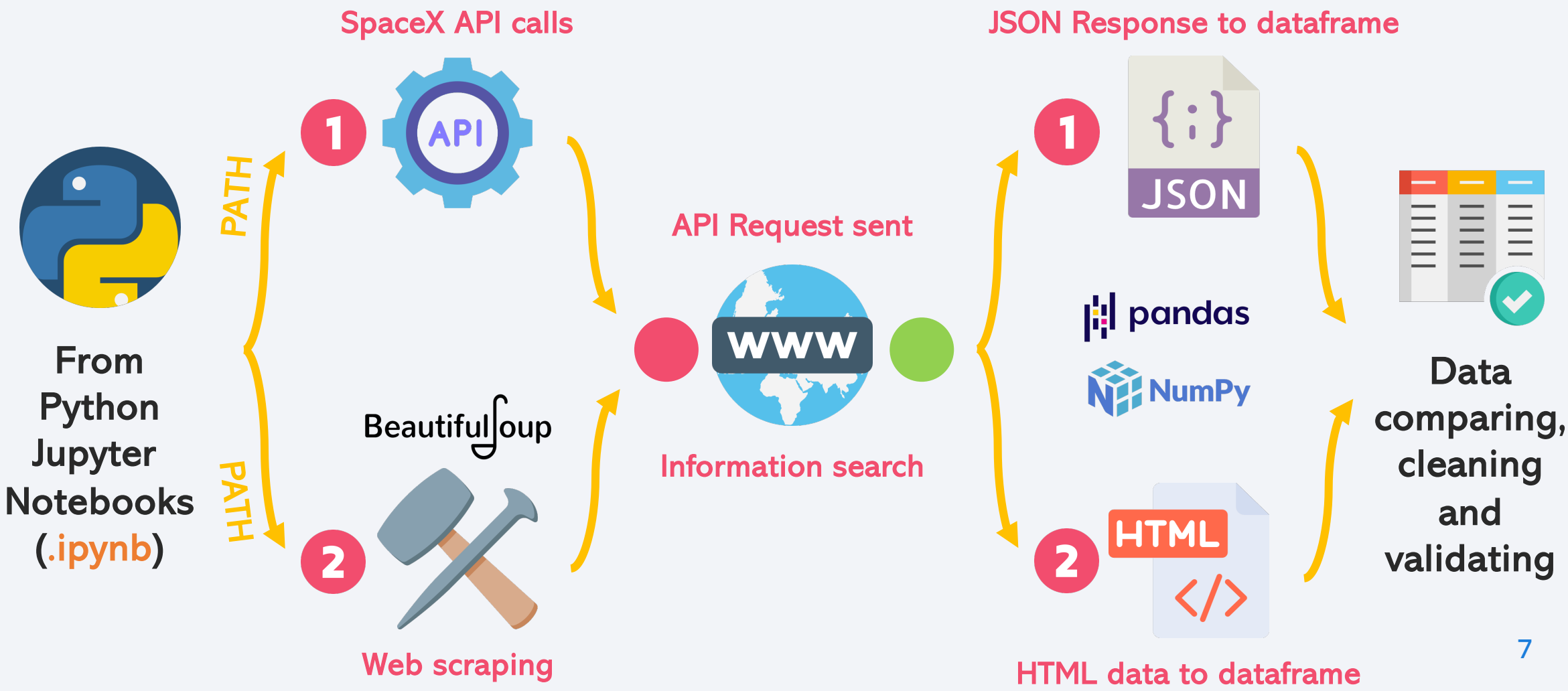
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from the SpaceX official REST API, and by scraping the internet for further information.
- Perform data wrangling
  - Identifying and handling missing data, and creating our target categorical variable (y).
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Performed Grid Search to find the best parameters for each one of our models, which were Logistic Regression, Support Vector Machine (SVM), and more.

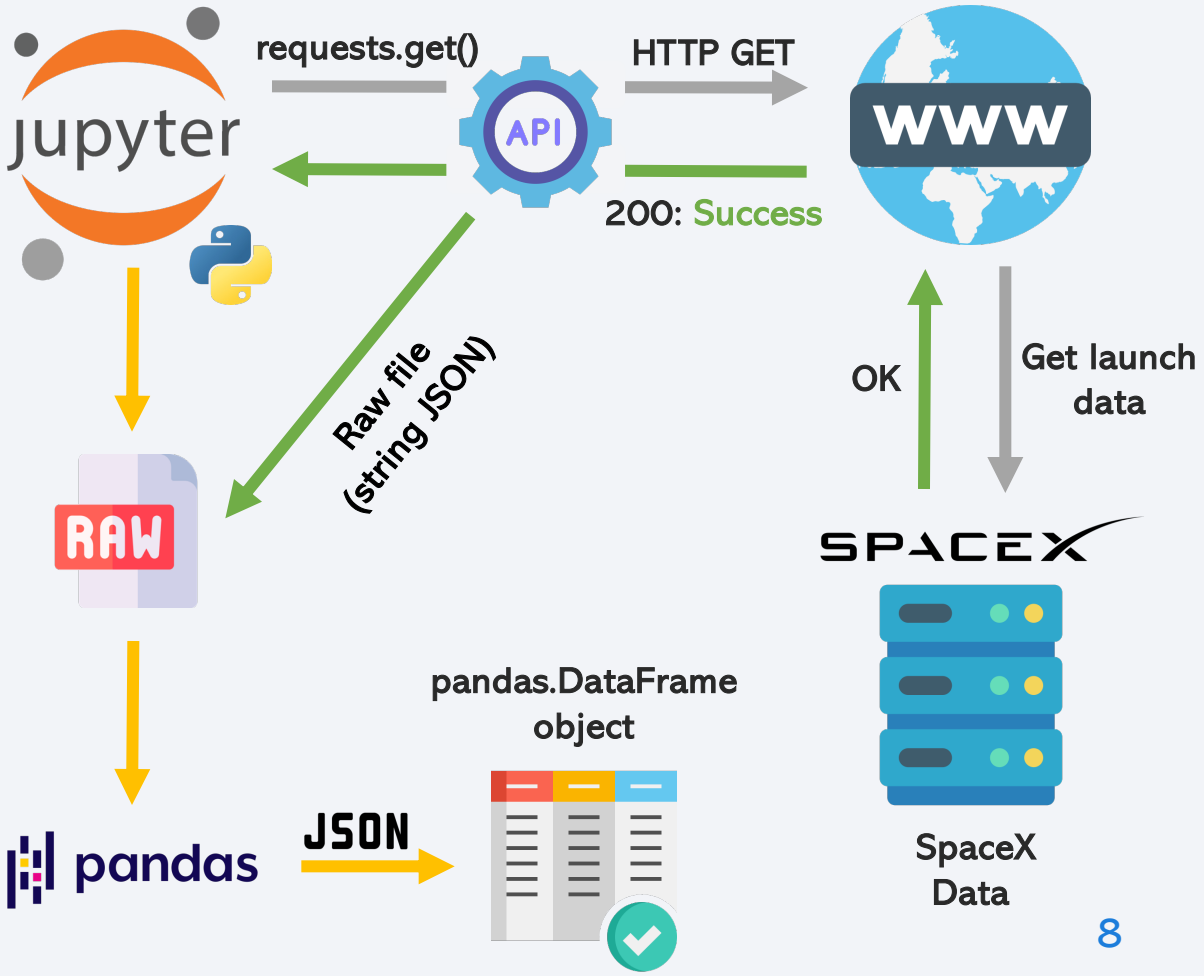
Data Collection



Data Collection – SpaceX API

1. With Python's "**requests**" module, call the SpaceX REST API (https://api.spacexdata.com/v4/rockets/)
2. Retrieve and load the JSON string response into Jupyter Notebooks.
3. Normalize the JSON string response into a pure JSON object so you can read it with "**pandas**" module into a DataFrame

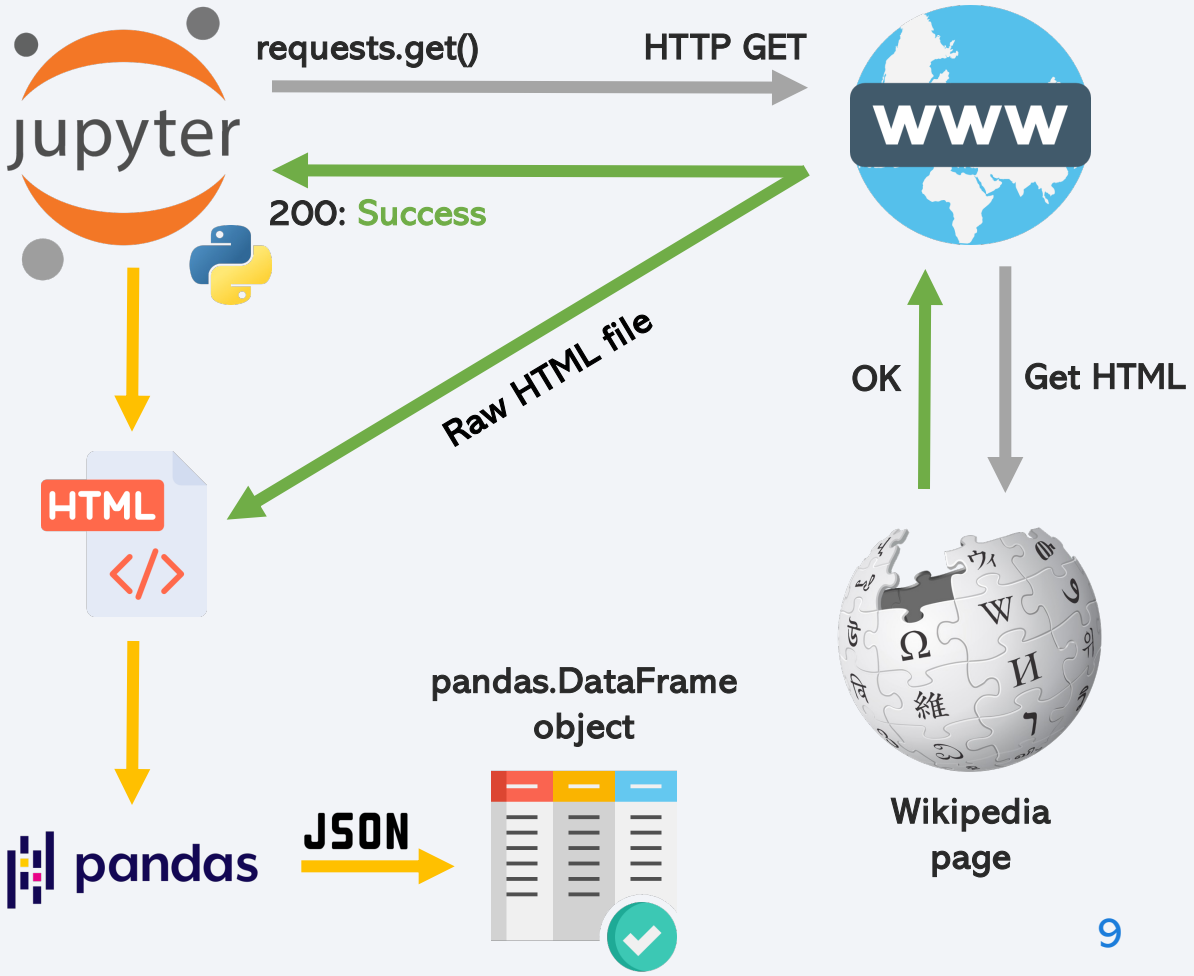
You can access the original Jupyter Notebooks file in GitHub by clicking the following link: jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

1. With Python's "**requests**" module, call an IP, or internet, address with available information (i.e. Wikipedia) on SpaceX rocket launches.
2. Retrieve and load the HTML information of the web page and parse it into a "**BeautifulSoup**" object.
3. Search for the launch information (i.e. <table> tag) and build a "**pandas**" DataFrame from it.

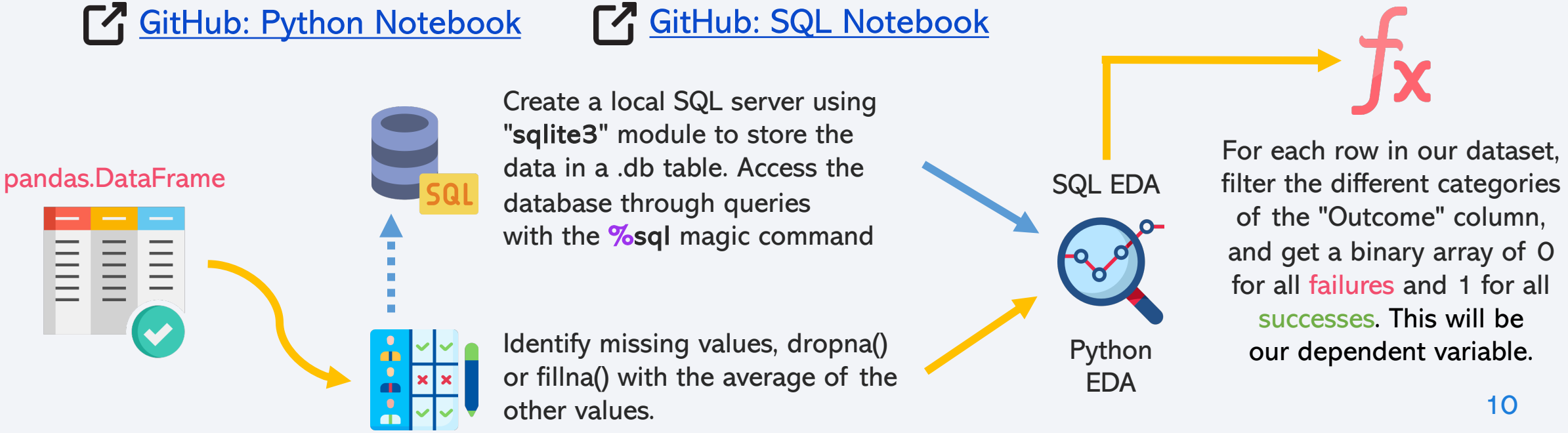
You can access the original Jupyter Notebooks file in GitHub by clicking the following link: [jupyter-labs-webscraping.ipynb]



Data Wrangling

- **SQL & Python:** Analyzed the structure of the data, its datatypes, trends and other useful insights for later on.
- **SQL:** Explored how the data was presented and how we could transform it for our model.
- **Python:** Fixed missing values, imputing them with the mean of the column if possible, or dropping them. Generated our dependent variable "class" using the data from the "Outcome" column.

GitHub: Python Notebook GitHub: SQL Notebook

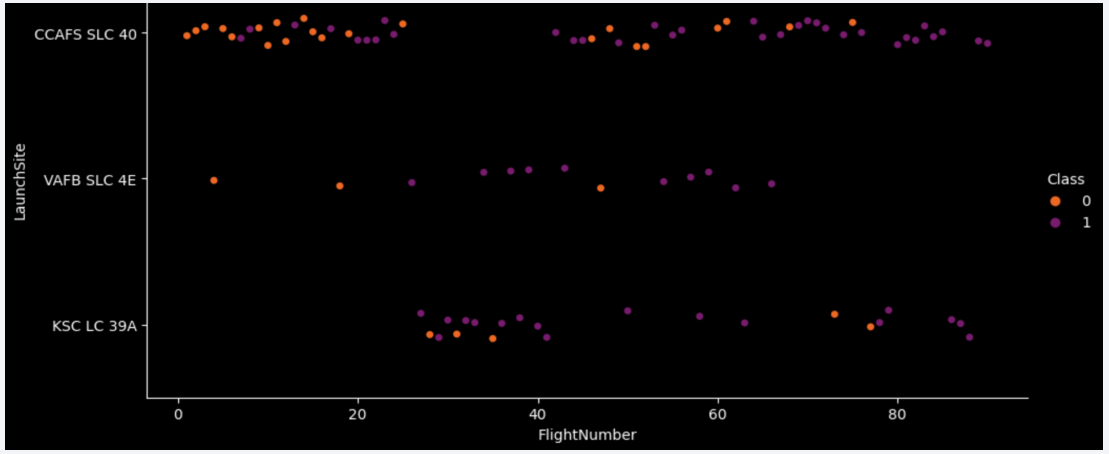


EDA with Data Visualization

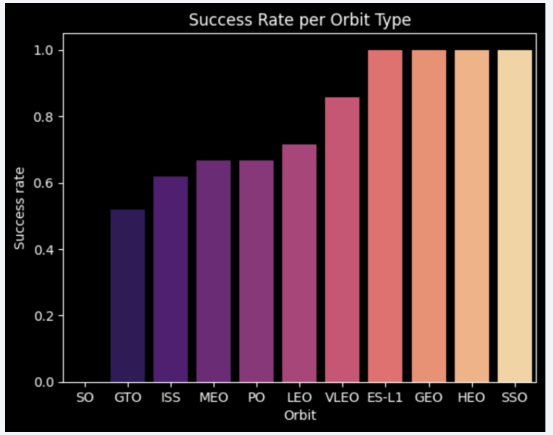
Primarily, our focus was on displaying how the payload mass, flight number and year correlated to the orbit types and launch sites for each rocket launch. The success rate, or the target variable, was visualized each time to see how its distribution changed.

We mainly used category plots due to our target variable being categorical, but scatter, bar and line plots were used as well to identify trends, such as the launch success rate over time or the success rates per orbit type.

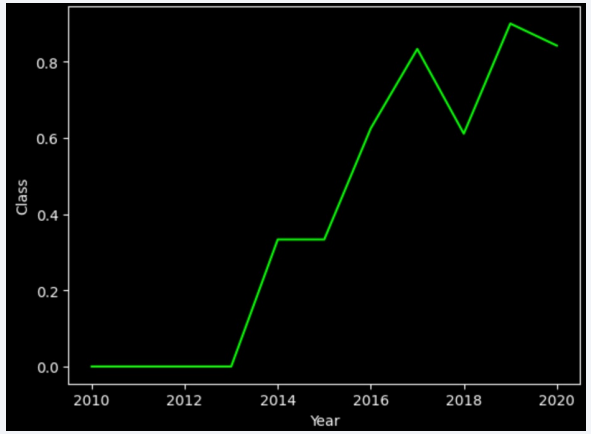
Launch outcomes grouped by launch site & number



Success rate per orbit type



Success rate over time



GitHub: jupyter-labs-eda-dataviz.ipynb

EDA with SQL

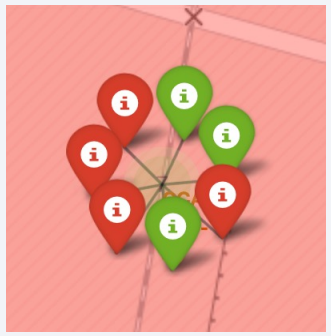
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with "CCA"
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- Get the date when the first successful landing outcome in a ground pad was achieved.
- Get the total number of "success" and "failure" mission outcomes
- With a subquery, list the names of the booster versions which have carried the maximum payload mass
- List the records which will display the month names, failure landing outcomes in a drone ship, booster versions, and launch site for the months in the 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.

GitHub: jupyter-labs-eda-sql-coursera_sqllite.ipynb

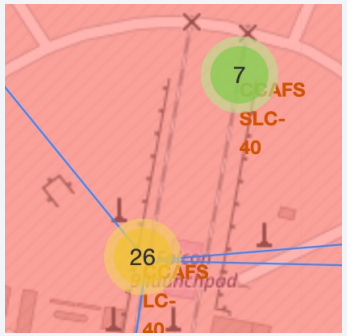
Build an Interactive Map with Folium

Added some elements to our map to better understand and visualize our geographical data:

- **Markers**: Display individual points on the map, such as successful (green), failed (red) rocket launches, and other landmarks such as nearby cities, shores and highways (blue)
- **Marker clusters**: When the map is zoomed out, to be able to easily identify the numberof records in a given geographical site.
- **Circles**: To locate each one of the landing sites of the SpaceX data.
- **Lines**: Trace the nearest path and distance to the nearest landmark markers (blue)

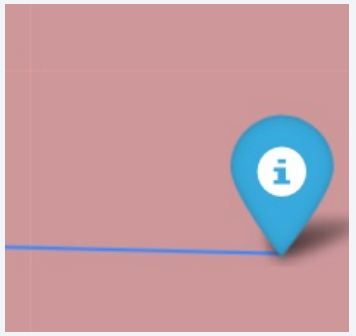


Marker



Marker Cluster

Circle



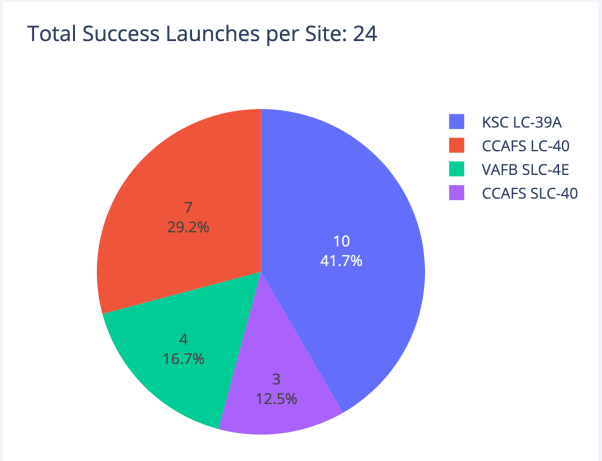
Line

GitHub: lab_jupyter_launch_site_location.ipynb

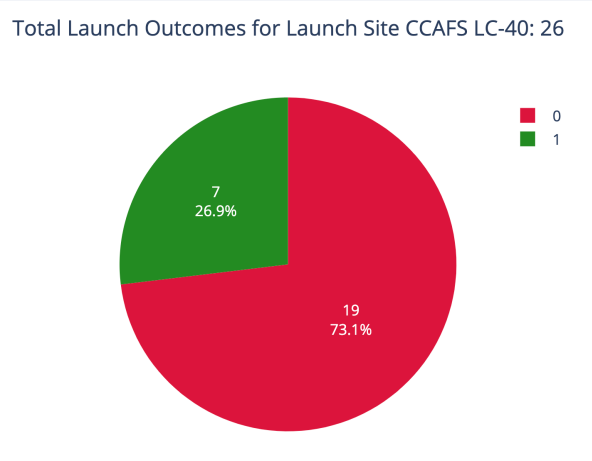
Build a Dashboard with Plotly Dash

The dashboard for the SpaceX launch data includes:

- **Launch Site Dropdown**: Allows users to filter the displayed data by launch site, with an option to view all sites or individual ones.
- **Pie Chart:** Shows the success rate of launches for either all sites[1] or a selected site[2], providing immediate visual feedback on performance.
- **Payload Range Slider**: Enables users to select a range of payload masses, offering insights into how payload size influences launch outcomes.
- **Scatter Plot**: Correlates payload mass with launch success, highlighting the effectiveness of different booster versions.



1: Total success launches per site



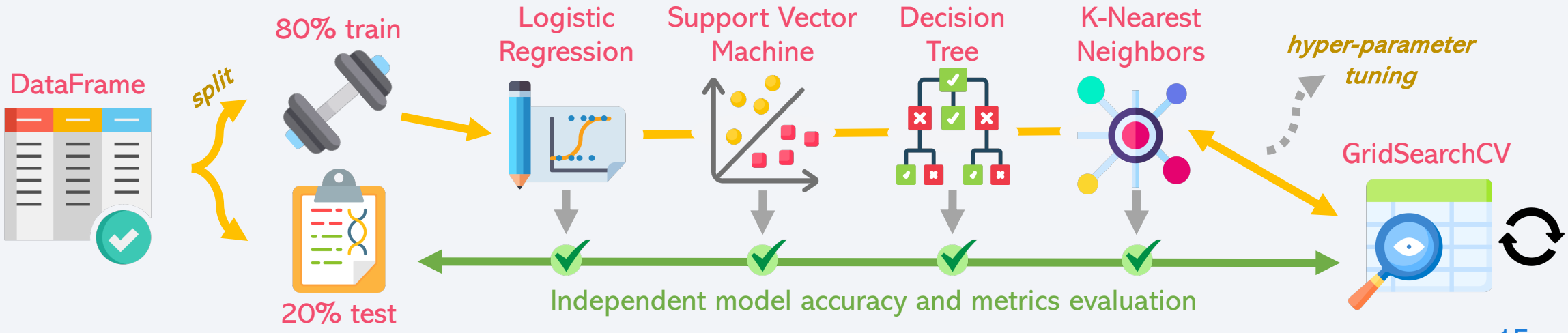
2: Outcomes for launch site

GitHub: dash_interactivity.ipynb

Predictive Analysis (Classification)

To build and select the best-performing classification model, the process included:

- **Data split**: The data was split into 80% training and 20% testing datasets.
- **Model Selection & Training**: Various classification algorithms (Logistic Regression, SVM, Decision Tree, KNN) were trained using GridSearchCV to tune hyperparameters.
- **Evaluation**: The models were evaluated on the test set, and their accuracy scores were stored.
- **Best Model Identification**: The model with the highest accuracy score was identified as the best performer.



GitHub: SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Exploratory Data Analysis Results:

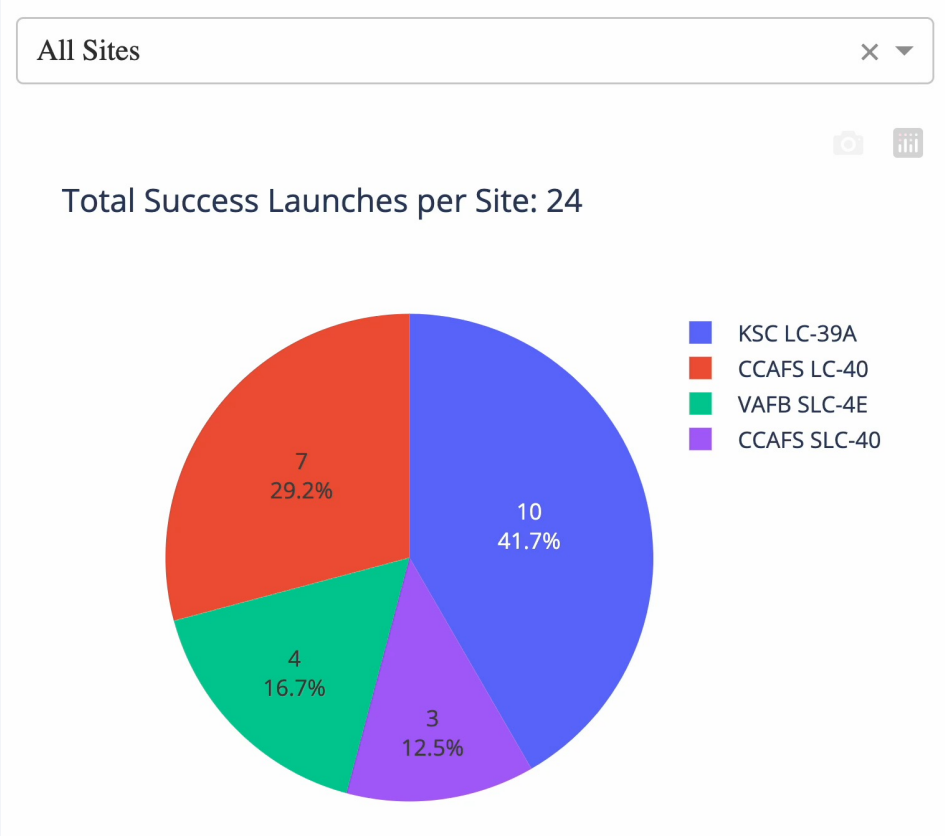
- **Visualized** the relationships between variables such as payload mass, flight number, year, and launch site, **understanding** how these factors correlate with launch success rates.
- **Analyzed** data through various types of plots, such as category plots, scatter and line plots; revealing **trends** and **distributions**, such as the variance in success rate by orbit type and its evolution over time.

Predictive analysis results

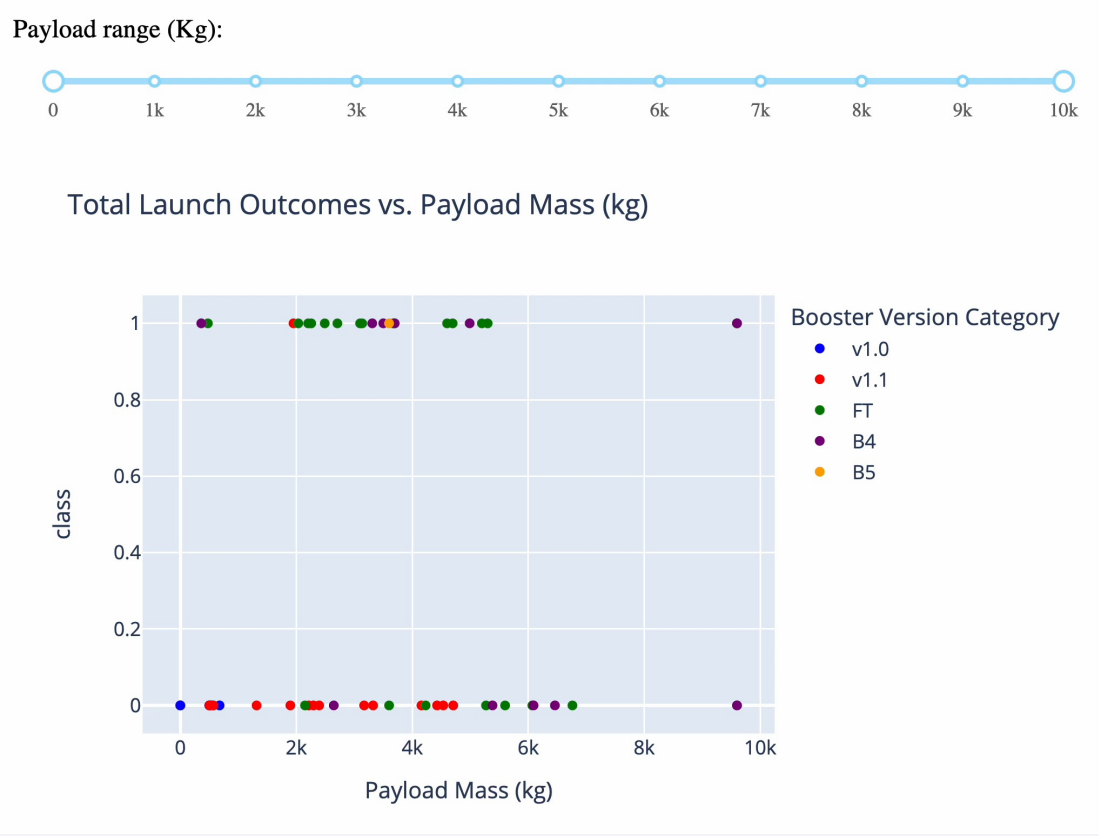
- **Trained** several **classification models** (Logistic Regression, SVM, Decision Tree, and KNN) with an 80/20 train-test split.
- **Hyperparameter tuning** was performed using **GridSearchCV** to optimize the models.
- The models' performances were compared based on accuracy, and the best model was identified as the one with the **highest accuracy** on the test set.

Results

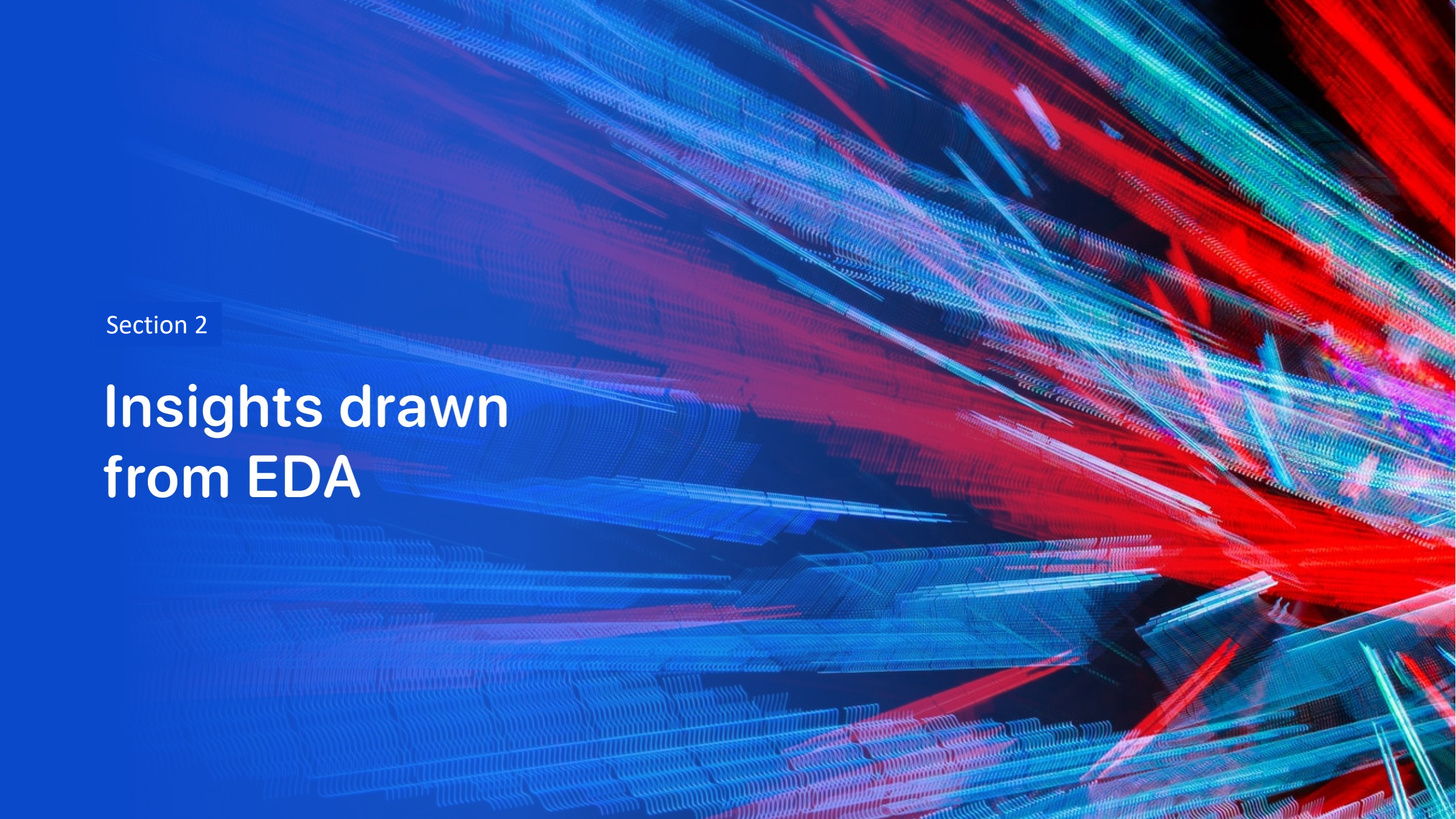
Interactive Analytics Demo



Get success % for each or all launch sites

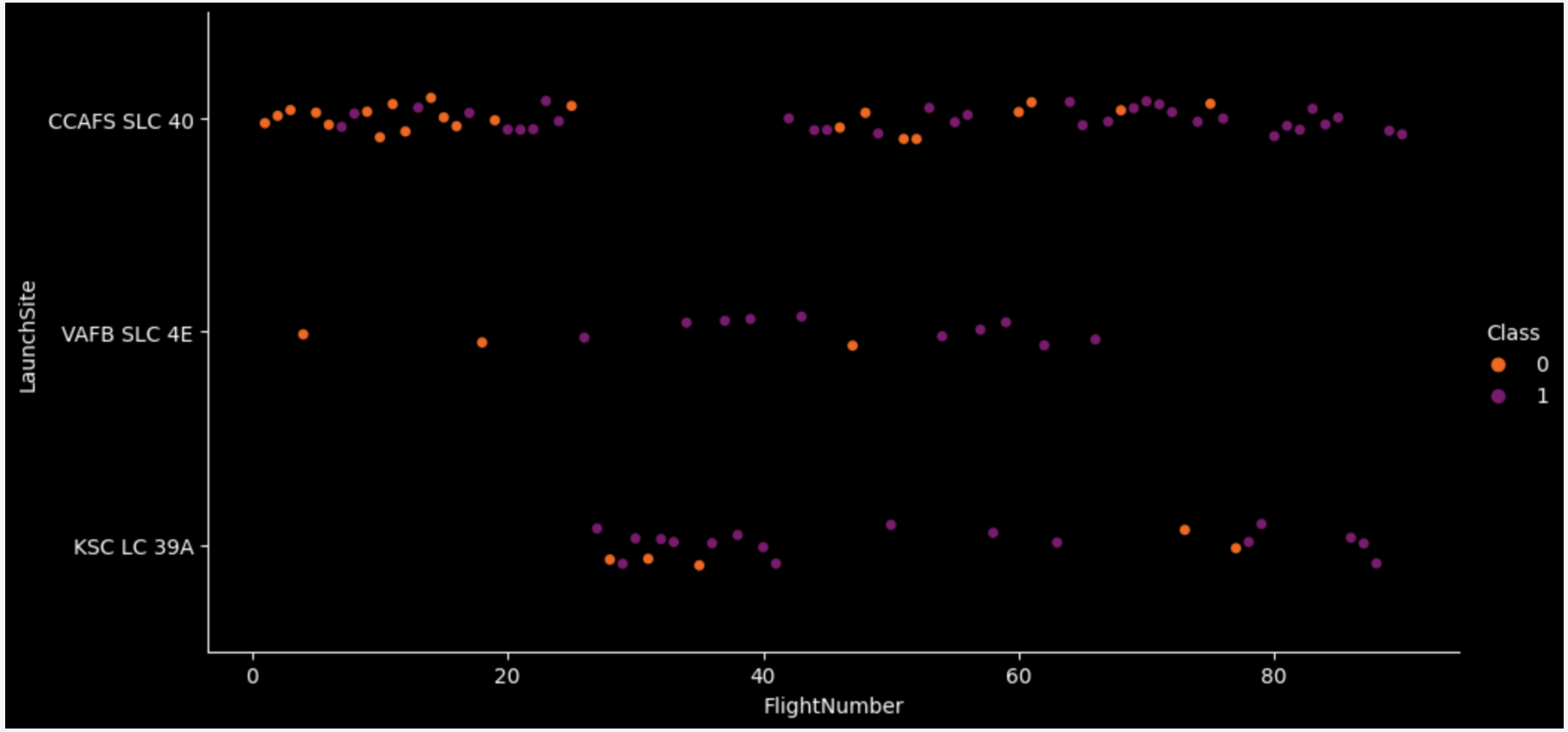


Filter by launch site and payload mass in real time


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

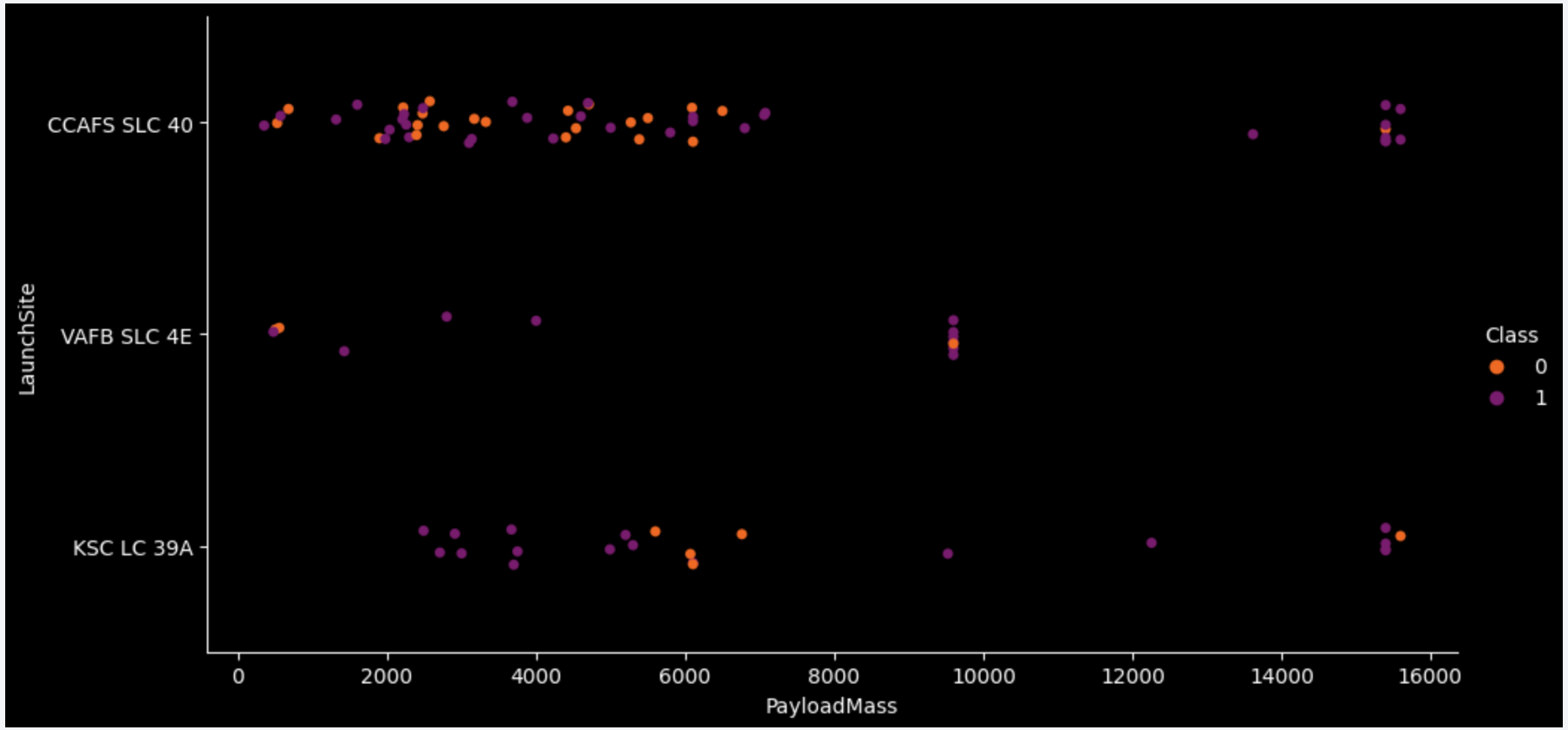
Although the graph shows that the success possibilities of a launch increase along with the flight number, there doesn't seem to be a lot of difference when the launch site changes.



Flight Number vs. Launch Site Category Plot

Payload vs. Launch Site

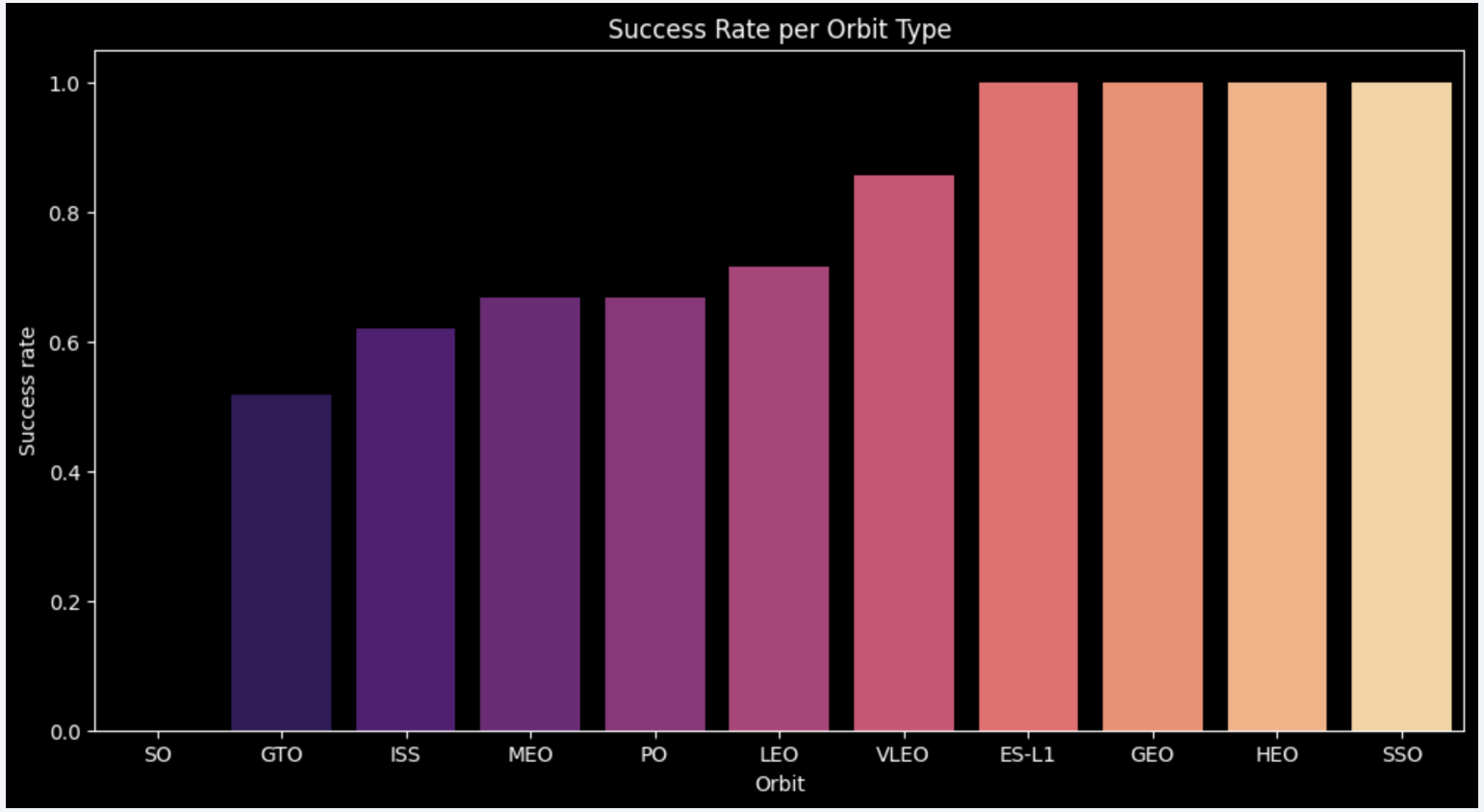
The graph shows that the probability of a failed launch outcome increases when the payload mass is around 500-7,000kg for the "CCAFS SLC-40" launch site, and 5,750-7,000 for "KSC LC 39A".



Payload Mass vs. Launch Site Category Plot

Success Rate vs. Orbit Type

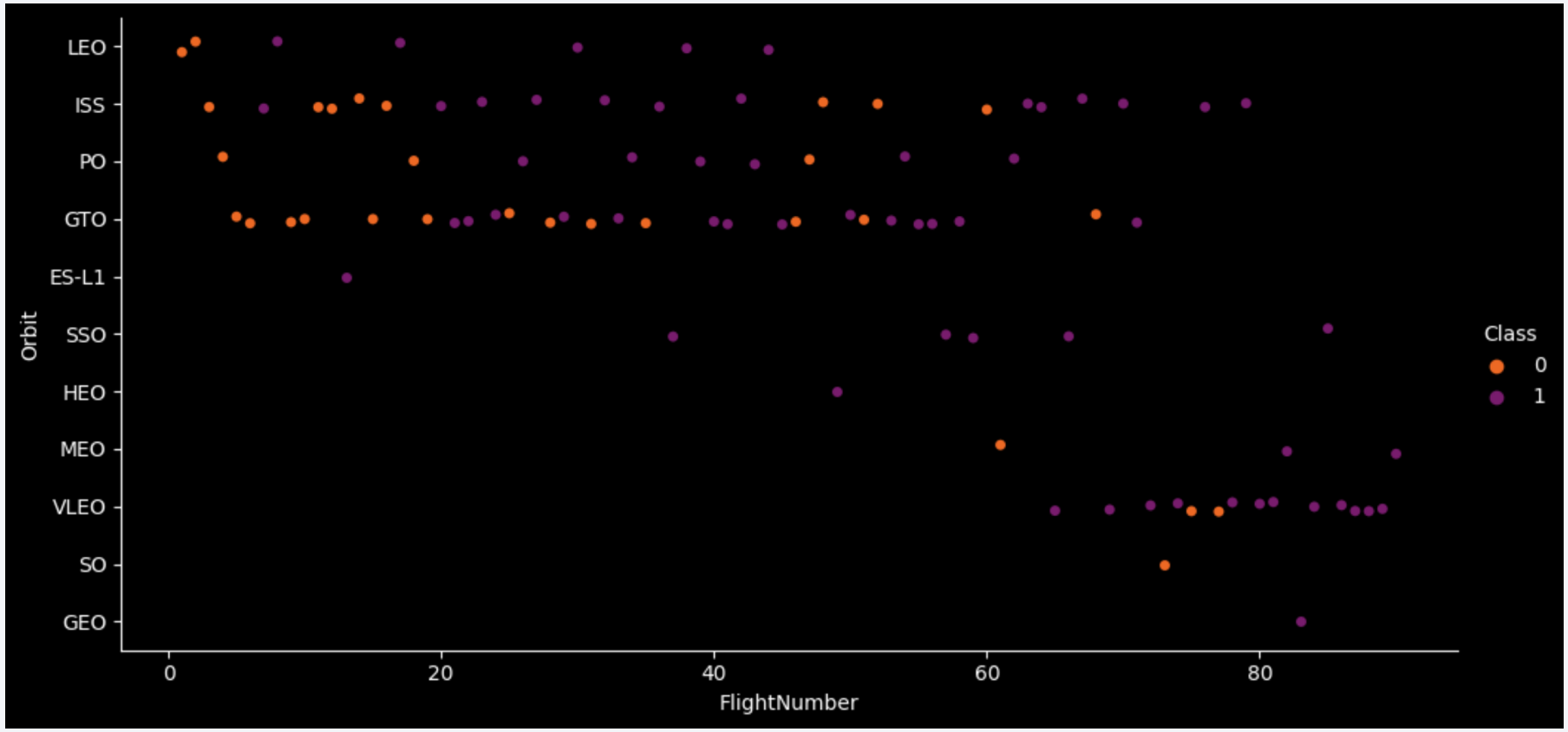
ES-L1, GEO, HEO and SSO orbits have a success probability of 100%, while SO has a 0%. The mode success probability is around 70%.



Orbit Type vs. Success Rate Bar Plot

Flight Number vs. Orbit Type

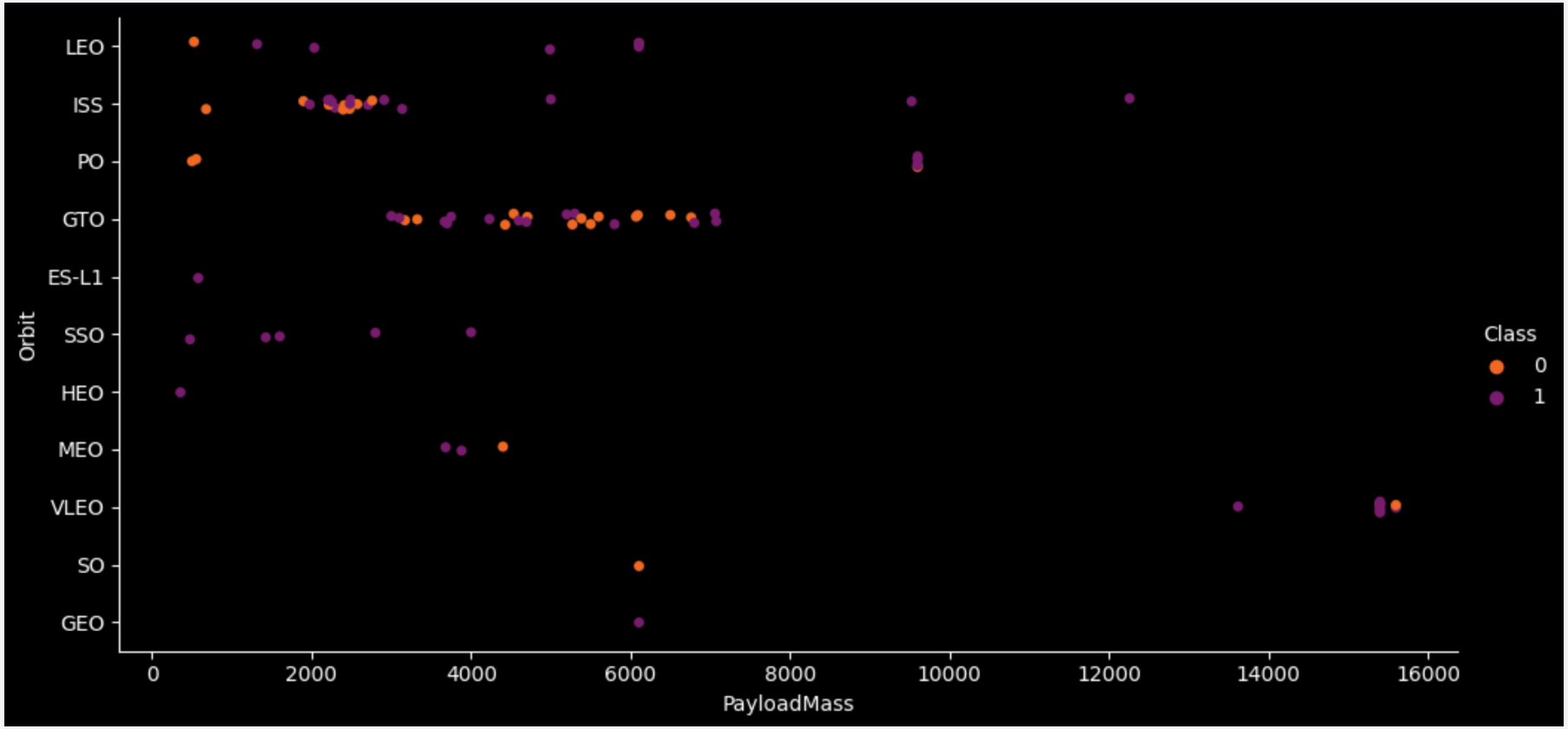
There appears to be a tendency for ISS and GTO orbits that contradict our theory proposed in slide #19, having more failures than usual. Also, as the flight number increases, certain orbits alternate in being reached or not by the rocket.



Flight Number vs. Orbit type Category Plot

Payload vs. Orbit Type

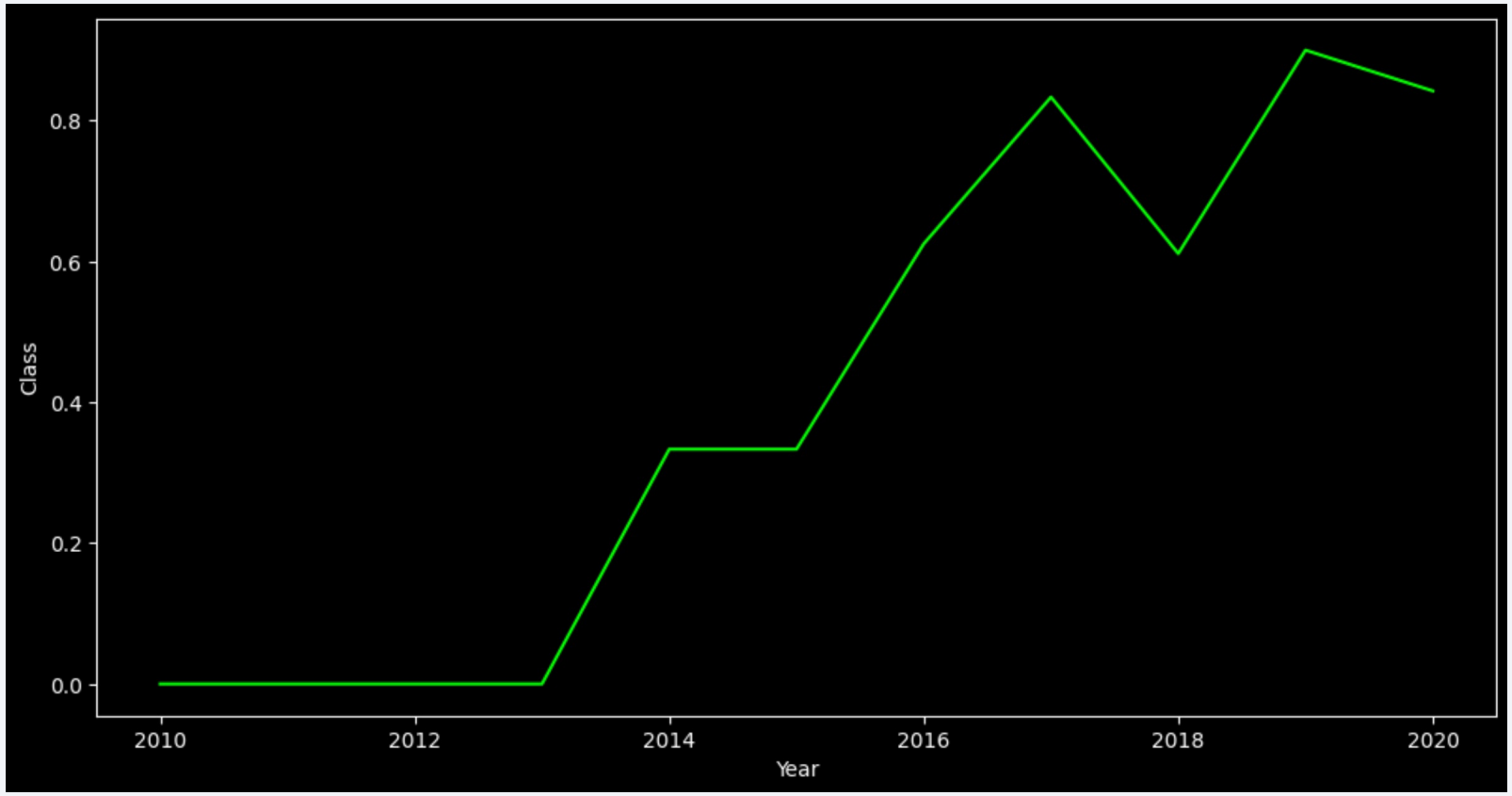
The most common payload mass and orbit type for our launches are 3,000-7,000kg for GTO, and 2,000-3,500kg for ISS. They are also the most prone to fail. The best and most consistent option seems to be the SSO orbit with payload mass around 500-4,000kg.



Payload Mass vs. Orbit type Category Plot

Launch Success Yearly Trend

This line graph confirms our theory that the success rate probability increases over launch number, hence time, as it is inherently correlated with it.



Launch Success Yearly Trend Line Plot

All Launch Site Names

This SQL query returns the different launch site names by using the SELECT DISTINCT command.

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

[9]

```
 * sqlite:///my_data1.db
Done.
```

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

SQL EDA: Task 1

Launch Site Names Begin with 'CCA'

This SQL query returns 5 records of launches having the NASA as their clients

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

[10] Python

```
 * sqlite:///my_data1.db
Done.
```

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

SQL EDA: Task 2

Total Payload Mass

This SQL query displays the total payload mass (in kilograms) for all launches having the NASA as their clients: 48,213kg.

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass_kg
FROM SPACEXTABLE
WHERE Customer LIKE '%CRS%'
```

[11]

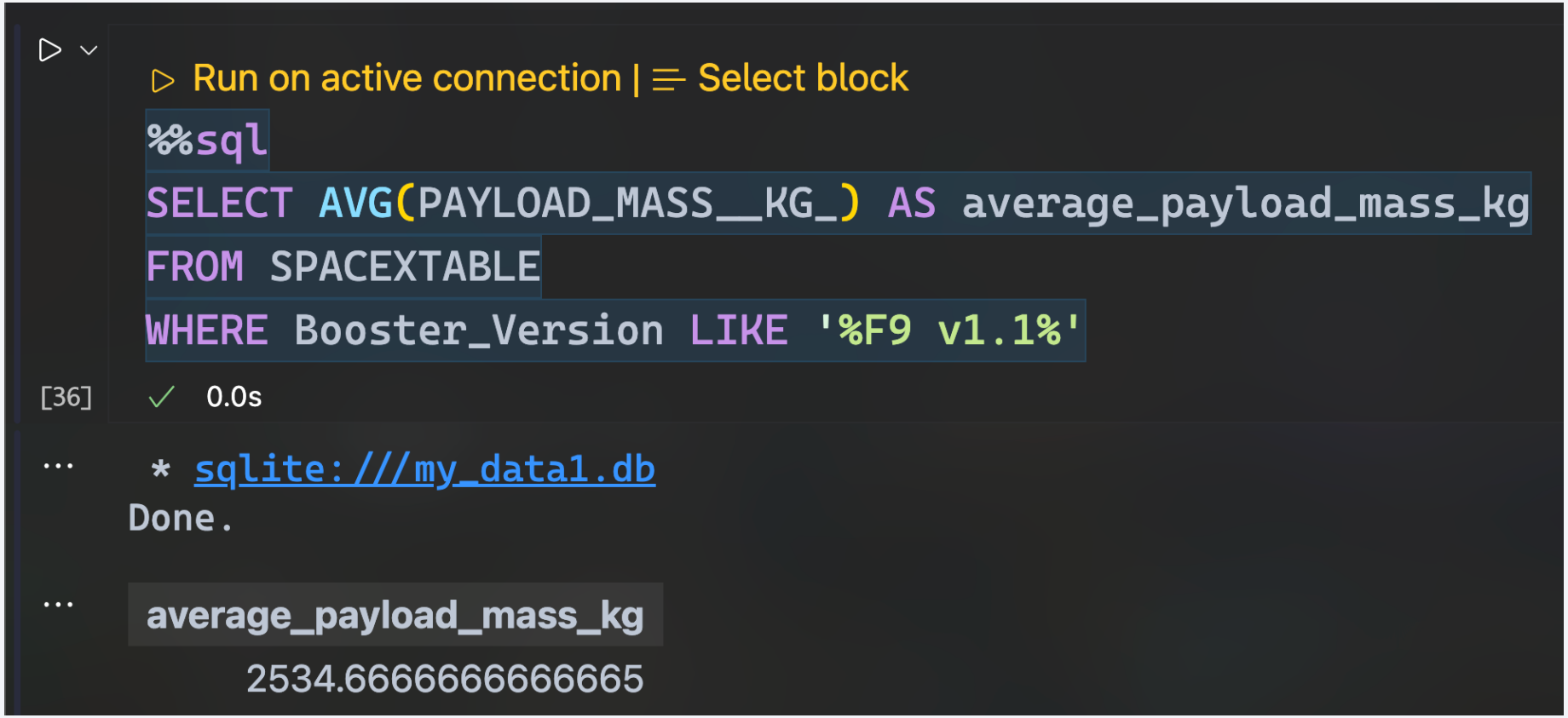
```
 * sqlite:///my_data1.db
Done.
```

total_payload_mass_kg
48213

SQL EDA: Task 3

Average Payload Mass by F9 v1.1

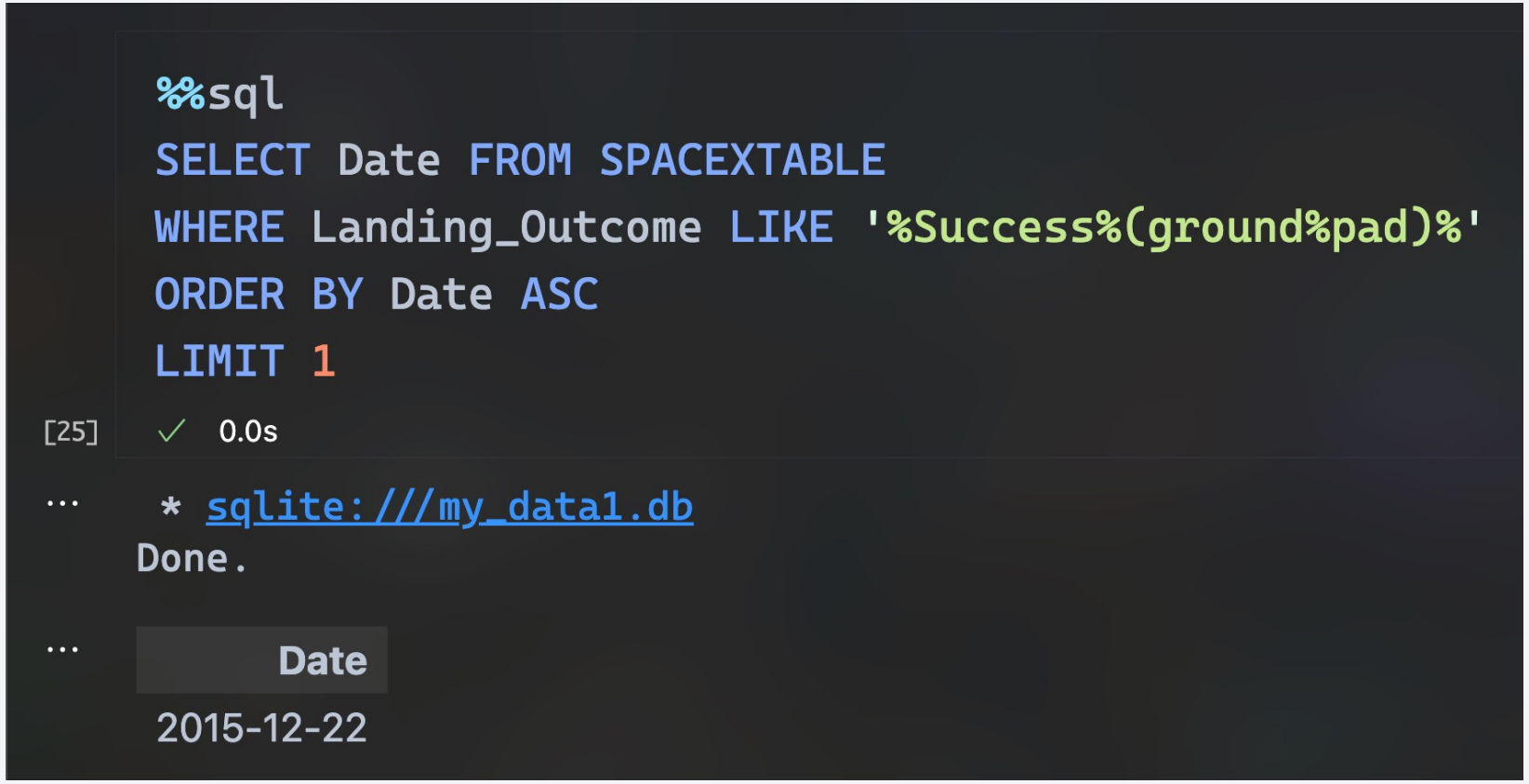
This SQL query returns the average payload mass for all the launches with booster version F9 v1.1, 2,534.667kg.



SQL EDA: Task 4

First Successful Ground Landing Date

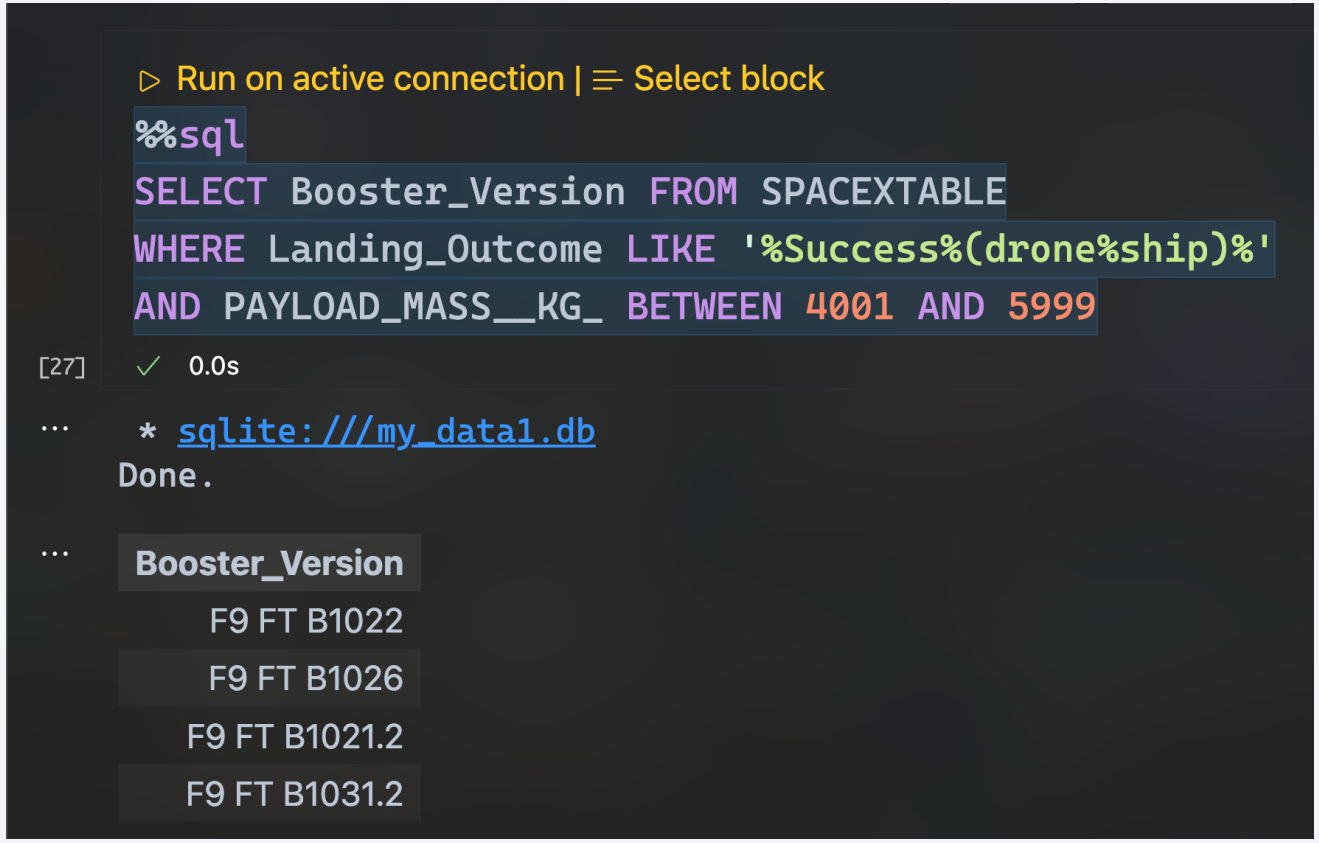
This SQL query retrieves the date of the first successful rocket launch in a ground pad: 22/December/2015. Almost a Christmas gift!



SQL EDA: Task 5

Successful Drone Ship Landing with Payload between 4000 and 6000

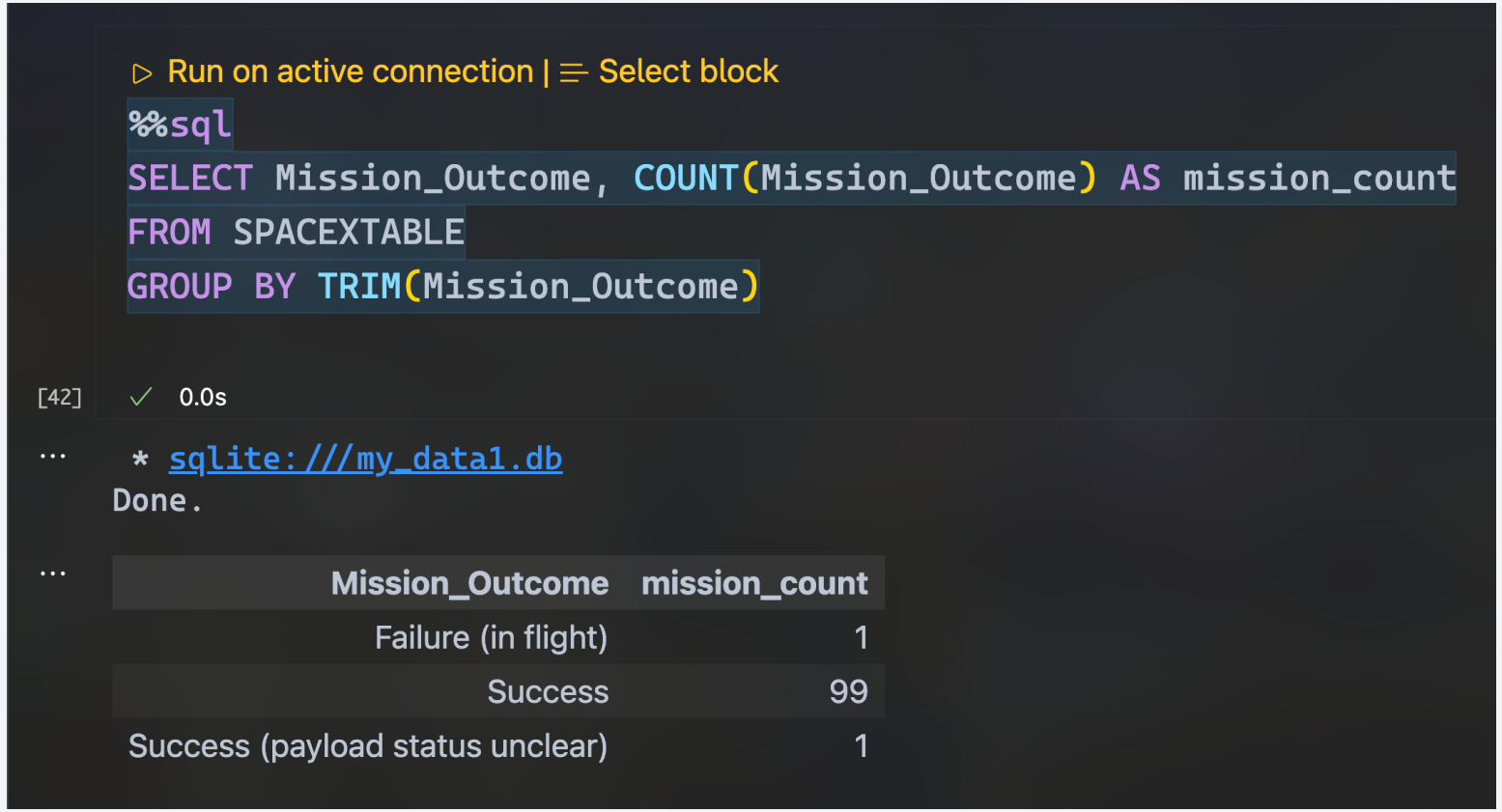
This SQL query retrieves all of the booster version names that have had success in a drone ship landing space, where their payload mass is between 4,000 and 6,000 kilograms.



SQL EDA: Task 6

Total Number of Successful and Failure Mission Outcomes

We see that most of the mission outcomes of these launches went overall successful, having a 99.01% probability of it.



SQL EDA: Task 7

Boosters Carried Maximum Payload

This SQL query retrieves all of the booster version names which have carried the maximum amount of payload mass in the data at least once. A subquery is used in order to correctly filter the information out.

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

[43] ✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

SQL EDA: Task 8

2015 Launch Records

```
▷ Run on active connection | ≡ Select block
%%sql
SELECT
    (CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END) AS month,
    sum(CASE
            WHEN Landing_Outcome LIKE '%Failure%(drone ship)%'
                THEN 1
            ELSE 0
        END
        ) AS drone_ship_failures,
    GROUP_CONCAT(DISTINCT Booster_Version) AS Booster_Versions,
    GROUP_CONCAT(DISTINCT Launch_Site) AS Launch_Sites
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
GROUP BY month
ORDER BY Date
✓ 0.0s
```

SQL EDA: Task 9 (query)

For this SQL query, two CASES are performed: the first one to get the month number and convert it into text in the results, and the second one to sum the amount of failed landing outcomes in drone ship landing spaces, contemplating them for each distinct booster versions and launch sites, which are concatenated to show all of their features in a single line, and only have one register per month.

The information is finally filtered for the year 2015, grouped by the "month" column we just defined in line 16, and in descending order using the "Date" column.

month	drone_ship_failures	Booster_Versions	Launch_Sites
January	1	F9 v1.1 B1012	CCAFS LC-40
February	0	F9 v1.1 B1013	CCAFS LC-40
March	0	F9 v1.1 B1014	CCAFS LC-40
April	1	F9 v1.1 B1015,F9 v1.1 B1016	CCAFS LC-40
June	0	F9 v1.1 B1018	CCAFS LC-40
December	0	F9 FT B1019	CCAFS LC-40

SQL EDA: Task 9 (result)

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

This SQL query displays that, between the dates 04/June/2010 and 20/March/2017, most of the landing outcomes were "no attempt": 10, "Success (drone ship)": 5 and "Failure (drone ship)": 5.

```
%%sql
SELECT Landing_Outcome, count(Landing_Outcome) AS 'count'
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY count(Landing_Outcome) DESC
```

[44] ✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

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

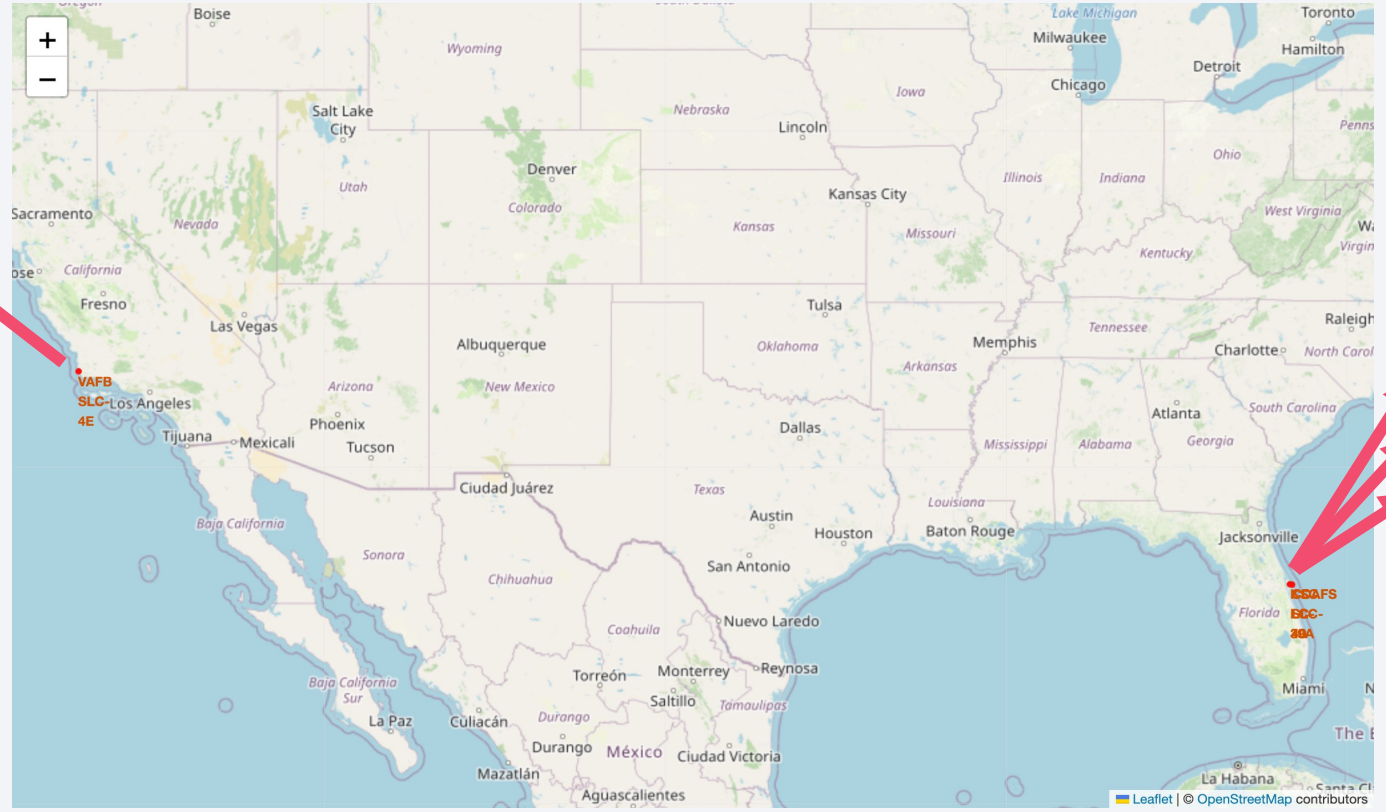
SQL EDA: Task 10


Section 3
Launch Sites
Proximities Analysis

SpaceX Launch Sites

The map displays SpaceX launch sites, strategically positioned near coastlines and transport paths, yet isolated from populated areas for safety. Apart from "VAFB SLC-4E", the sites are in close proximity, making individual site identification on the map difficult due to their tight clustering.

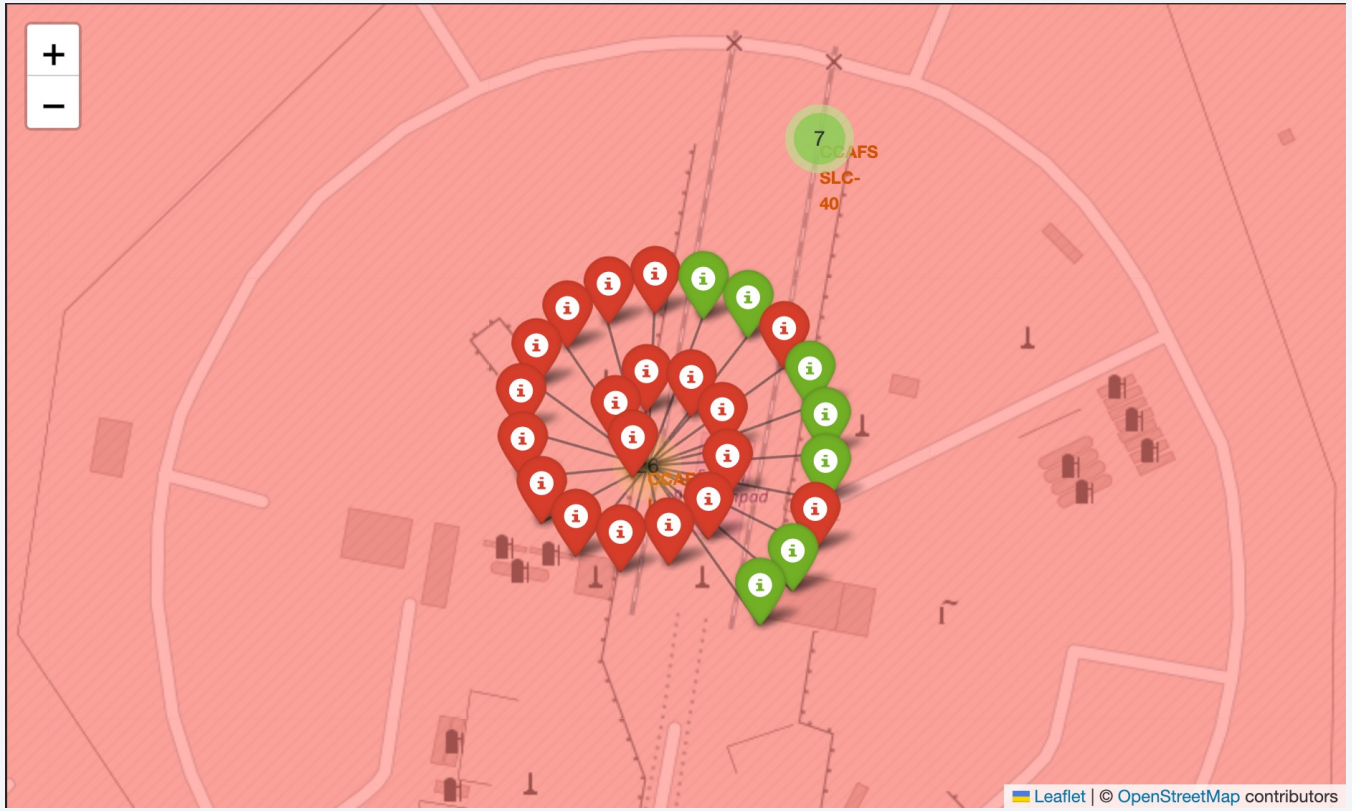
VVAFB SLC-4E



KSC LC-39A
CCAFS SLC-40
CCAFS LC-40

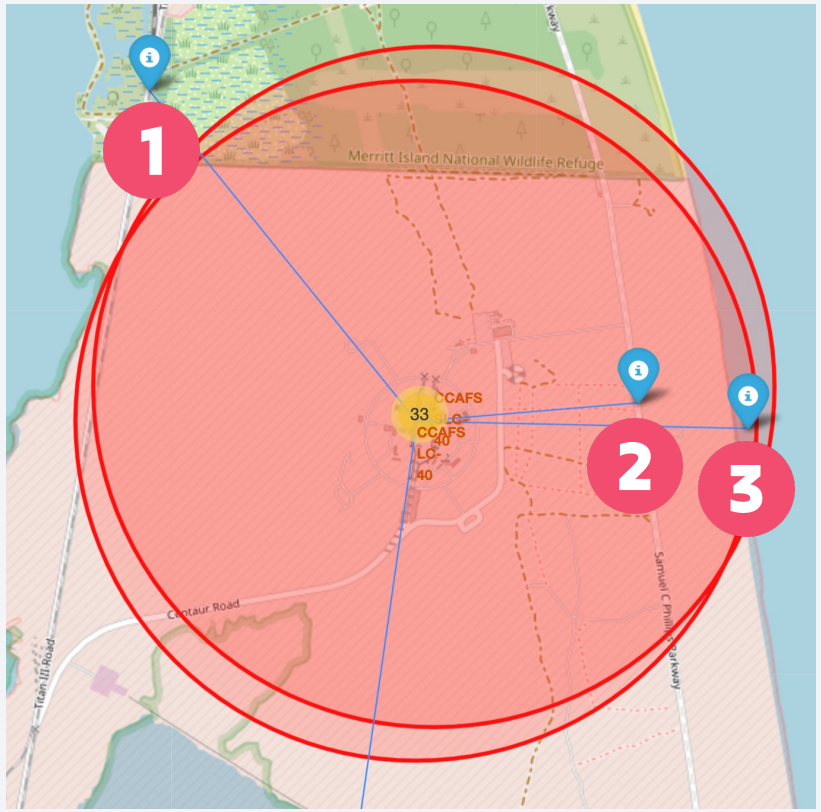
SpaceX Launch Outcome Markers

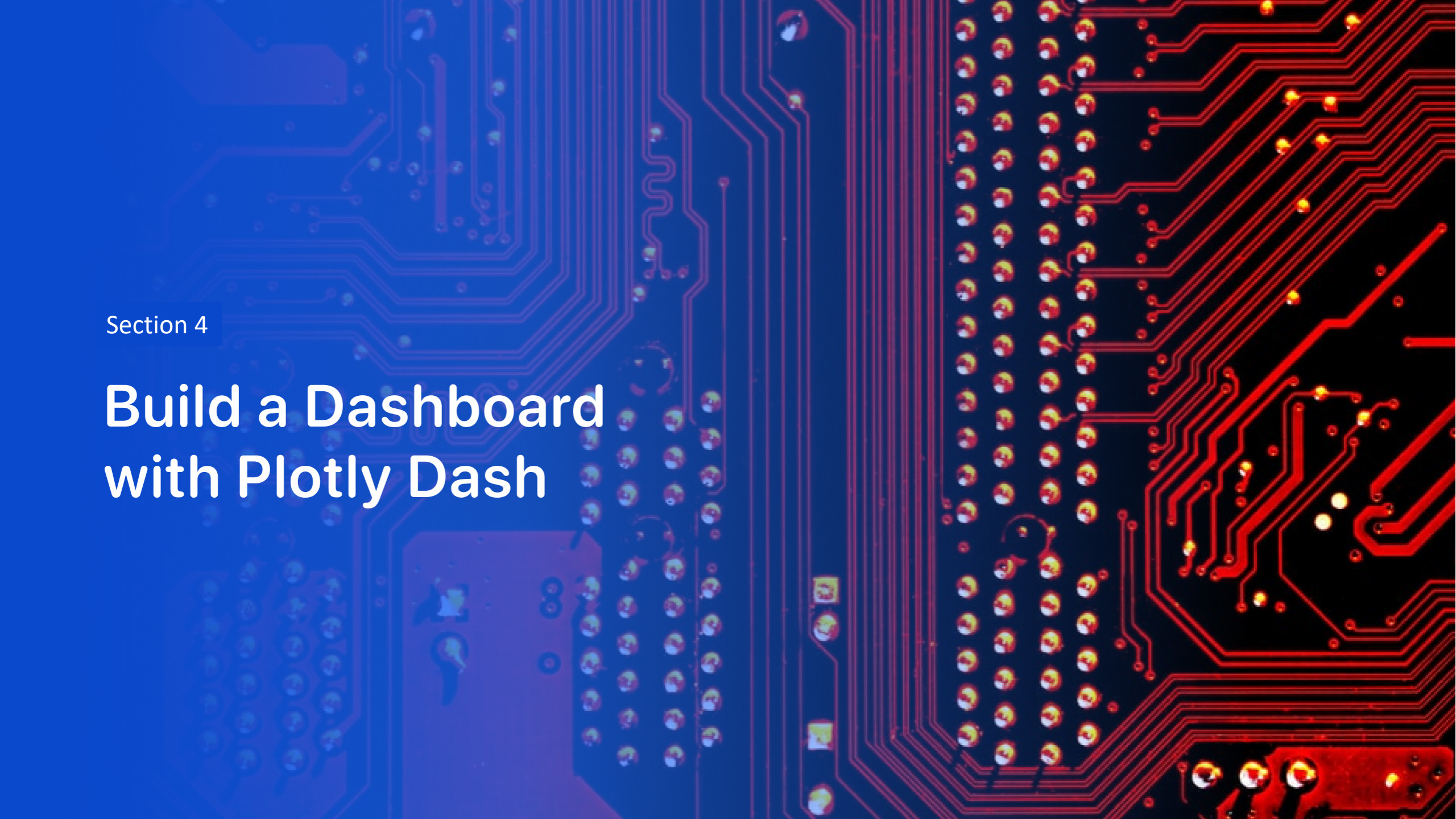
This map markers display all of the registers from the launch site "CCAFS LC-40", being 26 in total. The failed outcomes appear in red, and the successful ones in green.



SpaceX Launch Sites' Nearby Locations

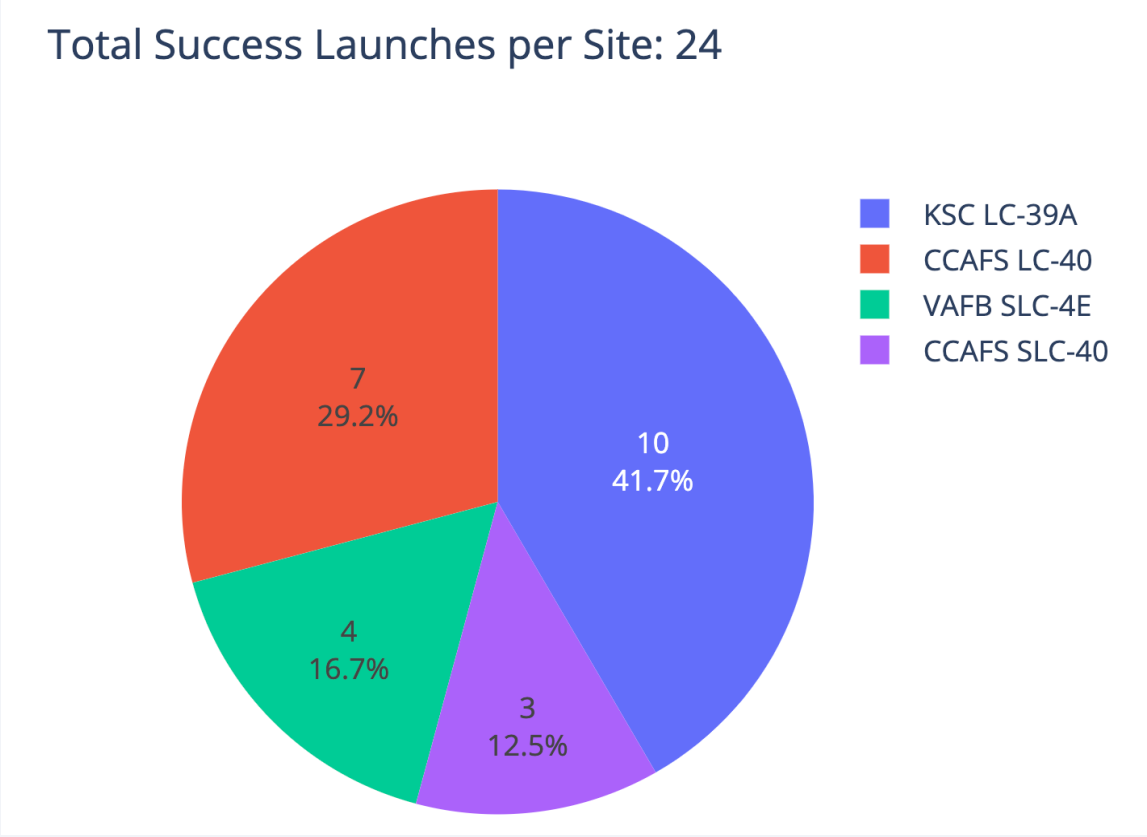
This map markers display some examples of the nearest locations of the launch site "CCAFS LC-40", such as the railway[1], highway[2], coastline[3] and city[4]. This launch site is strategically positioned to have multiple solutions for goods and employees transport, and being close to populated areas and cities in case of emergency.




Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches per Site

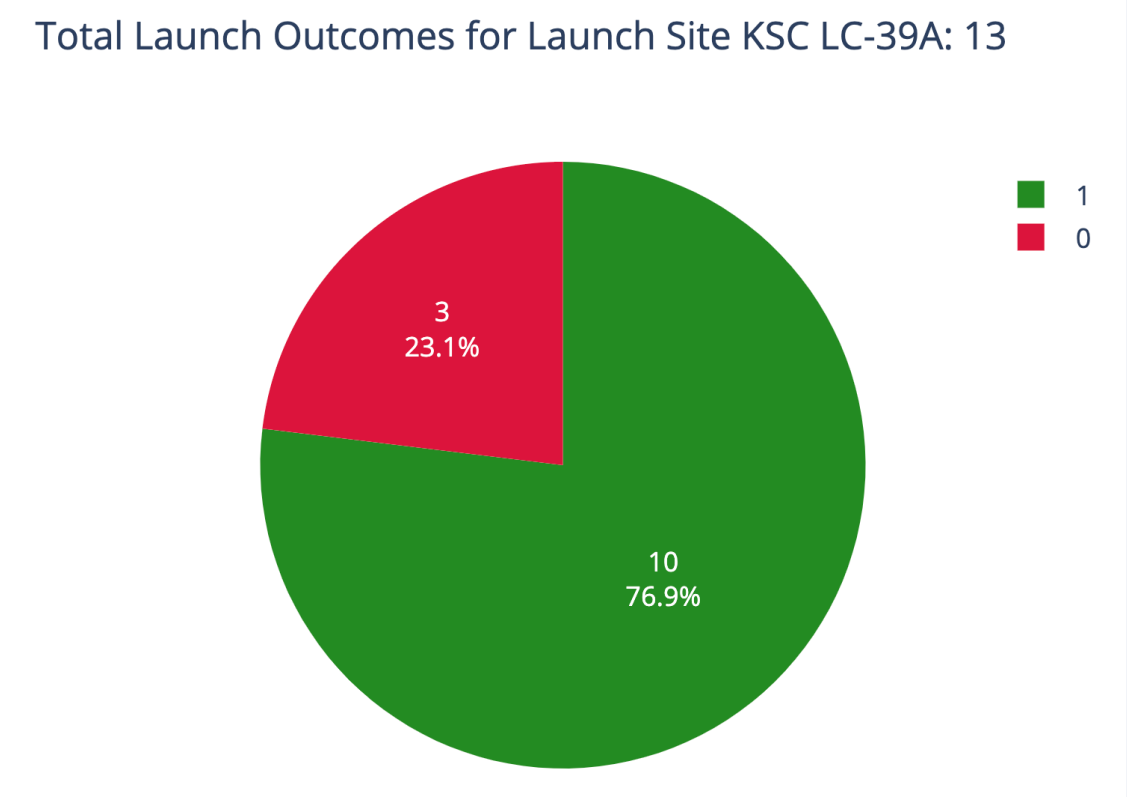
This graph indicates that the launch site with the most successful missions is "KSC LC-39A", with 41.7%, followed by "CCAFS LC-40" with 29.2%



Dashboard: Pie Chart with "All Sites" in dropdown

KSC LC-39A Launch Site Success/Failure Ratio

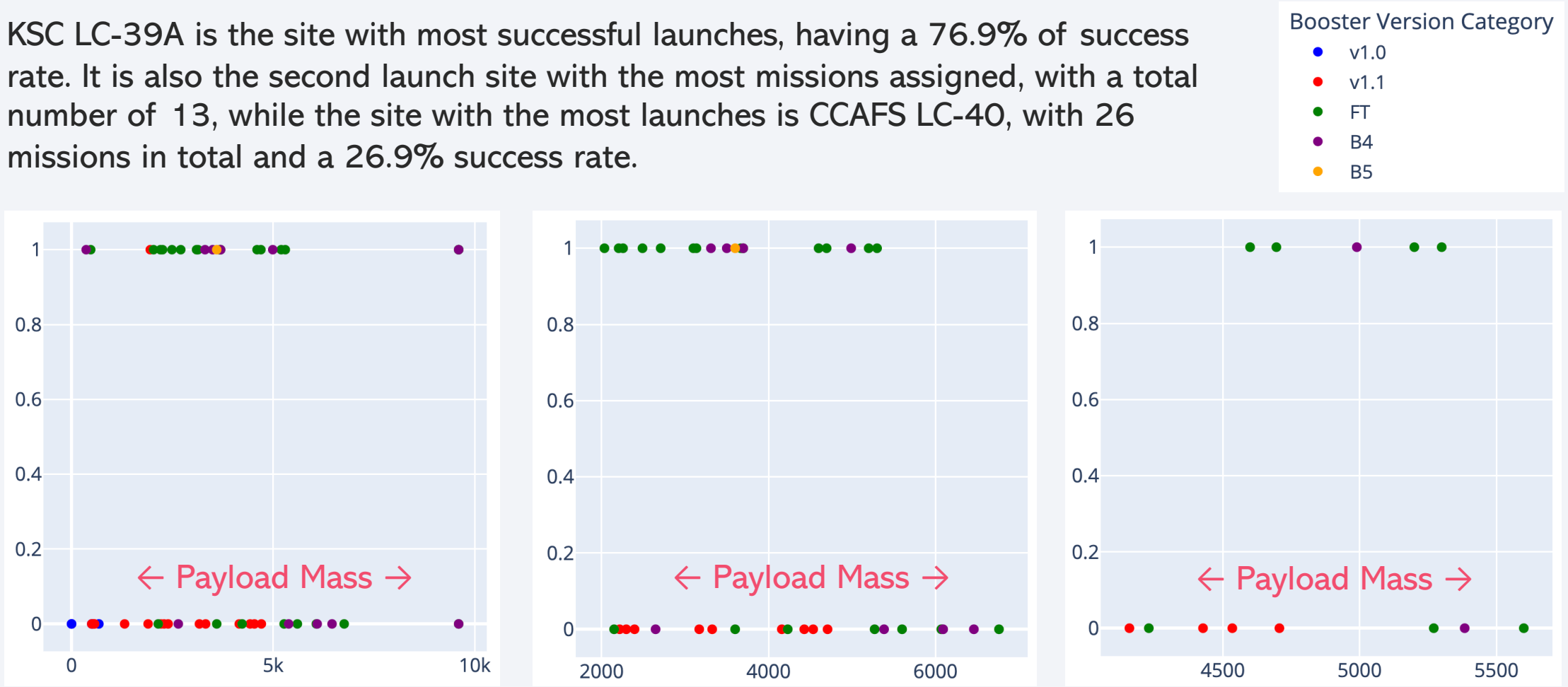
KSC LC-39A is the site with most successful launches, having a 76.9% of success rate. It is also the second launch site with the most missions assigned, with a total number of 13, while the site with the most launches is CCAFS LC-40, with 26 missions in total and a 26.9% success rate.



Dashboard: Pie Chart with "KSC LC-39A" in dropdown

Payload Mass vs. Success Rate for All Sites

KSC LC-39A is the site with most successful launches, having a 76.9% of success rate. It is also the second launch site with the most missions assigned, with a total number of 13, while the site with the most launches is CCAFS LC-40, with 26 missions in total and a 26.9% success rate.



Catplot with no range

Catplot with range 2,000-8,000kg

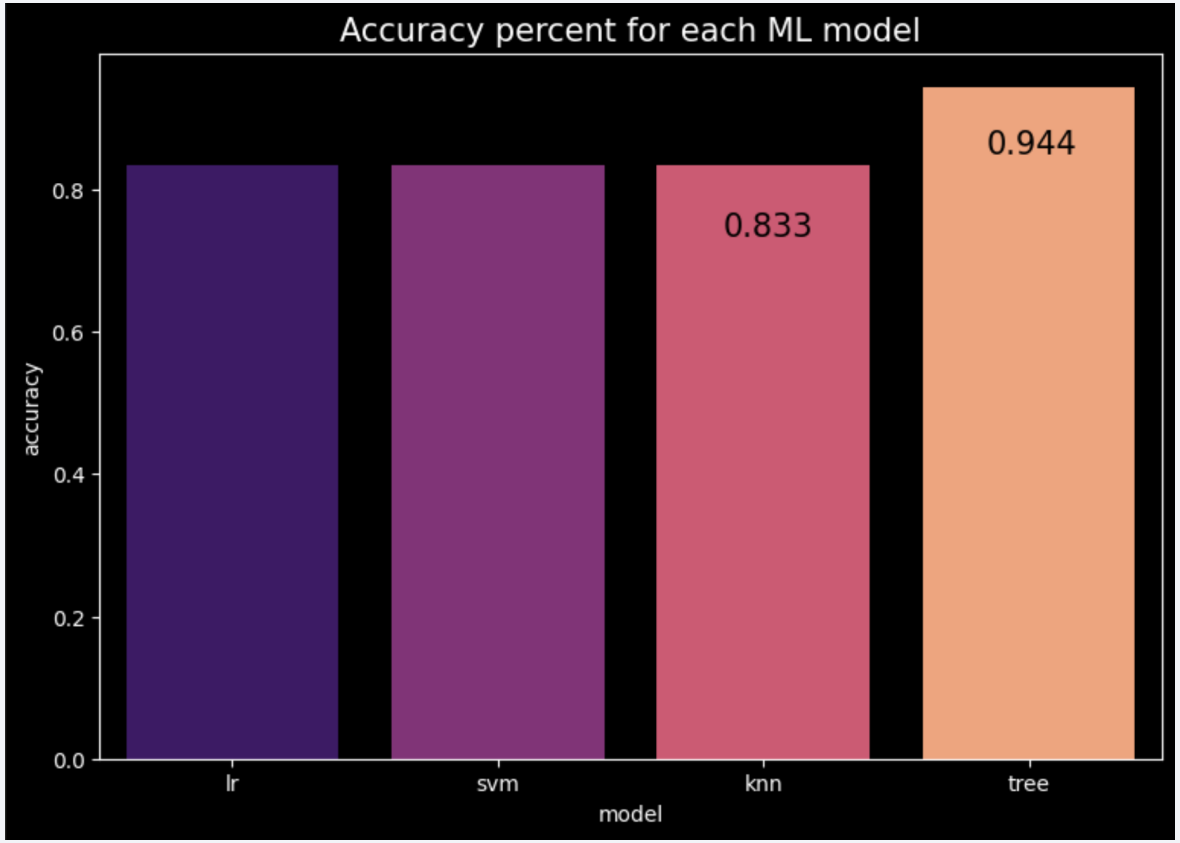
Catplot with range 4,000-6,000kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Here we can see that the Decision Tree (Tree) was the highest in accuracy evaluation with a score of 94.4%. The other three models K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Linear Regression (LR) tied up with a 83.3% score



Model accuracy evaluation

Confusion Matrix

We can appreciate that our Decision Tree Machine Learning model correctly predicted ALL of the successful launches (bottom-right) , and 5 out of 6 failed launches (top-left). The only mistake the model made was predicting one successful launch when in reality it was a failed one (top-right).

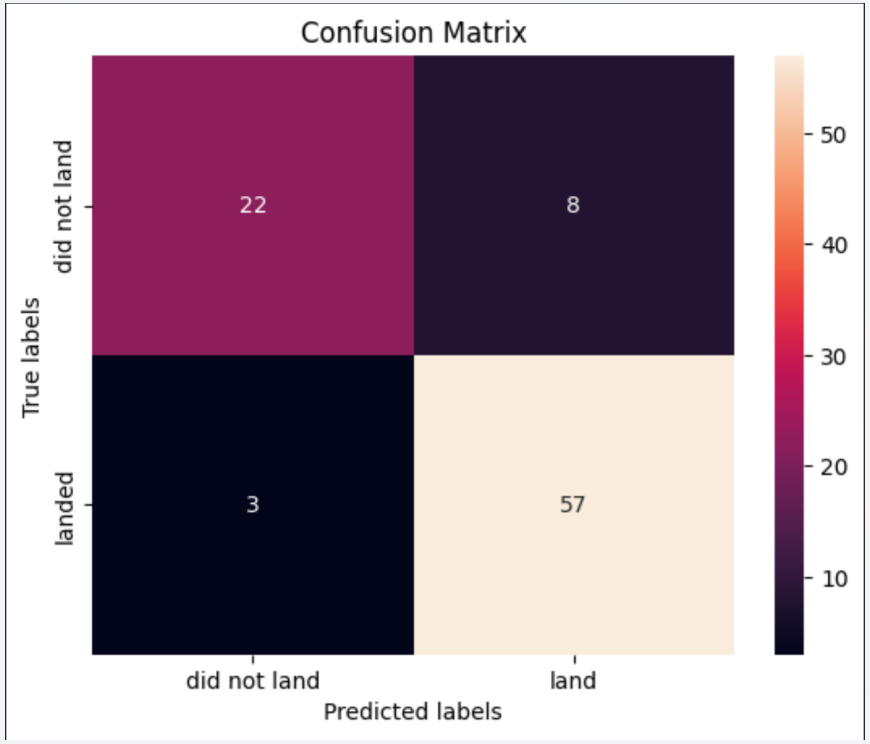
Confusion Matrix for Decision Tree Model

Conclusions

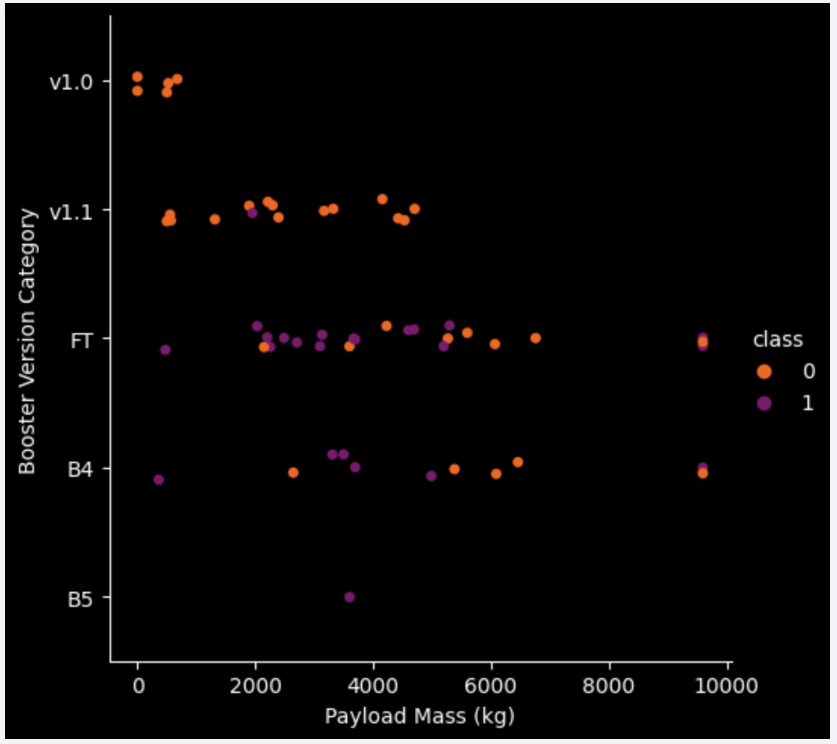
- From the insights drawn, we now know that the success rate of a rocket launch increases as time and launch number increase. This makes sense as every SpaceX launch generates data, and new technologies and improvements come up over time.
- The location of the launch sites are strategically planned in advance, positioned nearby coastlines, highways and railways, while being distant from cities and populated areas.
- Without contemplating time, the rocket launch with the highest probability of success seems to be the SSO orbit with a payload mass of 500-4,000kg, having a success rate of 100%. This is viable because it is inside our payload average mass for FT booster version rockets. ES-L1, GEO and HEO orbits, which also have a success probability of 100% are excluded because of their low launch count.
- The Machine Learning model that best performed, accuracy-wise, was the Decision Tree with the hyper parameters {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}. The model scored 94.4% on the accuracy test, while the others (LR, SVM and KNN) scored 83.3%

Appendix

This confusion matrix showcases the accuracy score when the Decision Tree model is trained with the entire dataset



This graph was generated afterwards to explain what the safest rocket launch was (Conclusion: Point 2), and why



Thank you!