IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Subramanya Adiga G

30 Jan 2025

Outline

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

Executive Summary

- Summary of methodologies
- **Data Collection Methodology :** Web scraping of SpaceX launch data, SpaceX Rest API Endpoints
- **Data Analysis Approach:** Comprehensive data wrangling and cleaning, Advanced data visualization techniques, Interactive visual analytics implementation, Statistical pattern analysis
- **Machine Learning Applications:** Multiple classification models tested, Feature importance ranking, Prediction accuracy optimization, Model performance evaluation

- Summary of all results
- Successfully gathered comprehensive launch data from multiple public sources including SpaceX API and web scraping
- Through EDA, identified critical features affecting landing success: Payload mass,Launch site location,Orbit type,Weather conditions.
- Machine Learning Prediction showed the best model to predict which characteristics are important to drive this opportunity by the best way, using all collected data.

Introduction

- In this project, we analyse SpaceX's Falcon 9 launch data to help Space Y develop a competitive pricing strategy in the commercial space industry.
- By leveraging public data and machine learning techniques, we predict the probability of successful first-stage landings - a crucial factor in determining launch costs.
- Our analysis focuses on identifying key factors that influence landing success, enabling accurate cost estimation for Space Y's future launches.
- This data-driven approach provides strategic insights for competing with SpaceX's reusable rocket technology, potentially reducing launch costs from $62 million to as low as $28 million

Section 1

Methodology

Methodology

- **Executive Summary**
- **Data collection methodology:**
- SpaceX launch data that is gathered from SpaxeX Rest API and Wiki pages
- **Perform data wrangling**
- Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analysing features
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- Our analysis encompassed creating scatter plots and bar charts in Python to visualize SpaceX launch data, performing extensive exploratory data analysis using Pandas, and executing SQL queries to extract meaningful patterns. Through these visualizations and data manipulations, we identified crucial relationships between launch parameters and landing success rates, laying the groundwork for our predictive modeling.
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- SpaceX launch data that is gathered from below two sources
  - The SpaceX REST API endpoints – https://api.spacexdata.com/v4/ (End point- /launches/past)
  - Web scraping the Wiki pages - (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)

Data Collection – SpaceX API

- SpaceX offers endpoint URL

  https://api.spacexdata.com/v4

  where the data collected about rocket identification number and then used below endpoints to get

- To get information about the launches using the IDs given for each launch(mainly rocket, payloads, launchpad, and cores.)

https://api.spacexdata.com/v4/rockets/

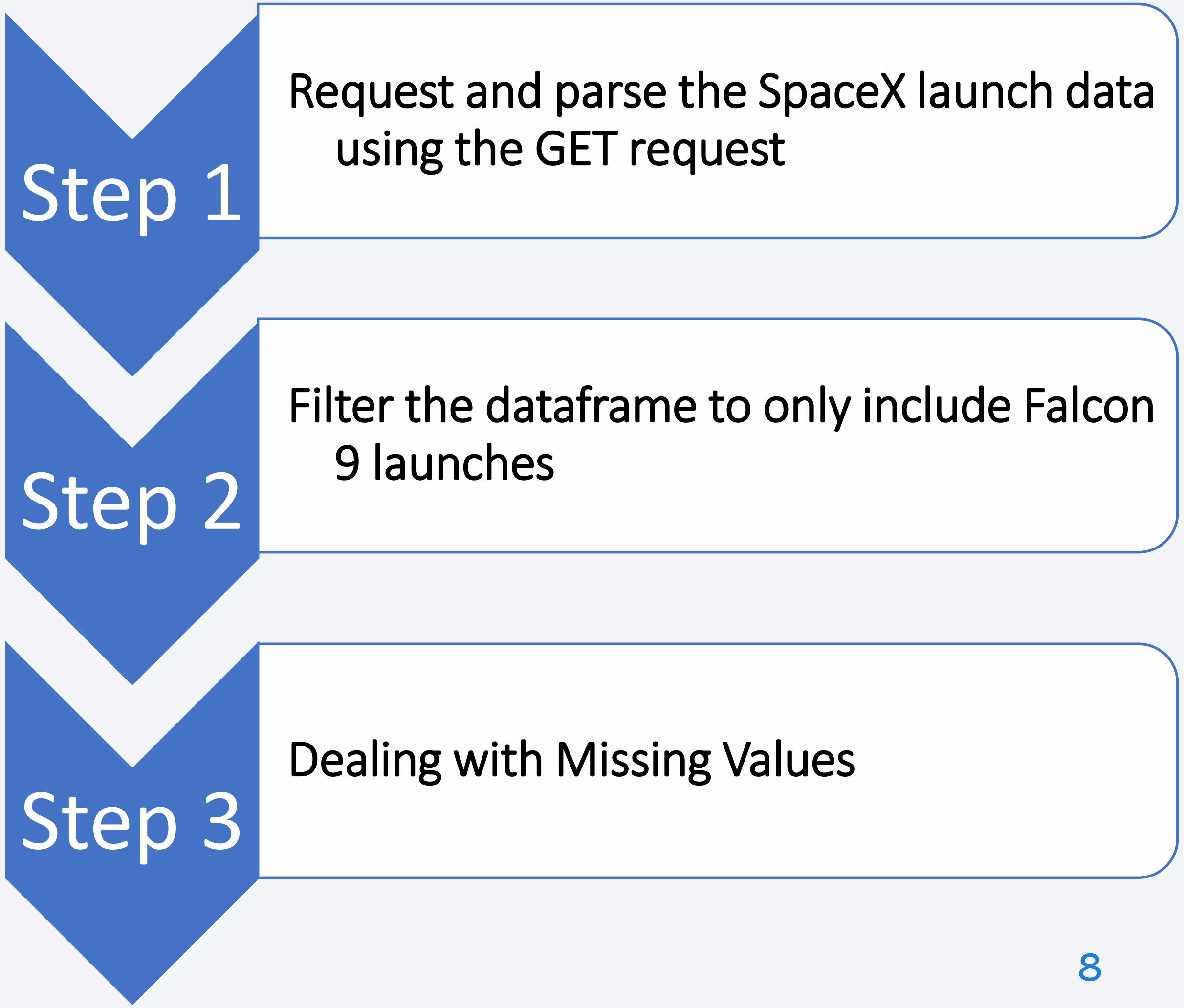
https://api.spacexdata.com/v4/launchpads

https://api.spacexdata.com/v4/payloads/

https://api.spacexdata.com/v4/cores/

My GIT Hub Link

https://github.com/sadigag/coursera-project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Step 1

Request and parse the SpaceX launch data using the GET request

Step 2

Filter the dataframe to only include Falcon 9 launches

Step 3

Dealing with Missing Values

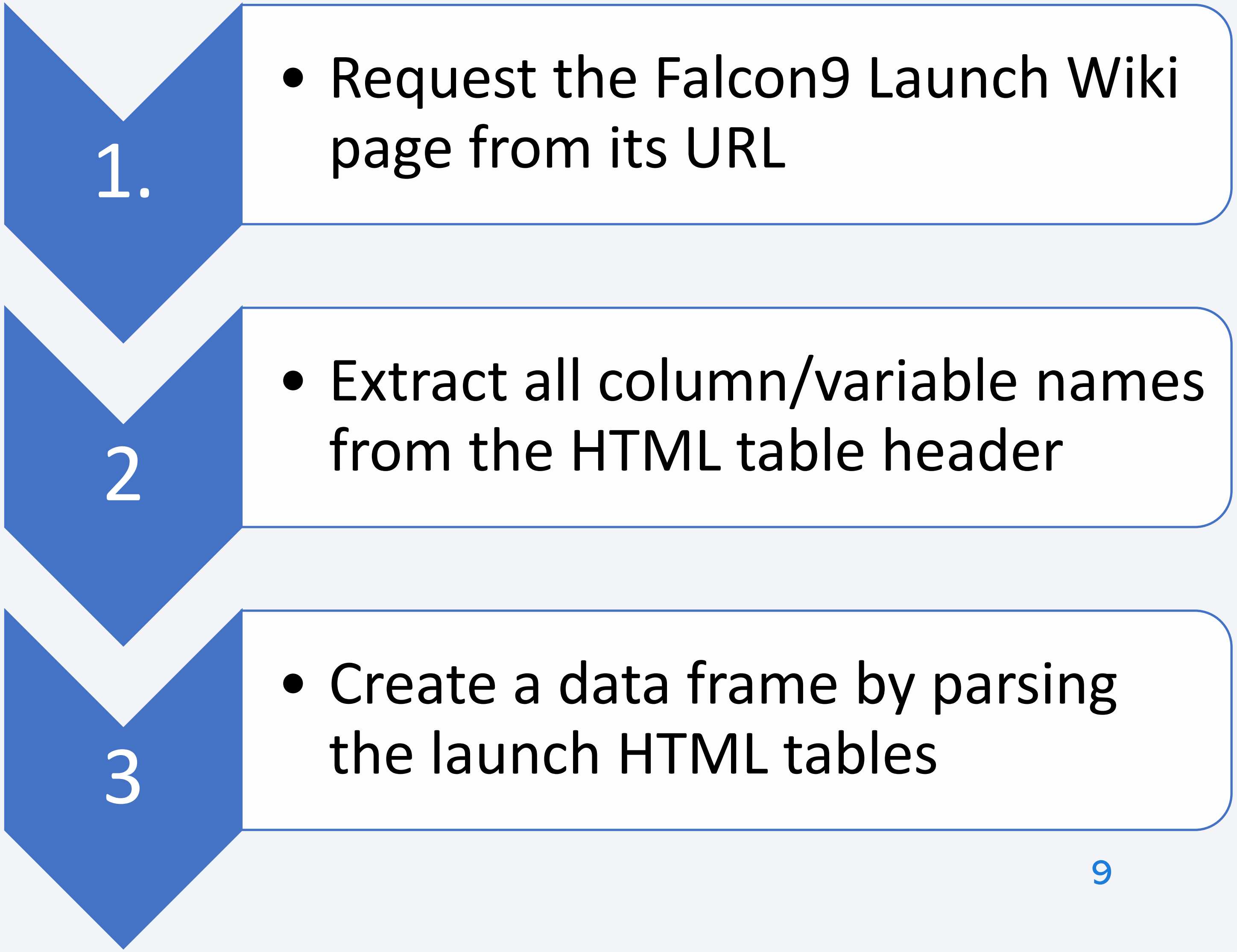
Data Collection - Scraping

- We have performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page: "List of Falcon 9 and Falcon Heavy launches“ below is the URL we referred:

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

- **My GIT Hub Link**

https://github.com/sadigag/coursera-project/blob/main/jupyter-labs-webscraping.ipynb



1. Request the Falcon9 Launch Wiki page from its URL

2. Extract all column/variable names from the HTML table header

3. Create a data frame by parsing the launch HTML tables

Data Wrangling

- Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then the summaries launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
- Finally, the landing outcome label was created from Outcome column.

-

My GIT Hub Link

- https://github.com/sadigag/coursera-project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Created scatter plots analyzing:
  - Payload mass vs. landing success
  - Launch site locations vs. success rates
  - Flight number vs. landing outcomes
- Developed bar charts showing:
  - Success rates by launch site
  - Orbit type distribution
  - Landing success trends over time

EDA with SQL

- **The following SQL queries were performed:**
- Names of the unique launch sites in the space mission;
- Top 5 launch sites whose name begin with the string 'CCA’;
- Total payload mass carried by boosters launched by NASA (CRS);
- Average payload mass carried by booster version F9 v1.1;
- Date when the first successful landing outcome in ground pad was achieved;
- Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
- Total number of successful and failure mission outcomes;
- Names of the booster versions which have carried the maximum payload mass;
- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015; and
- Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- **My GIT Hub Link**
- https://github.com/sadigag/coursera-project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps
- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.
- **My GIT Hub Link**
- https://github.com/sadigag/coursera-project/blob/main/lab_jupyter_launch_site_location1.ipynb

Build a Dashboard with Plotly Dash

- The following graphs and plots were used to visualize data
  - Percentage of launches by site
  - Payload range
- This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is best place to launch according to payloads.

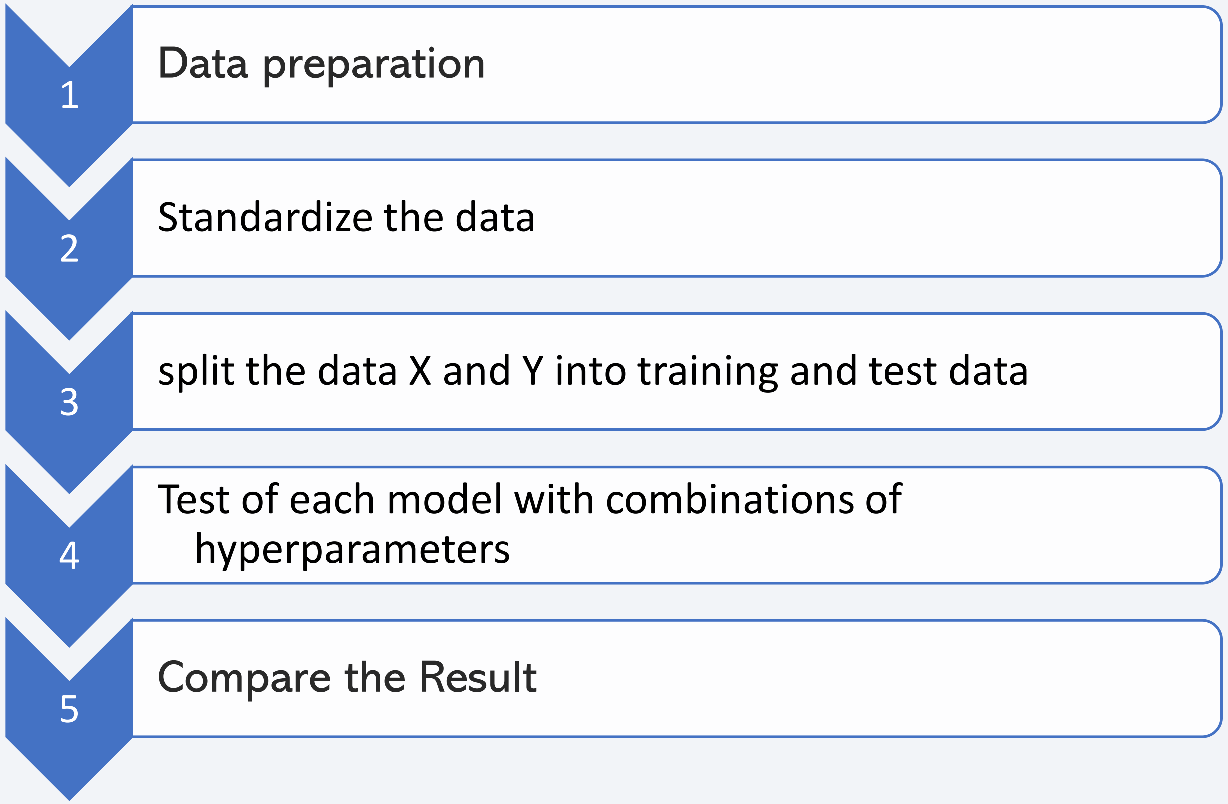
- **My GIT Hub Link**
- https://github.com/sadigag/coursera-project/blob/main/DashboardWithPlotyDash.py

Predictive Analysis (Classification)

- Four classification models were compared:
  - Logistic regression,
  - Support vector machine,
  - Decision tree
  - K nearest neighbors.

- **My GIT Hub Link**

- https://github.com/sadigag/coursera-project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



1 Data preparation

2 Standardize the data

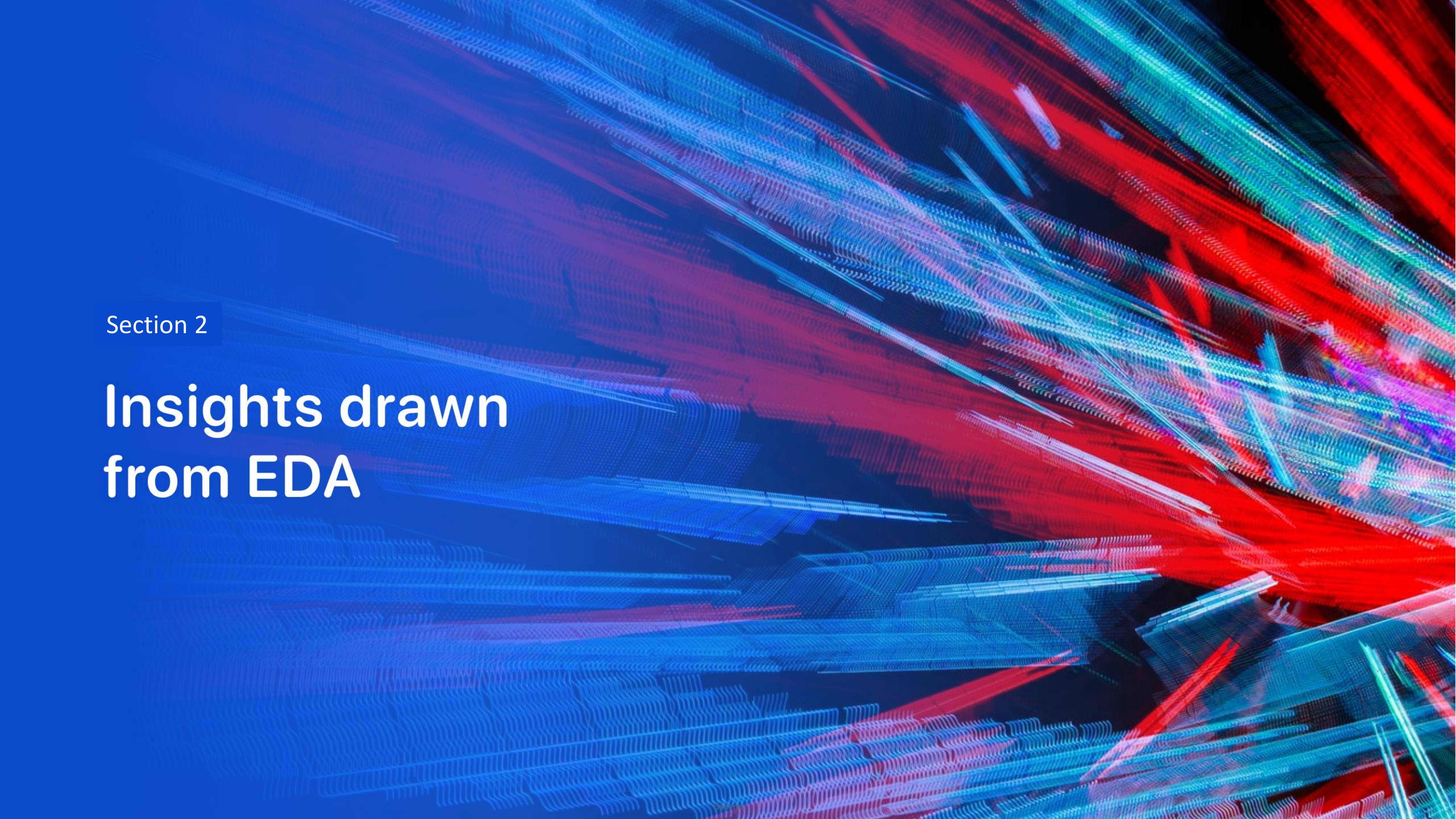
3 split the data X and Y into training and test data

4 Test of each model with combinations of hyperparameters

5 Compare the Result

Results

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

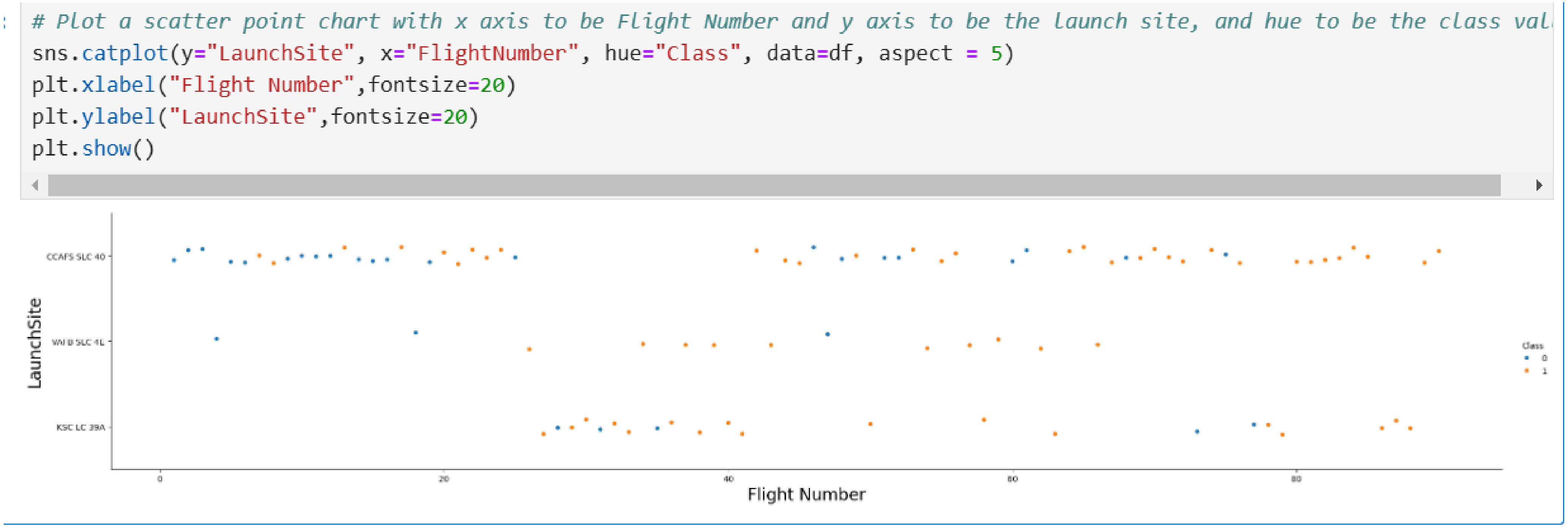


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

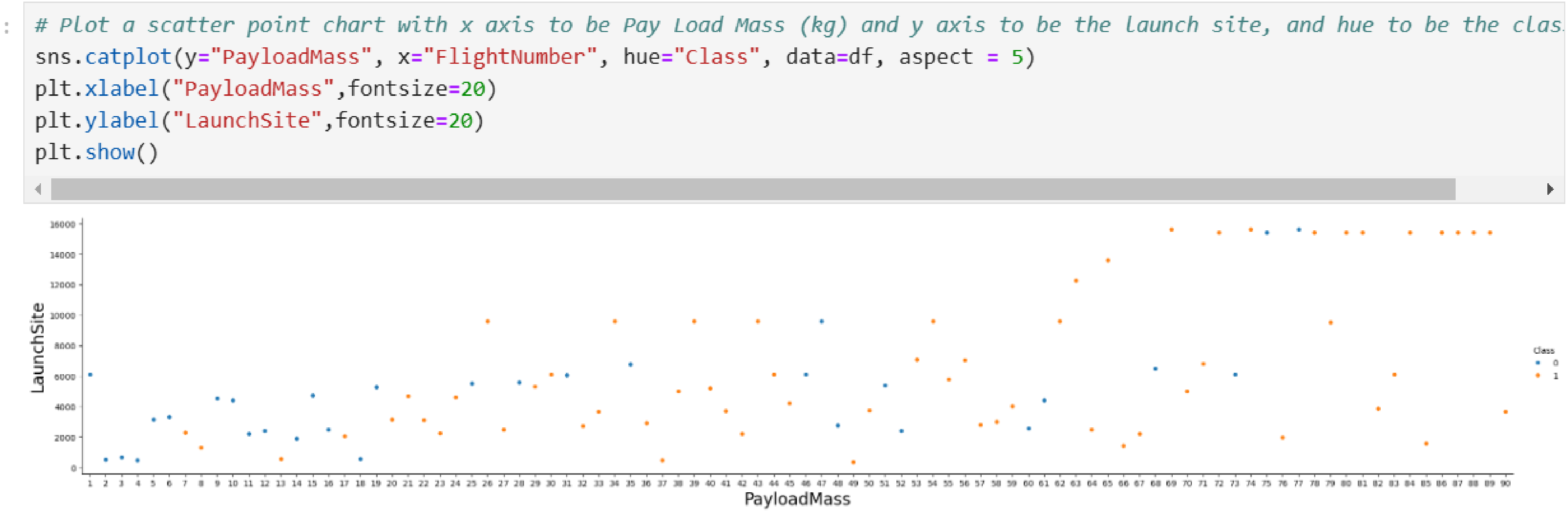
- According to the plot above, it’s possible to verify that the best launch site nowadays is CCAF5 SLC 40, where most of recent launches were successful;
- •In second place VAFB SLC 4E and third place KSC LC 39A;
- •It’s also possible to see that the general success rate improved over time.



```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class val
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

Payload vs. Launch Site

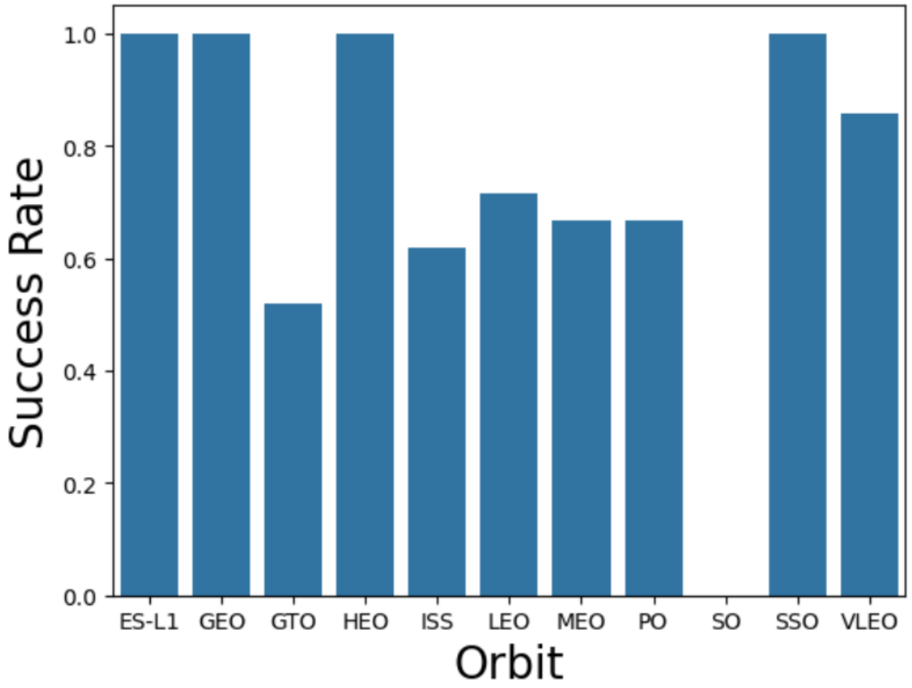
- Payloads over 9,000kg (about the weight of a school bus) have excellent success rate;
- Payloads over 12,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.



```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the clas
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

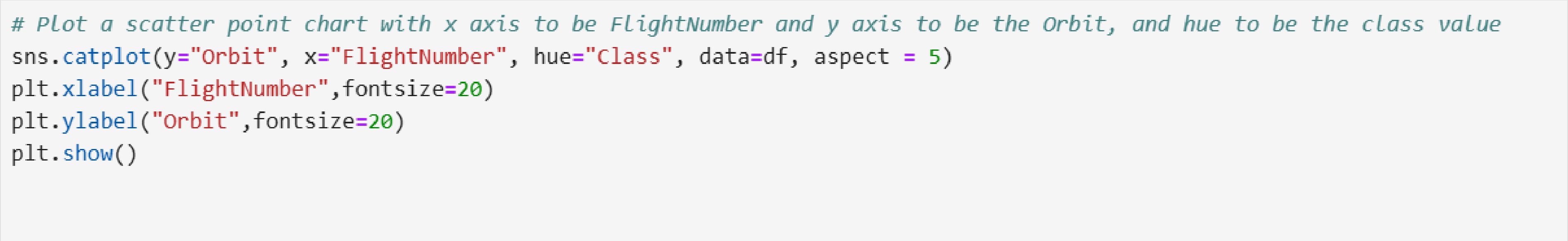
Success Rate vs. Orbit Type

- The biggest success rates happens to orbits:
- •ES-L1 GEO HEO ,SSO.
- •Followed by:
- •VLEO (above 80%); and LFO (above 70%).

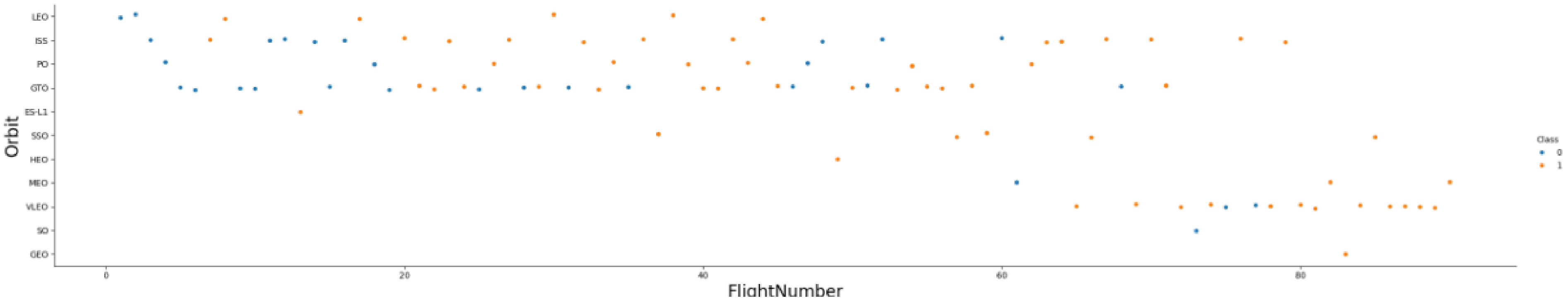


Flight Number vs. Orbit Type

- Apparently, success rate improved over time to all orbits;
- VLEO orbit seems a new business opportunity, due to recent increase of its frequency.

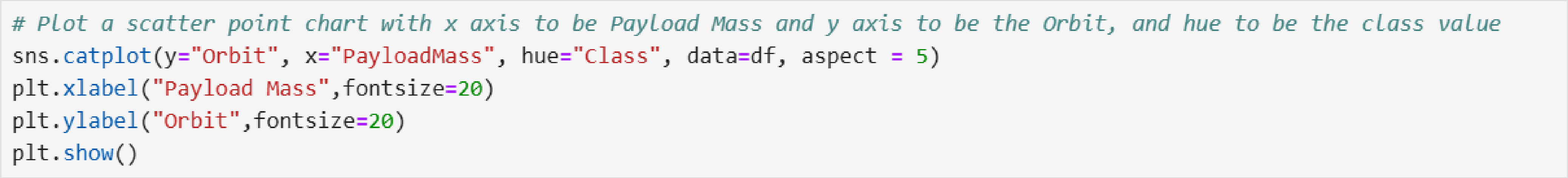


```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

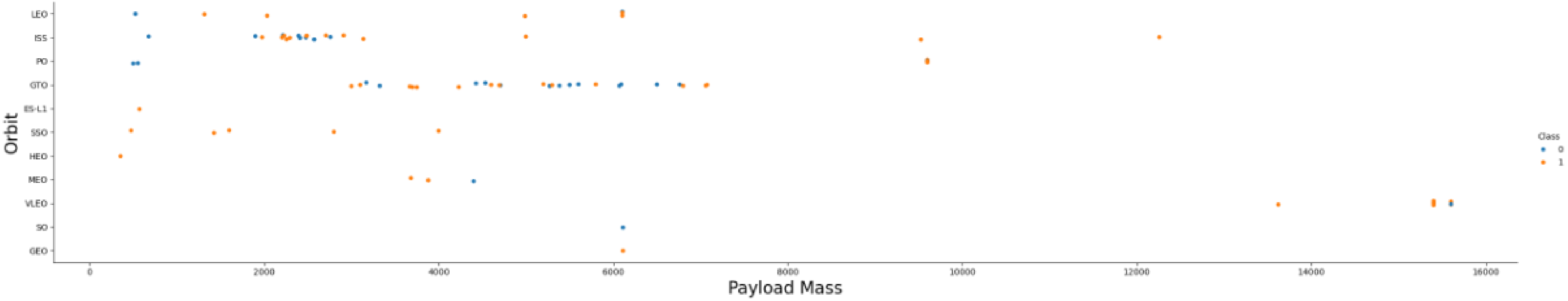


Payload vs. Orbit Type

- Apparently, there is no relation between payload and success rate to orbit GTO;
- ISS orbit has the widest range of payload and a good rate of success;
- There are few launches to the orbits SO and GEO.



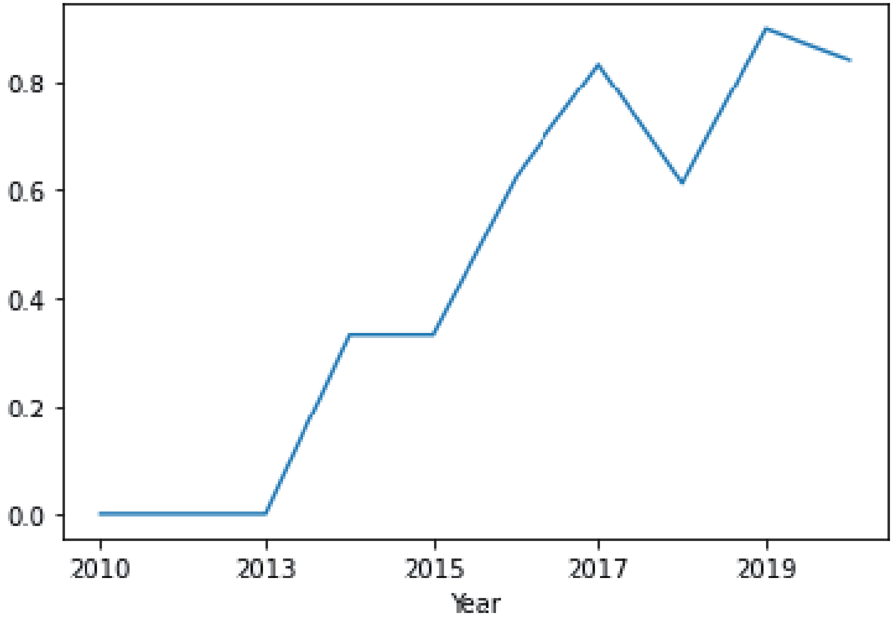
```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- Success rate started increasing in 2013 and
- kept until 2020

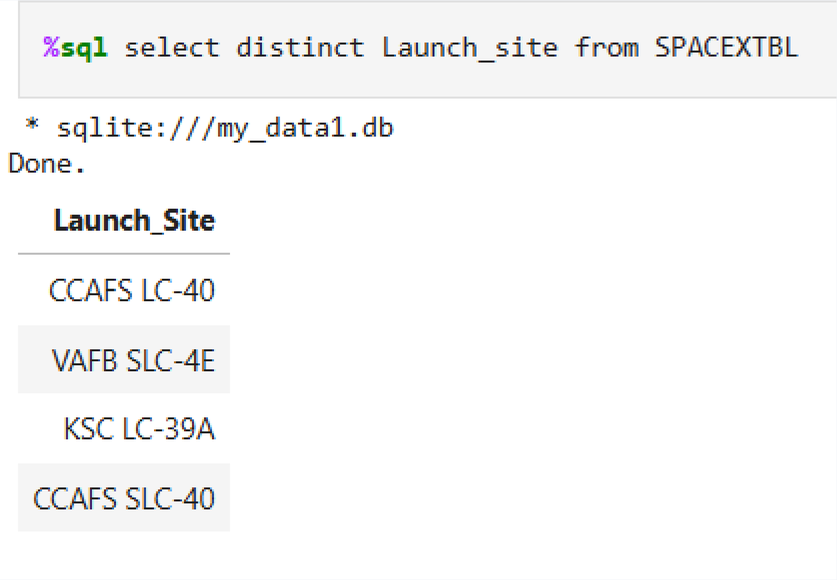
- It seems that the first three years were a
- period of adjusts and improvement of
- technology.



All Launch Site Names

- According to data, there are four launch sites:

- **LaunchSite**
- CCAFS LC-40
- CCAFS SLC-40
- KSC LC-39A
- VAFB SLC-4E



```
%sql select distinct Launch_site from SPACEXTBL
```

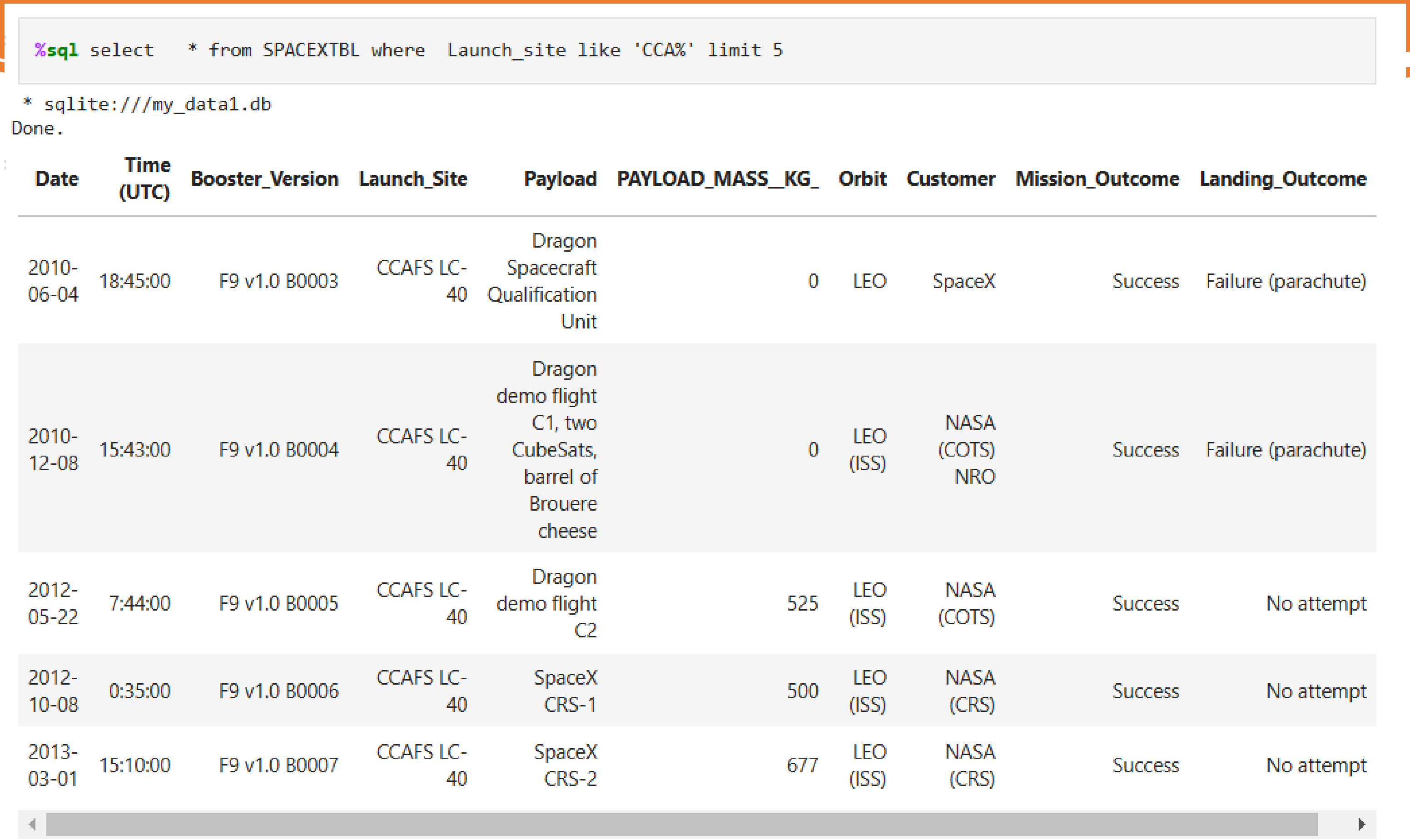
```
 * sqlite:///my_data1.db
Done.
```

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

- They are obtained by selecting unique(distinct) occurrences of “launch_site” values from the dataset. Above screenshot taken from Jupiter notebook

Launch Site Names Begin with 'CCA'

- SQL query with condition ‘like ‘CCA%’ will search launch site name starts with CCA
- Limit 5 will give only 5 row of result



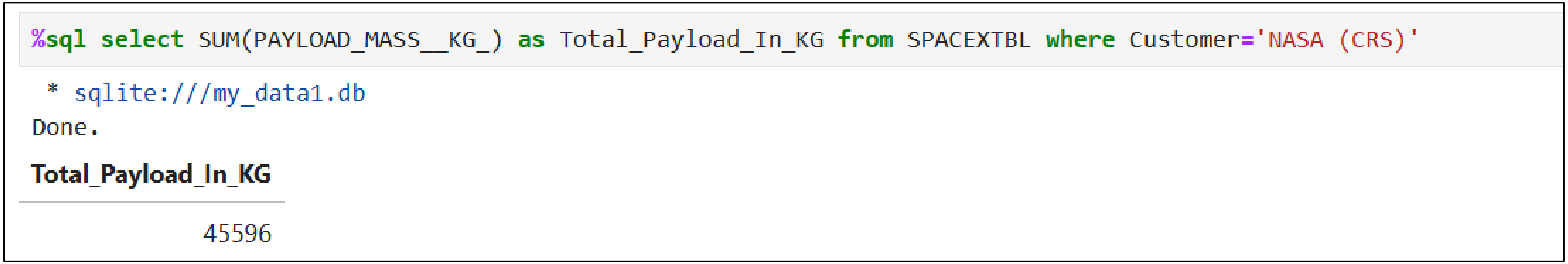
```
%sql select   * from SPACEXTBL where  Launch_site like 'CCA%' limit 5
```

```
 * 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

Total Payload Mass

- Calculate the total payload carried by boosters launched by NASA (CRS
- SUM is used to calculate total of a column
- Where condition is used to filter customer



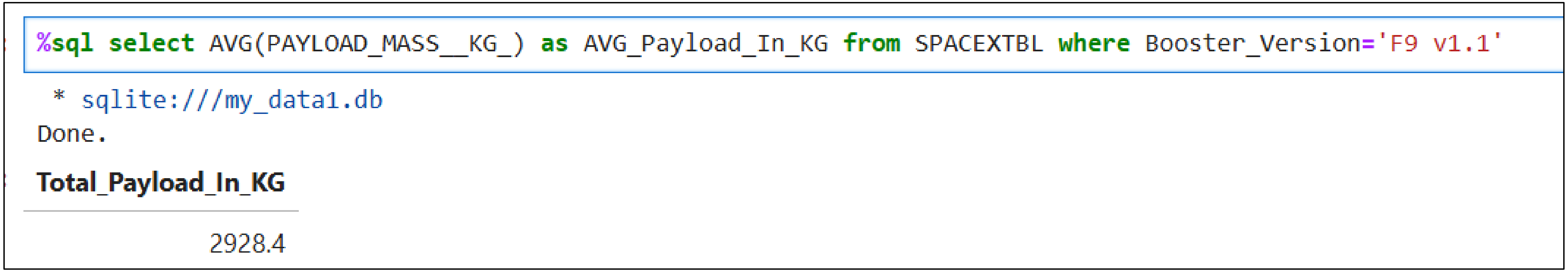
```
%sql select SUM(PAYLOAD_MASS__KG_) as Total_Payload_In_KG from SPACEXTBL where Customer='NASA (CRS)'
```

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

Total_Payload_In_KG
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- AVG is used to calculate average of a column
- Where condition is used to filter booster version



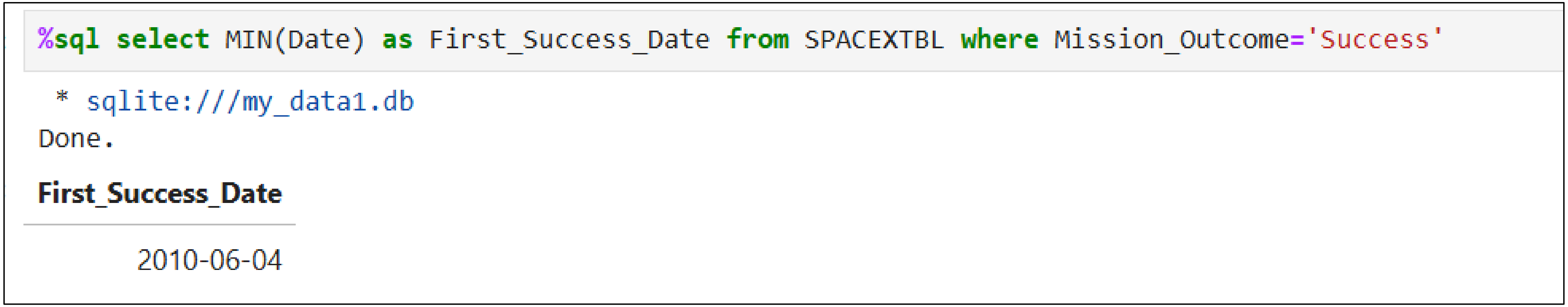
```
%sql select AVG(PAYLOAD_MASS__KG_) as AVG_Payload_In_KG from SPACEXTBL where Booster_Version='F9 v1.1'
```

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

Total_Payload_In_KG
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- MIN is used to calculate first date from the list.
- Where condition is used to filter success out come



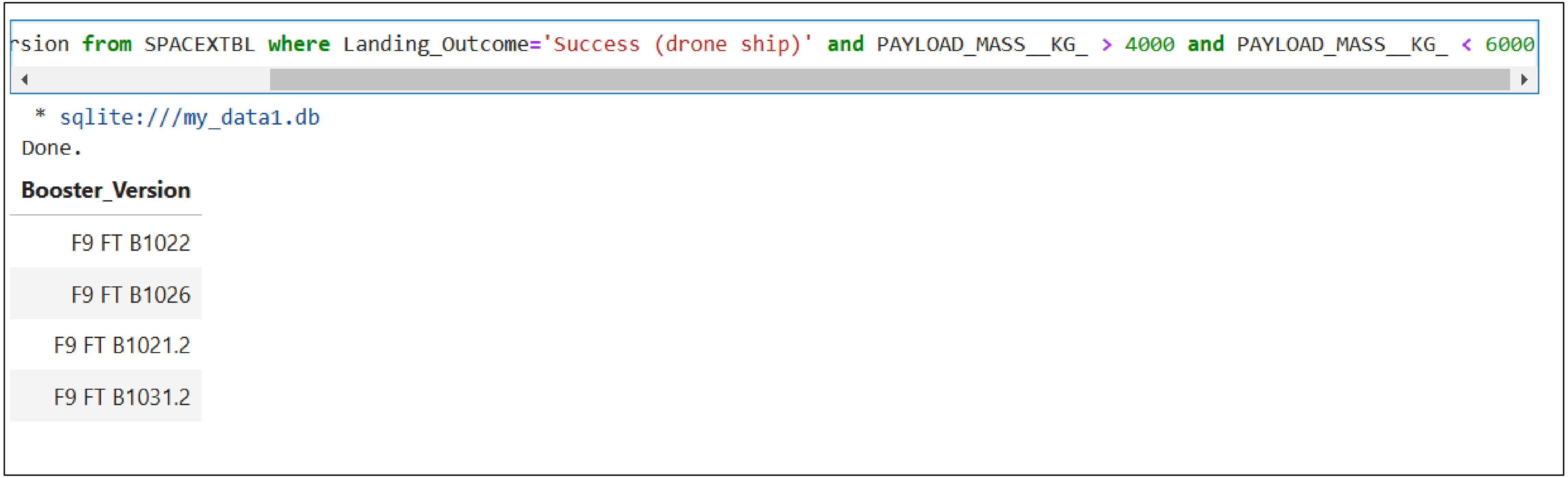
```
%sql select MIN(Date) as First_Success_Date from SPACEXTBL where Mission_Outcome='Success'
```

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

```
First_Success_Date
        2010-06-04
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



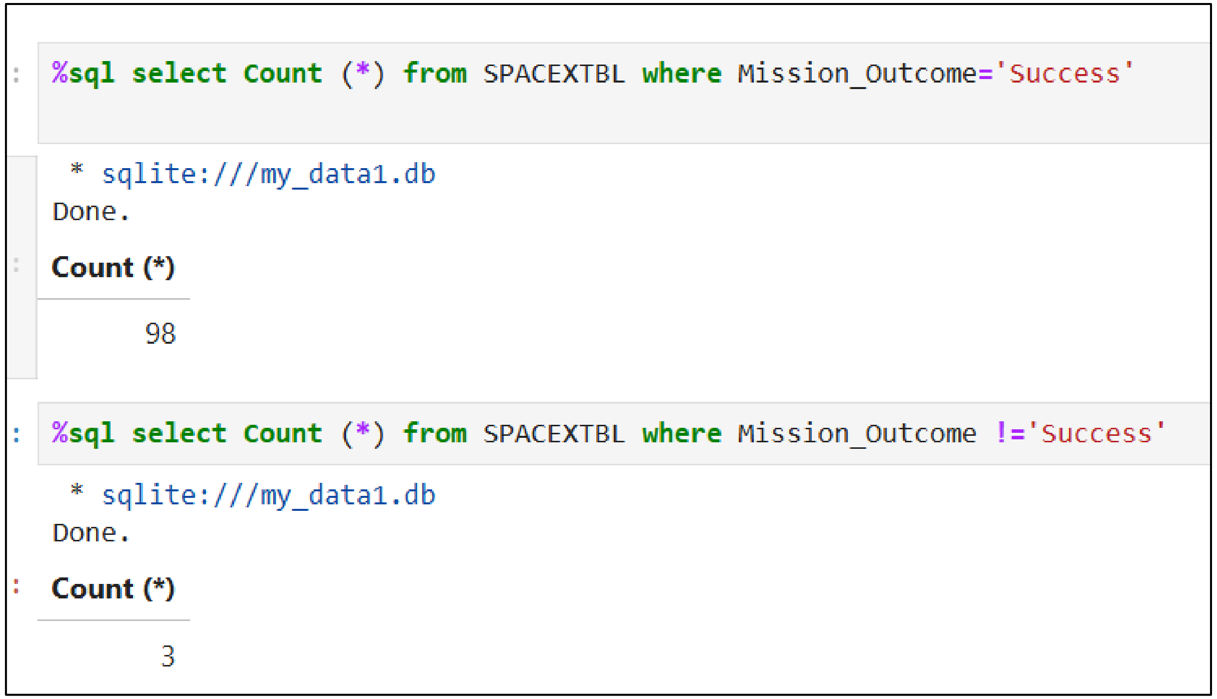
```
rsion from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Count is used to count number of Rows
- Condition is used equal and not equal as more ‘failure’ types are present

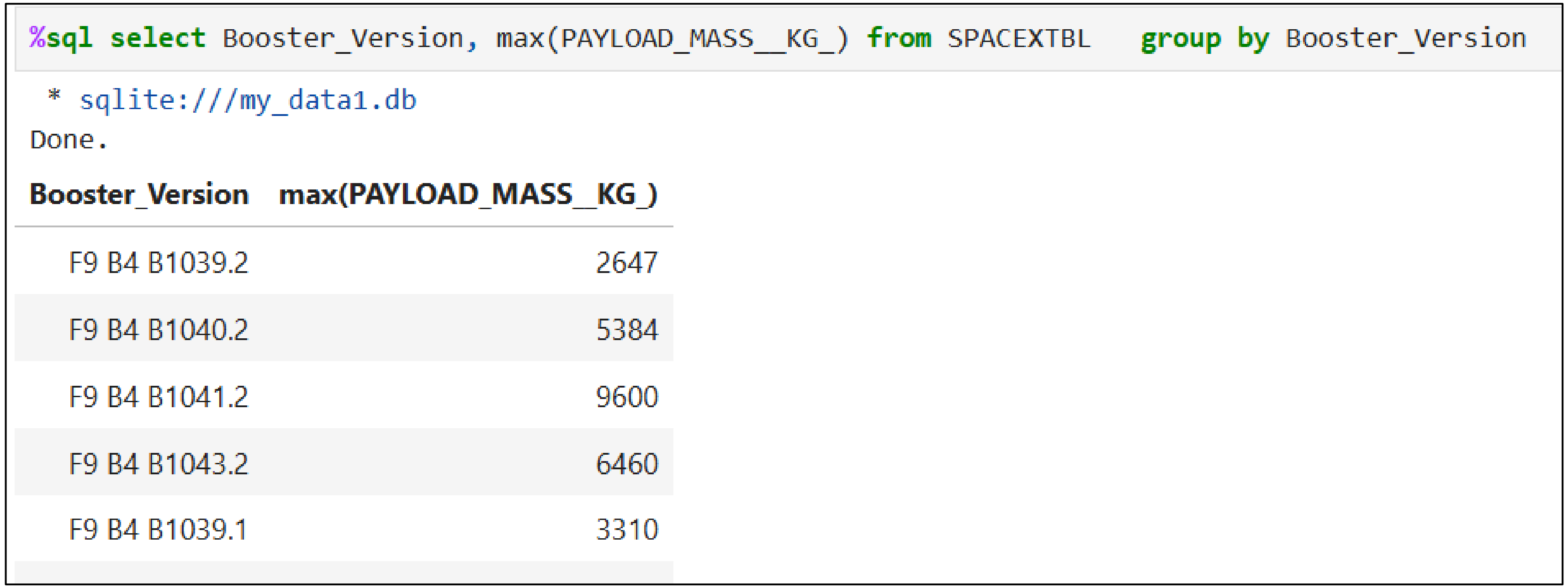


```
%sql select Count (*) from SPACEXTBL where Mission_Outcome='Success'

 * sqlite:///my_data1.db
Done.
Count (*)
      98
```

```
%sql select Count (*) from SPACEXTBL where Mission_Outcome !='Success'

 * sqlite:///my_data1.db
Done.
Count (*)
       3
```

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass



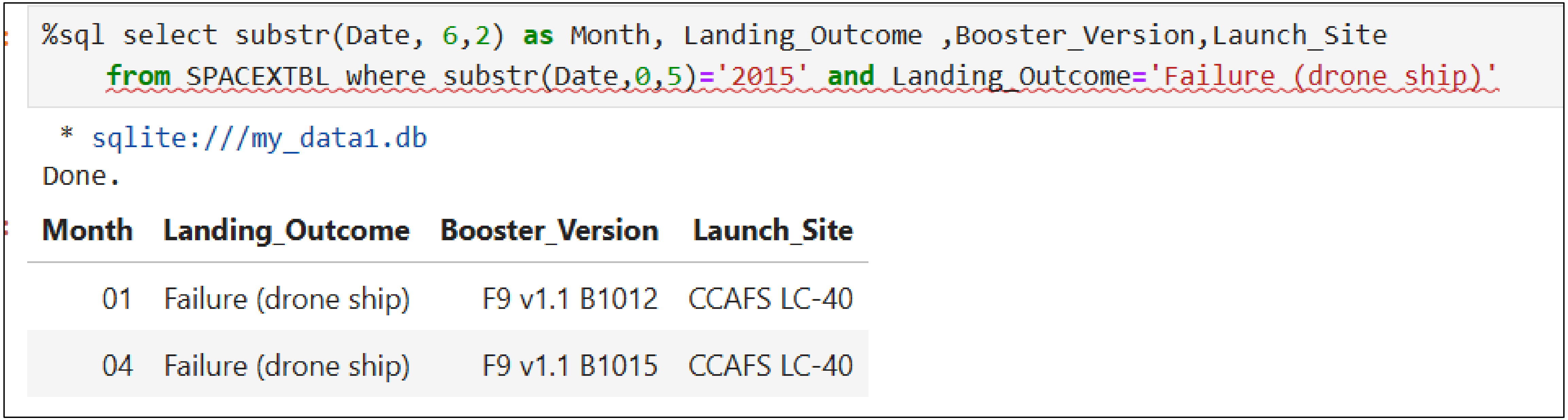
```
%sql select Booster_Version, max(PAYLOAD_MASS__KG_) from SPACEXTBL   group by Booster_Version
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B4 B1039.2	2647
F9 B4 B1040.2	5384
F9 B4 B1041.2	9600
F9 B4 B1043.2	6460
F9 B4 B1039.1	3310

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



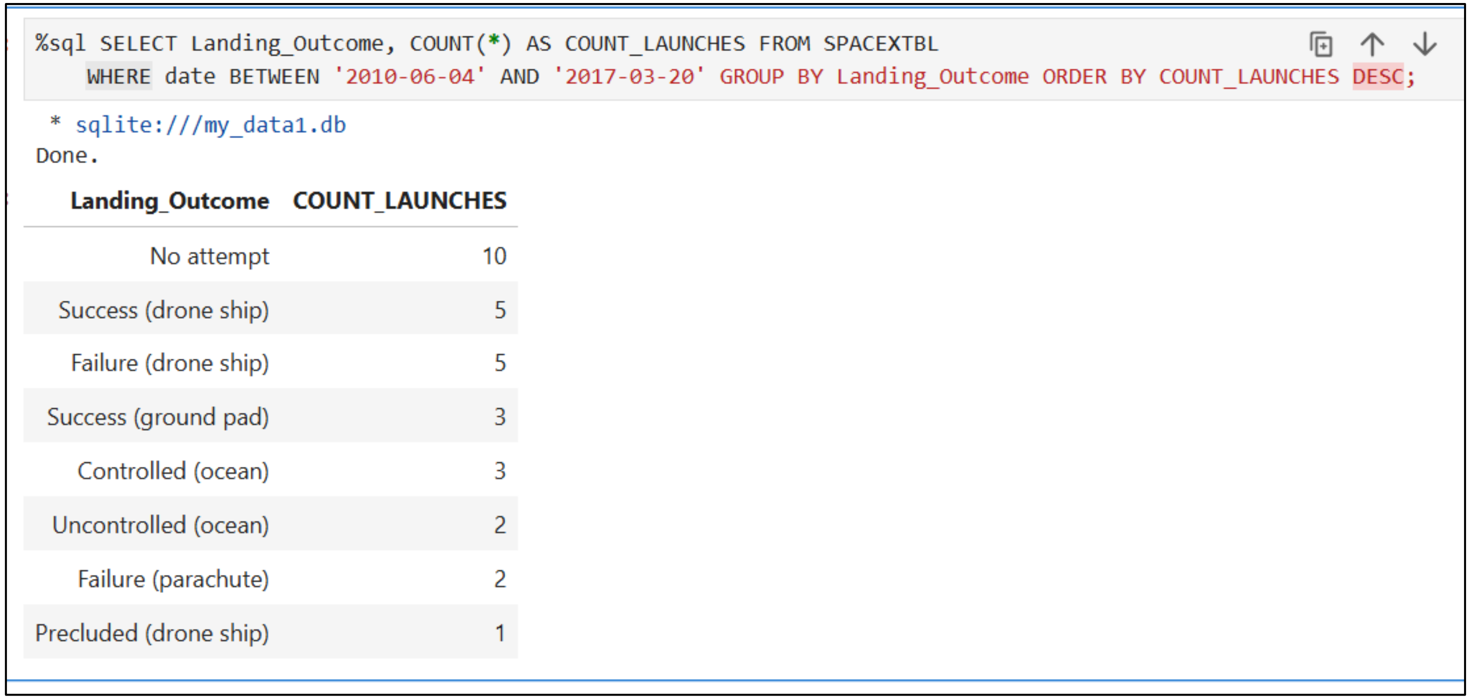
```
%sql select substr(Date, 6,2) as Month, Landing_Outcome ,Booster_Version,Launch_Site
    from SPACEXTBL where substr(Date,0,5)='2015' and Landing_Outcome='Failure (drone ship)'
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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
- Between condition is nused to check dates



```
%sql SELECT Landing_Outcome, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL
    WHERE date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT_LAUNCHES DESC;
```

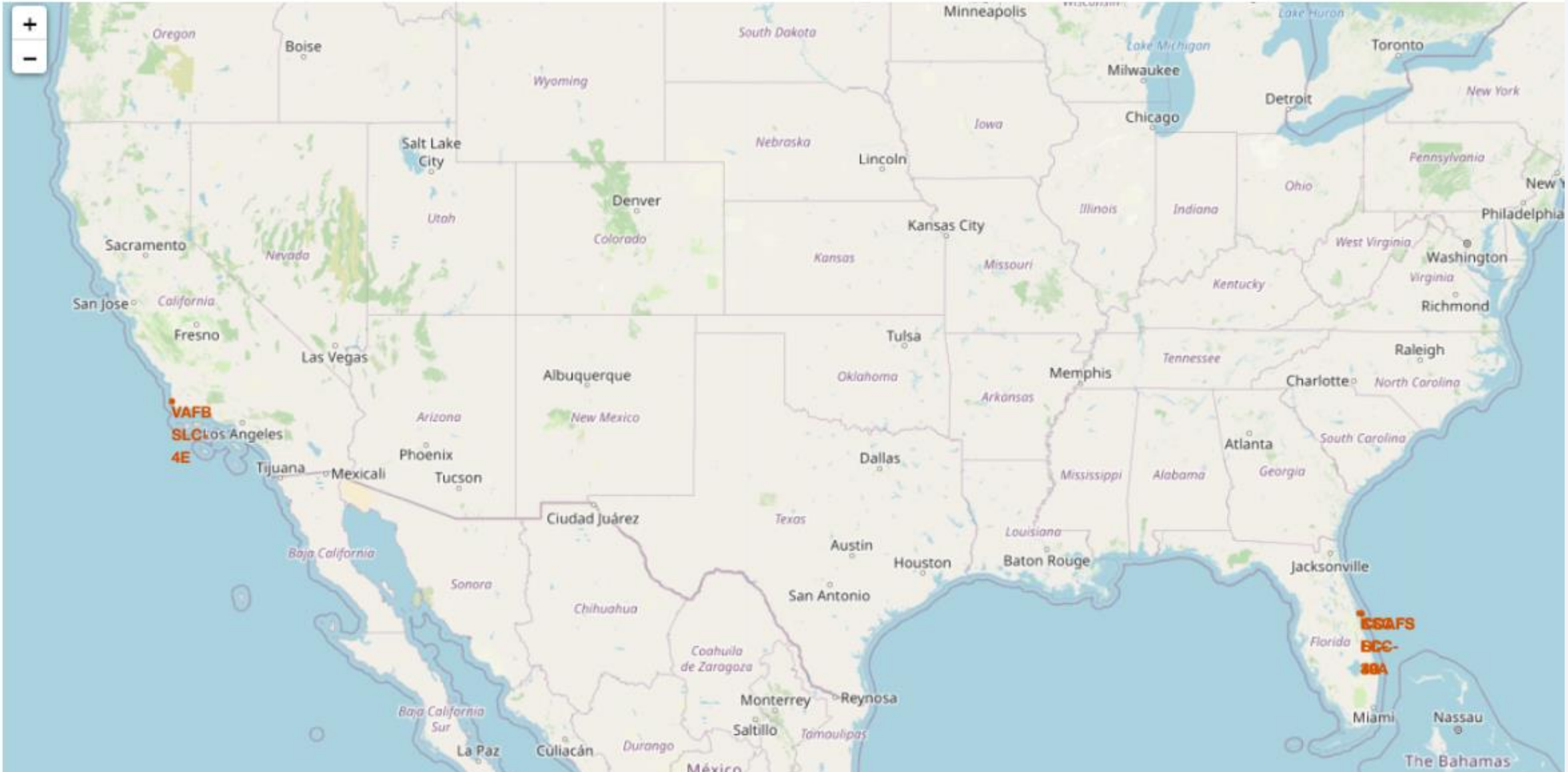
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT_LAUNCHES
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

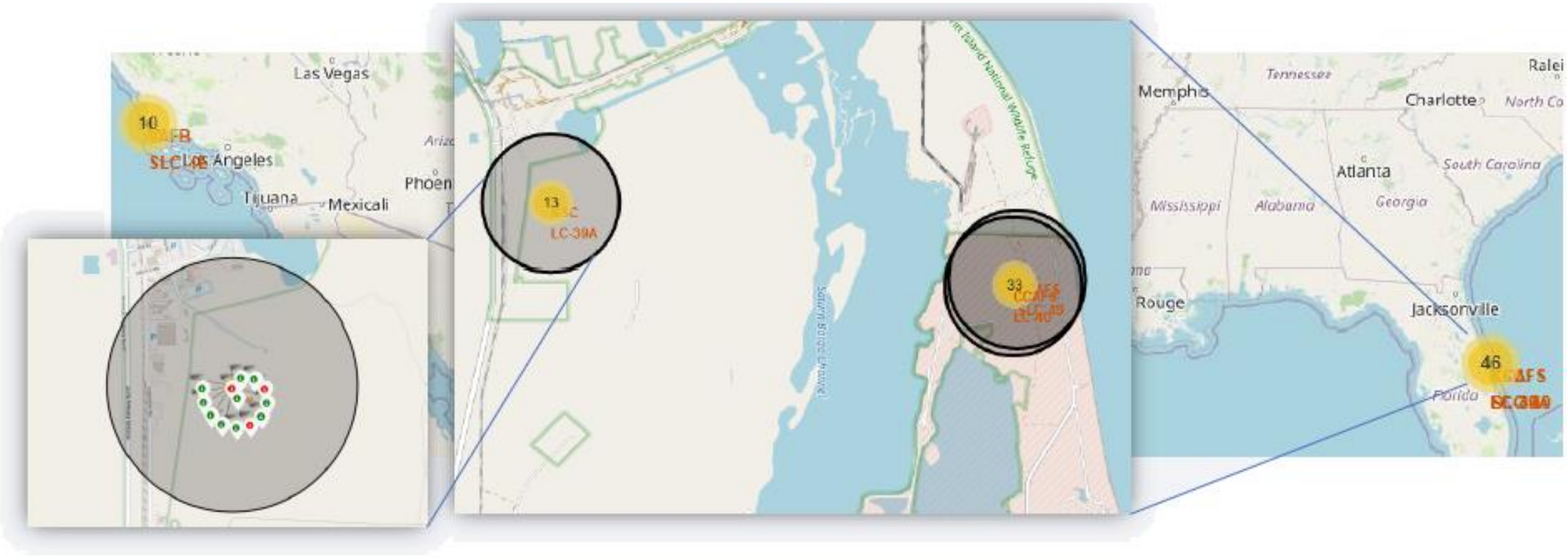
Launch Sites Proximities Analysis

All launch sites



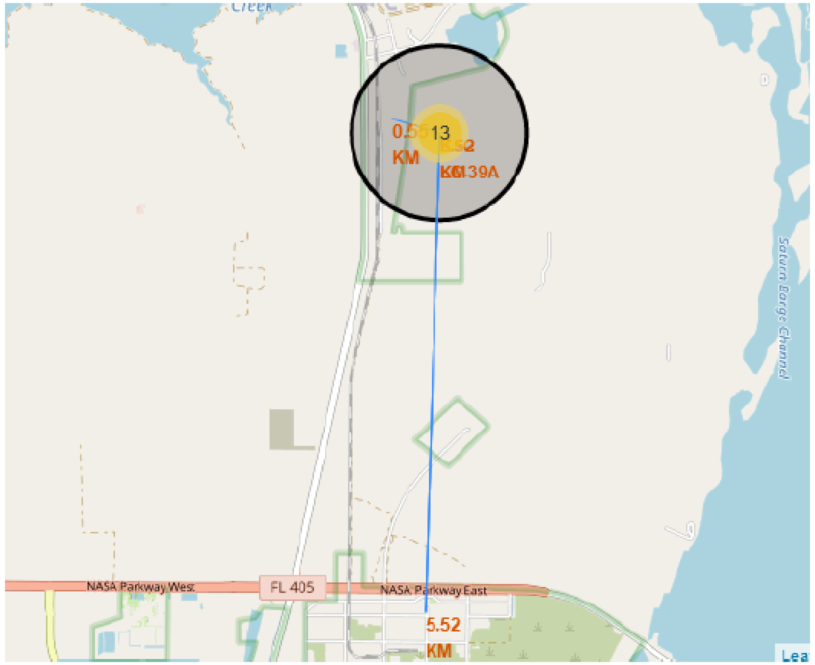
Launch Outcomes by Site

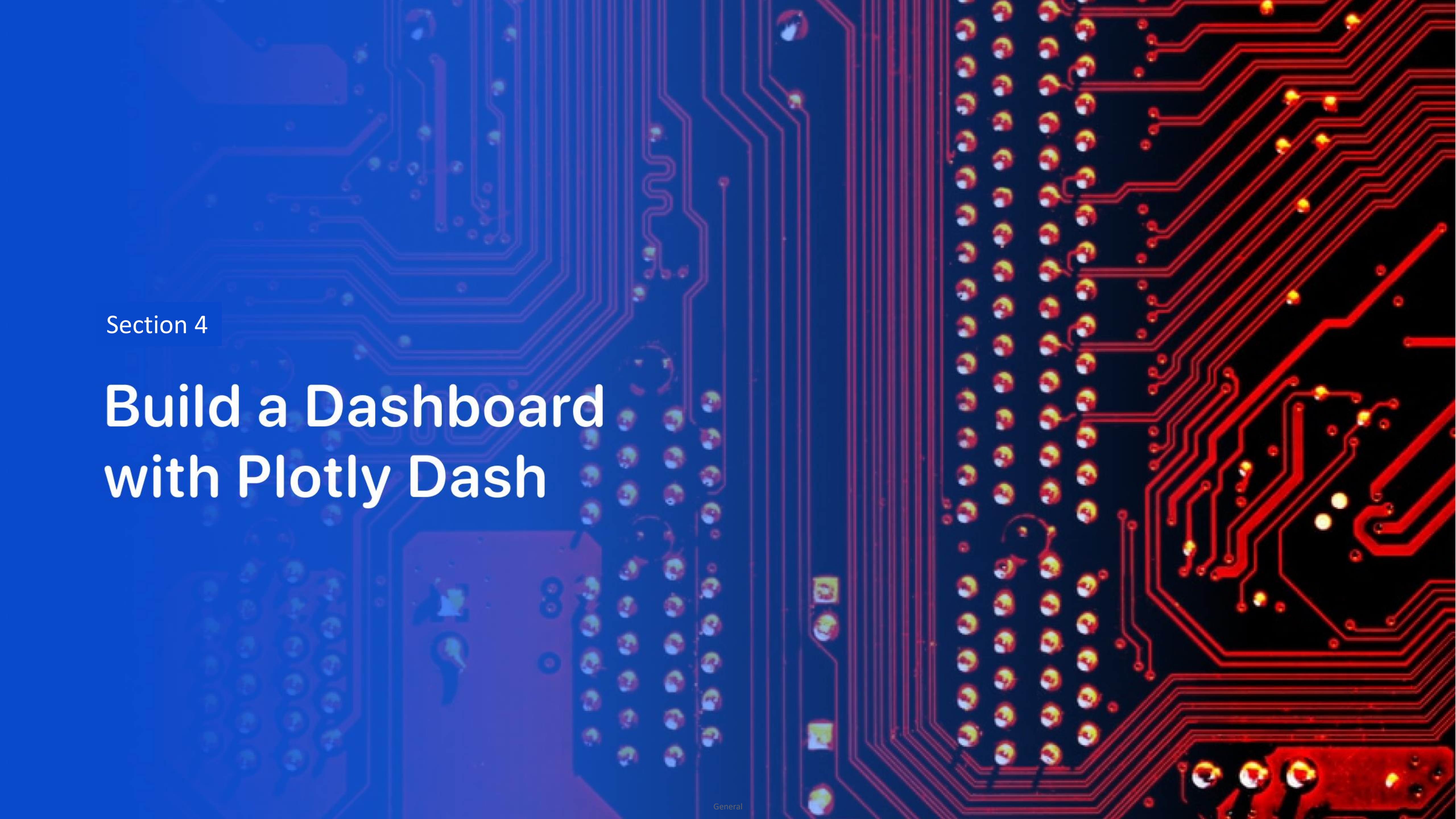
- Green markers indicate successful and red ones indicate failure.



Logistics and Safety

- Launch site KSC LC-39A has good logistics aspects, being near railroad and road and relatively far from inhabited areas.



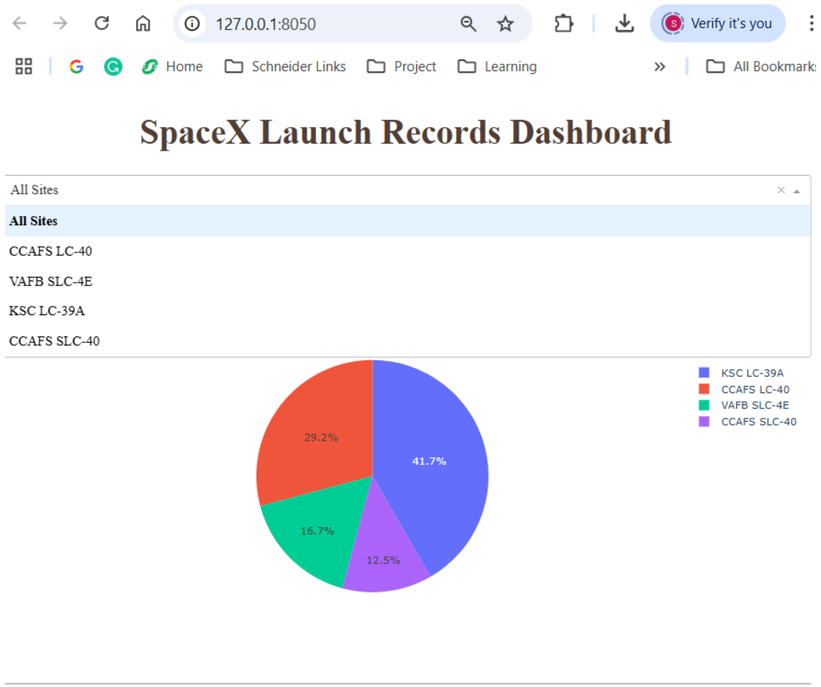


Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site

- The place from where launches are done seems to be a very important factor of success of missions.



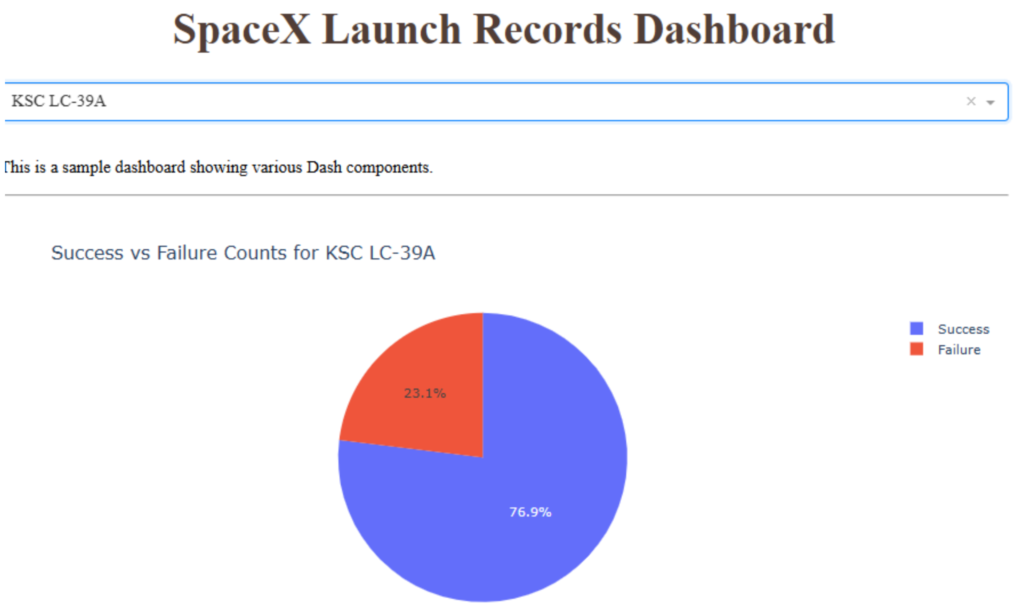
SpaceX Launch Records Dashboard

All Sites

All Sites
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Success Ratio for KSC LC-39A

- 76.9% of launches are successful in this site.



SpaceX Launch Records Dashboard

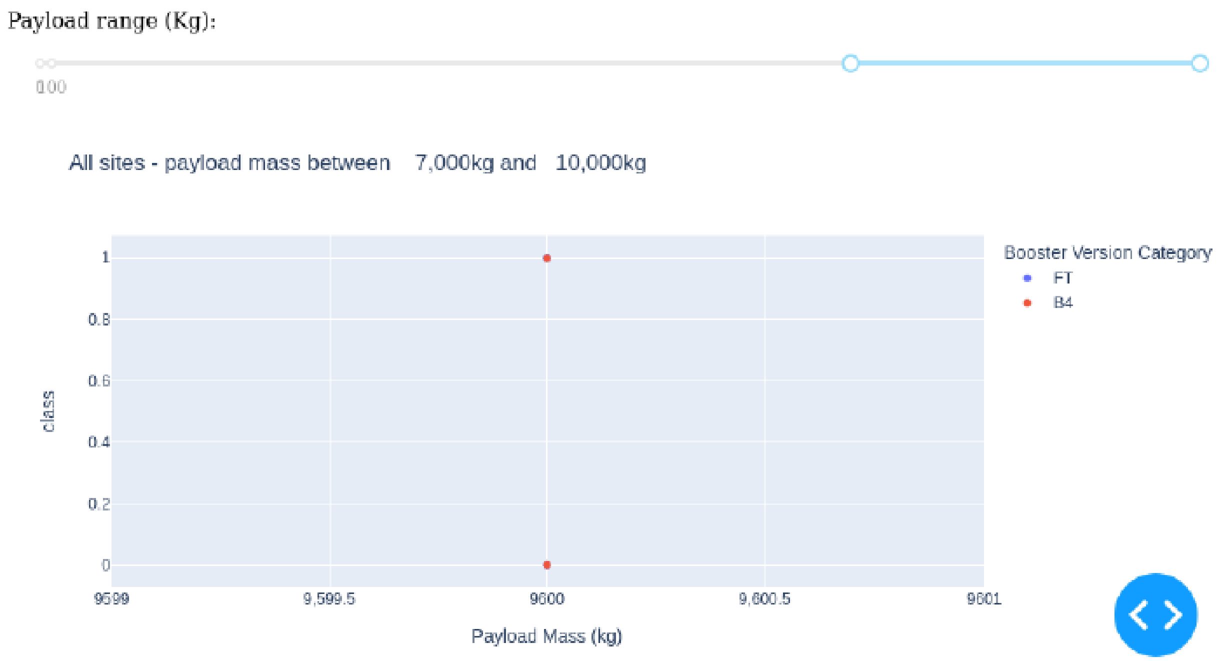
KSC LC-39A

This is a sample dashboard showing various Dash components.

Success vs Failure Counts for KSC LC-39A

Payload vs. Launch Outcome

- There’s not enough data to estimate risk of launches over 7,000kg



Payload range (Kg):

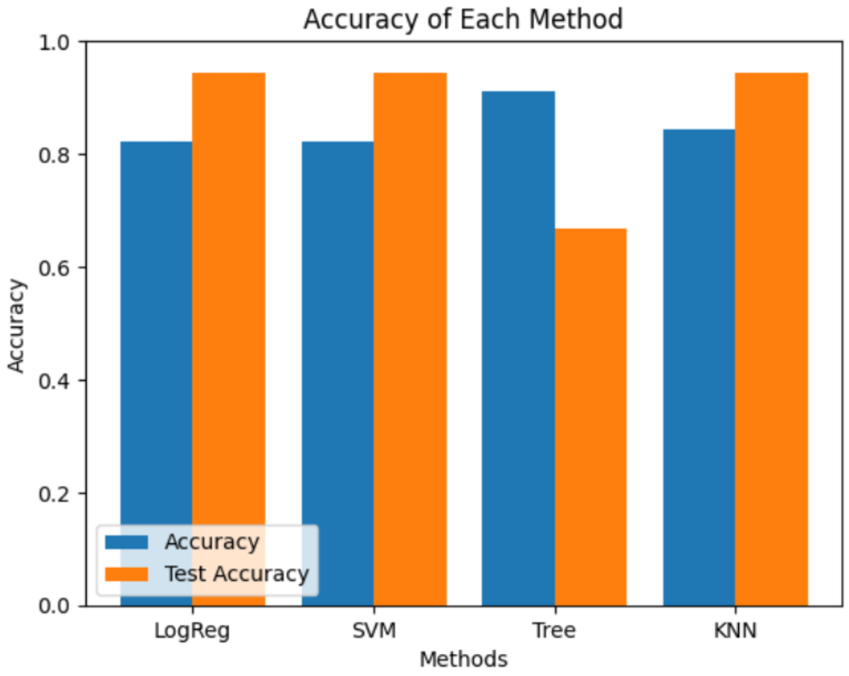
All sites - payload mass between 7,000kg and 10,000kg

Section 5

Predictive Analysis (Classification)

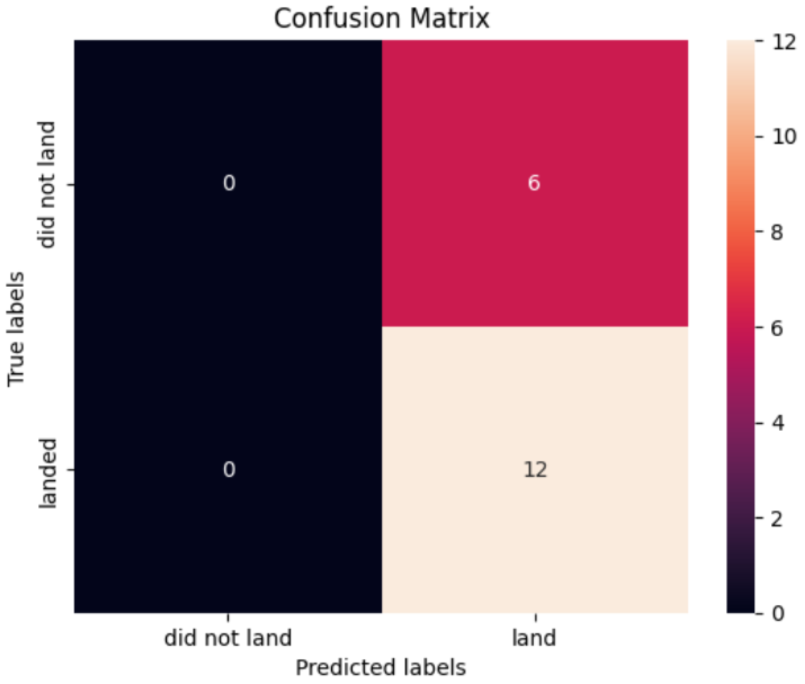
Classification Accuracy

- Four classification models were tested, and their accuracies are plotted beside;
- •The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies highest accuracy



Confusion Matrix

- Confusion matrix of Decision Tree Classifier proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.

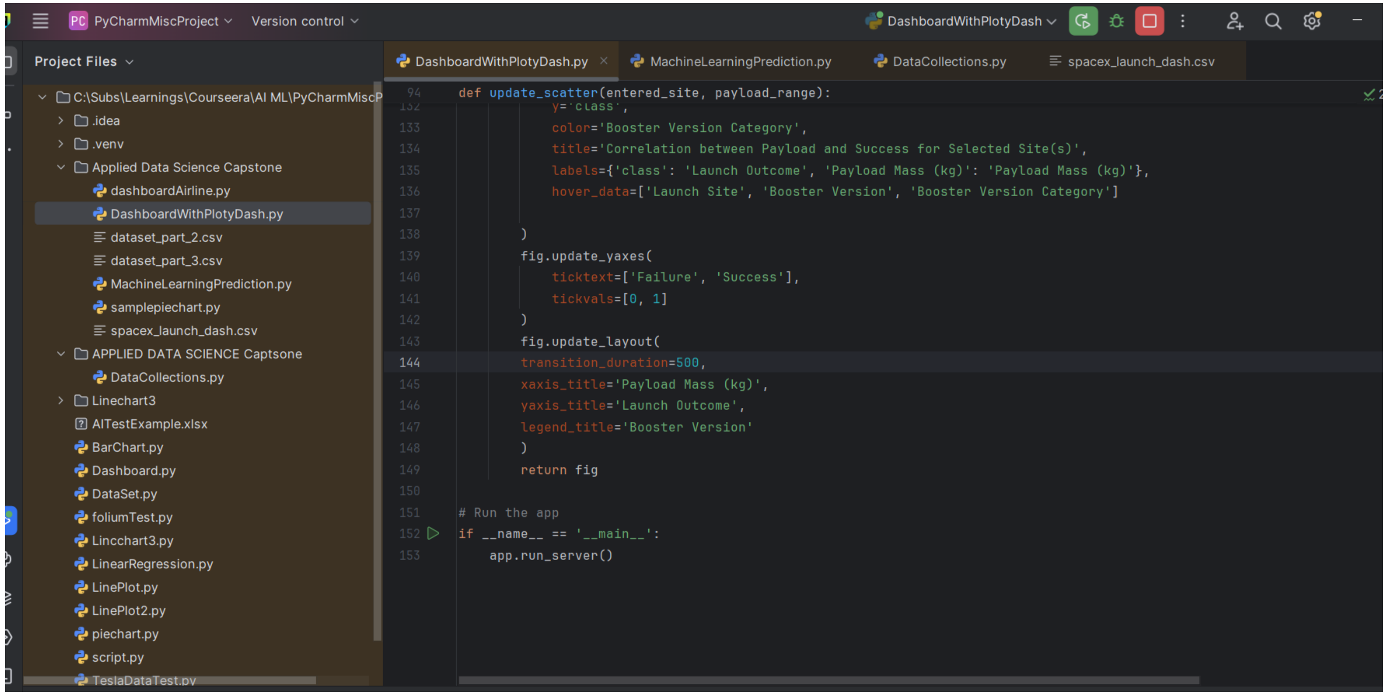


Conclusions

- Different data sources were analyzed, refining conclusions along the process;
- The best launch site is KSC LC-39A;
- Launches above 7,000kg are less risky;
- Although most of mission outcomes are successful, successful landing outcomes seem to improve over time, according the evolution of processes and rockets;
- Decision Tree Classifier can be used to predict successful landings and increase profits.

Appendix

- I have used PyCharm IDE for executing all python codes along with I have completed in Jupiter notebook as well



```
C:\Subs\Learnings\Courseera\AI ML\PyCharmMiscP
    .idea
    .venv
    Applied Data Science Capstone
        dashboardAirline.py
        DashboardWithPlotyDash.py
        dataset_part_2.csv
        dataset_part_3.csv
        MachineLearningPrediction.py
        samplepiechart.py
        spacex_launch_dash.csv
    APPLIED DATA SCIENCE Captsone
        DataCollections.py
    Linechart3
    AITestExample.xlsx
    BarChart.py
    Dashboard.py
    DataSet.py
    foliumTest.py
    Lincchart3.py
    LinearRegression.py
    LinePlot.py
    LinePlot2.py
    piechart.py
    script.py
    TeslaDataTest.py
```

```python
def update_scatter(entered_site, payload_range):
            y='class',
            color='Booster Version Category',
            title='Correlation between Payload and Success for Selected Site(s)',
            labels={'class': 'Launch Outcome', 'Payload Mass (kg)': 'Payload Mass (kg)'},
            hover_data=['Launch Site', 'Booster Version', 'Booster Version Category']

        )
        fig.update_yaxes(
            ticktext=['Failure', 'Success'],
            tickvals=[0, 1]
        )
        fig.update_layout(
        transition_duration=500,
        xaxis_title='Payload Mass (kg)',
        yaxis_title='Launch Outcome',
        legend_title='Booster Version'
        )
        return fig

# Run the app
if __name__ == '__main__':
    app.run_server()
```

Thank you!