IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Akinloye Magbagbeola
23/06/2022

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with plotly dash
  - Predictive analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive analysis
  - Predictive analysis results

# Introduction

- Project background and context
  - We predicted if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Problems you want to find answers
  - The main objective is to determine if the first stage of the SpaceX Falcon 9 rocket will land successfully.

Section 1

# Methodology

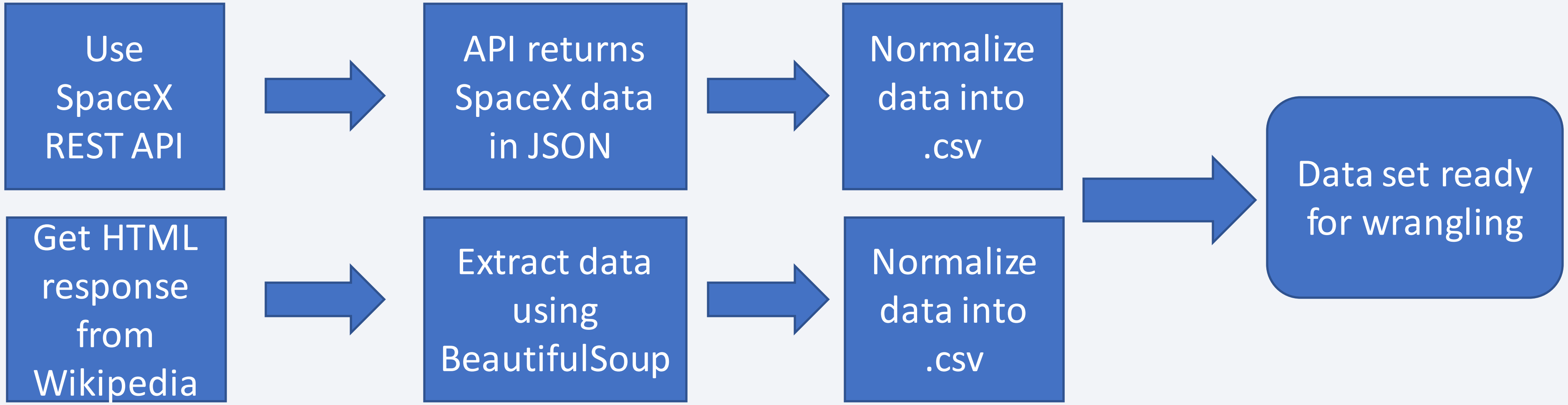
# Methodology

## Executive Summary

- Data collection methodology:
    - SpaceX Rest API
    - Web scrapping from Wikipedia
- Perform data wrangling
    - One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Linear Regression, K Nearest Neighbors, SVM and DT were built and evaluated for the best classifier

# Data Collection

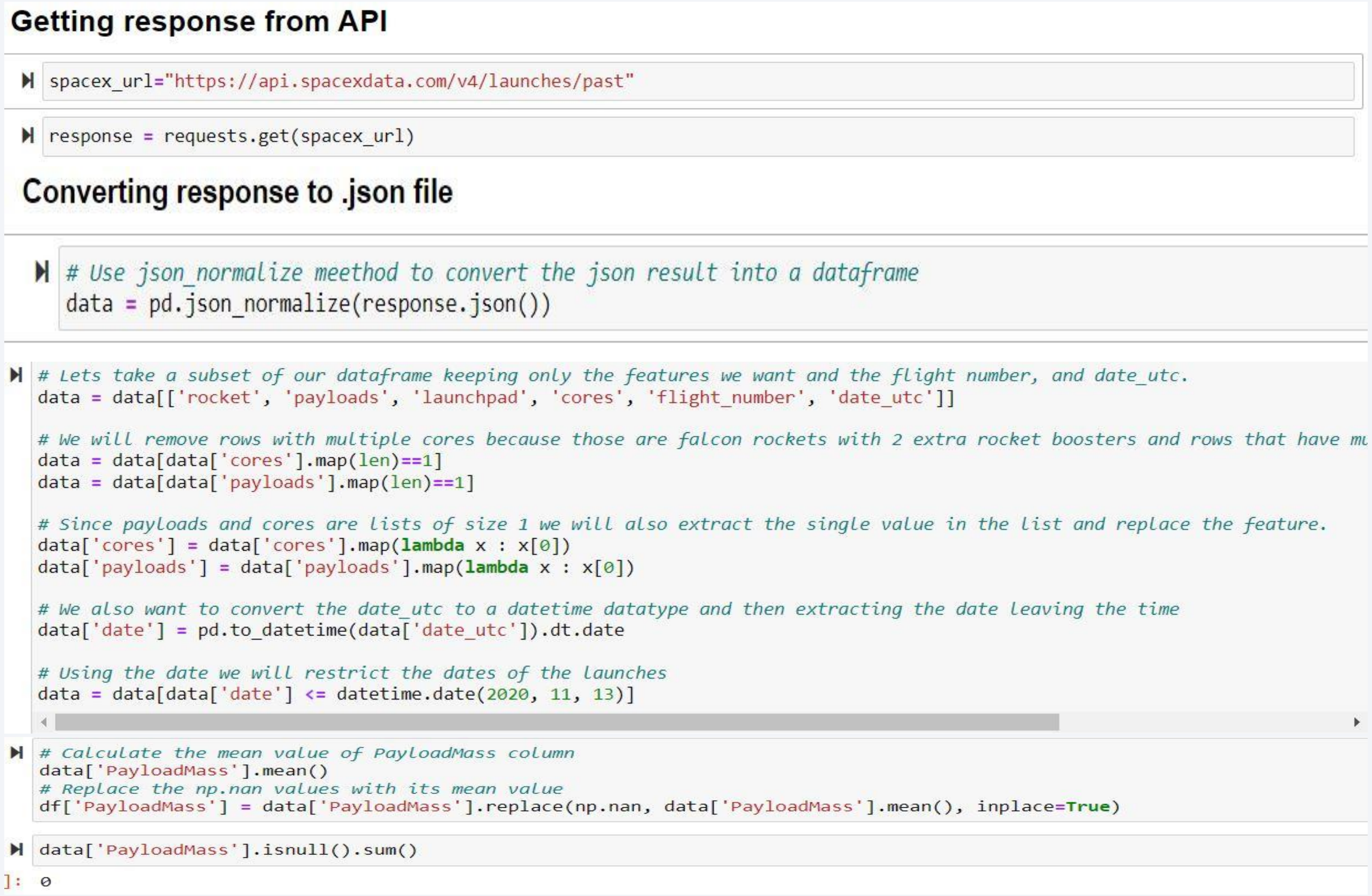
- SpaceX launch data was gathered from the SpaceX REST API
- The API gave us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications and landing outcome.
- The SpaceX REST API endpoints or URL starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scrapping Wikipedia using BeautifulSoup.


Use SpaceX REST API
API returns SpaceX data in JSON
Normalize data into .csv
Get HTML response from Wikipedia
Extract data using BeautifulSoup
Normalize data into .csv
Data set ready for wrangling

# Data Collection – SpaceX API

- SpaceX API was used to collect data

[https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb](https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb)



**Getting response from API**

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

**Converting response to .json file**

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have mu
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

```
# Calculate the mean value of PayloadMass column
data['PayloadMass'].mean()
# Replace the np.nan values with its mean value
df['PayloadMass'] = data['PayloadMass'].replace(np.nan, data['PayloadMass'].mean(), inplace=True)
```

```
data['PayloadMass'].isnull().sum()
```

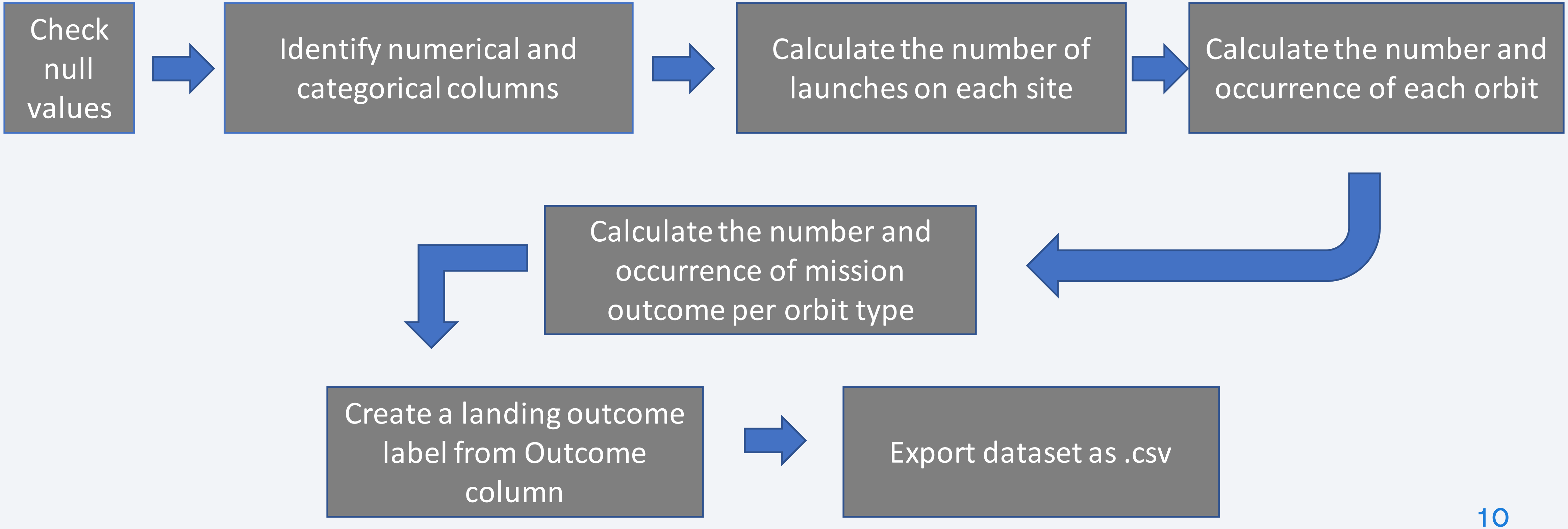
```
]: 0
```

# Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- I converted those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

# Data Wrangling Cont'd

## EDA Flow Chart



Check null values

Identify numerical and categorical columns

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Export dataset as .csv

# EDA with Data Visualization

- Scatter plot plotted are:
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Launch Site vs Payload Mass
  - Flight Number vs Orbit
  - Payload Mass vs Orbit

Scatter plot was used so that we could observe how one variable has affected the other.

- https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-dataviz.ipynb

- Bar chart was plotted to show the success of each orbit
- Line plot was plotted to show the success rate vs year. Line plots are useful because they show trends and it shows changes over a period of time.

# EDA with SQL

- SQL queries was performed on the dataset to gather some information:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in ground pad was achieved.
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Ranking the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
  - https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- All launch sites were marked and the longitudes and latitudes of each coordinates was used to create an interactive map while adding a circle marker around each launch site.
- The feature launch outcomes (failure and success) was assigned 0 and 1 respectively. 0 for failure, 1 for success.
- Green and red markers was used to identify launch sites with high success rate.
- Distance between two points on the map was calculated based on their latitude and longitude values to answer questions such as:
    - Proximity to railways
    - Proximity to highways.
    - Proximity to coastlines.
    - Do launch sites keep certain distance away from cities? The data revealed that they do

# Build a Dashboard with Plotly Dash

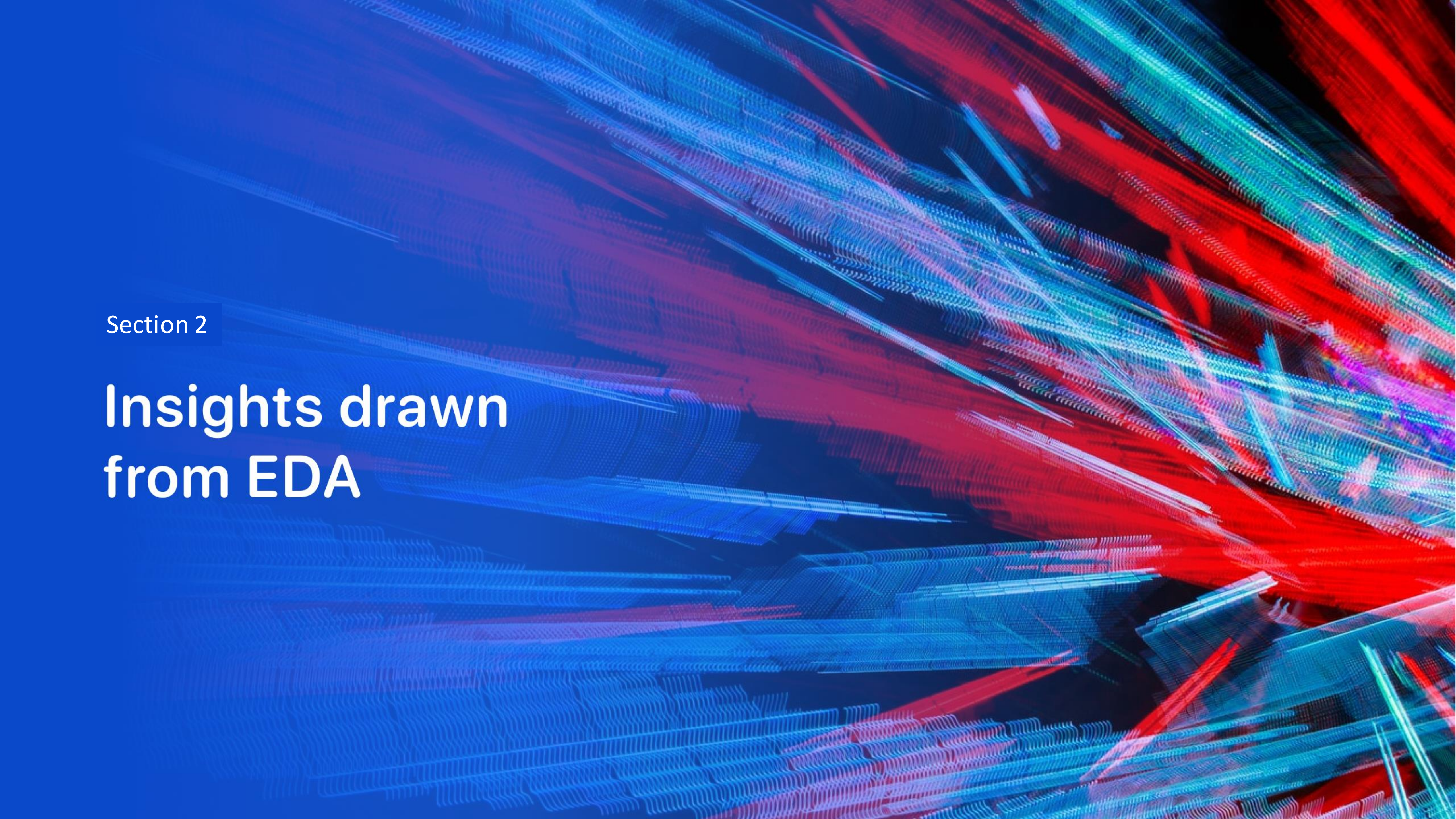
- An interactive dashboard was built with Plotly dash
- Total launches by certain sites was shown with pie charts
- Scatter graph showing the relationship with Outcome and Payload Mass (kg) for the different Booster Versions
- https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/Interactive%20map%20with%20folium.ipynb

# Predictive Analysis (Classification)

- Model Building
  - Loading of data using pandas and numpy
  - Data transform
  - Split data into train and test sets
  - GridSearchCV was used to tune the hyperparameters
  - Model was improved through feature engineering and algorithm tuning
  - Various classification methods such as (LR, SVM, DT, KNN) was used. Model with the best accuracy score wins the best performing model.
  - https://github.com/loyemagba/IBM-Data-Science-Capstone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

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

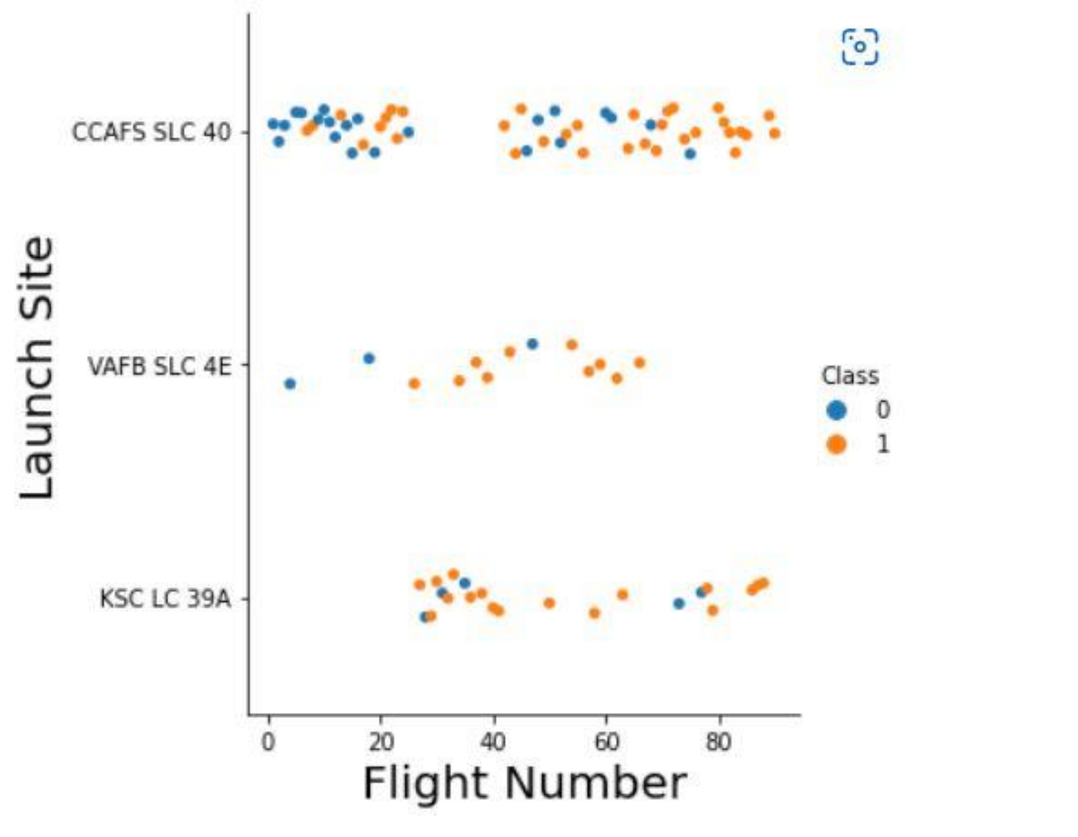


Section 2

# Insights drawn from EDA

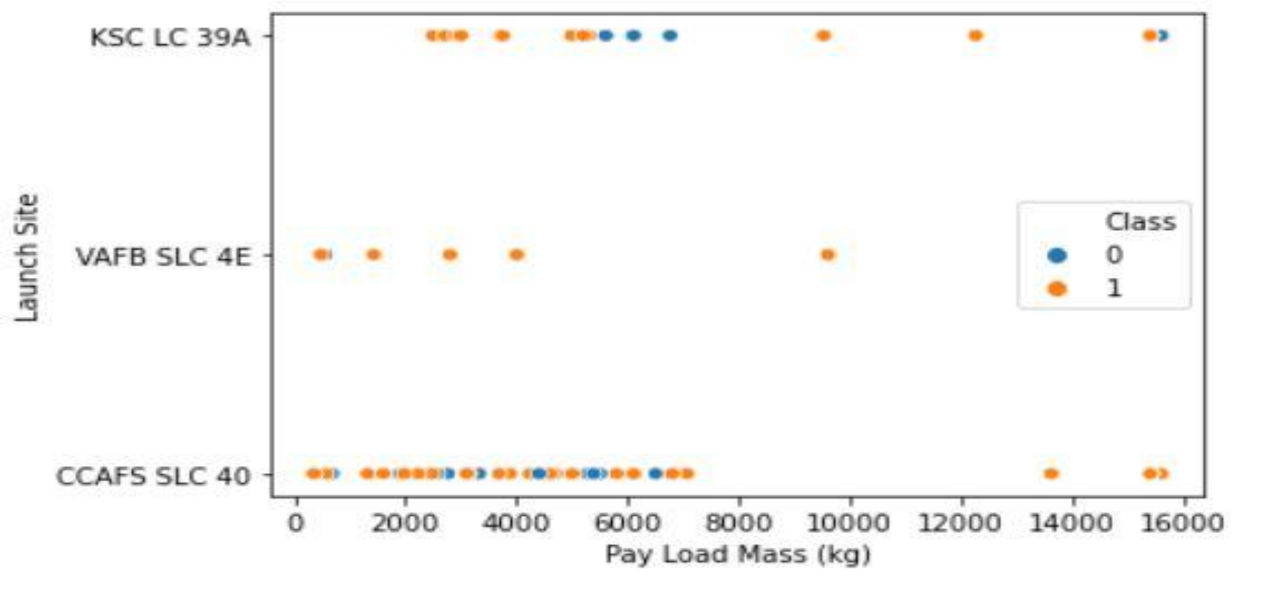
# Flight Number vs. Launch Site

- Scatter plot showing the relationship between Flight Number and Launch Site
- The scatter plot revealed that the larger the flight amount at a launch site, the greater the success rate.



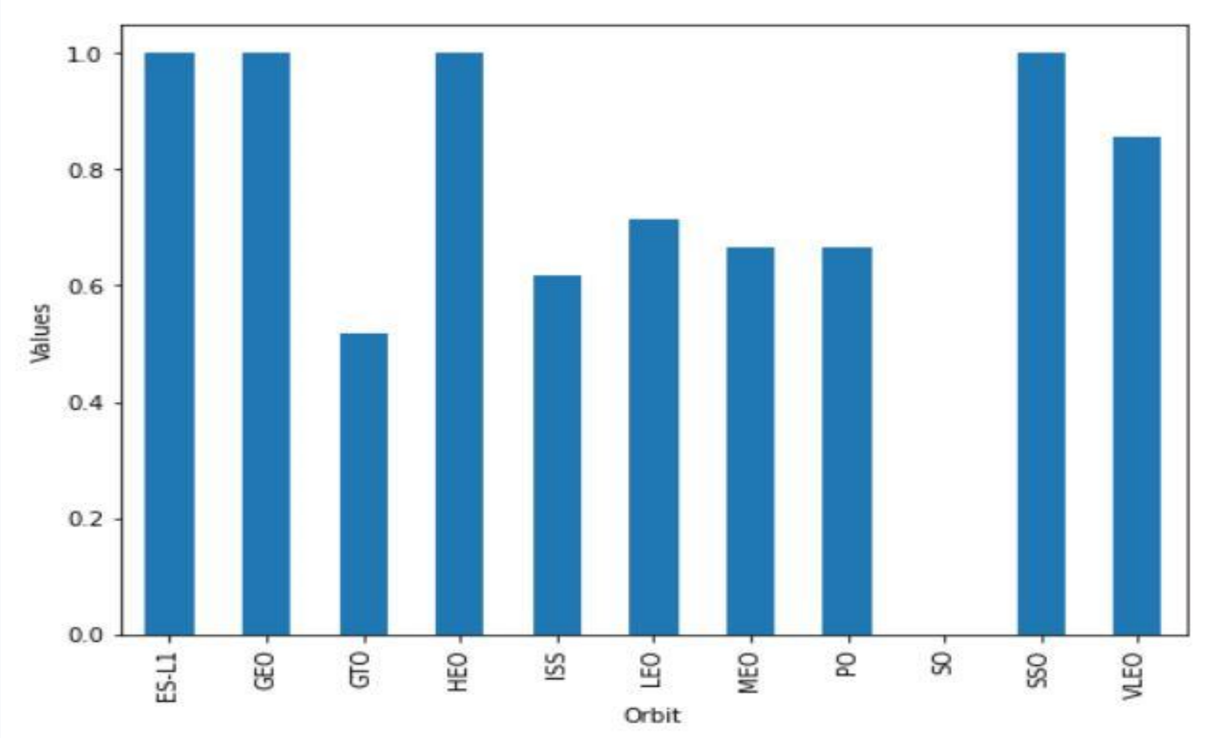
# Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- For CCAFS SLC 40, the greater the payload mass, the higher the success rate
- VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).



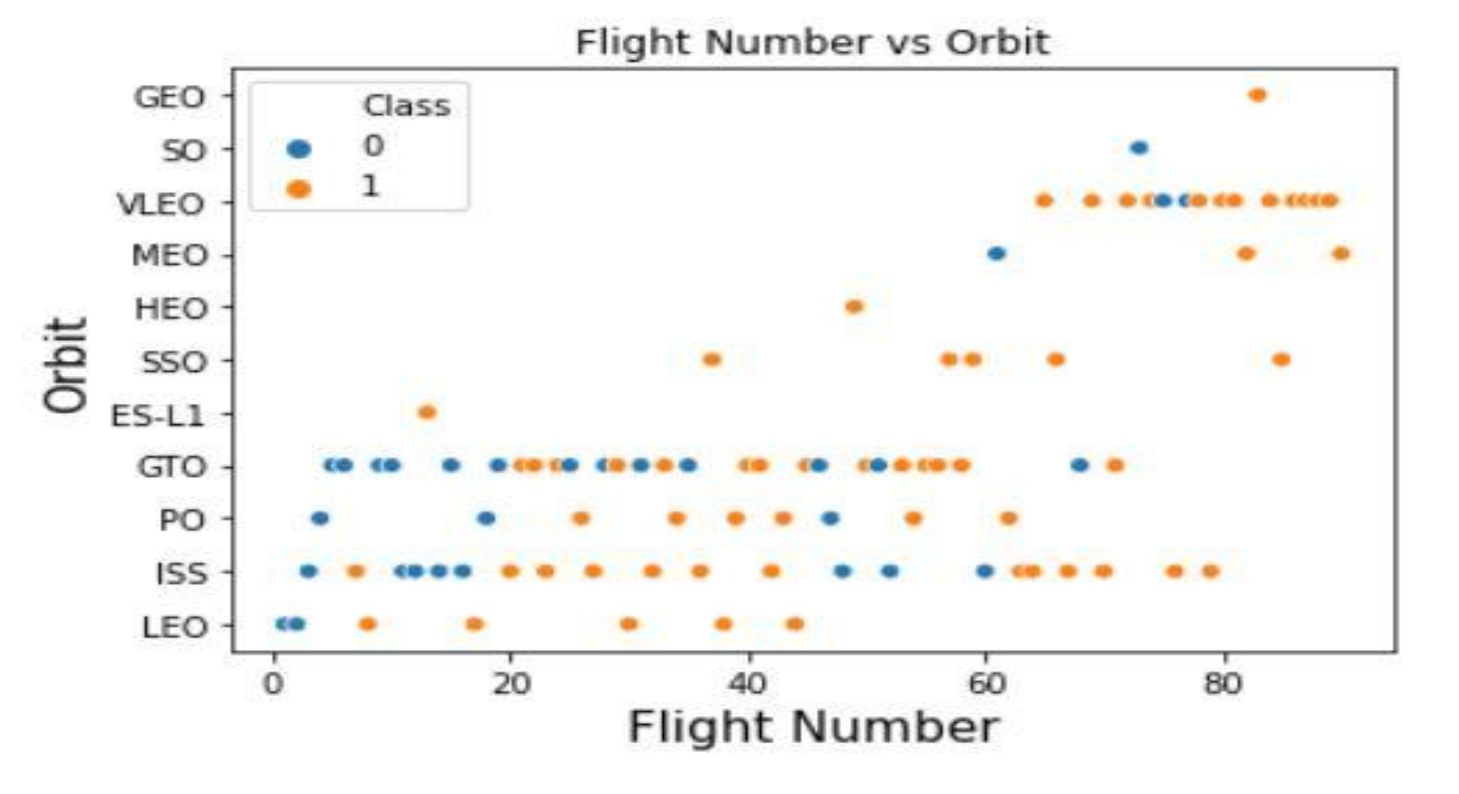
# Success Rate vs. Orbit Type

- Bar chart showing success rate vs orbit
- Orbits with the highest success rates are ES-L1, GEO, HEO and SSO. GTO has the lowest success rate amongst the orbits



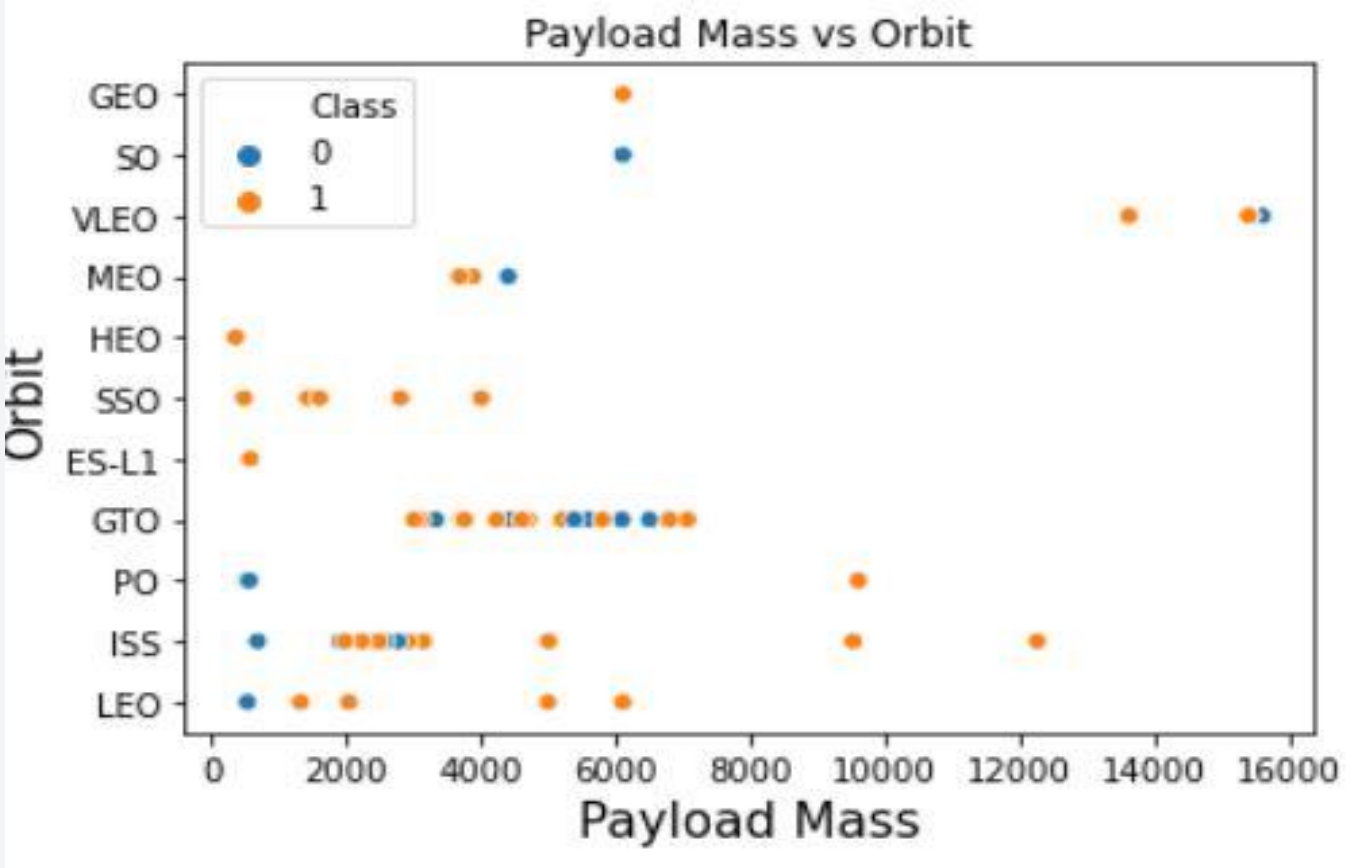
# Flight Number vs. Orbit Type

- There seems to be no relationship between flight number in GTO while in LEO, the higher the number of flights, the higher the success rate



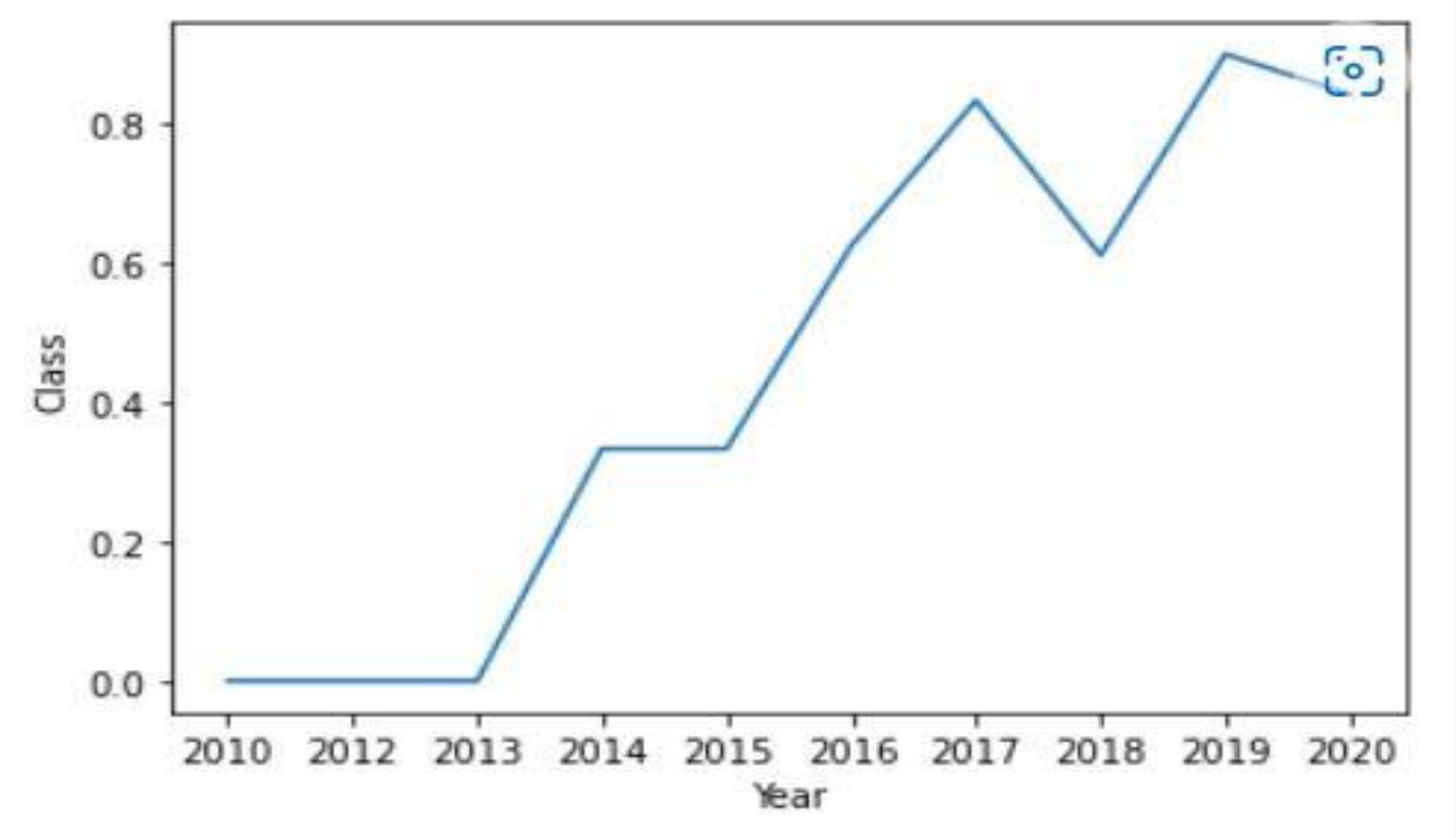
# Payload vs. Orbit Type

- Heavy payload have a higher success in ISS and LEO



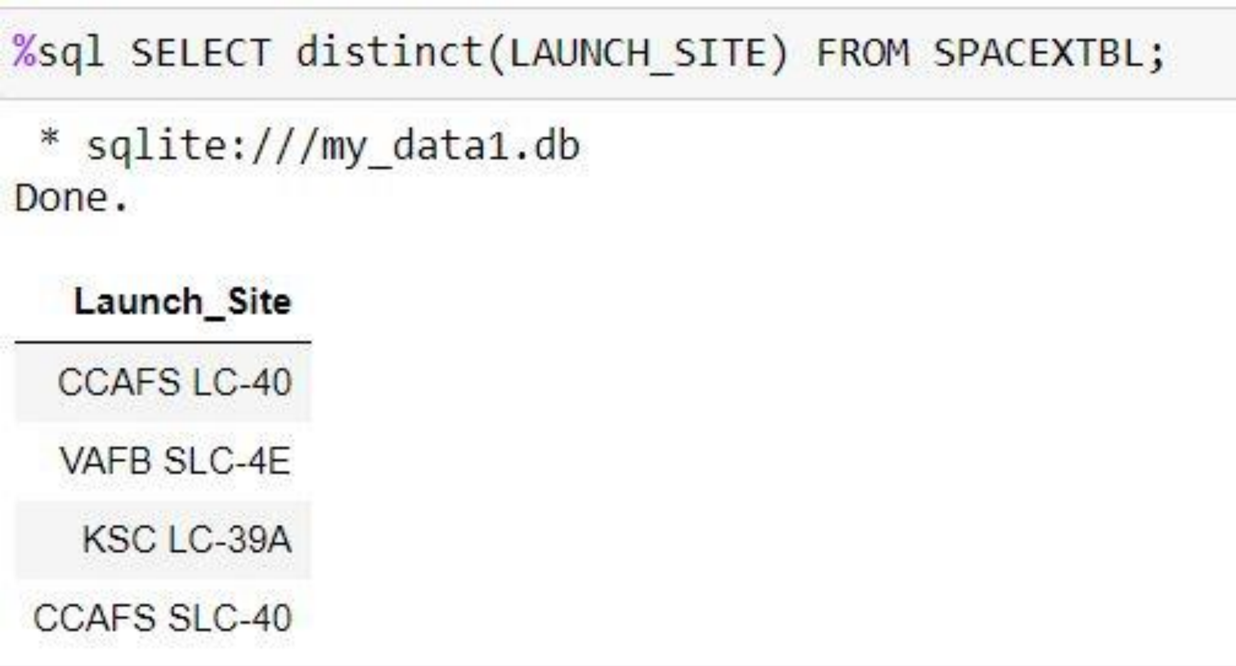
# Launch Success Yearly Trend

- We can observe that the sucess rate since 2013 kept increasing till 2020



# All Launch Site Names

- Using DISTINCT in the query means it only showed the unique values in launch site column from spacextbl data



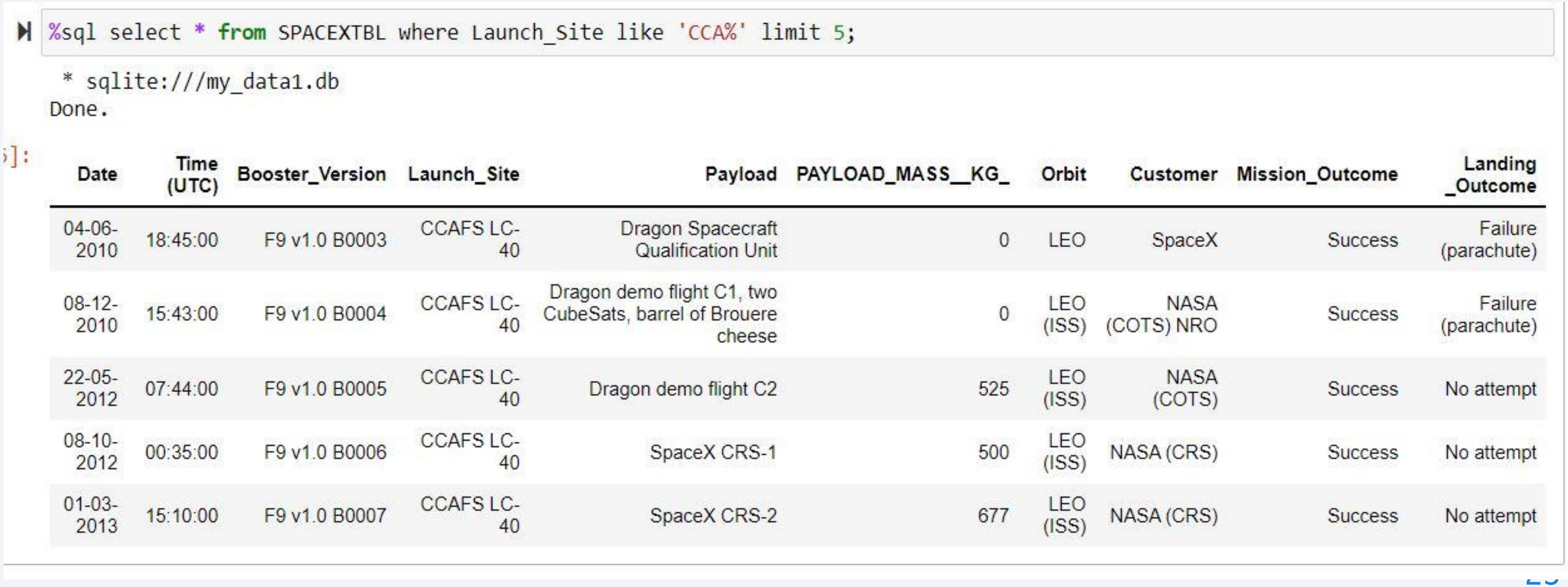
```
%sql SELECT distinct(LAUNCH_SITE) FROM SPACEXTBL;
```

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

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

# Launch Site Names Begin with 'CCA'

- The query in the code cell was used to display the table below



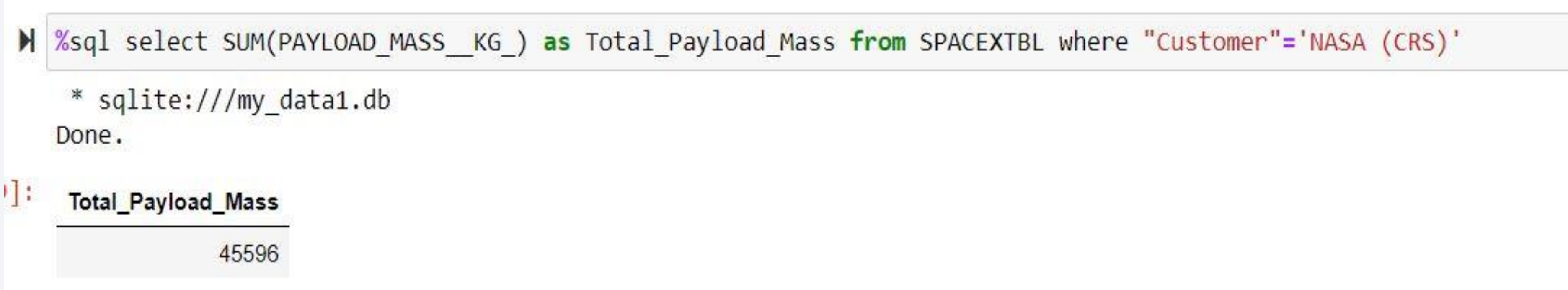
```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5;
```

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

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- The code in the cell was used to calculate the total payload mass



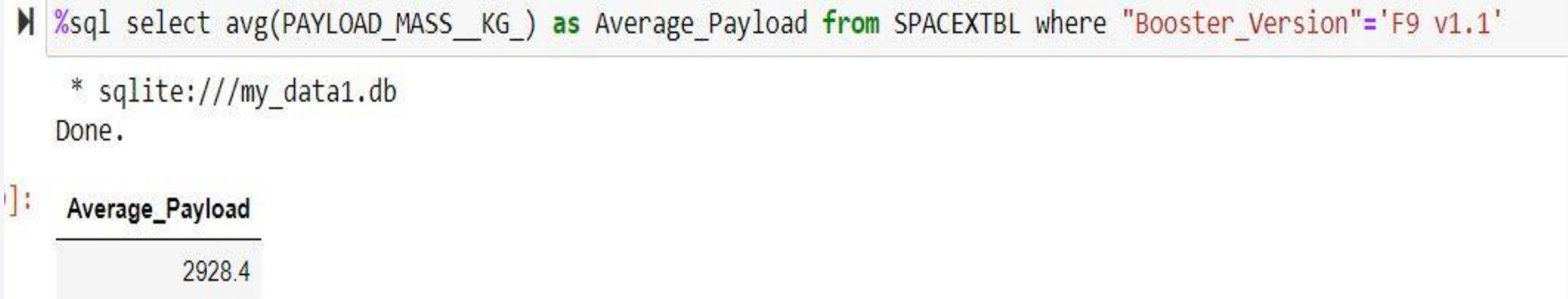
```
%sql select SUM(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTBL where "Customer"='NASA (CRS)'
```

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

| Total_Payload_Mass |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- The average payload mass is 2928.4



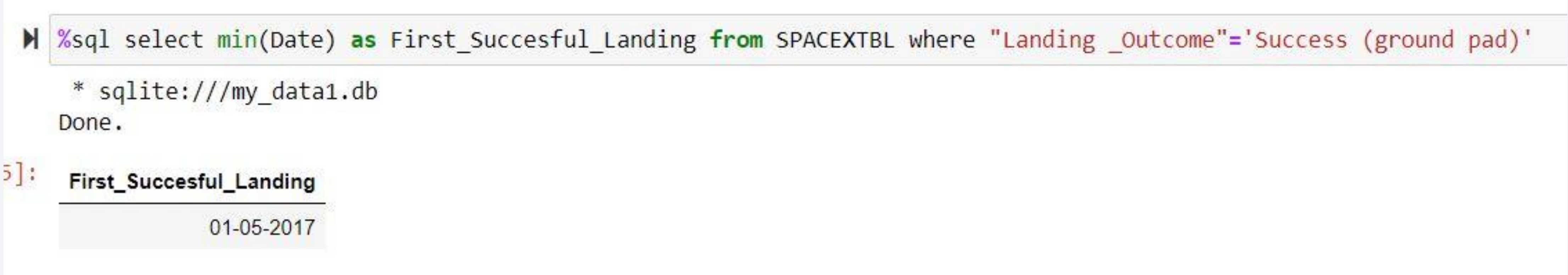
```
%sql select avg(PAYLOAD_MASS__KG_) as Average_Payload from SPACEXTBL where "Booster_Version"='F9 v1.1'

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

| Average_Payload |
|---|
| 2928.4 |

# First Successful Ground Landing Date

- The first successful ground landing date was 01-05-2017



```
%sql select min(Date) as First_Succesful_Landing from SPACEXTBL where "Landing _Outcome"='Success (ground pad)'
```

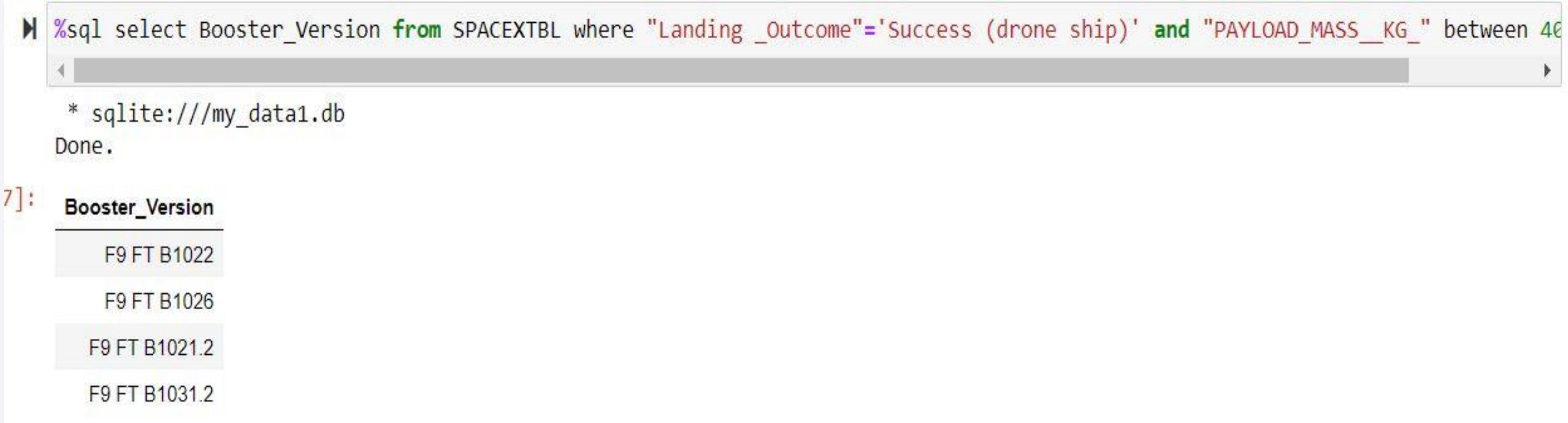
```
 * sqlite:///my_data1.db
Done.
```

5]:

| First_Succesful_Landing |
|---|
| 01-05-2017 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- The WHERE clause was used to filter for boosters which have successfully landed on drone ship as well as the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000



```
%sql select Booster_Version from SPACEXTBL where "Landing _Outcome"='Success (drone ship)' and "PAYLOAD_MASS__KG_" between 40
```

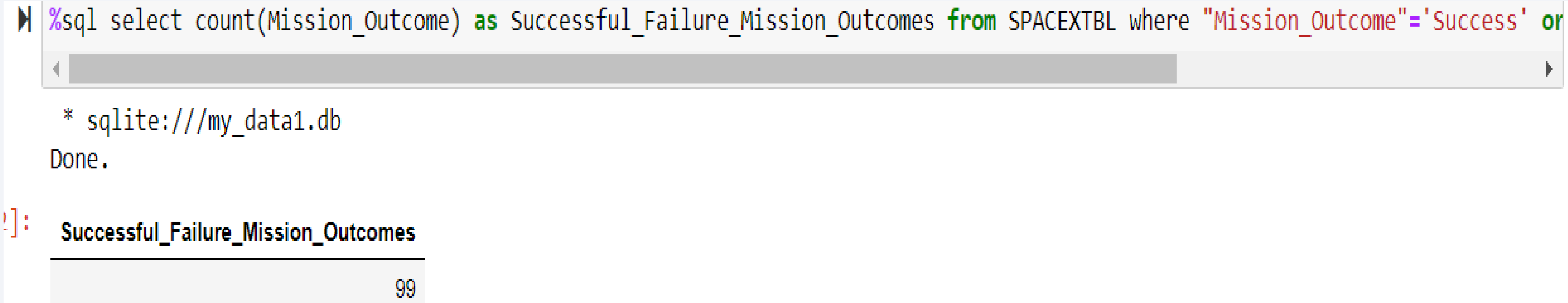
```
 * sqlite:///my_data1.db
Done.
```

7]:

| 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
- Present your query result with a short explanation here



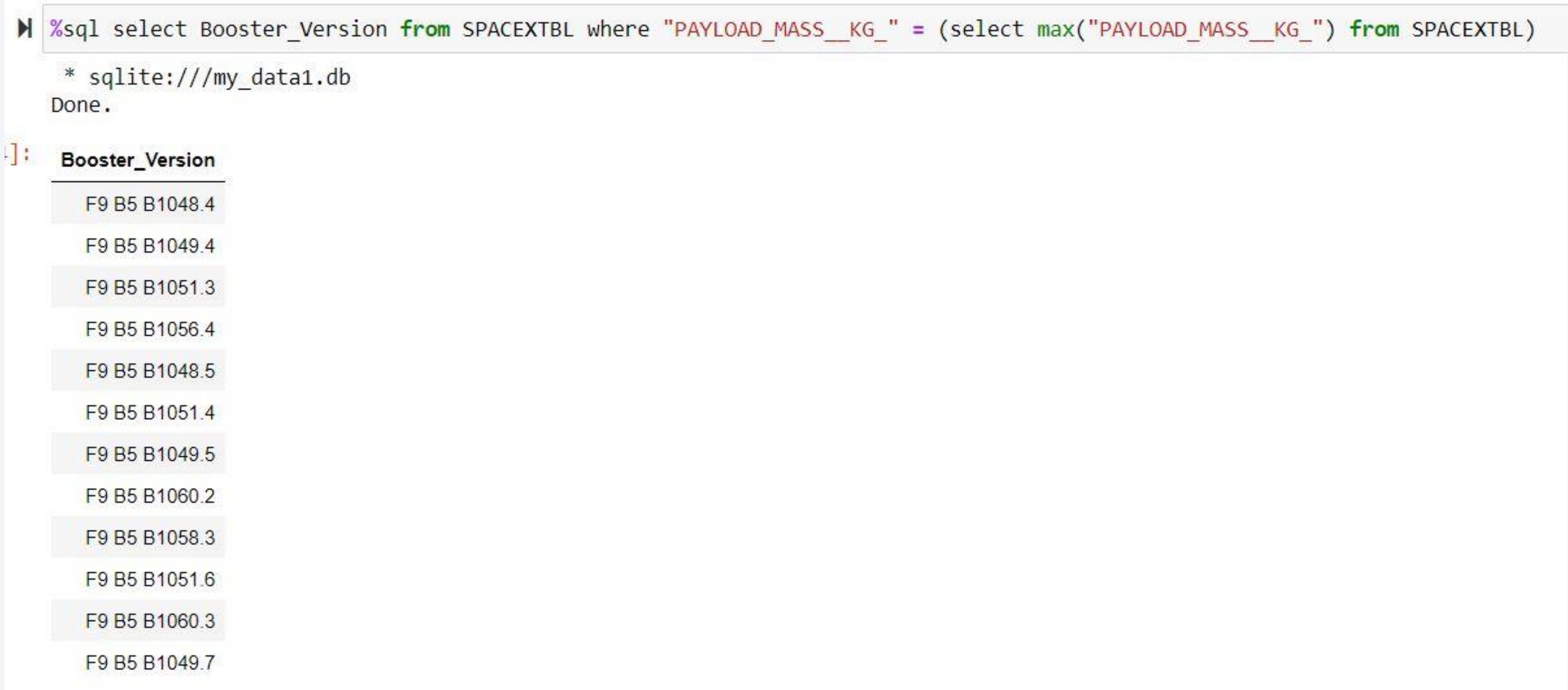
```
%sql select count(Mission_Outcome) as Successful_Failure_Mission_Outcomes from SPACEXTBL where "Mission_Outcome"='Success' or
```

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

| Successful_Failure_Mission_Outcomes |
|---|
| 99 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here



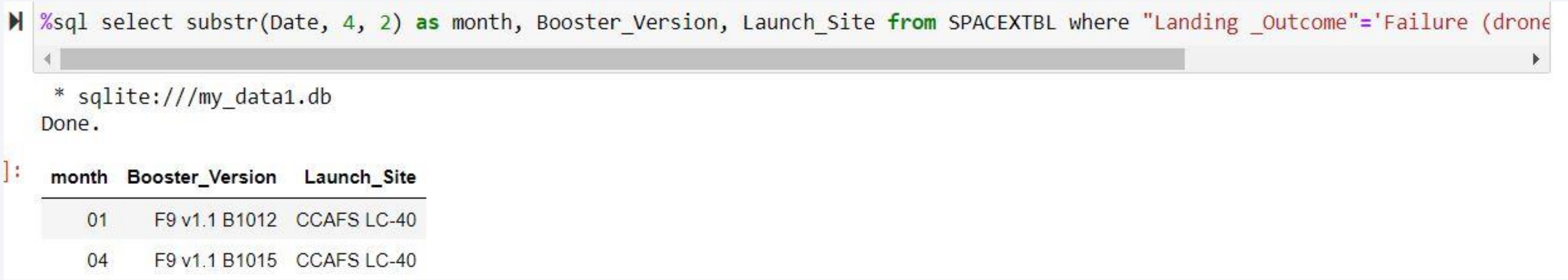
```
%sql select Booster_Version from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
```

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

| 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

- WHERE, LIKE, AND and BETWEEN conditions were used to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



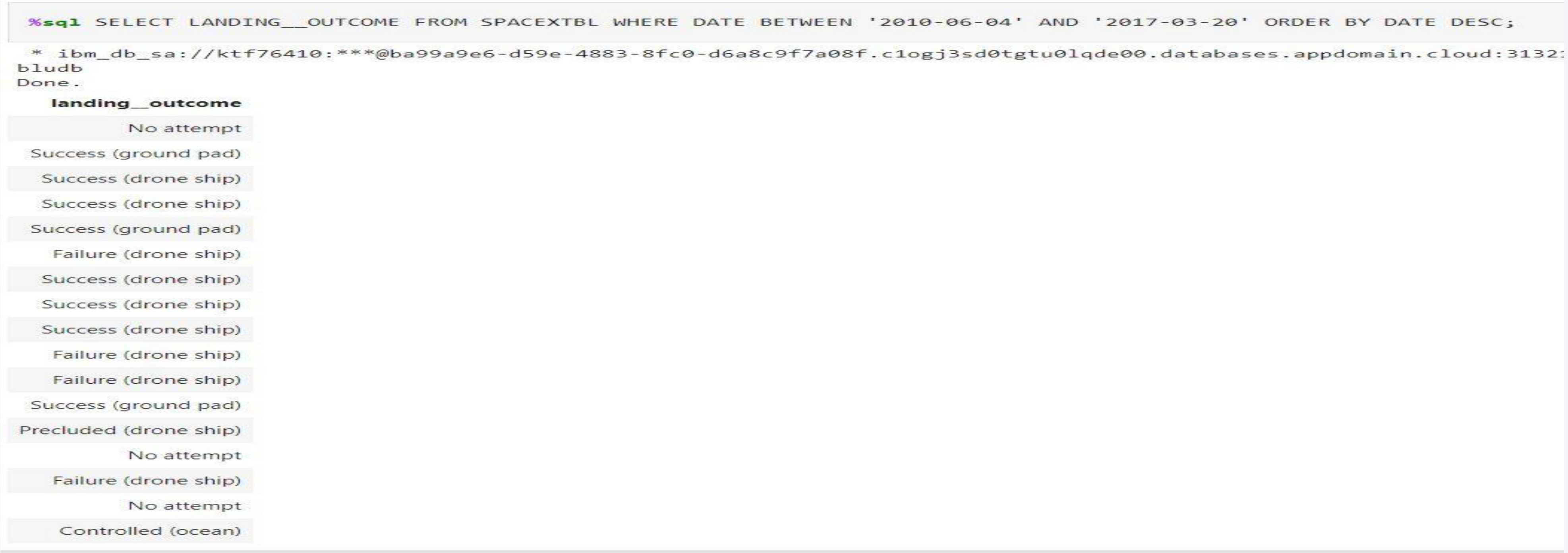
```
%sql select substr(Date, 4, 2) as month, Booster_Version, Launch_Site from SPACEXTBL where "Landing _Outcome"='Failure (drone
```

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

| month | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | 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



```
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

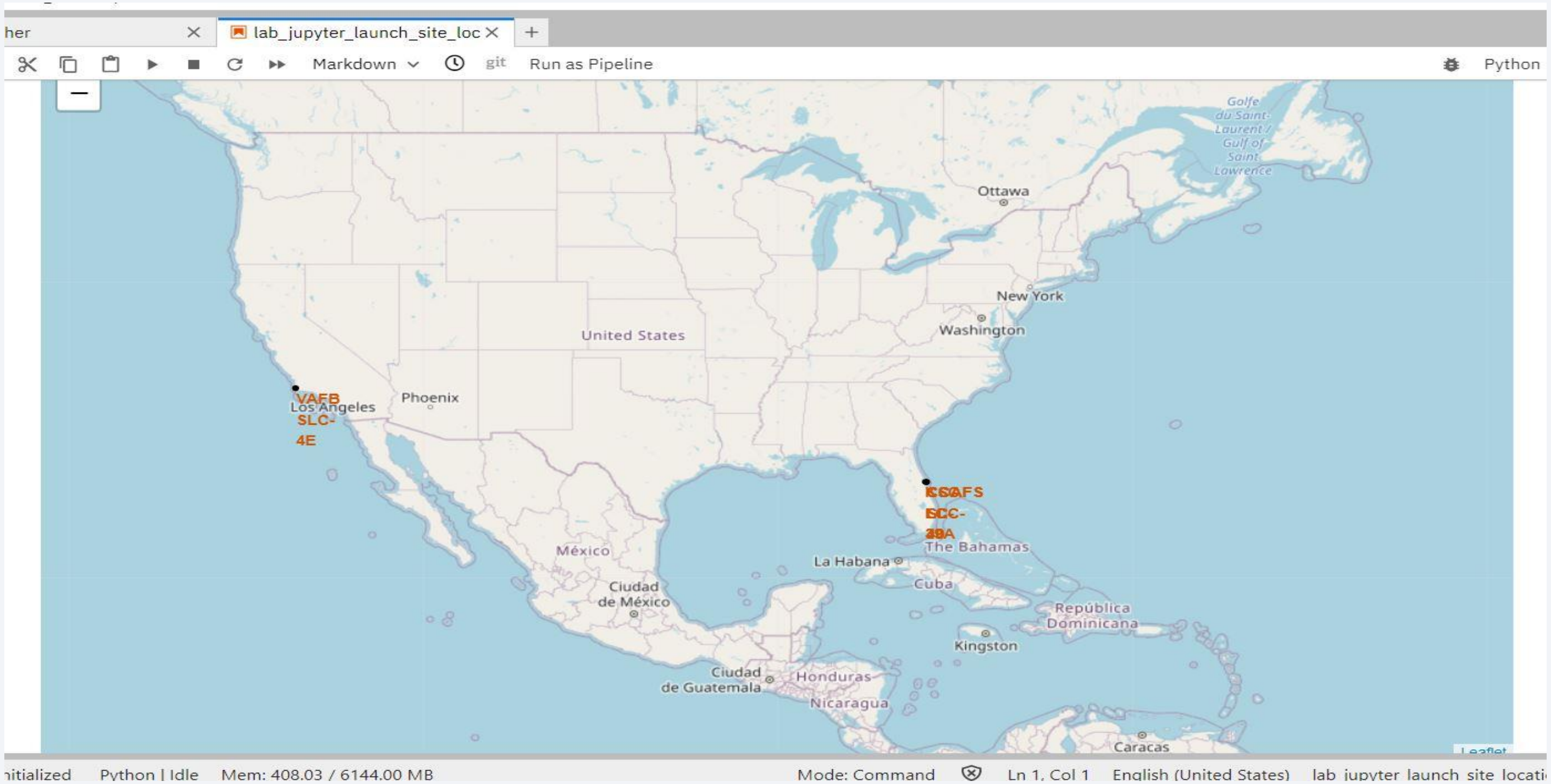
```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3132
bludb
Done.
    landing__outcome
          No attempt
Success (ground pad)
 Success (drone ship)
 Success (drone ship)
Success (ground pad)
  Failure (drone ship)
 Success (drone ship)
 Success (drone ship)
 Success (drone ship)
  Failure (drone ship)
  Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
          No attempt
  Failure (drone ship)
          No attempt
   Controlled (ocean)
```

Section 3

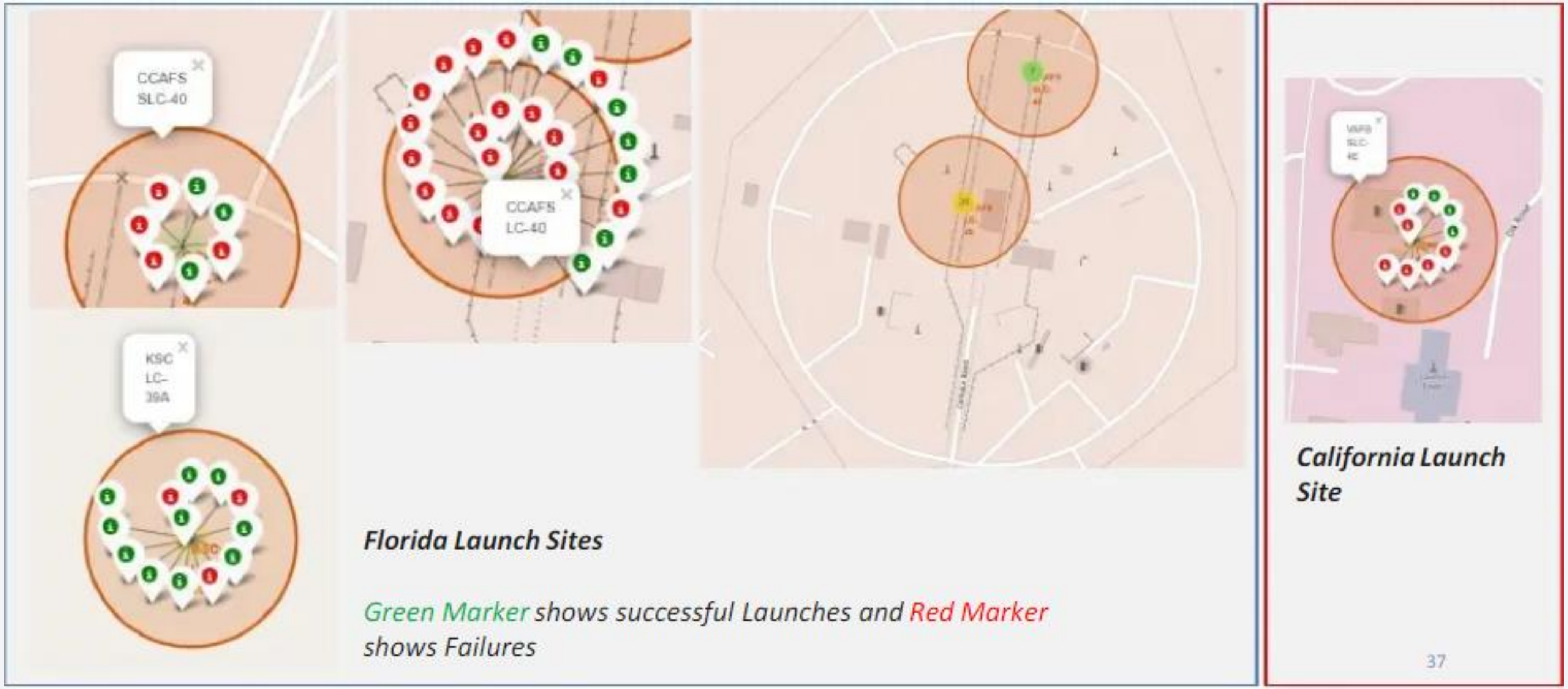
# Launch Sites Proximities Analysis

# All Launch Sites Global Map Markers

- The SpaceX launch sites are in the USA's coasts. Florida and California



# Color labelled markers



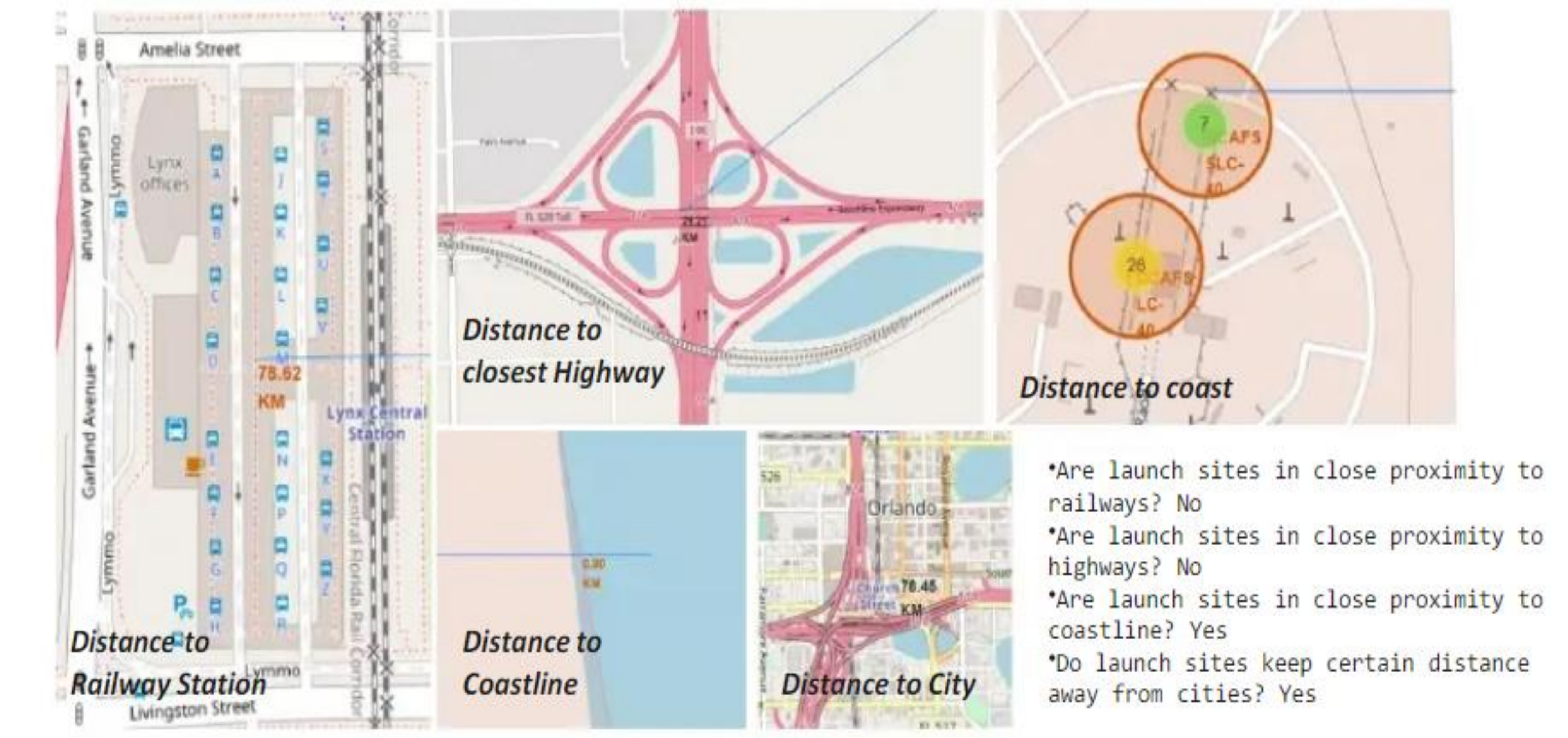
***Florida Launch Sites***

*Green Marker shows successful Launches and Red Marker shows Failures*

***California Launch Site***

37

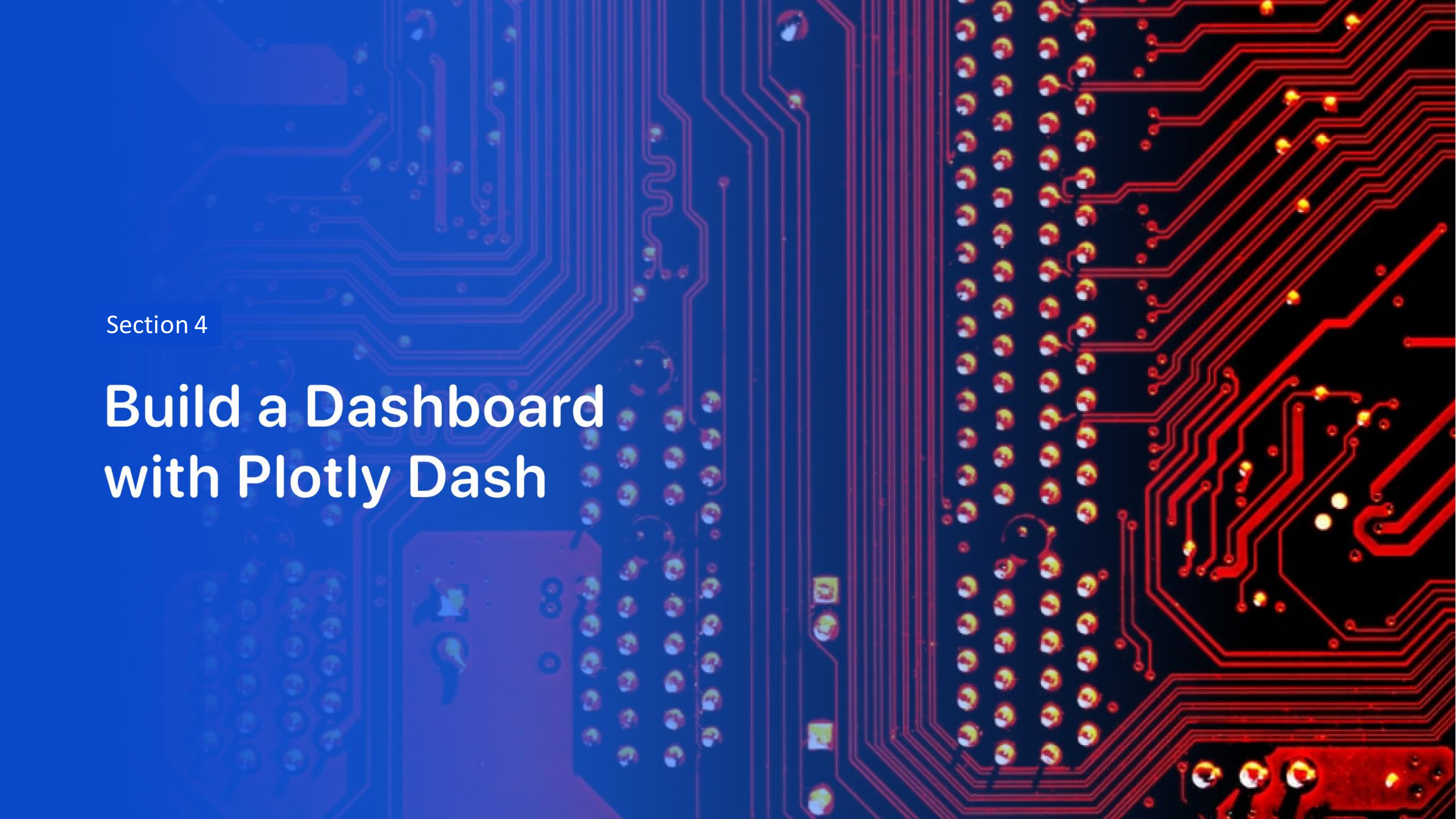
# Launch sites distance to landmarks



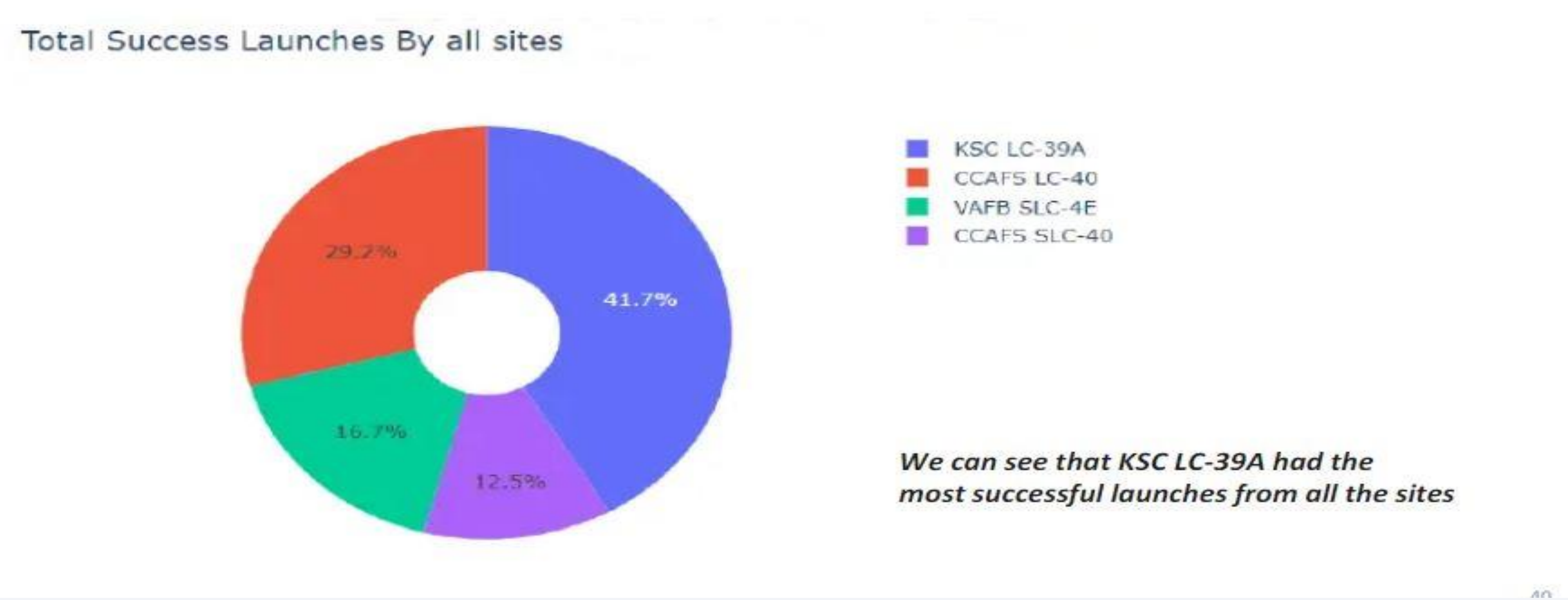
- •Are launch sites in close proximity to railways? No
- •Are launch sites in close proximity to highways? No
- •Are launch sites in close proximity to coastline? Yes
- •Do launch sites keep certain distance away from cities? Yes

Section 4

# Build a Dashboard with Plotly Dash

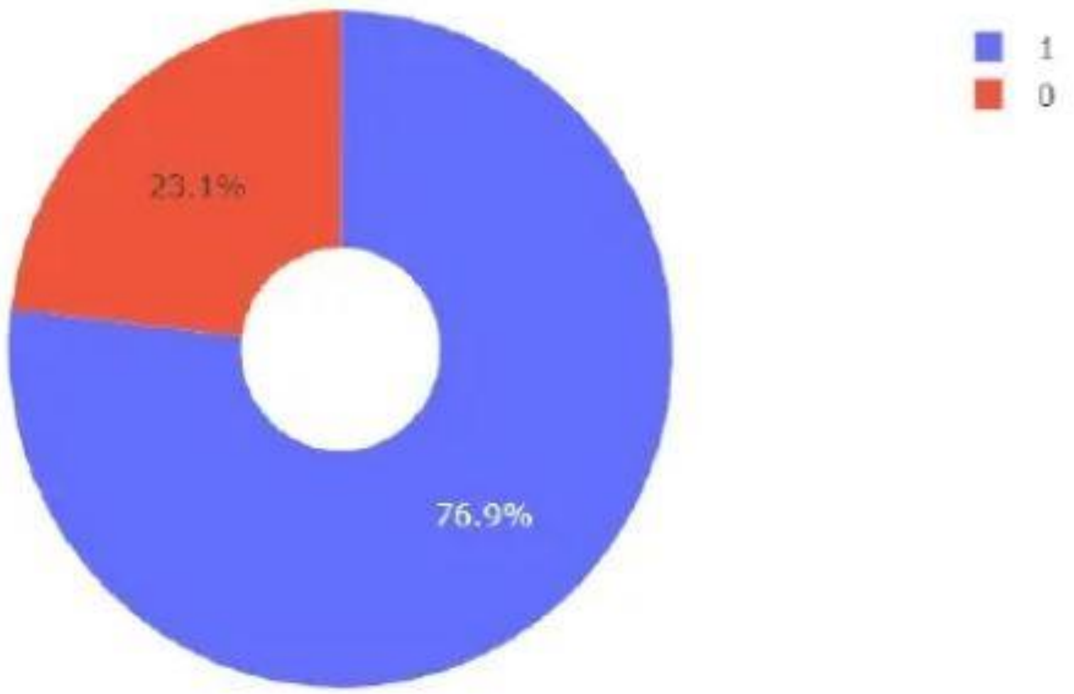


# Pie chart showing the success percentage achieved by each launch site



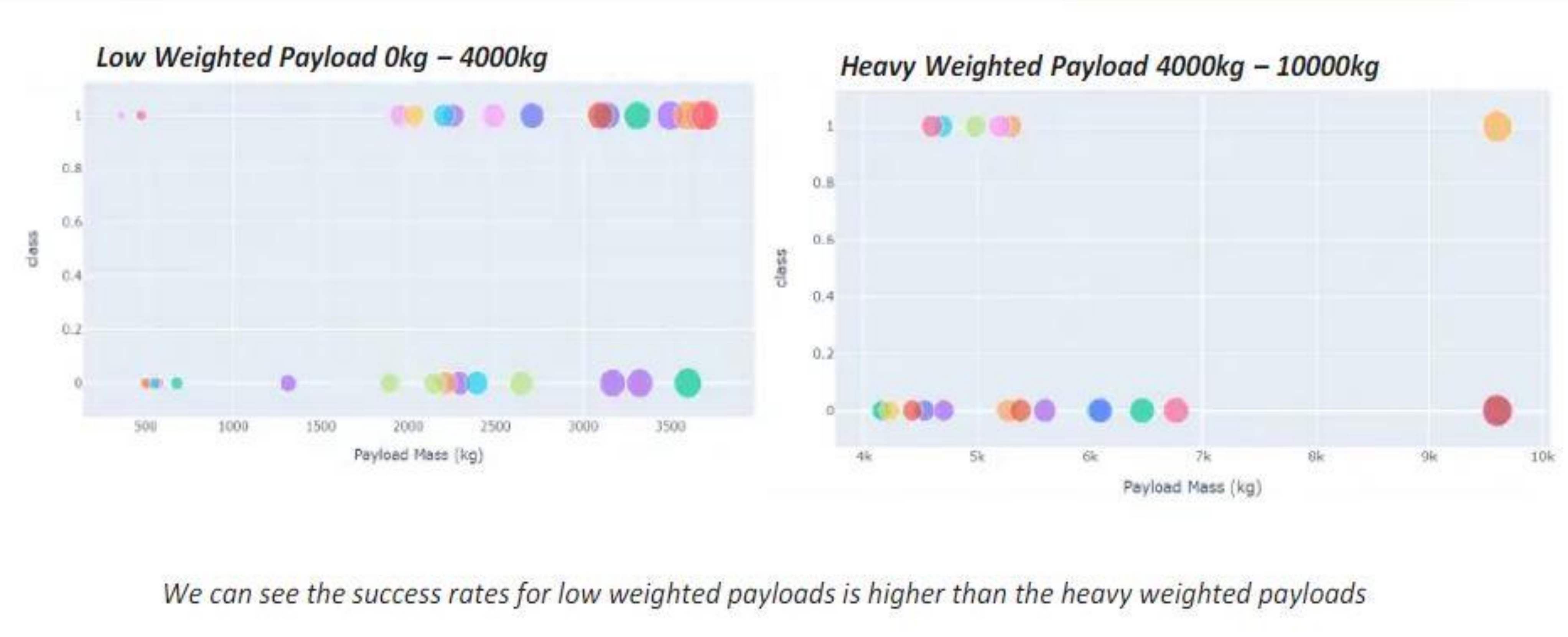
***We can see that KSC LC-39A had the most successful launches from all the sites***

# Pie chart showing the launch site with the highest launch success ratio



*KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate*

## Scatter plot of payload vs launch outcome for all sites with different payload selected in the range slider



Low Weighted Payload 0kg – 4000kg

Heavy Weighted Payload 4000kg – 10000kg

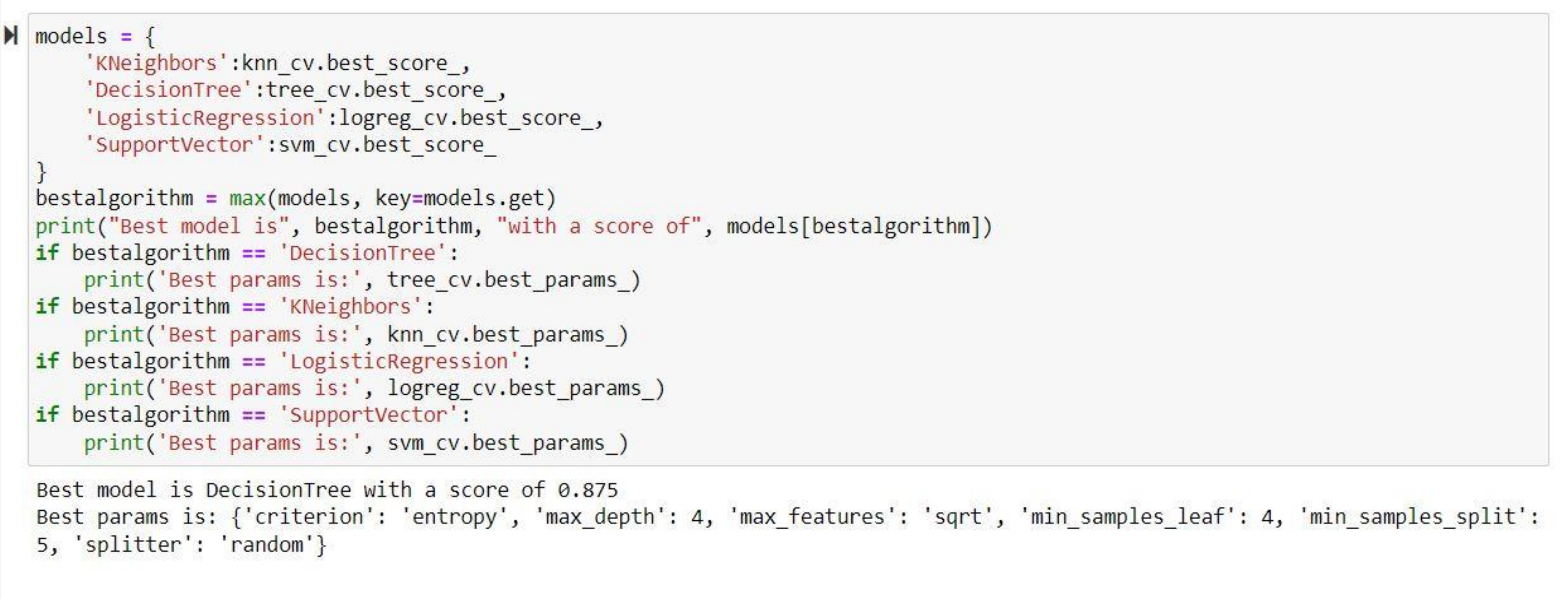
*We can see the success rates for low weighted payloads is higher than the heavy weighted payloads*

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- The decision tree is the model with the highest classification accuracy

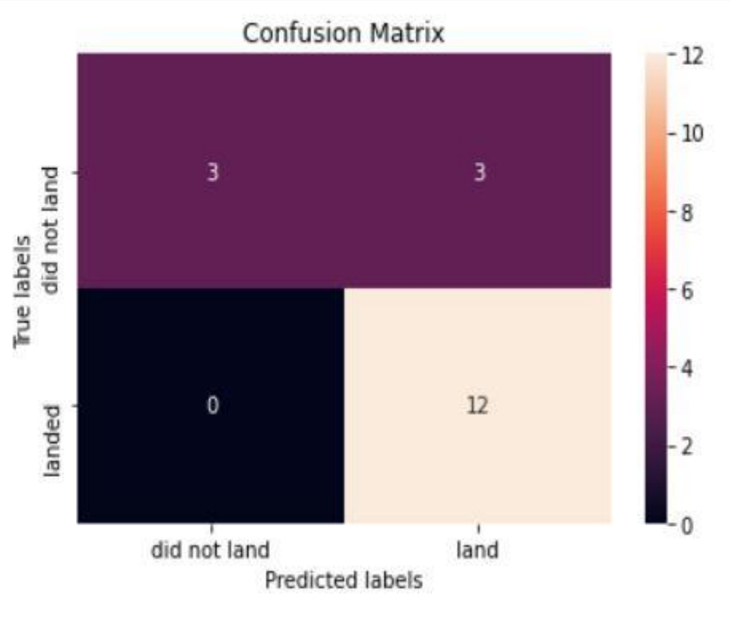


```
models = {
    'KNeighbors':knn_cv.best_score_,
    'DecisionTree':tree_cv.best_score_,
    'LogisticRegression':logreg_cv.best_score_,
    'SupportVector':svm_cv.best_score_
}
bestalgorithm = max(models, key=models.get)
print("Best model is", bestalgorithm, "with a score of", models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is:', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is:', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is:', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is:', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.875
Best params is: {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split':
5, 'splitter': 'random'}
```

# Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives.



# Conclusions

- Decision Tree is the best performing model for this data set
- The larger the flight amount at a launch site, the greater the success rate at a launch site
- Launch success rate started to increase in 2013 till 2020
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate
- KSC LC-39A had the most successful launches of any sites

# Thank you!