IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marlon CALISPA
15/06/2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context

This project is part of the Applied Data Science Capstone course, which in turn is part of the IBM Data Science Professional Certificate

In synthesizes the main ideas acquired in the specialization and shows in a summarized way the new concepts and knowledge.

- Problems you want to find answers

Was the course contents deep enough to solve the problems from the last module?

Are there any areas which need to be reinforced, as a prospective data scientist?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data for this module was collected mainly from two main sources:
    - SpaceX API (via jupyter Notebook app)
    - Web Scrapping (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

Methodology

- Perform data wrangling
  - Imported data was processed in python (pandas and Numpy libraries)
  - Wrangling consisted in missing values identification and initial data analysis (number of values, launches, etc).

Note: All the scripts were run locally (except the visualizations plotly which were run online)

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Libraries: pandas, numpy, seaborn, sqlite
  - Initial data analysis: check for potential correlations and trends (visual)
  - Data feature engineering.
  - Queries to understand the data
  - Create databases.

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
  - Libraries: folium, pandas
  - Maps (folium)
  - Interactive data displaying (dash, plotly)

Methodology

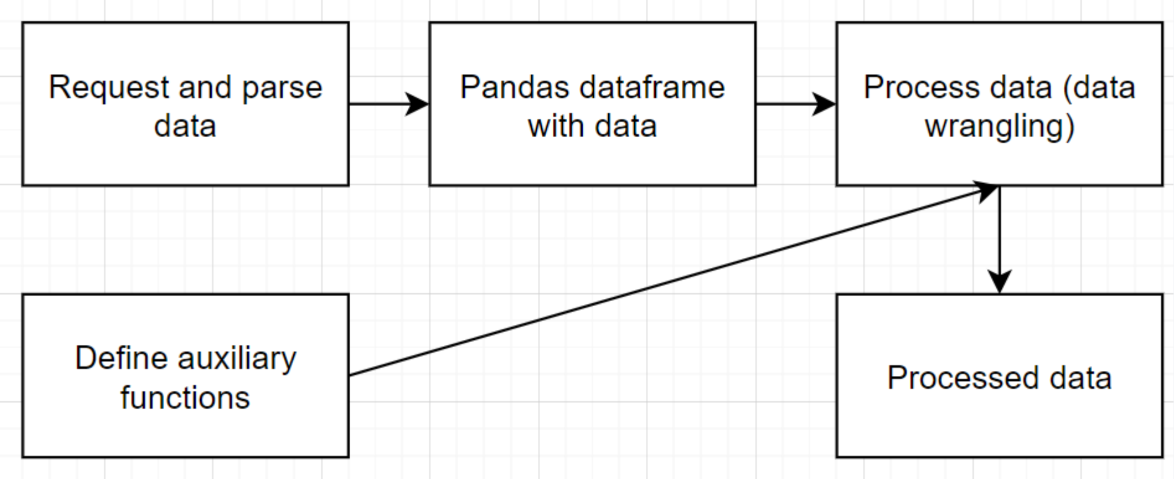
Executive Summary

- Perform predictive analysis using classification models
  - Libraries: mainly sklearn
  - Split the data
  - Fit models:
    - Logistic regression
    - Support vector machine
    - Decision trees
    - KNN
  - Assess the models (metrics)

Data Collection

Data Collection and wrangling – SpaceX API

- Data was downloaded from SpaceX's API
- Decode as pandas dataframe
- Processed (data wrangling)

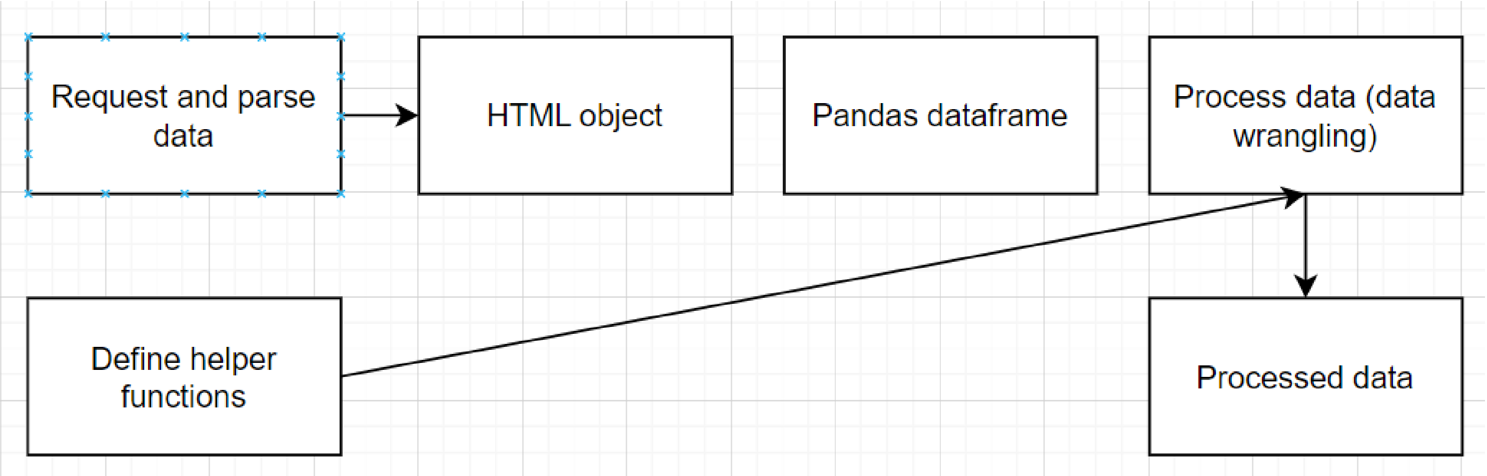

Request and parse data
Pandas dataframe with data
Process data (data wrangling)
Define auxiliary functions
Processed data

Summary chart of SpaceX data collection and processing

- GitHub URL: https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection and wrangling - Scraping

- Data requested from Wikipedia
- Converted to dataframe
- Processed (data wrangling)



Summary chart of Space launches data web scrapping and processing

- GitHub URLs: Web scrapping: https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/02_Data_collection_web_scrapping.ipynb
- Data wrangling: https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/03_Data_wrangling.ipynb

EDA with Data Visualization

The following plots were generated:

- Flight Number vs Payload Mass
- Flight Number and Launch site
- Payload vs Launch site
- Success rate and orbit type
- Flight number and orbit type
- Payload vs orbit type
- Launch success yearly trend

These charts were chosen to gain insight on the potential correlations and trends in the database. These correlations are helpful later, when models are chosen to make predictions.

Github URL:
https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/04_EDA_visualization.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 average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved

Github URL:
https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/05_EDA_SQL.ipynb

EDA with SQL

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- 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 URL:
https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/05_EDA_SQL.ipynb

Build an Interactive Map with Folium

- Three main stages on this point:
  - Add the launching sites
  - At each location add the outcome of each launch
  - Add distance calculations

Github URL: https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/06_Interactive_Map_Folium.ipynb

Build a Dashboard with Plotly Dash

- Two interactive plots:
  - Pie chart: Total success launches for each site and for all sites.
  - Slider: select payload range
  - Scatterplot: Payload range (selected by the slider) by booster category

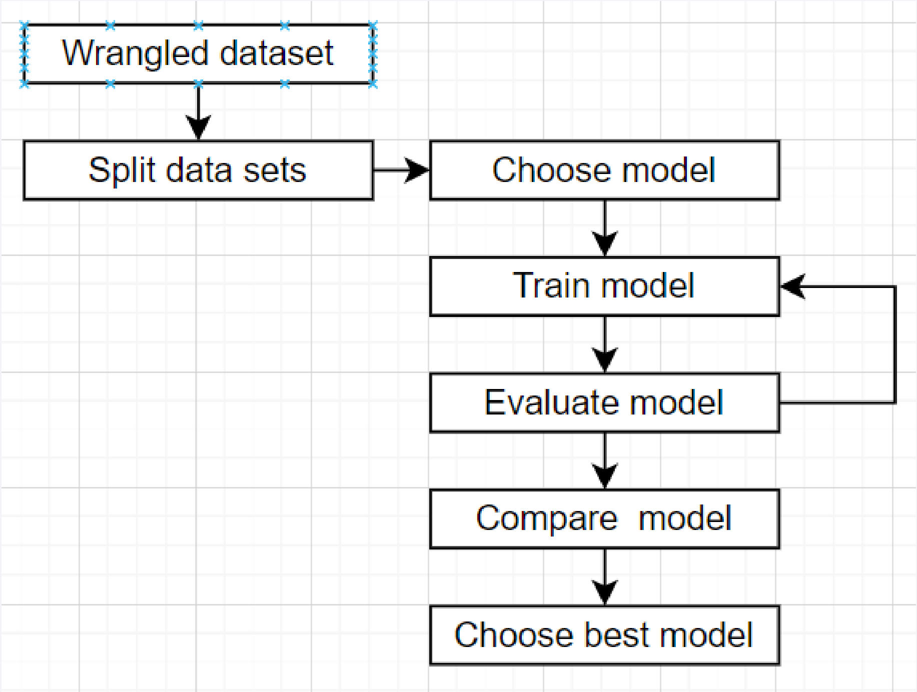
Github URL: https://github.com/marloncalispa/IBM_data_science_capstone/blob/main/07_espacex_dash_app.py

Predictive Analysis (Classification)

4 models were tested:

- Logistic regression
- Support vector machine
- Decision trees
- KNN

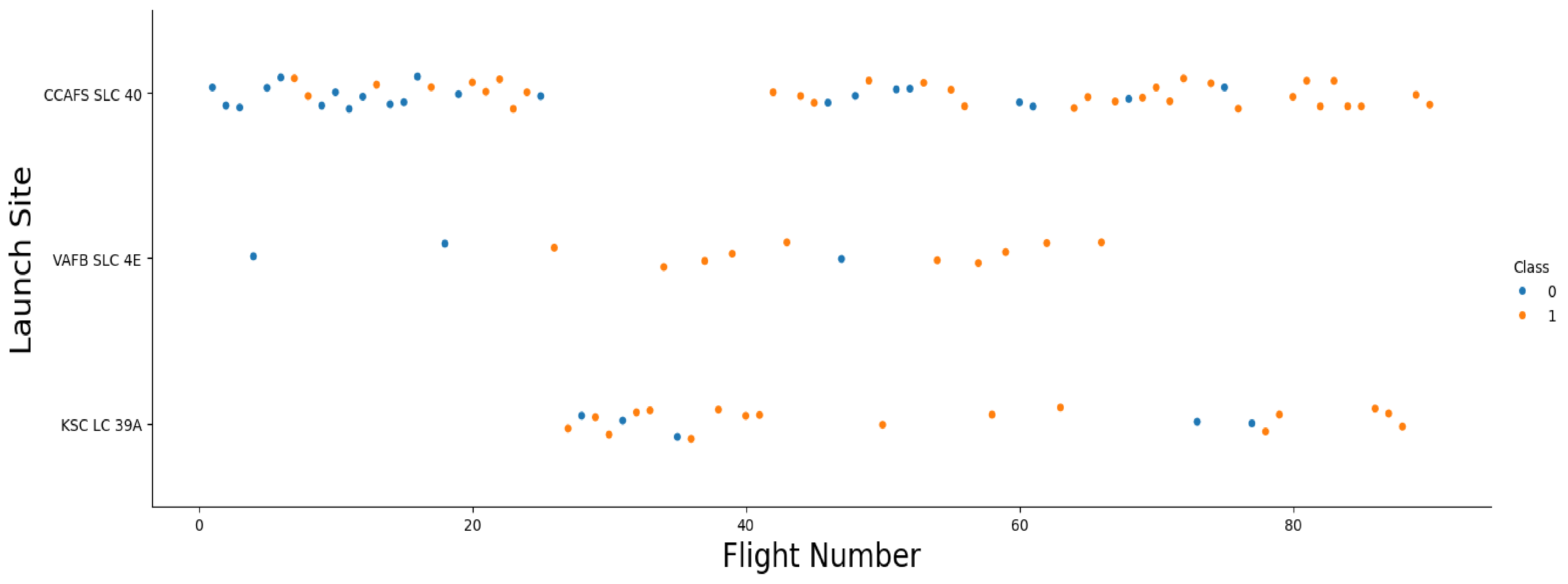
- For each model best parameters were chosen (hyp. Tuning)
- Metrics calculated for each one
- Best model chosen (best performance)



Section 2

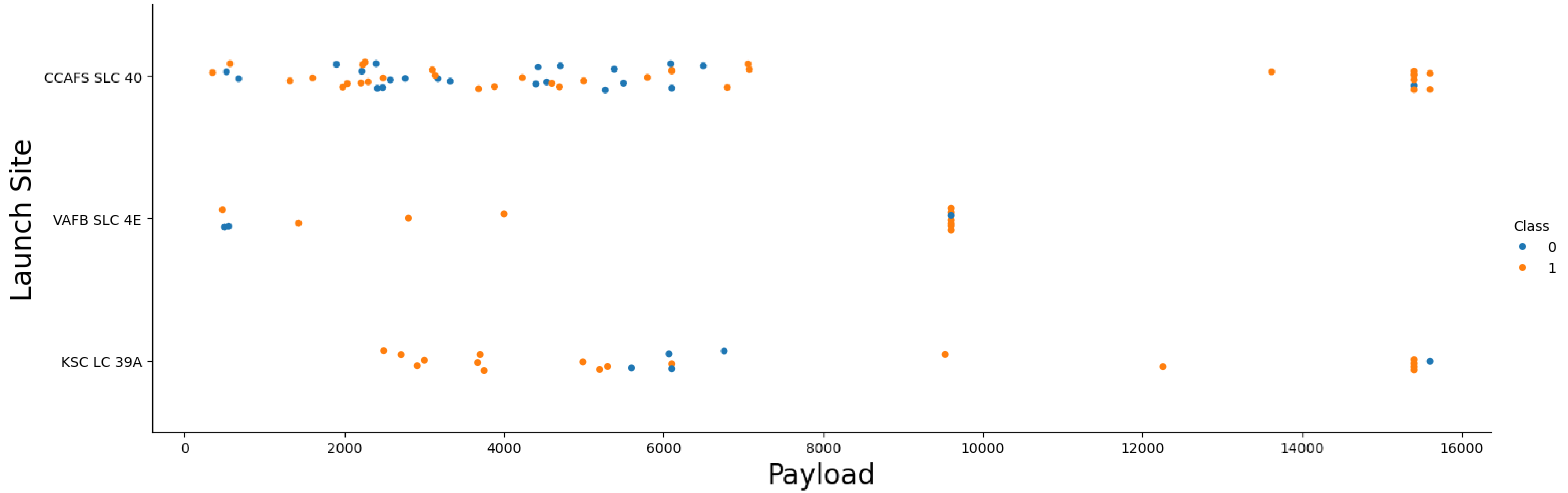
Insights drawn from EDA

Flight Number vs. Launch Site



- The plot shows the number of launches for 3 sites
- Site CCAFS SLC 40 shows the higher number of launches, both success/unsuccess

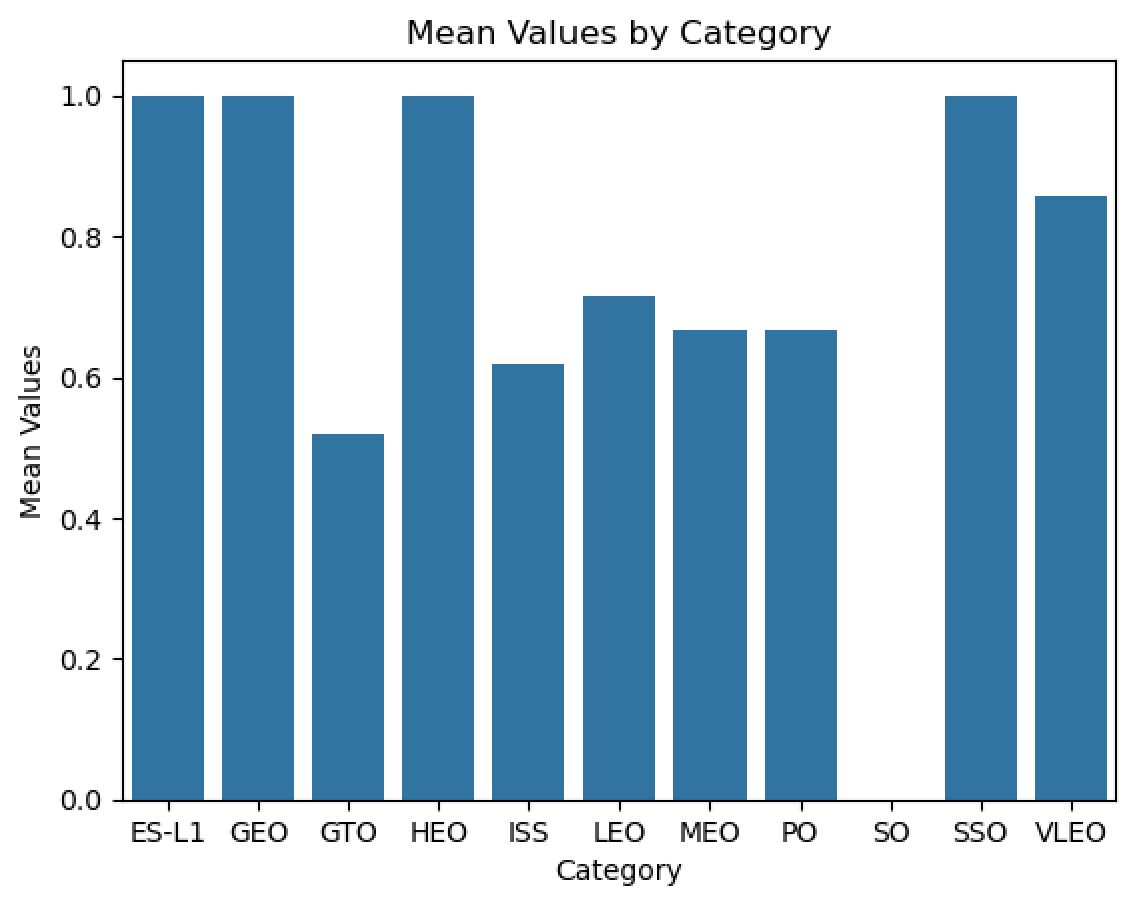
Payload vs. Launch Site



- Distribution of payloads is uneven
- For VAFB SLC 4E only payloads < 10000 kg are launched
- More common payloads are smaller than 8000 kg.

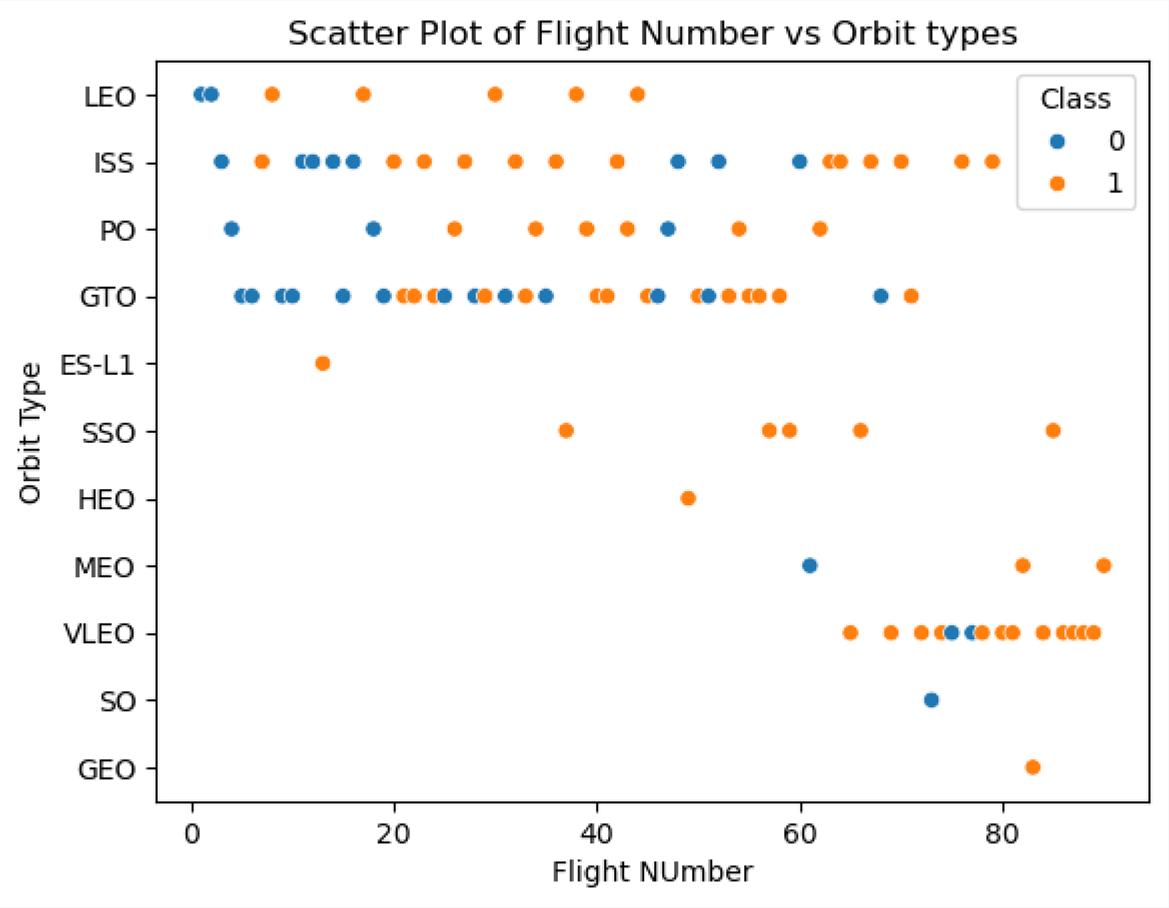
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO and VLEO orbits are the most successful.
- SO is unsuccessful
- The rate for the remaining types fluctuate between 0.5 and 0.7



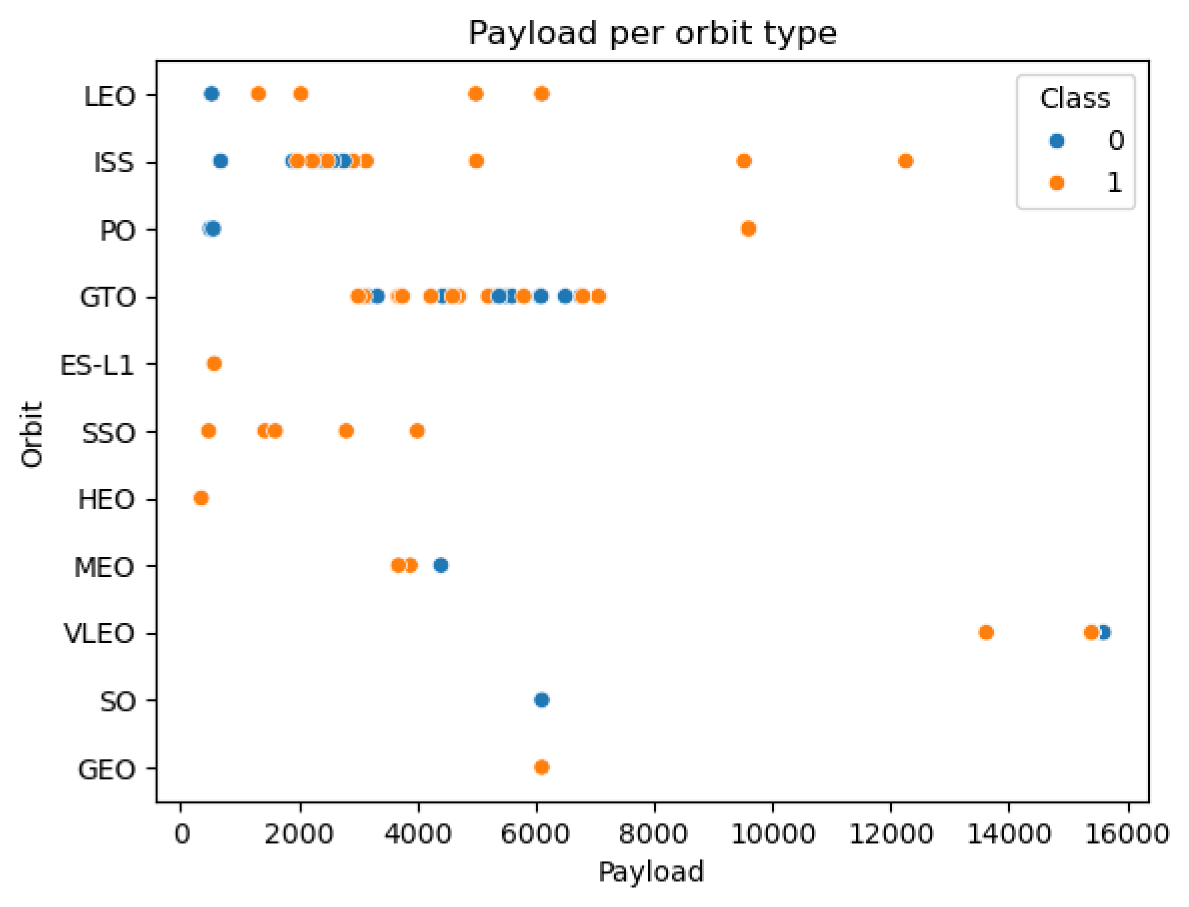
Flight Number vs. Orbit Type

- In some orbit types, flight number correlates with output
- LEO for example, low flight number correlates with unsuccess launches
- Other types seem to be less sensitive to flight number



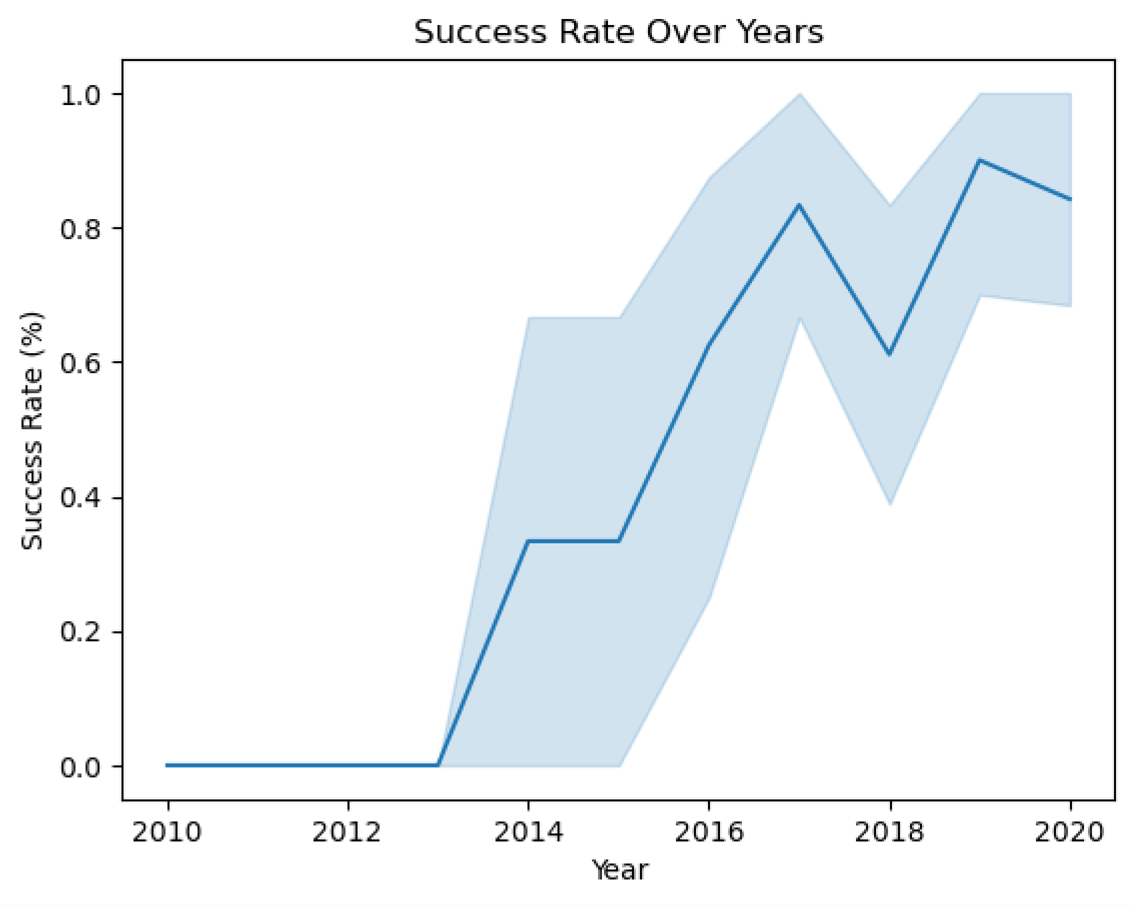
Payload vs. Orbit Type

- Payload ranges vary per orbit type.
- Only 3 orbit types (ISS, PO, VLEO) have payloads > 8000 kg.
- VLEO only has payloads > 12000 kg, due to the nature of it.



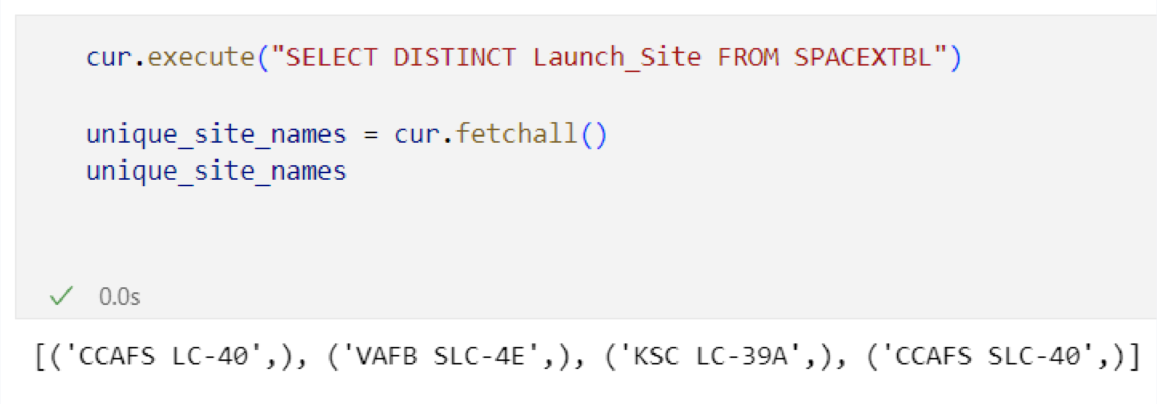
Launch Success Yearly Trend

- Success rate increases with time (as usually expected in engineering)
- Since 2013 the success rate increases steadily
- Small drop in 2018, but then increases again.



All Launch Site Names

- 4 unique sites
- Used the SELECT DISTINCT query from the created database (using a cursor)
- Then fetch all the results from the cursor.



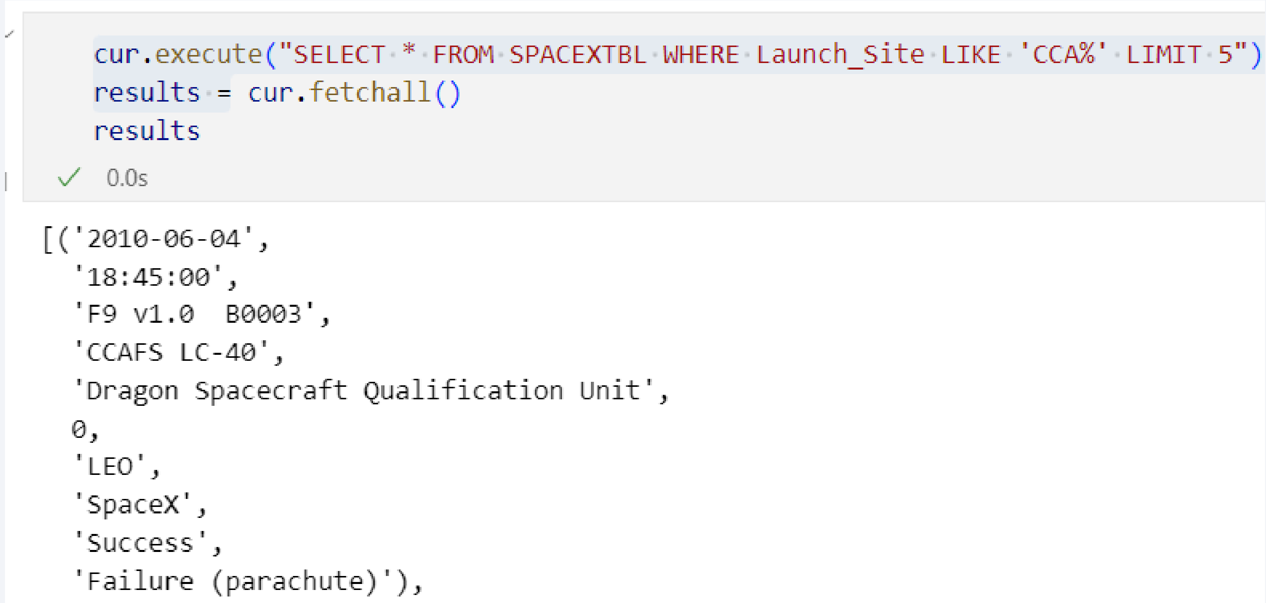
```
cur.execute("SELECT DISTINCT Launch_Site FROM SPACEXTBL")

unique_site_names = cur.fetchall()
unique_site_names
```

✓ 0.0s

```
[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

Launch Site Names Begin with 'CCA'

- Used the SELECT * query combined with the WHERE and LIKE
- Five sites retrieved:
  - CCAFS LC-40
  - 3 successes, 2 failures.



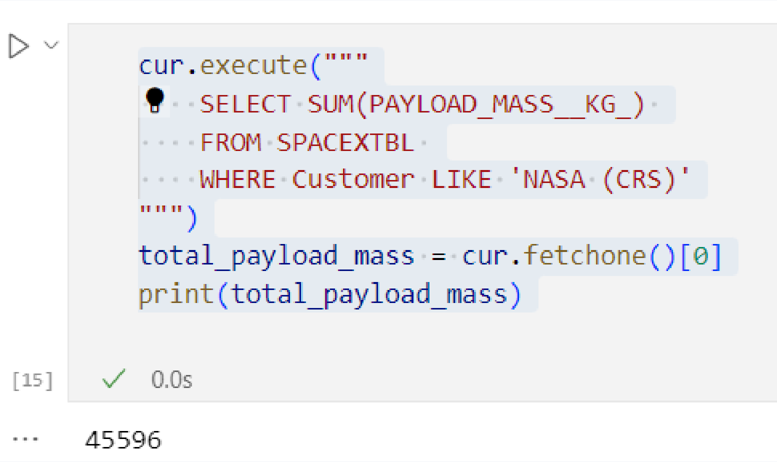
```
cur.execute("SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5")
results = cur.fetchall()
results
```

✓ 0.0s

```
[('2010-06-04',
  '18:45:00',
  'F9 v1.0  B0003',
  'CCAFS LC-40',
  'Dragon Spacecraft Qualification Unit',
  0,
  'LEO',
  'SpaceX',
  'Success',
  'Failure (parachute)'),
```

Total Payload Mass

- 45596 kg in total, for NASA mission's payload
- Combination of SELECT with SUM (to obtain the total) and WHERE to specify the customer, NASA in this case.

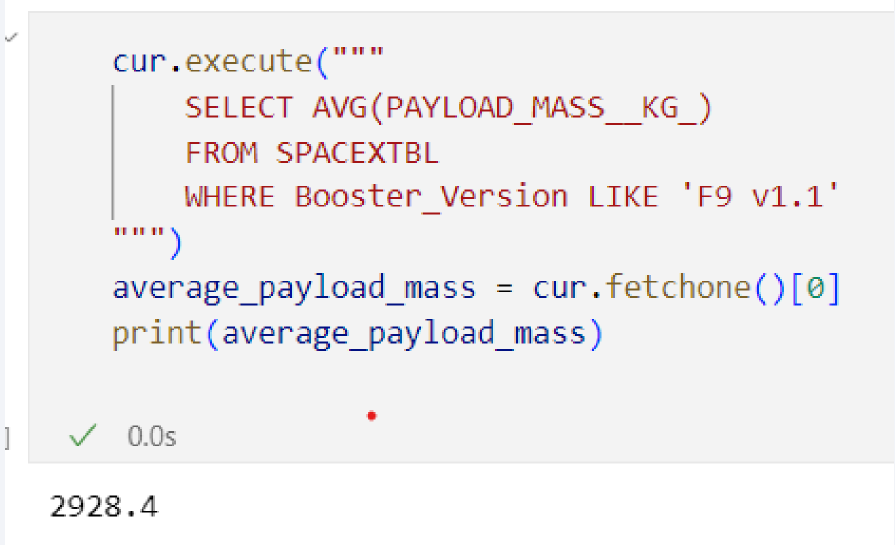


```
cur.execute("""
    SELECT SUM(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL
    WHERE Customer LIKE 'NASA (CRS)'
""")
total_payload_mass = cur.fetchone()[0]
print(total_payload_mass)
```

```
45596
```

Average Payload Mass by F9 v1.1

- 2984.4 kg on average are sent in F9 v1.1 booster
- Used a combination of SELECT with AVG (to get the mean value of payload) and WHERE to specify the booster type.

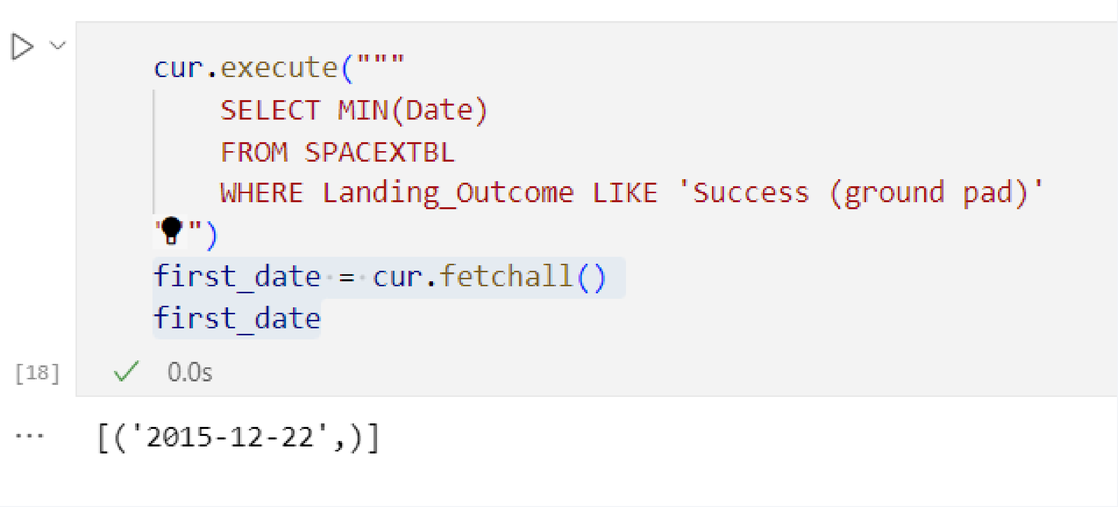


```
cur.execute("""
    SELECT AVG(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL
    WHERE Booster_Version LIKE 'F9 v1.1'
""")
average_payload_mass = cur.fetchone()[0]
print(average_payload_mass)
```

```
2928.4
```

First Successful Ground Landing Date

- 2015 – 12 – 22
- Used a combination of SELECT and MIN to get the first date and WHERE + LIKE to select the Landing Outcome and Ground pad

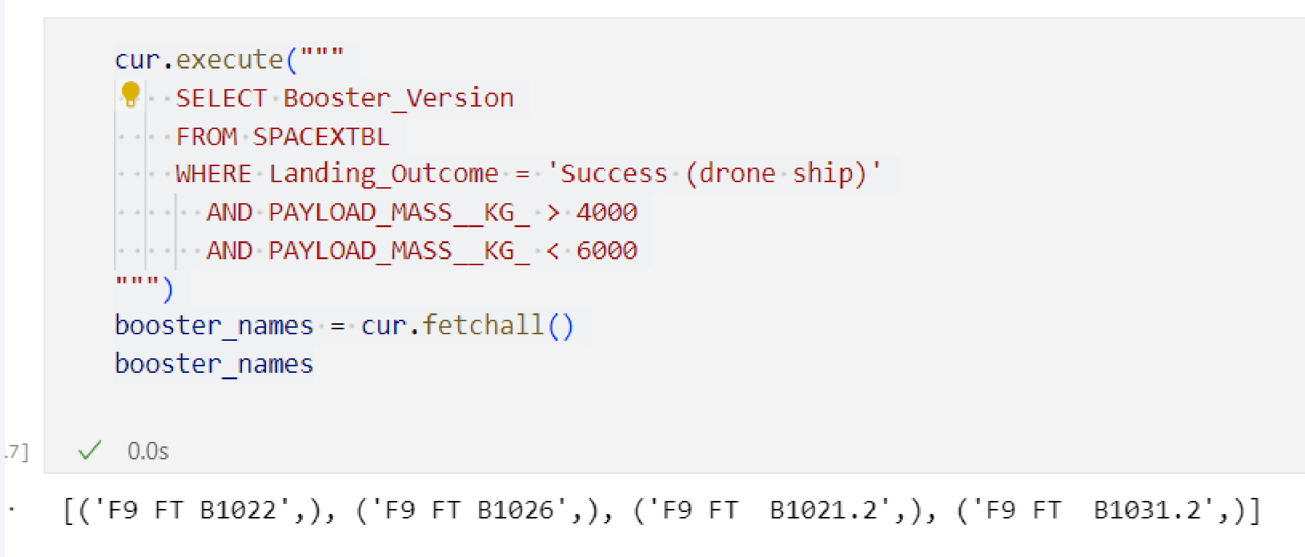


```
cur.execute("""
    SELECT MIN(Date)
    FROM SPACEXTBL
    WHERE Landing_Outcome LIKE 'Success (ground pad)'
""")
first_date = cur.fetchall()
first_date
```

```
[('2015-12-22',)]
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- 4 boosters selected
- Combinations of SELECT and WHERE with conditionals

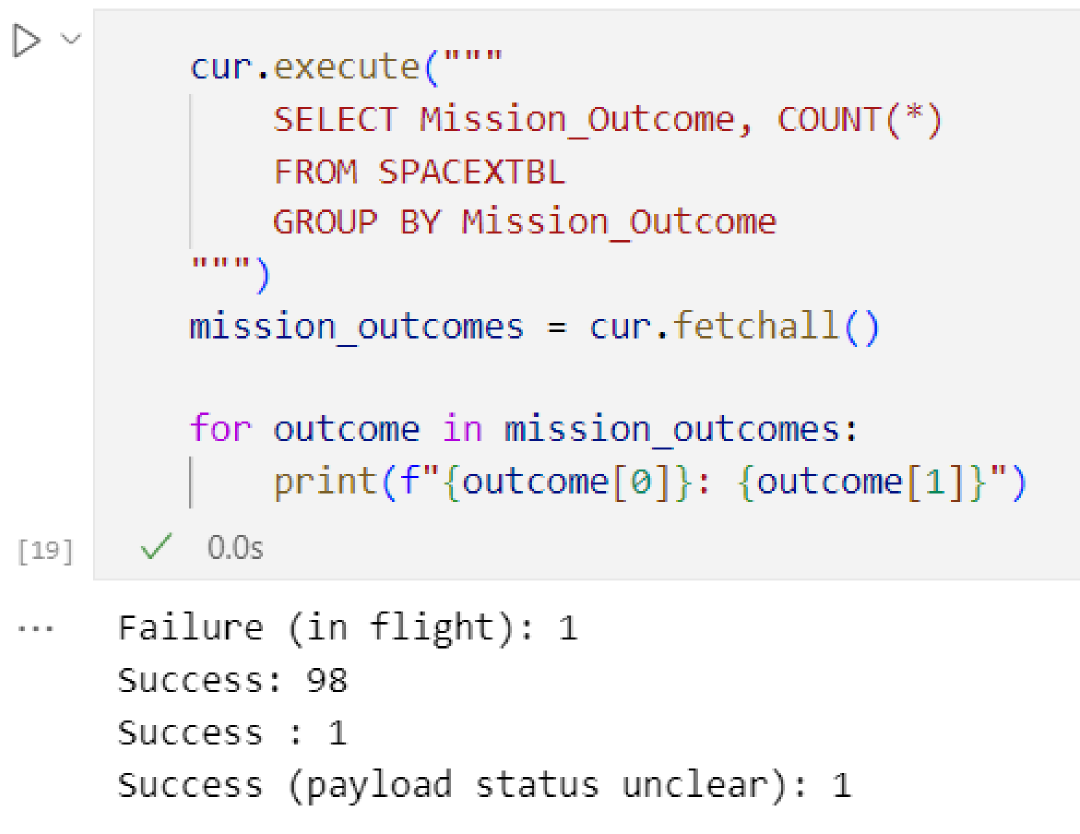


```
cur.execute("""
    SELECT Booster_Version
    FROM SPACEXTBL
    WHERE Landing_Outcome = 'Success (drone ship)'
      AND PAYLOAD_MASS__KG_ > 4000
      AND PAYLOAD_MASS__KG_ < 6000
""")
booster_names = cur.fetchall()
booster_names
```

```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

Total Number of Successful and Failure Mission Outcomes

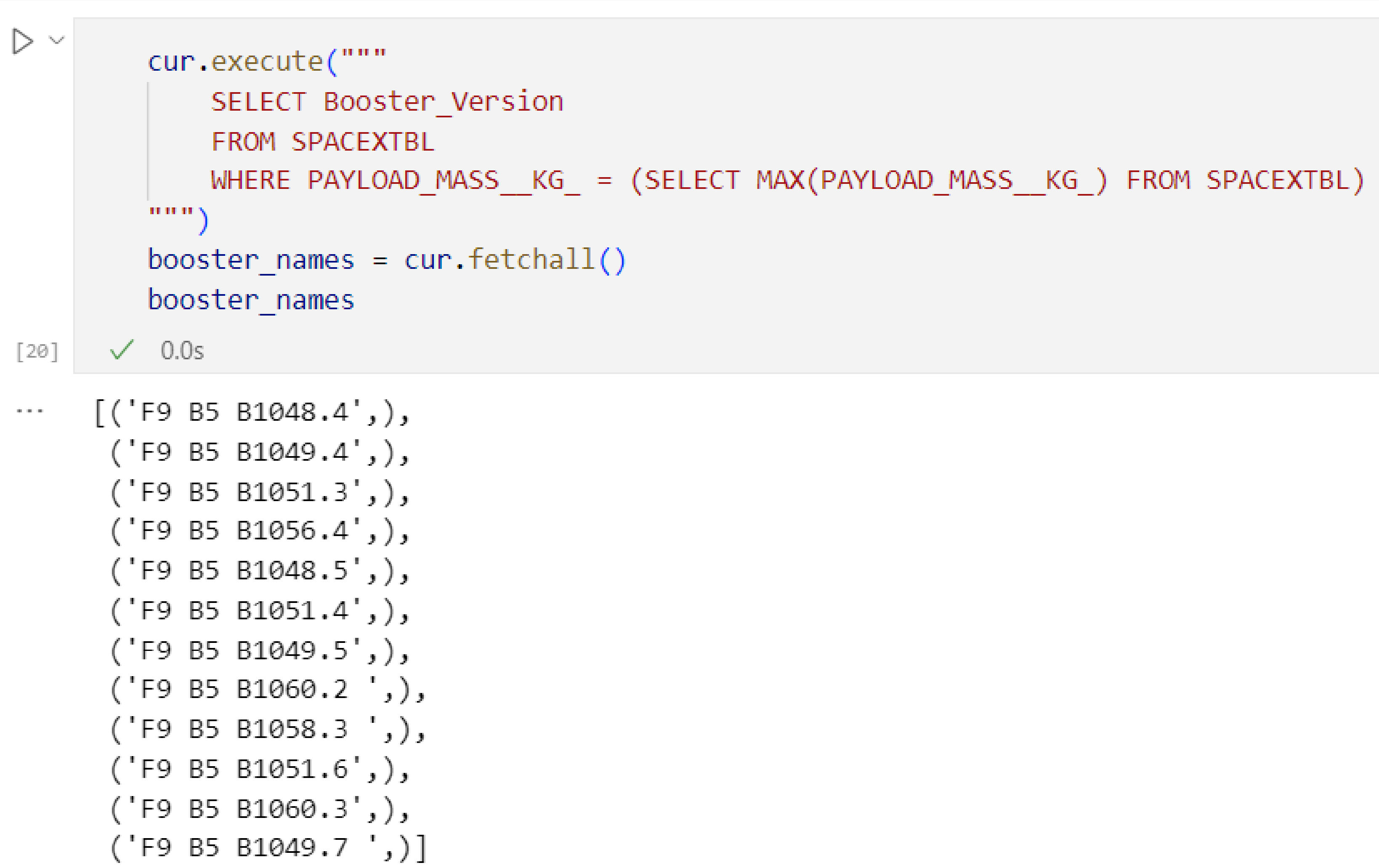
- I used a combination of SELECT, but counting all the values (COUNT (*)) and grouping by Mission Outcome



```
cur.execute("""
    SELECT Mission_Outcome, COUNT(*)
    FROM SPACEXTBL
    GROUP BY Mission_Outcome
""")
mission_outcomes = cur.fetchall()

for outcome in mission_outcomes:
    print(f"{outcome[0]}: {outcome[1]}")
```

```
Failure (in flight): 1
Success: 98
Success : 1
Success (payload status unclear): 1
```

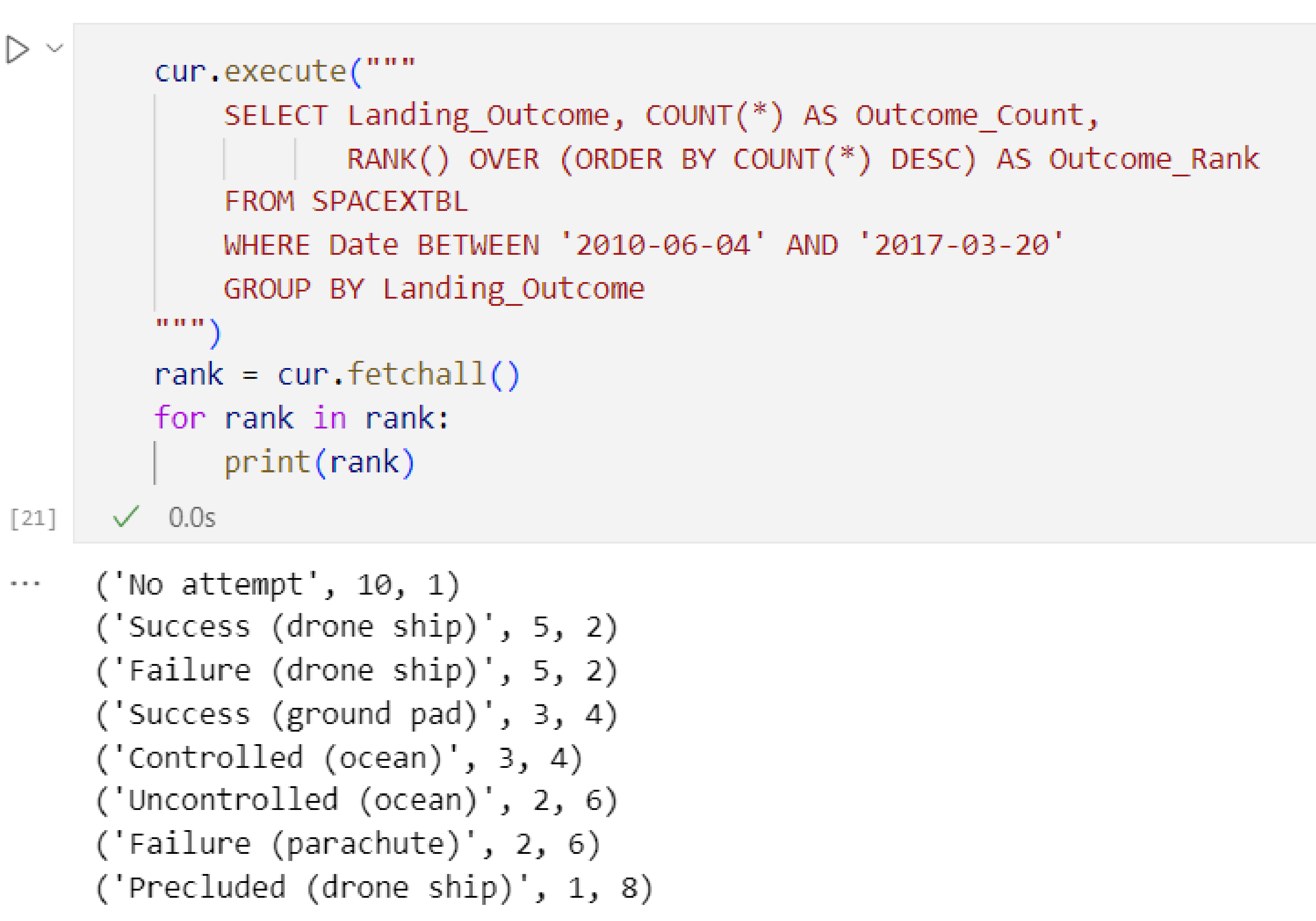
Boosters Carried Maximum Payload



```
cur.execute("""
    SELECT Booster_Version
    FROM SPACEXTBL
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
""")
booster_names = cur.fetchall()
booster_names
```

```
[('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 ',)]
```

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



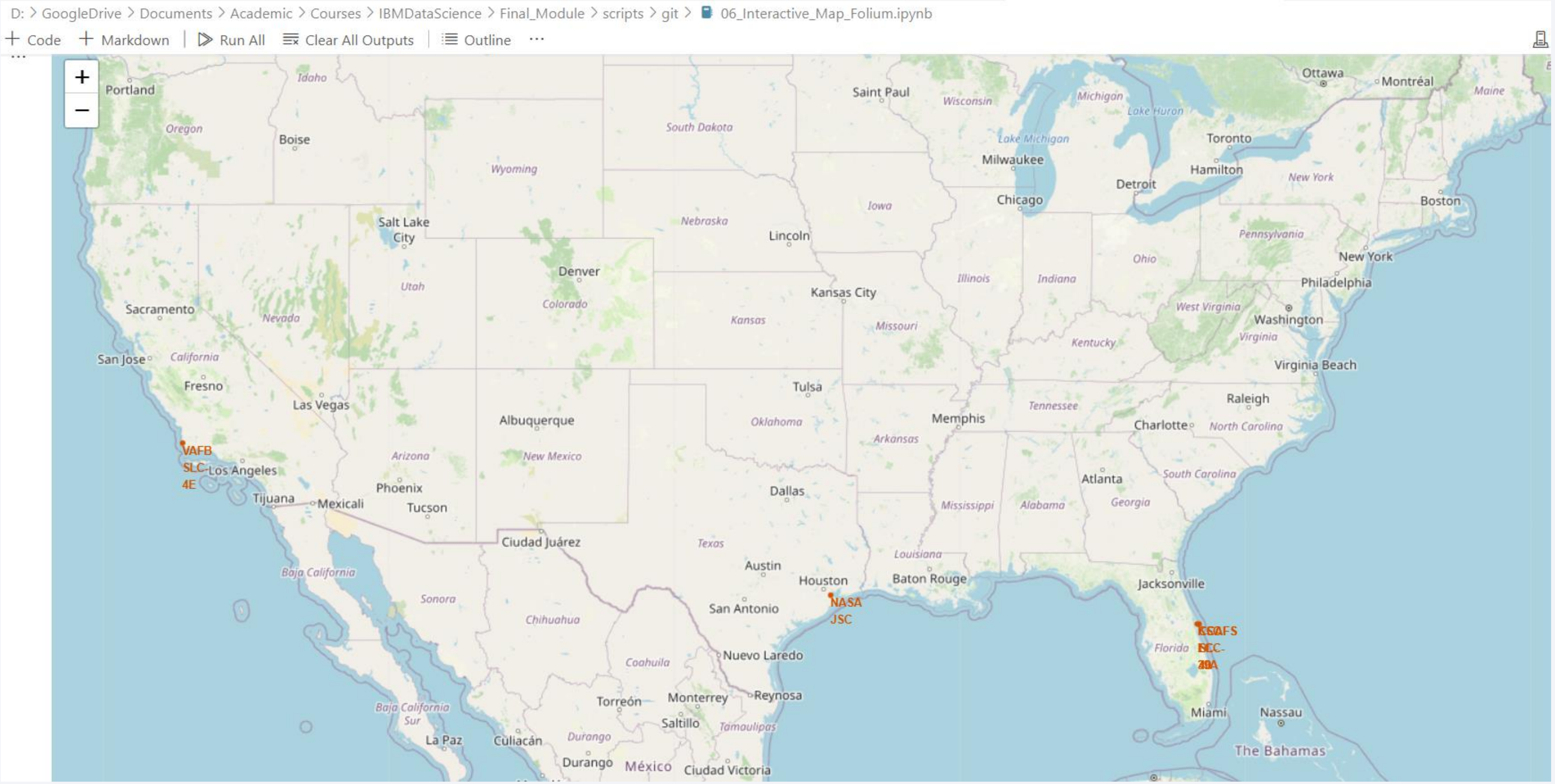
```
cur.execute("""
    SELECT Landing_Outcome, COUNT(*) AS Outcome_Count,
           RANK() OVER (ORDER BY COUNT(*) DESC) AS Outcome_Rank
    FROM SPACEXTBL
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
""")
rank = cur.fetchall()
for rank in rank:
    print(rank)
```

```
('No attempt', 10, 1)
('Success (drone ship)', 5, 2)
('Failure (drone ship)', 5, 2)
('Success (ground pad)', 3, 4)
('Controlled (ocean)', 3, 4)
('Uncontrolled (ocean)', 2, 6)
('Failure (parachute)', 2, 6)
('Precluded (drone ship)', 1, 8)
```

Section 3

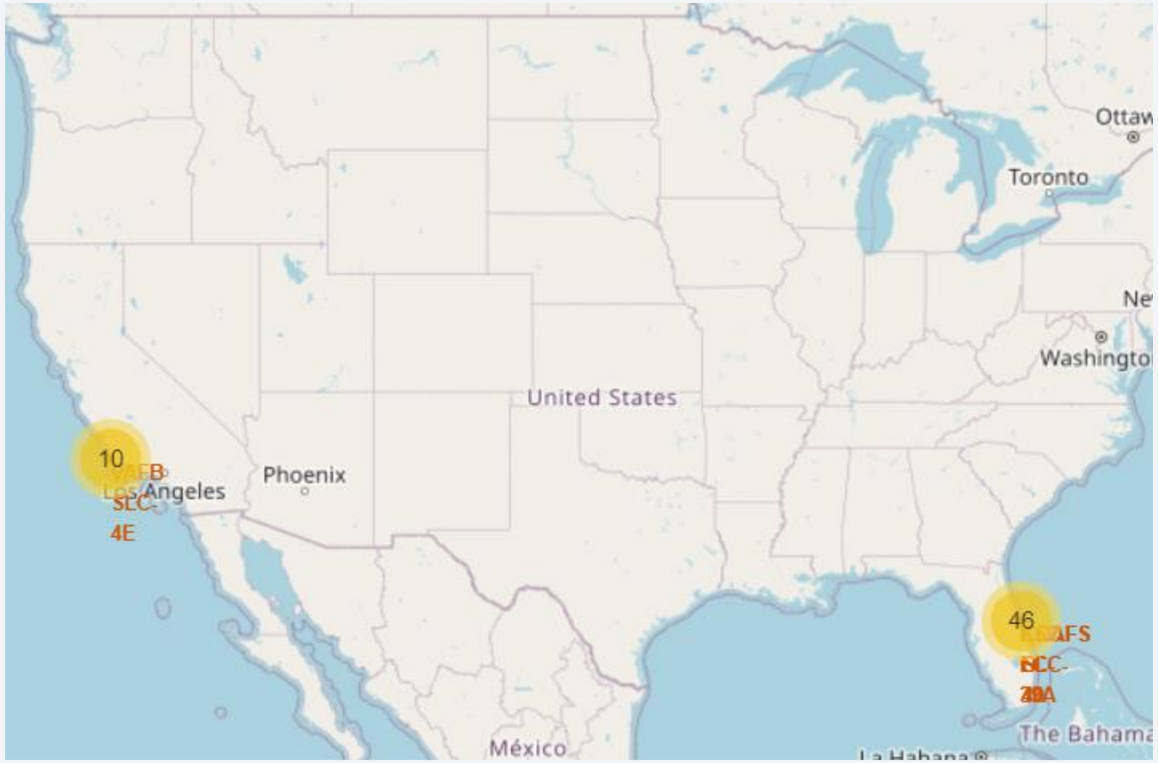
Launch Sites Proximities Analysis

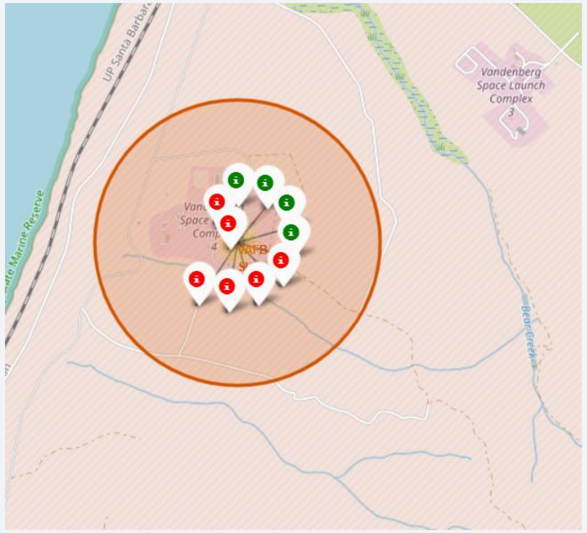
SpaceX launching sites Map

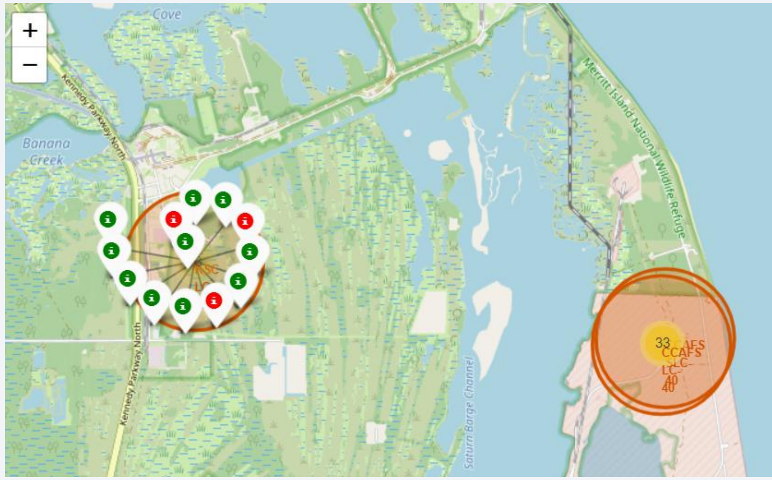


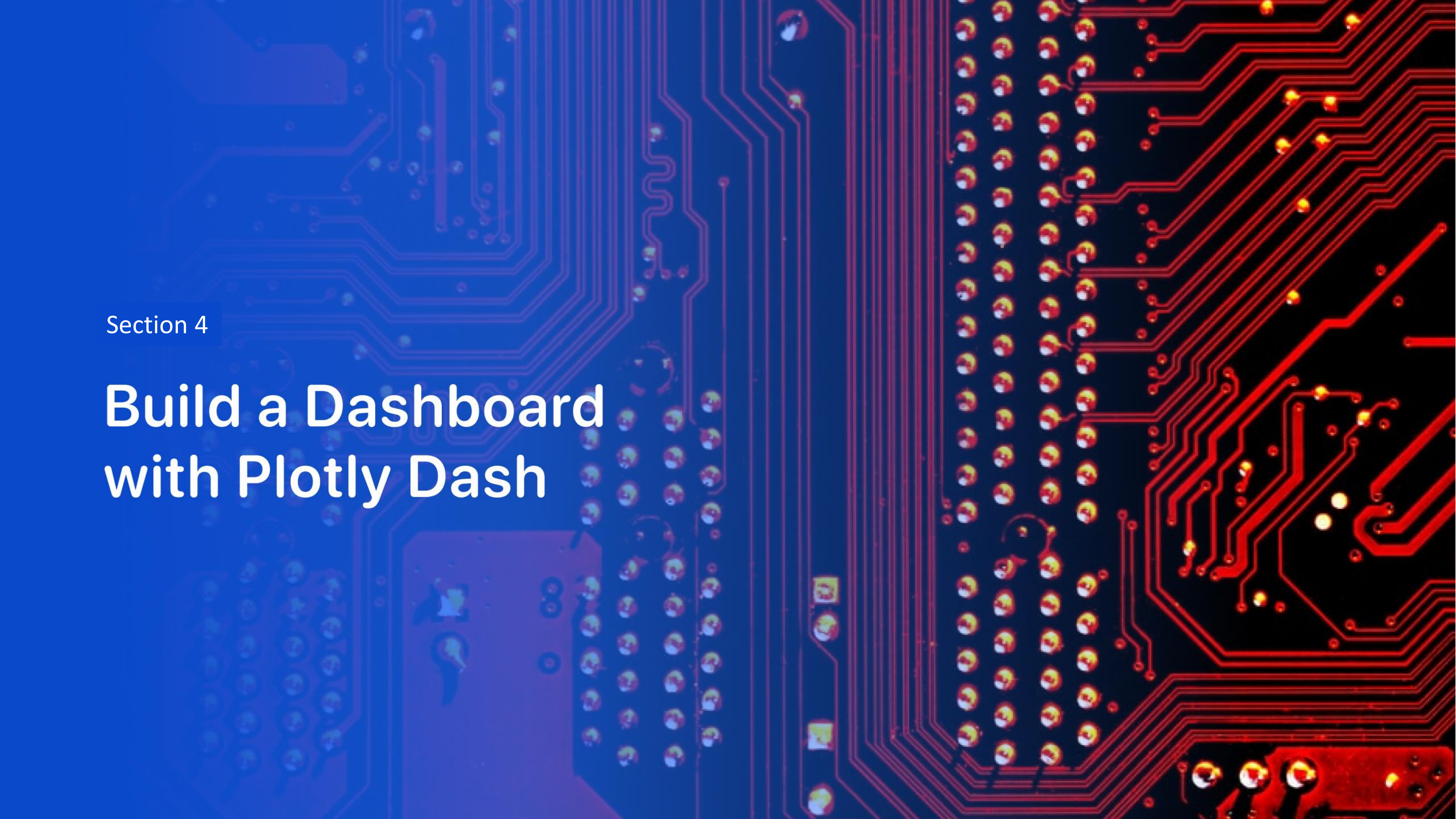
D: > GoogleDrive > Documents > Academic > Courses > IBMDataScience > Final_Module > scripts > git > 06_Interactive_Map_Folium.ipynb

Cluster mapping with Folium





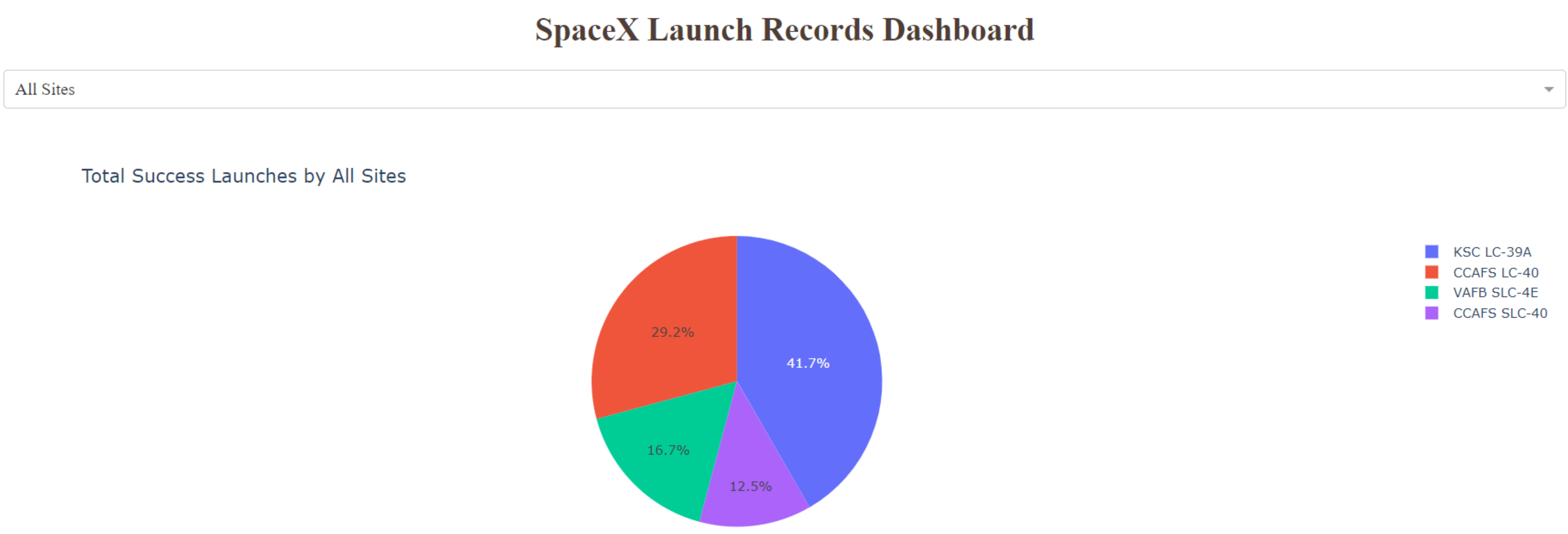




Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



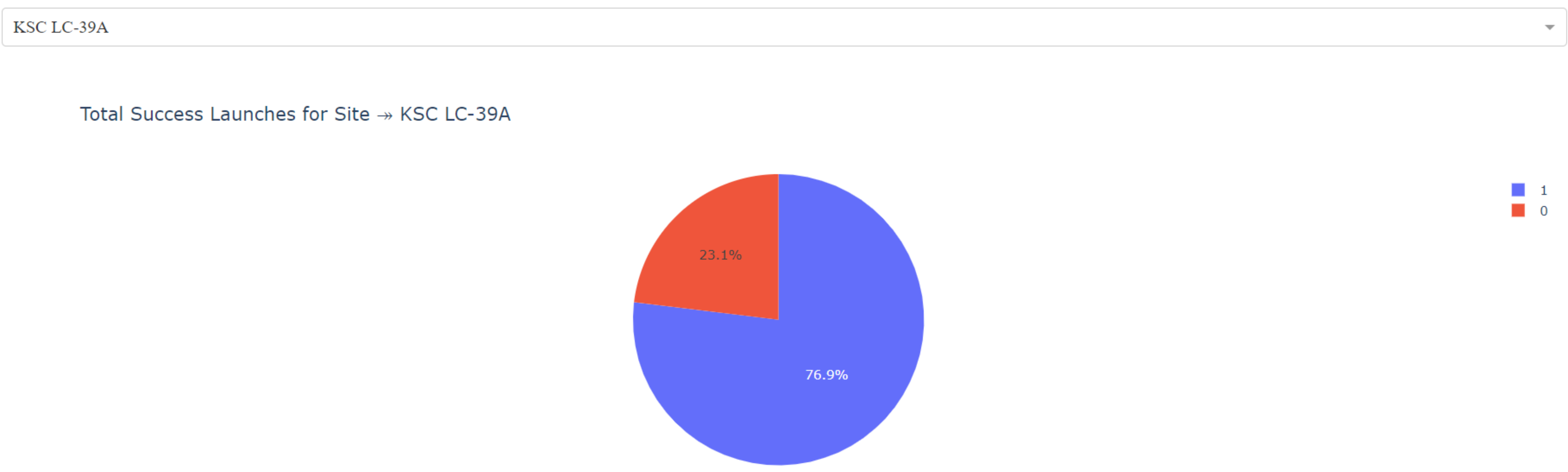
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by All Sites

- Most used launching site is KSC LC 39A

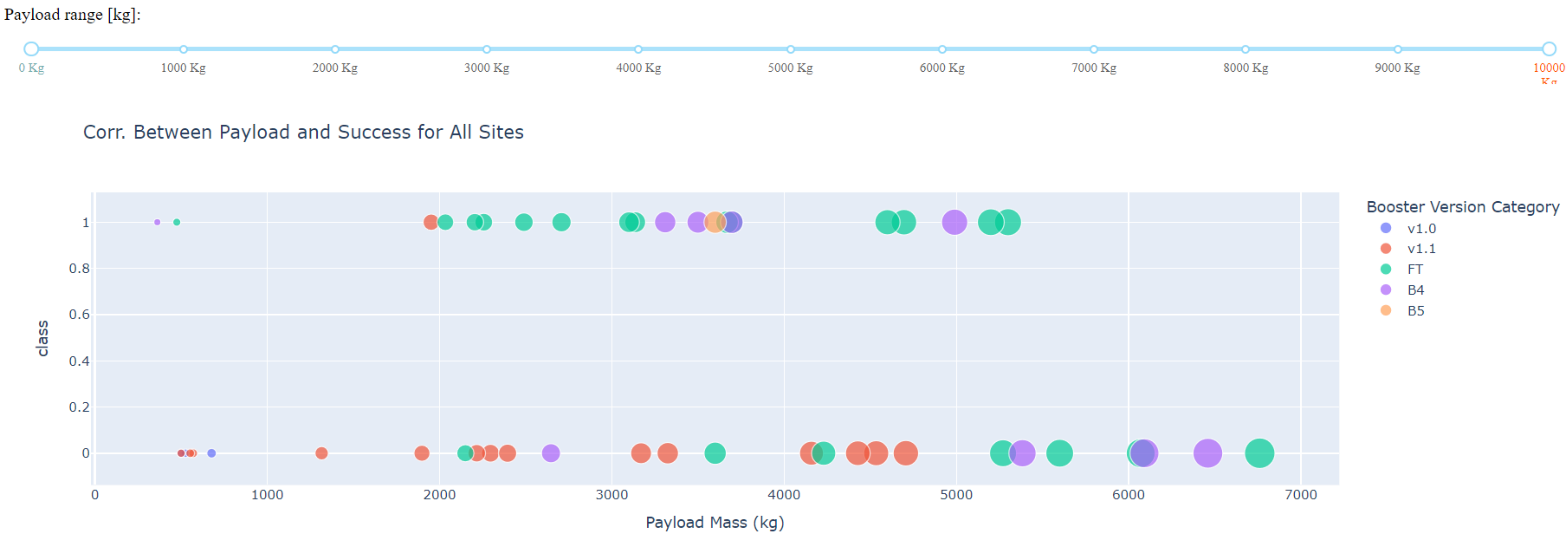
SpaceX KSC LC-39A Records Dashboard



KSC LC-39A

Total Success Launches for Site → KSC LC-39A

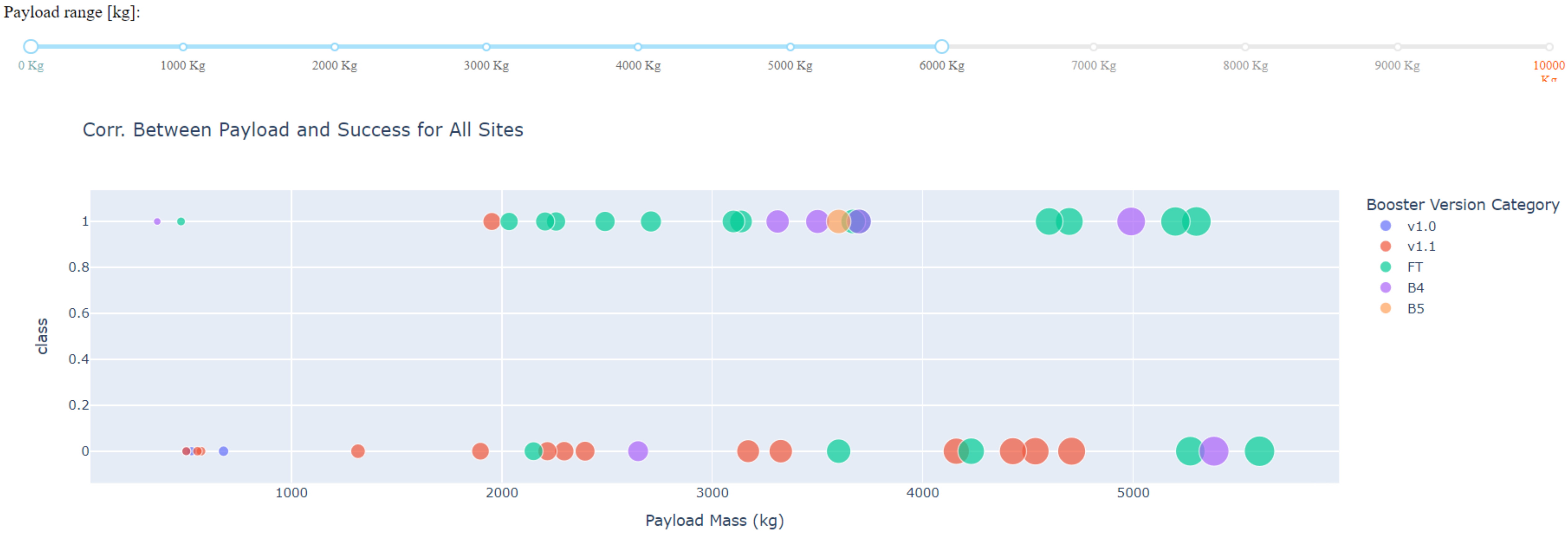
Dashboard with slider and scatterplot



Payload range [kg]:

Corr. Between Payload and Success for All Sites

Dashboard with slider and scatterplot



Payload range [kg]:

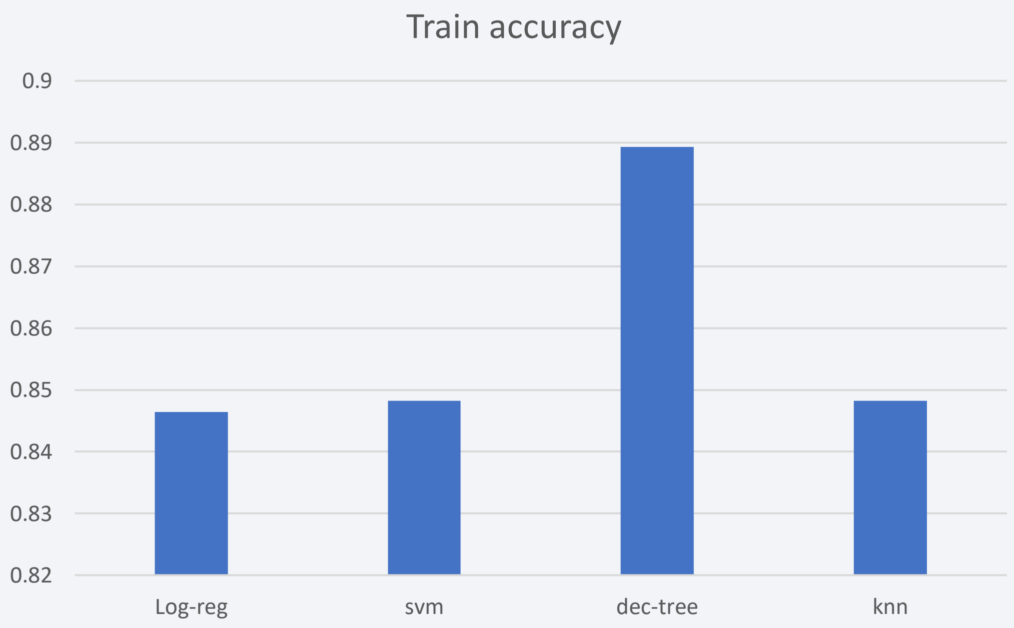
Dashboard with slider and scatterplot

- V1.1 has the most unsuccess launches in different Payload ranges.
- FT cover the widest payload range (both success and unsuccess)
- Failures are not related with payload, since they occur no matter the payload
- Successful launches appear to be prevented for payloads greater than 5.5 k

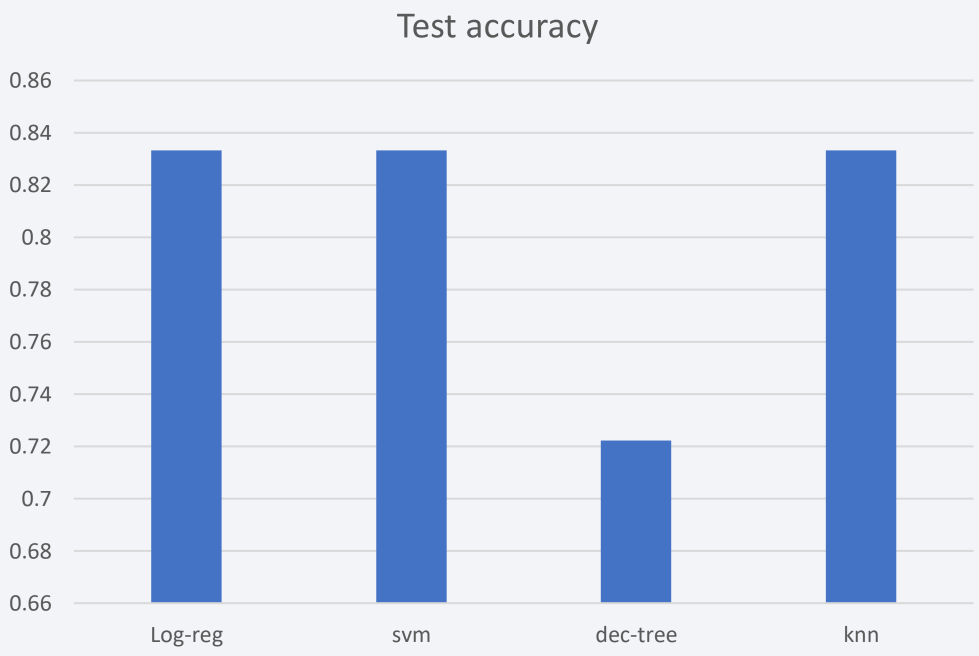
Section 5

Predictive Analysis (Classification)

Classification Accuracy



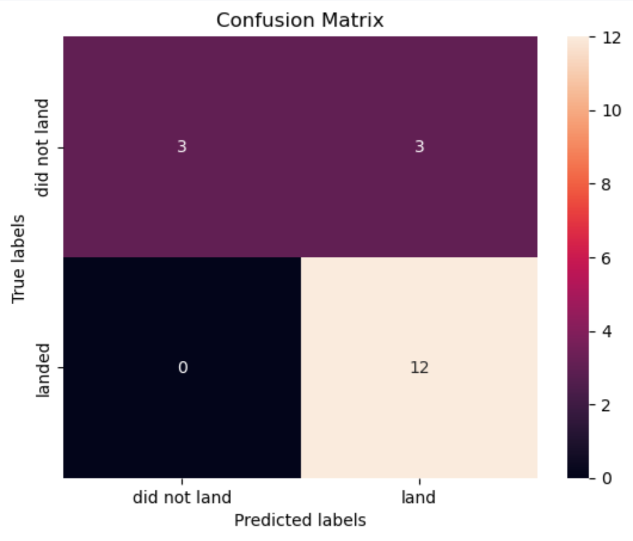
Train accuracy

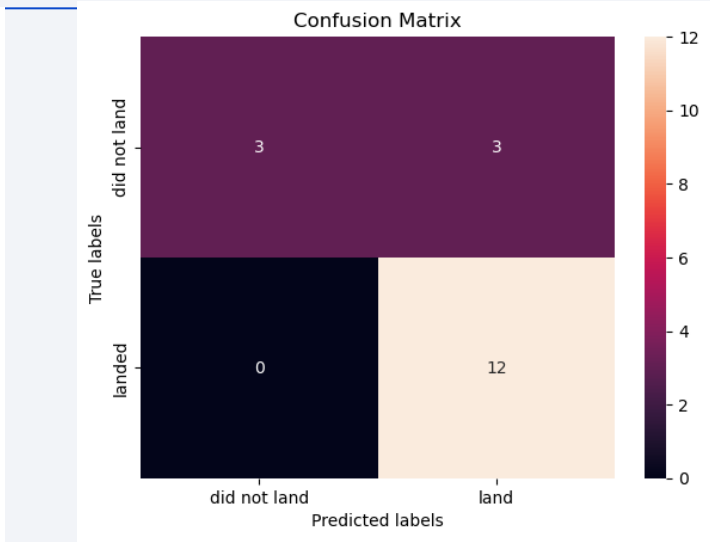


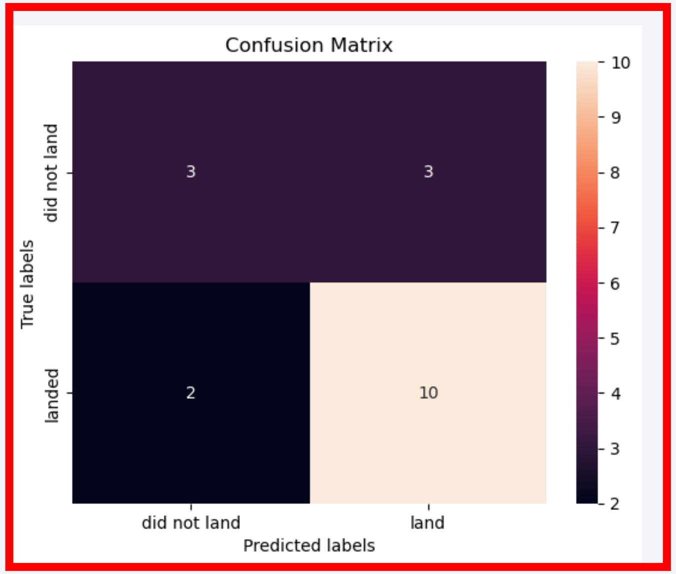
Test accuracy

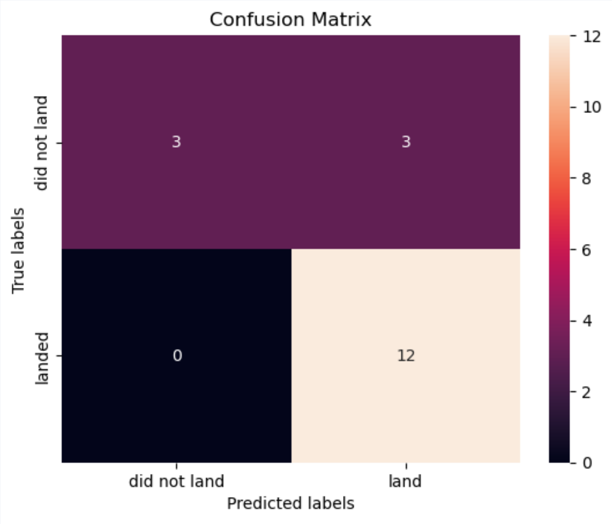
- Decision-tree classifier performs better in the training dataset, however it performs worse in the test dataset.

Confusion Matrix









Conclusions

- Best model to make predictions is Decision-Tree, in terms of overall accuracy in the training data
- Despite of its higher accuracy, all selected models perform relatively well (accuracy > 80%)
- Even though decision-tree performs well in the training dataset, with the test dataset if performs not that well, while the other models perform similarly
- I would choose SVM model since it has the second highest training accuracy and performs well in the test dataset.
- Additionally, the confusion matrices point in this direction. Although higher training accuracy is achieved by Decision-tree model, it has the more misclassification records.

Thank you!