IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Gerrit Lamm
11th April 2025

# Outline

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

# Executive Summary

- Task: Prediction whether Falcon9 rockets from SpaceX land successfully or not

- Main steps to address the ask
    1. Data Acquisition using the SpaceX API & historical data on Falcon9/Falcon9 heavy launches
    2. Data wrangling as a preparatory step for data analysis
    3. Exploratory data analysis (EDA) using Python & Visualization to retrieve first trends and what variables should be considered further
    4. Creation of interactive Visualization on the mission outcomes and surroundings of used launch sites
    5. Predict mission outcomes using machine learning algorithms, e.g. logistic regression

- Main fact causing an increase in success rate of Falcon9 missions is the experience in performing such missions

# Introduction

- SpaceX is a company providing several kinds of rockets that can be used in space missions
- Main advantage compared to competitors: costs per Falcon9 rocket launch
  - SpaceX: 62 million US dollars  Competitors: >165 million US dollars
  - Advantage provided by reusability of rockets first stage for future missions

- Question: Is it reasonable to start a competition with SpaceX in this field?

- Task: Predict the mission outcome based on mission parameters such as Payload or Launch Site using machine learning

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - 2 different sources & methodologies were used
    - 1st: the SpaceX API providing access to data directly from the competitor SpaceX
    - 2nd: Web scrapping was used to retrieve historical data on Falcon9 and Falcon9 heavy launches from the respective Wikipedia entry
    - In both cases, basic formatting and cleaning was applied afterwards to have suitable table to proceed with in the next steps
- Perform data wrangling
  - Python was used to get first insights, e.g. amount of missing values, amount of launches per launch site
  - mission outcome was classified into two numerical variables, (0 and 1)

# Methodology

## Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - SQL queries on the fully prepared dataset were executed to get further insights into the dataset
    - Names of unique launch sites, carried payloads, date of 1st successful landing
  - Visualizations were created using Seaborn library to investigate the relationship of certain provided information
    - e.g. has the failure rate of a mission improved over time? Is this in correlation with the used launch site?
    - Basic visualization created using the python library seaborn

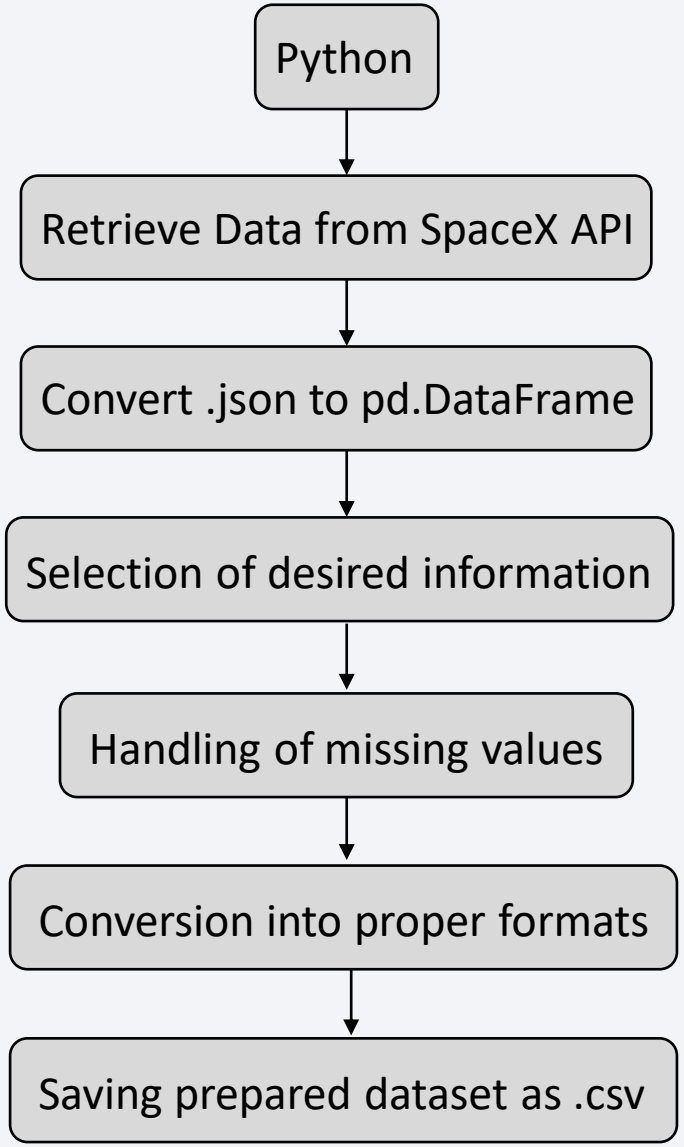
# Methodology

## Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
  - Folium: Creation of a geographical map, with launch sites and the distances to critical/useful objects/places around them highlighted
  - Plotly Dash: Creation of an interactive dashboard → information of carried payloads per launch sites or for all launch sites
- Perform predictive analysis using classification models with the scikit learn library
  - Split the available data into test and train subsets
  - Different models (logistic regression, support vector machine, decision tree, k-nearest neighbors) were trained using a cross-validation approach → GridSearchCV
  - Evaluation was done based on accuracy on predicting test data, and false positive rates

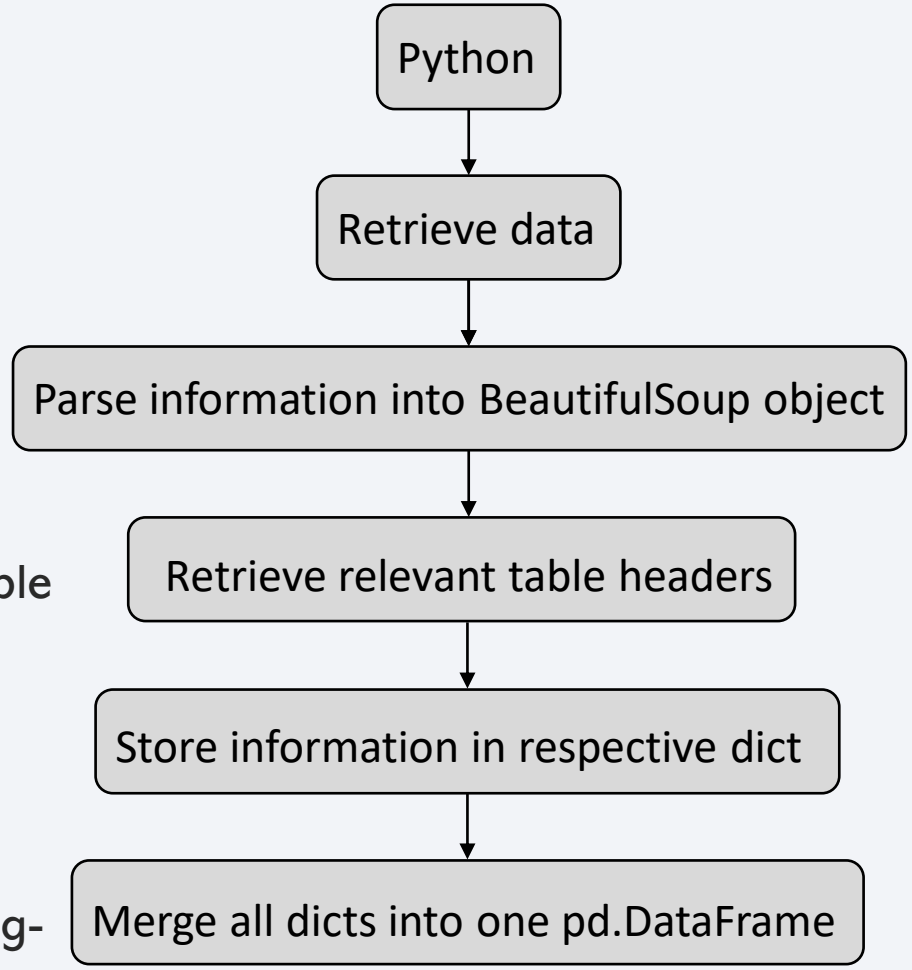
# Data Collection – SpaceX API

- Used python libraries: requests, numpy (np), pandas (pd)
- Helper functions were provided
- Requests used to retrieve data from provided URL as .json file [requests.get() → response.json()]
- .json file converted to pandas DataFrame [pd.json_normalize()]
- Only retrieved columns with necessary information, e.g. only Falcon9 data
- Final dataset was cleaned from missing values
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



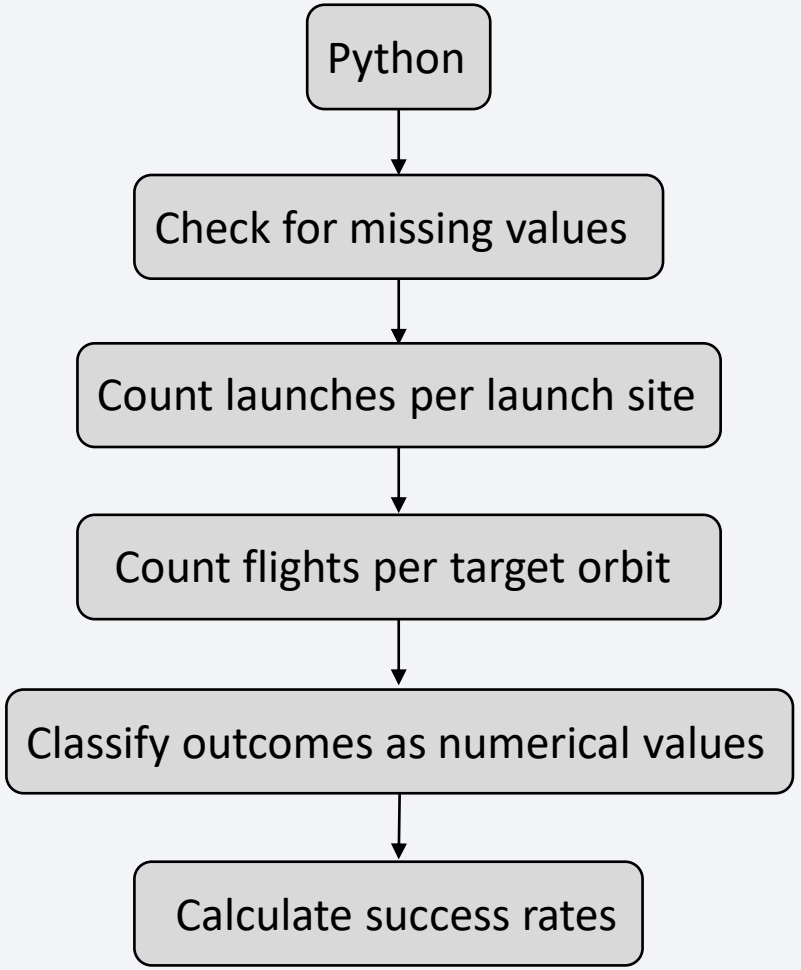
# Data Collection - Scraping

- Used python libraries: requests, pandas (pd), beautifulsoup
- Helper functions were provided
- Requests used to retrieve data from provided URL as BeautifulSoup object [requests.get() → BeautifulSoup()]
- Retrieve all tables [soup.find_all(tables)]
- Find all table headers → create empty dictionary for each table header → store respective information
- Convert final all the dictionaries into 1 pd.DataFrame
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



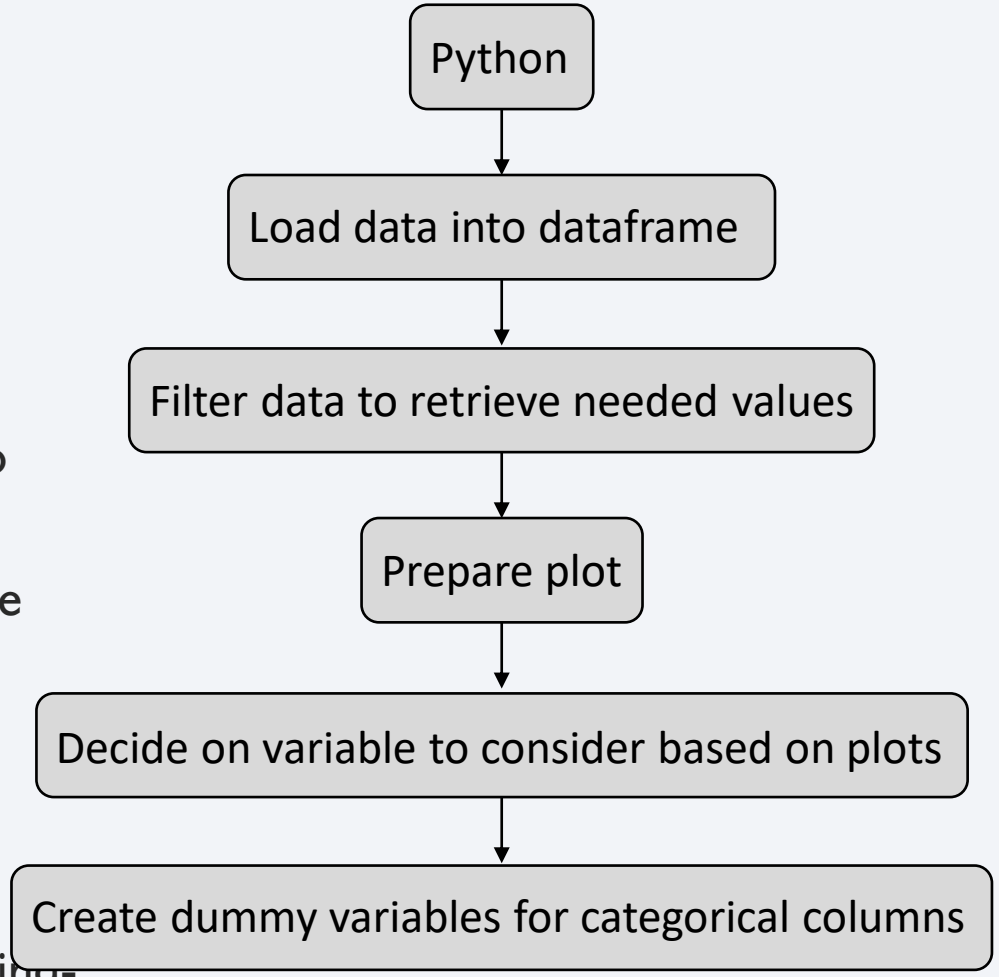
# Data Wrangling

- Used python libraries: requests, numpy (np), pandas (pd)
- Helper functions were provided
- Check for missing values using df.isnull() function
- Count rocket launches per launch sites using value_counts() function on ‘LaunchSite’
- Calculate the amount of flight to each orbit using value_counts() function on ‘Orbit’
- Classify good and bad outcomes as numerical values for machine learning → bad outcome =1; otherwise = 0
- Calculate success rates by calculation of mean values for row ‘Class’ using .mean() function
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



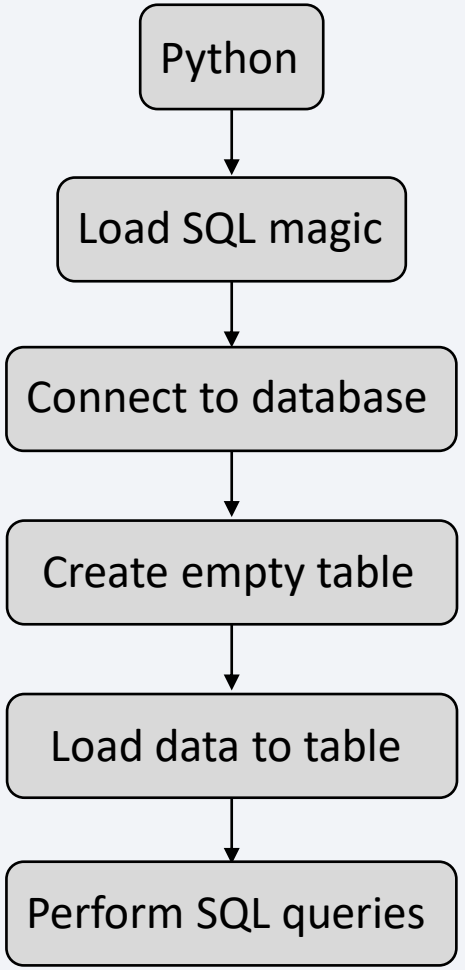
# EDA with Data Visualization

- Used python libraries: seaborn, numpy (np), pandas (pd)
- Load data into pd.DataFrame
- Filter data to have the needed data to prepare the plots to retrieve correlations
- Use hue='Class' to color data points in graphs according to the mission outcome
- Based on prepared graphs, decide which features should be further considered in the process to follow
- Create dummy variables for categorial columns pd.get_dummies()
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



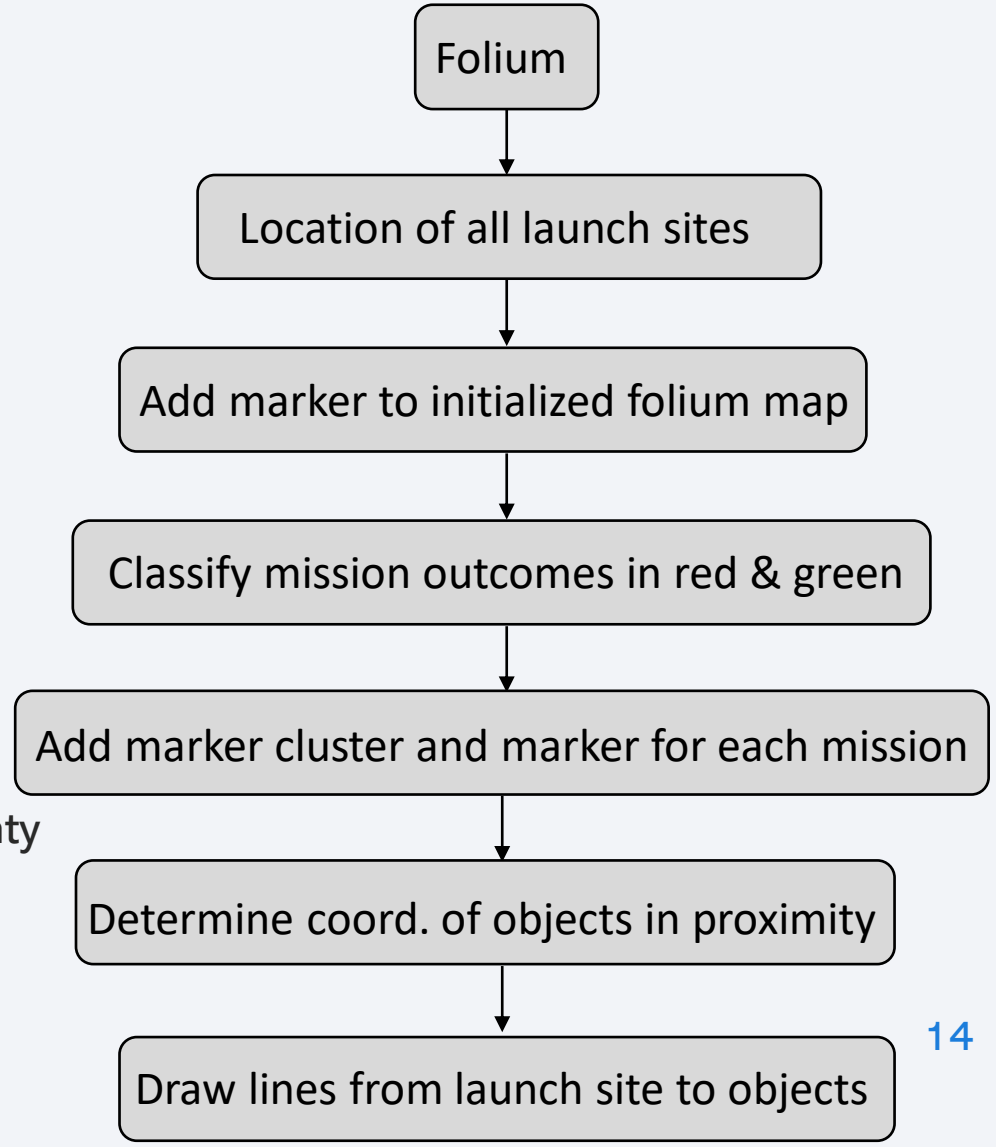
# EDA with SQL

- Used python libraries: pandas (pd), ipython-sql. sqlalchemy, sql magic
- Connect to database: con = sqlite3.connect(), cur = con.cursor()
- Read data and send to sql: pd.read_csv, df.to_sql()
- %sql for SQL magic in every code cell
- Perform required queries to retrieve information
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



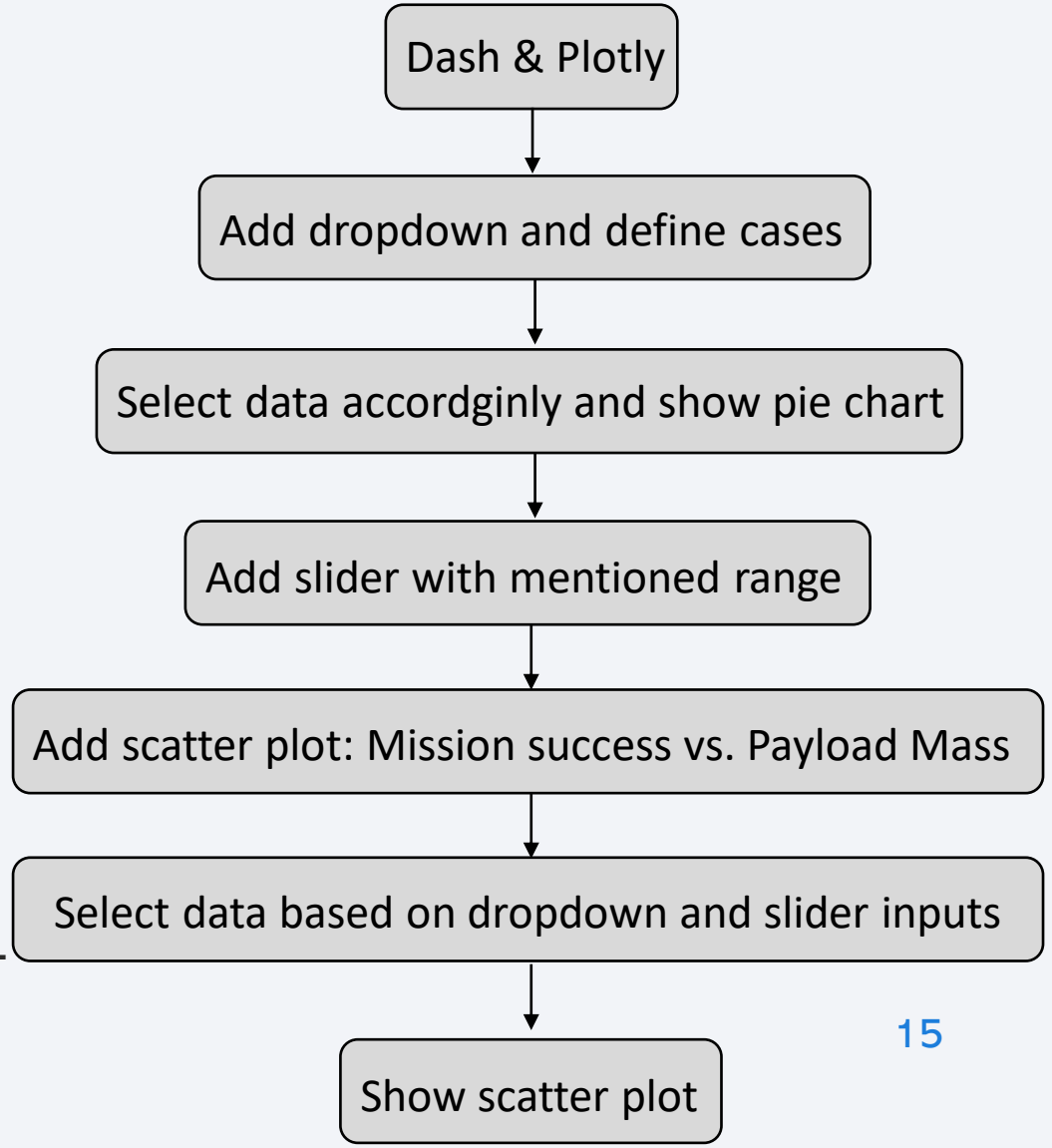
# Build an Interactive Map with Folium

- Used python libraries: folium, pandas (pd)
- Retrieve locations of all launch sites from data
- Initialize folium map
- Use circle marker to highlight launch sites
- Classify mission outcomes in colors
  - Positive: green negative:red
- Add marker cluster to map, and add markers for all missions
- Determine locations of closest highway, railway, city, coast
- Draw line from launch site to considered objects in the proximinty
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



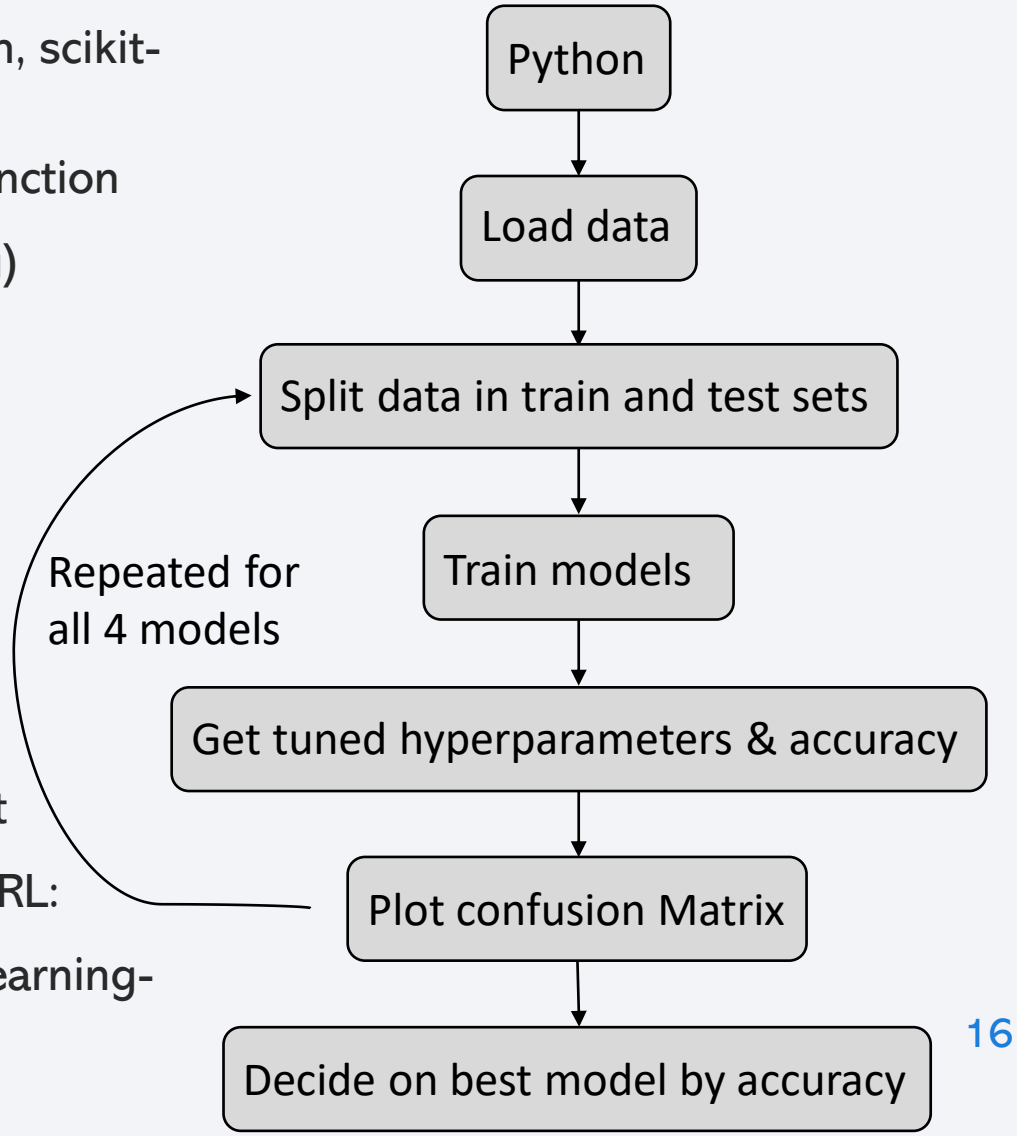
# Build a Dashboard with Plotly Dash

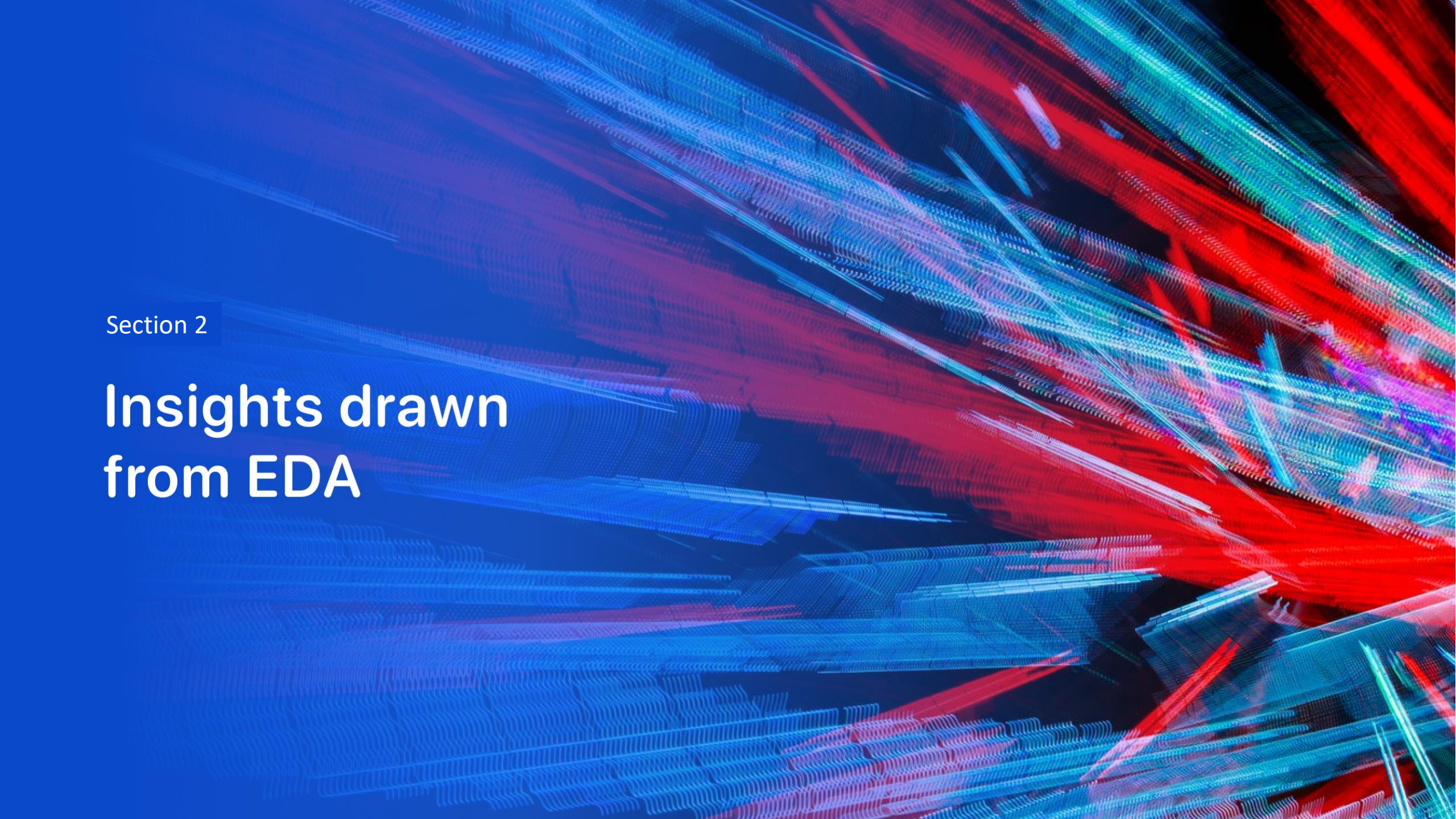
- Used python libraries: Plotly und Dash
- Add dropdown field to choose between individual launch sites or all launch sites combined
  - Data needs to be selected accordingly
- Display results as a pie chart
- Add slider (0 kg to 10000 kg) to adjust considered payload mass range
- Add a scatter plot: Mission success vs. Payload Mass
  - Data selection based on dropdown and slider inputs
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project



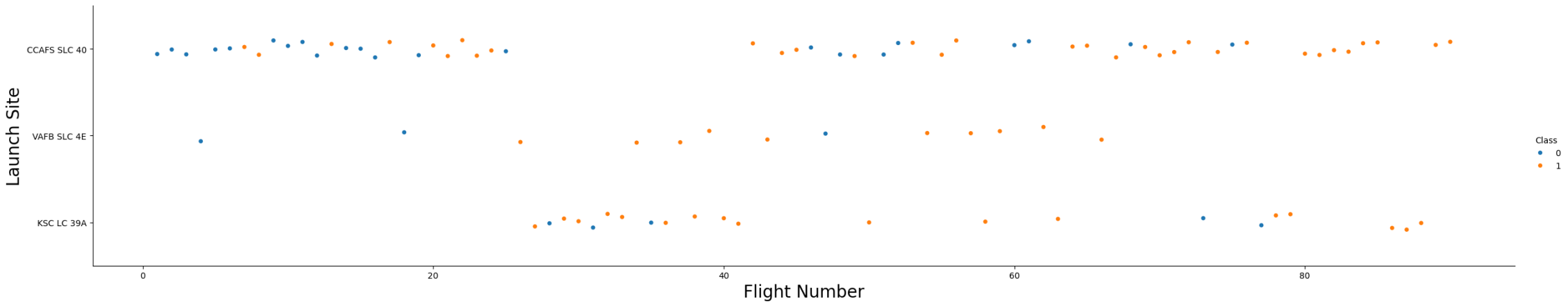
# Predictive Analysis (Classification)

- Used python libraries: numpy (np), pandas (pd), seaborn, scikit-learn
- Load data and standardize it using .StandardScaler() function
- Split dataset in train and test datasets (20% for testing)
  - train_test_split()
- Define parameters and model type for GridSearch
- Using 10 iterations of GridSearchCV to find best hyperparameters
- Calculate accuracy score and plot confusion matrix
- Repeat for all models
- Compare accuracy and decide which model fits the best
- The notebook can be access via the following GitHub URL:
- https://github.com/gerrla/Data-Science-and-Machine-Learning-Capstone-Project




Section 2
Insights drawn from EDA

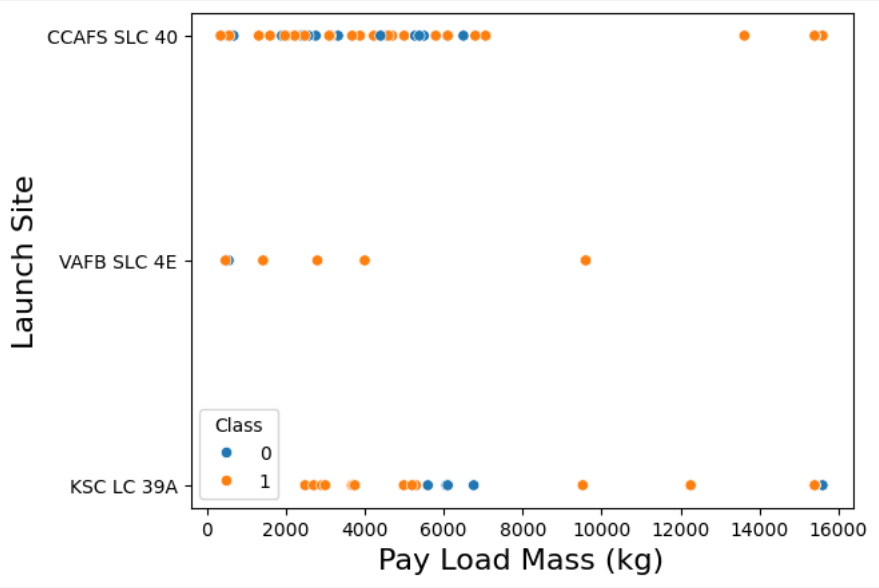
# Flight Number vs. Launch Site




- Majority of performed flights started at CCAFS SLC 40 launch site
- Success rate of performed flights increased over time → more problems in the beginning
- CCAFS SLC 40 seems to have a higher failure rate, but one needs to consider the beginning of the project and the total amount of performed flights per launch sites
- No strong impact on launch site on mission outcome

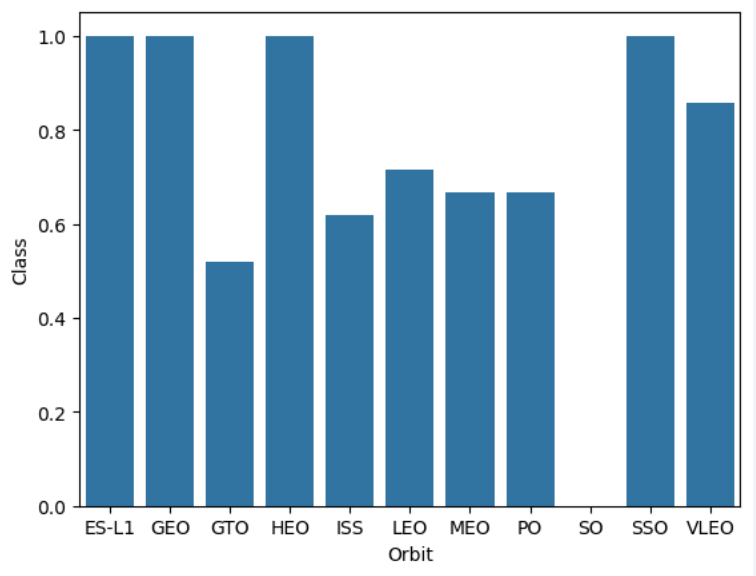
# Payload vs. Launch Site

- Majority of performed flights has a carried payload < 8000 kg
- VAFB SLC 4E launch site not used for flights with payloads. > 10000 kg
- CCAFS SLC 40 was most frequently used



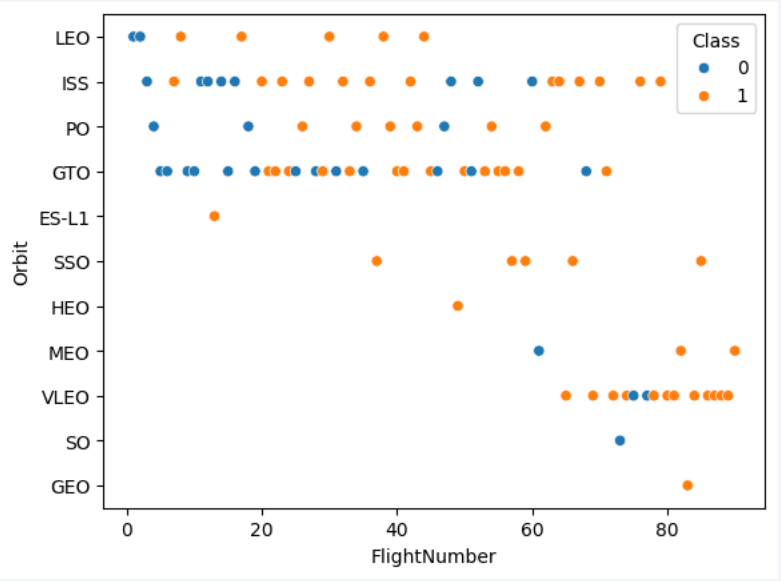
# Success Rate vs. Orbit Type

- Success rate of flights seems to depend on the used launch site
- Orbits: ES-L1, GEO, HEO, SSO have 100% success rate
- Orbit SO has 0% success rate
- This data needs to be carefully interpretated since the total amount of flights per launch site is not considered!
    - If extremely low amount of flights for one launch site, the retrieved success rate is not statistically significant



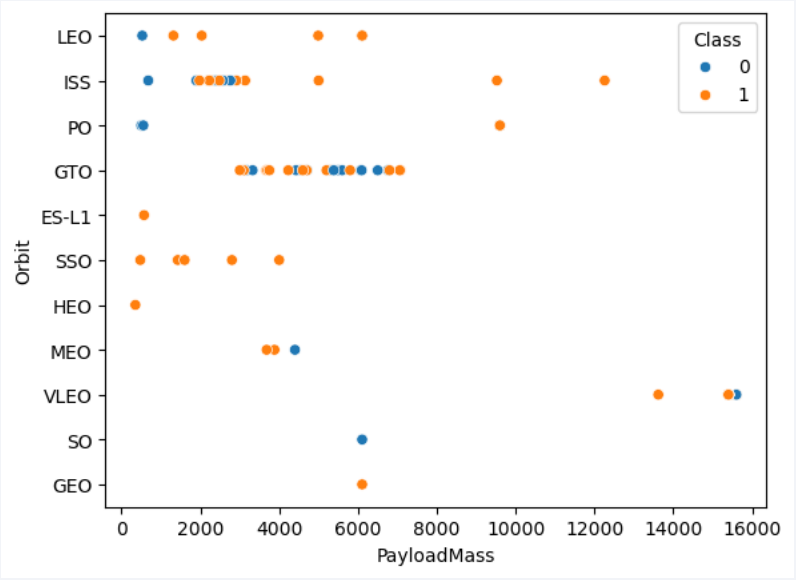
# Flight Number vs. Orbit Type

- Success of flights to LEO orbit is related to the number of flights to this orbit
- For GTO orbit this is not the case
- Trend continues that the failure rate is higher for lower flight numbers
- Concerns raised from one slide before are nicely illustrated
  - Only 1 flight to GEO orbit → successful → 100% success rate
  - Only 1 flight to SO orbit → failure → 0% success rate
  - Data on both orbits is not statistically relevant



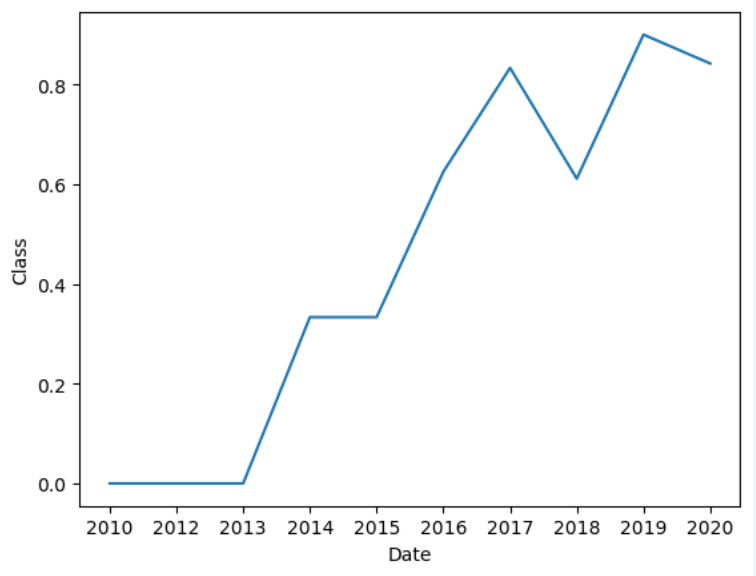
# Payload vs. Orbit Type

- High payload flights (> 8000 kg) only performed to orbits: ISS, PO, VLEO
- Success rate for GTO orbit is quite distributed which makes predictions for this orbit harder



# Launch Success Yearly Trend

- Success rate significantly increases with advancement of the project
- Experience gained from the various performed flights helps in increasing the success rate



# All Launch Site Names

- Task: Find the names of the unique launch sites

```
[11]: %config SqlMagic.style = 'DEFAULT'
%sql select Distinct "Launch_Site" from SPACEXTBL;
```

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

[11]:

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

- Using Distinct command on column “Launch_Site” to only retrieve unqiue launch sites and not the launch sites for each recorded flight

# Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`

Task 2

Display 5 records where launch sites begin with the string 'KSC'

```
[9]: %sql select * from SPACEXTBL where "Launch_Site" like 'KSC%' Limit 5;
```

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

[9]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2017-02-19 | 14:39:00 | F9 FT B1031.1 | KSC LC-39A | SpaceX CRS-10 | 2490 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-03-16 | 6:00:00 | F9 FT B1030 | KSC LC-39A | EchoStar 23 | 5600 | GTO | EchoStar | Success | No attempt |
| 2017-03-30 | 22:27:00 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO | SES | Success | Success (drone ship) |
| 2017-05-01 | 11:15:00 | F9 FT B1032.1 | KSC LC-39A | NROL-76 | 5300 | LEO | NRO | Success | Success (ground pad) |
| 2017-05-15 | 23:21:00 | F9 FT B1034 | KSC LC-39A | Inmarsat-5 F4 | 6070 | GTO | Inmarsat | Success | No attempt |

- 'KSC%' was used to account for the remaining part of the launch site name
- Limit was set to 5 to only retrieve 5 entries

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
[15]: %sql select sum("PAYLOAD_MASS__KG_") from SPACEXTBL where Customer is "NASA (CRS)"
```

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

[15]:

| sum("PAYLOAD_MASS__KG_") |
|---|
| 45596 |

- Sum(“PAYLOAD_MASS__KG_) was used to calculate the total payload mass
- Customer was filtered for “NASE (CRS)” using the where function

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Task 4 ¶

Display average payload mass carried by booster version F9 v1.1

```
[17]: %sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTBL where "BOOSTER_VERSION" is "F9 v1.1"
```

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

[17]:

| AVG("PAYLOAD_MASS__KG_") |
|---|
| 2928.4 |

- AVG("PAYLOAD_MASS__KG_) was used to calculate the average payload mass
- "BOOSTER_VERSION was filtered for "F9 v1.1" using the where function

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.Present your query result with a short explanation here

**Task 5**

List the date where the succesful landing outcome in drone ship was acheived.

*Hint:Use min function*

```
[16]: %sql select min(DATE) from SPACEXTBL where "LANDING_OUTCOME" LIKE "Success (drone ship)";
 * sqlite:///my_data1.db
Done.
[16]: min(DATE)
2016-04-08
```

- Min(Date) was used to retrieve the first date for such a case
- Landing outcome was specified to “Success (drone ship)” using the where function

# 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

**Task 6**

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
[31]: %sql select "Booster_Version" from SPACEXTBL where "LANDING_OUTCOME" = "Success (ground pad)" and "PAYLOAD_MASS__KG_" between "4000" and "6000";
 * sqlite:///my_data1.db
Done.
```

[31]:

| Booster_Version |
|---|
| F9 FT B1032.1 |
| F9 B4 B1040.1 |
| F9 B4 B1043.1 |

- Booster Version was specified as output
- Landing outcome was specified as “Success (ground pad)” using where function
- PAYLOAD_MASS__KG_ was specified to meet the given range using the between function

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[23]: %%sql
SELECT number_of_success_outcomes, number_of_failure_outcomes FROM (SELECT COUNT(*) AS number_of_success_outcomes FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%') success_table,
(SELECT COUNT(*) number_of_failure_outcomes FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%') failure_table
```

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

[23]:

| number_of_success_outcomes | number_of_failure_outcomes |
|---|---|
| 100 | 1 |

- 2 subtables (success_table & failure_table) were created since otherwise counting two different outcomes in the same row is not possible
- Count function was used to retrieve the amount of such missions
- Mission outcome was specified with flexibility for additional terms, e.g. (ground pad)

# Boosters Carried Maximum Payload

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

Task 8

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
[49]: %sql SELECT DISTINCT("Booster_Version") FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

[49]:

| 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 |

- Only retrieve unique results → DISTINCT("BOOSTER_VERSION")
- Set carried Payload mass to the maximum value within this column using a subquery

# 2015 Launch Records

- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

**Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.**

```
[53]: %sql SELECT SUBSTR(Date,6,2),"Booster_Version","Landing_Outcome","Launch_Site" FROM SPACEXTBL WHERE SUBSTR(Date,0,5)='2017' AND "Landing_Outcome" = "Success (ground pad)";
 * sqlite:///my_data1.db
Done.
```

[53]:

| SUBSTR(Date,6,2) | Booster_Version | Landing_Outcome | Launch_Site |
|---|---|---|---|
| 02 | F9 FT B1031.1 | Success (ground pad) | KSC LC-39A |
| 05 | F9 FT B1032.1 | Success (ground pad) | KSC LC-39A |
| 06 | F9 FT B1035.1 | Success (ground pad) | KSC LC-39A |
| 08 | F9 B4 B1039.1 | Success (ground pad) | KSC LC-39A |
| 09 | F9 B4 B1040.1 | Success (ground pad) | KSC LC-39A |
| 12 | F9 FT B1035.2 | Success (ground pad) | CCAFS SLC-40 |

- Results filtered using where function and the correct way to display year = 2017 as provided by the note

# 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

**Task 10**

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

```
[57]: %sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT("Landing_Outcome") DESC;
```

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

[57]:

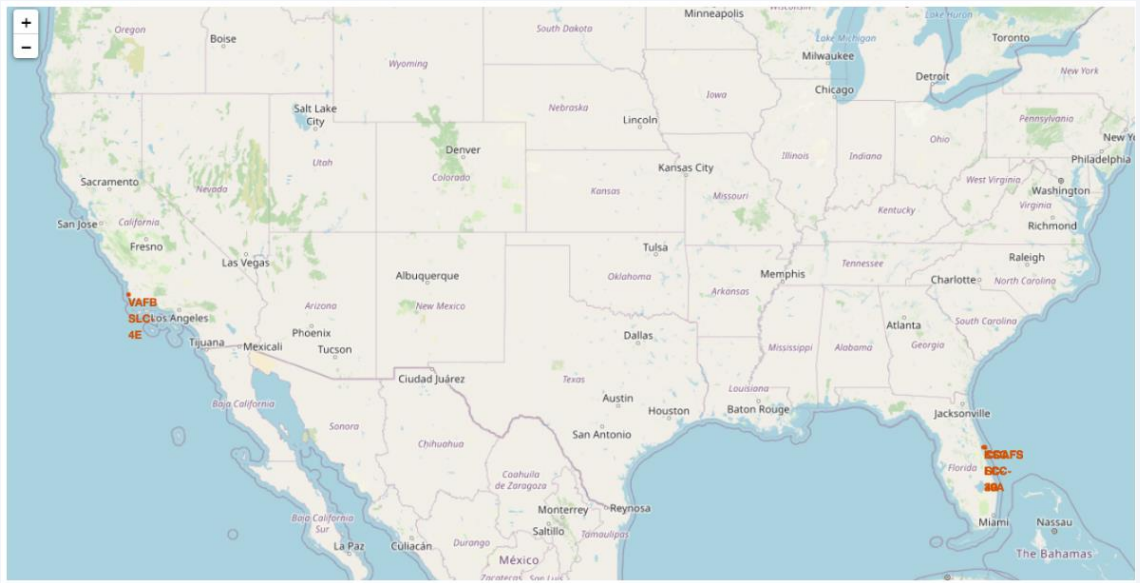
| Landing_Outcome | COUNT("Landing_Outcome") |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

- Date specified to meet the given time window (2010-06-04 to 2017-03-20)
- Grouped by Landing outcome to count the amount of each of them
- Order by amount of the specific landing outcome in descending order using


Section 3
Launch Sites
Proximities Analysis

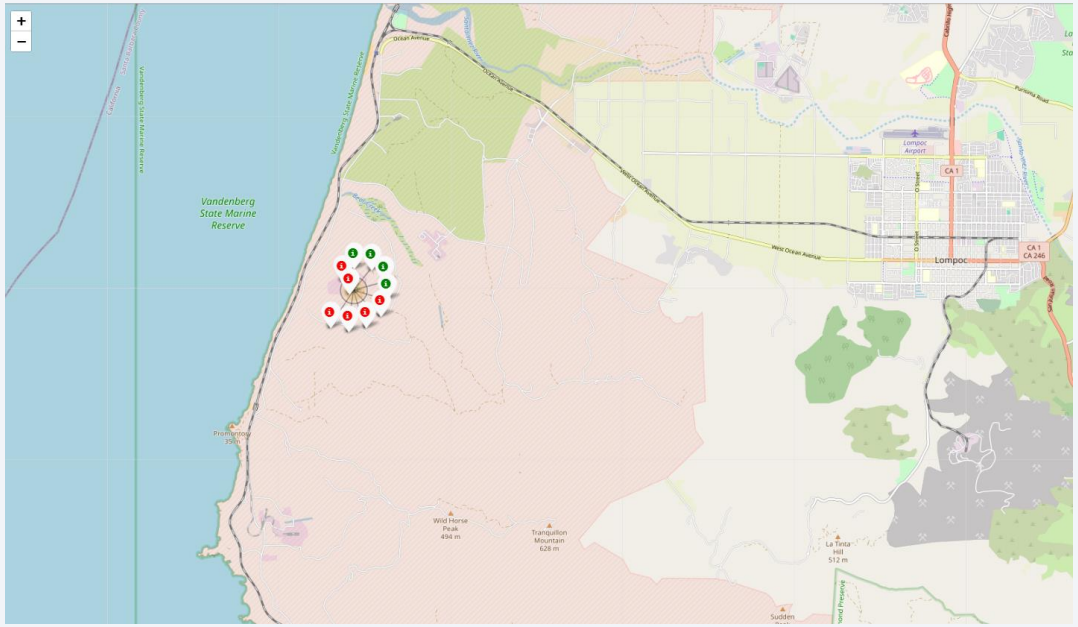
# Locations of Launch Site in the U.S.

- Needed (Folium) objects:
  - Initialize folium map folium.map()
  - Folium.Circle for marker
  - Folium.map.Marker() function to add indicators with
    - Coordinates
    - icon
    - Name
    - color



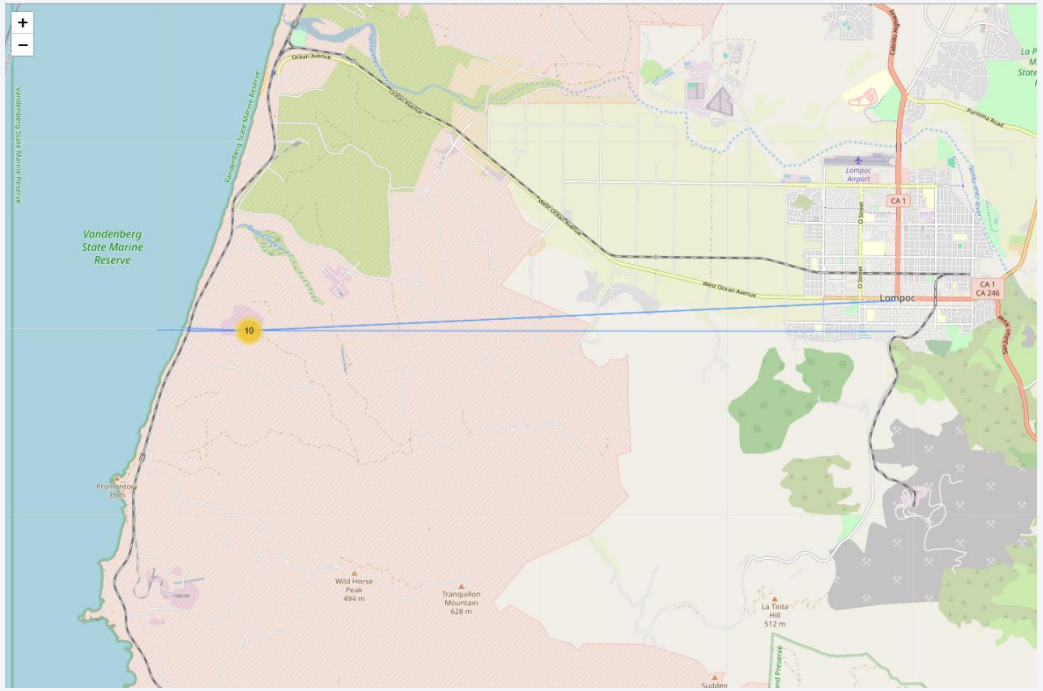
# Mission Outcomes at VAFB SLC 4E Launch Site

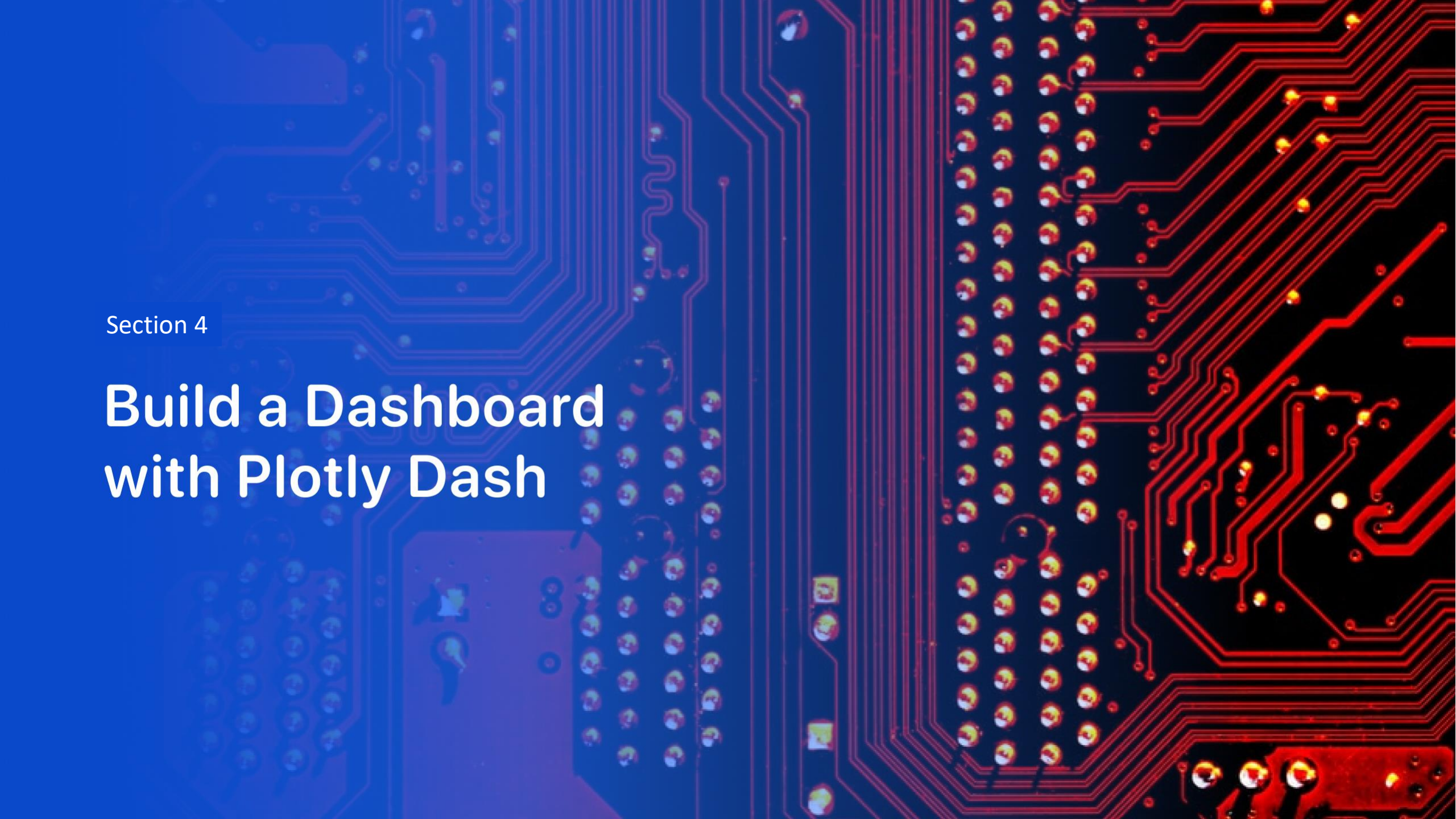
- Needed (Folium) objects:
  - Classification of mission outcomes as green (successful) or red (failure)
  - Marker_cluster in folium
  - Folium.map.Marker() function to add indicators with
    - Coordinates
    - Name for each marker
    - Color according to mission outcome



# Proximities of the VAFB SLC 4E Launch Site

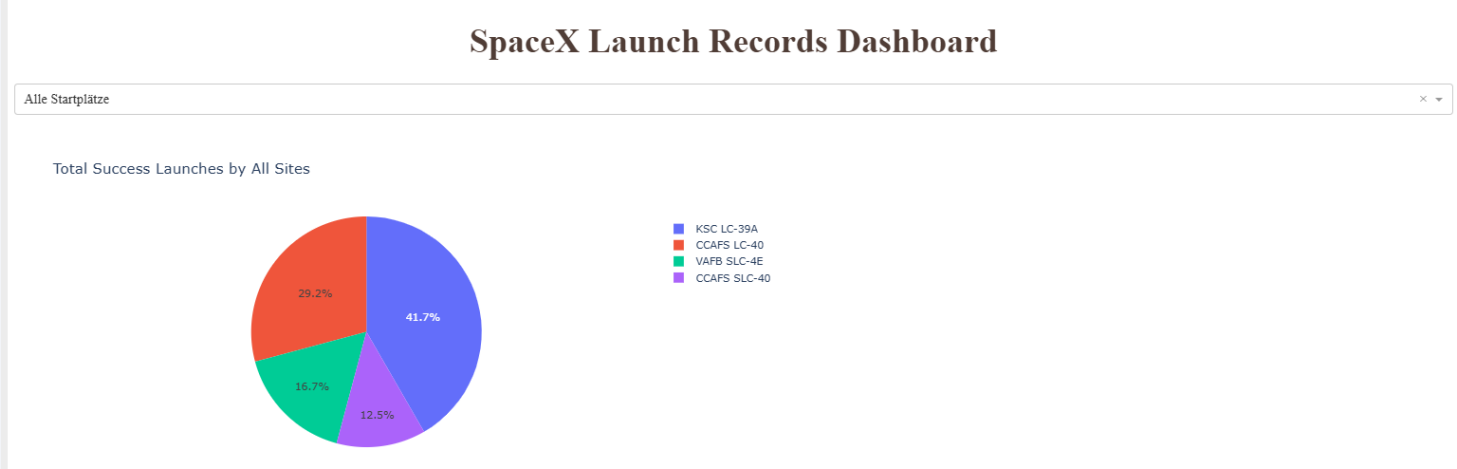
- Needed (Folium) objects:
  - Coordinates for nearest City, nearest Railway, and nearest Highway
  - Folium.Polyline() function to draw lines from coordinates of launch site to the coordinates of the nearest object
  - Use map.add_child() to add the lines after they are created




Section 4
Build a Dashboard with Plotly Dash

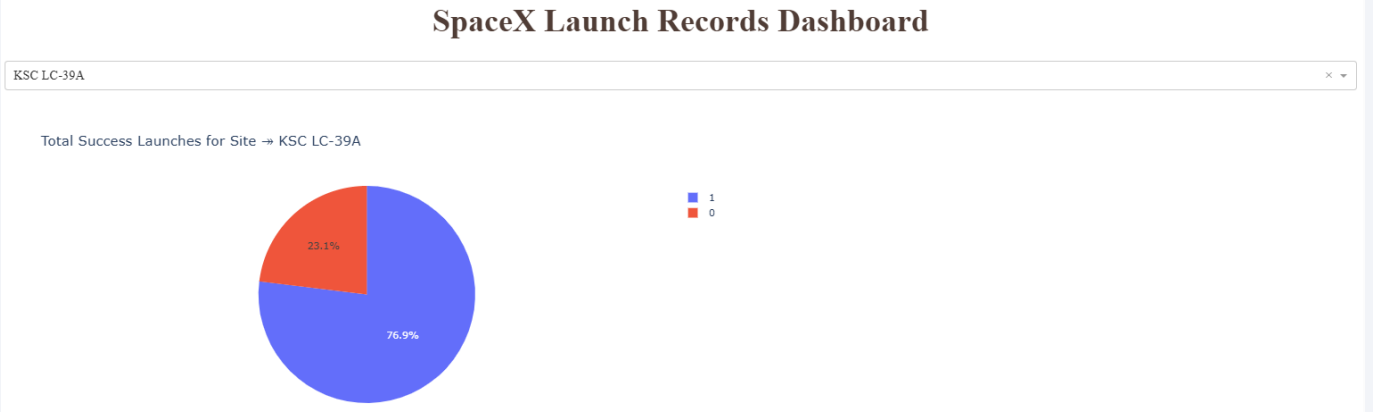
# Distribution of Launch Sites in Successful Missions

- Pie chart representation shows the distribution of launch sites among all successful missions
- 2 launch sites have significantly higher contribution to all successful missions
  - KSC LC-39A (43.7%)
  - CCAFS LC-40 (29.2%)
- Resembles that both launch sites have the highest number of performed flights



# The Most Successive Launch Site

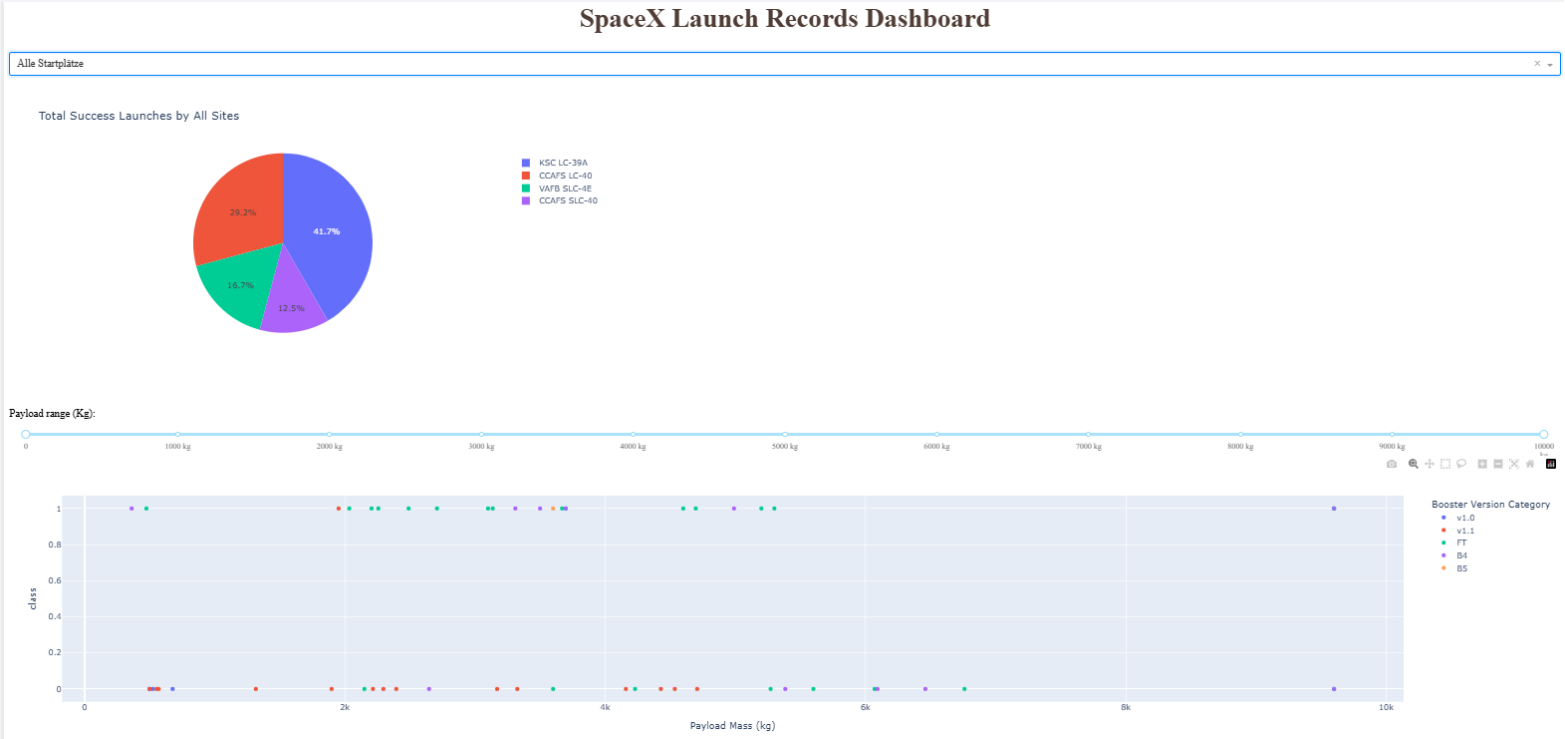
- 0 or blue color represents successful mission
- 1 or red color represents failures
- KSC LC-39A has the highest success rate of all considered launch site with 76.9%



# Impact of Payload Mass on the Mission Outcome - I

0000 kg ≤ Payload Mass ≤ 10000 kg

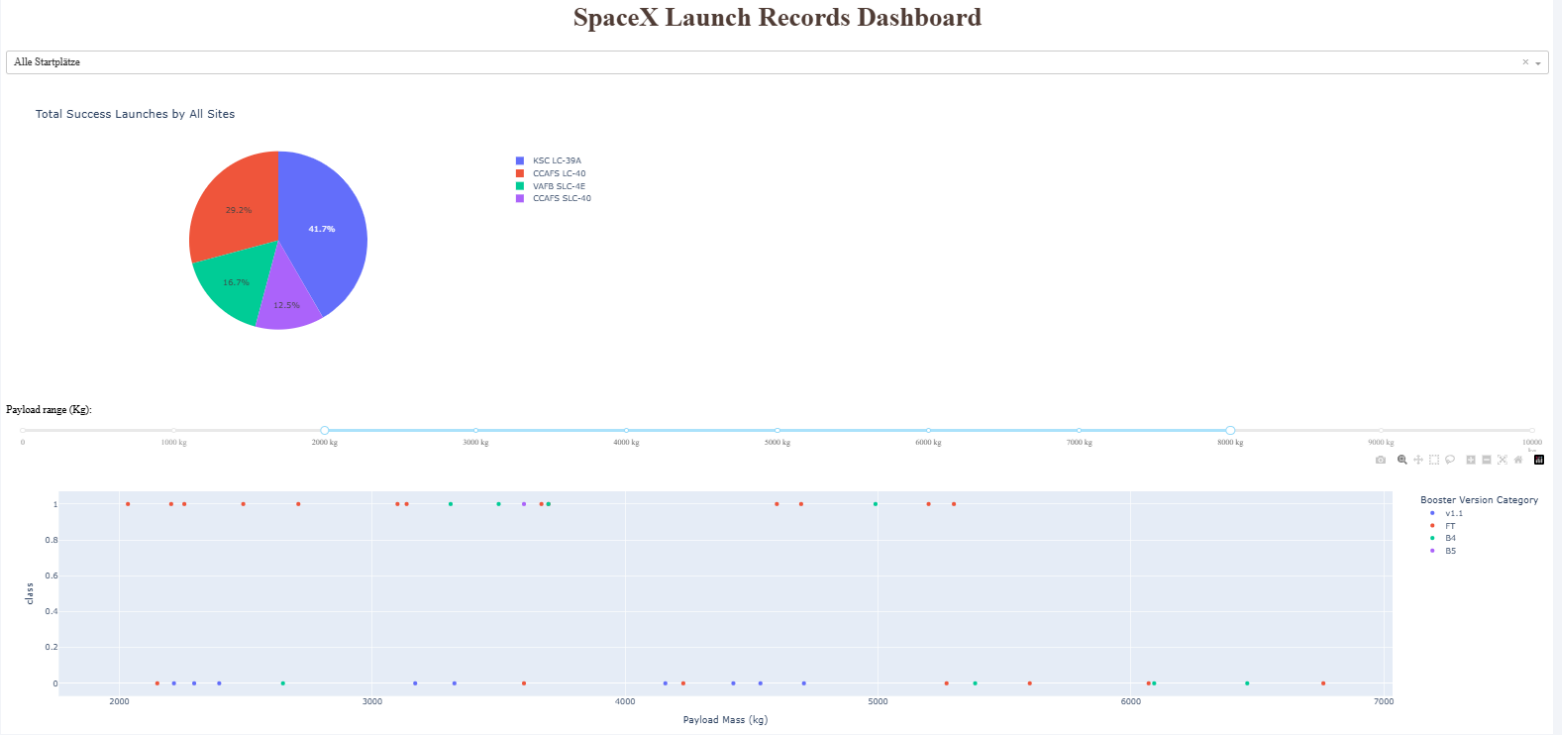
- The scatter plot provides distribution of failures (0) or successes (1) depending on the considered payload mass range



# Impact of Payload Mass on the Mission Outcome - II

2000 kg ≤ Payload Mass ≤ 8000 kg

- The scatter plot provides distribution of failures (0) or successes (1) depending on the considered payload mass range

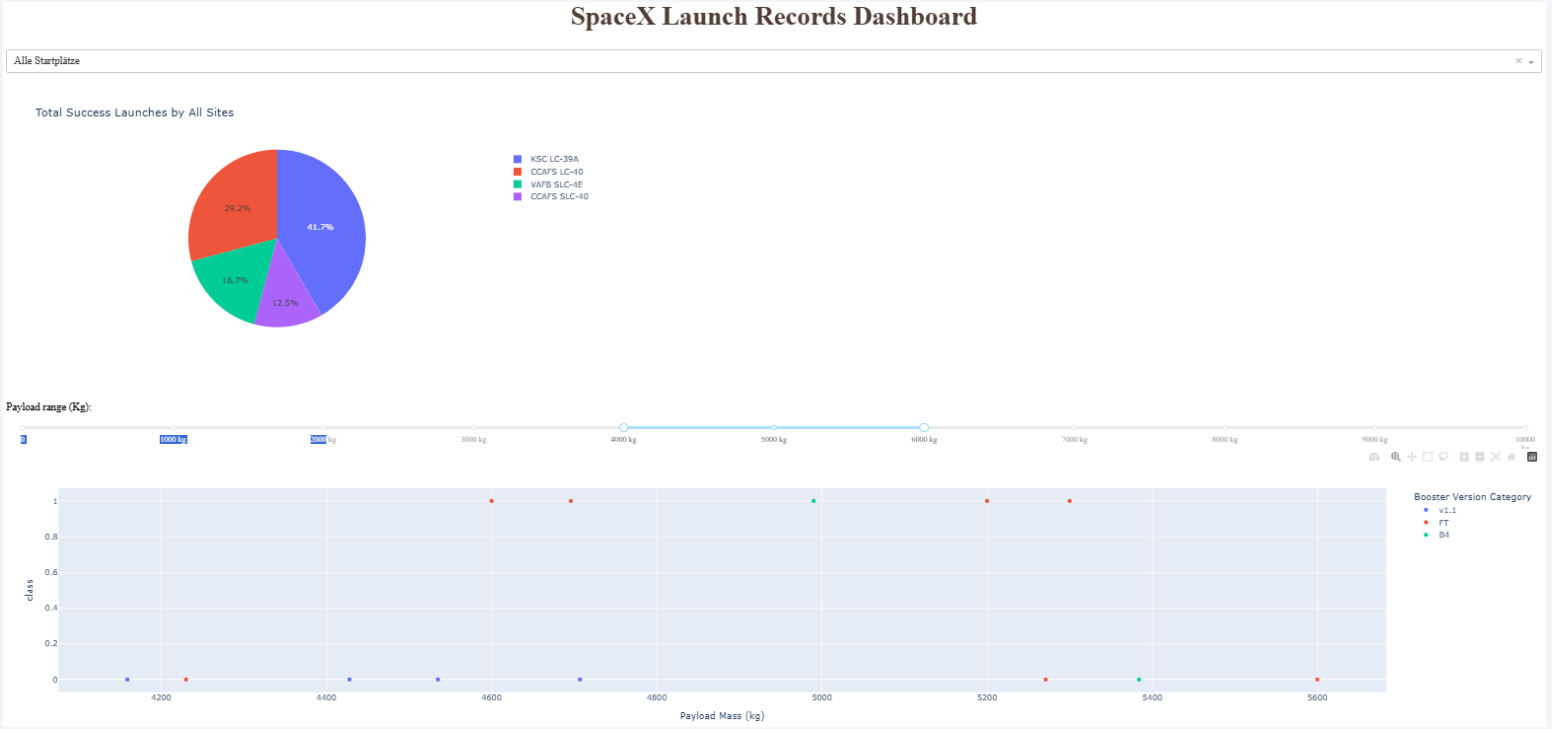


# Impact of Payload Mass on the Mission Outcome - III

4000 kg ≤ Payload Mass ≤ 6000 kg

- title with an appropriate The scatter plot provides distribution of failures (0) or successes (1) depending on the considered payload mass range

- All 3 scatter plots together do not show a direct correlation in between payload mass and success rate

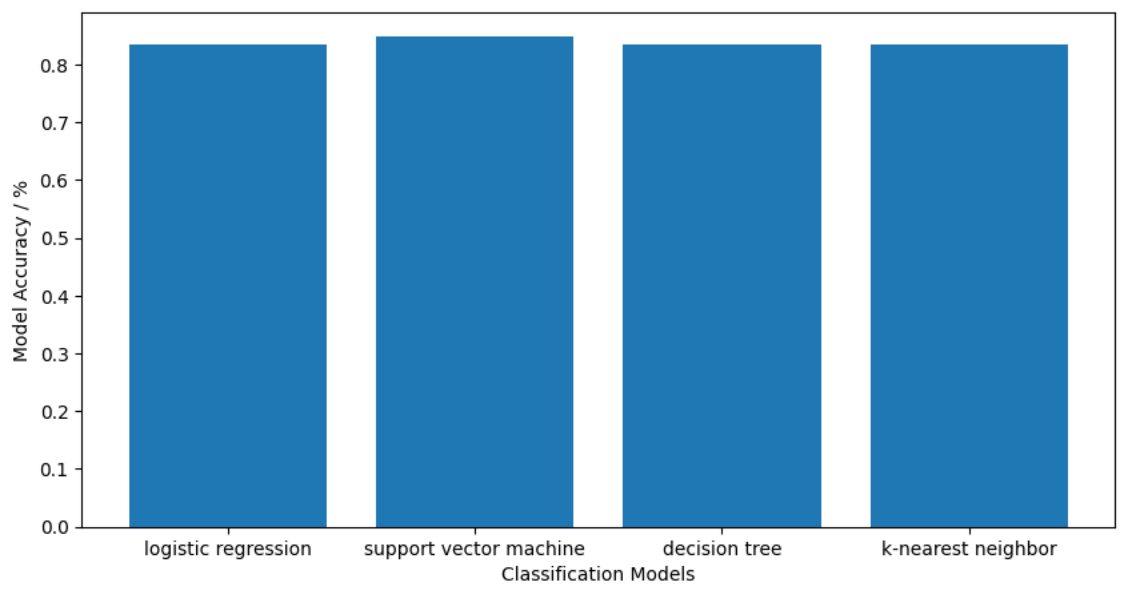


Section 5

# Predictive Analysis (Classification)

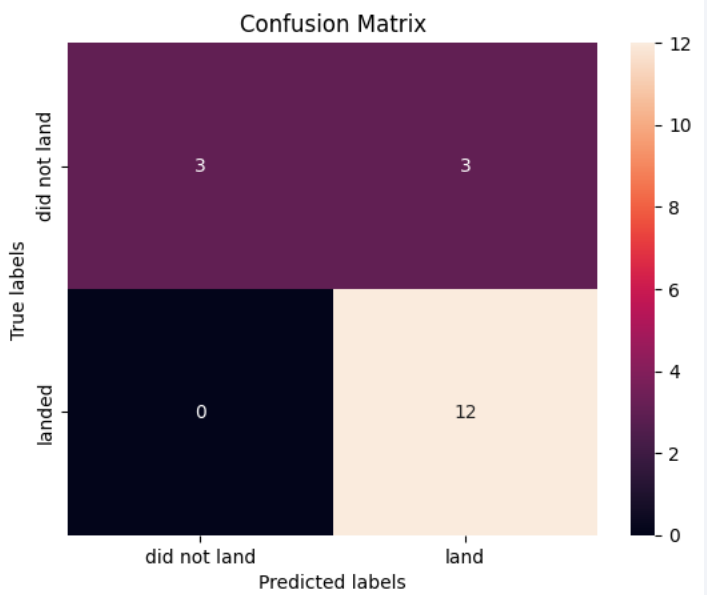
# Classification Accuracy

- All models have roughly the same accuracy with ~83%
- No best model can be chosen since all perform equally well



# Confusion Matrix

- No best performing model since all have the same accuracy of ~83%
- Confusion matrix of the Logistic Regression model is shown exemplarily
- Out of the 18 cases the model had to classify
  - **12** were classified as **true positive** (correct classification)
  - **3** were classified as **true negative** (correct classification)
  - **3** were classified as **false positive** (incorrect classification)
    - Model predicted successful landing, but it did not land
  - **0** were classified as **false negative** (incorrect classification)



# Conclusions

- Retrieved data and data preparation led to successful prediction of mission outcomes
- Either no correlation observed, or amount of data was not large enough to large enough to derive relyful correlation between launch sites, payload mass, or target orbits to the mission outcome
- Created visualizations show nicely that most important for mission outcome is the experience in performing such missions
    - Number of yellow dots increases significantly with increasing flight numbers
- 4 different machine learning models have been trained to predict mission outcomes
- All 4 models perform equally well with accuracy values of ~83%
- Increase in prediction accuracy can be achieved by either more complex models or larger amount of data

- Will be extremely hard to compete with SpaceX, since one would need to overcome their advantage in experience since they perform such missions for more than one decade
    - Experience is key to increase success rate of the performed missions!

# Thank you!