IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Denis Keil
November 2022

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data collection via different proceedings (API-Interface, Web Scraping, API Scraping)
  - ➢ Data wrangling
  - ➢ Exploratory data analyzing with SQL and visualization
  - ➢ Interactive visual analytics with Folium and Dashboard
  - ➢ Predictive Analysis for different classification model by using machine learning
- Summary of all results
  - ➢ Exploratory data analysis for the results of the different classification models
  - ➢ Interactive analytics demo in screenshoots
  - ➢ Predictive analysis results

Introduction

- Project background and context

  Spaxe X advertises its Falcon 9 rocket launches at the low cost of $62 million, while other vendors charge launch costs in excess of $165 million. The immense difference in cost is due to the reuse of the first stage. Therefore, there is a great interest for alternative companies to Space X to be able to predict whether or not after a rocket launch the first stage can land again and be reused. Thus, the declared goal of this project is to create a pipeline for machine learning with different classifiers in order to be able to predict as best as possible whether there will be a successful landing of the first stage.

- Problems you want to find answers

  Are there certain factors influencing the success rate for a successful first stage landing. Is there a correlation between various factors for a successful land maneuver?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data were collected bei using SpaceX-API und web scraping from Wikipedia page
- Perform data wrangling
  - Data were cleaned from irrelevant informations and creating one-hot-encoding datafields for machine learning
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Data were visualized bei different plotkinds for searching for pattern
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data were collected by different proceedings
    - Requesting the SpaceX-API
        - Decoding the response content, coming in json-format, and turn into a Pandas dataframe
        - Cleaning dataframe from uninteresting columns and multiple rows
        - Filter according to desired parameters
        - Checking the dataset for missing data and filling missing data up where required
    - Web-Scraping an Wikipedia page for Falcon 9 launch records by using BeautifulSoup
        - Transforming the received html-table into a Pandas dataframe

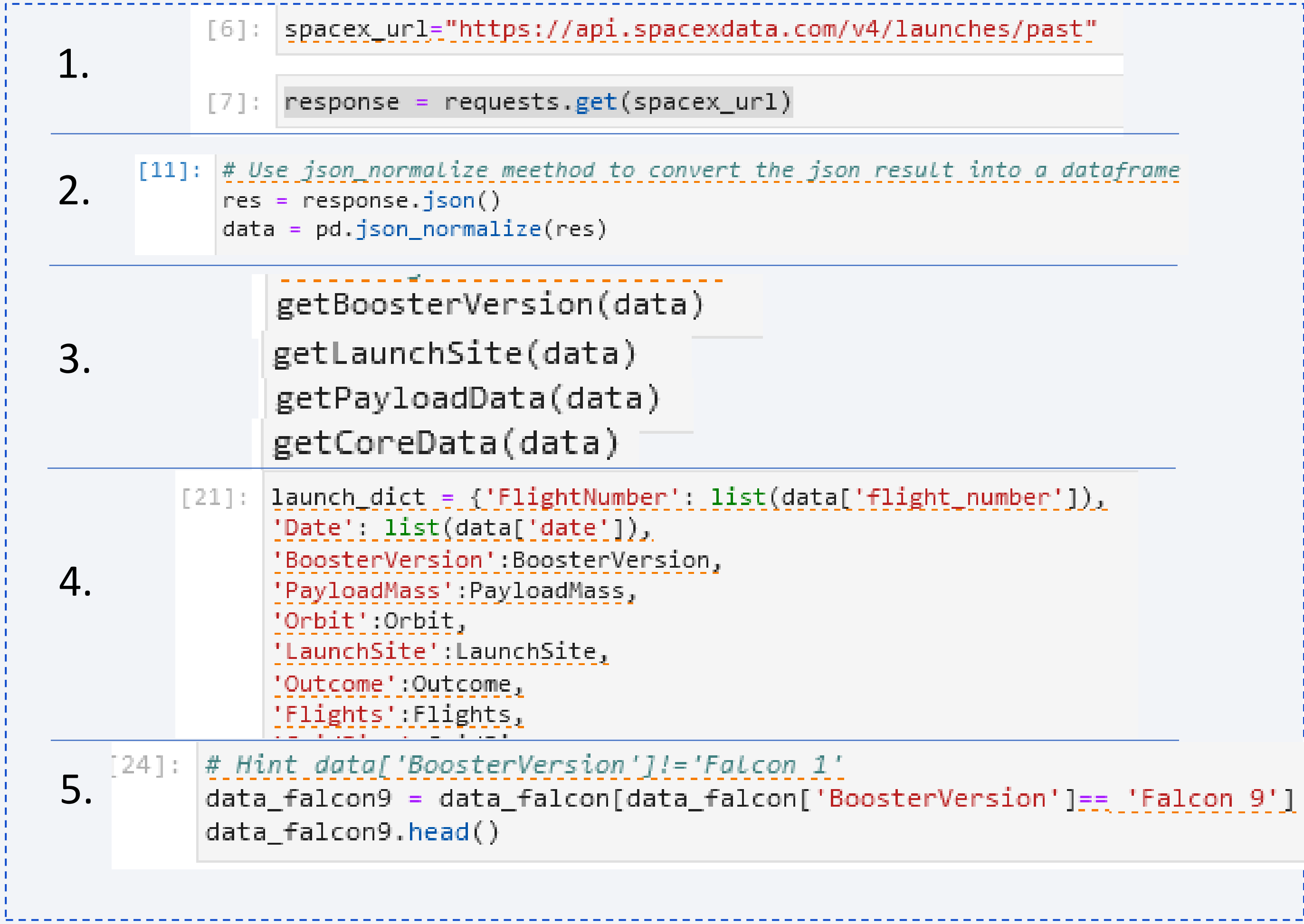
Data Collection – SpaceX API

The following workflow was used

1. Getting response from SpaceX-API
2. Converting response to json-file and transferring into a dataframe
3. Clean up the data using prepared functions
4. Creating a final dataset with the columns of interest
5. Filtering dataset for selected BoosterVersion

GitHub-link to notebook:

https://github.com/dekeil/Coursera_Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api_01.ipynb



1.

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

[7]: response = requests.get(spacex_url)
```

2.

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

3.

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4.

```
[21]: launch_dict = {'FlightNumber': list(data['flight_number']),
      'Date': list(data['date']),
      'BoosterVersion':BoosterVersion,
      'PayloadMass':PayloadMass,
      'Orbit':Orbit,
      'LaunchSite':LaunchSite,
      'Outcome':Outcome,
      'Flights':Flights,
```

5.

```
[24]: # Hint data['BoosterVersion']!='Falcon 1'
      data_falcon9 = data_falcon[data_falcon['BoosterVersion']== 'Falcon 9']
      data_falcon9.head()
```

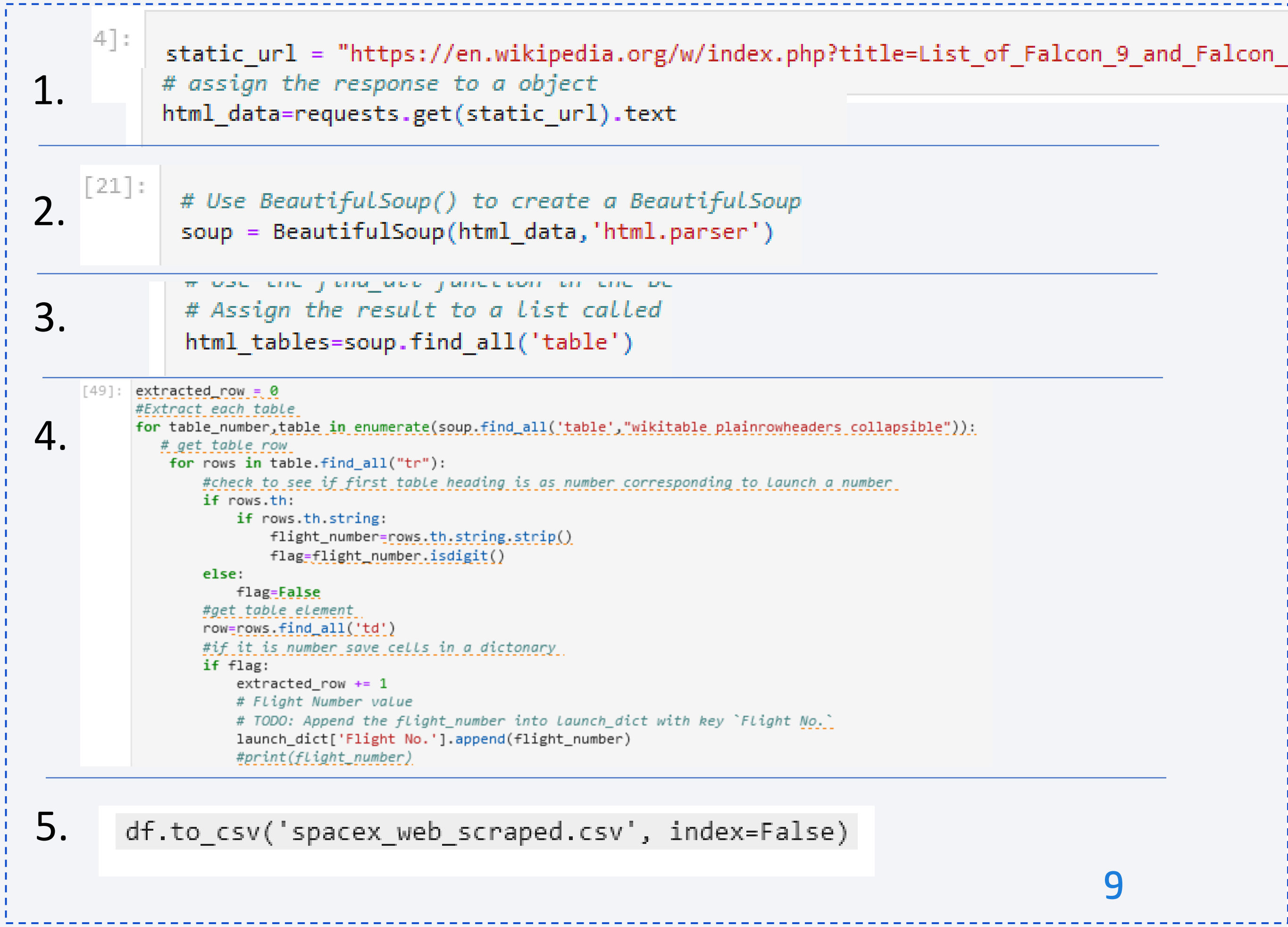
Data Collection - Scraping

The following workflow was used

1. Requesting Falcon9 Launchdata from Wiki-page
2. Creating Beautiful-soup object from HTML response
3. Extract all column/variable names from the HTML table header
4. Create a data frame by parsing the launch HTML tables
5. Exporting data to CSV-file

GitHub-link to notebook:

https://github.com/dekeil/Coursera_Capstone-Project/blob/main/jupyter-labs-webscraping_02.ipynb



1.

```
4]:
 static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_
# assign the response to a object
html_data=requests.get(static_url).text
```

2.

```
[21]:
 # Use BeautifulSoup() to create a BeautifulSoup
 soup = BeautifulSoup(html_data,'html.parser')
```

3.

```
# Use the find_all function in the BS
# Assign the result to a list called
html_tables=soup.find_all('table')
```

4.

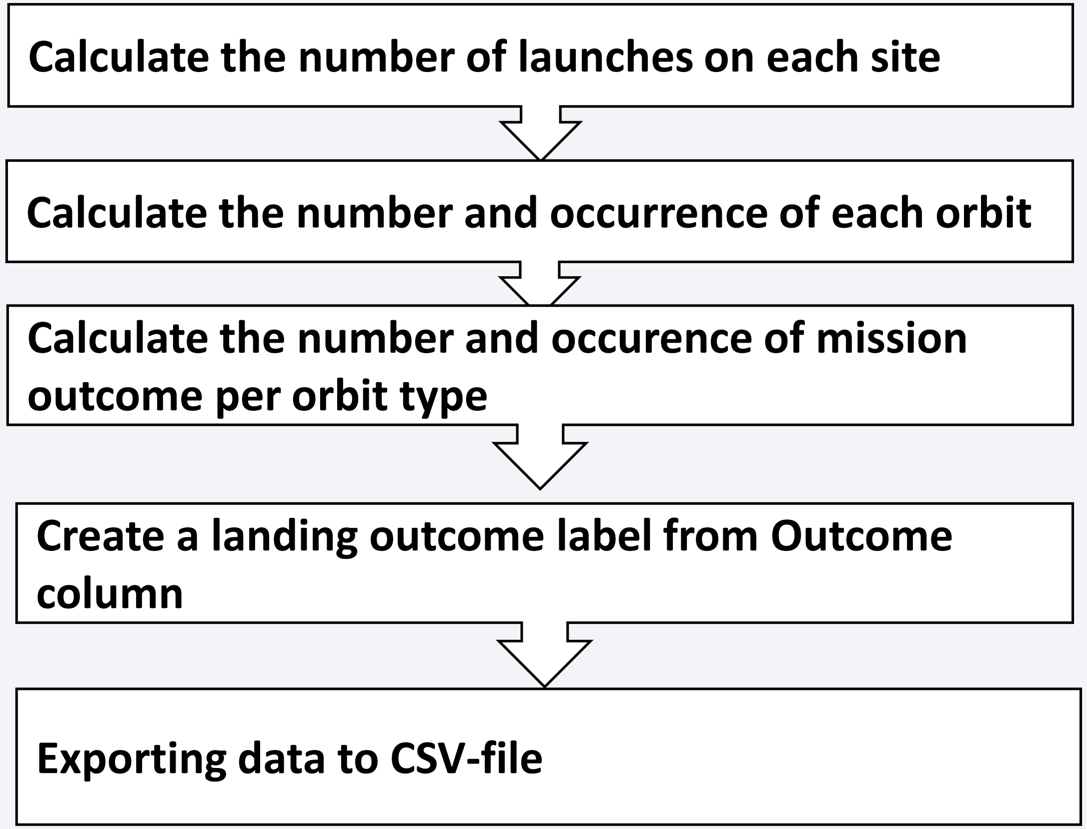
```
[49]: extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
```

5.

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Exploratory Data Analysis and Determine Training Labels



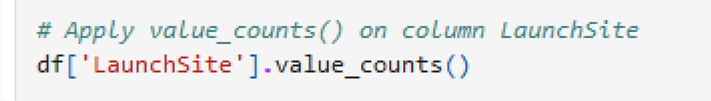
Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

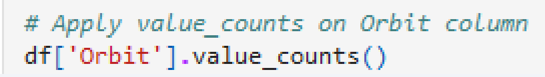
Calculate the number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

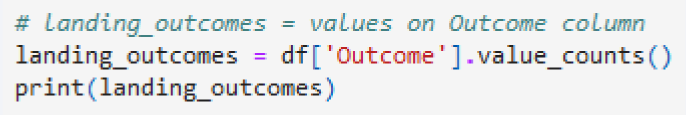
Exporting data to CSV-file



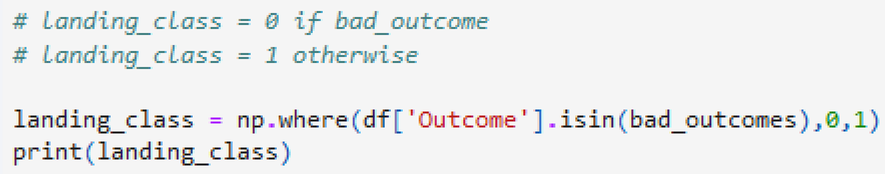
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```



```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

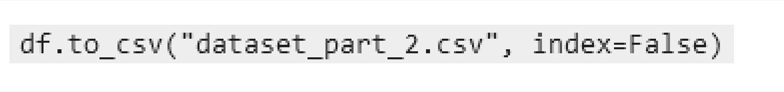


```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
print(landing_outcomes)
```



```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class = np.where(df['Outcome'].isin(bad_outcomes),0,1)
print(landing_class)
```



```
df.to_csv("dataset_part_2.csv", index=False)
```

GitHUB URL: [labs-jupyter-spacex-Data%20wrangling.ipynb]

EDA with Data Visualization

Different chart types and dependencies were plotted

- Payload-Mass vs. Flight Number / Launchsite vs. Flight Number / Launchsite vs. Payload-Mass
- orbit vs. success rate / Flight number vs. orbit / Payload Mass vs. orbit
- Launch success yearly trend

The aim is to find patterns and dependencies in the plots, which can then be used to train the machine learning engine. The different chart types have different advantages for this.

- scatterplots are useful to show relationships between variables
- barcharts are suitable for comparing the ratio of a variable in discrete classes with one another, if necessary grouping them as well
- lineplots show the progression of variables over time

GitHUB URL: jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived
- 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

GitHUB URL: jupyter-labs-eda-sql-coursera_sqllite_01.ipynb

Build an Interactive Map with Folium

objects added to the interactive map

- all launch sites are marked on a map
  - NASA Johnson Space Center marked as a point using latitude and longitude coordinates and set as the center of the map, additionally identified with a text label
  - all X-space launch sites marked as a point, provided with a text label and plotted using the latitude and longitude coordinates. Thus marking the geographic location and distance to the equator
- Drawing and color coding of the launch outcome for each start per launch site
  - adding a colored marker to the exact starting point for each launch site, green for successful launches, red for failed, this allows a quick overview of the success rate per launch site
- Calculate the distances between a launch site to its proximities
  - Insertion of connecting lines from a selected launch site (CCAFS SLC-40) to the coast and nearby infrastructure such as the nearest railway line, highway or city. Determine an entry of the respective distances

GitHUB URL: lab_jupyter_launch_site_location.ipynb

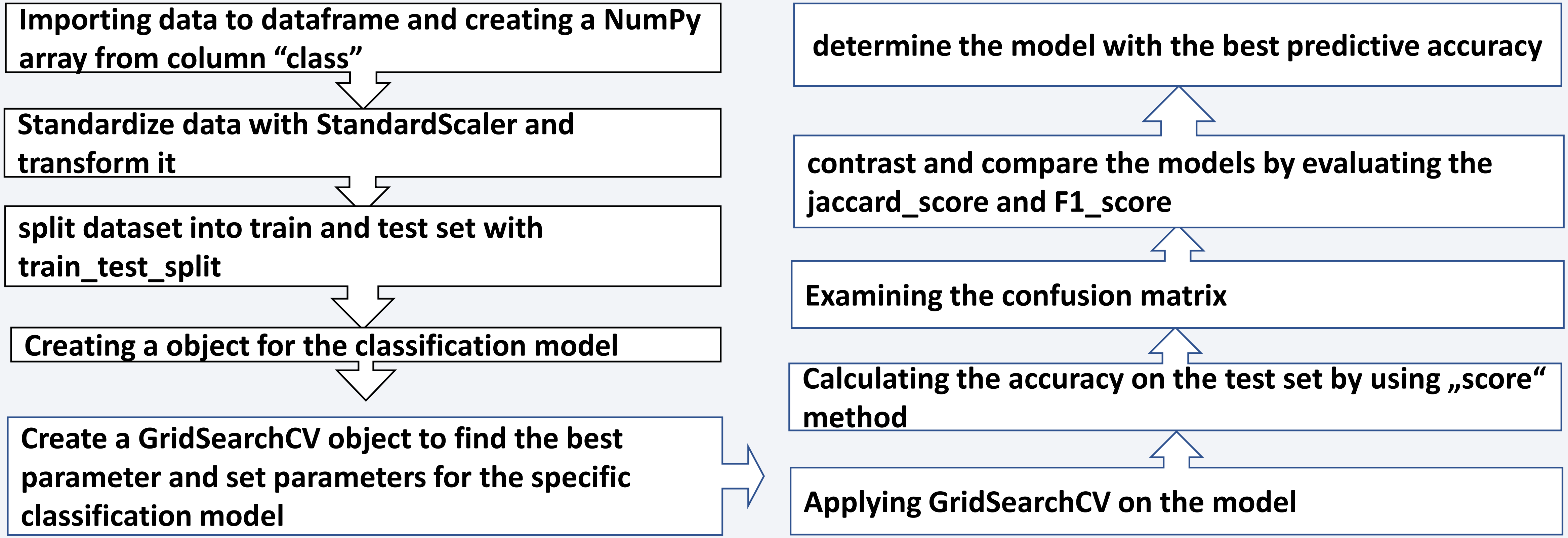
Build a Dashboard with Plotly Dash

- Dropdown list of all launch sites
  - Adding a dropdown list to select the possible launch sites
- Creation of a pie chart with successful and failed launches (all sites combined / specific site)
  - Added a pie chart to show for all sites aggregated or a selected site - depending on selection - the total number of successful launches vs. failed launches for the sites
- Include range slider for payload
  - Add a Range Slider to Select Payload
- Scatterplot showing payload mass vs. success rate, selectable for the different booster versions
  - add a scatterplot to show the dependency between payload and launch success/success rate

GitHUB URL: [spacex_dash_app.py]()

Predictive Analysis (Classification)

A total of 4 classification models were created and their results compared with each other (logistic regression, SVM, decision tree, k nearest neighbors) - the workflow is, however, basically similar for all of them



Importing data to dataframe and creating a NumPy array from column “class”

Standardize data with StandardScaler and transform it

split dataset into train and test set with train_test_split

Creating a object for the classification model

Create a GridSearchCV object to find the best parameter and set parameters for the specific classification model

Applying GridSearchCV on the model

Calculating the accuracy on the test set by using „score“ method

Examining the confusion matrix

contrast and compare the models by evaluating the jaccard_score and F1_score

determine the model with the best predictive accuracy

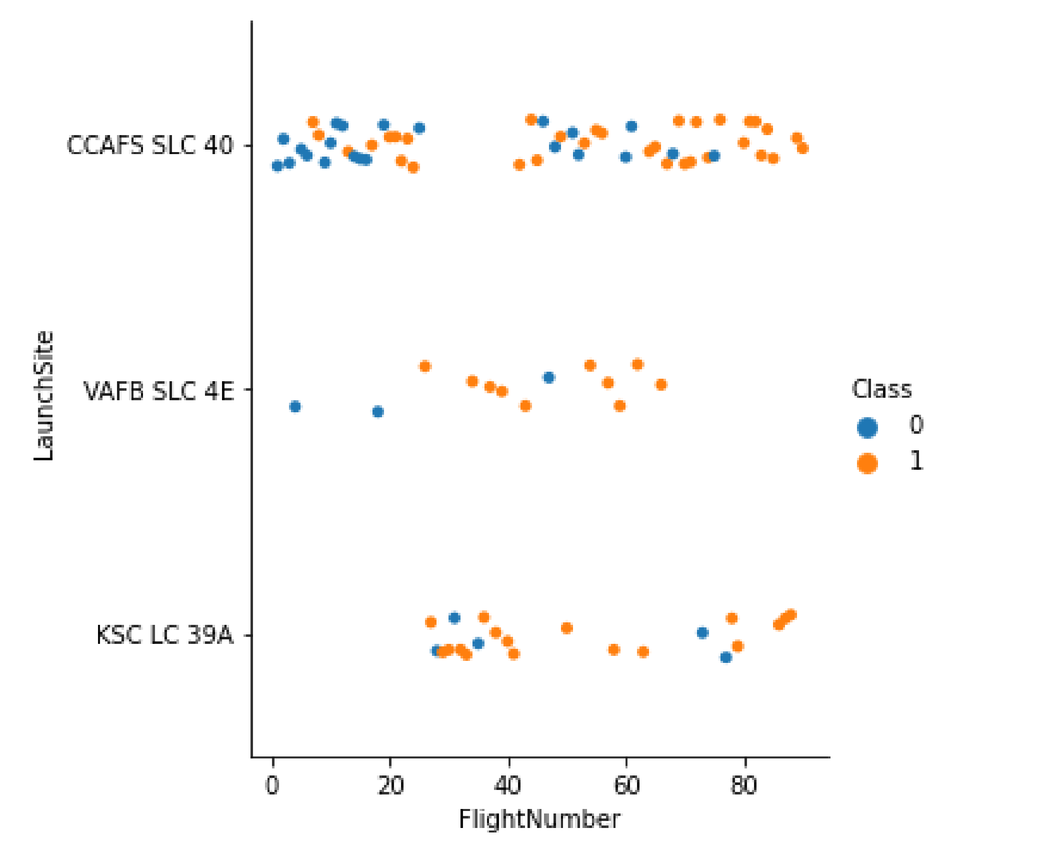
GitHUB URL: SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Section 2

Insights drawn from EDA

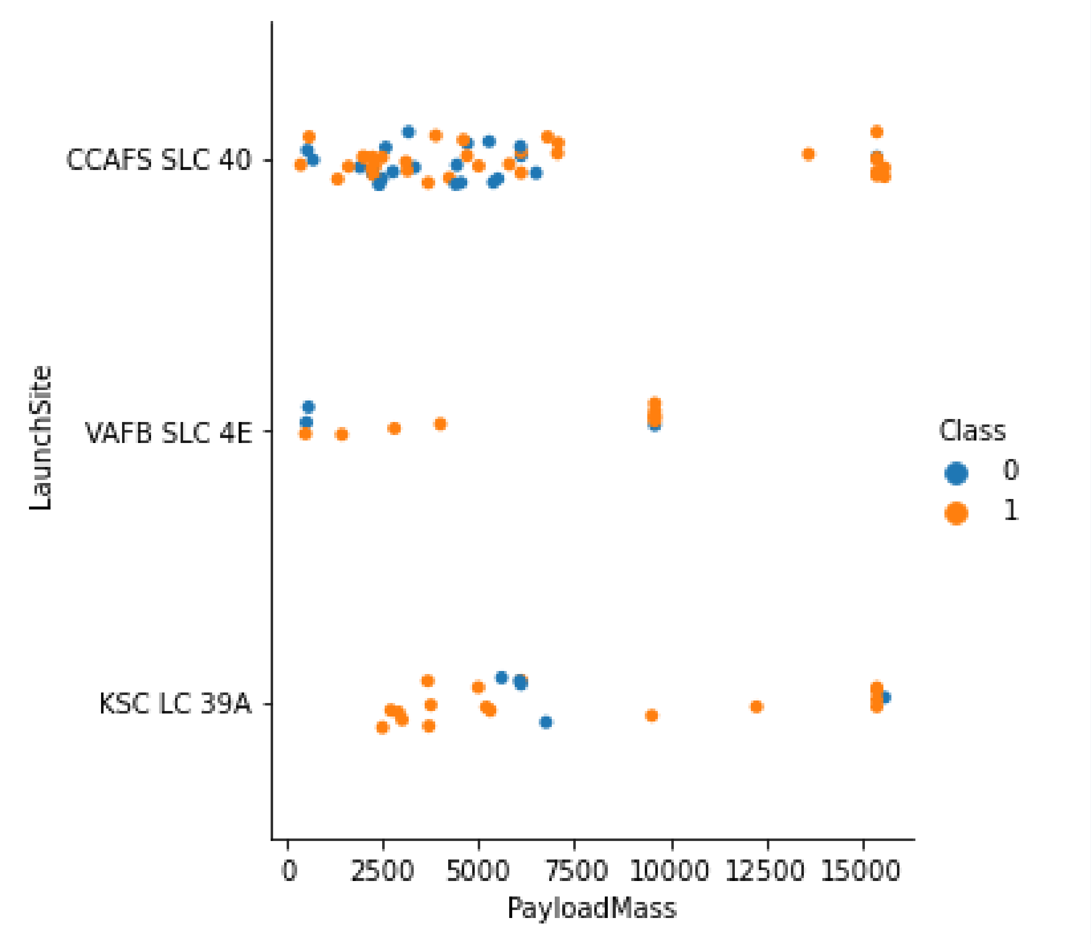
Flight Number vs. Launch Site

- the first launch attempts were made from the CCAFS SLC 40 launch site
- the probability of success increased as the starting attempts progressed
- Starting places with the best success rate are VAFB SLC 4E and KSC LC 39A



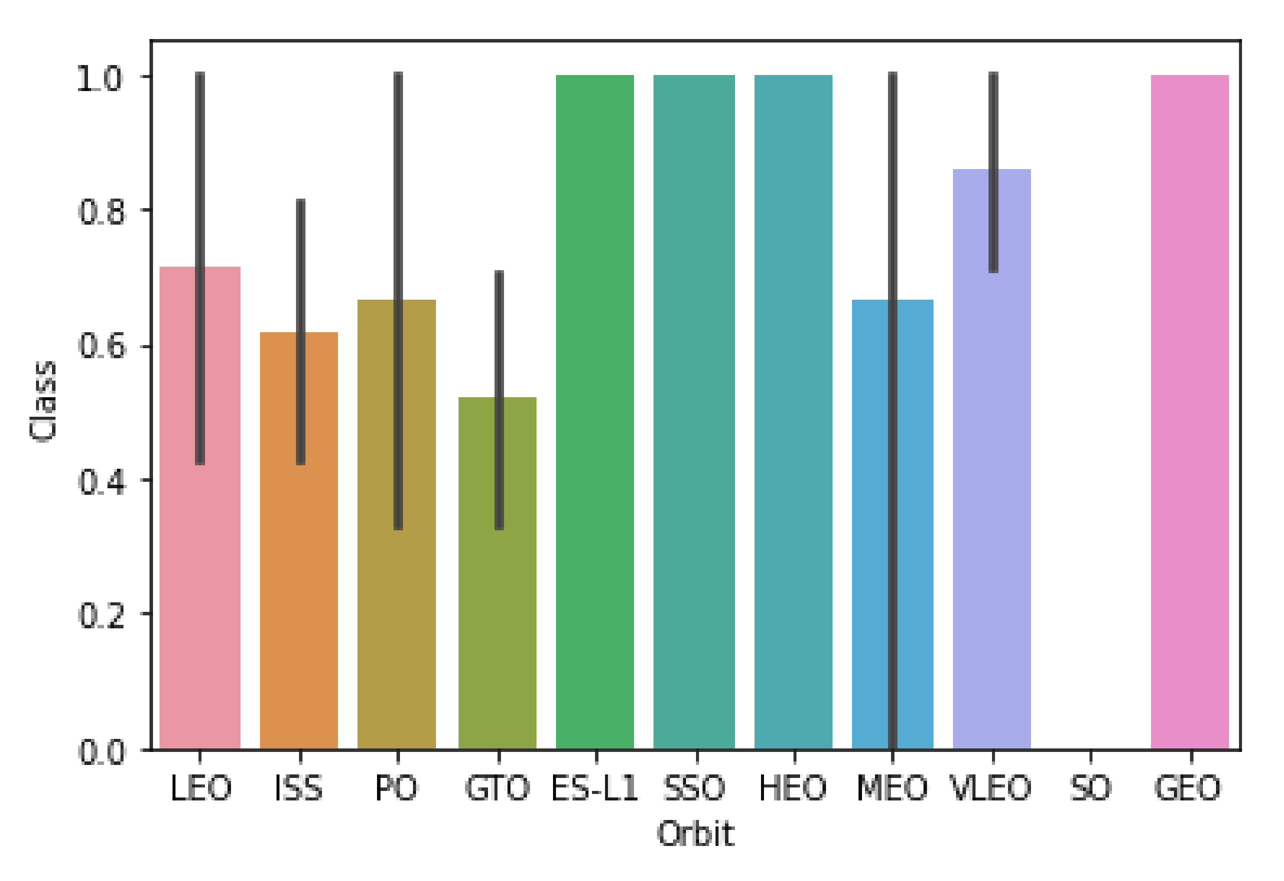
Payload vs. Launch Site

- all launch sites show a high success rate for a high payload
- Launch site VAFB SLC 4E seems unsuitable for launches with high payloads



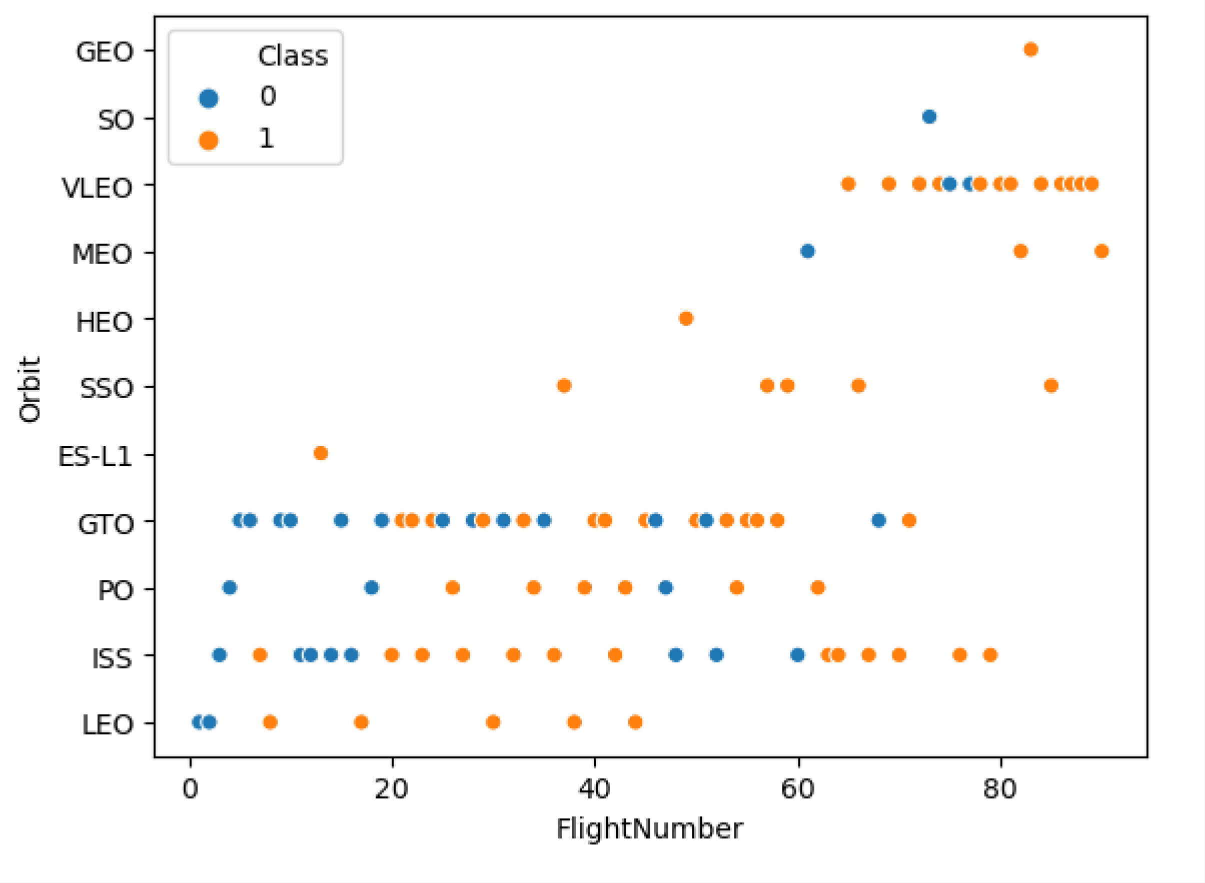
Success Rate vs. Orbit Type

- Orbits with success rate greater 80% are ES-L1, SSO, HEO, VLEO and GEO
- Orbits with success rates between 50 – 80% are LEO, ISS, PO, GTO and MEO
- Orbit without successful launches is SO



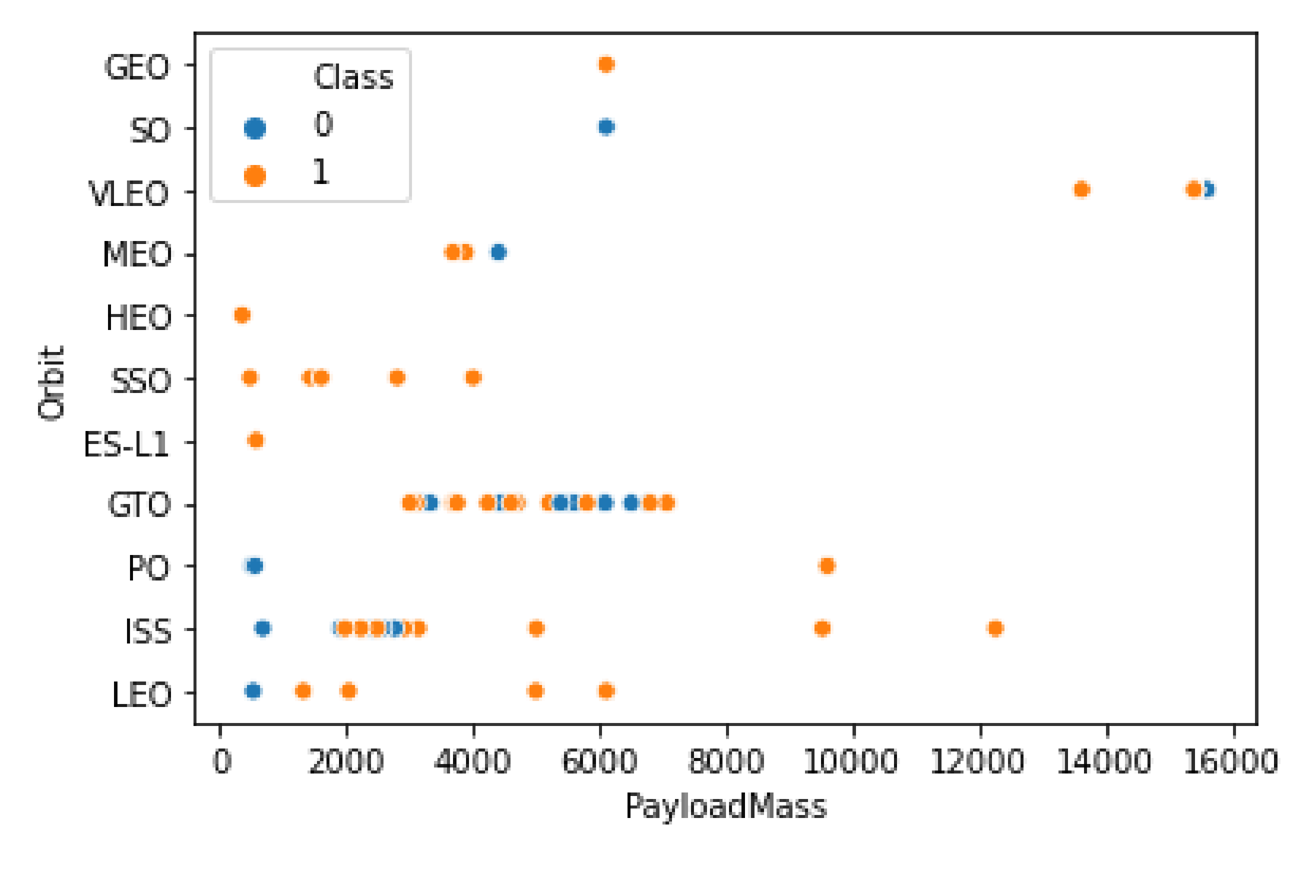
Flight Number vs. Orbit Type

- most and also first launches were in low orbit up to the GTO, here the chances of success increased with the number of launches
- most of the tests in orbit VLEO were successful



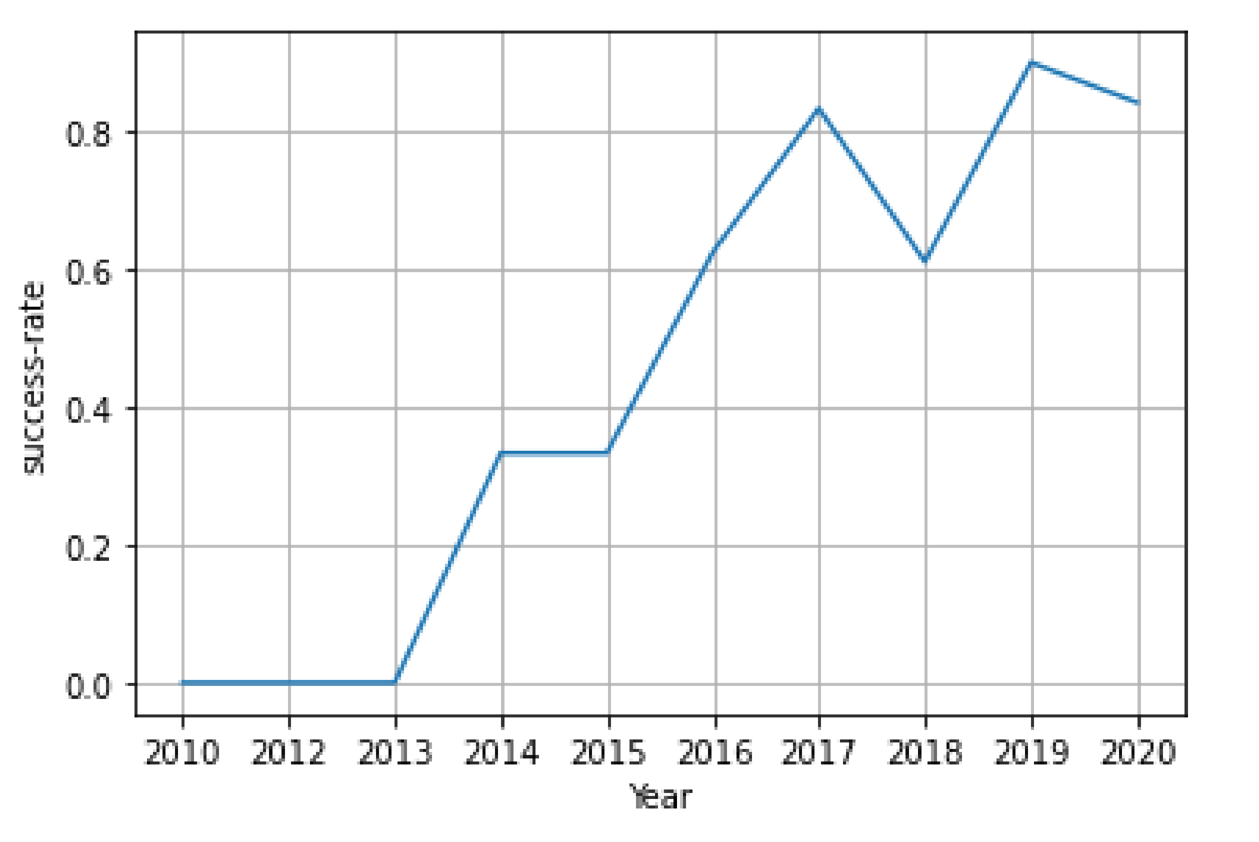
Payload vs. Orbit Type

- apparently there is no general relationship between payload and orbit
- low orbits show a low success rate with a low payload, but this could also have been the first launch attempts and caused by this
- The orbits GTO and ISS had the most launches, while ISS also had the widest spread of payloads

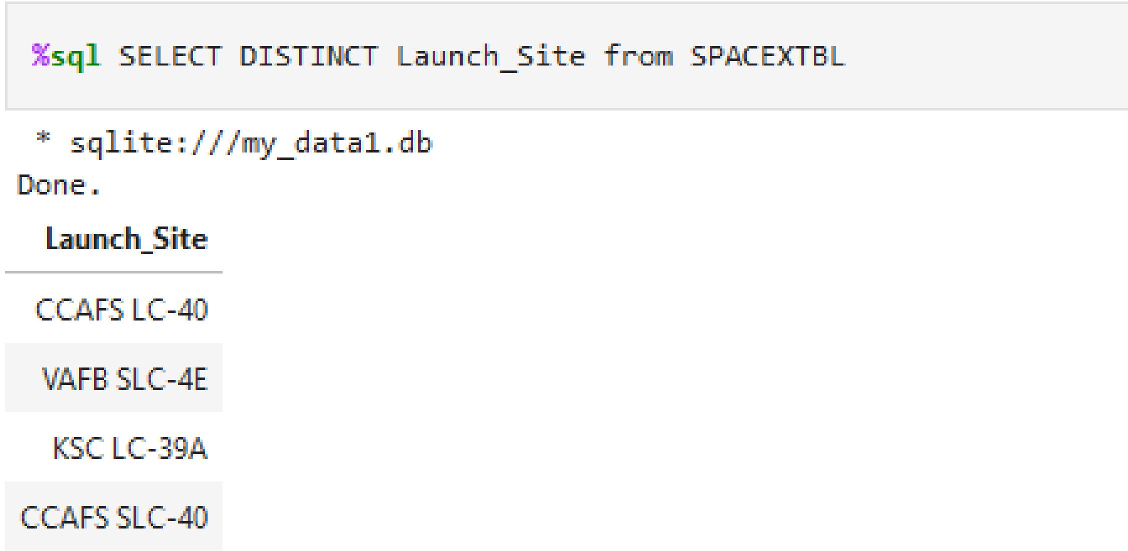


Launch Success Yearly Trend

- in the years 2010 - 2013 there were no successful launch attempts
- in the following years there was a sharp increase in the probability of success
- the year with the highest probability of success (> 90%) was 2019



All Launch Site Names



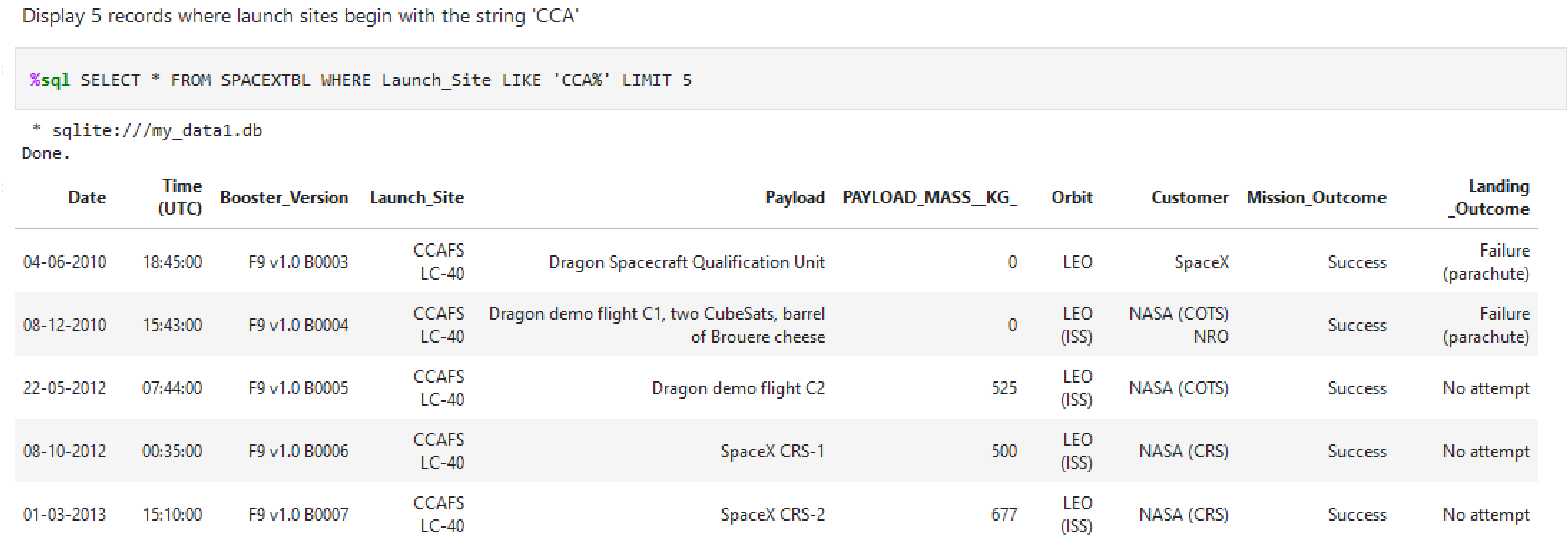
```
%sql SELECT DISTINCT Launch_Site from SPACEXTBL
```

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

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

- Query with the keyword "DISTINCT" returns all launch sites once

Launch Site Names Begin with 'CCA'



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

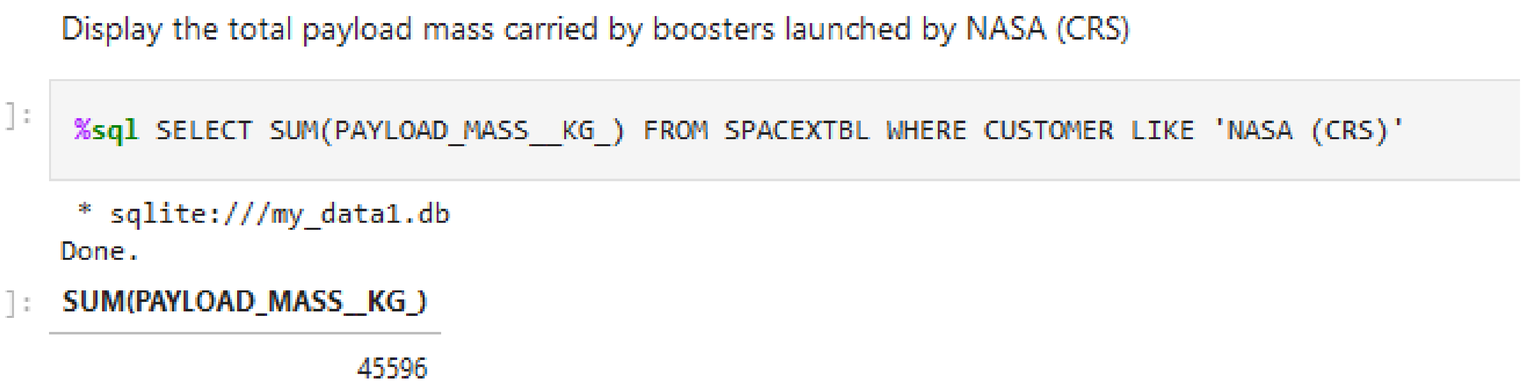
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

```
 * 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

- The query result shows a selection of 5 entries in the "Launch_site" column which start with "CCA"

Total Payload Mass



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

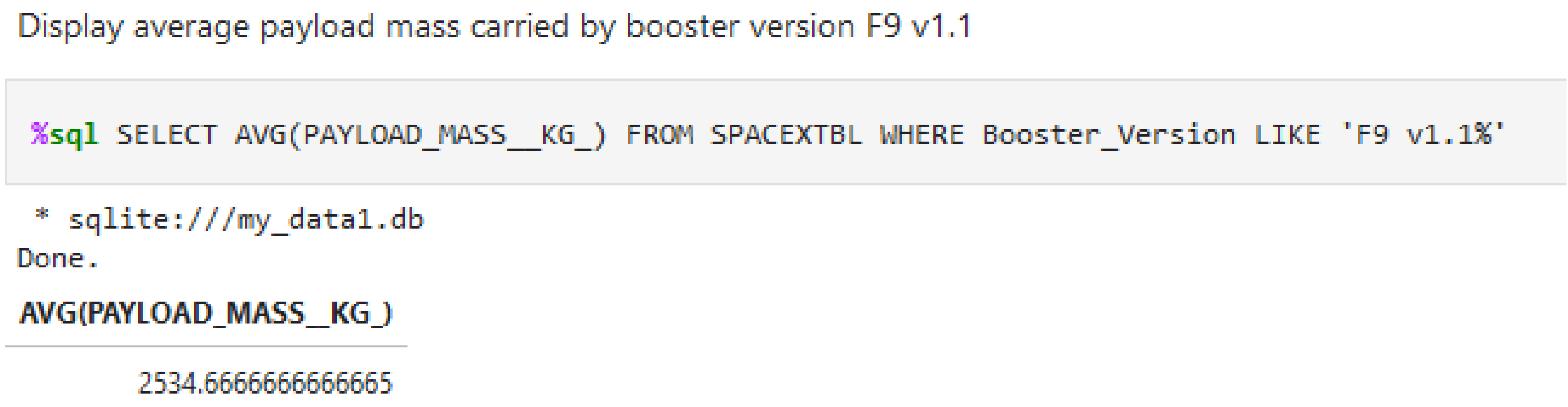
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS_KG_)
45596

- the result returns the total payload of all launches on behalf of NASA

Average Payload Mass by F9 v1.1



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

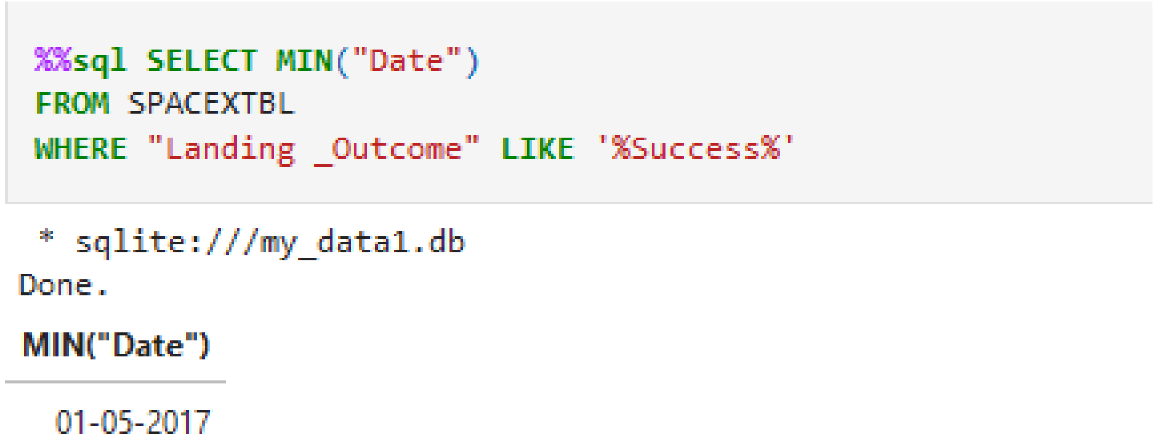
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- the result gives the average payload of the launches with the booster version "F9 v1.1"

First Successful Ground Landing Date



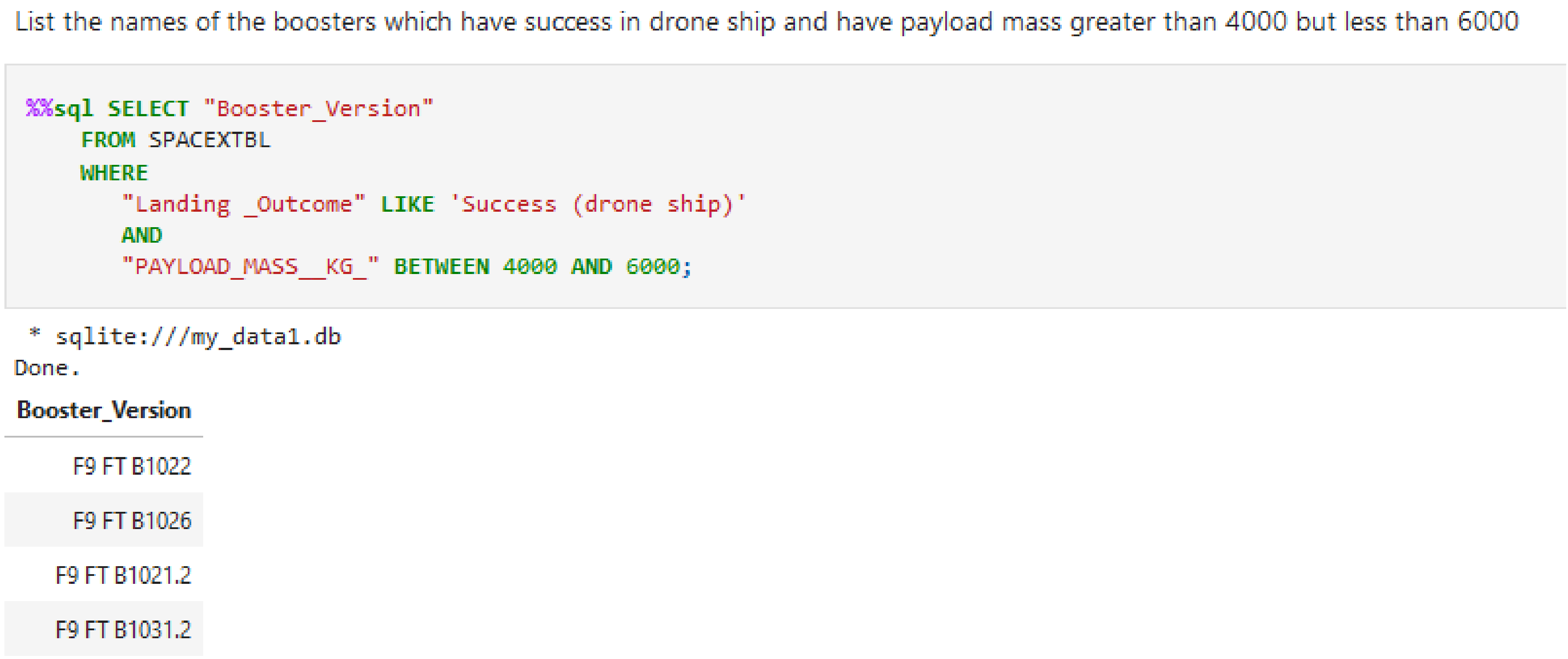
```
%%sql SELECT MIN("Date")
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success%'
```

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

MIN("Date")
01-05-2017

- Query returned the first successful landing on 01/05/2017

Successful Drone Ship Landing with Payload between 4000 and 6000



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

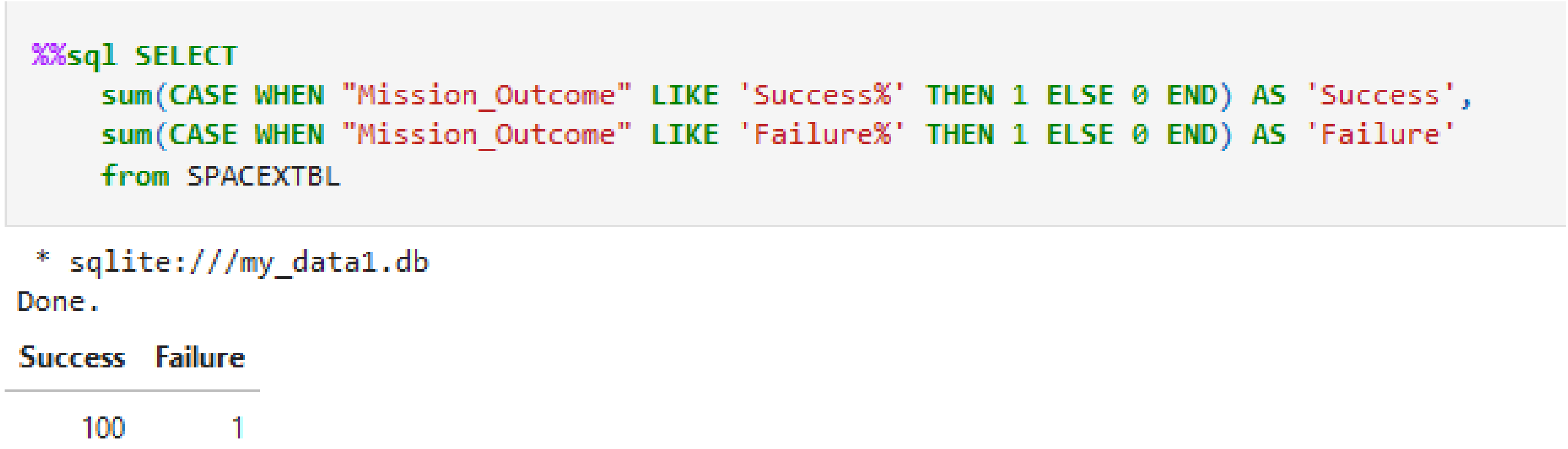
```
%%sql SELECT "Booster_Version"
    FROM SPACEXTBL
    WHERE
       "Landing _Outcome" LIKE 'Success (drone ship)'
       AND
       "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

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

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

Total Number of Successful and Failure Mission Outcomes



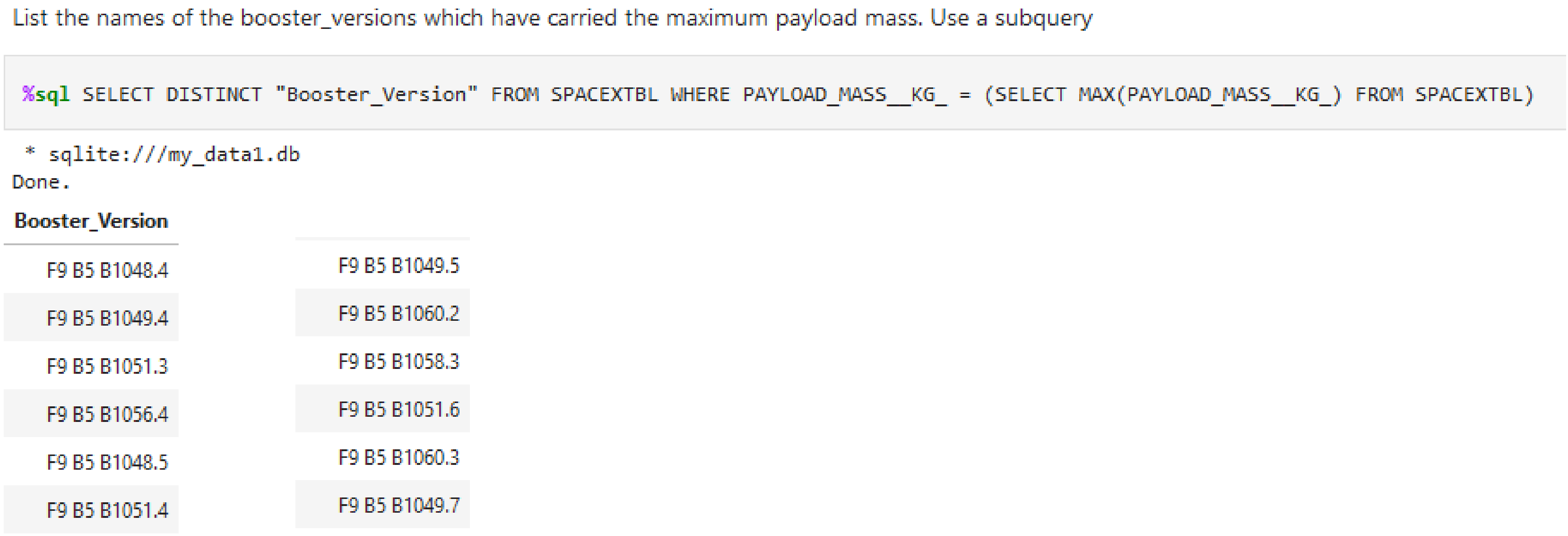
```
%%sql SELECT
    sum(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 ELSE 0 END) AS 'Success',
    sum(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 ELSE 0 END) AS 'Failure'
    from SPACEXTBL
```

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

Success	Failure
100	1

- Returning the query returns the sum of the successful and failed mission outcomes

Boosters Carried Maximum Payload



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

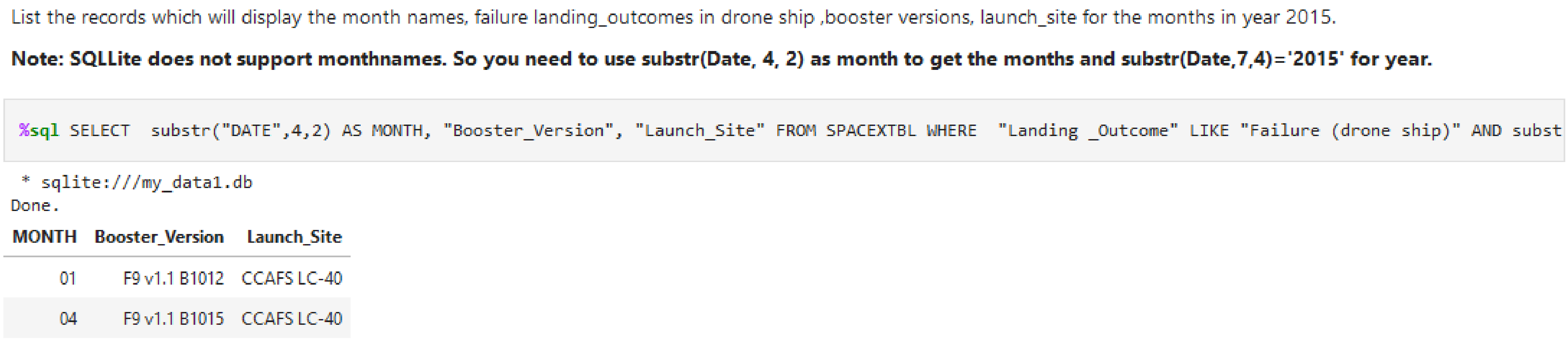
```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

- Query returns the names of the booster versions that carried the maximum payload, 12 booster versions in total

2015 Launch Records



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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

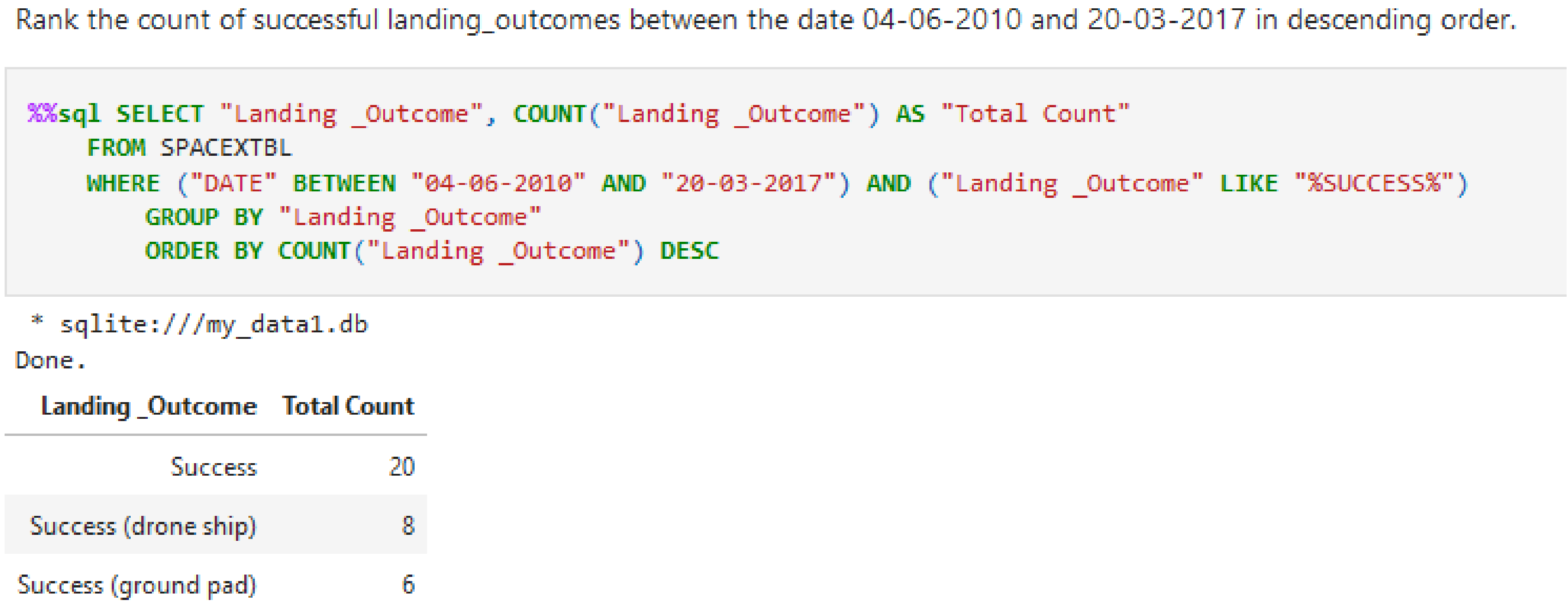
```
%sql SELECT  substr("DATE",4,2) AS MONTH, "Booster_Version", "Launch_Site" FROM SPACEXTBL WHERE  "Landing _Outcome" LIKE "Failure (drone ship)" AND subst
```

```
 * 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

- the combined query returns the names of the booster versions, the launch site and the landing outcome for all launches in 2015 that had a failed landing outcome in drone ships

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



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") AS "Total Count"
    FROM SPACEXTBL
    WHERE ("DATE" BETWEEN "04-06-2010" AND "20-03-2017") AND ("Landing _Outcome" LIKE "%SUCCESS%")
        GROUP BY "Landing _Outcome"
        ORDER BY COUNT("Landing _Outcome") DESC
```

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

Landing _Outcome	Total Count
Success	20
Success (drone ship)	8
Success (ground pad)	6

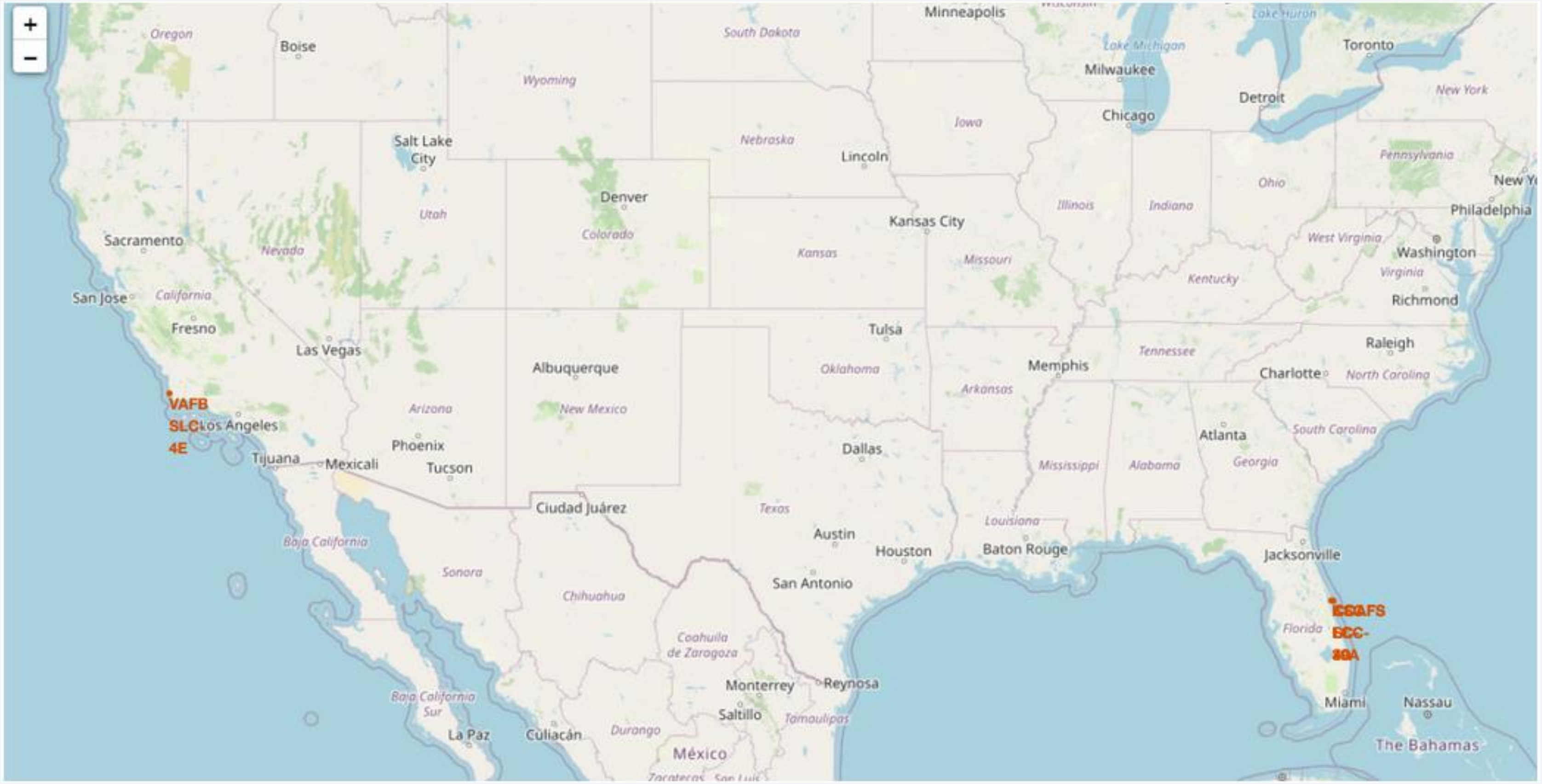
- the combined query returns the landing outcome and the number of landing outcomes for all launches between 2010-06-04 to 2017-03-20. Here, the summation is carried out over the different landing outcomes and an output in descending order

Section 3

Launch Sites Proximities Analysis

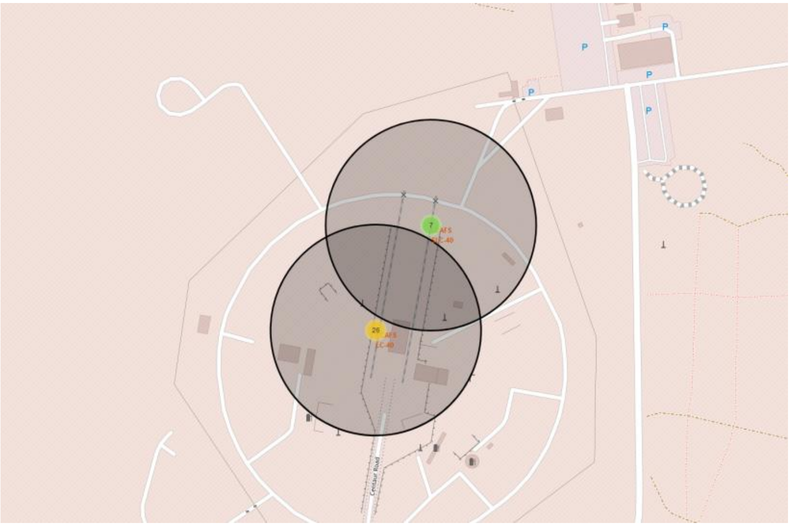
Location of all launch sites

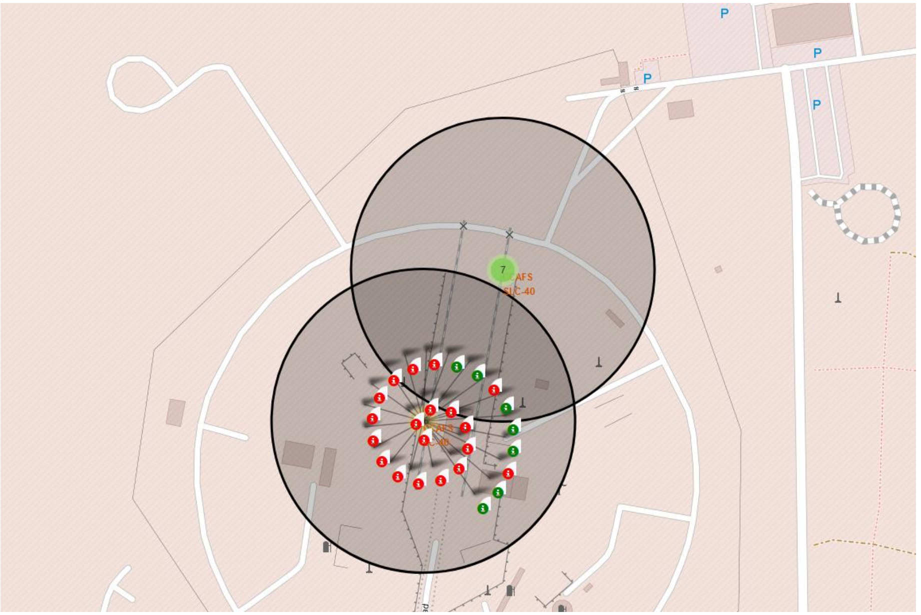
All the launch sites for the Space X missions are marked on the map.The proximity to the coast in the west and east can be clearly seen, as well as the relative proximity to the equator.



Launch results for site CCAFS

Maps give an overview of the total number of starts at the respective starting place





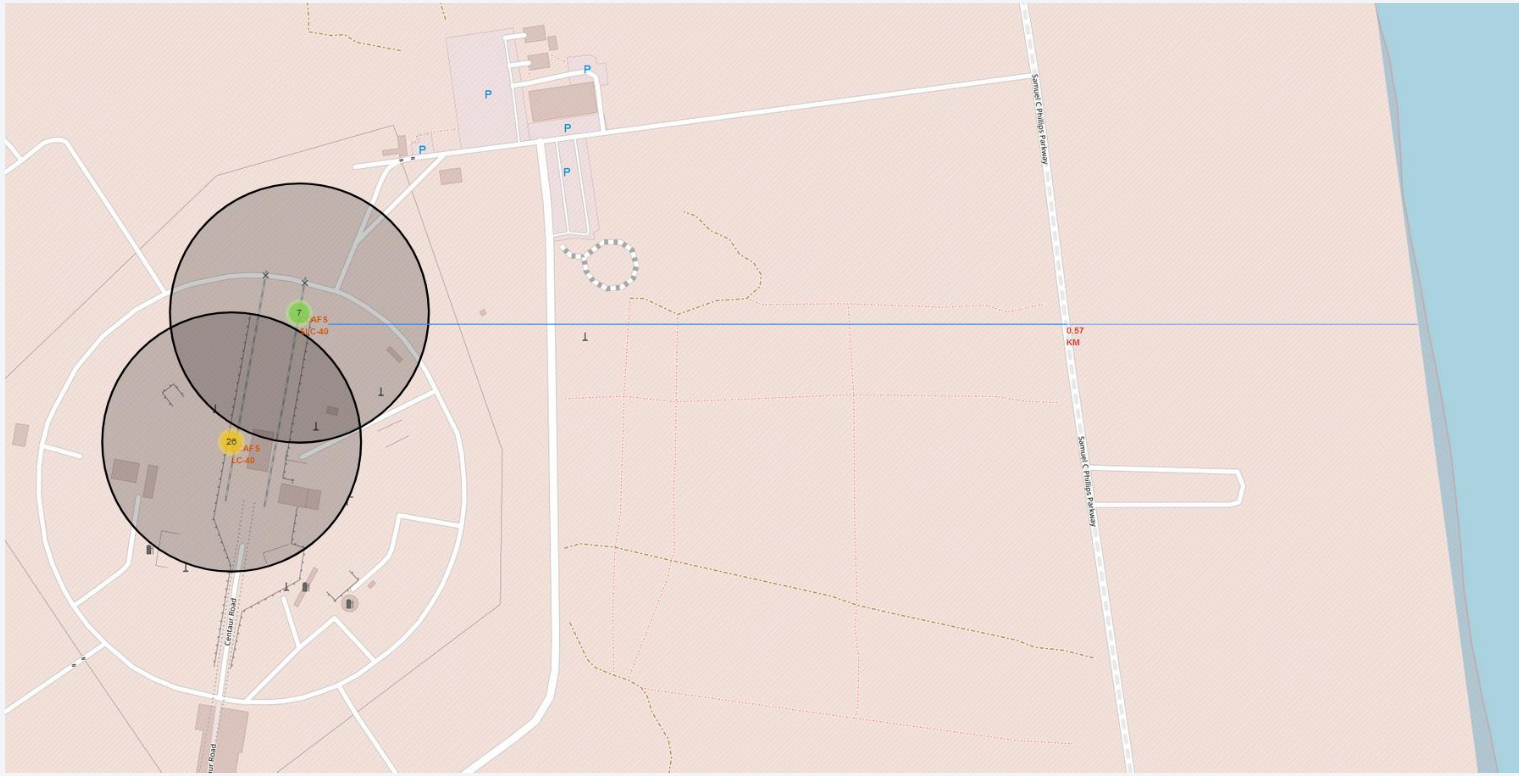
Map section provides a detailed insight into the launch attempts carried out. On the one hand the exact starting position and then the mission result - red marker means failure, green marker means success

Infrastructure around the launch site

The section shows that the launch site is in close proximity to the coast (0.57 km).

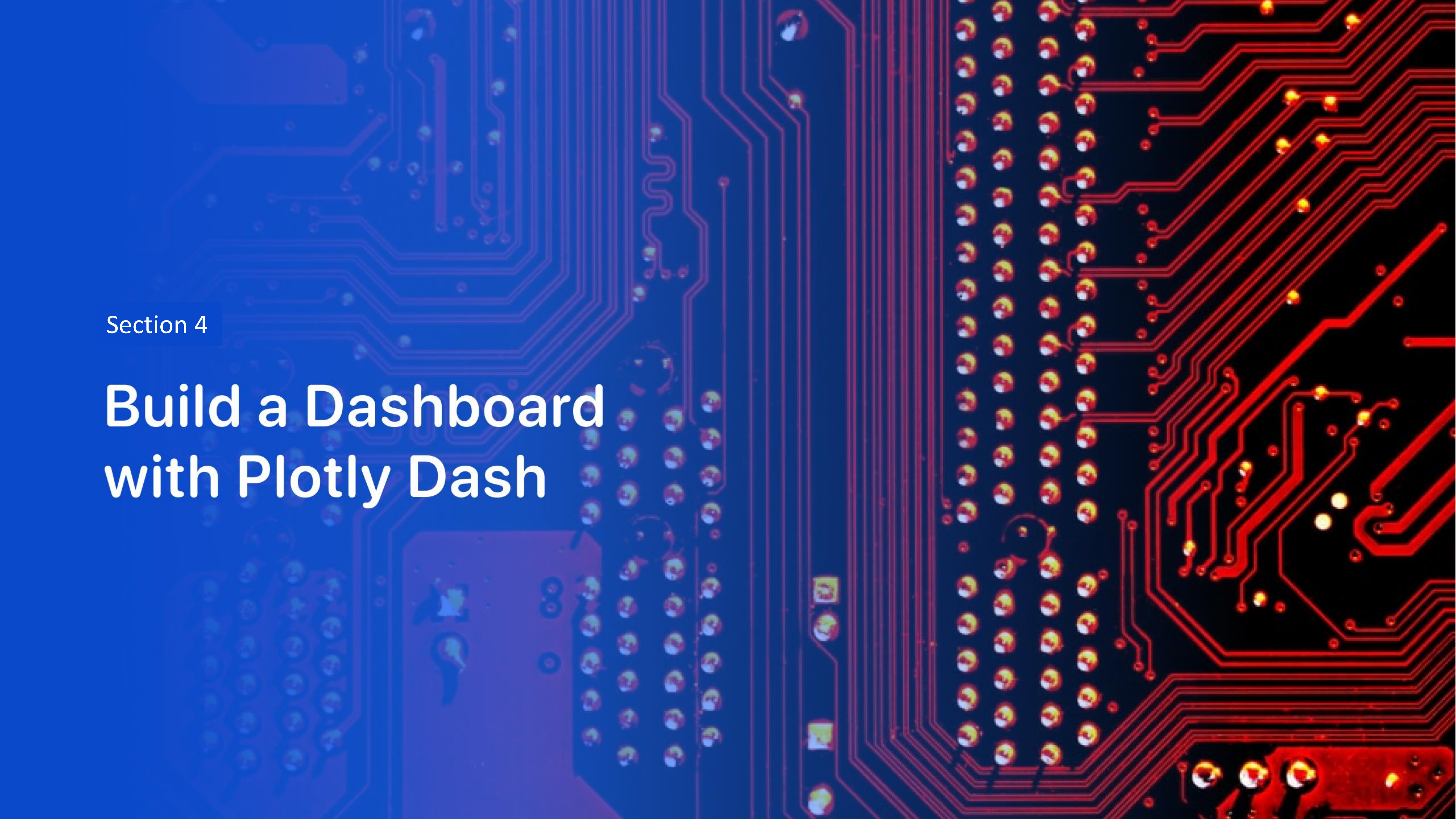
Other infrastructure such as roads and railways are also in close proximity.

Distance to inhabited areas and cities, on the other hand, is greater for safety reasons.

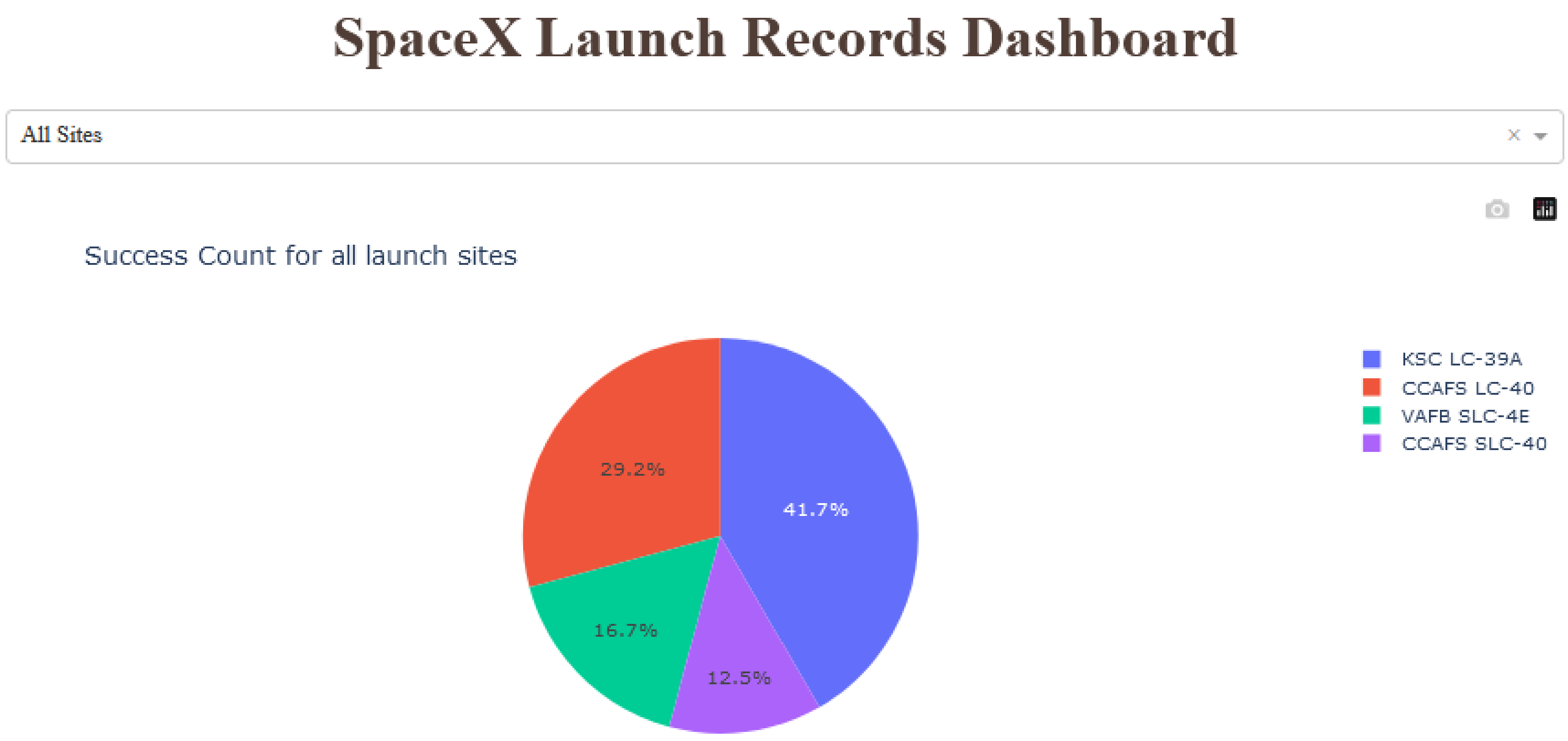


Section 4

Build a Dashboard with Plotly Dash



Start success overview for all start sites

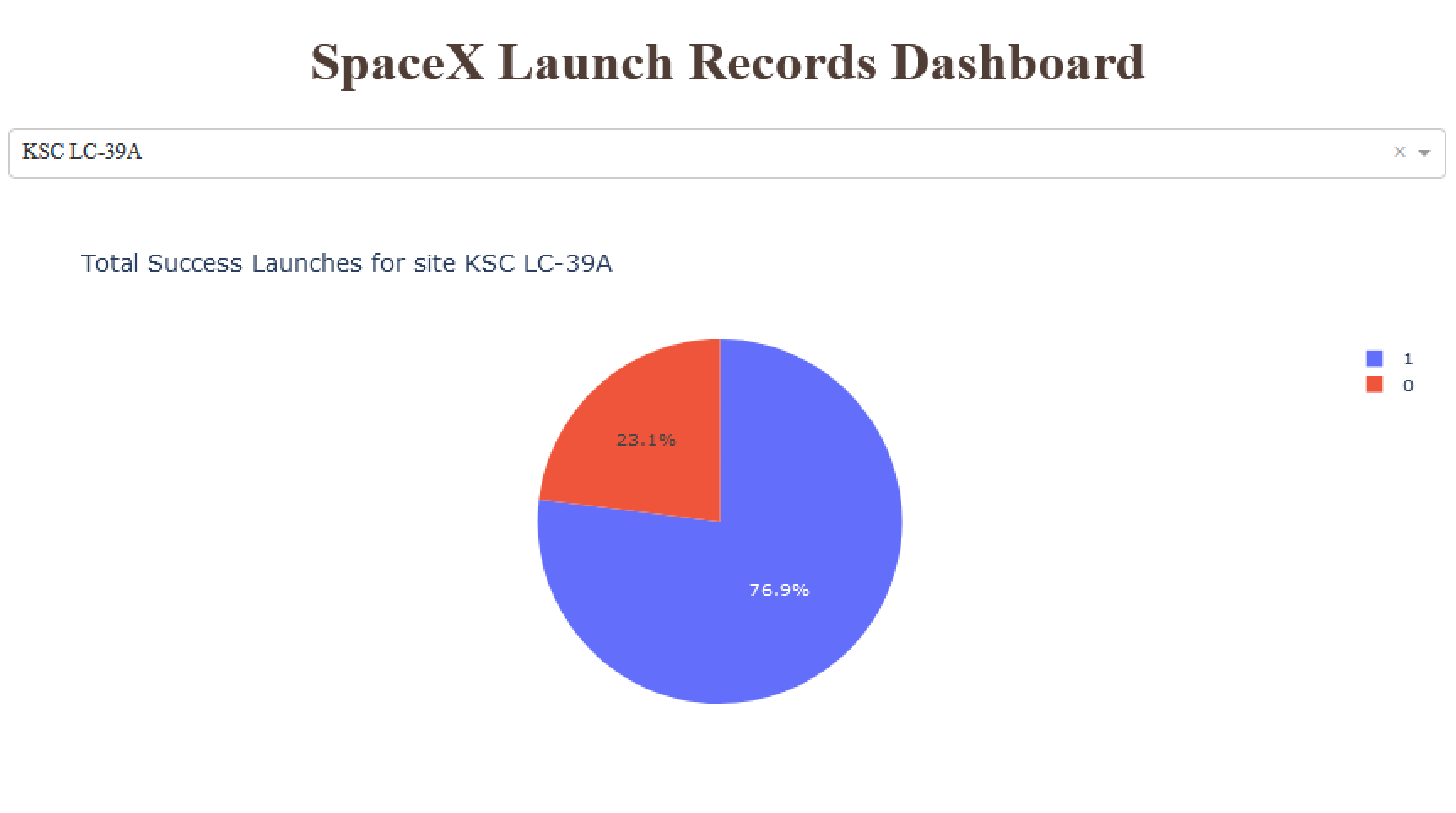


SpaceX Launch Records Dashboard

All Sites

The graphic shows that most successful launch attempts took place at the launch site "KSC LC-39A". In total, more than 2/3 of the total successful missions ran at the two launch sites "KSC LC-39A" and "CCAFS LC-40".

Launch successes at KSC LC-39A



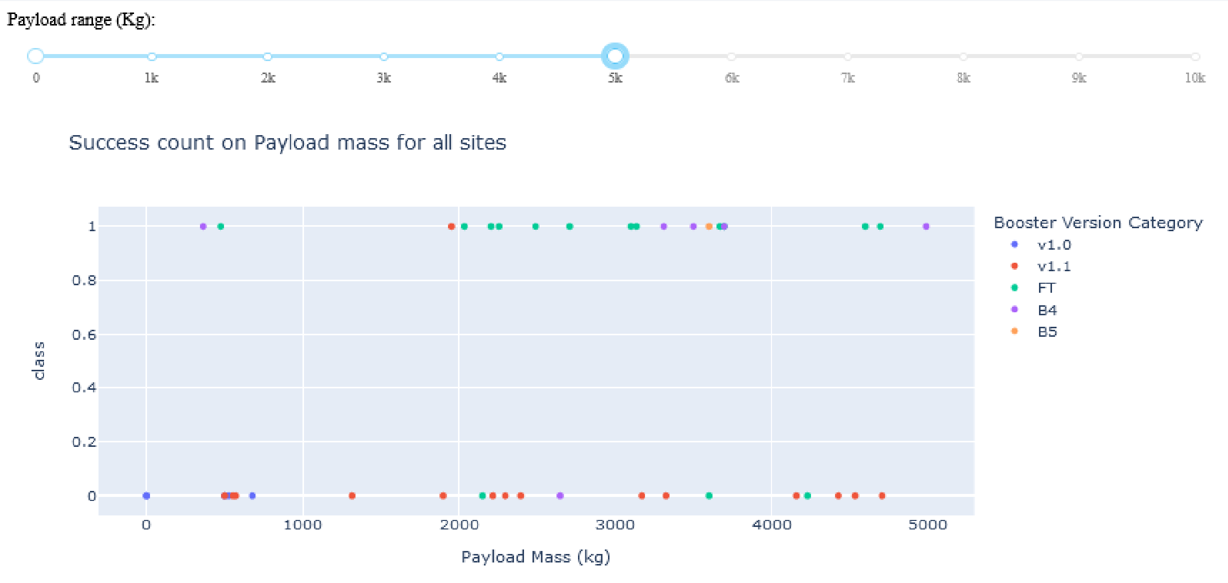
SpaceX Launch Records Dashboard

KSC LC-39A

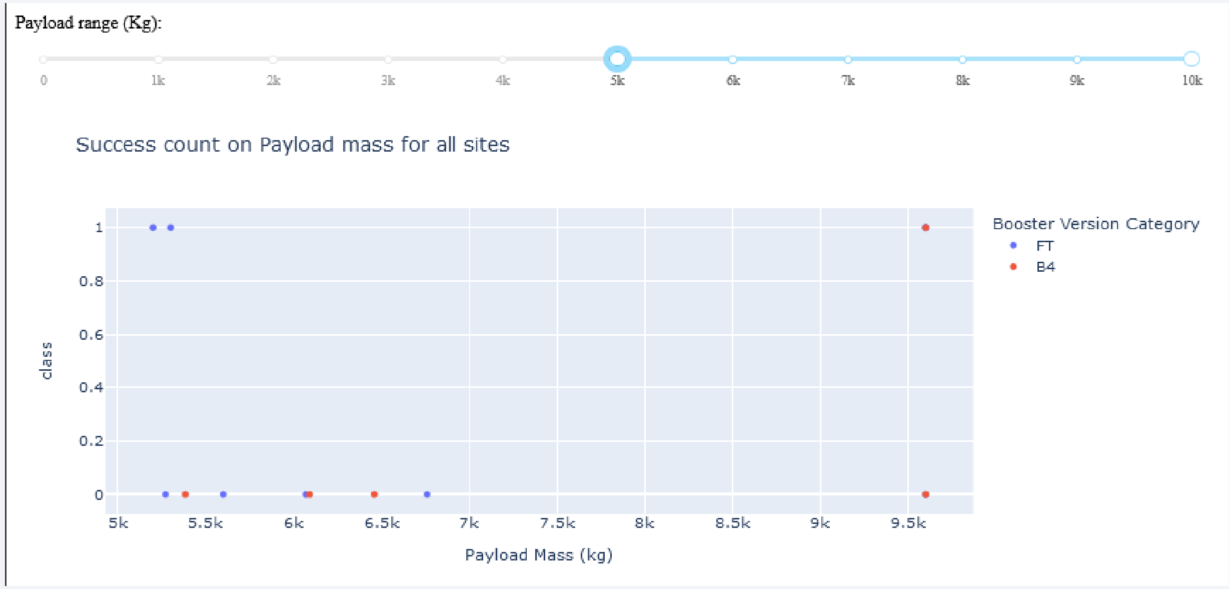
Total Success Launches for site KSC LC-39A

Of the missions performed at the "KSC LC-39A" launch site, 76.9% were successful and 23.1% failed.

Launch successes at different payload masses



Lower range of payload mass



Higher range of payload mass

The dashboard shows a variety of information:

- with the booster version v 1.0 and v1.1 there were almost only failed attempts, the payload was also still quite low
- Booster version "FT" had the highest chance of success, especially in the payload between 2000 - 5500kg
- with a payload of more than 5500kg, the probability of success decreased significantly

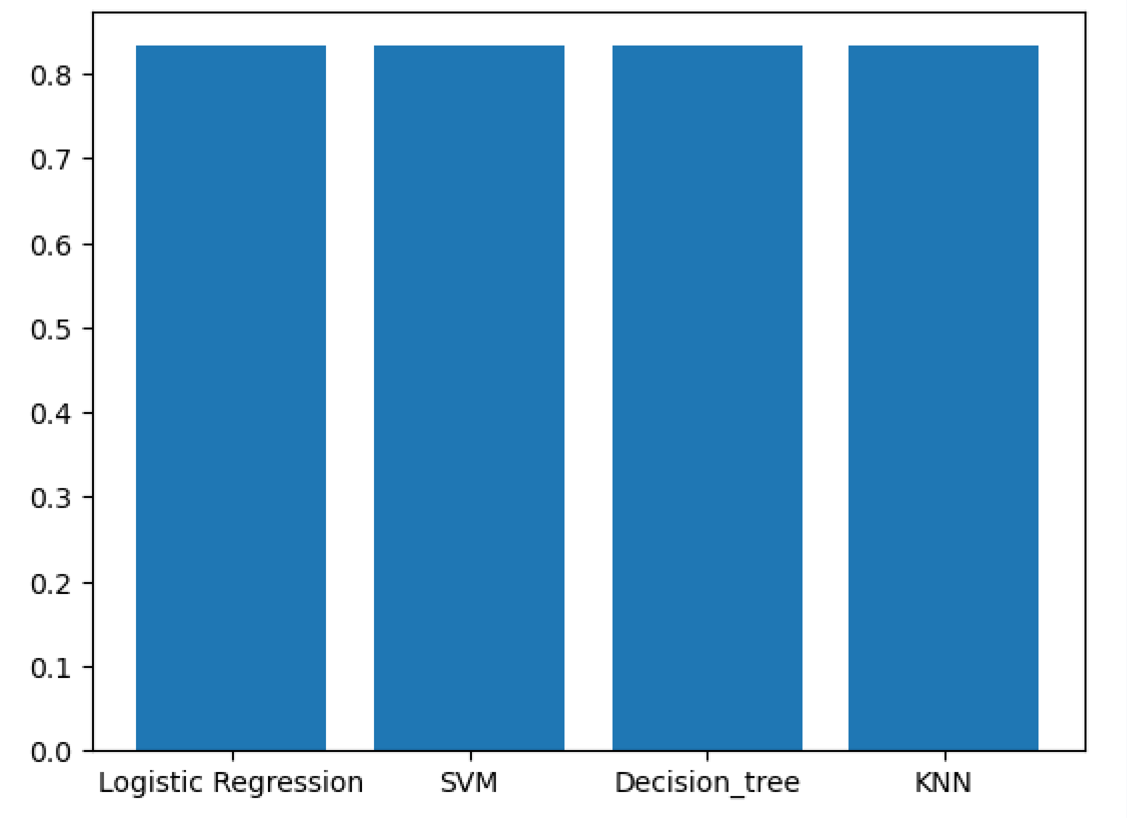
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- the accuracy for all models are the same by using the test dataset.

- When using the training data set, however, different accuracies were found. here the decicion tree performed best.

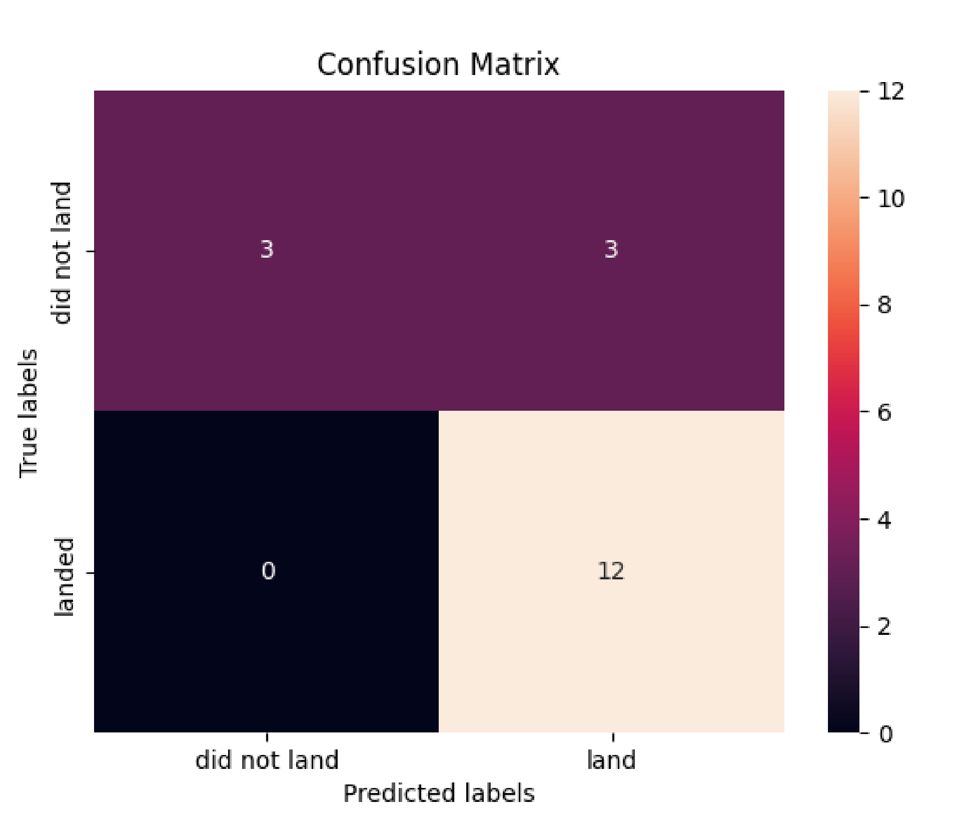


Confusion Matrix

The confusion matrix of the decicion tree is shown. The classifier had to make a total of 18 predictions.

12 successful landings were correctly predicted, or 3 cases in which the landing failed.

There were also 3 erroneous predictions. All three were real failed landing attempts which were predicted to be successful. There was no case where a failed landing was predicted as a successful one.



Conclusions

- the years 2010 to 2013 were less successful (possibly also used to adapt and gather experience) and from 2013 to 2020 the success rate rose continuously
- with an increasing number of launch attempts at a launch site, the chance of success also increased
- A payload that is too high or very low also has a negative effect on the probability of success
- Launch site with the highest success rate was KSC LC-39A
- decision tree classifier is the best classifier for this task.

Appendix

- Folium does not render map views in the github, so the maps are only visible in the presentation

Thank you!