IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thorge Spehr
4 August 2024

Outline

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

Executive Summary

- This case study showcases the analysis of the success rate of SpaceX – launches using Machine Learning algorithms. From gathering the data via webscraping, connecting to an API, cleaning and standardizing the data to making predictions via Machine learning the whole process is shown.
- Several key insights can be shown about factors that influence the outcome of launches. On top of that predictions can be made with high certainty about future launches. Through this the success of launches can be improved.

Introduction

- Launches of rockets are expensive. SpaceX is leading in providing cheaper rocket launches to customers. An important success factor for that is whether or not the booster stage will land back on earth successfully.

- This is why it is important to find the answers for why launches are successful so that the success rate of launches can be improved upon. Through Machine Learning predictions of future launches can be made that show whether or not a launch shall be done given the return values of the predictions.

Section 1

Methodology

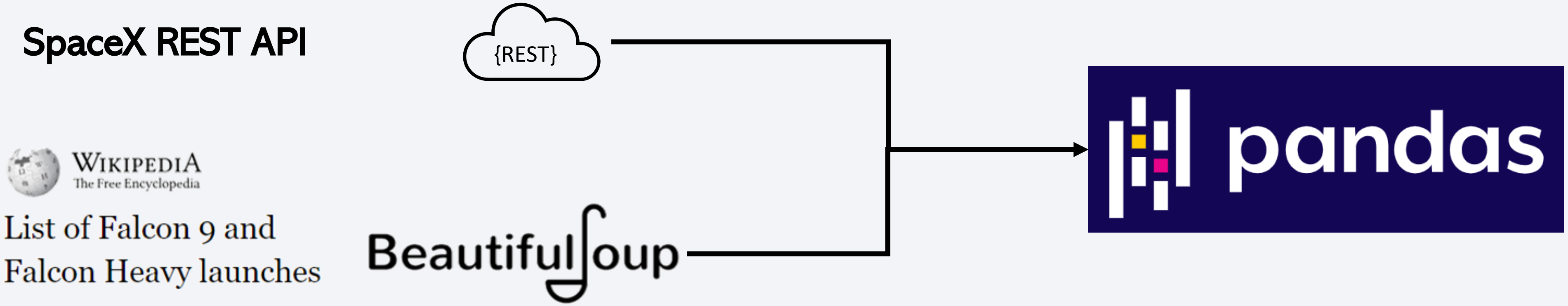
Methodology

Executive Summary

- Data collection methodology:
  - Collecting launch data via SpaceX-API and webscraping
- Perform data wrangling
  - Cleaning the data and preparing it Machine Learning Algorithms
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune and evaluate classification models

Data Collection

- Data sets were collected using the open source SpaceX-API (github.com/r-spacex/SpaceX-API)
- Additional data was collected from the list of Falcon 9 and Falcon Heavy launches` Wikipage updated 9th June 2021 using webscraping

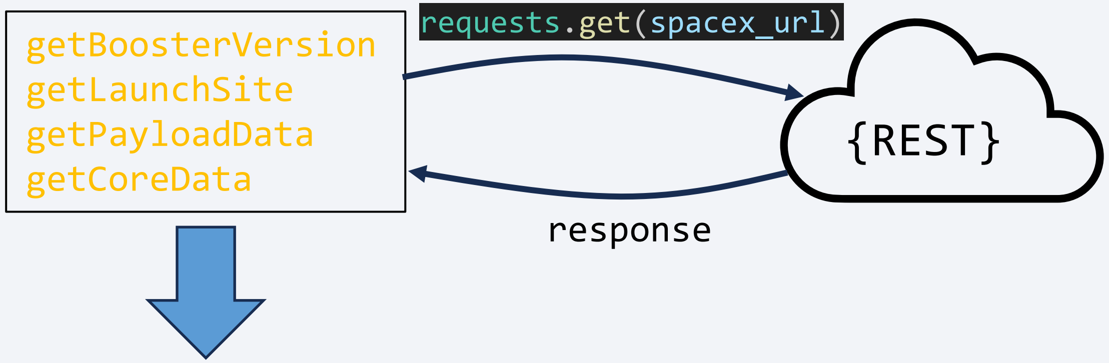


Data Collection – SpaceX API

- Data was taken from API answers and relevant fields were extracted

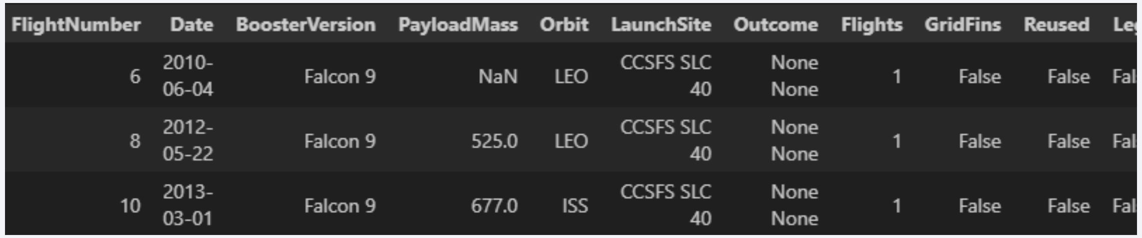
- GitHub URL to corresponding Notebook for reference:

https://github.com/spehr95/IBM-Data-Science-Capstone/blob/7f751e6de7d438ae67cce7b432639325fa19021b/jupyter-labs-spacex-data-collection-api.ipynb



```
requests.get(spacex_url)
```

```
getBoosterVersion
getLaunchSite
getPayloadData
getCoreData
```



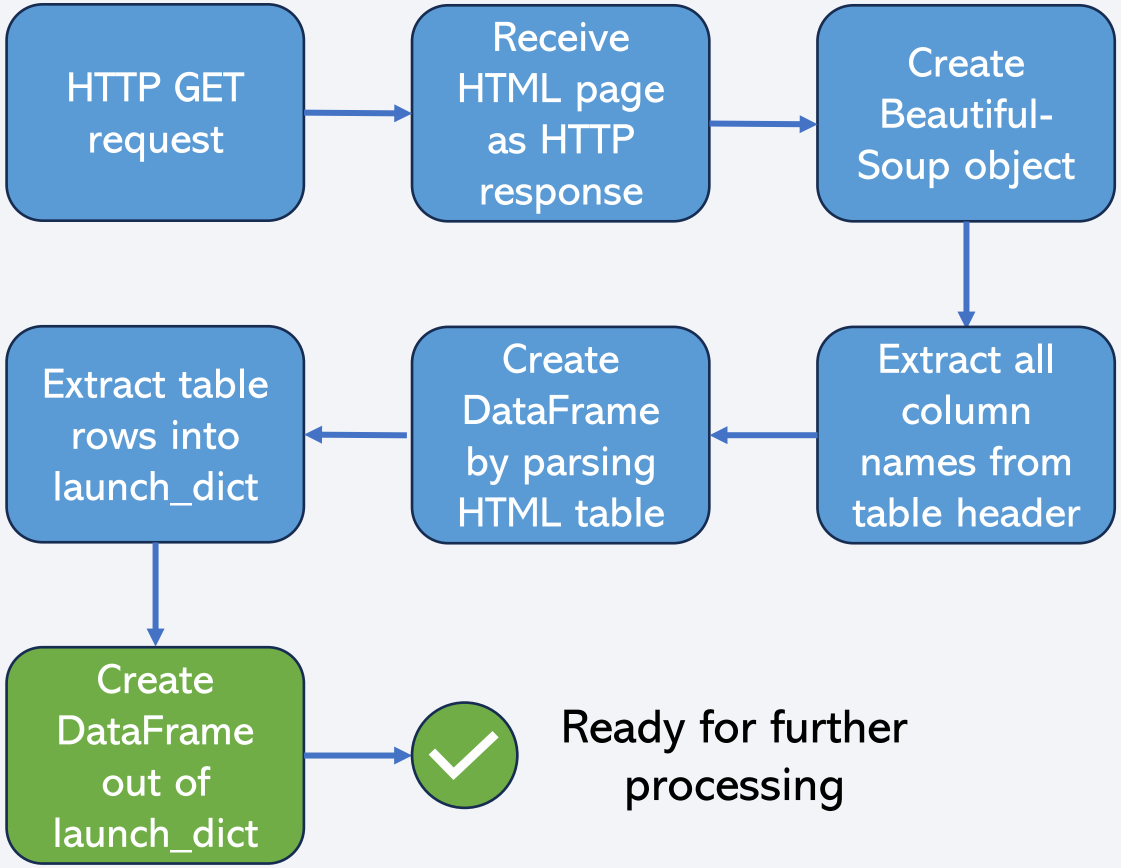
FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Le
6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	Fal
8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	Fal
10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	Fal

Data Collection - Scraping

- Additional data was scraped from the Wikipedia page of Falcon Heavy launches
  (last updated 9th June 2021)

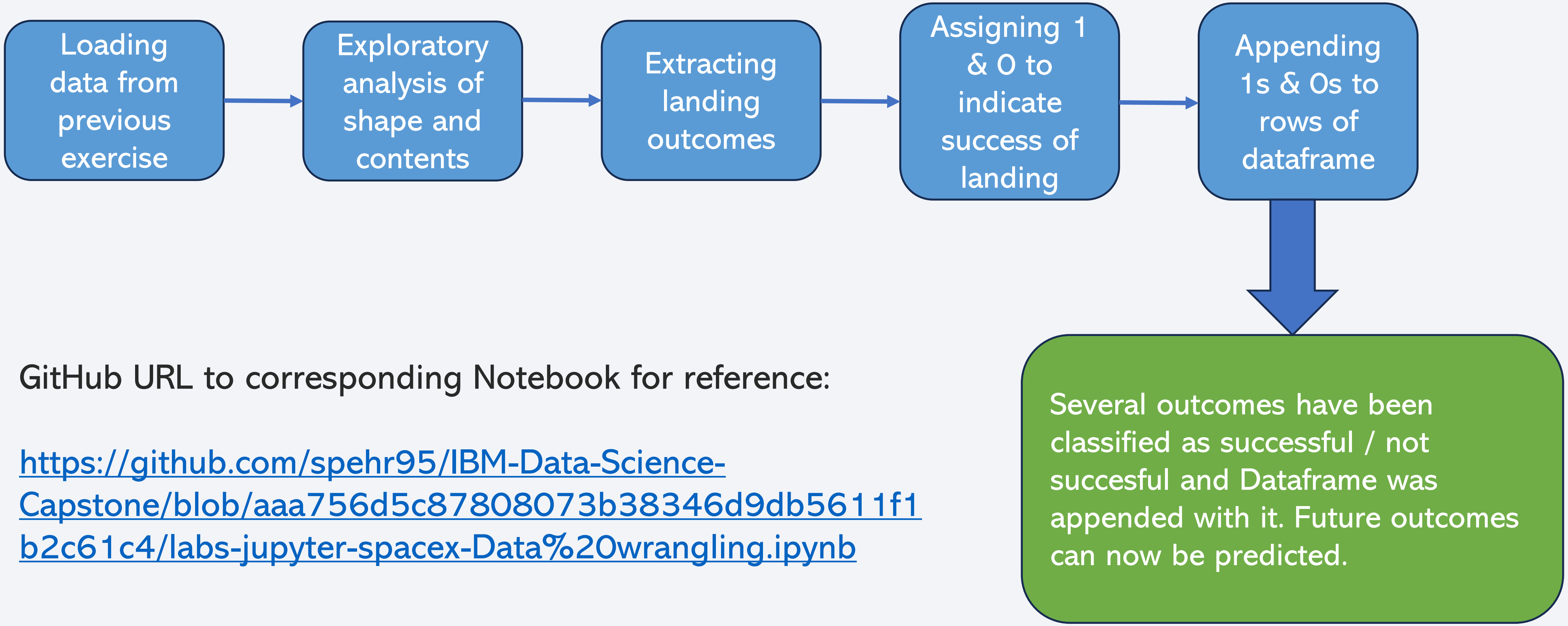
- GitHub URL to corresponding Notebook for reference:

https://github.com/spehr95/IBM-Data-Science-Capstone/blob/a72c3f72047a224536a477b399b0014511eca5a8/jupyter-labs-webscraping.ipynb


HTTP GET request
Receive HTML page as HTTP response
Create Beautiful-Soup object
Extract all column names from table header
Create DataFrame by parsing HTML table
Extract table rows into launch_dict
Create DataFrame out of launch_dict
✓
Ready for further processing

Data Wrangling

- Collected data needed to be set up to use it for Machine Learning Algorithms



Loading data from previous exercise

Exploratory analysis of shape and contents

Extracting landing outcomes

Assigning 1 & 0 to indicate success of landing

Appending 1s & 0s to rows of dataframe

Several outcomes have been classified as successful / not succesful and Dataframe was appended with it. Future outcomes can now be predicted.

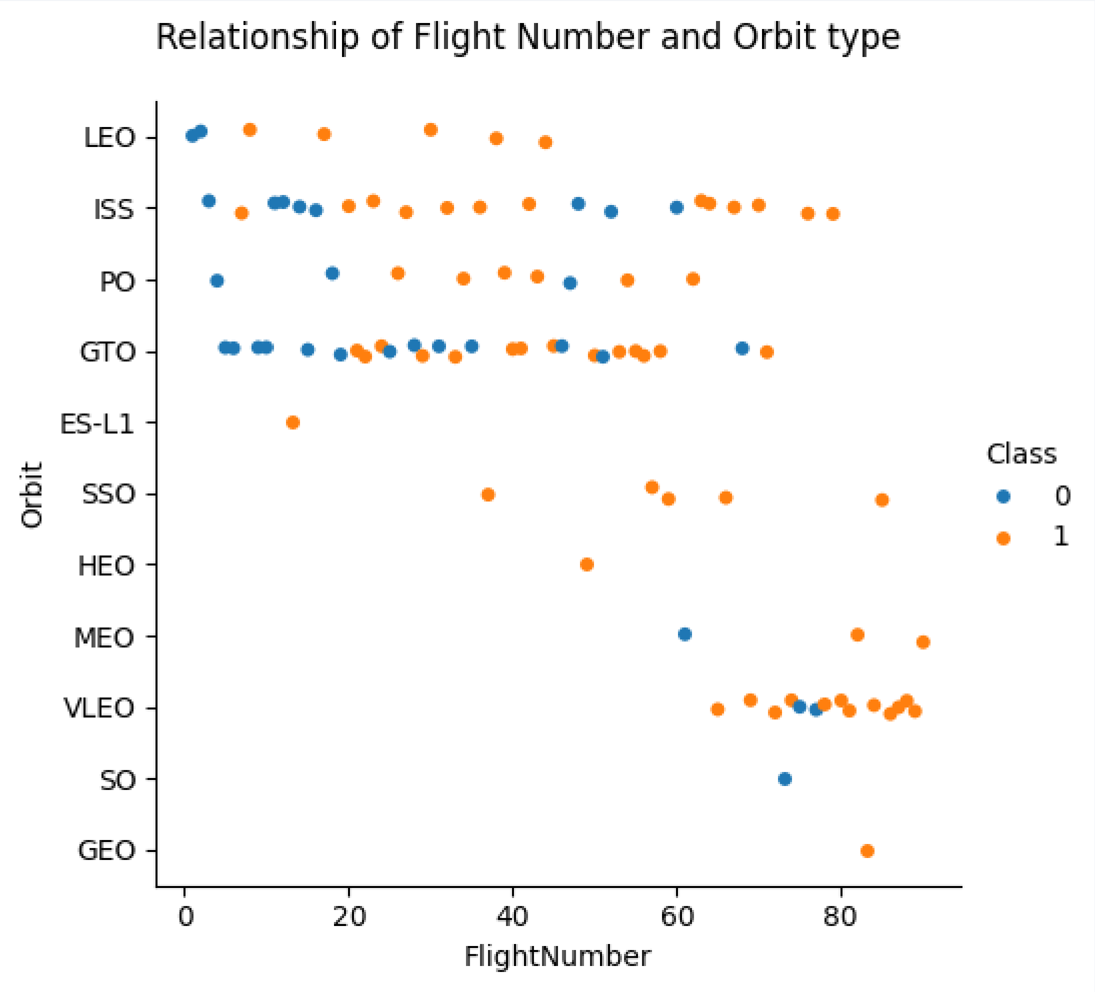
GitHub URL to corresponding Notebook for reference:

https://github.com/spehr95/IBM-Data-Science-Capstone/blob/aaa756d5c87808073b38346d9db5611f1b2c61c4/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

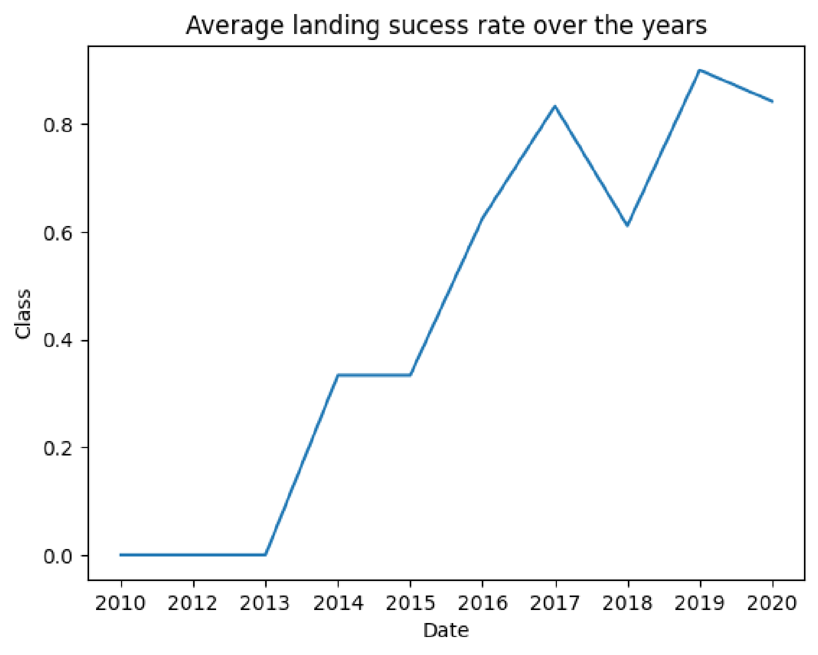
- Data was analyzed to find the driving factors behind landing success
- Mainly Seaborn-catplots were used to find correlations between landing success and:
  - Flight number & Payload mass
  - Launch site & Flight Number
  - Payload mass & Launch site
  - Orbit and Flight Number

GitHub-Link for Notebook: https://github.com/spehr95/IBM-Data-Science-Capstone/blob/d7d5016b10acc67c6d11bba5933f92b283225394/Independent%20Labs_module_2_edadataviz.ipynb



EDA with Data Visualization

- Further analysis was made with bar plots and lineplots
- Goal was to find out average success rate for launch sites and years
- Insights gathered:
  - Success rate increased over the years dramatically. This is probably due learning effects in manufacturing and launching the rockets
  - Out of the features examined, Payload Mass seemed to be a good estimator for landing success.
  - **Important to note:** Payload Mass increased with Flight Number



EDA with SQL

- Connected to database using **sqlite3**
- Extracted unique launch sites using sql- `DISTINCT` – command
- Listed Payload masses and found average to be at **2,928.4 kg.**
- Found out date of first successful landing through `MIN` – function
- Queried for booster names of successful landings between 4 tons and 6 tons of Payload mass
- Used `COUNT` – function to query for number of successful landings per type of landing
- Used subquery to filter for boosters who carried max payload
- ...

GitHub: https://github.com/spehr95/IBM-Data-Science-Capstone/blob/d0dcc56d195791672370d5b55d1f7c06cade71b9/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers together with circles were created to highlight single sites (e.g. NASA HQ)
  - Popups were added to show further information of markers
- Marker Clusters were used to show different landing outcomes for the same locations (green for successful landing, red for unsuccessful landing)
  - Marker Clusters automatically expand upon clicking on them
  - This makes them also useful when there are multiple markers close by on the map
- MousePosition was added to the map. This enables the indication of the coordinates of the current mouse position on the map
- Lines were created to show distances (i.e. distance to sea)

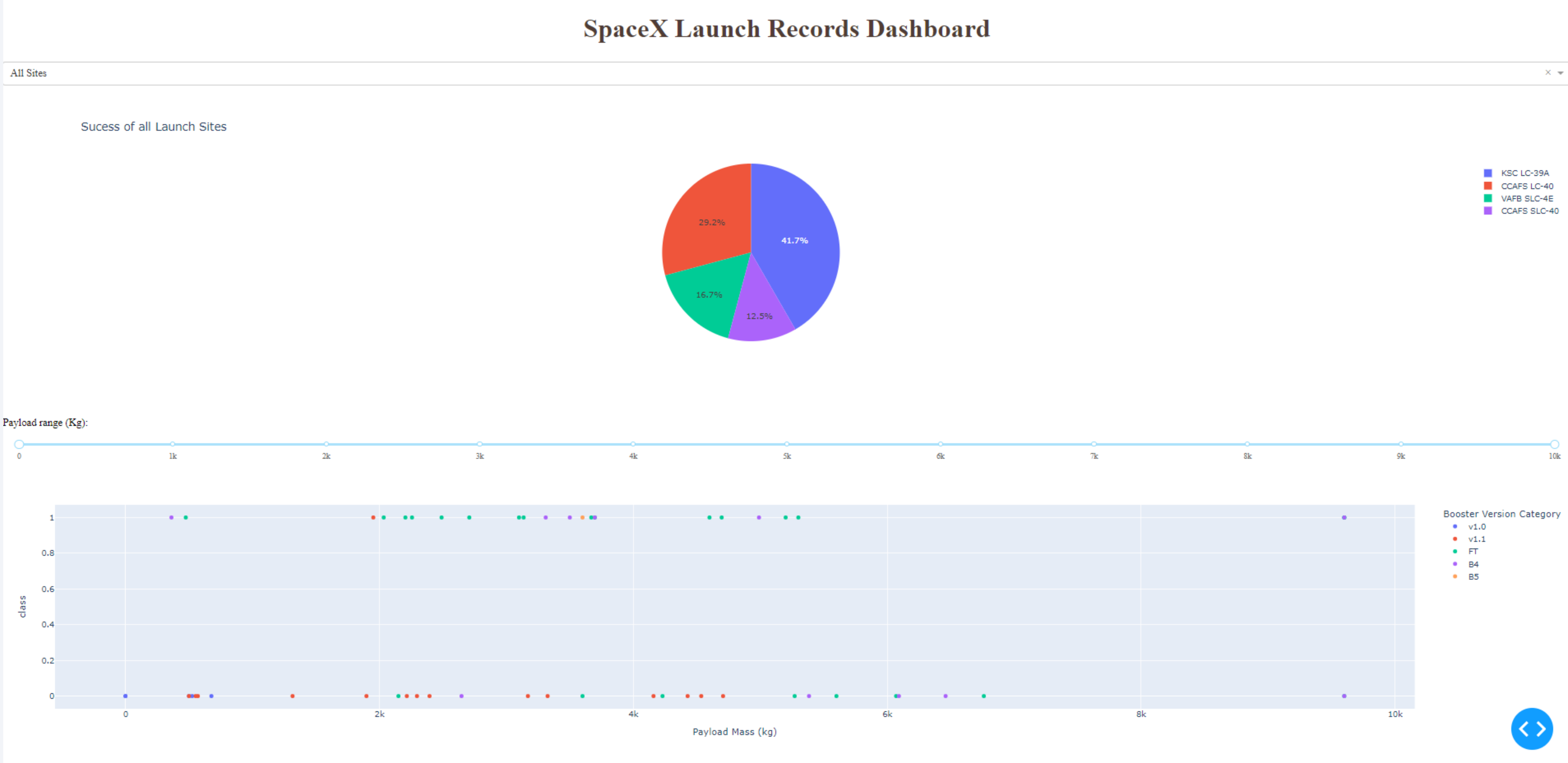
GitHub:https://github.com/spehr95/IBM-Data-Science-Capstone/blob/301e5e08f7ce6af5da4fd621dd72188191e4a9de/DS0203EN_module_3_Independent%20Labs_module_3_lab_jupyter_launch_site_location.backup.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was created to let the user find how different features of the dataset are related.
- Through a dropdown menu the user can select launch sites whose success rate is shown in the following pie chart.
  - Selections can be made either for single launch sites or for all launch sites at once.
  - This was done to make the user quickly see which launch site has been most succesful
- Using a range slider the user can select payload ranges. The choices made there are reflected in the scatter plot below.
  - Scatter plot shows payload mass, indication of success and booster version category
  - This way the user can easily see how landing outcome is influenced by payload mass

GitHub: https://github.com/spehr95/IBM-Data-Science-Capstone/blob/87f1830d0257a4caf7dd920b35f9fd288068a477/spacex_dash_app.py

Build a Dashboard with Plotly Dash



SpaceX Launch Records Dashboard

All Sites

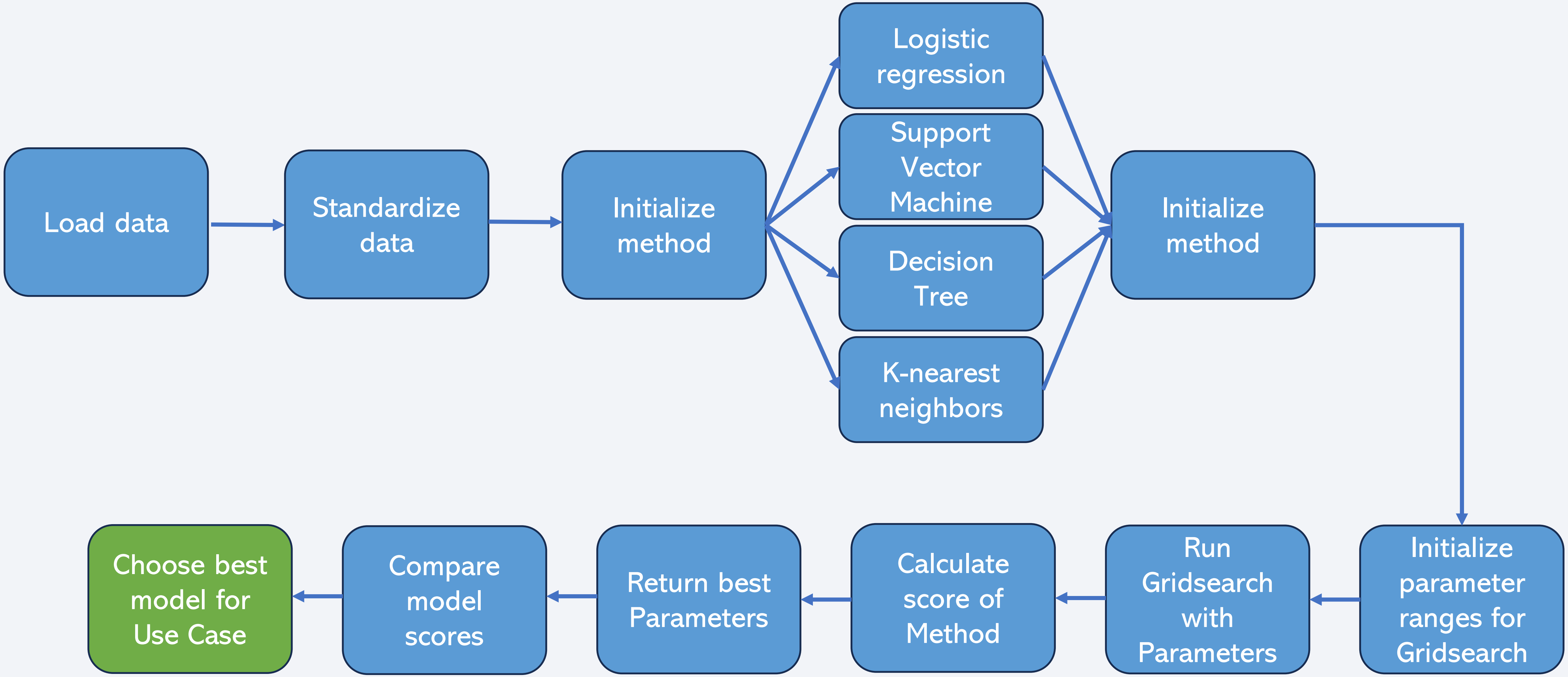
Payload range (Kg):

Predictive Analysis (Classification)

- Data had to be transformed to be able to be used in Machine Learning algorithms
- Features were standardized using the `preprocessing.StandardScaler()` – method → all columns now have average of 0 and standard deviation of 1.
- 80% of data entries were split into the training data set
- Logistic Regression, Support Vector Machines, Decision Trees, k-nearest neighbors were tested on the data set
- Through GridSearch many different parameters were tested
- Score was calculated to determine the performance of every method

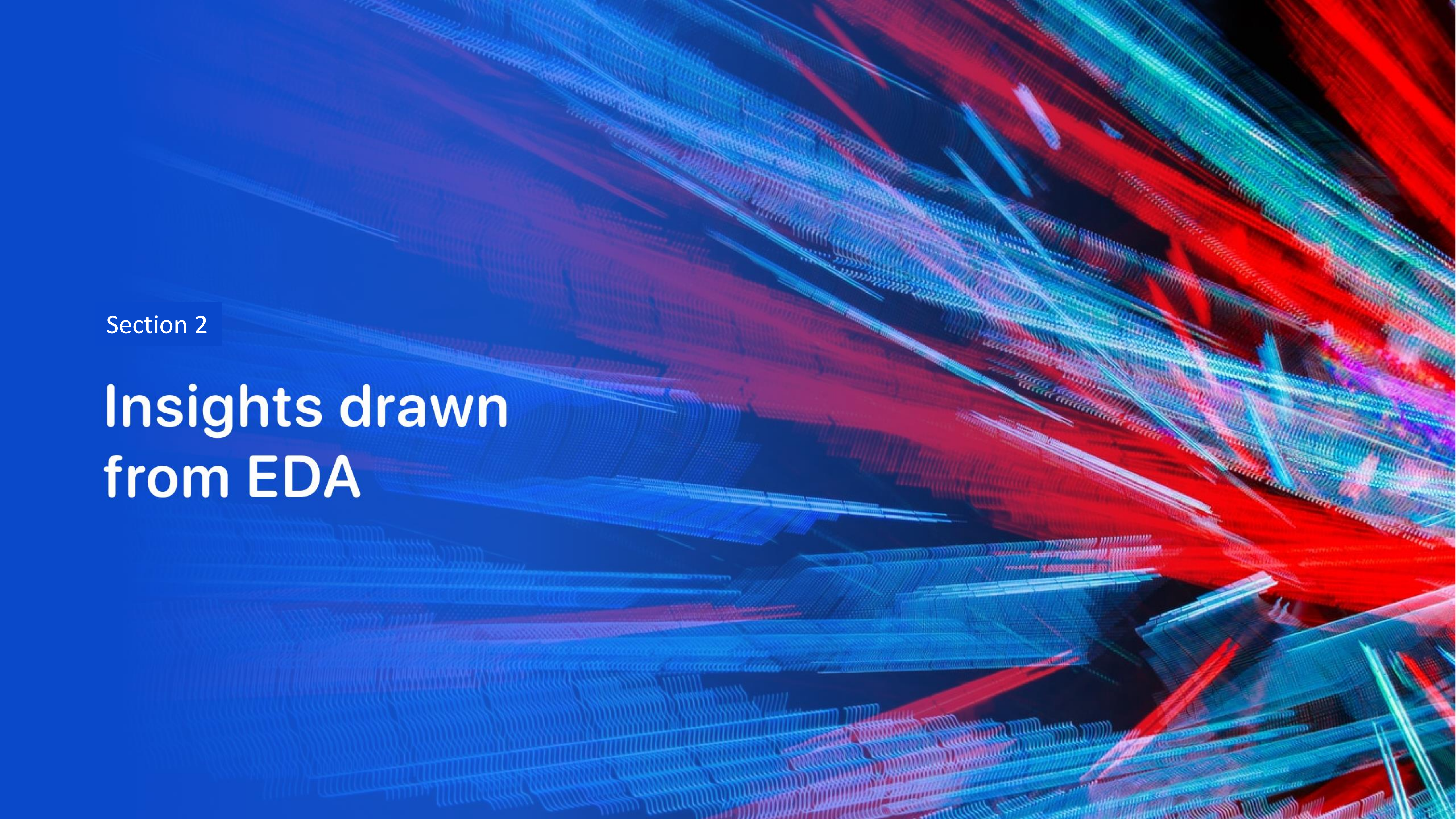
GitHub: https://github.com/spehr95/IBM-Data-Science-Capstone/blob/136deb6ca6272bf43b97be2f20df7c6c4762358b/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)



Results

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

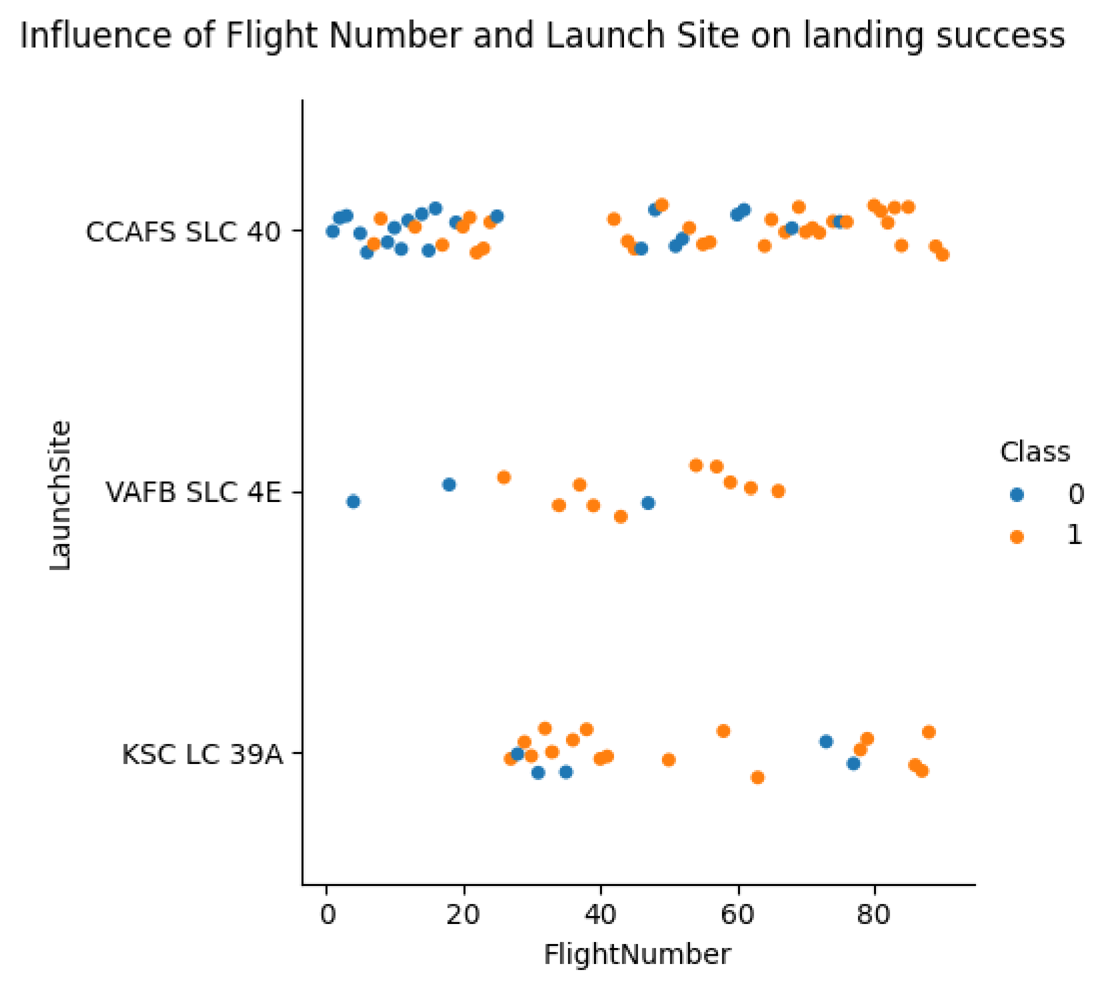


Section 2

Insights drawn from EDA

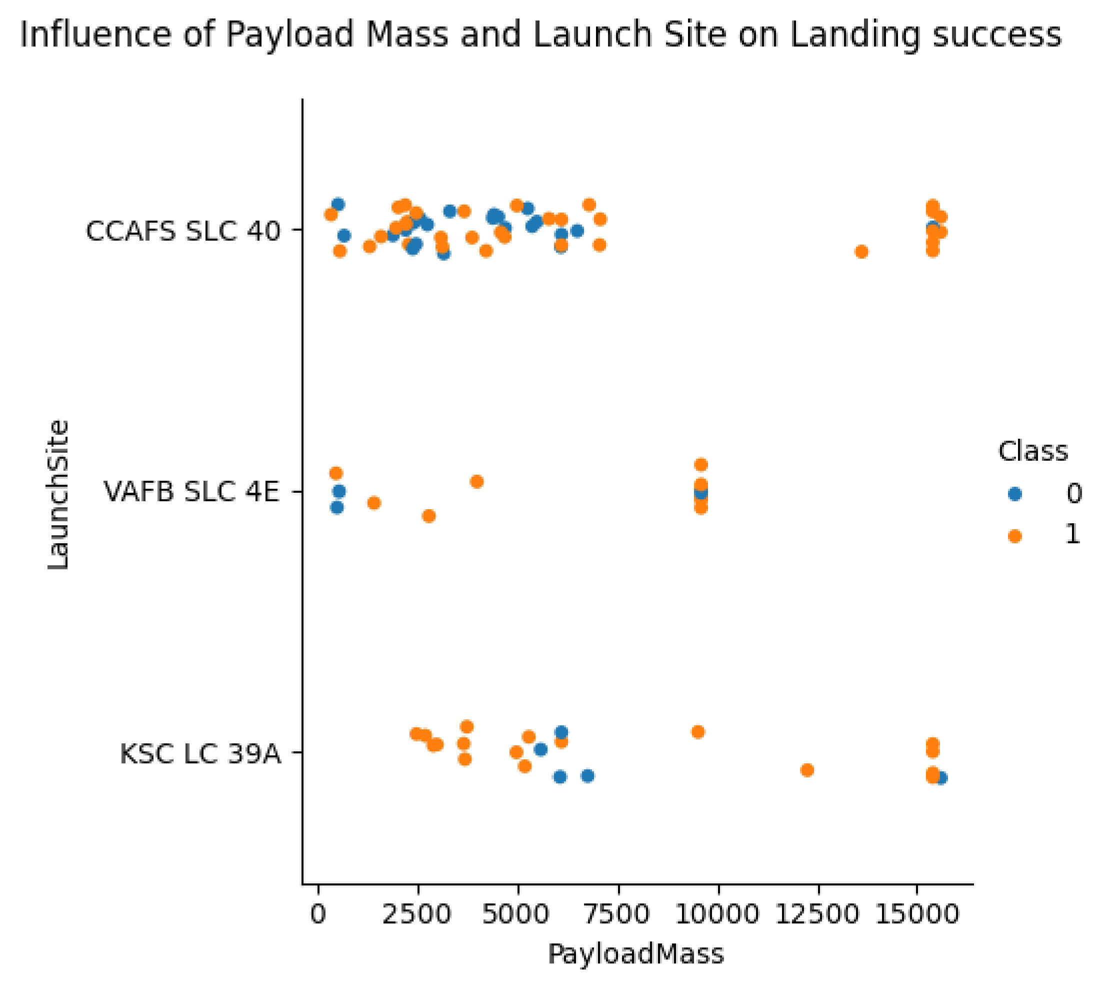
Flight Number vs. Launch Site

- Data shows that FlightNumber has a tremendous influence on Landing success.
- This is why over all launch sites the success rate is improving over time
- KSC LC39A looks most successful, but it did not have starts from the beginning



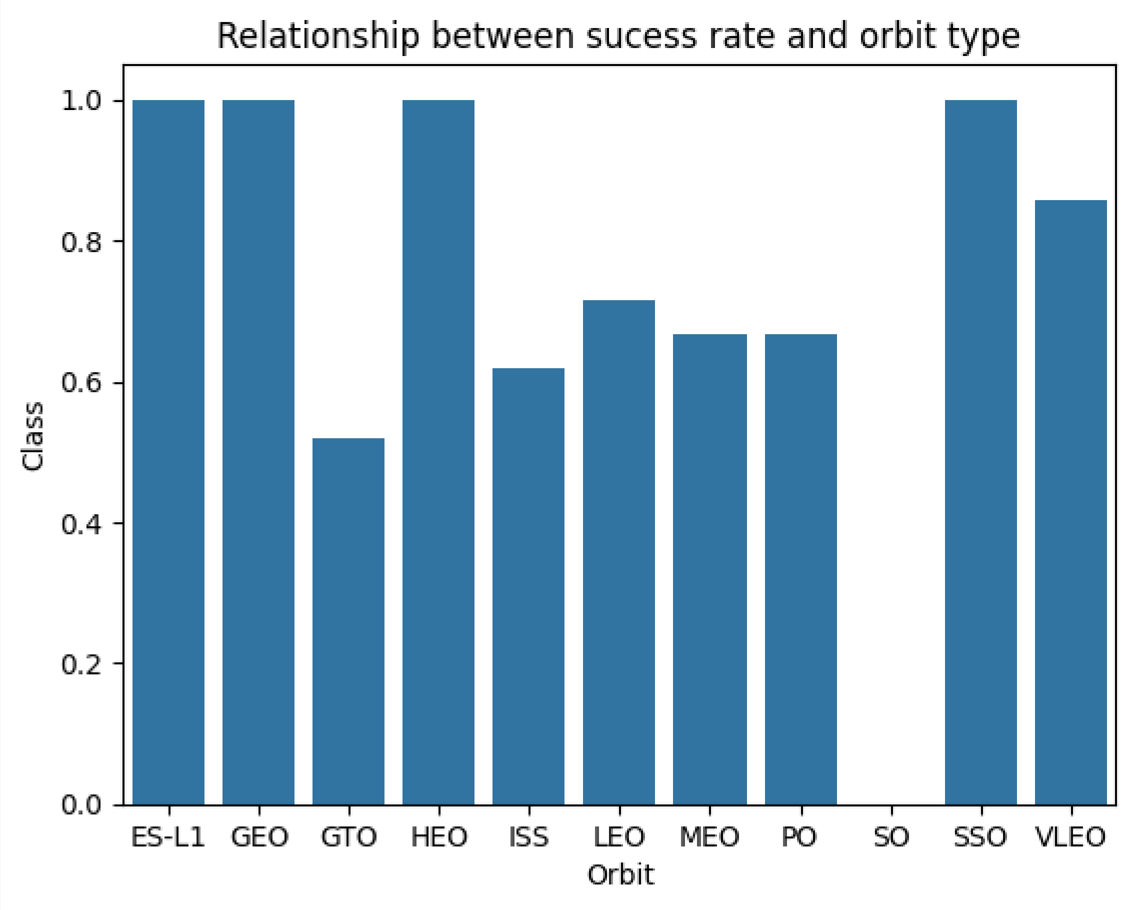
Payload vs. Launch Site

- The higher the payload, the higher the success rate suggests the data
- KSC LC 39A is very succesful for low payloads, CCAFS SLC 40 is successful with high payloads
- VAFB SLC 4E does not seem to be suited for high payloads.



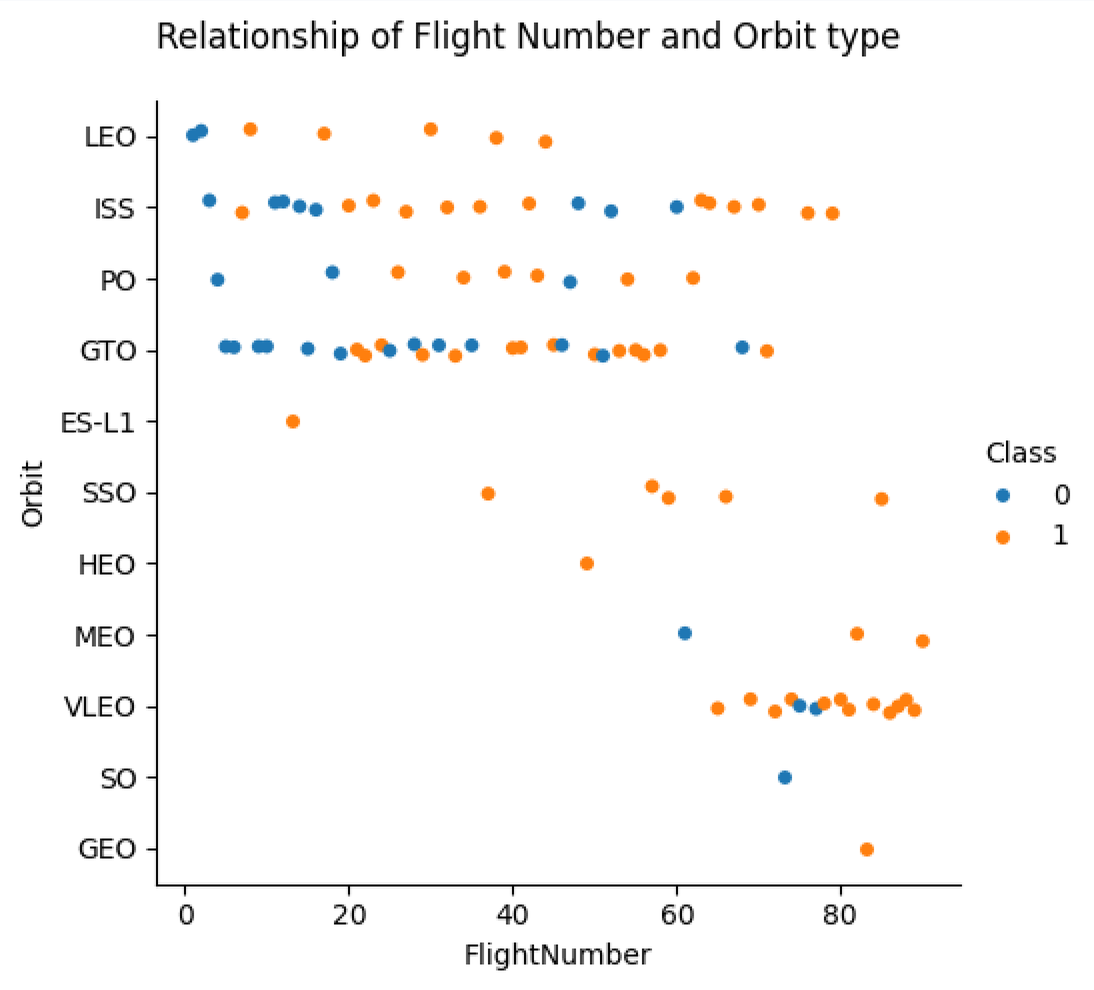
Success Rate vs. Orbit Type

- The rather high orbits have the highest success rate of landing on Earth again



Flight Number vs. Orbit Type

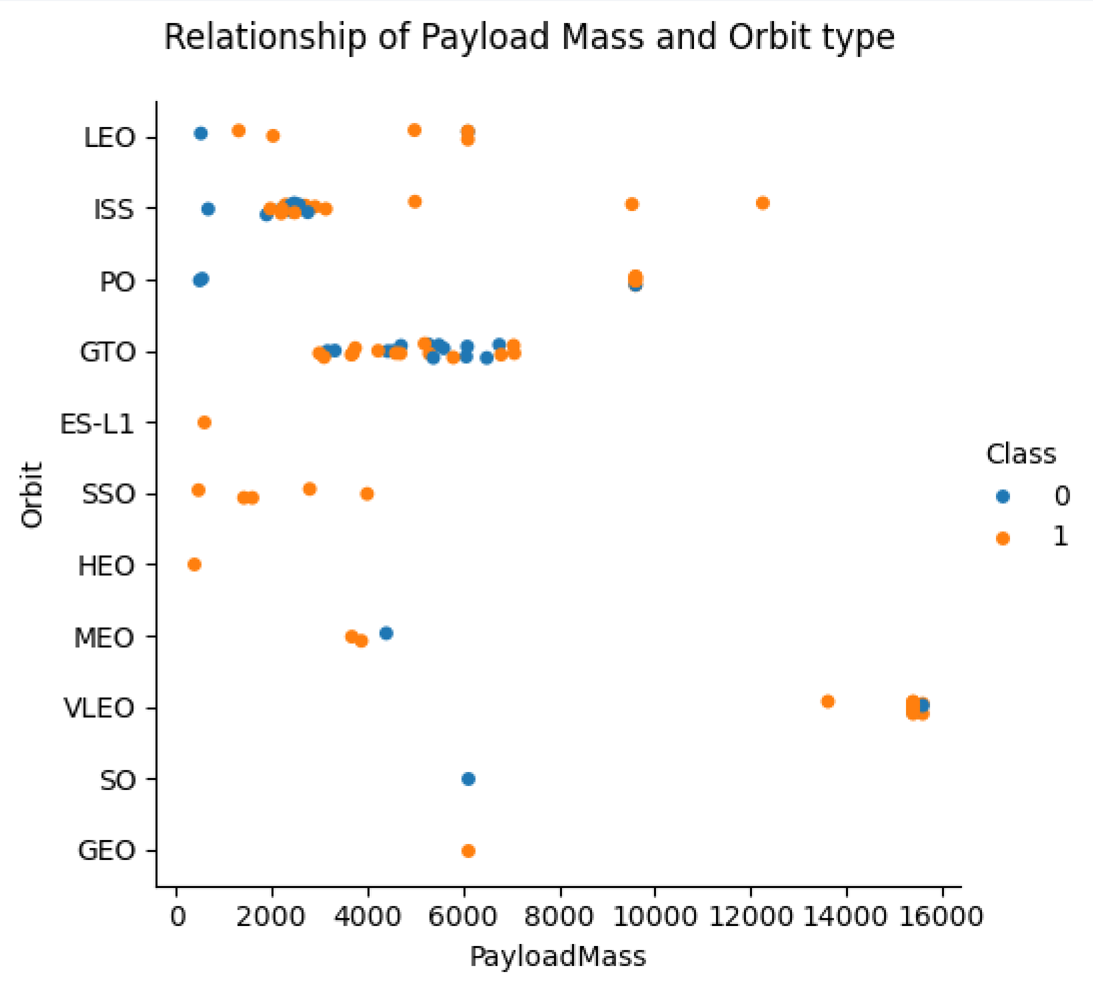
- You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



Payload vs. Orbit Type

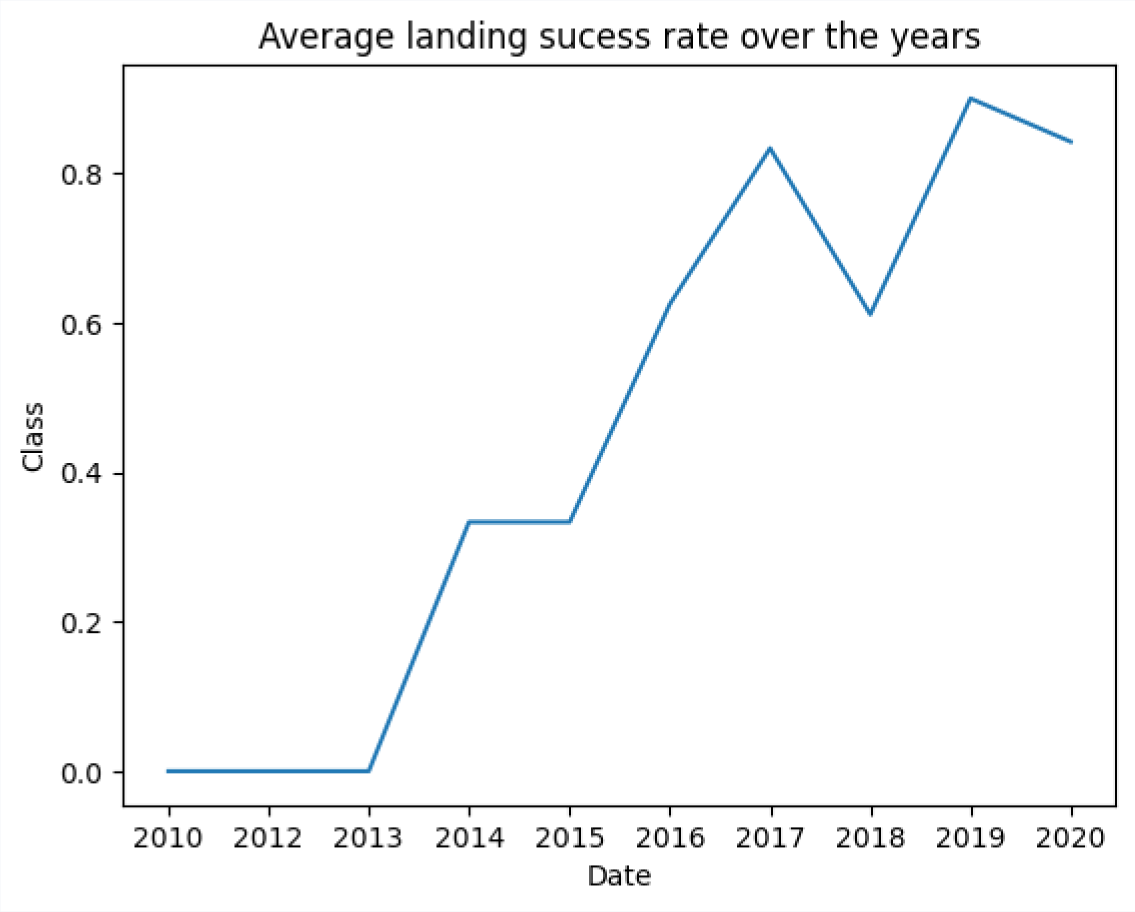
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



Launch Success Yearly Trend

- It can be observed that the success rate since 2013 kept increasing until 2020



All Launch Site Names

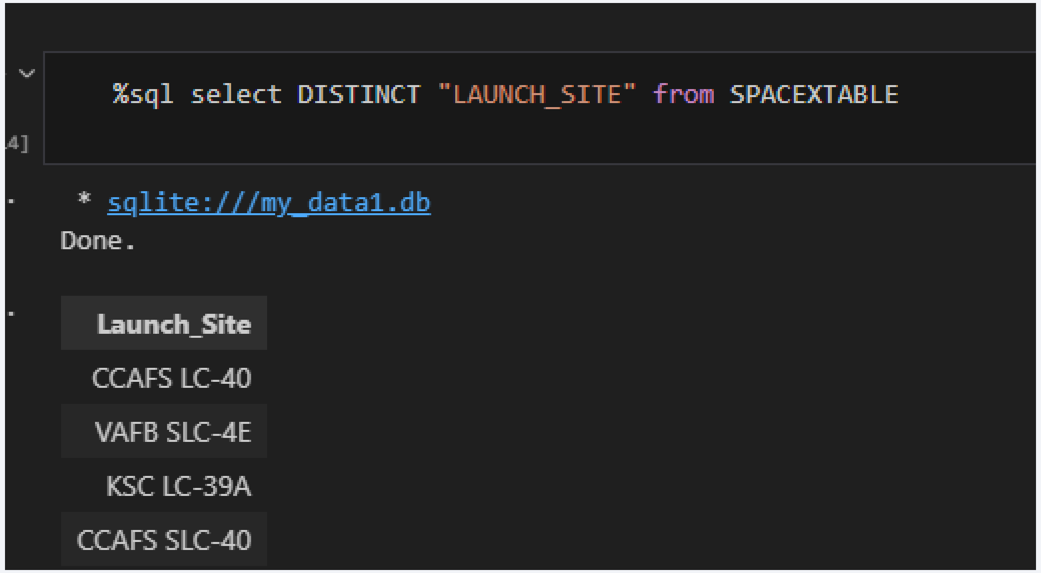
CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

- Unique launch site names were obtained from the sql-Database using the `DISTINCT` function



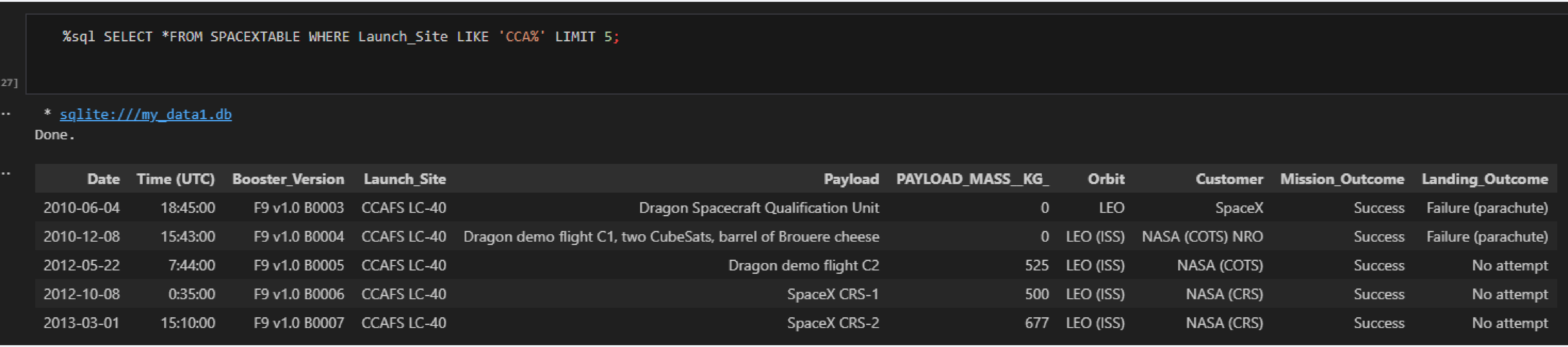
```
%sql select DISTINCT "LAUNCH_SITE" from SPACEXTABLE
```

```
 * 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'

- Using the `LIKE 'CCA%' LIMIT 5` –method, the first entires of launch sites beginning with ‘CCA’ could be obtained.



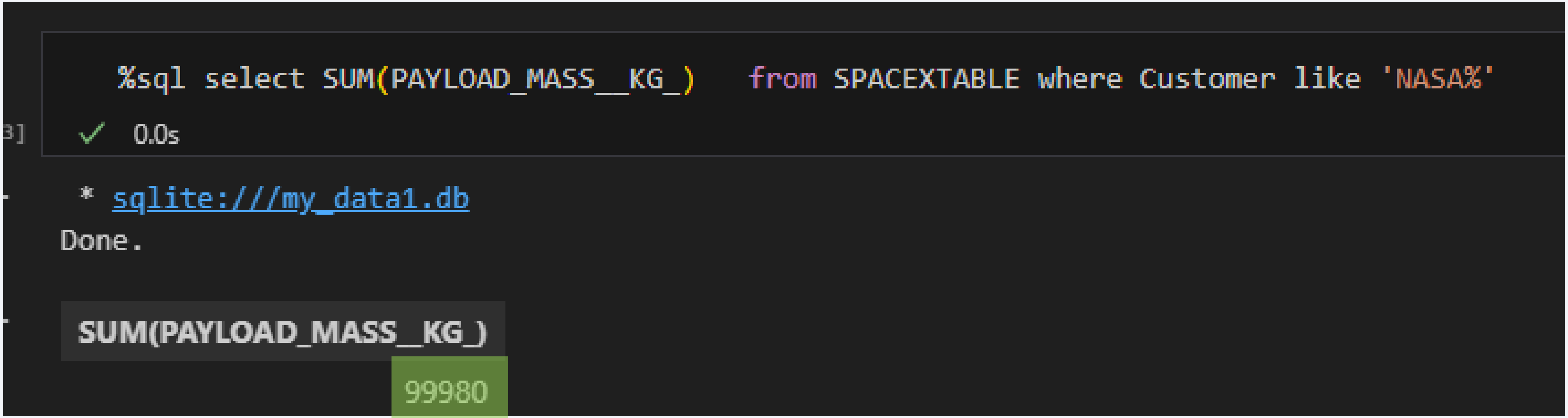
```
%sql SELECT *FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Total payload carried by boosters from NASA: **99,980 kg**
- By using the `LIKE “NASA%”` –method, the total payload mass for all launches NASA was a part of, could be calculated.



```
%sql select SUM(PAYLOAD_MASS__KG_)   from SPACEXTABLE where Customer like 'NASA%'
```

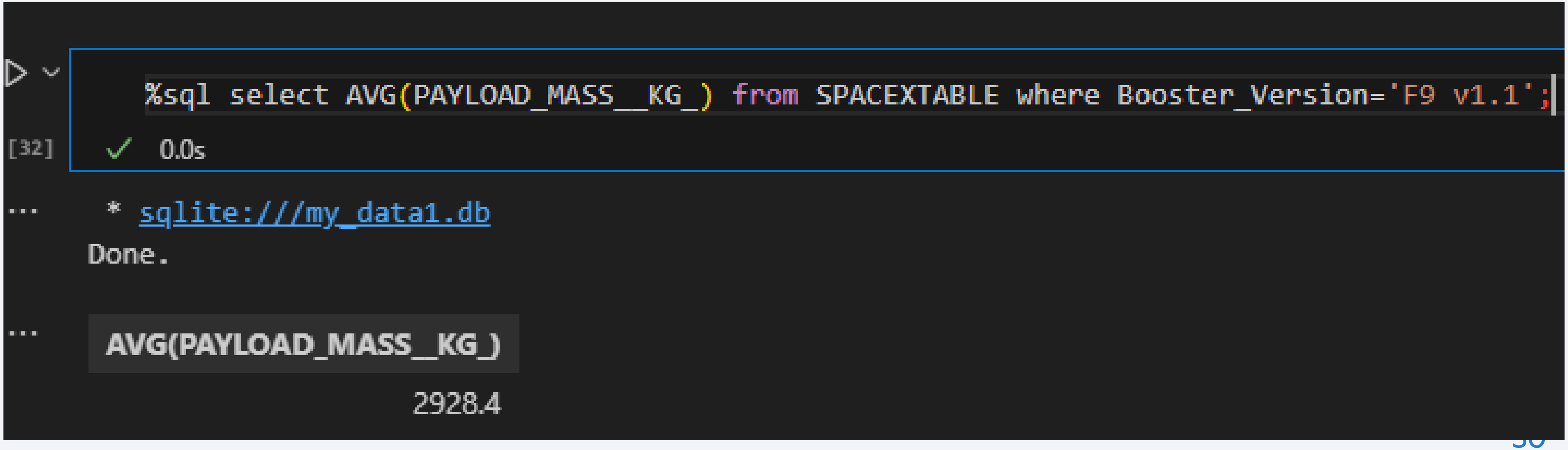
✓ 0.0s

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

SUM(PAYLOAD_MASS__KG_)
99980

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1: **2,928.4 kg**
- Using the AVG() – function the value could be calculated



```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version='F9 v1.1';
```

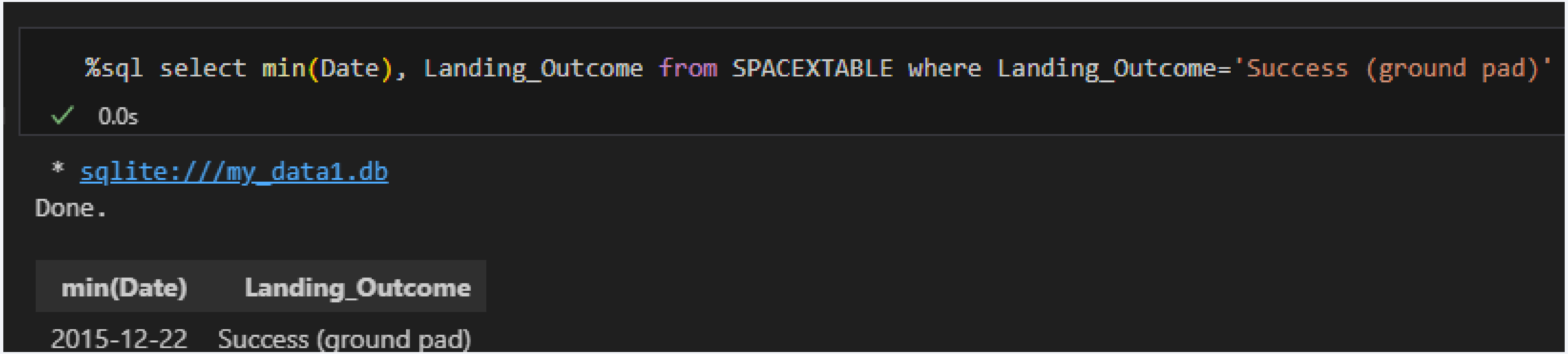
[32] ✓ 0.0s

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad: **2015-12-22**
- SQL Database detected time format of Date-column automatically which is why the min-function could be used.



```
%sql select min(Date), Landing_Outcome from SPACEXTABLE where Landing_Outcome='Success (ground pad)'
```

✓ 0.0s

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

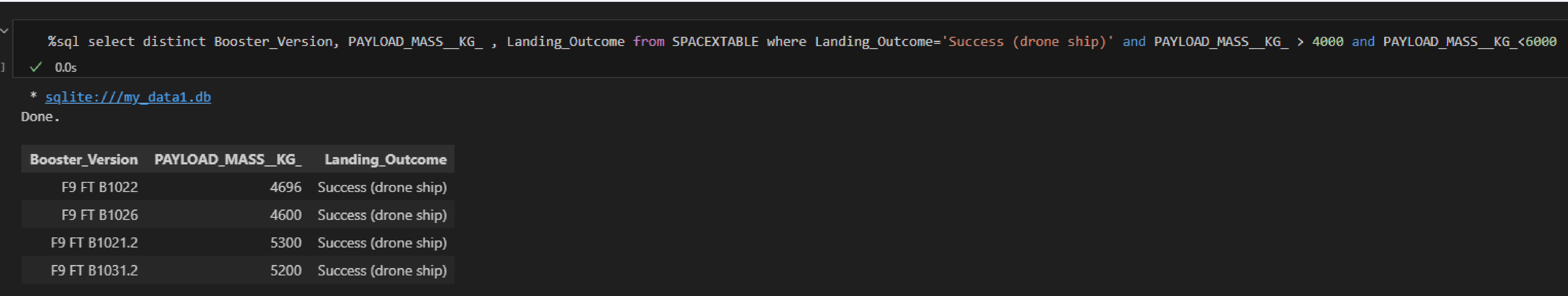
min(Date)	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- With AND- function multiple conditions can be put in.



```
%sql select distinct Booster_Version, PAYLOAD_MASS__KG_ , Landing_Outcome from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_<6000
```

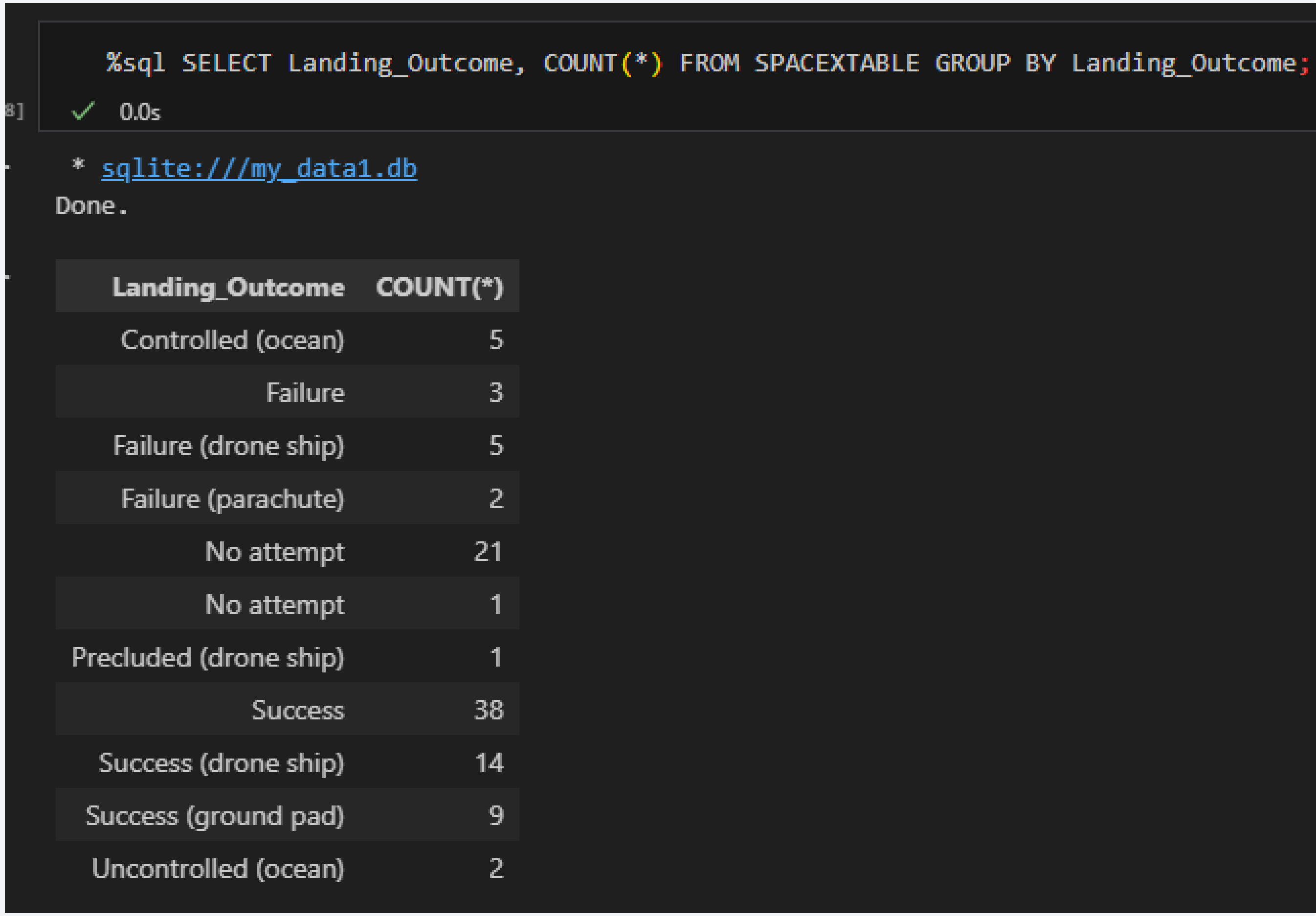
✓ 0.0s

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

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- GROUPBY – function can be used to count the occurrences of distinct categories.



```
%sql SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Landing_Outcome;
✓ 0.0s
```

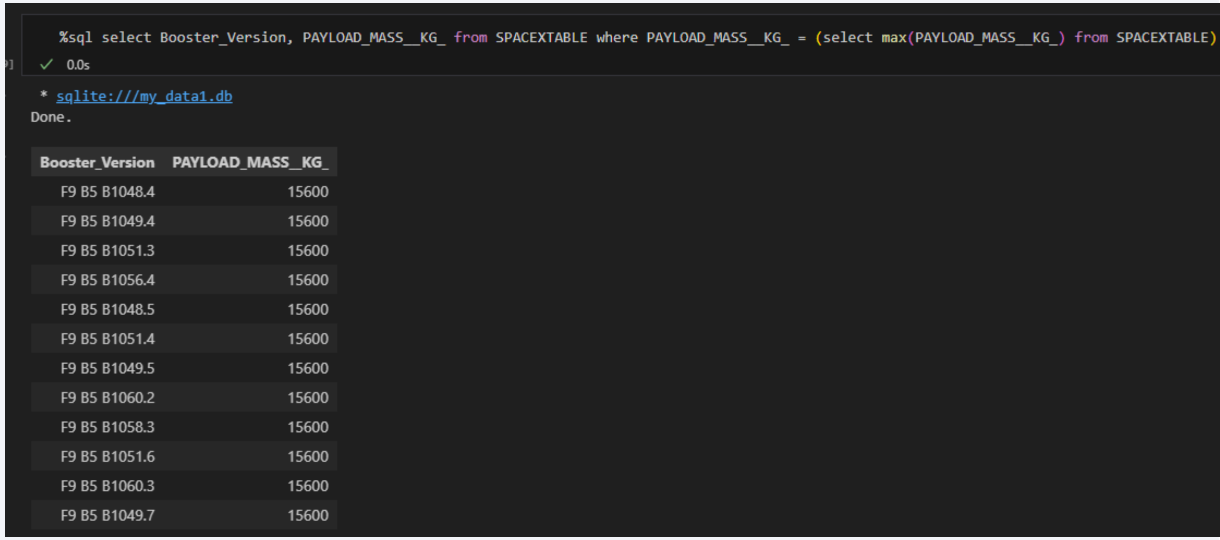
```
* sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT(*)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- List of the names of the booster which have carried the maximum payload mass:
- Subquery enables the user to ask multiple things at once from SQL –data base (calculate maximum, search for maximum)

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



```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
✓ 0.0s
```

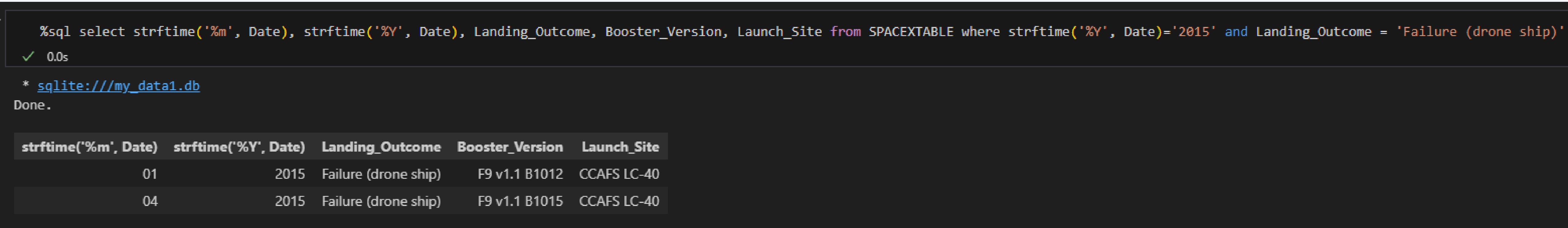
```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- List of the failed landing outcomes in drone ship, their booster versions, and launch site names in year 2015

- Function `strftime('%m', Date), strftime('%Y', Date)` was used to extract month and year from Date - column



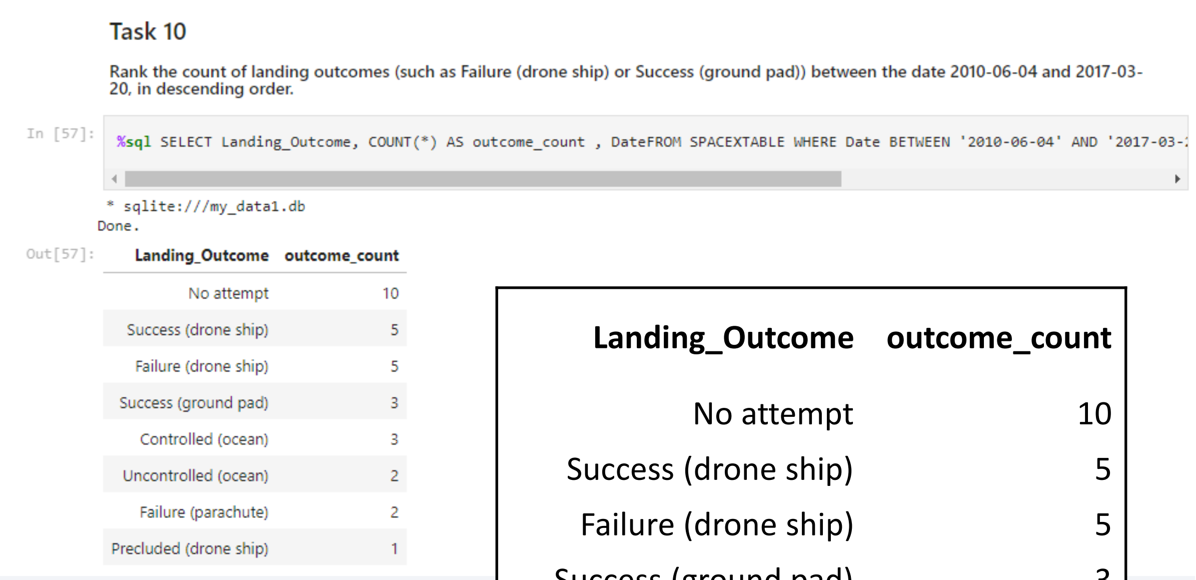
```
%sql select strftime('%m', Date), strftime('%Y', Date), Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where strftime('%Y', Date)='2015' and Landing_Outcome = 'Failure (drone ship)'
✓ 0.0s

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

strftime('%m', Date)	strftime('%Y', Date)	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- `ORDER BY` –method enables the user to rank entries.



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.

```
In [57]: %sql SELECT Landing_Outcome, COUNT(*) AS outcome_count , DateFROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-
```

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

Out[57]:

Landing_Outcome	outcome_count
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

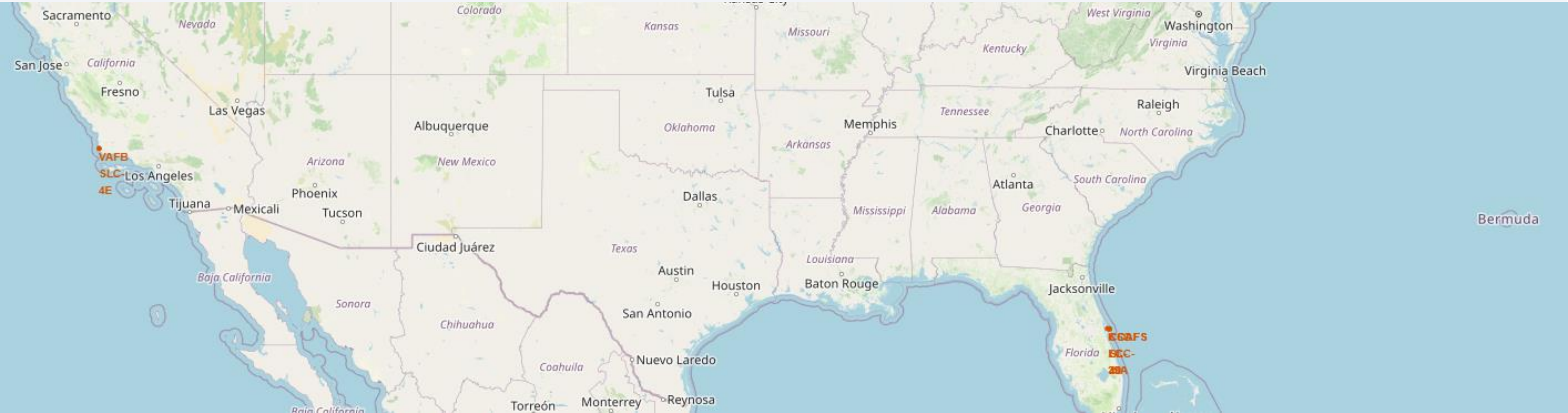
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

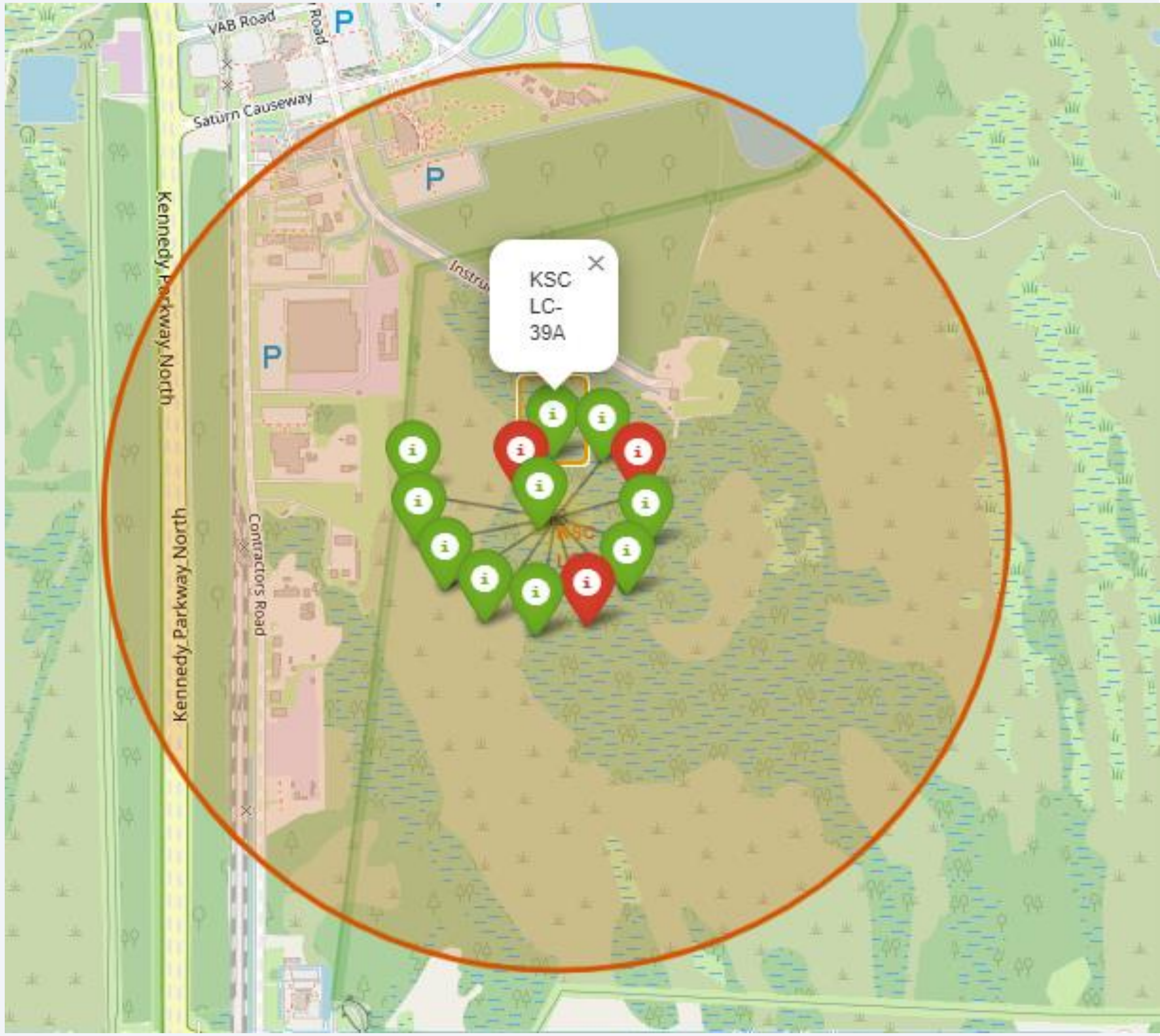
Overview of all launch sites

- All launch locations are positioned in the south of the US. This is because the vicinity to the Equator is important in order to save fuel.
- From the map it can also be seen that launch sites are close to the sea. This is due to protection of nearby residents.



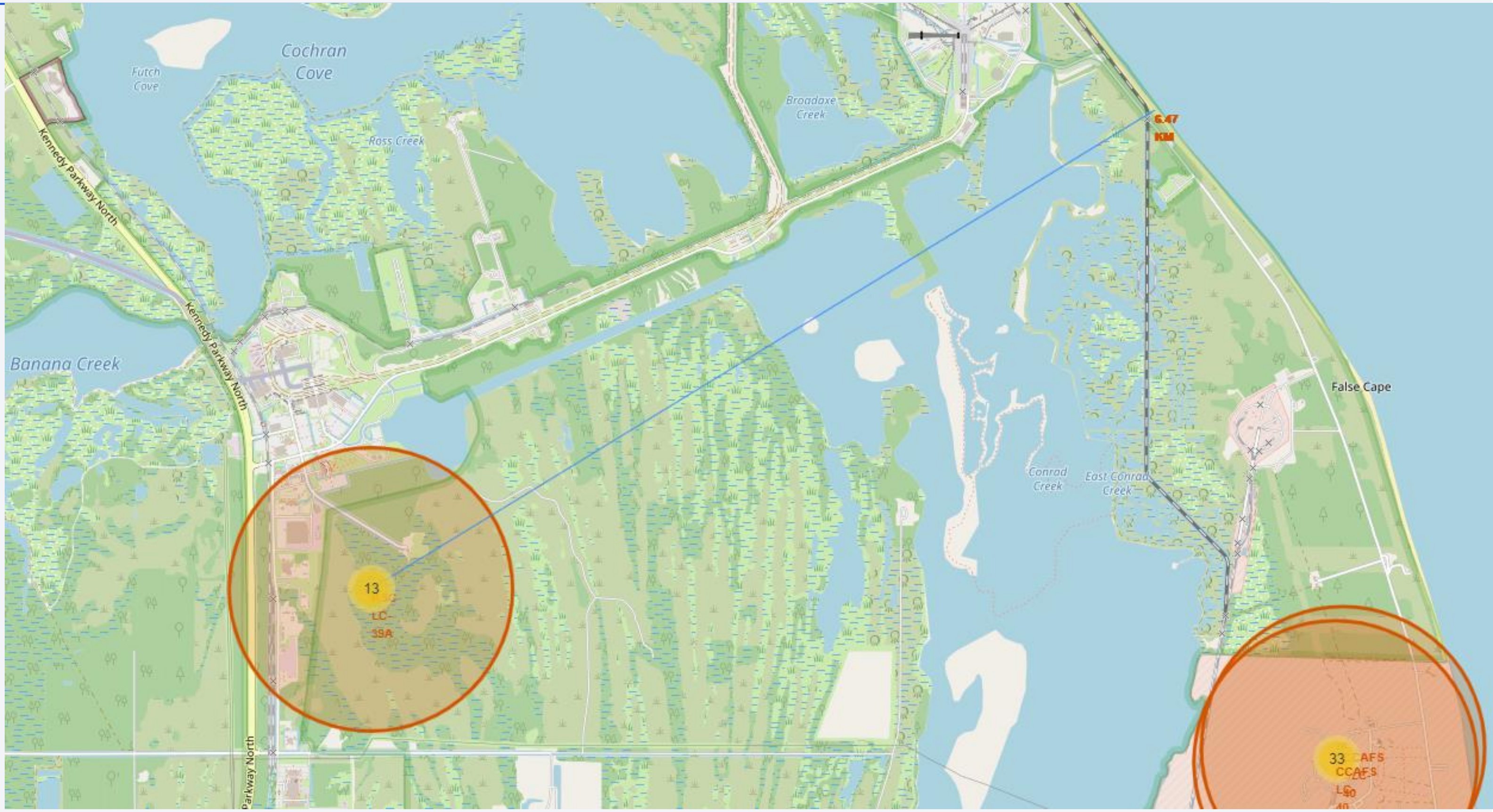
Visualization of landing outcomes

- By the use of marker clusters every launch from the same location can be easily distinguished from each other
- Coloring the markers by their respective landing outcome makes it easily visible how many landings were successful from this location and how many were not.



Analysis of landmarks in the vicinity

- Distance from KSC LC 39A to the nearest ocean is **6.47km.**
- This further emphasizes the vicinity of launch sites to the sea.
- Although the site is not directly at the ocean, its expected launch path (headed east) still leads only over uninhabited land.

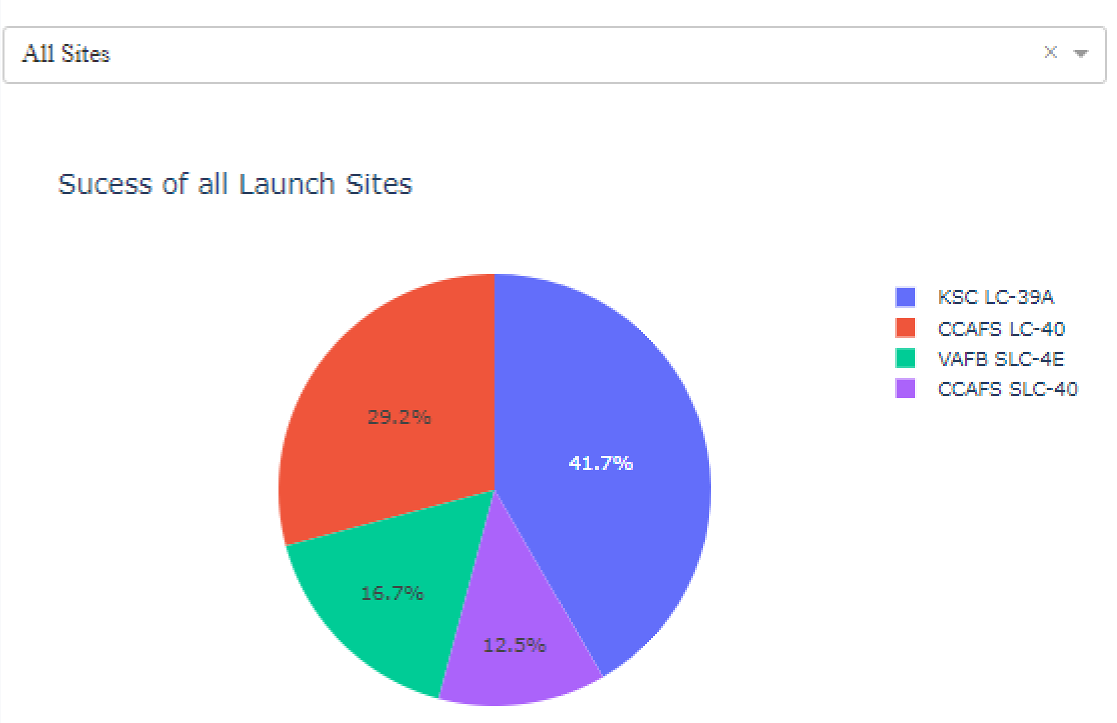


Section 4

Build a Dashboard with Plotly Dash

Pie chart of success rate of launch sites

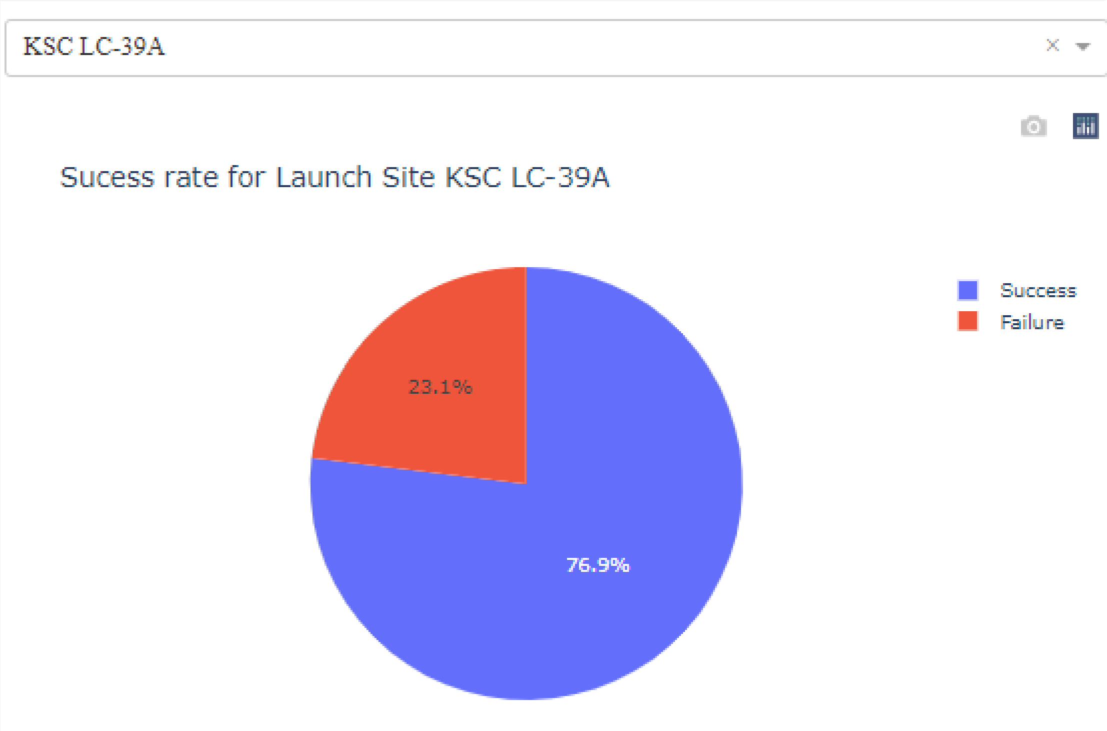
- It can be seen that overall KSC LC-39A contributed the most to successful landings, followed by CCAFS LC-40.
- CCAFS SLC-40 contributed the least.



All Sites

Pie Chart of most successful Launch Site

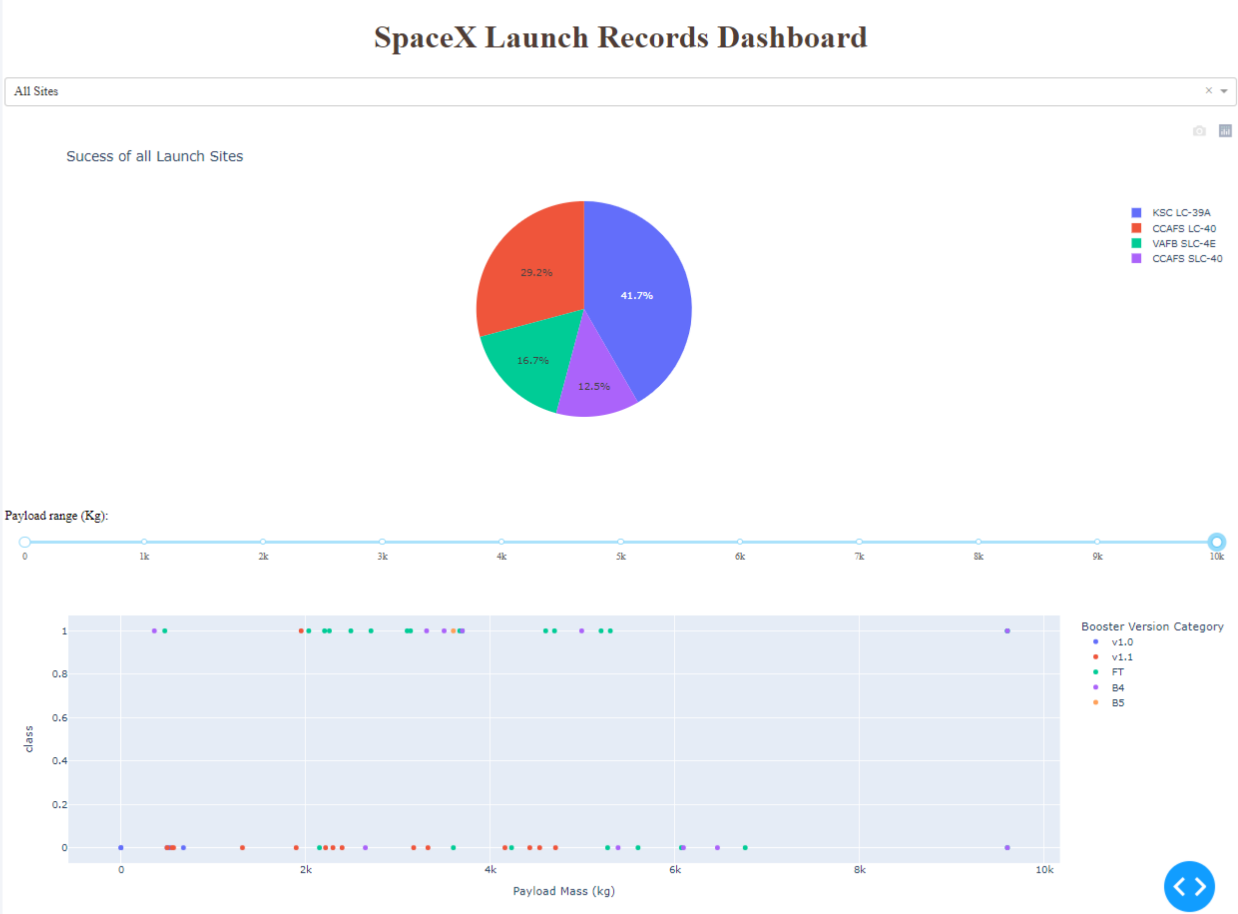
- KSC LC-39A has a high success rate
- The success rate of this site is higher than the average over all launch sites (-66%)



KSC LC-39A

Analysing payload mass against launch outcome

- The payload range between 5 tons and 8 tons has an exceptionally poor success rate
- FT booster had the most successful launches
- Launches are most successful for payload ranges between 2 tons and 4 tons.

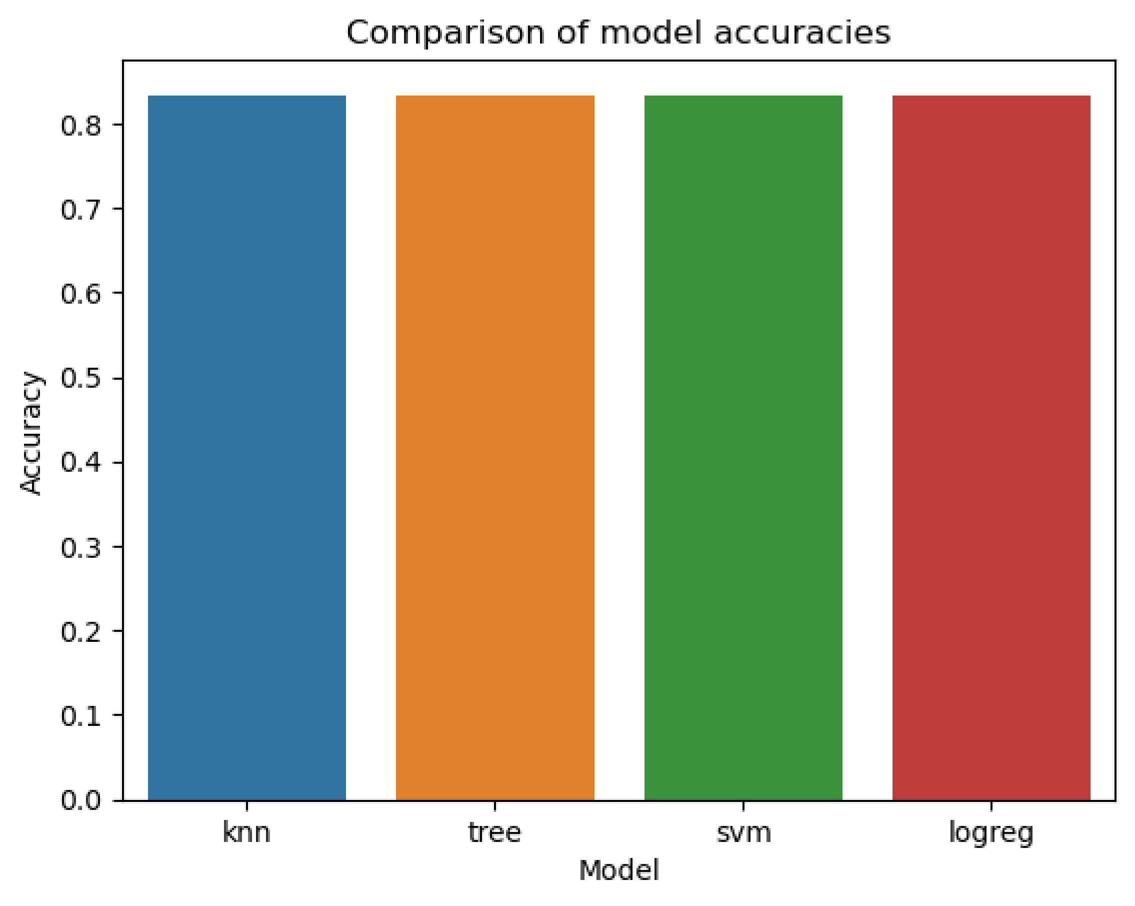


Section 5

Predictive Analysis (Classification)

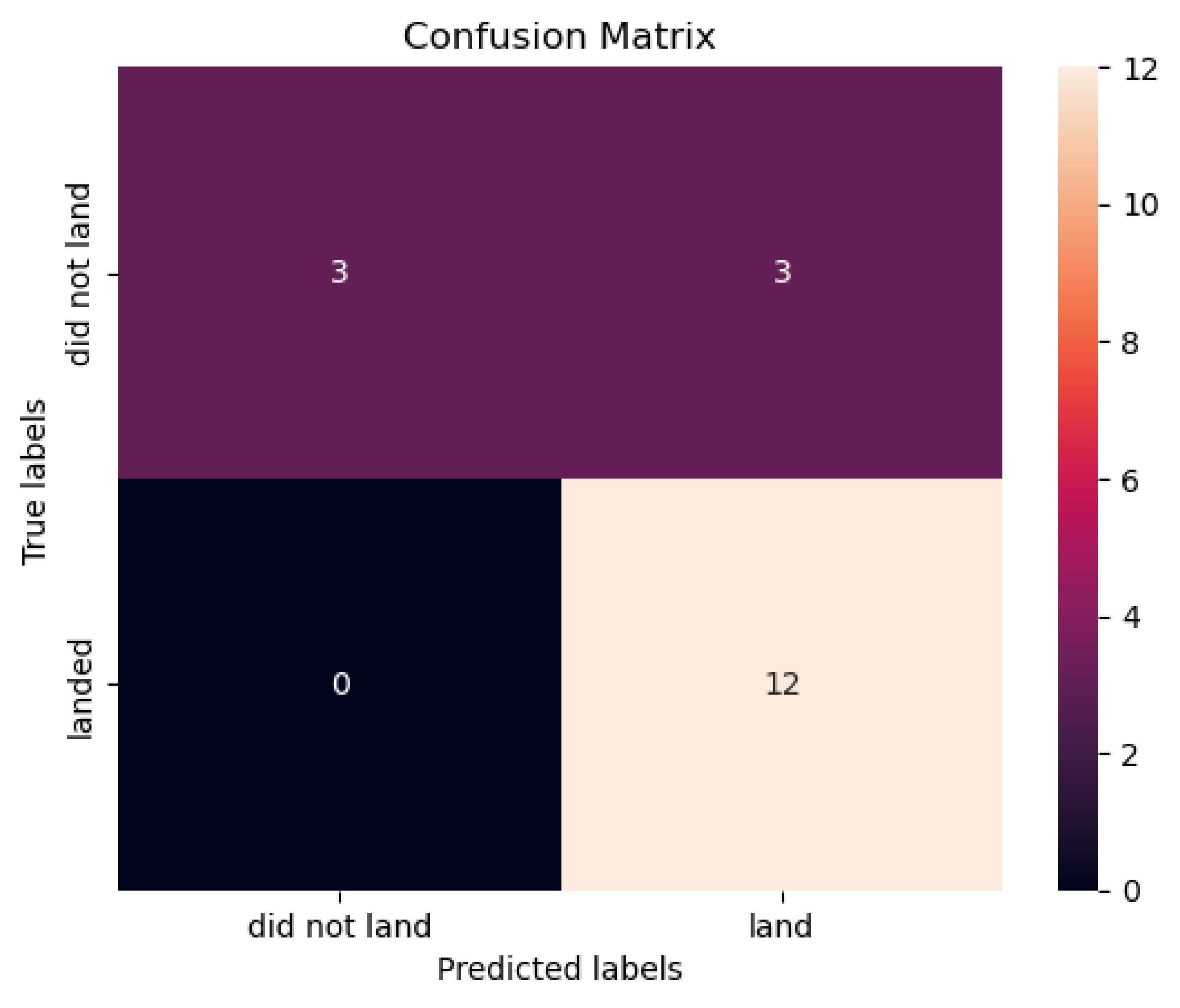
Classification Accuracy

- All models seem to have the same accuracy being at 0.833.



Confusion Matrix

- The best performing model is very good at calculating true positives.
- Only 16% (3 in total) were falsely classified.
- This explains the accuracy of 0.8333.



Conclusions

- Even with the very limited dataset a good performing machine learning model could be built
- The remaining uncertainty can be attributed to the multitude of reasons for why a launch may be a failure or a success; The dataset is still missing some information about mechanical components of the rocket.
- With the help of Data Science, measures can be taken that improve success probability in the future

Thank you!