IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yair Tenorio
06 Nov 2022

https://github.com/YiroTen/IBM_Capstone

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection SpaceX API
  - Data collection web scraping
  - Data Wrangling
  - EDA with visualization
  - EDA with SQL
  - Interactive map with Folium
  - Plotly Dash interactive HTML charts
  - Predictive Analysis with LR, SVM, DT and KNN

Executive Summary

- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context
    - In this project we will obtain and analyze the data from landings of Space X, to gain insight of the success rate and type of rockets that positively landed, also informatios about payload, places, and dates of the launches are of relevance, We want better understand what is happening

- Problems you want to find answers
    - The Falcon 9 will land successfully, recovering the first stage represents cost savings
    - Can we re-use the first stage?
    - Which places and booster versions have the higest landing rate
    - What is the mean payload puted in orbit for each rocket
    - Which places have more launches
    - Which rockets are worst
    - How is the Landing success rate trough the years?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected from the Wik page of Space X through web scrapping, also with the API of SpaceX, the data from falcon 9 was extracted and converted into a pandas DF
- Perform data wrangling
    - With the information of the data set the landings was categorized in successful and failed landings
- 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, evaluate classification models

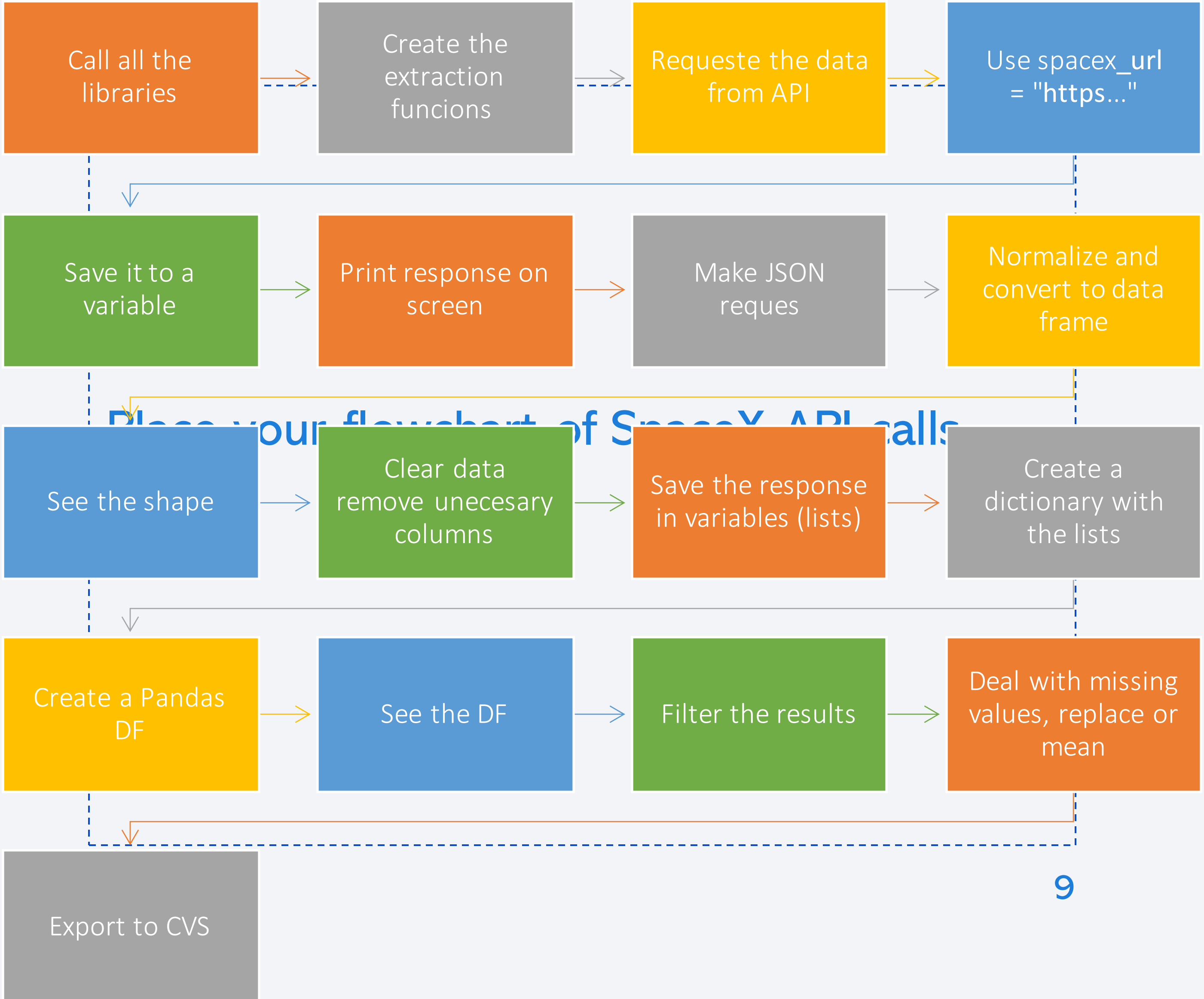
Data Collection

- Describe how data sets were collected.
- The data was collected from two sources, from the API of space X and from the wiki page trough web srapping, filtering, cleaning, and normalization of the data was performed
- You need to present your data collection process use key phrases and flowcharts
- Call all the libraries, Create the extraction funcions, Requeste the data from API, Use spacex_url = "https...", Save it to a variable, Print response on screen, Make JSON request, Normalize and convert to data frame
- See the shape, Clear data remove unecesary columns, Save the response in variables (lists), Create a dictionary with the lists, Create a Pandas DF, See the DF, Filter the results, Deal with missing values, replace or mean and finally Export to CVS

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/YiroTen/IBM_Capstone/blob/7e70a4ac071aae060f5554b45e323e1fa2b2a3f4/jupyter-labs-spacex-data-collection-api%20(1).ipynb), as an external reference and peer-review purpose



Call all the libraries

Create the extraction funcions

Requeste the data from API

Use spacex_url = "https..."

Save it to a variable

Print response on screen

Make JSON reques

Normalize and convert to data frame

Place your flowchart of SpaceX API calls

See the shape

Clear data remove unecesary columns

Save the response in variables (lists)

Create a dictionary with the lists

Create a Pandas DF

See the DF

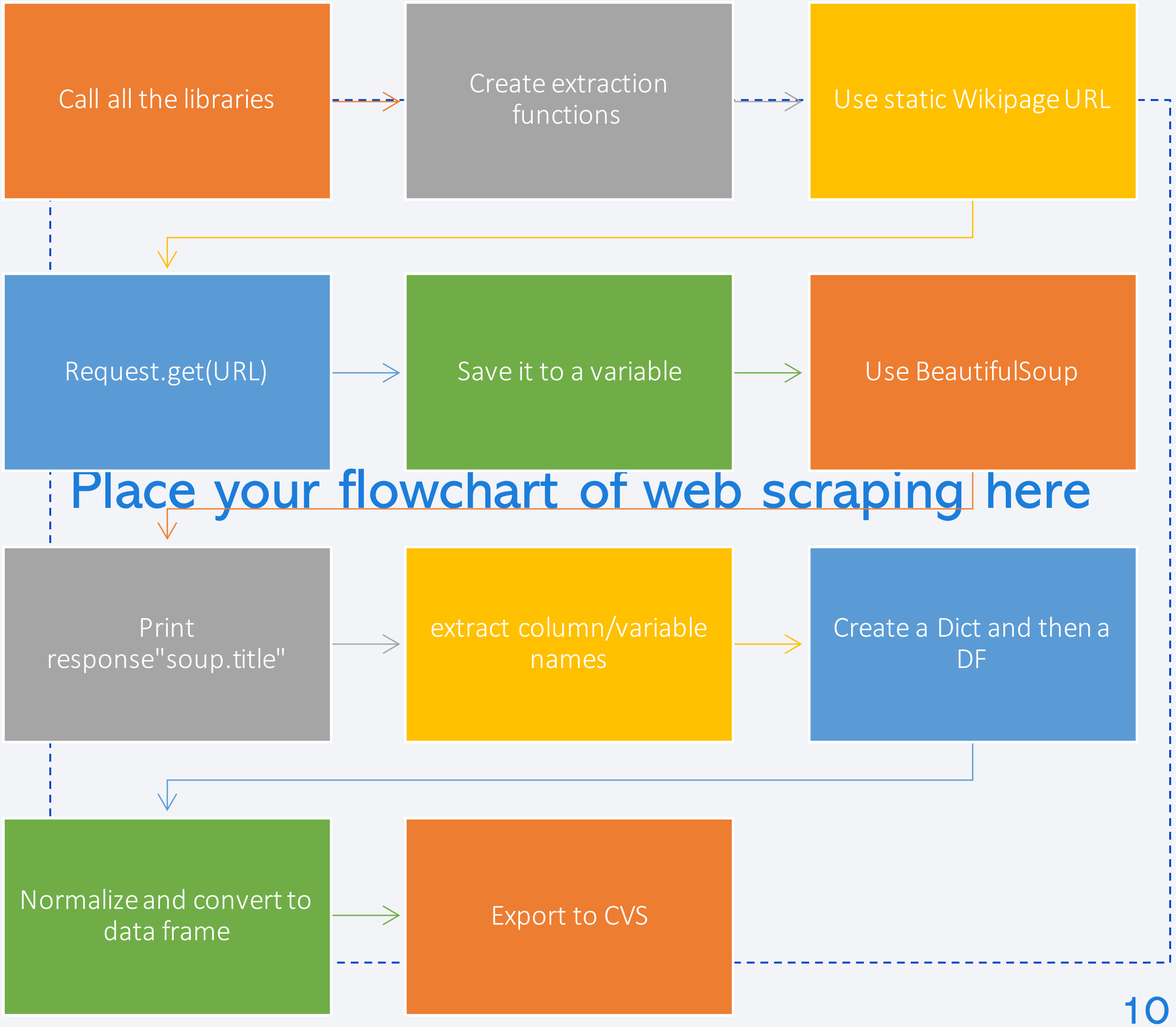
Filter the results

Deal with missing values, replace or mean

Export to CVS

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose. https://github.com/YiroTen/IBM_Capstone/blob/d0d503fe80df1ac757e0b3a01ac92d1ef41b9091/jupyter-labs-webscraping.ipynb



Call all the libraries

Create extraction functions

Use static Wikipage URL

Request.get(URL)

Save it to a variable

Use BeautifulSoup

Place your flowchart of web scraping here

Print response"soup.title"

extract column/variable names

Create a Dict and then a DF

Normalize and convert to data frame

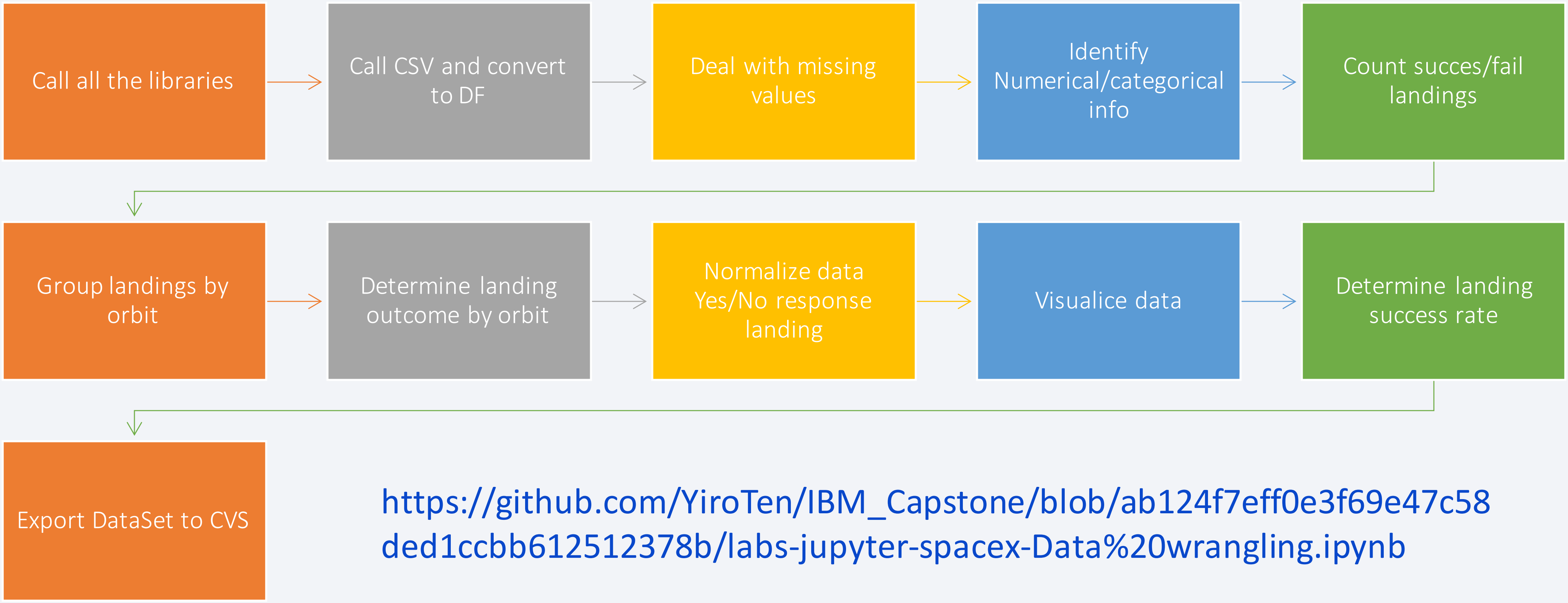
Export to CVS

Data Wrangling

- Describe how data were processed
  - We perform EDA to find some patters in the data an determine the variables to predict successful landing, firstly we import all the necessary libraries, and then we load the data into a data frame, the data was cleaned and "none values" was removed, numerical or categorical information was identified, total of launches for each site were calculated and separated depending on the Orbits
  - Also, information was categorized in successful or unsuccessful landings (Normalized) only YES/NO, 1 or 0. And succesful rate was calculated
  - The response was saved in CSV format

Data Wrangling

- You need to present your data wrangling process using key phrases and flowcharts



Call all the libraries
Call CSV and convert to DF
Deal with missing values
Identify Numerical/categorical info
Count succes/fail landings
Group landings by orbit
Determine landing outcome by orbit
Normalize data Yes/No response landing
Visualice data
Determine landing success rate
Export DataSet to CVS

https://github.com/YiroTen/IBM_Capstone/blob/ab124f7eff0e3f69e47c58ded1ccbb612512378b/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - **"Flight"** VS "**Payload Mass"** and the success/fail landing encoded to group the landings we observe an increase of success throug time
  - **"Flight number"** vs **"Launch site"**, some places have success landing rate's higher than others
  - Also **"Launch site"** vs **"Payload Mass"**, the higher payloads have higher success landing rate
  - Normalized **Bar Chart** of orbits shown that some orbits also had higher success rates
  - **"Flight number"** vs **"Orbit"**, we observe though time if the success is increasing
  - **"Payload mass"** vs **"Orbit"** shows a "**range**" of the min/max load destined to each orbit
  - **"Annual success rate"** vs **"year"** we observe an increase of success landings though the years.
  - https://github.com/YiroTen/IBM_Capstone/blob/a18481b8bf7f138535d3fa699e71180eadf4e80e/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

1. %sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;
2. %sql SELECT Launch_Site FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
3. %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL;
4. %sql SELECT avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version='F9 v1.1';
5. %sql SELECT LandingOutcome, Date from SPACEXTBL where LandingOutcome LIKE '%ground%' ;
6. %sql SELECT Booster_Version, LandingOutcome, PAYLOAD_MASS__KG_ from SPACEXTBL where LandingOutcome LIKE 'Success%drone%' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
7. %sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Total from SPACEXTBL GROUP BY Mission_Outcome;
8. %sql SELECT Distinct(Booster_Version), PAYLOAD_MASS__KG_ from SPACEXTBL WHERE PAYLOAD_MASS__KG_ > (SELECT AVG(PAYLOAD_MASS__KG_) FROM spacextbl) GROUP BY Booster_Version order by PAYLOAD_MASS__KG_ DESC;
9. %sql select LandingOutcome,Booster_Version,Launch_Site, substr(Date,1,2) as Day, substr(Date,4,2) as Month, substr(Date,7,4) as Year from spacextbl WHERE LandingOutcome LIKE 'Failure%drone%' AND Year='2015';
10. %sql select LandingOutcome,Booster_Version,Launch_Site, substr(Date,1,2) as Day, substr(Date,4,2) as Month, substr(Date,7,4) as Year from spacextbl WHERE LandingOutcome LIKE 'Success%' and Year<='2017' limit 8;

Add the GitHub URL of your completed EDA with SQL notebook,

https://github.com/YiroTen/IBM_Capstone/blob/08ac816bffdd7b17e3abbad9814b65c6b9a63968/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- We add markers, circles, and lines to observe the number of success/fail landings per place and watch it better, also circles help us to easy localize the places of interest in the globe, lines with
- Distances to the nearest places of interest was calculated, such as railways, cities and highways, also marked with a line on the map for easy identification
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- https://github.com/YiroTen/IBM_Capstone/blob/5e2f7c98b40967028310adaf709747aeec4e851e/3_1_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

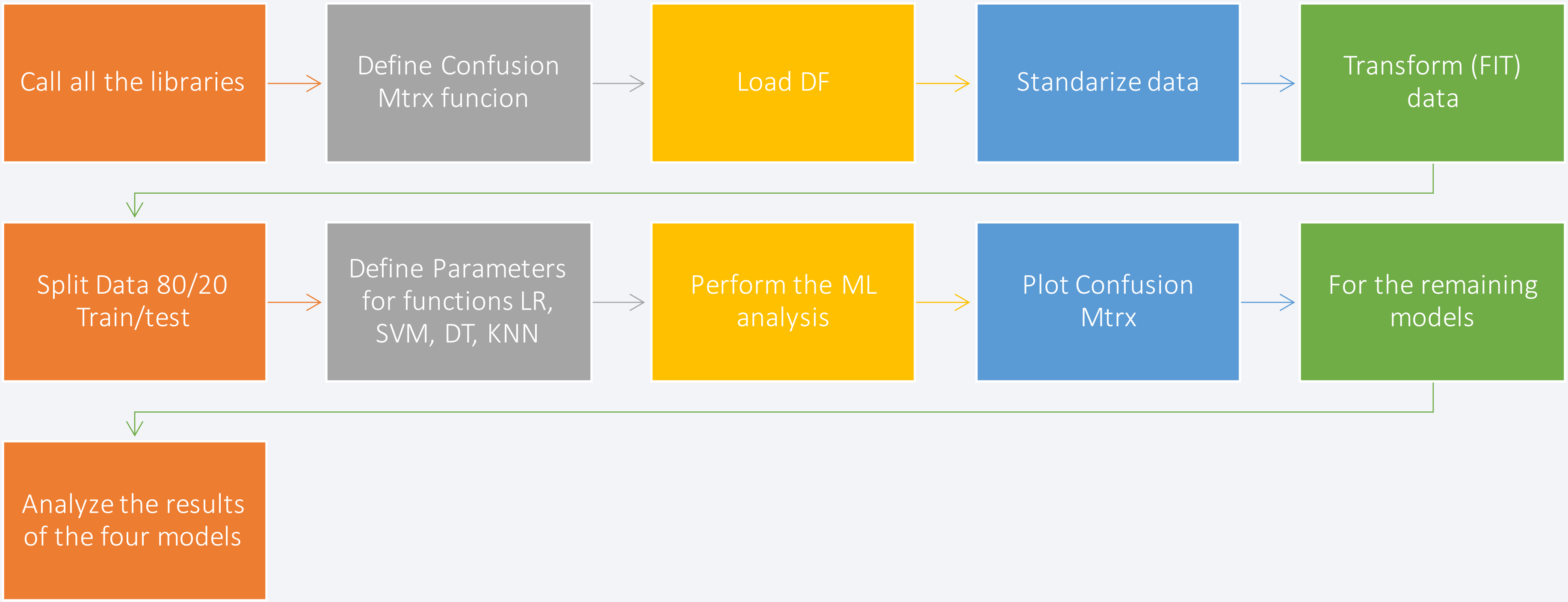
- Summarize what plots/graphs and interactions you have added to a dashboard
- success count for all sites, in a piechart
- piechart for the launch site with highest launch success/fail ratio in %
- Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain why you added those plots and interactions
- These interactions where added because they represent the strongest correlation between variables of interest such as landing site, booster version, and payload, we are analyzing them
- https://github.com/YiroTen/IBM_Capstone/blob/9ce4f4104b52020871826a8515ab75117e95bd15/spacex_dash_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- The several models was builded using the succes/fail landing spliting the data in a train / test ratio of 80/20, four models was used, logistic regression, support vector machines, Decision tree and KNN,
- tuned hpyerparameters :(best parameters) {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'} **accuracy : 0.8464285714285713** logistic regression
- tuned hpyerparameters :(best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'} **accuracy : 0.8482142857142856** SVM
- tuned hpyerparameters :(best parameters) {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'best'} **accuracy : 0.8910714285714285** Deccisson tree
- tuned hpyerparameters :(best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1} **accuracy : 0.8482142857142858** KNN

Predictive Analysis (Classification)

- You need present your model development process using key phrases and flowchart


Call all the libraries

Define Confusion Mtrx funcion

Load DF

Standarize data

Transform (FIT) data

Split Data 80/20 Train/test

Define Parameters for functions LR, SVM, DT, KNN

Perform the ML analysis

Plot Confusion Mtrx

For the remaining models

Analyze the results of the four models

- https://github.com/YiroTen/IBM_Capstone/blob/4de67f9f438b8984433fa20d7808f8c3a6463850/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

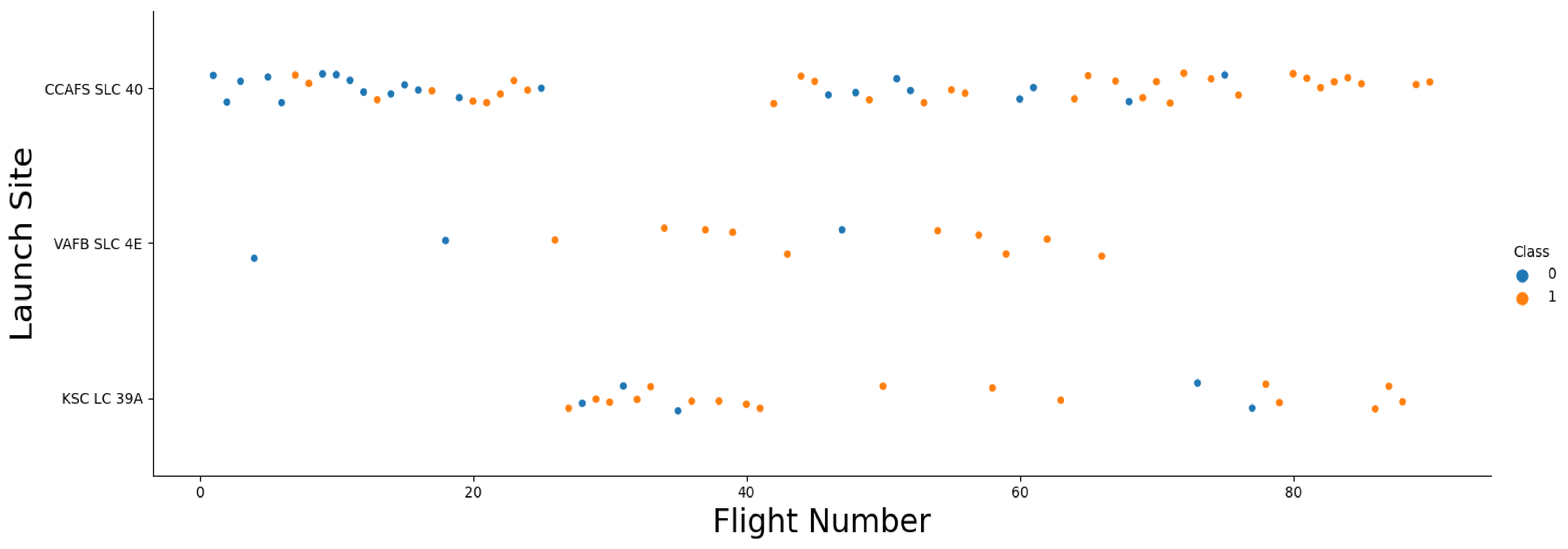
Results

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

Section 2

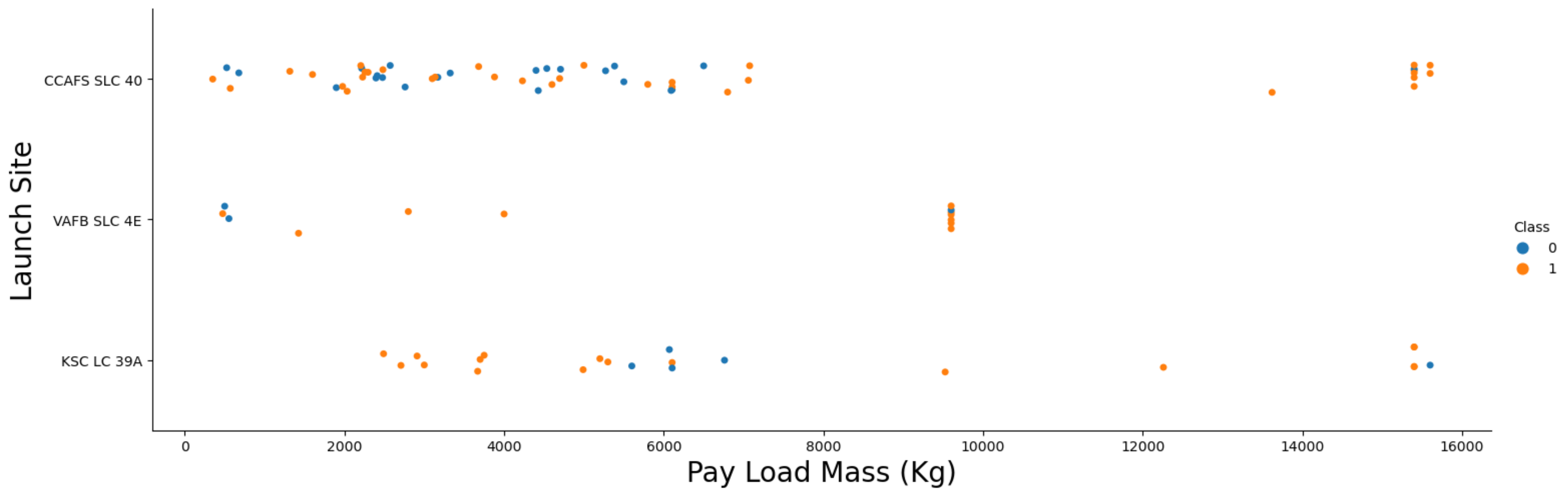
Insights drawn from EDA

Flight Number vs. Launch Site



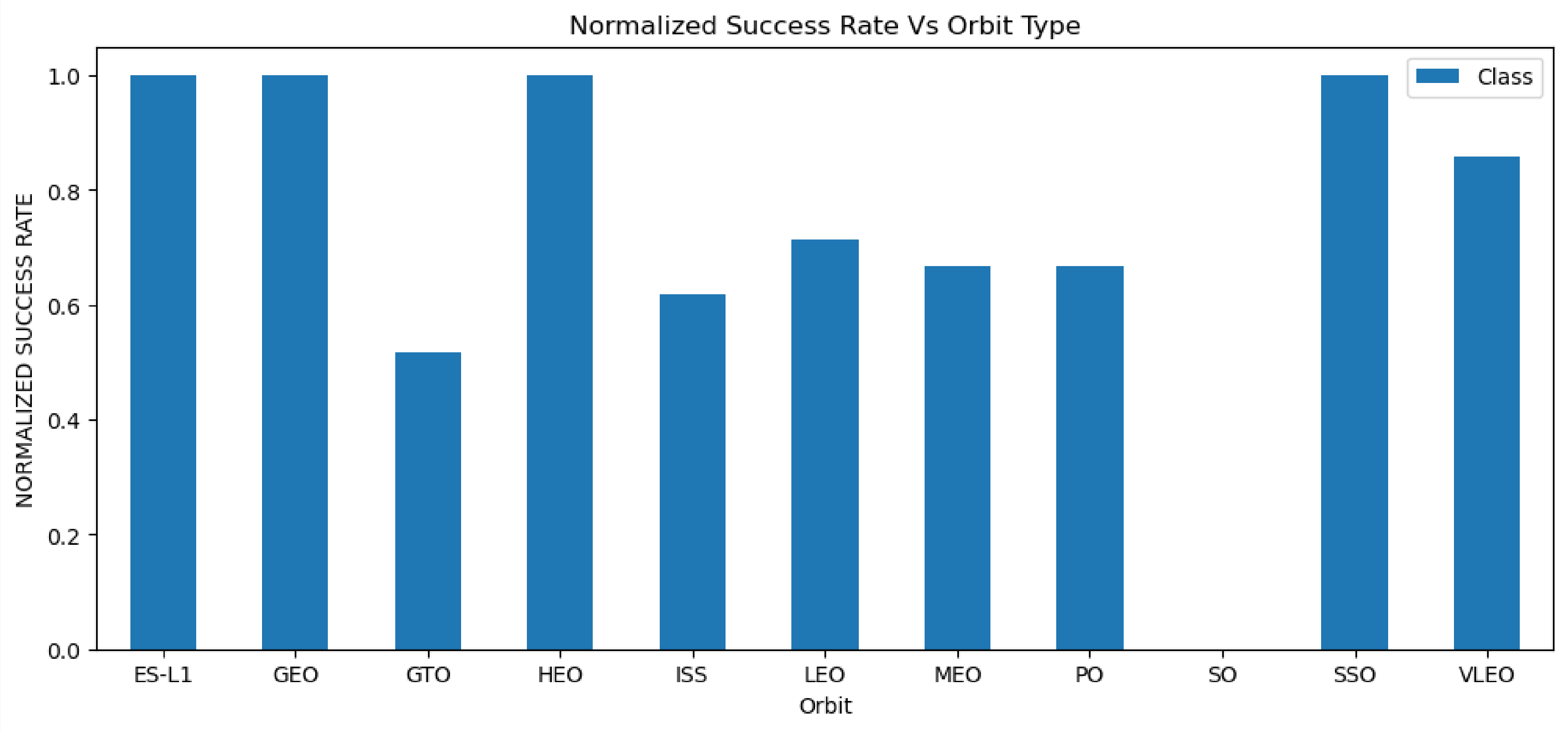
- some places have success landing rate's higher than others

Payload vs. Launch Site



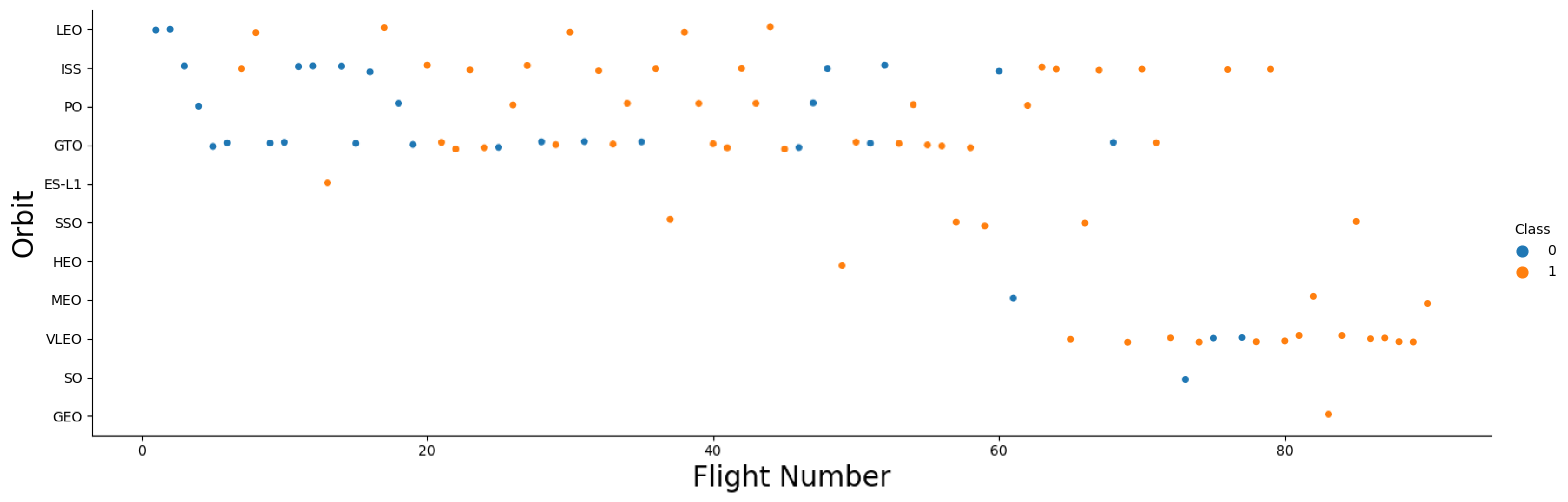
- The higher payloads have higher success landing rate

Success Rate vs. Orbit Type



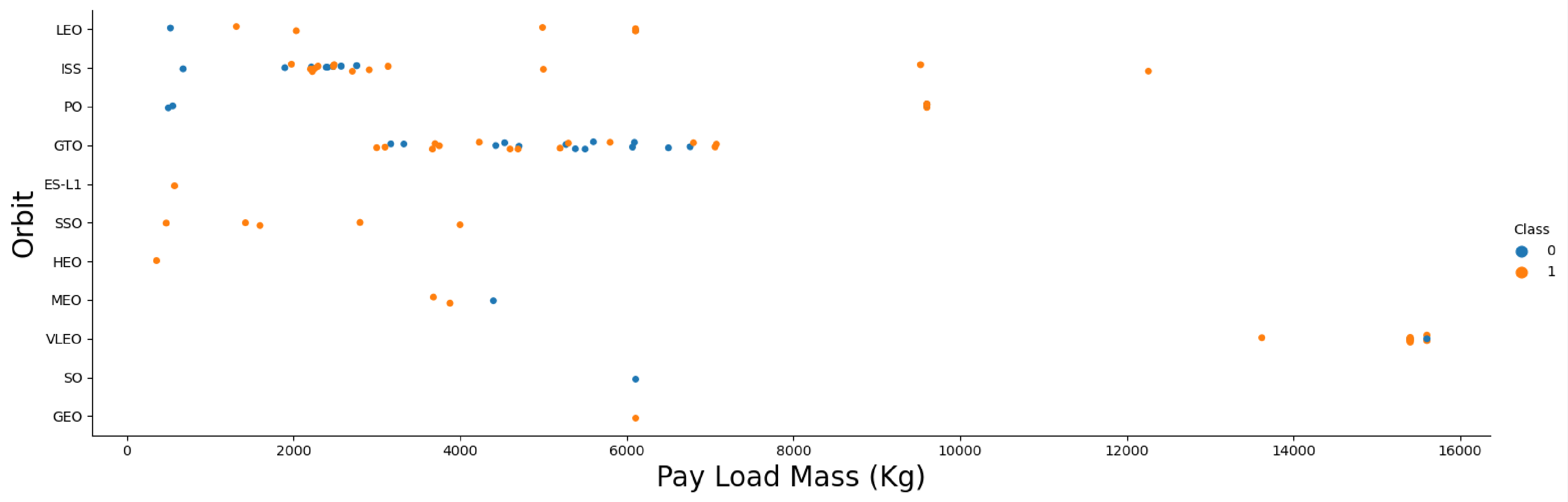
- orbits shown that some orbits also had higher success rates

Flight Number vs. Orbit Type



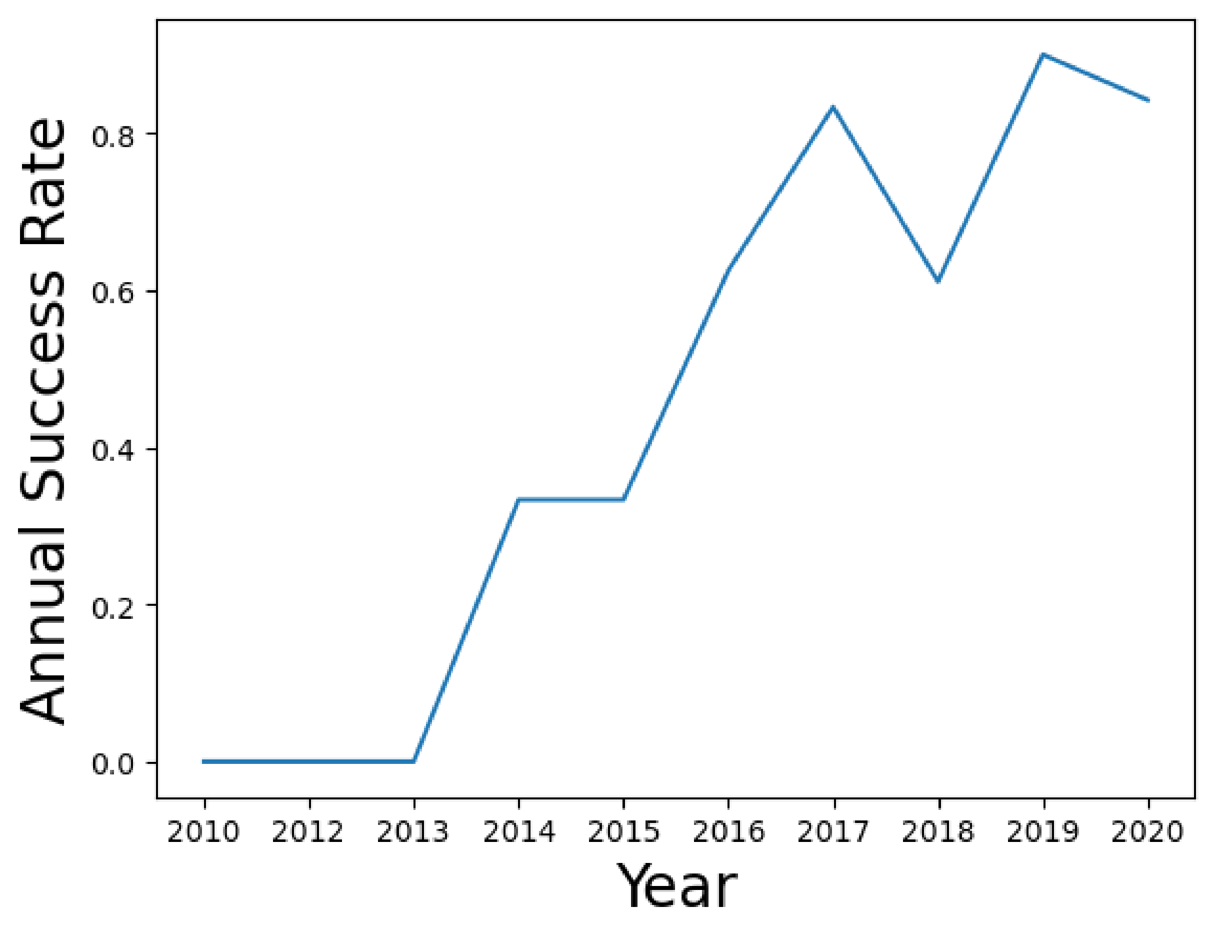
we observe though time if the success is increasing depending on the orbit

Payload vs. Orbit Type



- Some orbits have ranges of weight or payload mass,
- higher payloads have higher success rates

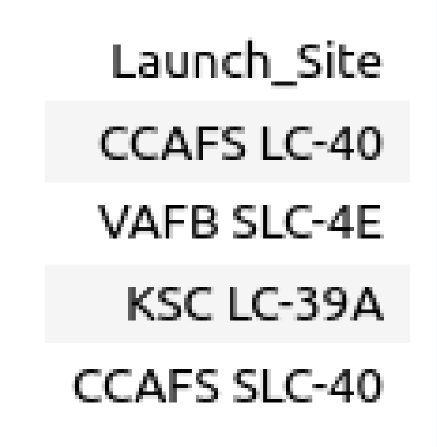
Launch Success Yearly Trend



- The Overall landing success rate has been increased over the years

All Launch Site Names

- Find the names of the unique launch sites

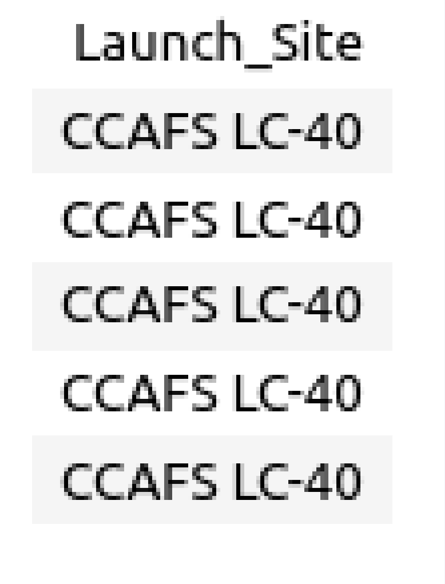


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- Present your query result with a short explanation here
- From the DB we extract unique values from column "Launch_Site"
- %sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

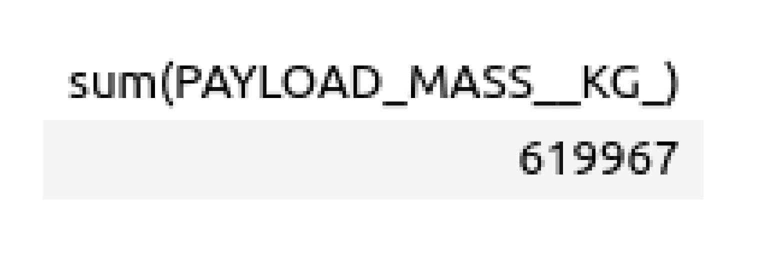


Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- Present your query result with a short explanation here
- If we want to perform filtering to obtain only some specific values, we use the next query
- %sql SELECT Launch_Site FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



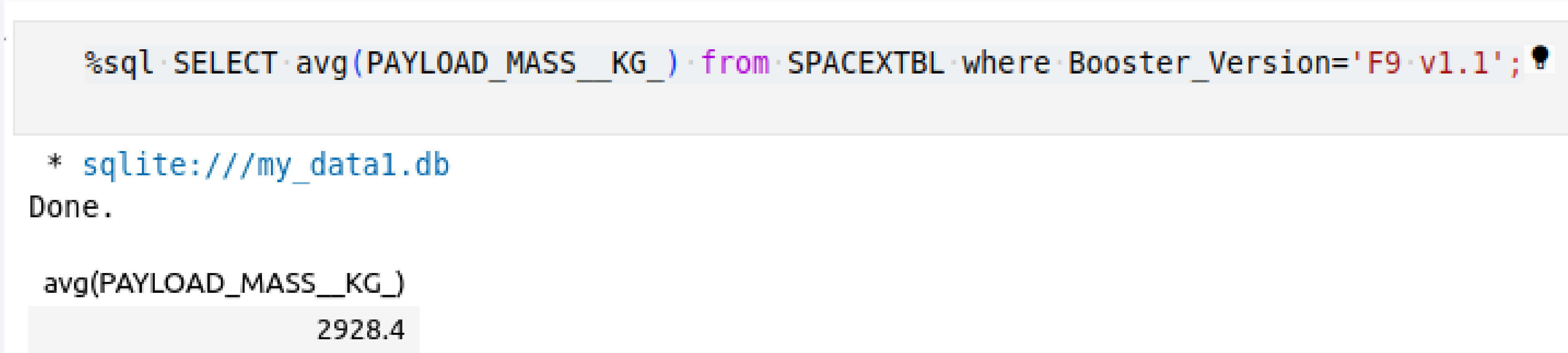
```
sum(PAYLOAD_MASS__KG_)
                619967
```

- Present your query result with a short explanation here, we performed o sum over the column payload mass, the total of "Payload put it on orbit" from all the lauches
- %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL;

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here,

- The average payload mass launched by falcon 9 V1.1, is obtained with the next query, this could give us insight of the cost of a launch per booster



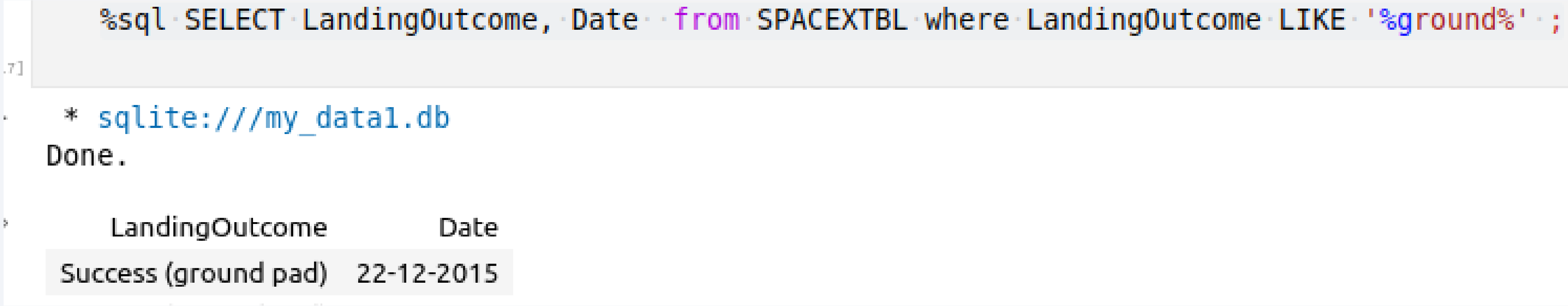
```
%sql SELECT avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version='F9 v1.1';
```

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

 avg(PAYLOAD_MASS__KG_)
                  2928.4
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here
    - The first success landig on ground pad was on Dec- 22 -2015,
        - ¡ it was Tuesday !



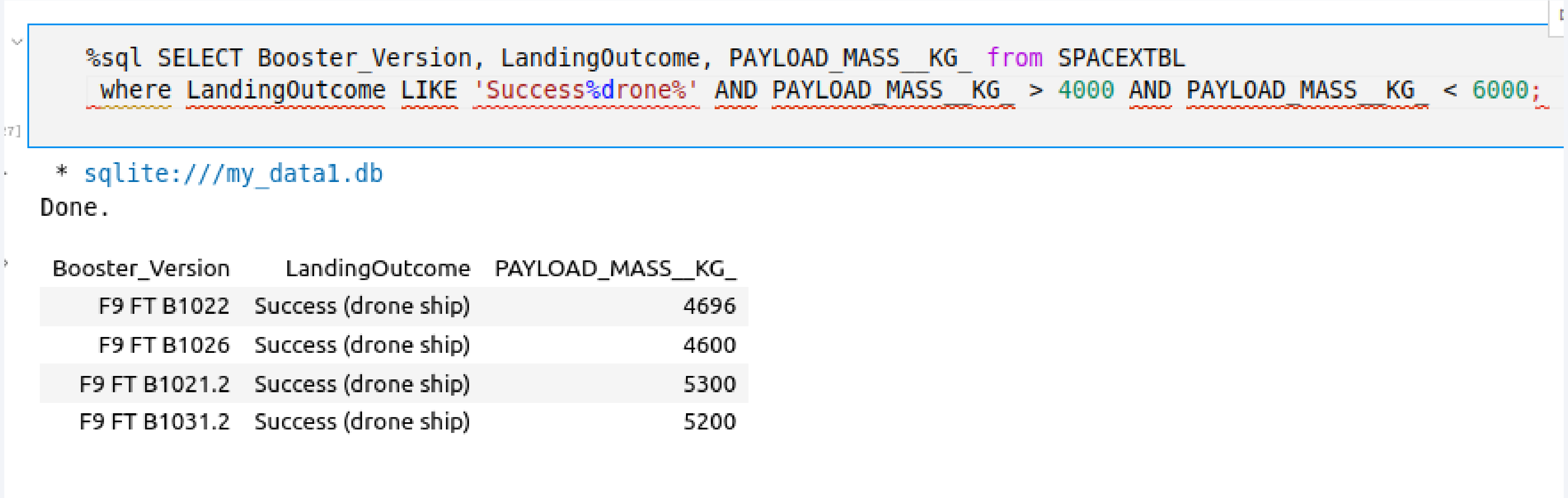
```
%sql SELECT LandingOutcome, Date  from SPACEXTBL where LandingOutcome LIKE '%ground%' ;
```

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

LandingOutcome	Date
Success (ground pad)	22-12-2015

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
- Present your query result with a short explanation here, if we want to limit the results between some ranges or get insight some payloads success, this query is very useful



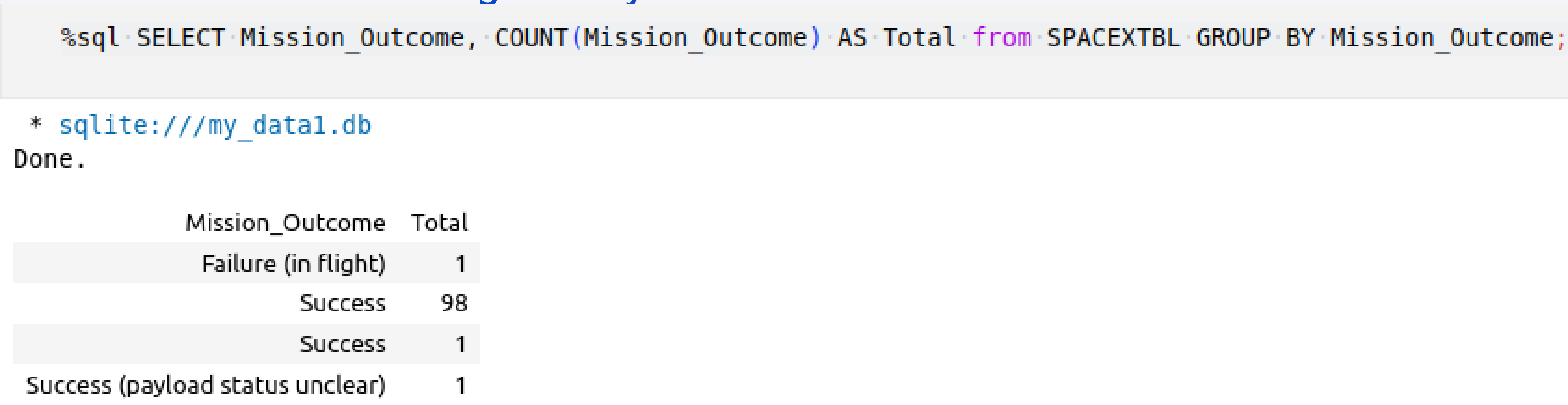
```
%sql SELECT Booster_Version, LandingOutcome, PAYLOAD_MASS__KG_ from SPACEXTBL
 where LandingOutcome LIKE 'Success%drone%' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here,
- We observe from the data that Falcon9 has a very high success rate,
- From a total of 101 flights only one failed and one is unclear.

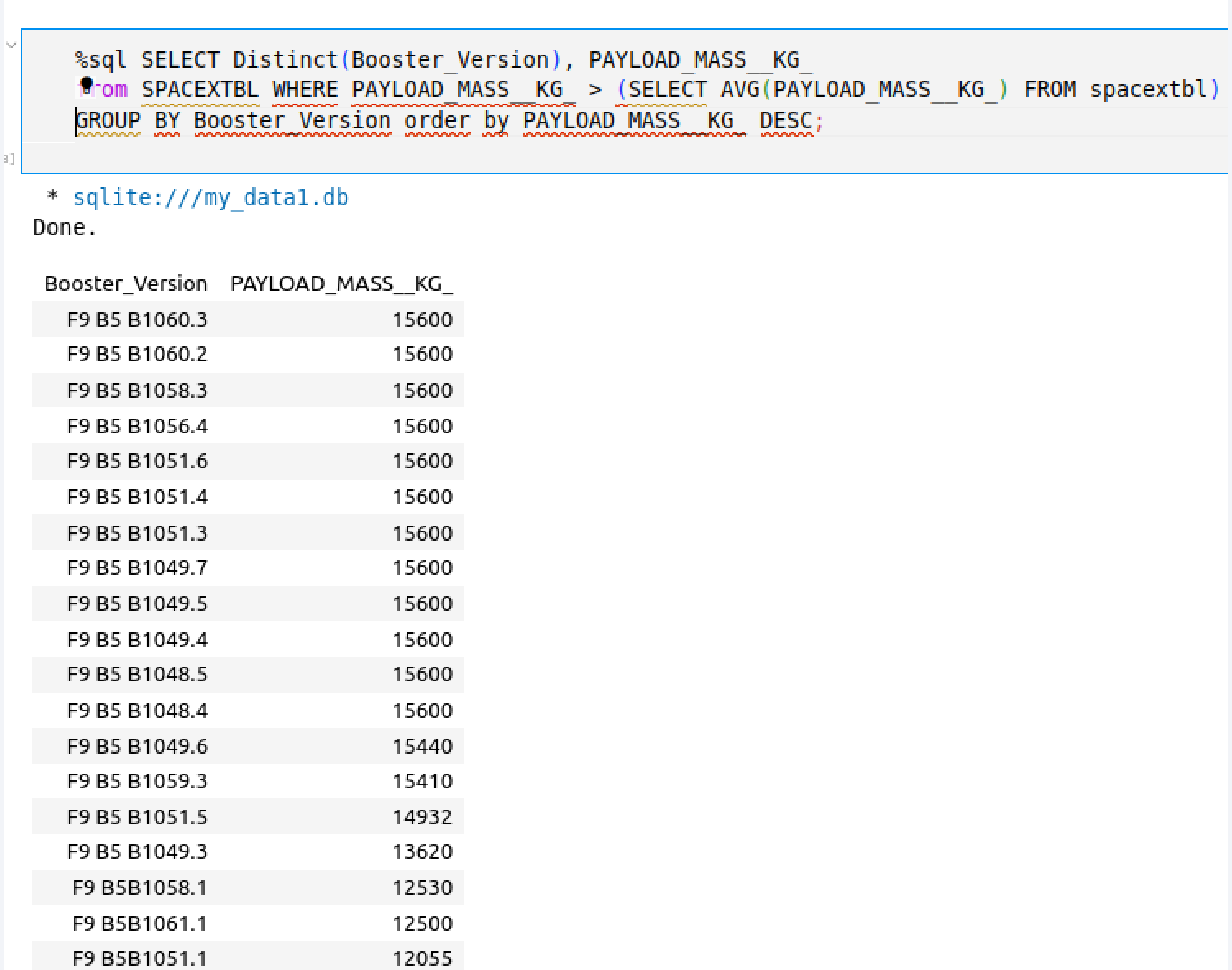


```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Total from SPACEXTBL GROUP BY Mission_Outcome;
```

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

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload



```
%sql SELECT Distinct(Booster_Version), PAYLOAD_MASS__KG_
from SPACEXTBL WHERE PAYLOAD_MASS__KG_ > (SELECT AVG(PAYLOAD_MASS__KG_) FROM spacextbl)
GROUP BY Booster_Version order by PAYLOAD_MASS__KG_ DESC;
```

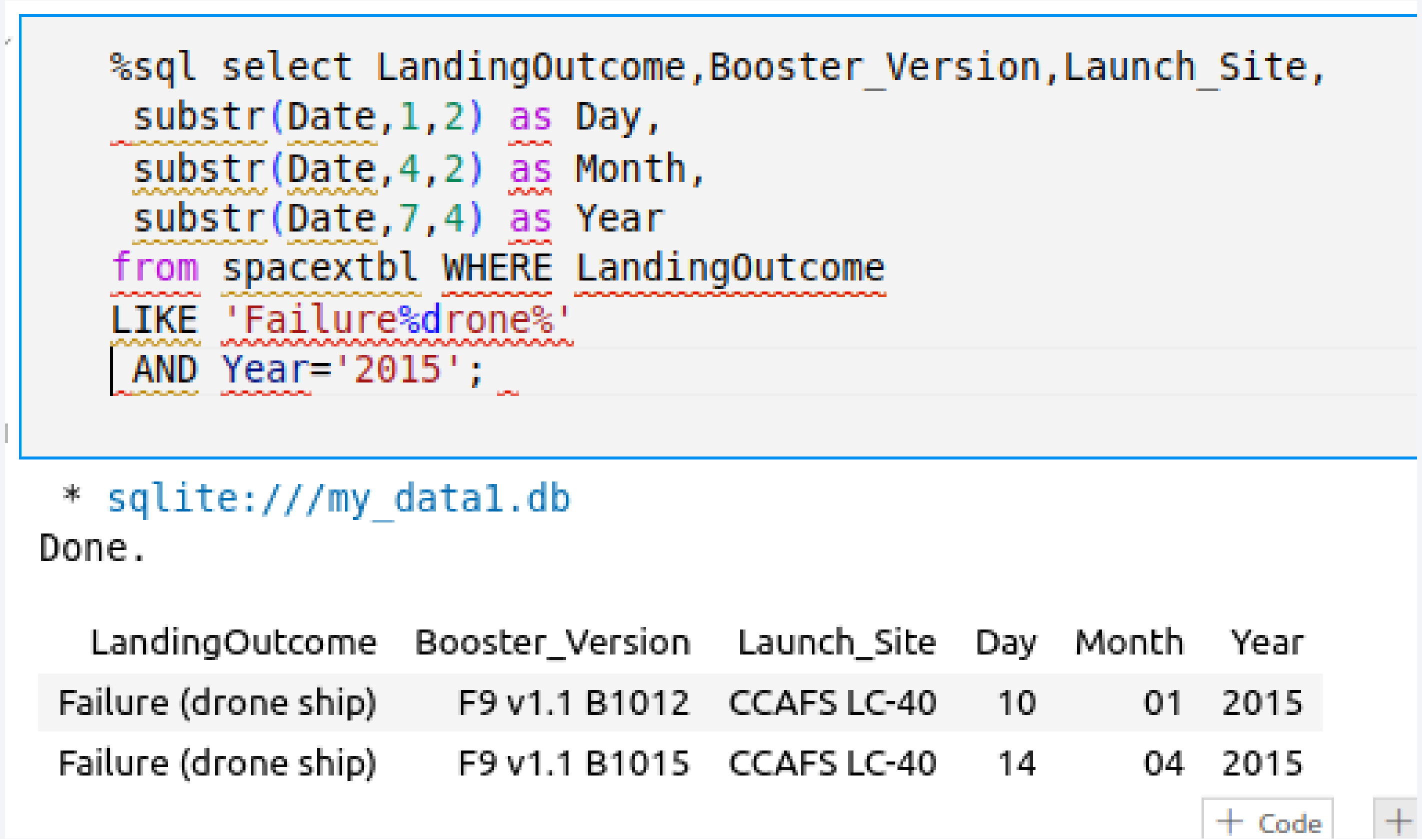
```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1060.3	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1056.4	15600
F9 B5 B1051.6	15600
F9 B5 B1051.4	15600
F9 B5 B1051.3	15600
F9 B5 B1049.7	15600
F9 B5 B1049.5	15600
F9 B5 B1049.4	15600
F9 B5 B1048.5	15600
F9 B5 B1048.4	15600
F9 B5 B1049.6	15440
F9 B5 B1059.3	15410
F9 B5 B1051.5	14932
F9 B5 B1049.3	13620
F9 B5B1058.1	12530
F9 B5B1061.1	12500
F9 B5B1051.1	12055

- F9-B5 an F9-B4 and F9-FT are the heavy-duty guys, the charge high loads
- From 6000 to 15 600 Kg

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here



```
%sql select LandingOutcome,Booster_Version,Launch_Site,
 substr(Date,1,2) as Day,
 substr(Date,4,2) as Month,
 substr(Date,7,4) as Year
from spacextbl WHERE LandingOutcome
LIKE 'Failure%drone%'
 AND Year='2015';
```

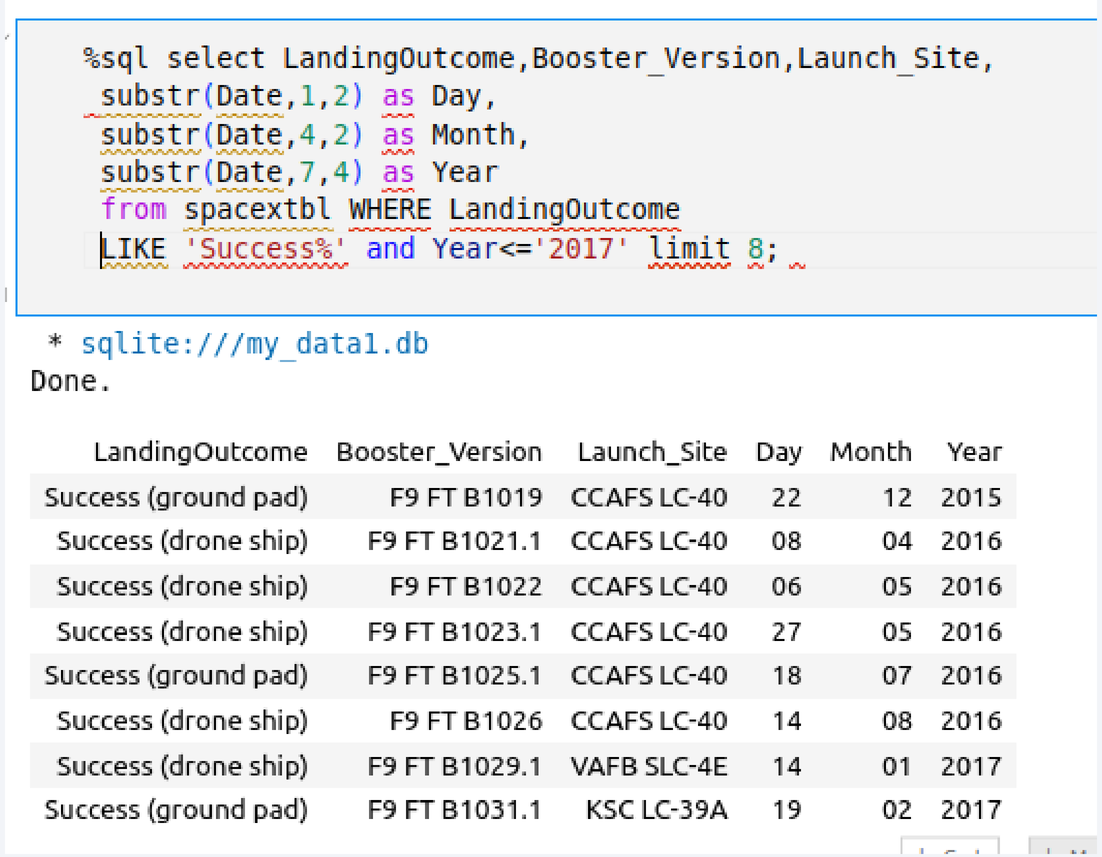
```
 * sqlite:///my_data1.db
Done.
```

LandingOutcome	Booster_Version	Launch_Site	Day	Month	Year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10	01	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	14	04	2015

To obtain the date we extract the information in a string maner and then the landing outcome from the booster version, and we observe wich ones failed, and when.

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

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



```
%sql select LandingOutcome,Booster_Version,Launch_Site,
 substr(Date,1,2) as Day,
 substr(Date,4,2) as Month,
 substr(Date,7,4) as Year
 from spacextbl WHERE LandingOutcome
 LIKE 'Success%' and Year<='2017' limit 8;
```

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

LandingOutcome	Booster_Version	Launch_Site	Day	Month	Year
Success (ground pad)	F9 FT B1019	CCAFS LC-40	22	12	2015
Success (drone ship)	F9 FT B1021.1	CCAFS LC-40	08	04	2016
Success (drone ship)	F9 FT B1022	CCAFS LC-40	06	05	2016
Success (drone ship)	F9 FT B1023.1	CCAFS LC-40	27	05	2016
Success (ground pad)	F9 FT B1025.1	CCAFS LC-40	18	07	2016
Success (drone ship)	F9 FT B1026	CCAFS LC-40	14	08	2016
Success (drone ship)	F9 FT B1029.1	VAFB SLC-4E	14	01	2017
Success (ground pad)	F9 FT B1031.1	KSC LC-39A	19	02	2017

If we want the success/fail outcomes in a specific period of time, we can observe it with the query provided, here.

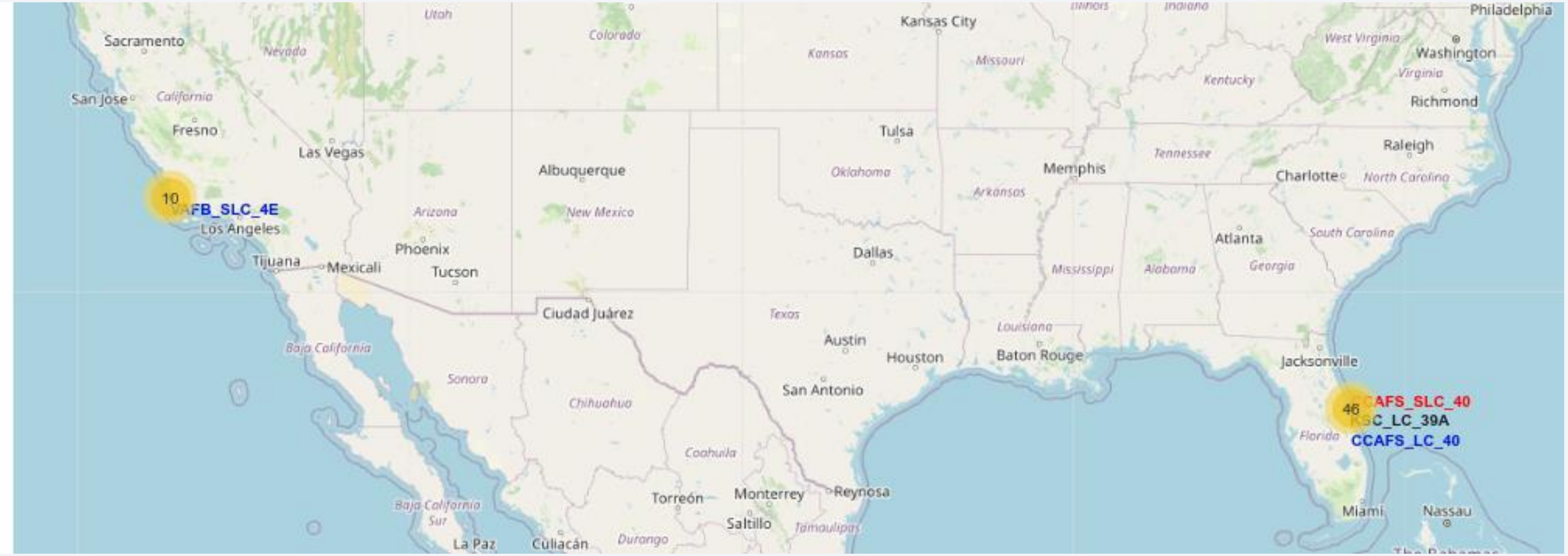
Maybe some events are important between those dates.

Section 3

Launch Sites Proximities Analysis

All Launch sites and places

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

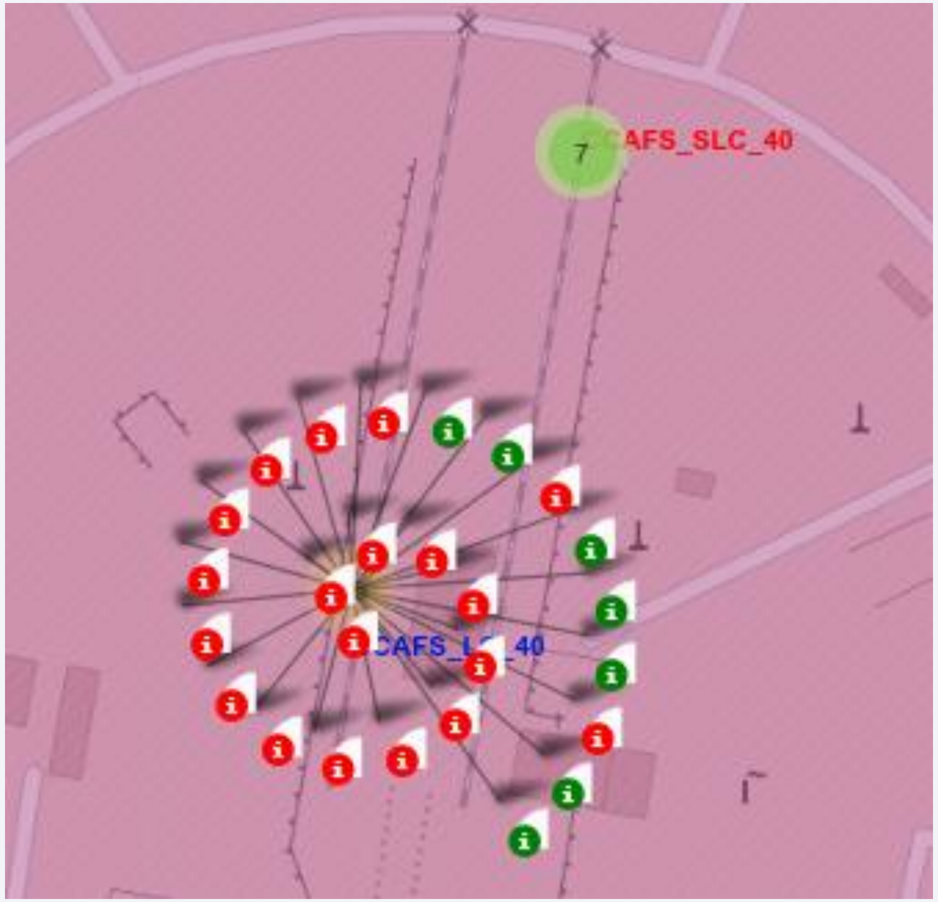


- We observe more launches from the East side than the West side,

Launches divided by place

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

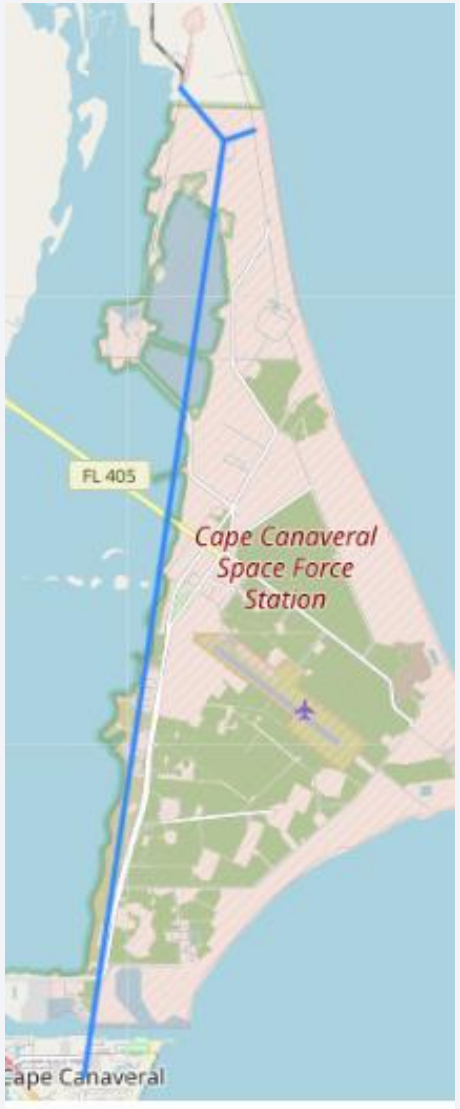
Vandenberg State Marine Reserve
10
FB_SLC_4E



- In the east with more landings, we can observe with green the success and red the fail landings

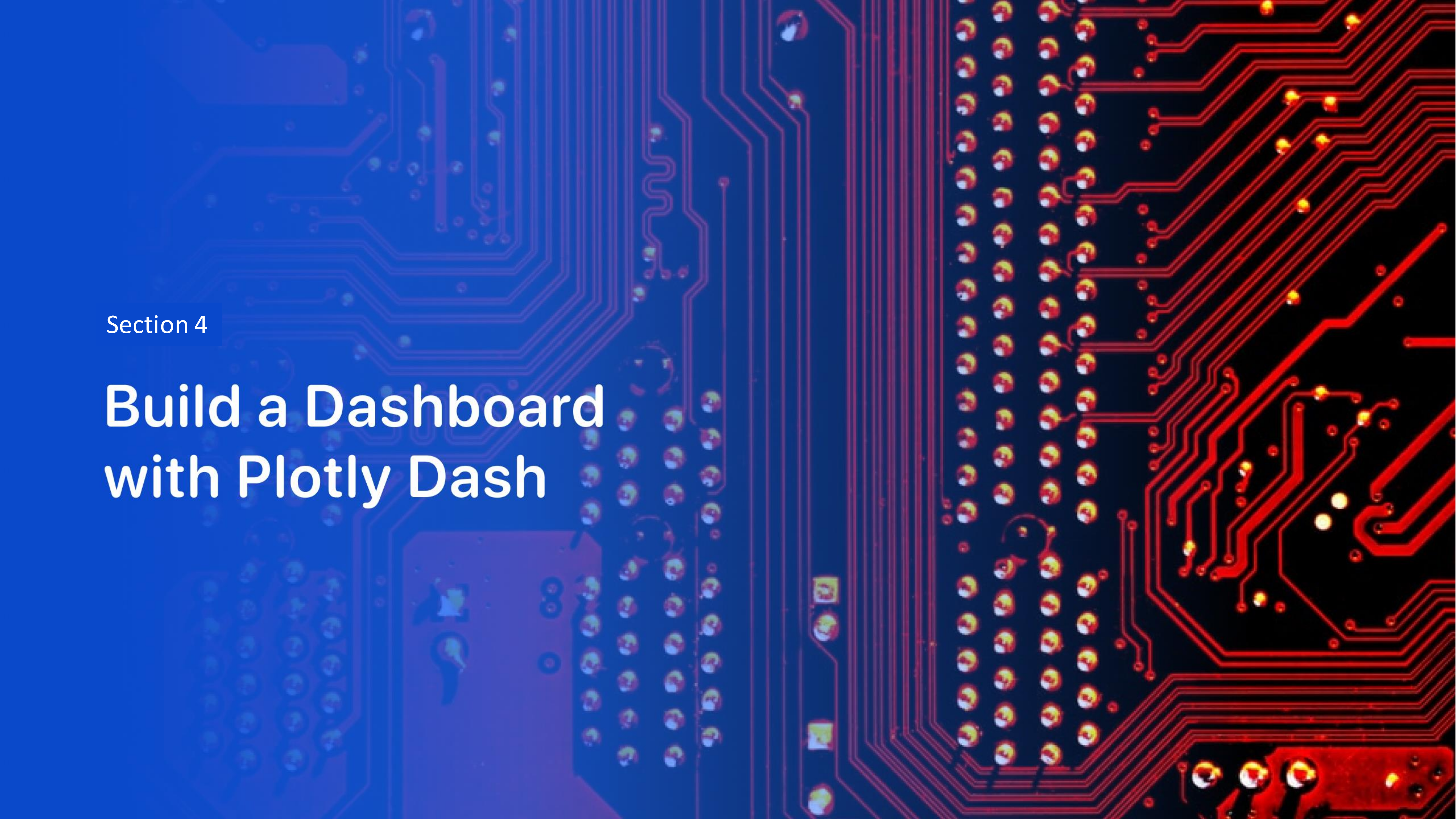
Distance to important sites from launching site

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- The distances to
- railway = 1.29 Km , city = 18.12 Km and highway = 0.59 Km
- https://github.com/YiroTen/IBM_Capstone/blob/5e2f7c98b40967028310adaf709747aeec4e851e/3_1_lab_jupyter_launch_site_location.ipynb

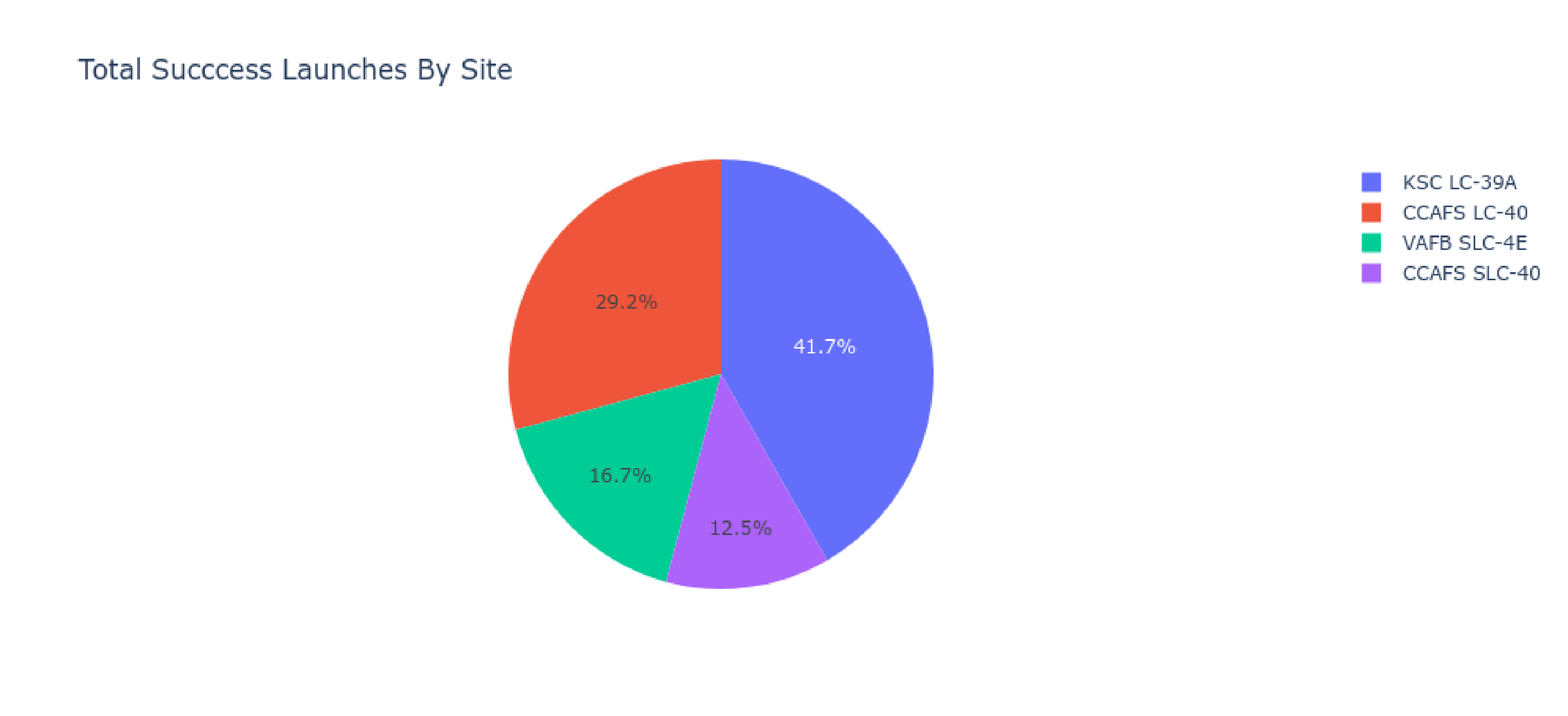


Section 4

Build a Dashboard with Plotly Dash

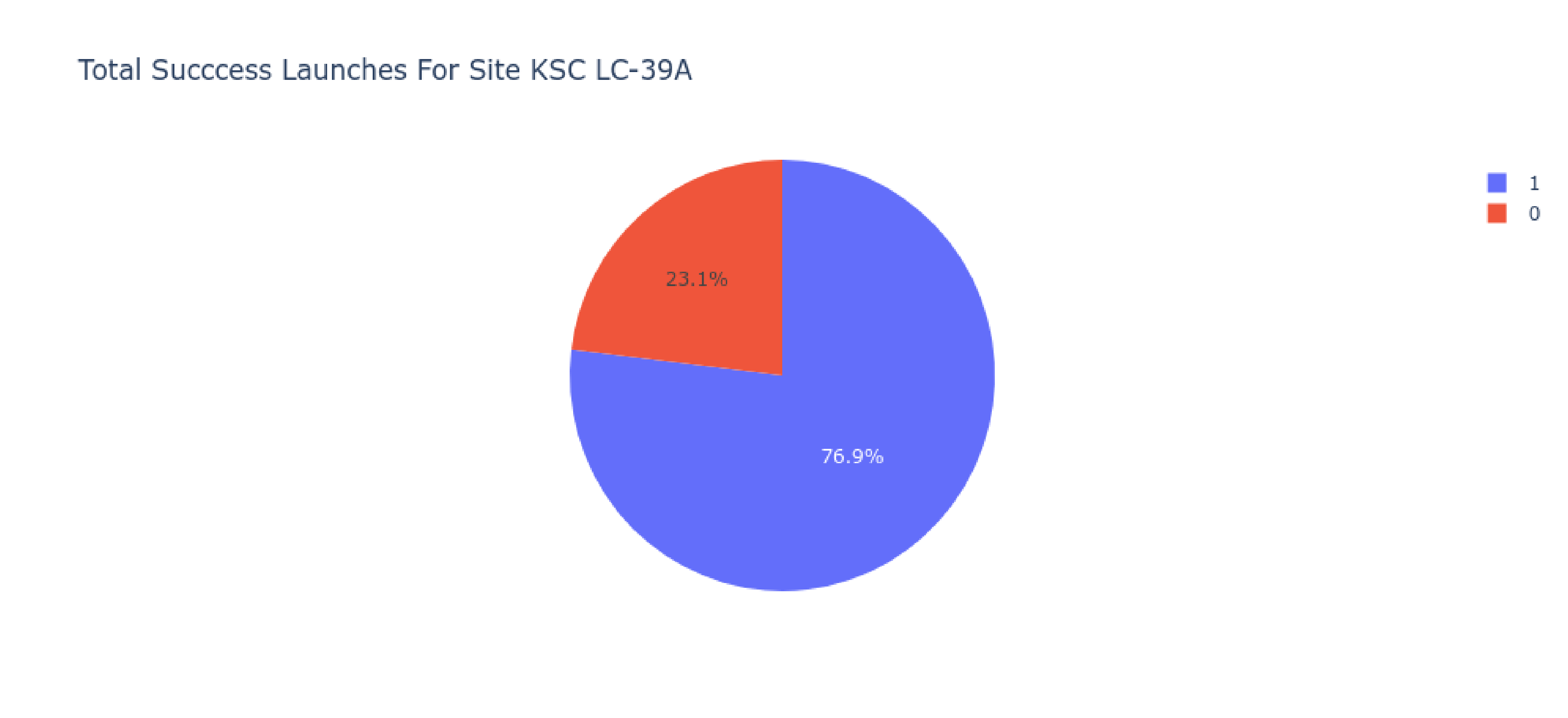


Launch success for all sites



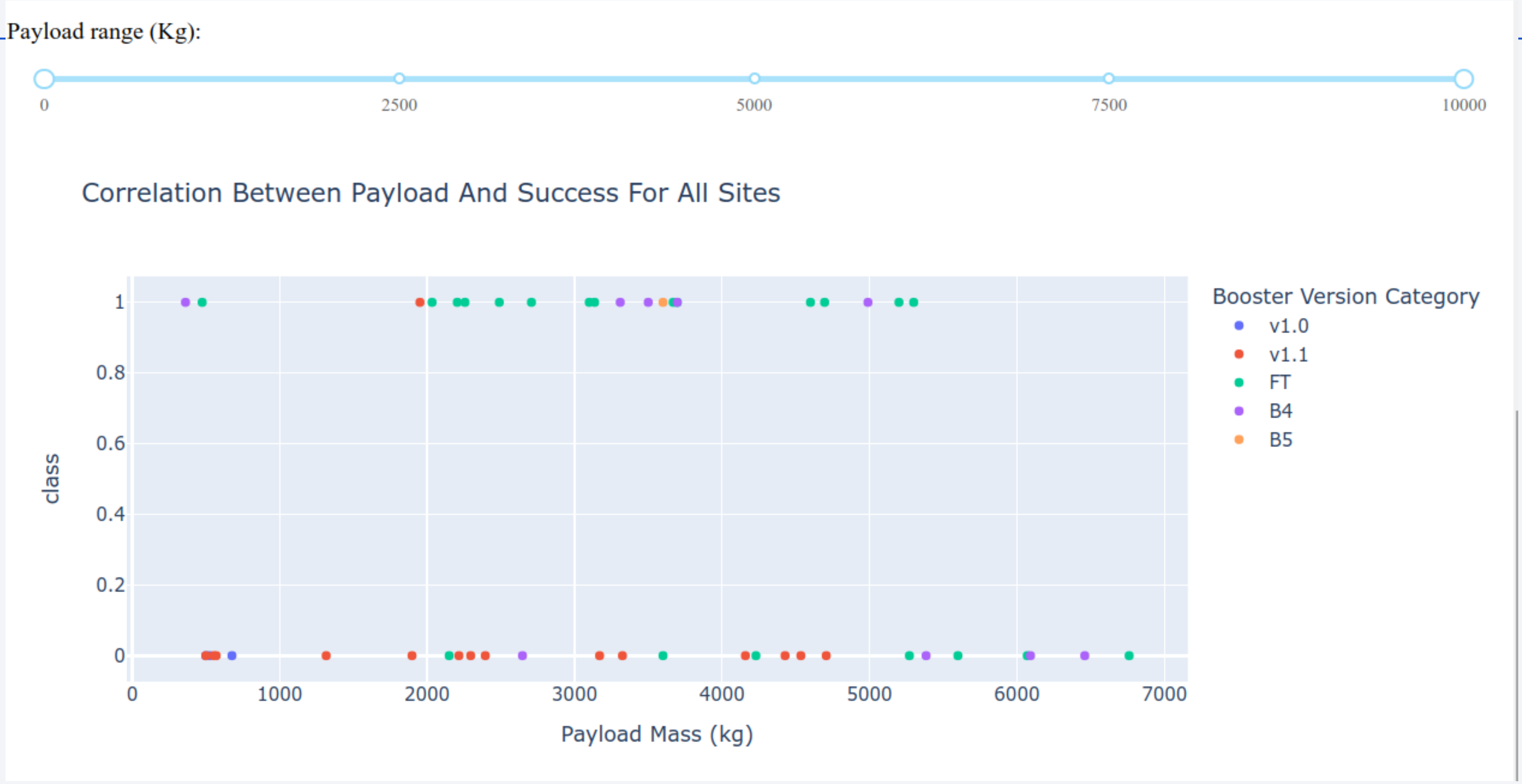
- We observe the total contribution of succesful launches by site

Place with the high success launch rate



- in the site KSC LC 39A the launches have a high landing success rate with 76.9 %

Payload vs Launch Outcome



Payload range (Kg):

- The payload range from 2500 to 7000 from KSC-LC-39A booster version FT has the largest success rate,
- https://github.com/YiroTen/IBM_Capstone/blob/9ce4f4104b52020871826a8515ab75117e95bd15/spacex_dash_app.py

Section 5

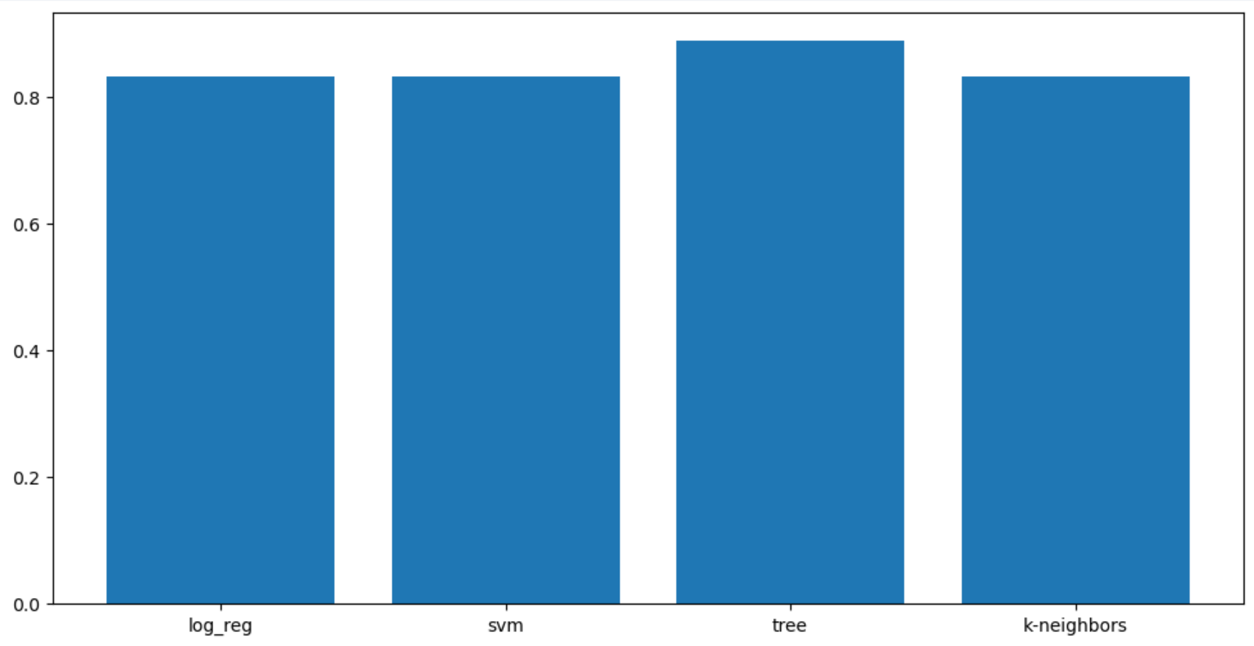
Predictive Analysis (Classification)

Classification Accuracy

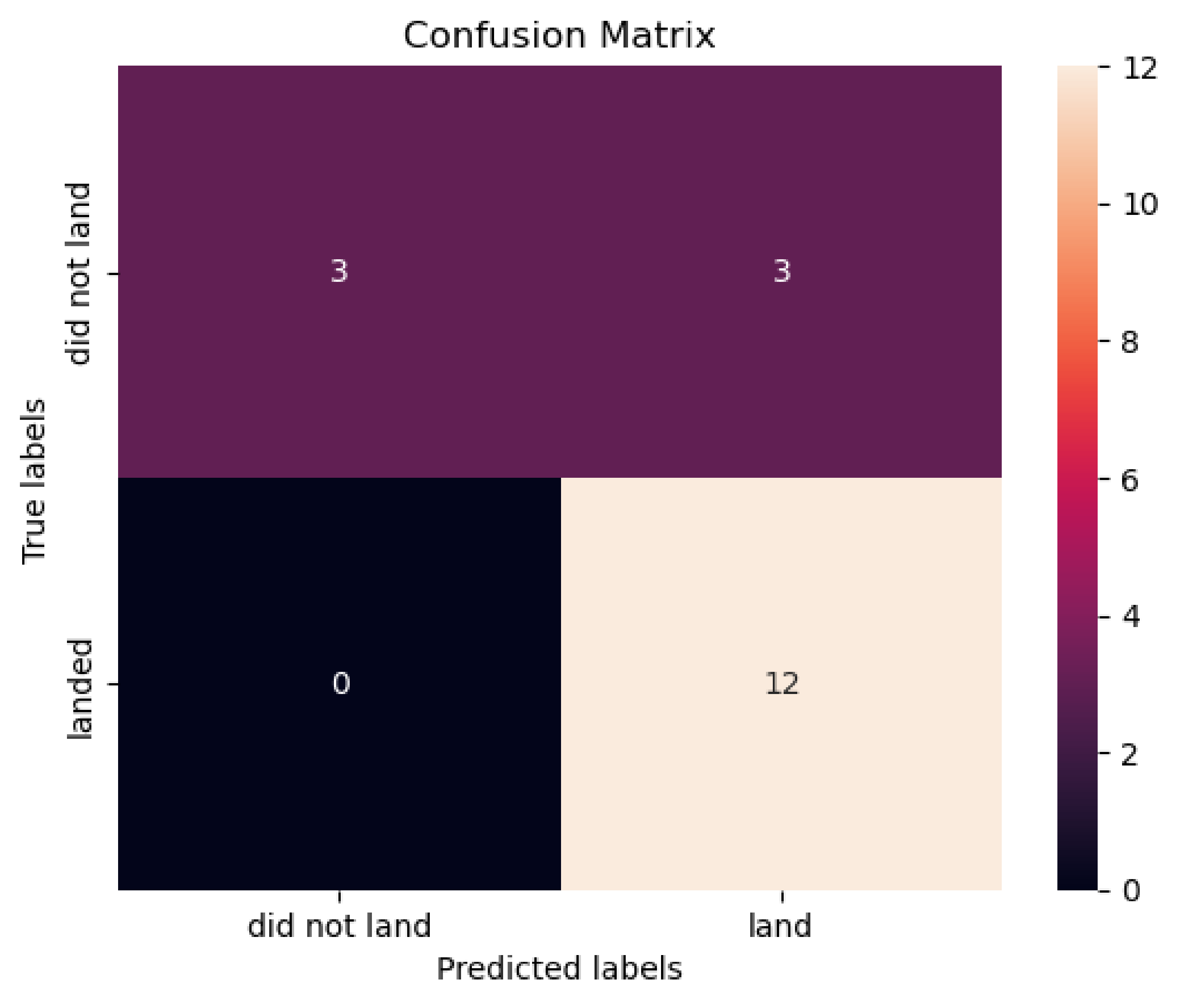
- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

- The best model is the tree classification model with an overall of 0.88 from 1.0 of correct prediction



Confusion Matrix



- Show the confusion matrix of the best performing model with an explanation
- The confusion matrix show us that the model is good at predicting landings but is bad for predict Fail landings, from it predicts all correct landing but NONE of the fail landings

Conclusions

- The EDA shows a strong correlation between place and success
- Also payload mass and kind of booster is important to ensure success landing
- Some orbits have better success rate and higher payloads are destined
- The success rate has an increase over the years
- The EAST coast of Florida is the place with more launches
- The best rocket is Falcon9 with 97% and site KSC-LC-39A with success rate of 76.9%
- The first success landing was on Dec –22- 2015
- There is a total of 619 Tons of Payload launched to space from SpaceX
- Falcon 9 has the highest success landing rate with 98/101 success landings
- Our models can accurately predict 88% of the success landings but the model is bad predicting FAIL landings,
- Decision Tree is the best predictive model from four tested LR,SVM, DT and KNN.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!