IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amine Azaiez

05/09/2023

Outline

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

Executive Summary

- The data bases will be imported, cleaned and wrangled, then interrogated by sql and python code snippets to reveal relationships between various variables
- Classification methods will be tested to find out the best prediction model to implement
- Answers to this questions and the optimum choices to ensure the success of a launch will be inspected and detailed in the following presentation

Introduction

- The Space X Falcon 9 first stage databases are detailed records about all launches aspects.

- Can we get use of this databases to find out the potential correlation between the various variables?

- Is it possible to train a model that predicts the best the outcome of launch ?

Section 1

Methodology

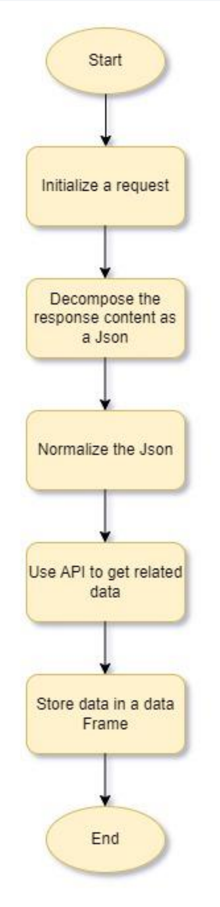
Methodology

Executive Summary

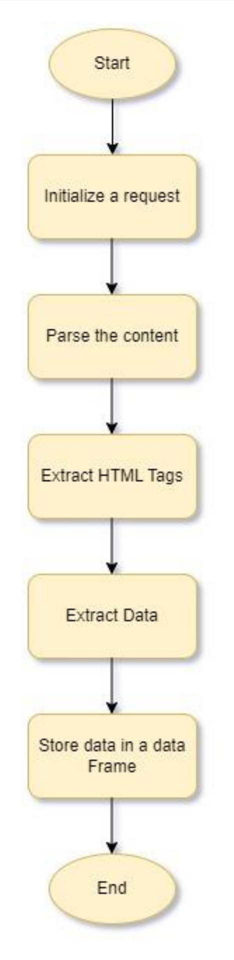
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Via API


Start
Initialize a request
Decompose the response content as a Json
Normalize the Json
Use API to get related data
Store data in a data Frame
End

Via Web Scrapping



Data Collection – SpaceX API

- The source of data:

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

https://api.spacexdata.com/v4/ ...

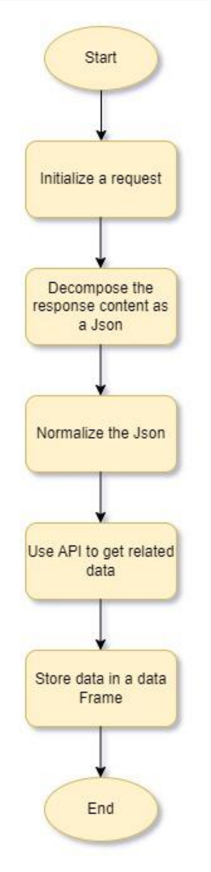
- Initialize the request:

- Decompose and Normalize the response:

```
JsonResponse=response.json()

data=pd.json_normalize(JsonResponse)
```

- Use API to get related data to feed main data frame

- GitHub Link: https://github.com/Amine-Azaiez/Capstone/blob/1798d3418ad395b96d0dbe3f5340b6707a6bb67a/jupyter-labs-webscraping.ipynb


Start
Initialize a request
Decompose the response content as a Json
Normalize the Json
Use API to get related data
Store data in a data Frame
End

Data Collection - Scraping

- The source of data:

https://api.spacexdata.com/v4/launches/past

- Initialize the request:

```
response= requests.get(static_url)
```

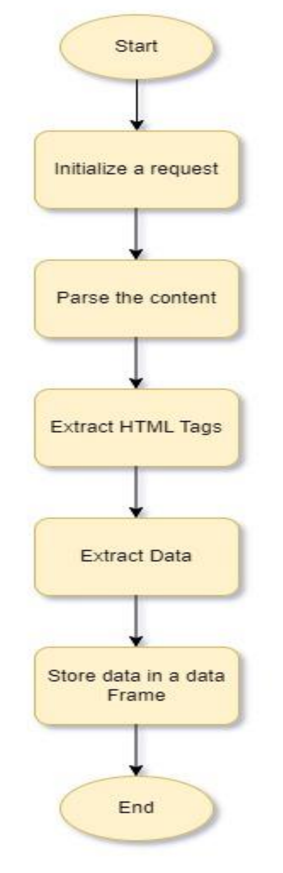
- Parse the content:

```
soup= BeautifulSoup(response.content,'html.parser')
```

- Extract Data:

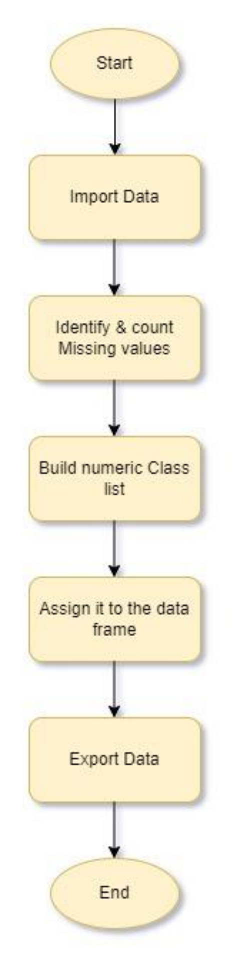
```
• for table_number,table in
  enumerate(soup.find_all('table',"wikitable plainrowheaders
  collapsible")):

•    # get table row

•     for rows in table.find_all("tr"):

         ….
```

- GitHub Link: https://github.com/Amine-



Data Wrangling

- Identify and count missing values: `df.isnull().sum()/df.shape[0]*100`
- Build numeric 'landing_class' list: `landing_class=[ 0 if outcome in bad_outcomes  else 1 for outcome in df['Outcome']]`
- Add a 'Class' column to the data frame: `df['Class']=landing_class`
- Export data: `df.to_csv("dataset_part\_2.csv", index=False)`
- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with Data Visualization

- Scatter plots are exploited to reveal if there is a correlation between specific data variables.

- Vertical bar chart is implemented to compare the successful rates relative to each orbit

- A line chart serves to present the evolution of the success rate over years

- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.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 average payload mass carried by booster version F9 v1.1
- Display the total payload mass carried by boosters launched by NASA (CRS):
- List the date when the first successful landing outcome in ground pad was achieved

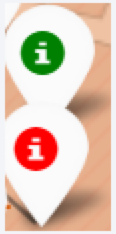
EDA with SQL

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes:
- List the names of the booster_versions which have carried the maximum payload mass
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015:
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Used Map objects:
  - **Circles:** to highlight the area of the launch site

  - Markers: to trace whether a launch in a specific site was successful or failed



  - Lines: the lines are drawn to show the distance between a launch site and a key positon
- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

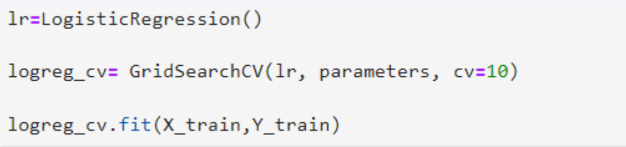
Build a Dashboard with Plotly Dash

- A pie chart is used to present the contribution of every launch site into the successful launches, this can help to visualize witch launch sites are more relevant.
- The pie chart can be used also to show the successful rate of a selected launch site
- A scatter plot is implemented to display the launch outcome by a selected payload range and launch site.
- The booster version will be characterized by the color of the dots.
- We can exploit this scatter plot to infer what booster version is better for what payload range

- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/Dash.ipynb

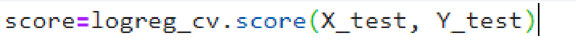
Predictive Analysis (Classification)

- Creation and training of a classification model example:



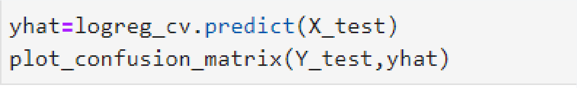
```
lr=LogisticRegression()

logreg_cv= GridSearchCV(lr, parameters, cv=10)

logreg_cv.fit(X_train,Y_train)
```

- Get score as an evaluation :



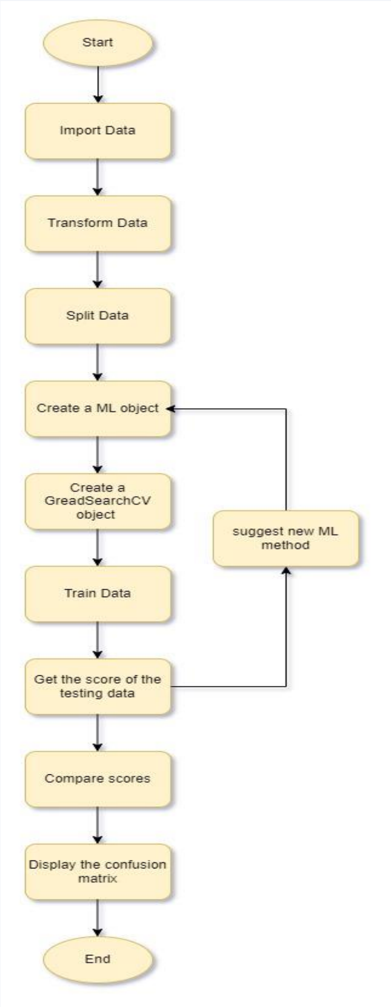
```
score=logreg_cv.score(X_test, Y_test)
```

- Get the confusion matrix:



```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

- GitHub URL: https://github.com/Amine-Azaiez/Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Start

Import Data

Transform Data

Split Data

Create a ML object

Create a GreadSearchCV object

Train Data

Get the score of the testing data

Compare scores

Display the confusion matrix

End

suggest new ML method

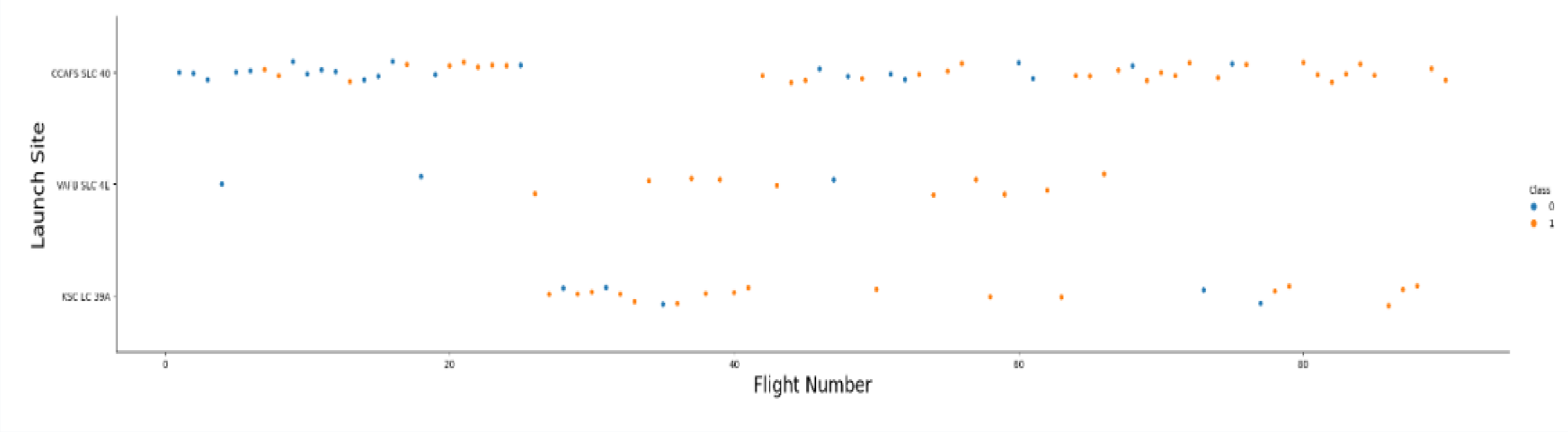
Results

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

Section 2

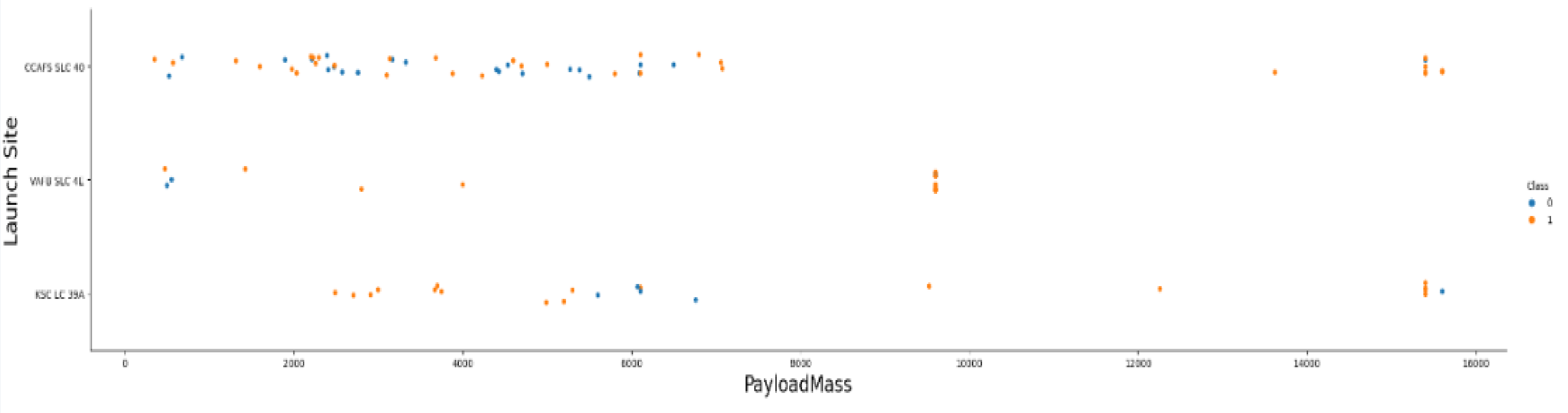
Insights drawn from EDA

Flight Number vs. Launch Site



- This screen shot already depicts that CCAFS SLC 40 and KSC LC 39A were able to provide successful flight numbers superior to 80.

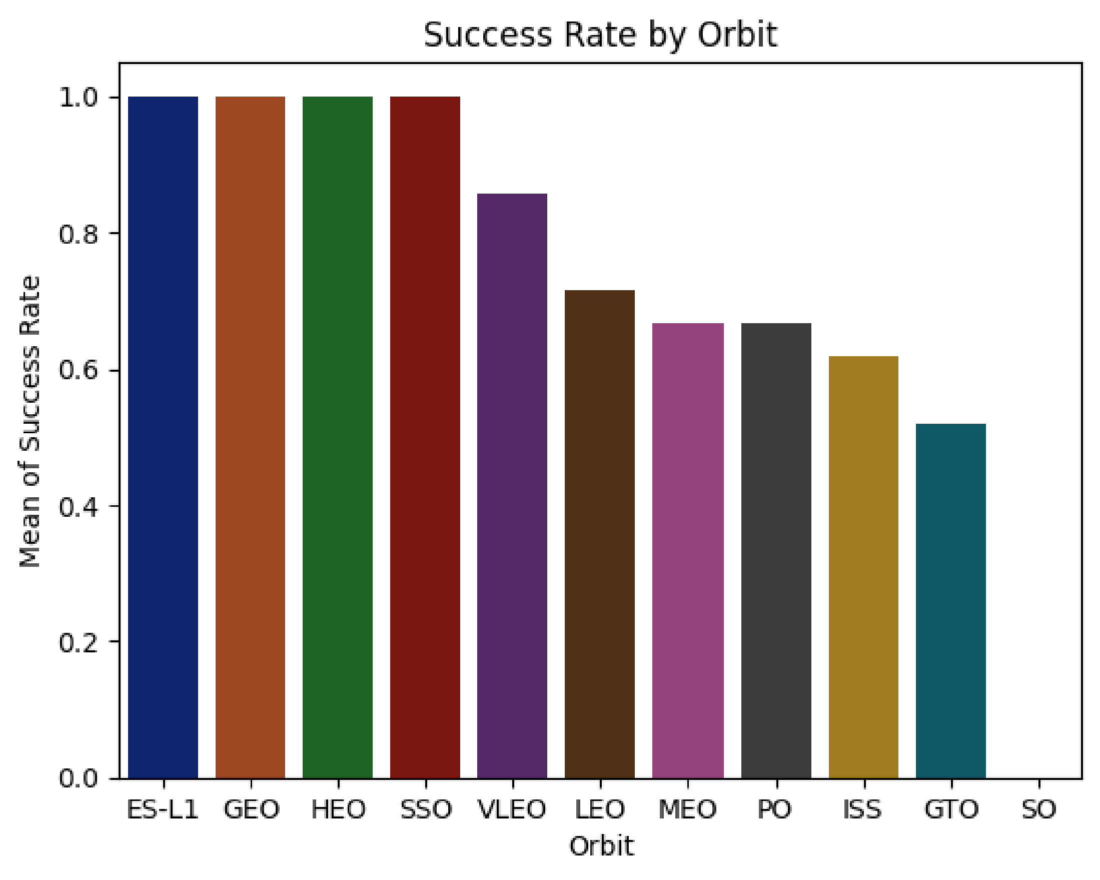
Payload vs. Launch Site



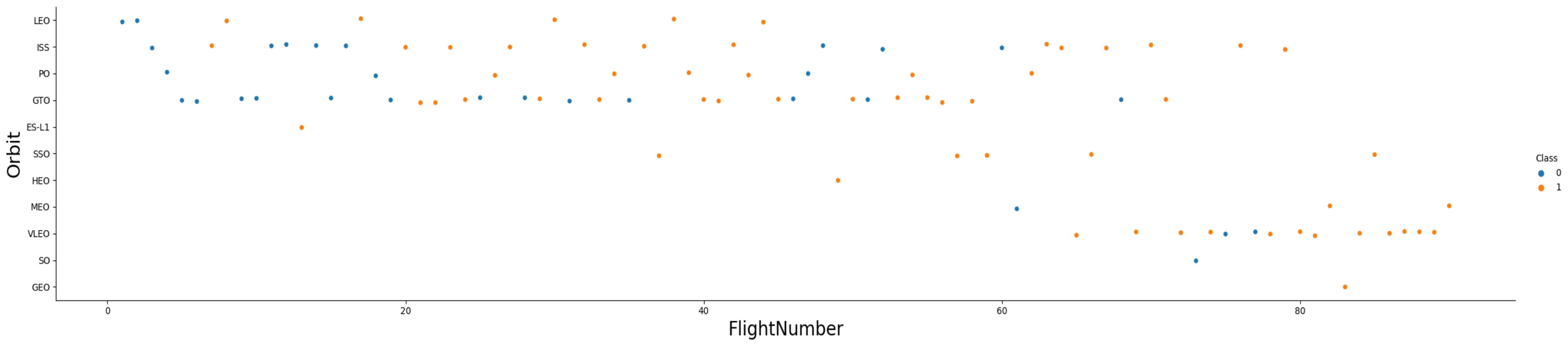
- This screen shot reveals that the three launch sites are used for payloads under 8000 Kg, but CCAFS SLC 40 and KSC LC 39A are able also to launch payloads near 16000 Kg

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO turned out to be always successful orbit destinations.
- GTO and ISS are the least successful orbit destinations (<60 %)

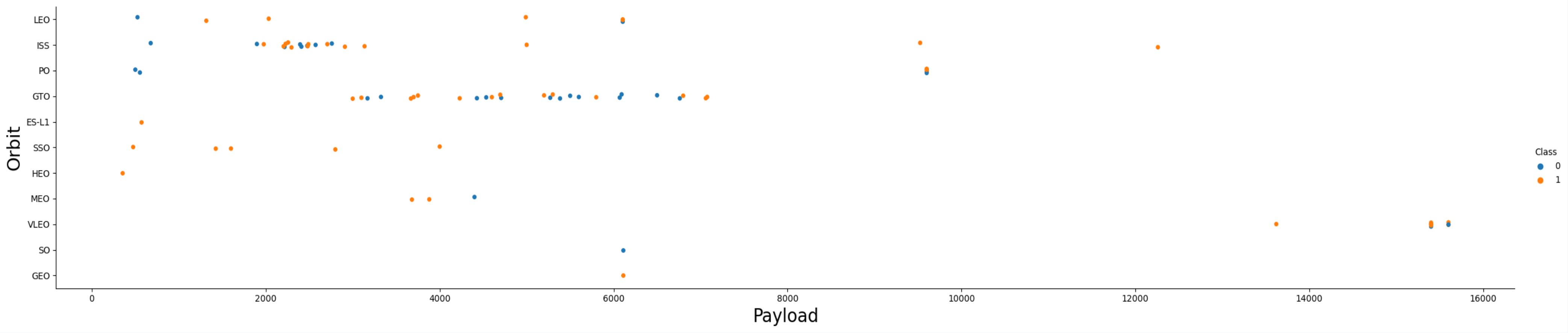


Flight Number vs. Orbit Type



- The screen shot reveals that for higher flight numbers VLEO is the best orbit destination

Payload vs. Orbit Type

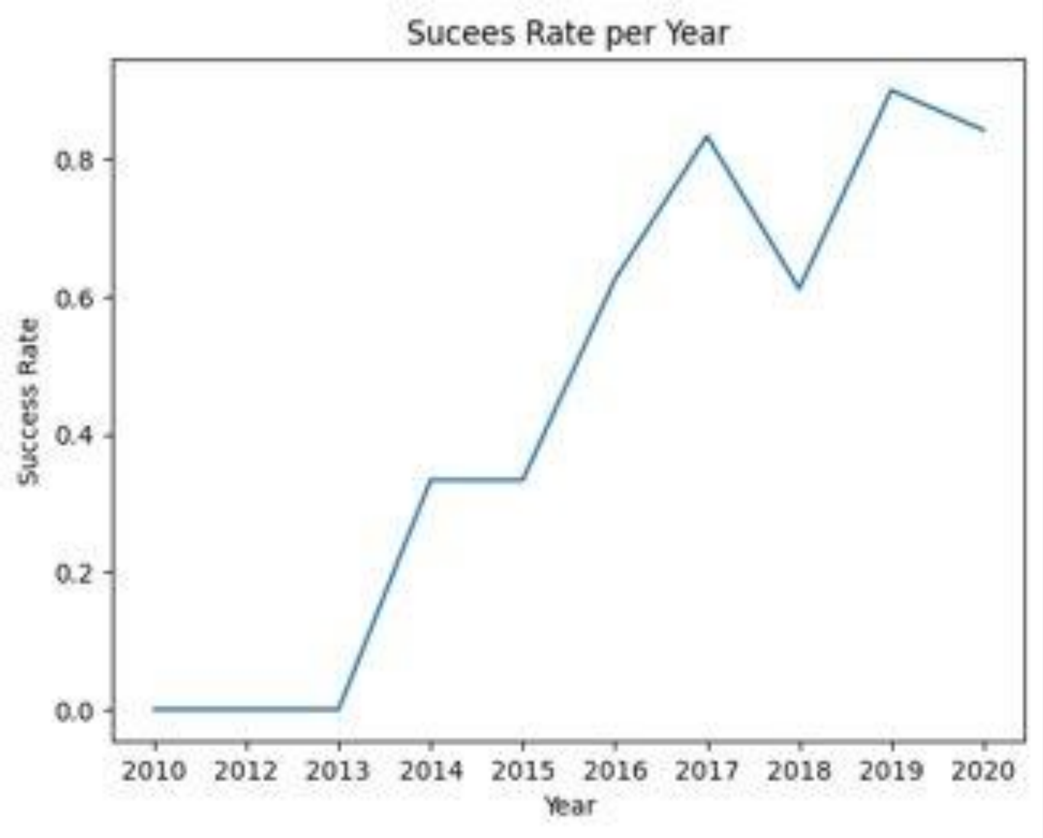


- Most payloads up to 8000 Kg can be ensured to orbits like: LEO, ISS, PO and GTO

- For payloads up to 16000 Kg, VLEO is mainly the optimum orbit destination

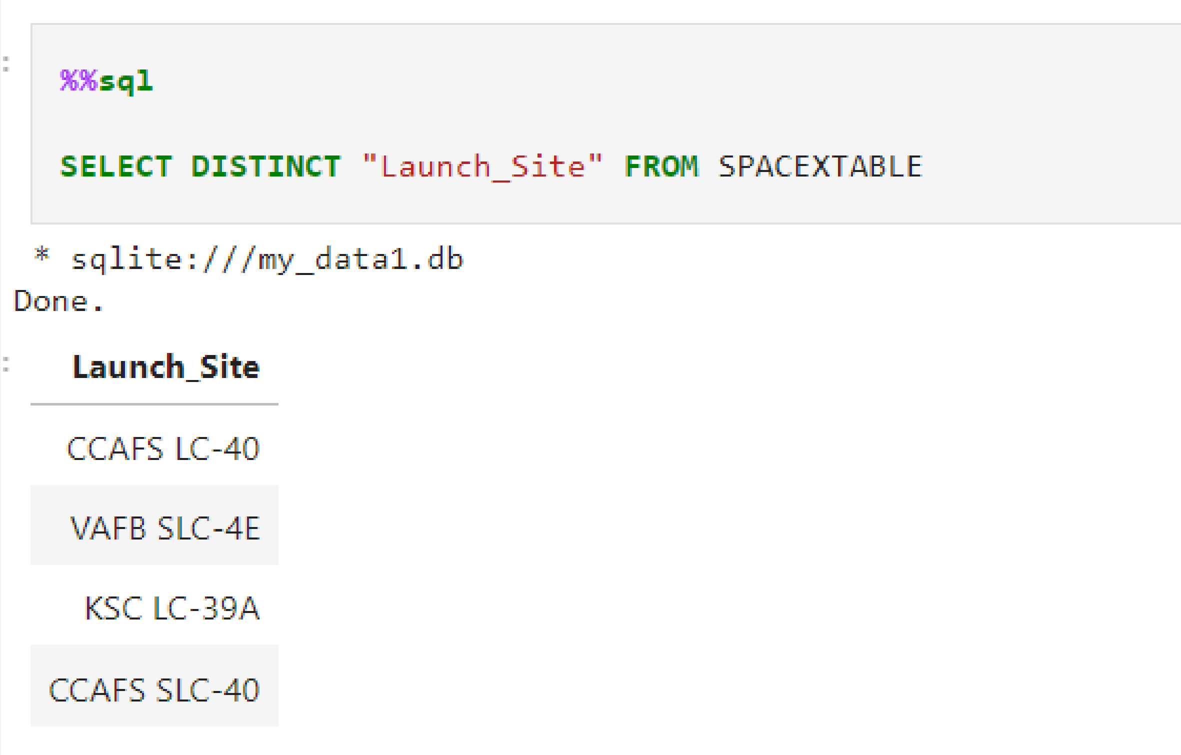
Launch Success Yearly Trend

- There has been a steady improvement in the success rate over the years.



All Launch Site Names

- Using "distinct" feature in a sql query :



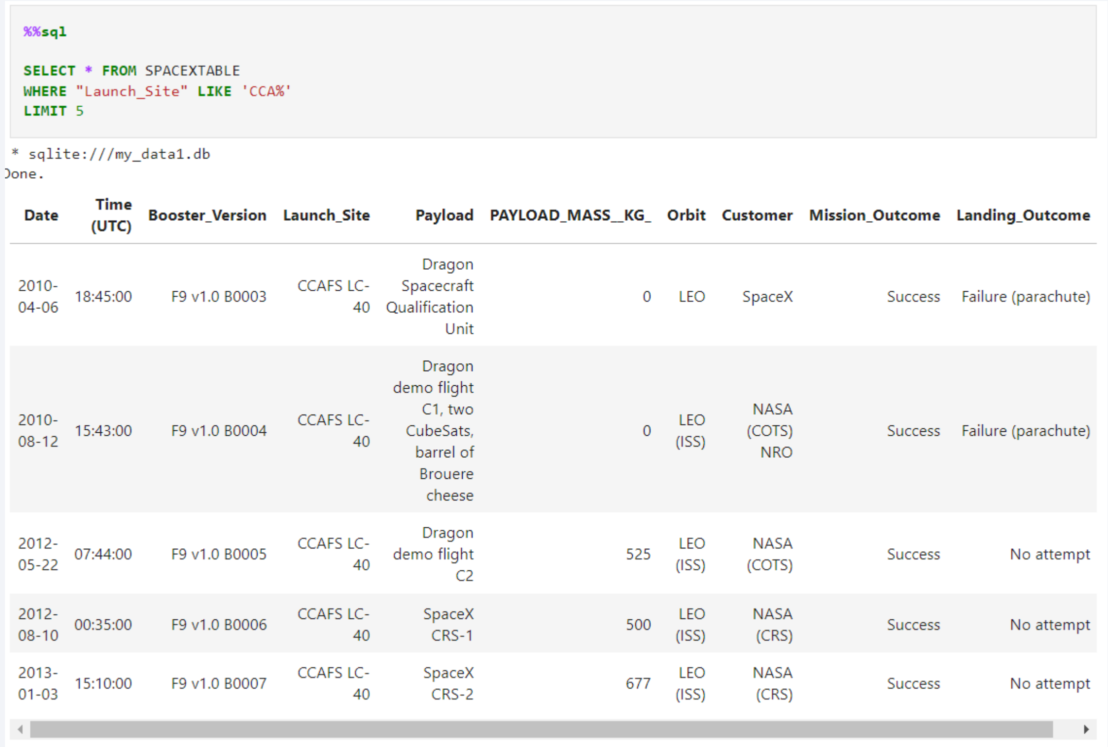
```
%%sql

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 'Like' sql feature in the 'where' clause



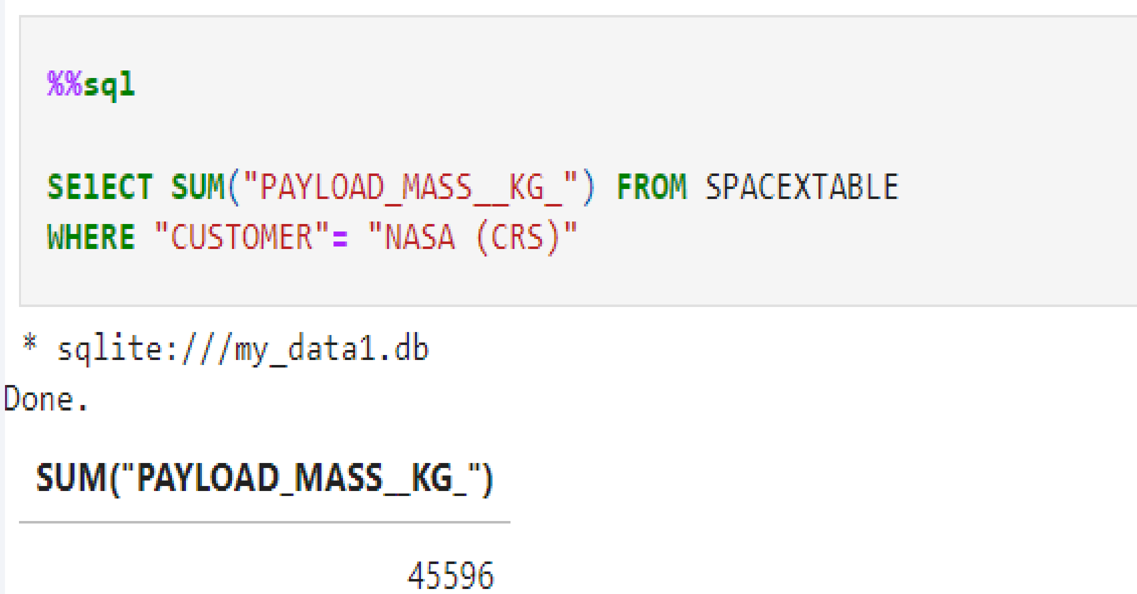
```
%%sql

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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We used the 'SUM' sql feature and filter on customer = NASA



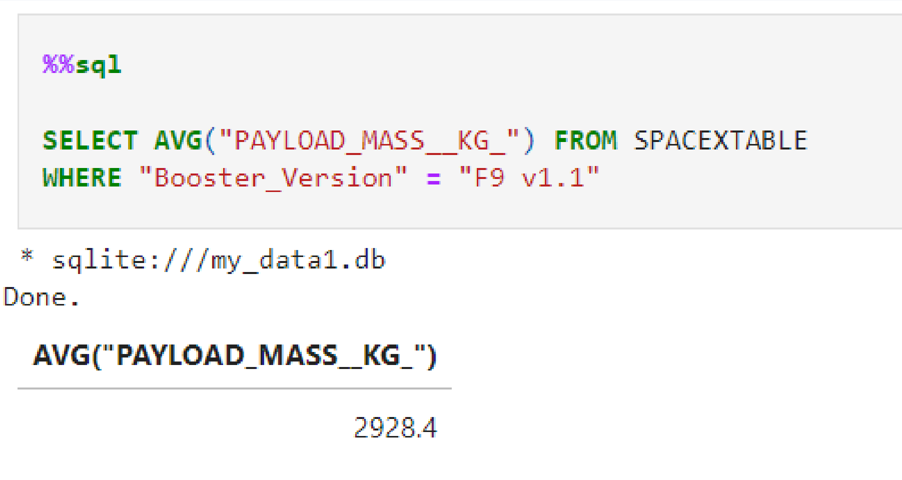
```
%%sql

SElECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
WHERE "CUSTOMER"= "NASA (CRS)"
```

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- We used 'AVG' sql feature and filter on Booster version = "F9 v1.1"



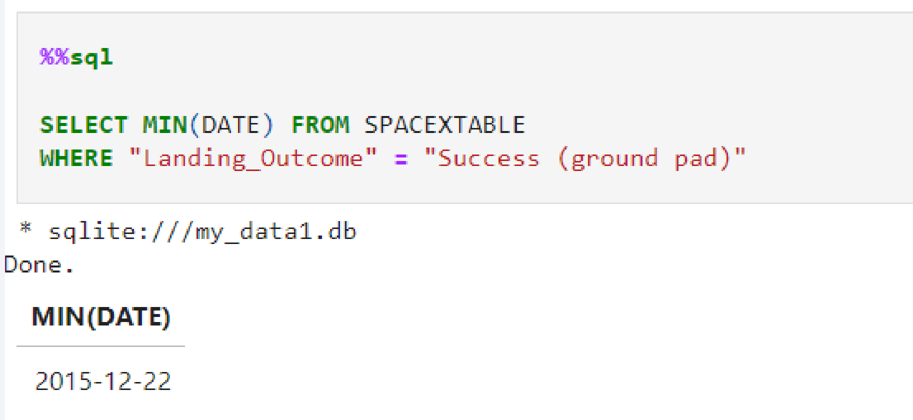
```
%%sql

SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
WHERE "Booster_Version" = "F9 v1.1"
```

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

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- 'MIN' is used to get the first date, and a filter on the landing Outcome as a Success on ground pad is applied



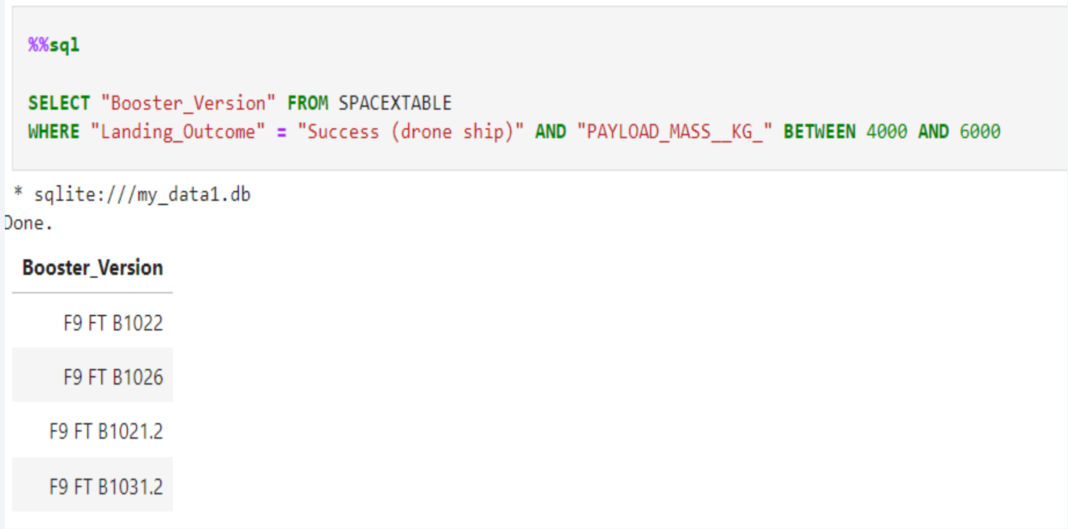
```
%%sql

SELECT MIN(DATE) FROM SPACEXTABLE
WHERE "Landing_Outcome" = "Success (ground pad)"
```

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

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The landing Outcome is filtered as success on a drone ship and the payload is filtered to be in a range of [4000,6000]



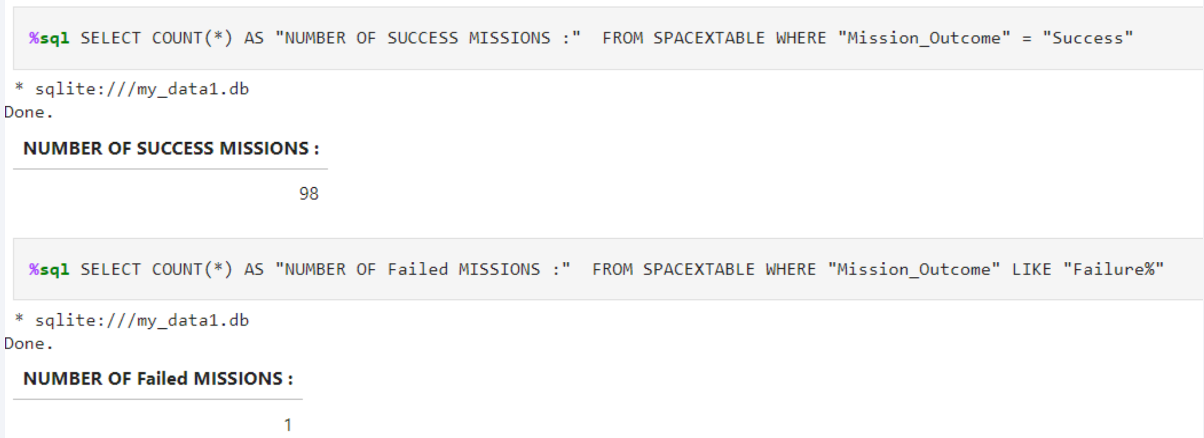
```
%%sql

SELECT "Booster_Version" FROM SPACEXTABLE
WHERE "Landing_Outcome" = "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

Total Number of Successful and Failure Mission Outcomes

- A filter is applied on the Mission outcome, and a count(*) of the results is performed.



```
%sql SELECT COUNT(*) AS "NUMBER OF SUCCESS MISSIONS :"  FROM SPACEXTABLE WHERE "Mission_Outcome" = "Success"
```

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

```
NUMBER OF SUCCESS MISSIONS :
                          98
```

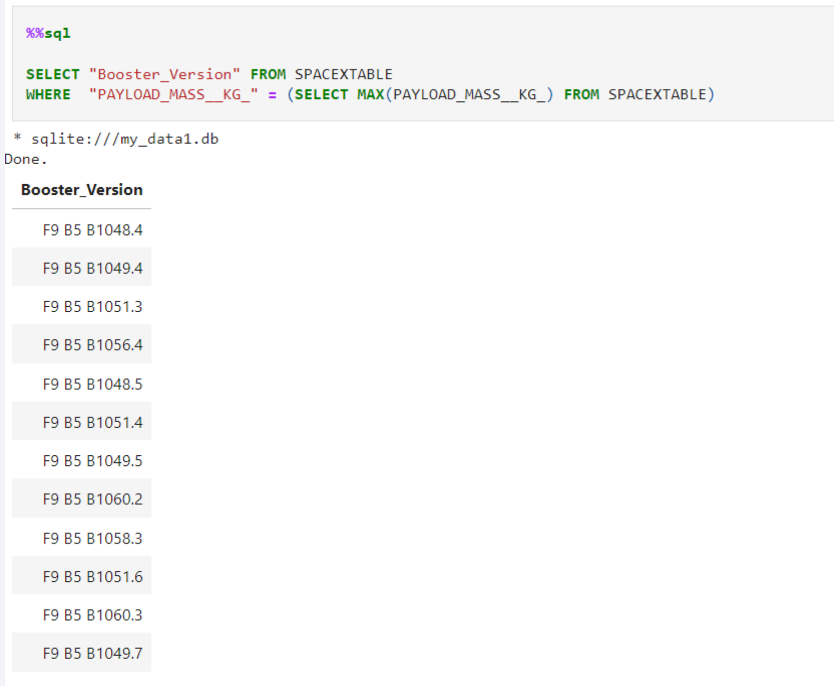
```
%sql SELECT COUNT(*) AS "NUMBER OF Failed MISSIONS :"  FROM SPACEXTABLE WHERE "Mission_Outcome" LIKE "Failure%"
```

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

```
NUMBER OF Failed MISSIONS :
                          1
```

Boosters Carried Maximum Payload

- A nested query is launched to determine the max payload
- The main query will filter the booster versions witch have been used for that maximum payload



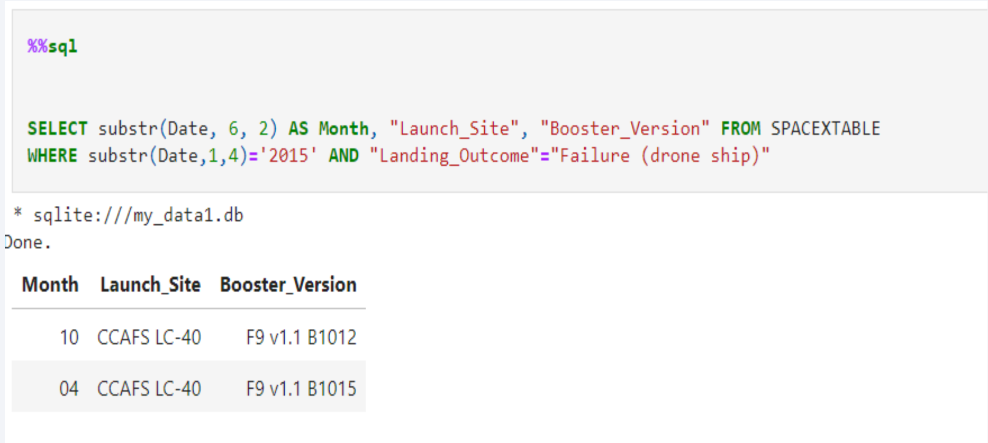
```
%%sql

SELECT "Booster_Version" FROM SPACEXTABLE
WHERE  "PAYLOAD_MASS__KG_" = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * 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

2015 Launch Records

- A filter is applied on the year
- Another filter is applied on the landing outcome



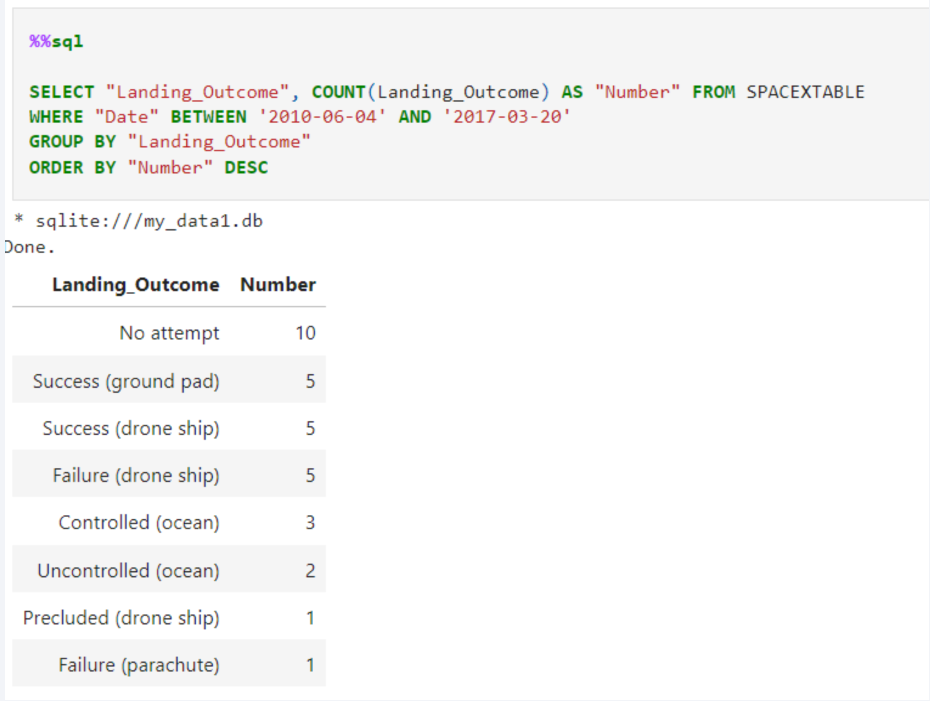
```
%%sql

SELECT substr(Date, 6, 2) AS Month, "Launch_Site", "Booster_Version" FROM SPACEXTABLE
WHERE substr(Date,1,4)='2015' AND "Landing_Outcome"="Failure (drone ship)"
```

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

Month	Launch_Site	Booster_Version
10	CCAFS LC-40	F9 v1.1 B1012
04	CCAFS LC-40	F9 v1.1 B1015

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

- A filter on the date range is applied
- The results are grouped by the landing outcome
- The count of every landing outcome is performed
- The list is ordered desc by the number of occurrence counted



```
%%sql

SELECT "Landing_Outcome", COUNT(Landing_Outcome) AS "Number" FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY "Number" DESC
```

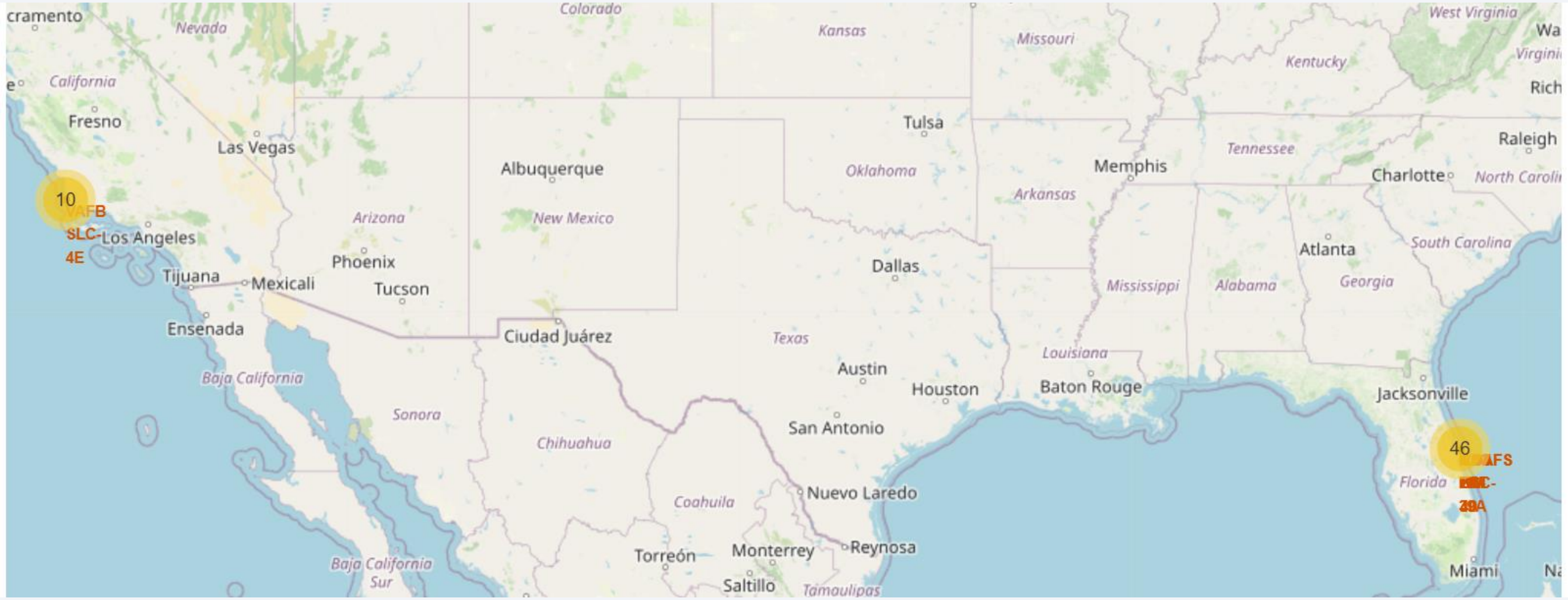
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

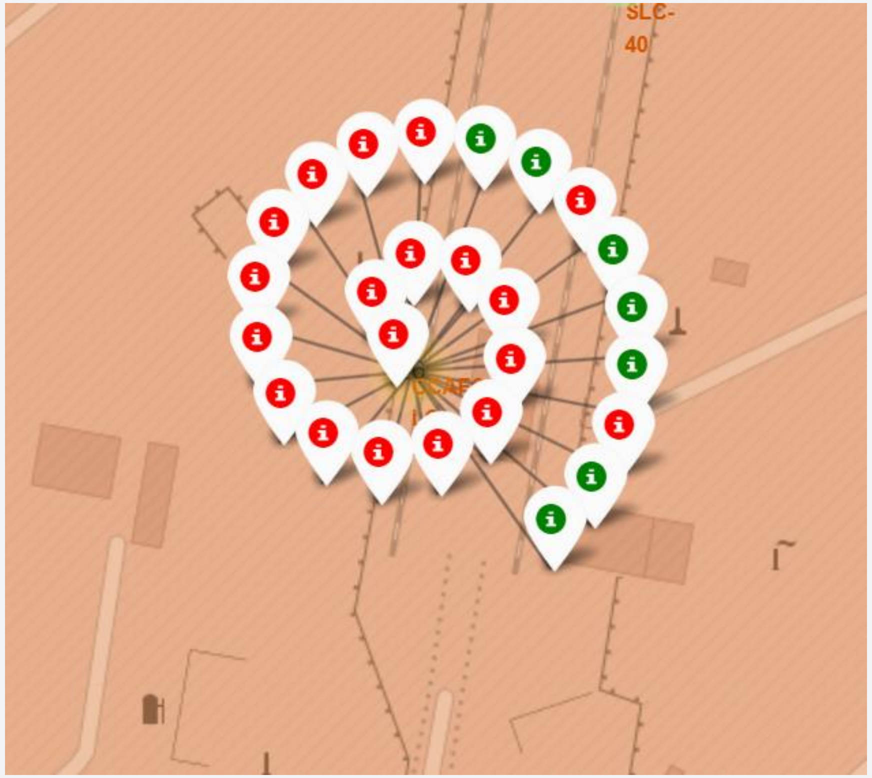
Launch Sites Proximities Analysis

Global Launch Sites Overview



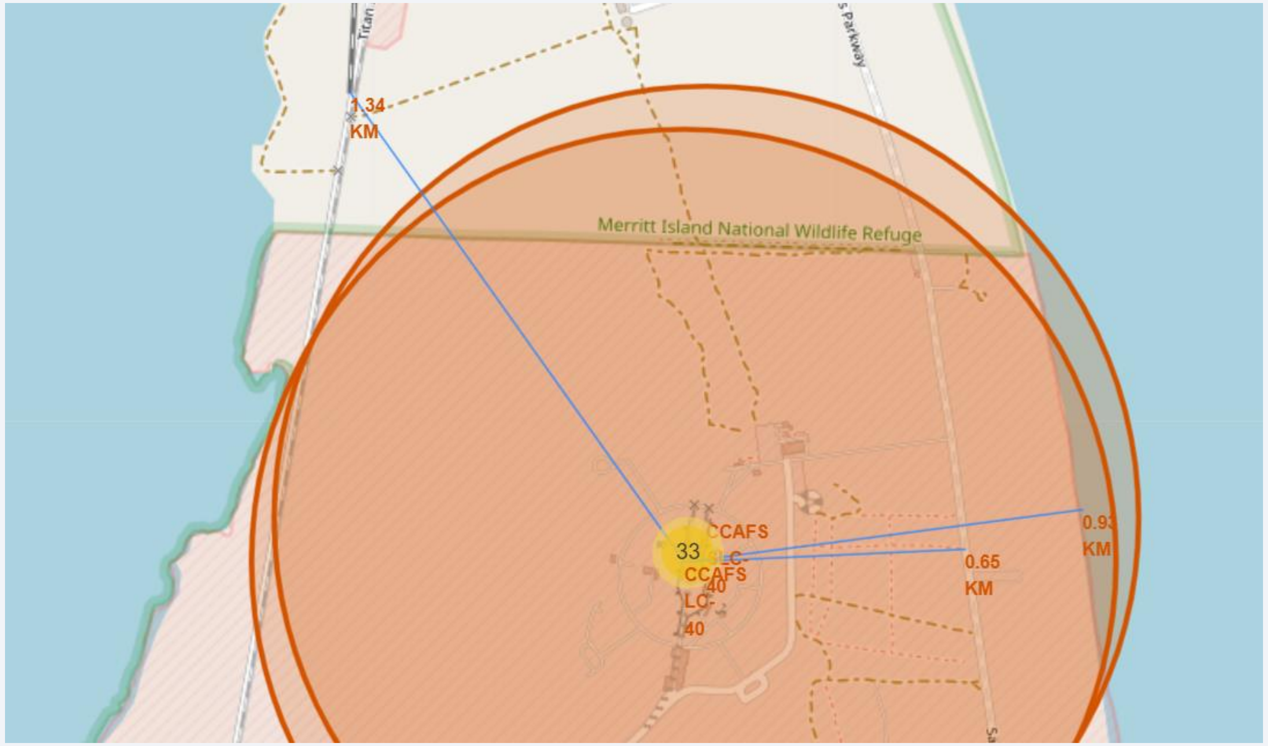
- While launch sites are laid on both sides of the USA, Eastern sites are responsible for the majority of launches.

Launch Outcomes per one site

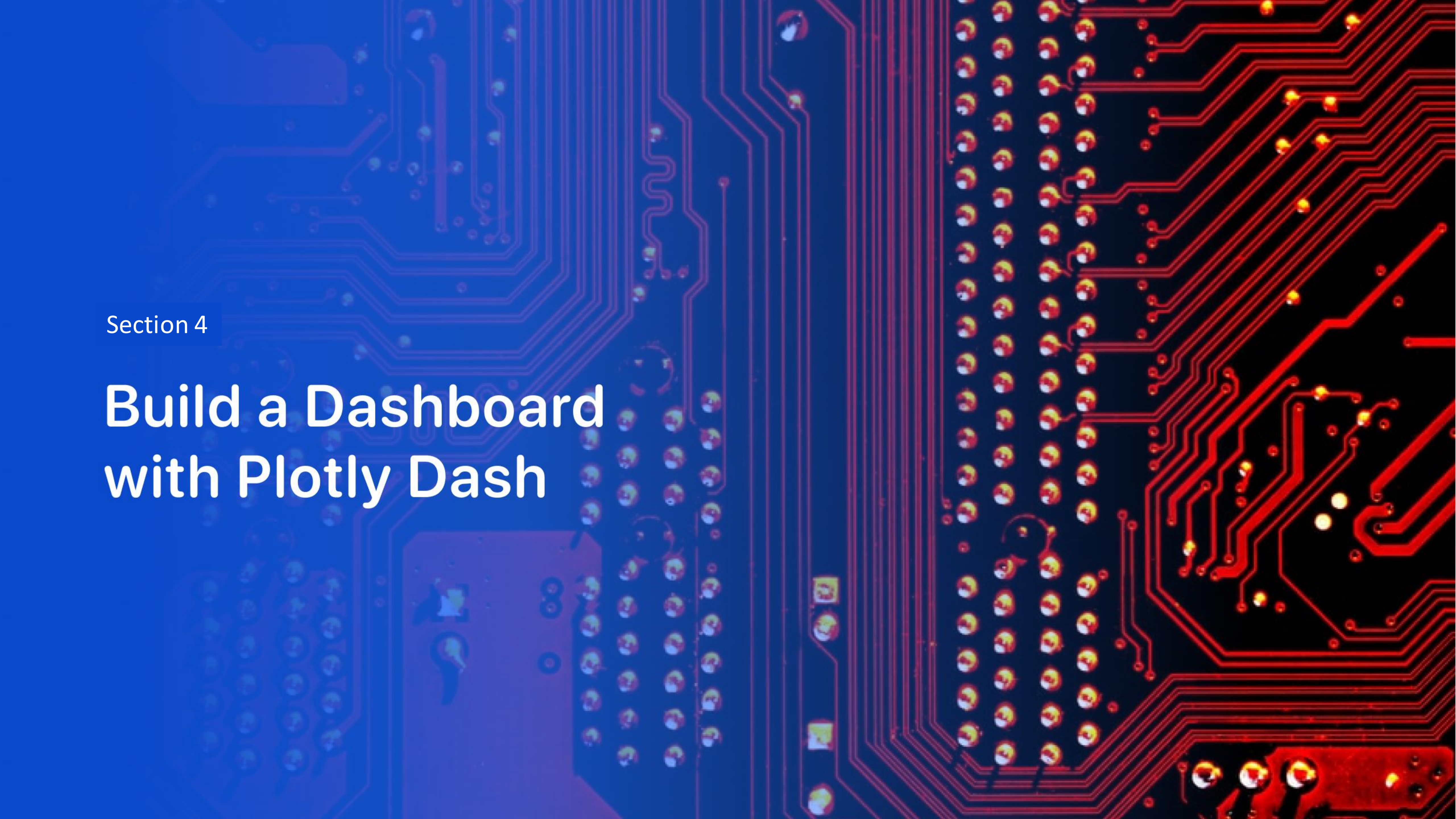


- While the majority of the markers are in red, the fact that they are in chronological order reveals that the majority of last launches are successful.

Launch Site distance to main key points



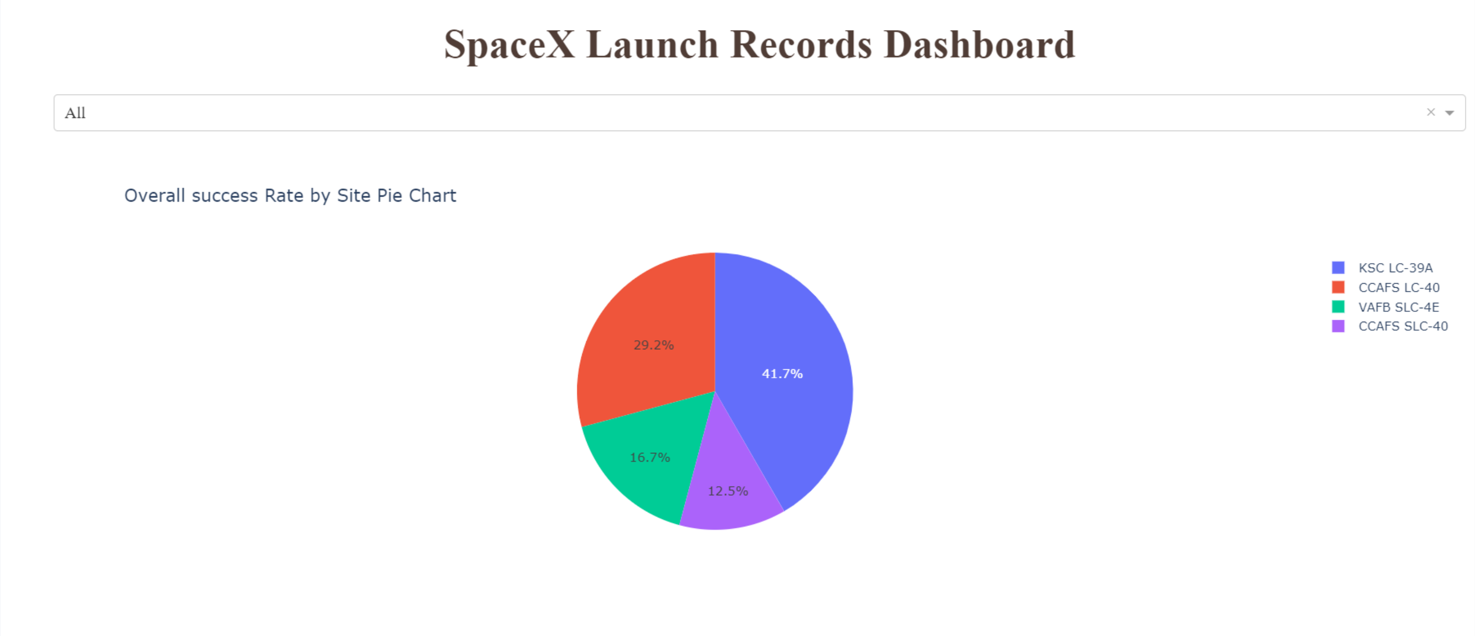
- All key points like coastline, highway or railway are about 1 km far of the launch site.



Section 4

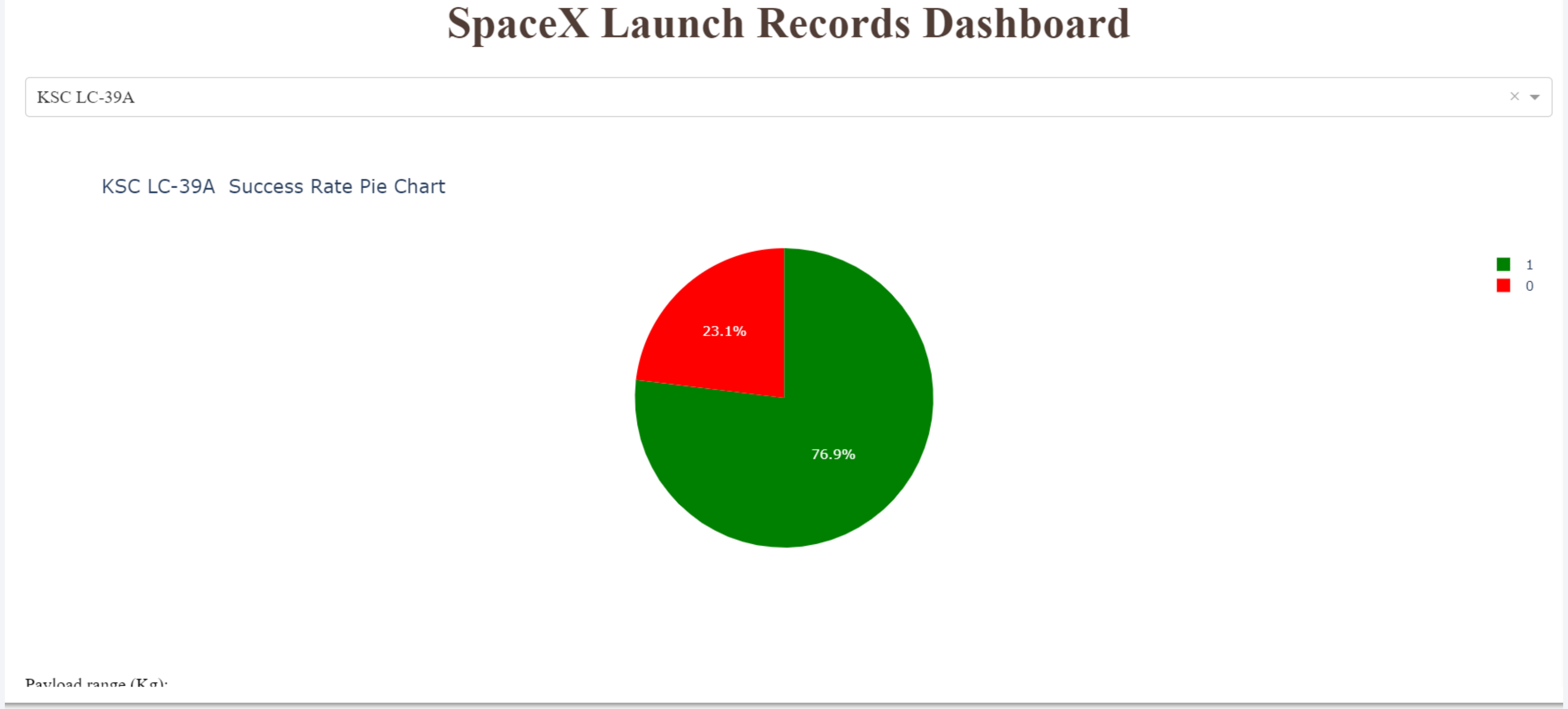
Build a Dashboard with Plotly Dash

Success Count for All Launch Sites


SpaceX Launch Records Dashboard
All
Overall success Rate by Site Pie Chart
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40

- KSC LC-39A Launch Site alone accounted for 41,7 % of the overall successful launched.
- KSC LC-39A and CCAFS LC-40 together made up 71% of the overall successful launched.

KSC LC-39A Success Rate



SpaceX Launch Records Dashboard

KSC LC-39A ×

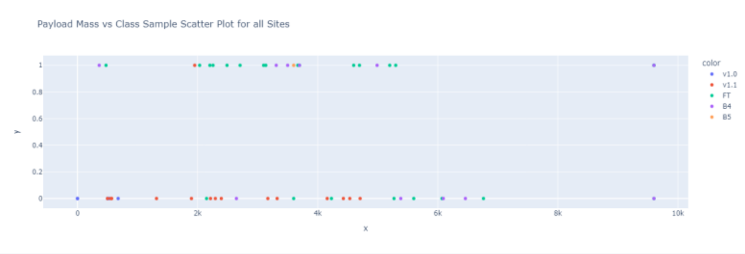
KSC LC-39A Success Rate Pie Chart

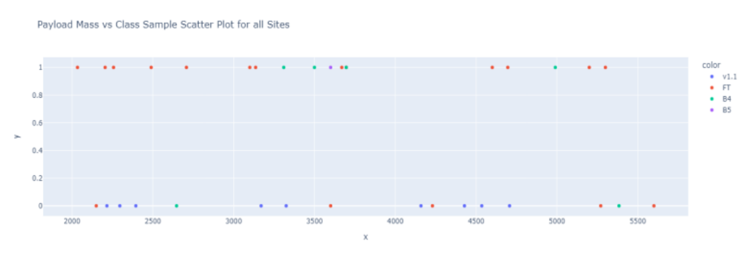
Payload range (Kg):

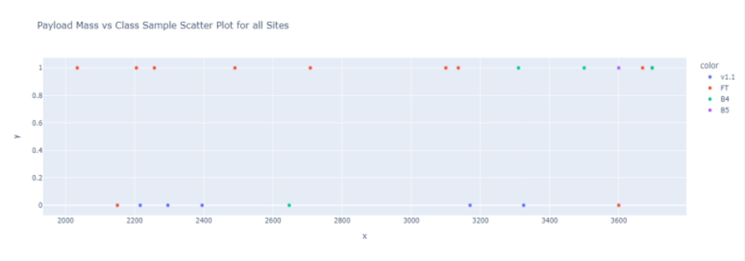
- While KSC LC-39A represents the highest number of successful launch records, it represents also the highest successful ratio: 77%

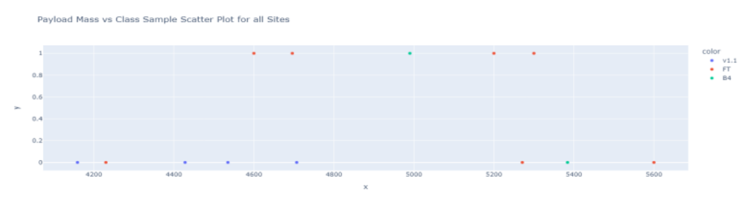
Payload vs Launch Outcome Scatter Plot

- Scatter plot with the whole payload range: seems that successful launches are relative to payloads between 2000 and 6000 Kg.
- Scatter plot with payload range between 2000 Kg and 6000 Kg: We can still divide this range into two halves
- Scatter plot with payload range between 2000 Kg and 4000 FT Booster showed the best results, B5 the worst
- Scatter plot with payload range between 4000 Kg and 6000 Kg: FT booster still presents the best results but v1.1 is the worst


Payload Mass vs Class Sample Scatter Plot for all Sites


Payload Mass vs Class Sample Scatter Plot for all Sites


Payload Mass vs Class Sample Scatter Plot for all Sites

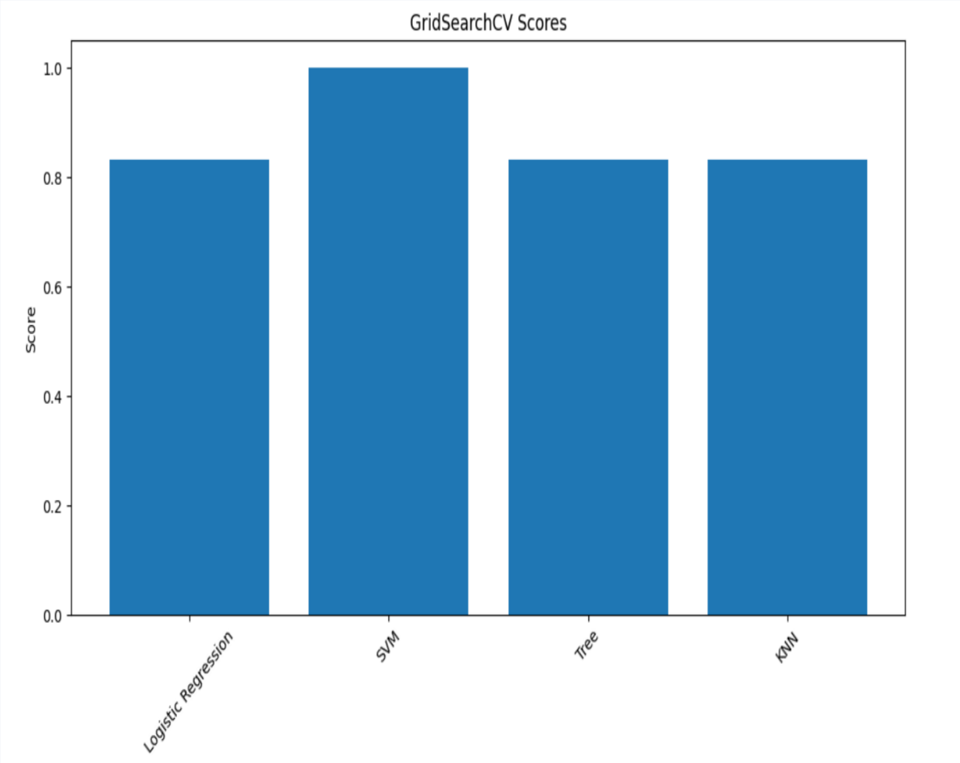

Payload Mass vs Class Sample Scatter Plot for all Sites

Section 5

Predictive Analysis (Classification)

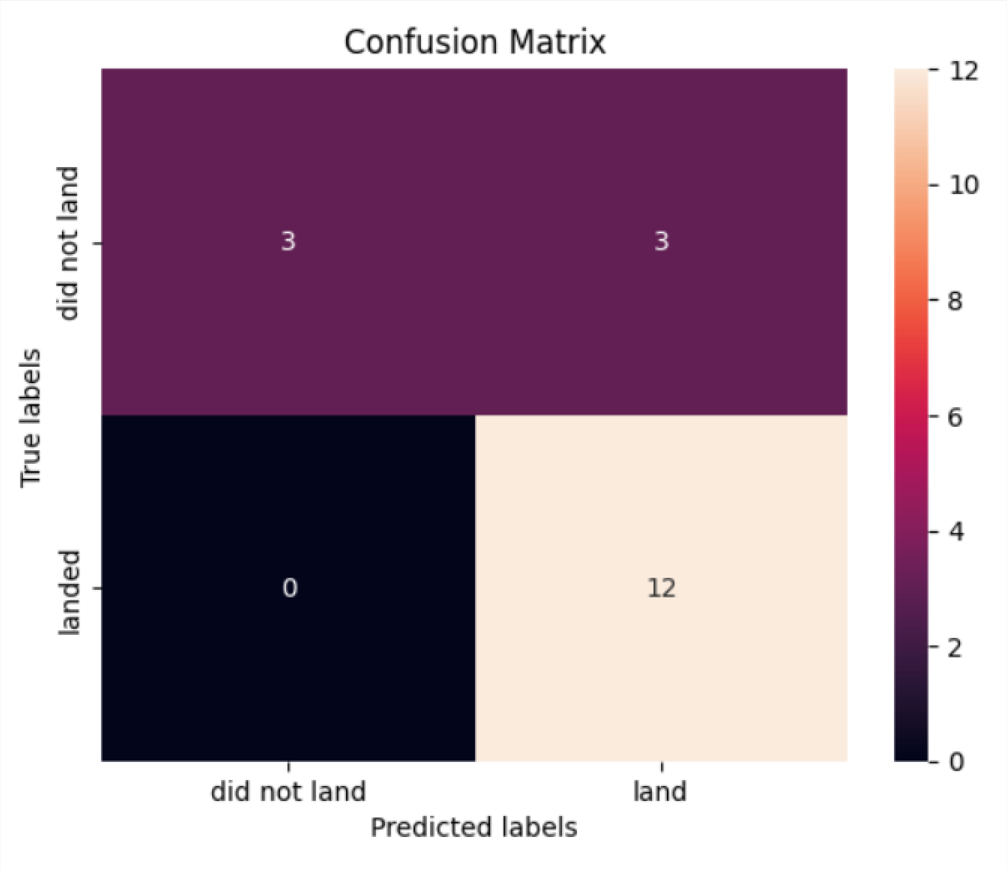
Classification Accuracy

- SVM proved to be the best method with an accuracy score up to 1



Confusion Matrix

- The confusion matrix of the SVM Method reveals the following facts:
- No predicted fail launch proved to be a success launch
- 12 out of 15 launches predicted as successful turned out to be really successful



Conclusions

- CCAFS SLC 40 and KSC LC 39A are the better launch sites
- CCAFS SLC 40 and KSC LC 39A can be used to launch payloads near 16000 Kg
- ES-L1, GEO, HEO and SSO turned out to be always successful orbit destinations
- For payloads up to 16000 Kg, VLEO is mainly the optimum orbit destination
- There has been a steady improvement in the success rate over the years.
- While KSC LC-39A represents the highest number of successful launch records, it represents also the best launching site regarding the success rate
- In order to predict a potential launch, the SVM approach is the best

Appendix

- Git Hub URL for all Notebooks:

https://github.com/Amine-Azaiez/Capstone/tree/main

Thank you!