IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

TACYARA SAGAE
01/03/2022

Outline

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

Executive Summary

SUMMARY OF METHODOLOGIES

- Data Collection
- Data Wrangling
- EDA with Data Visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive Analysis (Classification)

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 predict if the Falcon 9 first stage would land successfully.
The Space X company announced on its website that it would be able to launch Falcon 9 rockets with a cost of **62 million dollars**, while other providers cost upward of **165 million dollars** each.

The significant lower cost and advantage of Space X its because the company claims it is able to reuse the first stage. Therefore, if it is possible to determine if the first stage will land, it is possible to predict the cost of a launch.

This information is valuable if an alternate company wants to bid against Space X for a rocket launch!

THE PURPOSE OF THE PROJECT TO FIND ANSWERS

- What are the main influencers in the rocket launch?

- What is the effect of the relationship between the variables? What will determine the success rate of the rocket launch?

- How can Space X achieve the best results to ensure a successful launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using API and Webscrapping
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and dropping unnecessary columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plots: Bar Graphs, Scatter Graphs, Folium Maps and Plotly Dash in order to perform exploratory data analysis.
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

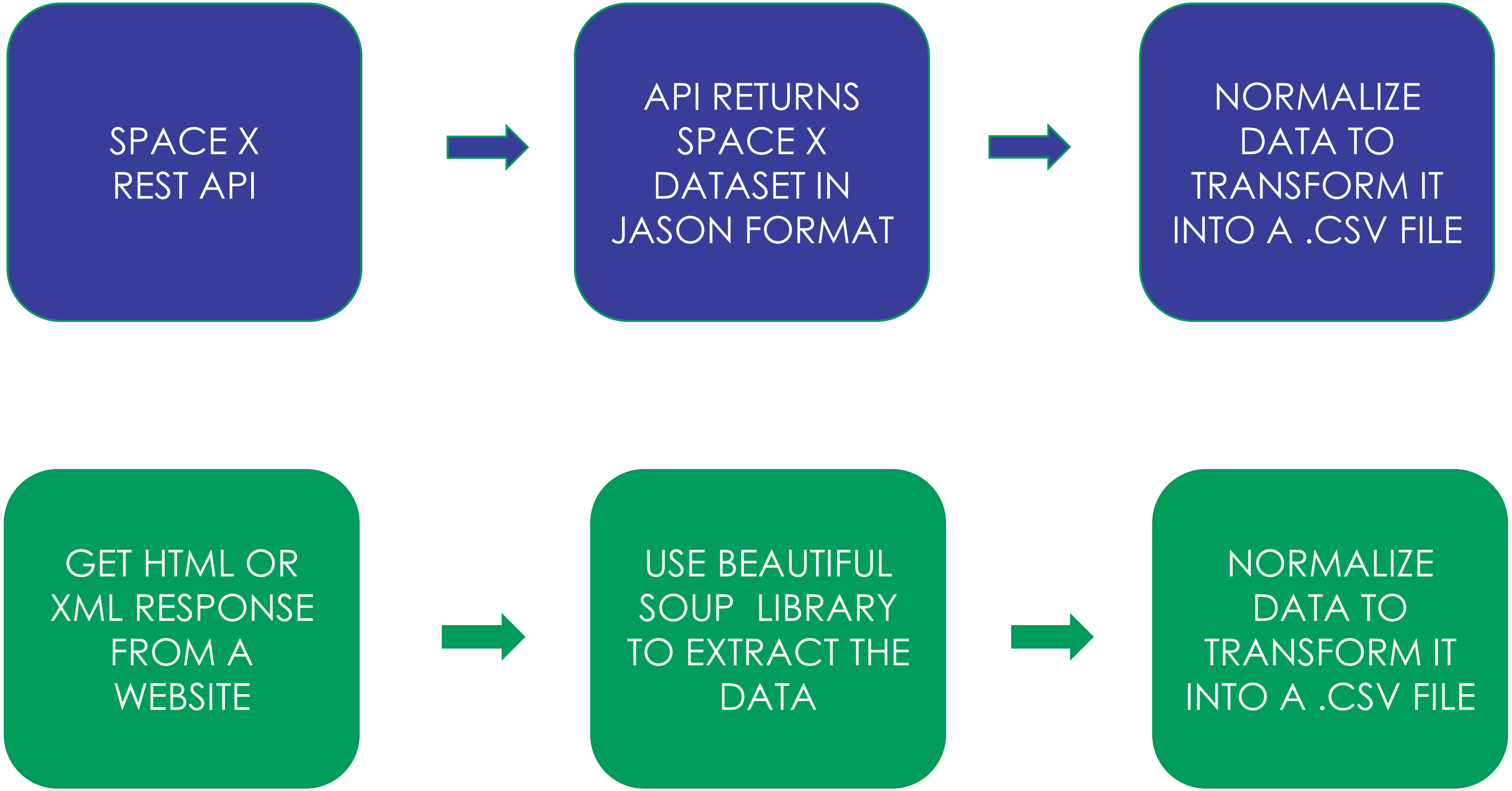
Data Collection

THE PROCESS TO DATA SET COLLECTION

- Extracting data collected through the Space X launch REST API
- The API contains informations about the rocket used, payload delivered, launch specifications, landing informations and outcomes.
- The goal is to exploit this dataset to find out whether Space X is able to launch the rocket under the assumed conditions or not.
- The Space X REST API endpoints ou URL starts with api.spacexdata.com/v4/.
- Another way to get the data Falcon 9 Launch is Webscrapping Wikipedia through Beautiful Soap.

* Beautiful Soap is a Python library that is used for webscrapping purposes to pull the data out of HTML and XML files

SPACE X API



SPACE X REST API

API RETURNS SPACE X DATASET IN JASON FORMAT

NORMALIZE DATA TO TRANSFORM IT INTO A .CSV FILE

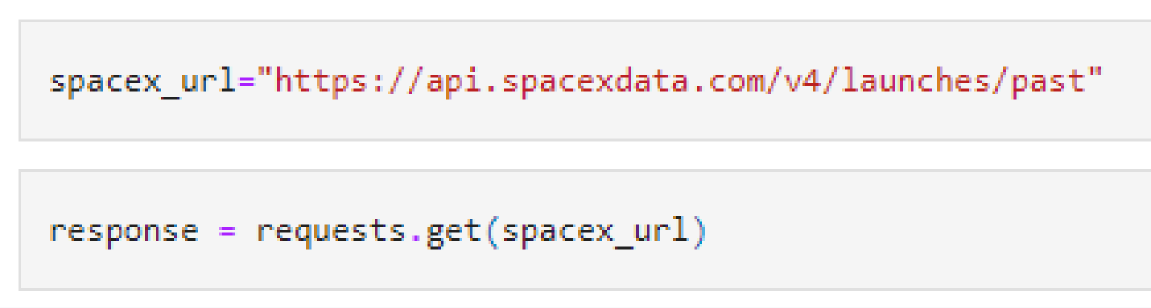
GET HTML OR XML RESPONSE FROM A WEBSITE

USE BEAUTIFUL SOUP LIBRARY TO EXTRACT THE DATA

NORMALIZE DATA TO TRANSFORM IT INTO A .CSV FILE

Data Collection – SpaceX API

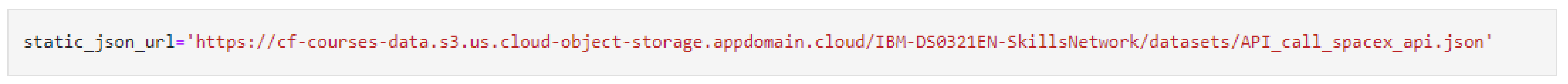
1. Getting responde from the URL



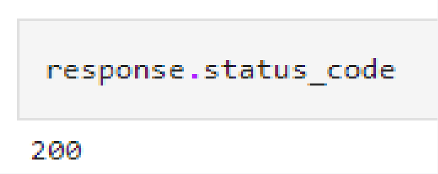
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

2. Converting to a JSON file

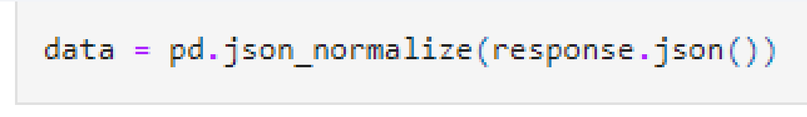


```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```



```
response.status_code
```

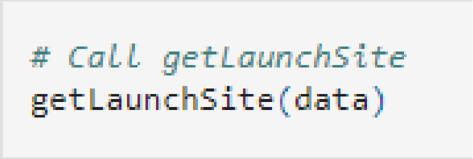
```
200
```



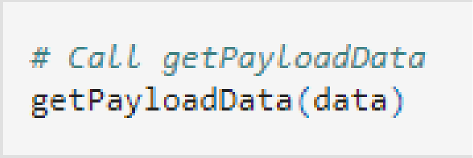
```
data = pd.json_normalize(response.json())
```

https://github.com/Tacyara/Capstone/blob/main/API%20with%20webscraping.ipynb

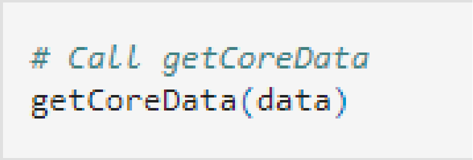
3. Apply custom functions to clean the dataset



```
# Call getLaunchSite
getLaunchSite(data)
```

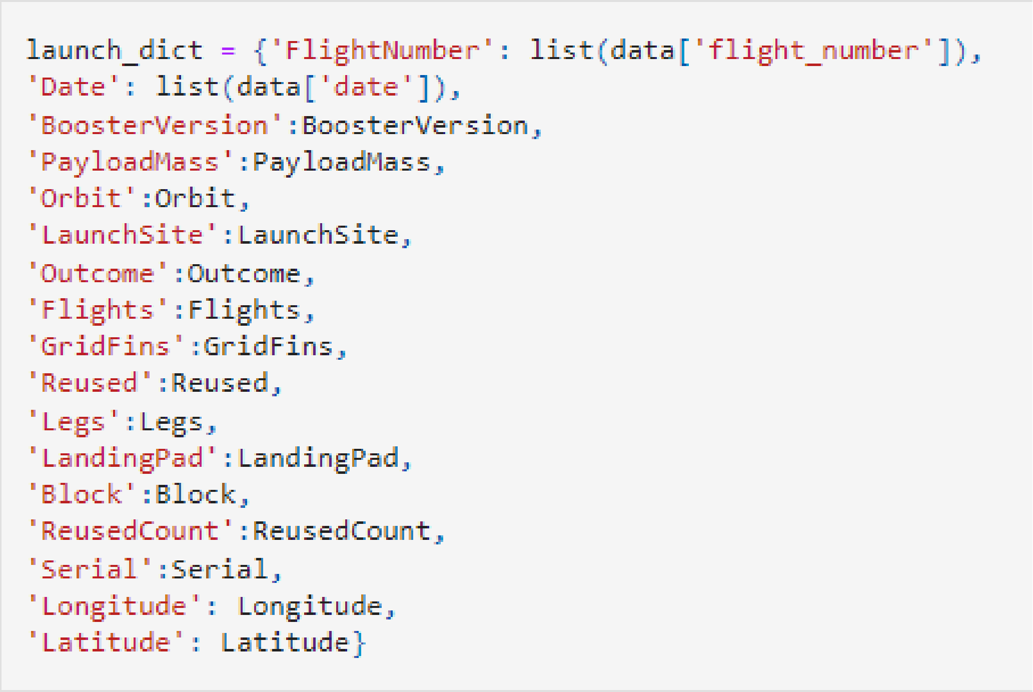


```
# Call getPayloadData
getPayloadData(data)
```



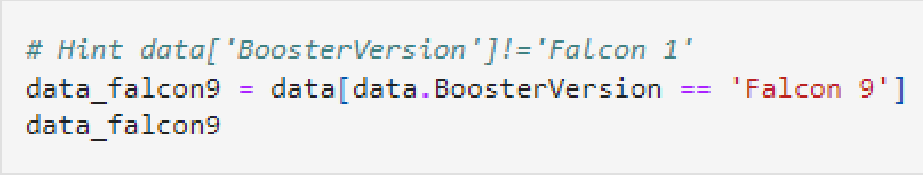
```
# Call getCoreData
getCoreData(data)
```

4. Assign list to dictionary then dataframe

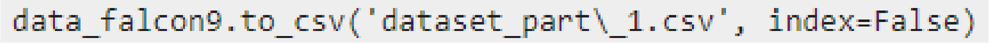


```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Filter dataframe and export to flat file(.csv)

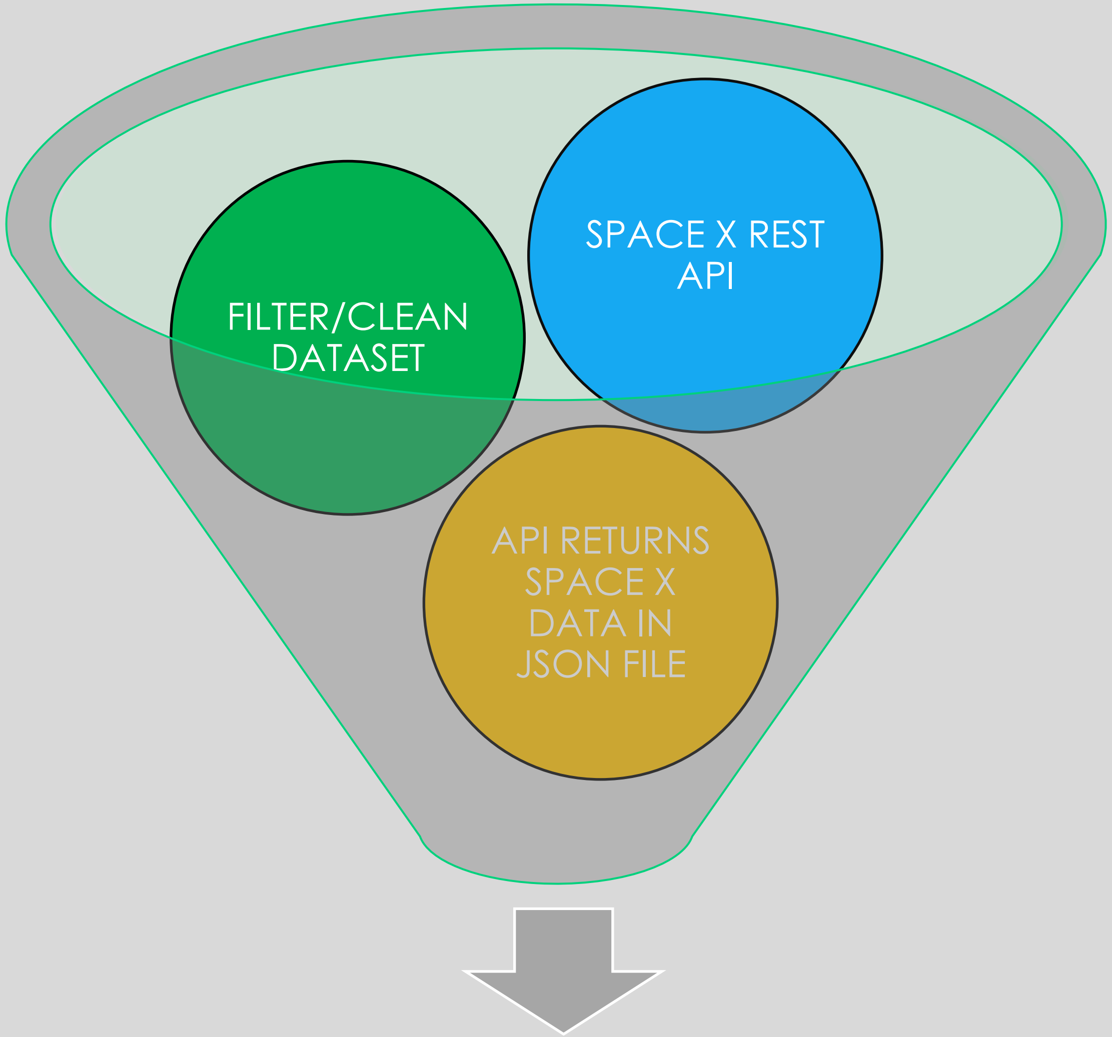


```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
data_falcon9
```



```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

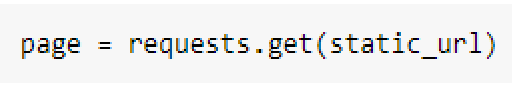
DATA COLLECTION SPACE X API



NORMALIZE DATA TO TRANSFORM IT INTO A .CSV FILE

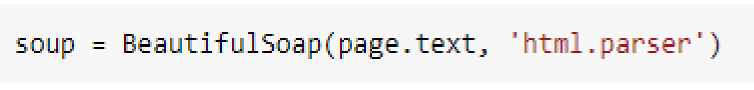
Data Collection - Scraping

1. Getting response from XML or HTML



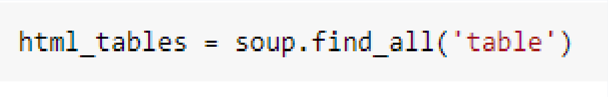
```
page = requests.get(static_url)
```

2. Creating Beautiful Soap object



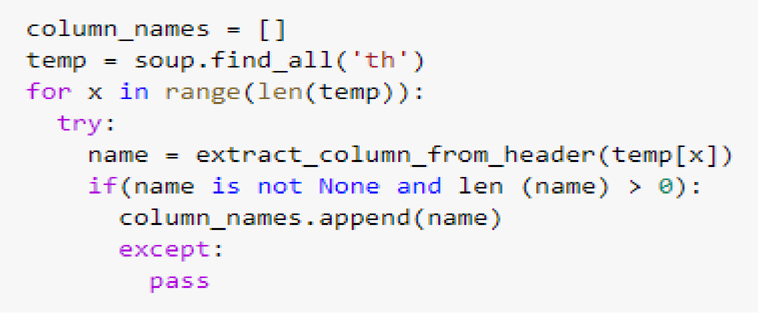
```
soup = BeautifulSoap(page.text, 'html.parser')
```

3. Finding Tables



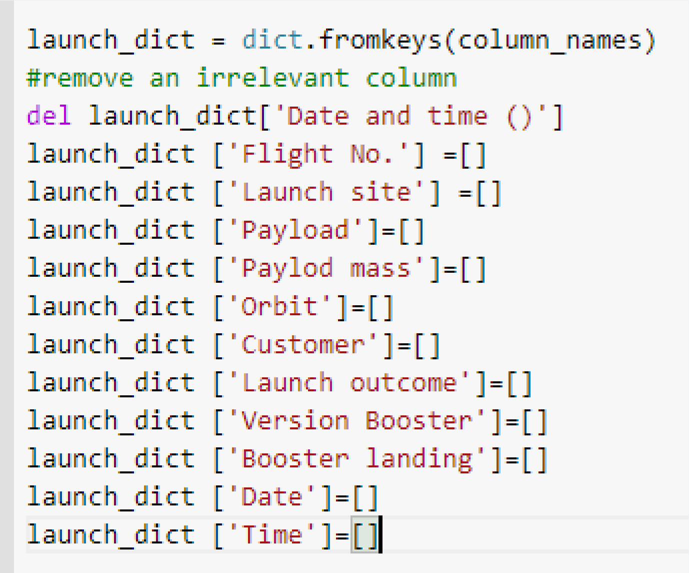
```
html_tables = soup.find_all('table')
```

4. Getting Column Names



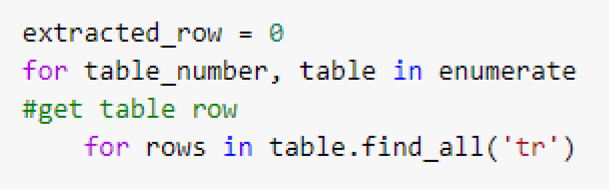
```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
  try:
    name = extract_column_from_header(temp[x])
    if(name is not None and len (name) > 0):
      column_names.append(name)
      except:
        pass
```

5. Creating a Dictionary



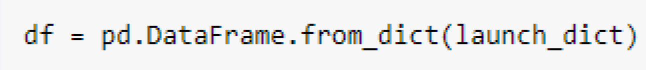
```
launch_dict = dict.fromkeys(column_names)
#remove an irrelevant column
del launch_dict['Date and time ()']
launch_dict ['Flight No.'] =[]
launch_dict ['Launch site'] =[]
launch_dict ['Payload']=[]
launch_dict ['Paylod mass']=[]
launch_dict ['Orbit']=[]
launch_dict ['Customer']=[]
launch_dict ['Launch outcome']=[]
launch_dict ['Version Booster']=[]
launch_dict ['Booster landing']=[]
launch_dict ['Date']=[]
launch_dict ['Time']=[]
```

6. Appending data to keys



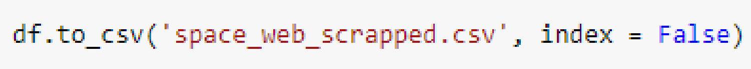
```
extracted_row = 0
for table_number, table in enumerate
#get table row
    for rows in table.find_all('tr')
```

7. Converting dictionary to dataframe



```
df = pd.DataFrame.from_dict(launch_dict)
```

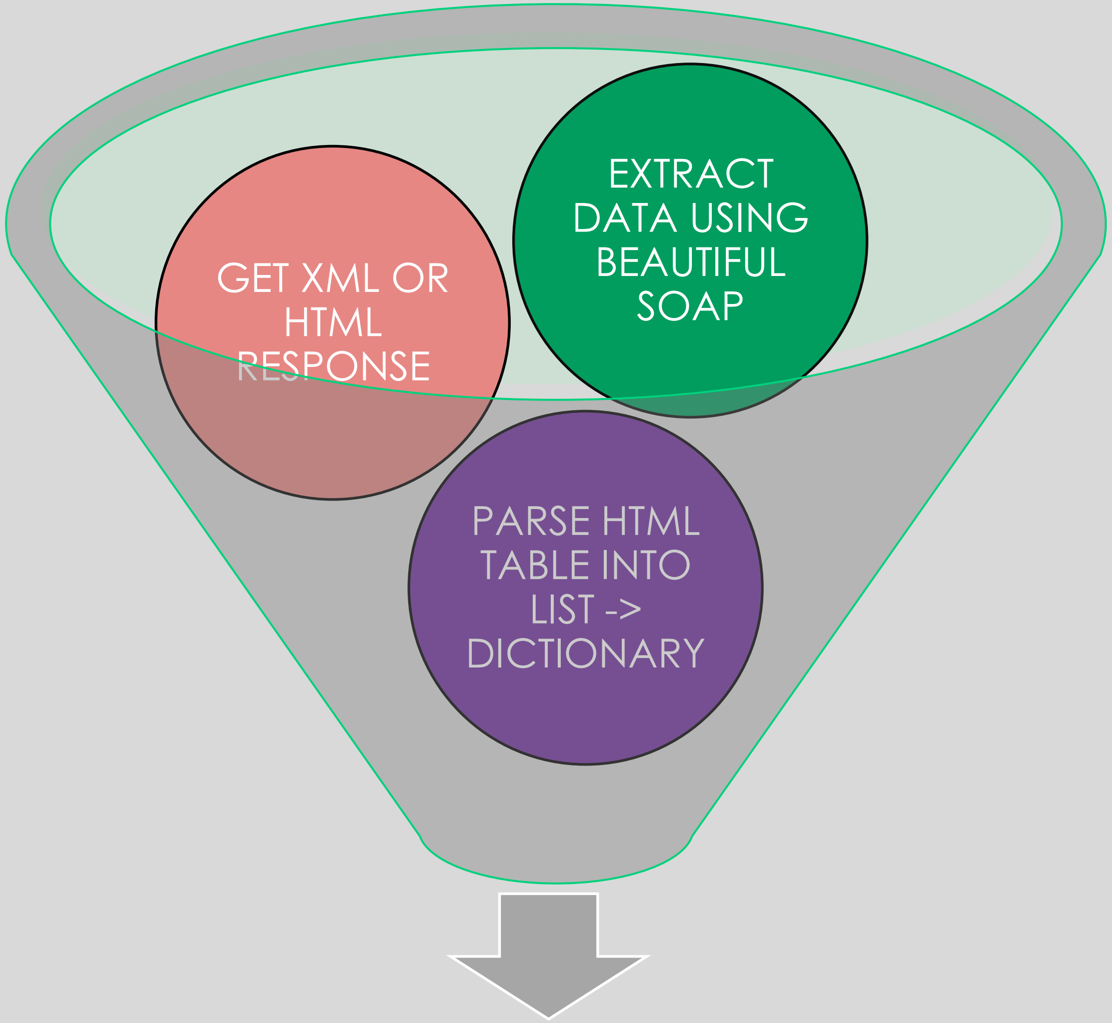
8. Finally: Dataframe to .CSV



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

https://github.com/Tacyara/Capstone/blob/master/EDA.ipynb

DATA COLLECTION WEBSCRAPPING

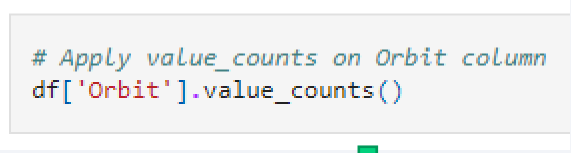

GET XML OR HTML RESPONSE
EXTRACT DATA USING BEAUTIFUL SOAP
PARSE HTML TABLE INTO LIST -> DICTIONARY

NORMALIZE DATA TO TRANSFORM IT INTO A .CSV FILE

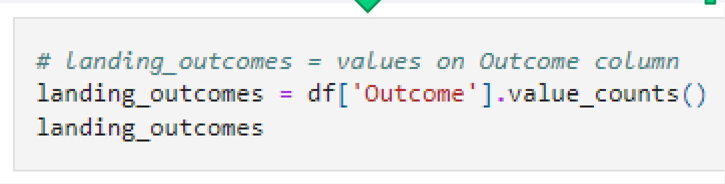
Data Wrangling

- Describe how data were processed

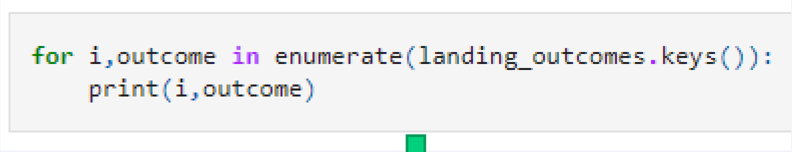
- Parse data type before any manipulation
- Change data type if necessary
- Check if there is null values
- Treat null values
- Structure de dataset to bring out the best approach for analytics exploration
- Use tools like “value_counts()” to better understand your dataset.



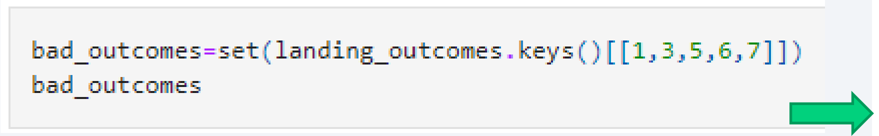
```python
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```



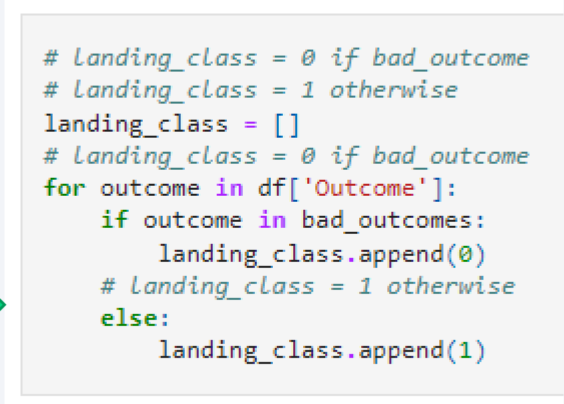
```python
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```



```python
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```



```python
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```



```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
# landing_class = 0 if bad_outcome
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    # landing_class = 1 otherwise
    else:
        landing_class.append(1)
```

https://github.com/Tacyara/Capstone/blob/master/EDA.ipynb

Each launch aims to a dedicated orbit, and here are some common orbit type

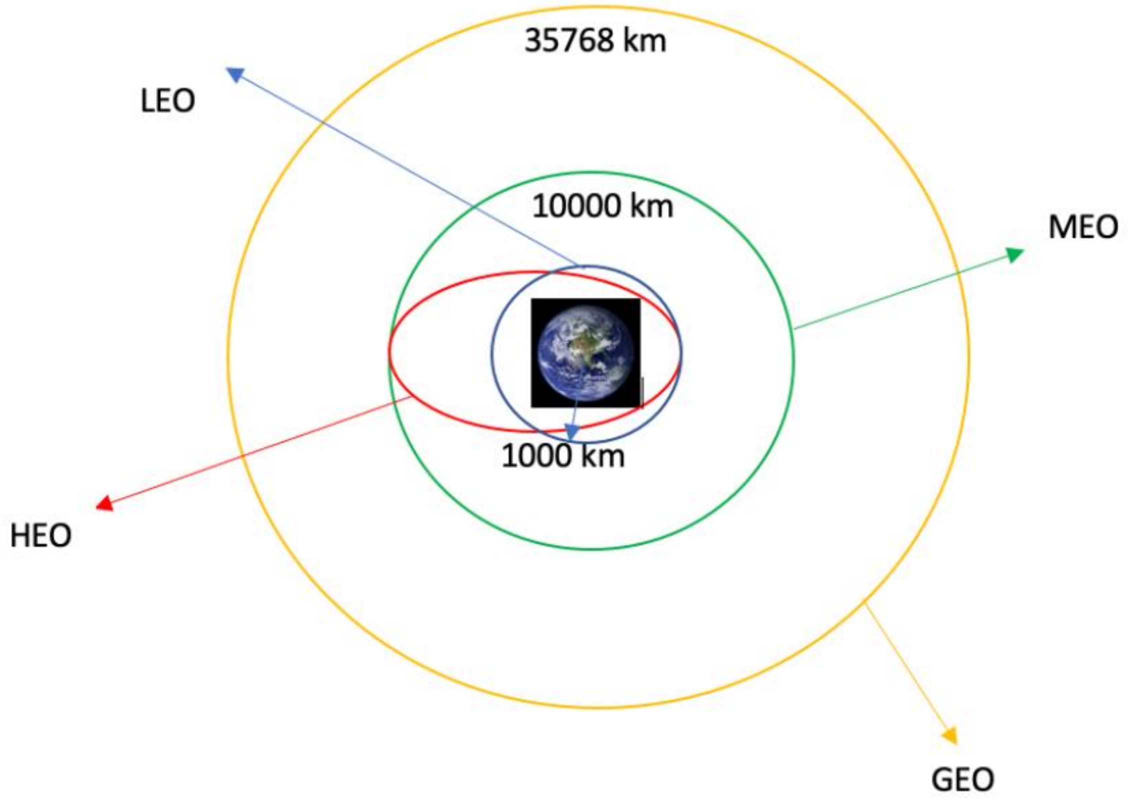
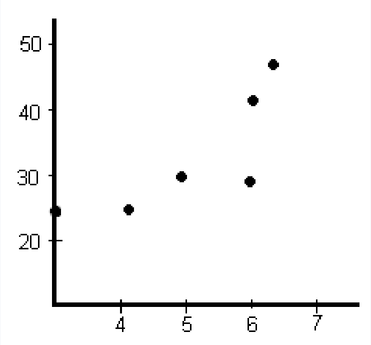


Diagram showing commom orbit types Space X uses

EDA with Data Visualization

SCATTER GRAPHS:



- Flight Number Vs. Payload Mass
- Flight Number Vs. Launch Site
- Payload Vs. Launch Site
- Orbit Vs. Flight Number
- Payload Vs. Orbit Type
- Orbit Vs. Payload Mass

* **SCATTER PLOTS** are representations of data from two or more variables that are arranged in a graph. It us used Cartesian coordinates to display values from the dataset. It shows how much one variable is affected by another.

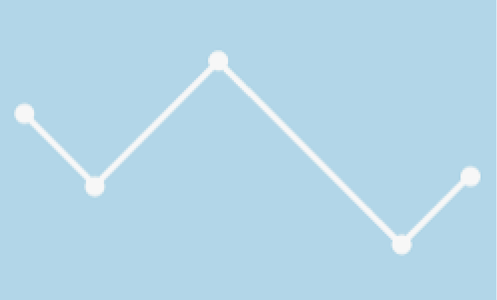
BAR GRAPHS:

- Mean Vs. Orbit

* **BAR GRAPHS** are used to compare things between diferente groups
or to track changes over time. It makes it easy to compare sets of data
Between diferente groups at a glance. The graph represents catergories on
one axis and a discrete value in the other. The goal is to represent the relation
between them.

https://github.com/Tacyara/Capstone/blob/master/EDA%20with%20Visualization%20lab.ipynb

EDA with Data Visualization



LINE GRAPH:

- Success rate Vs. Year

A **Line Graph** is a type of chart used to show information that changes over time. We plot **line graphs** using several points connected by straight lines.

https://github.com/Tacyara/Capstone/blob/master/EDA%20with%20Visualization%20lab.ipynb

EDA with SQL

SUMMARIZING THE SQL QUERIES PERFORMED IN THIS PROJECT:

- Displaying the names of the unique launch sites in the space mission;
- Displaying 5 records where launch sites begin with string ‘KSC’;
- Displaying the total payload mass carried by boosters launched by NASA (CRS);
- Displaying average payload mass carried by booster version F9 v1.1;
- Listing the date where the successful landing outcome in drone ship was achieved;
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 400 and less than 6000;
- Listing the total number of successful and failure mission outcomes;
- Listing the names of the boosters_versions which have carried the maximum payload mass;
- Listing the records wich will display the month names, successful landing_outcomes in ground pad, booster versions, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

https://github.com/Tacyara/Capstone/blob/master/EDA%20with%20SQL%20lab.ipynb

Build an Interactive Map with Folium

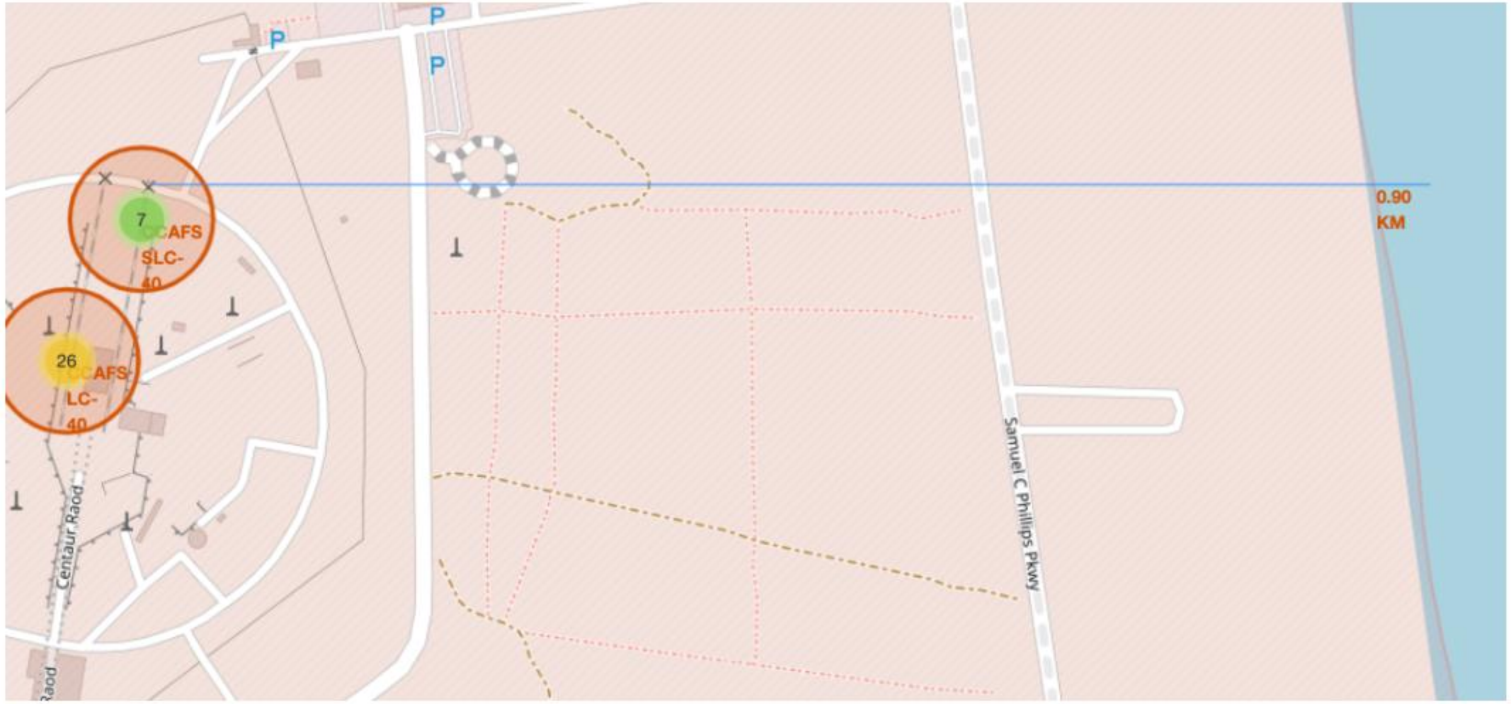
This map was built to visualize the Launch Data into an interactive map. For this stage, its is required the Latitude and Longitude coordinates at each launch site. It was also added a Circle Marker around each launch site with a label of the name of the launch site.

We assigned the dataframe launch_outcomes (failure / success) to classes 0 and 1 with Green and Red markers on the Map in a MarkerCluster()

- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

https://github.com/Tacyara/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build an Interactive Map with Folium



Some trends example:

- Are launch sites in close proximity to railways?

A: **No**

- Are launch sites in close proximity to highways?

A: **No**

- Do launch sites keep certain distance away from cities?

A: **Yes**

https://github.com/Tacyara/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium

Predictive Analysis (Classification)

BUILDING MODEL

- Load the dataset into NumPy and Pandas libraries
- Transform data
- Split the dataset – train (X_train, Y_train) and test (X_test and Y_test)
- Check the test sample to validate the proportion
- Make data balancing if necessary
- Set the parameters and algorithms to Grid SearchCV
- Fit the dataset into GridSearch CV objects and train the dataset

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins the best performing model
- In the bottom of the notebook there is a dictionary of algorithms with scores.

https://github.com/Tacyara/Capstone/blob/master/Machine%20Learning%20Prediction%20lab.ipynb

Results

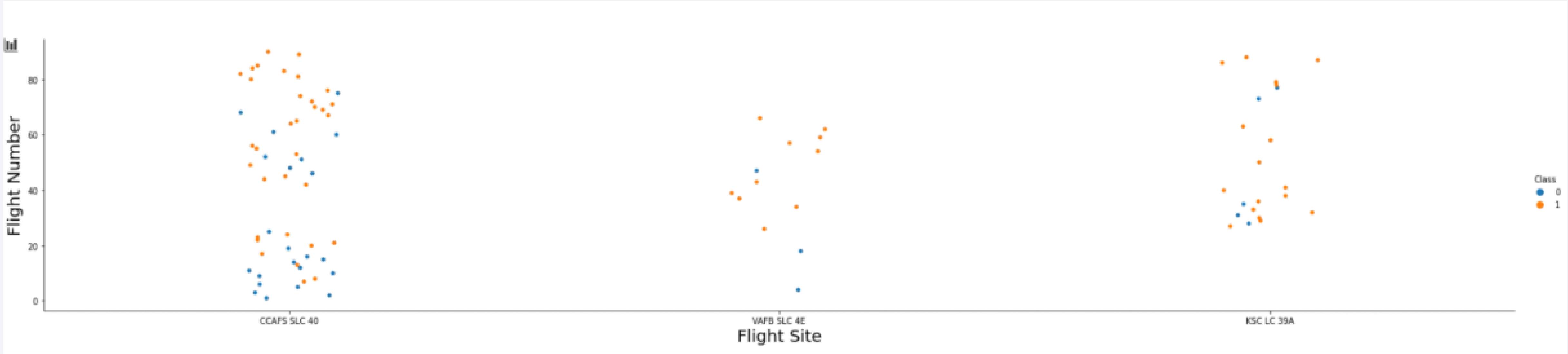
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

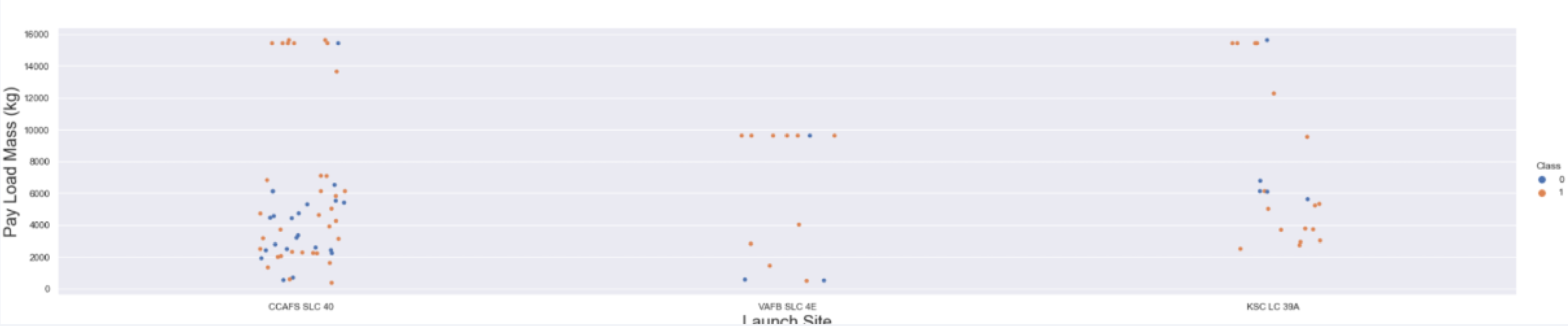
SCATTER PLOT OF FLIGHT NUMBER VS. LAUNCH SITE



Conclusion: the greater the number of launches, the greater the success rate

Payload vs. Launch Site

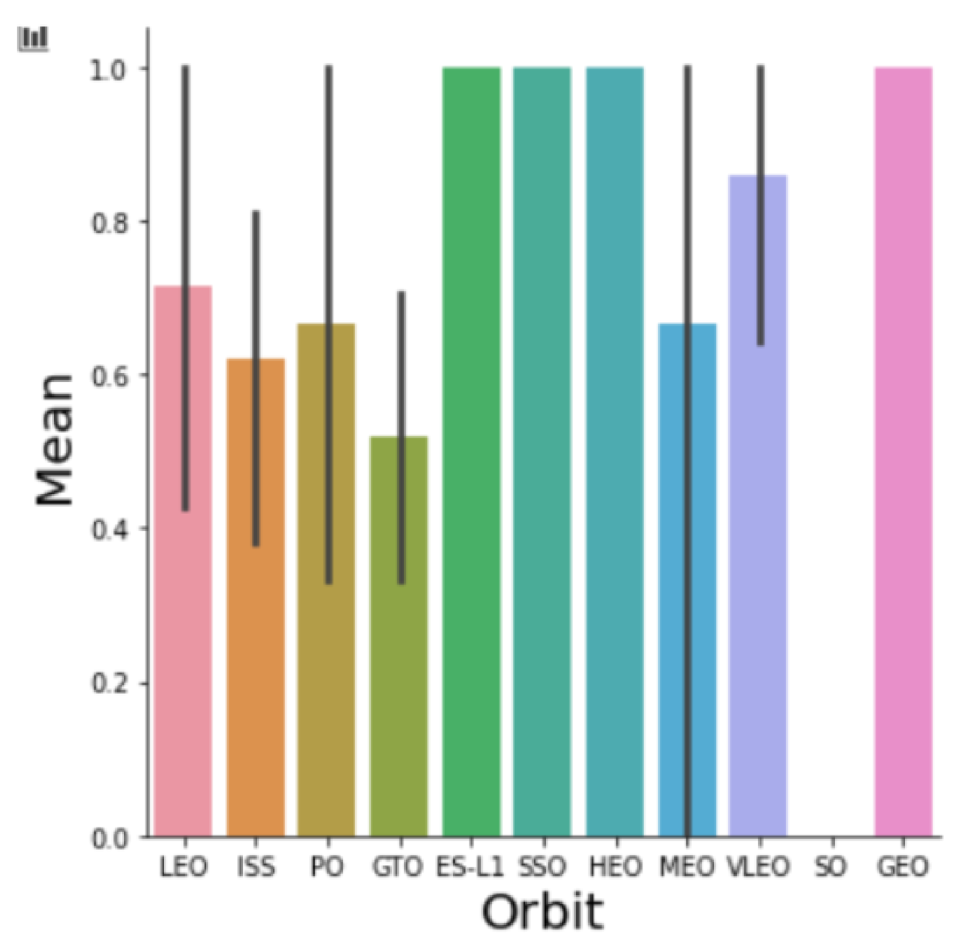
SCATTER PLOT OF PAYLOAD VS. LAUNCH SITE



Conclusion: The greater the payload mass for launch site CCAFS SLC 40, the greater the success rate fo the Rocket.

Success Rate vs. Orbit Type

BAR CHART FOR THE SUCCESS RATE OF EACH ORBIT TYPE

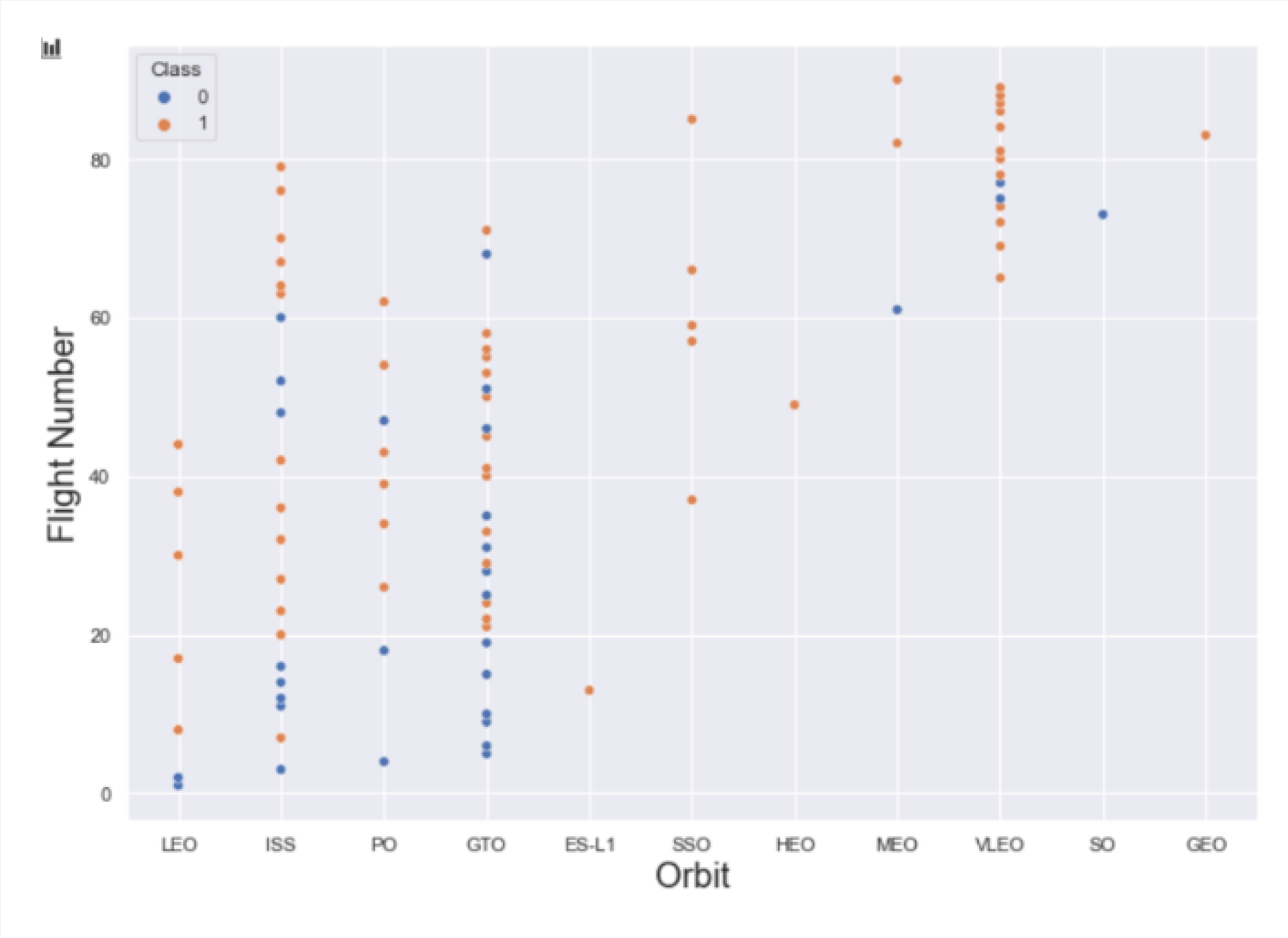


Conclusion: Orbit GEO, HEO, SSO, ES-L1 has the higher success rate

Flight Number vs. Orbit Type

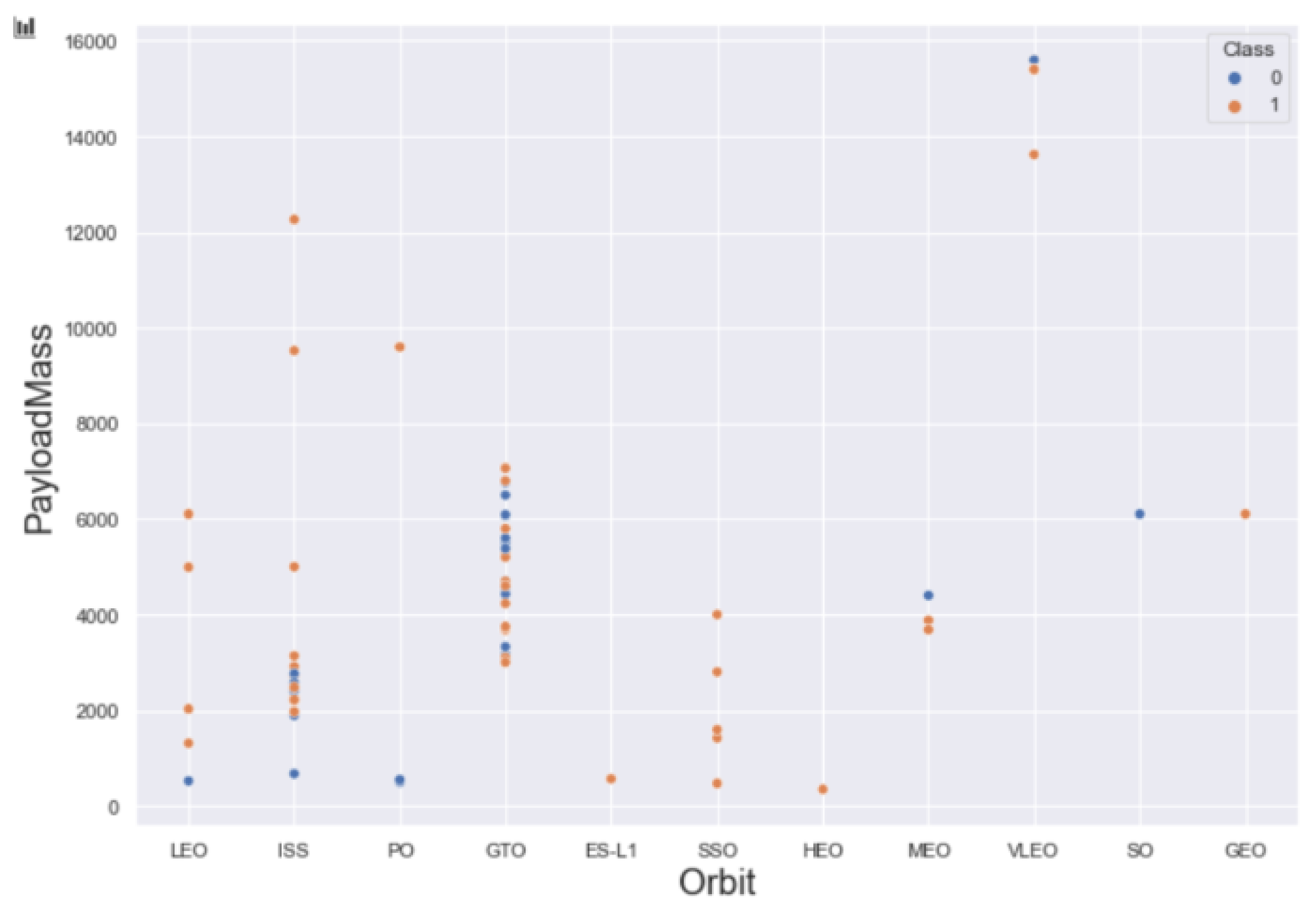
SCATTER POINT OF FLIGHT NUMBER VS. ORBIT TYPE

Conclusion: At LEO orbit, the success appears to be related to the numbers of flights.
On the other hand, it seems that there is no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

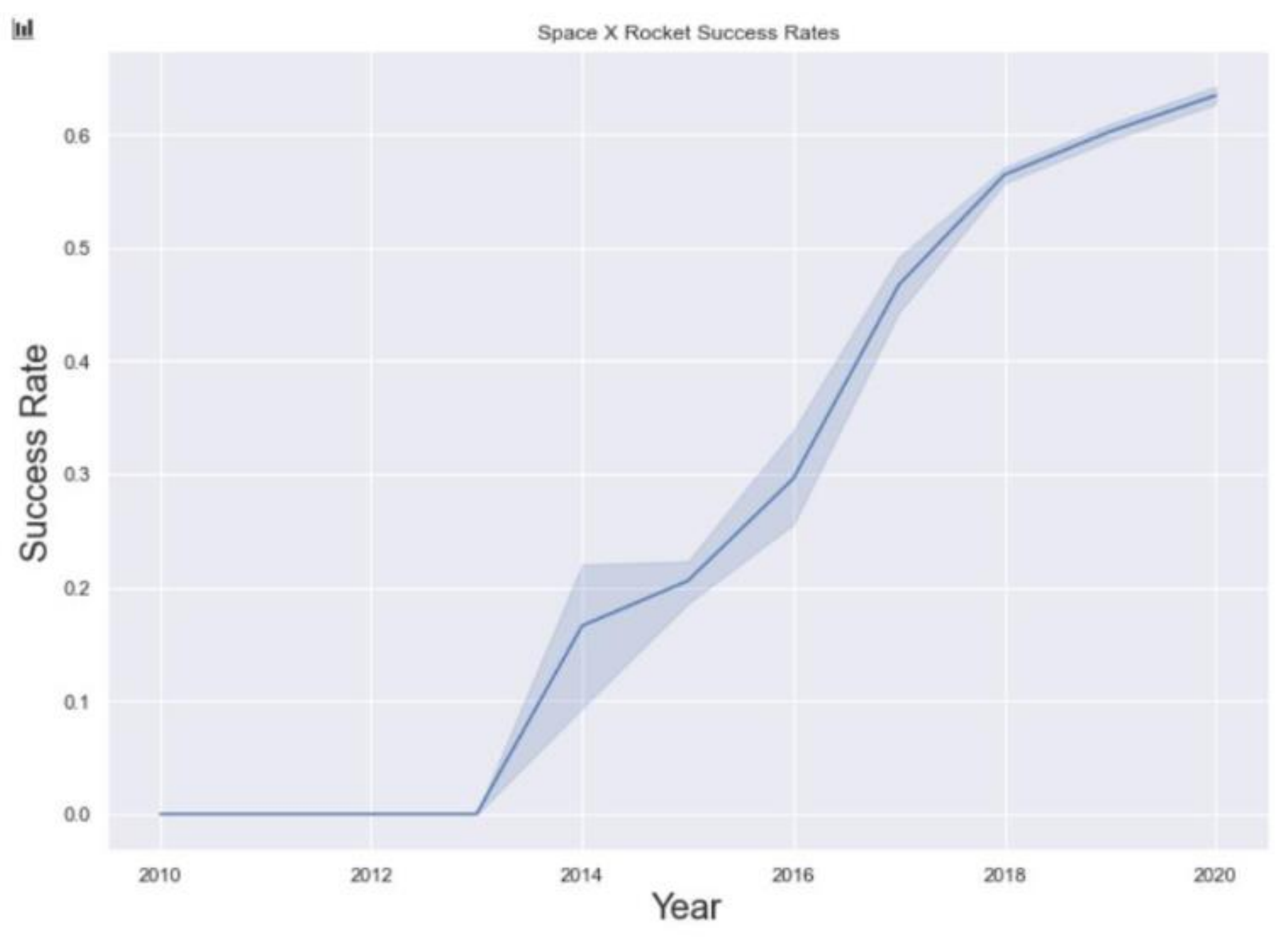
SCATTER POINT OF PAYLOAD VS. ORBIT TYPE



Conclusion: Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

LINE CHART OF YEARLY AVERAGE SUCCESS RATE



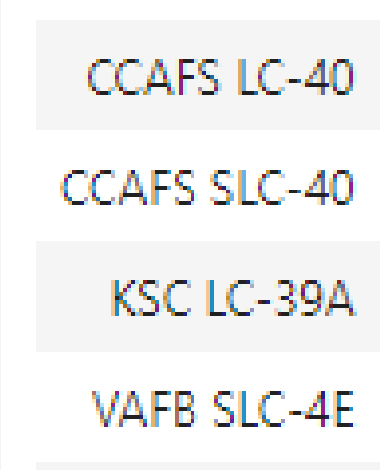
Conclusion: The success rate kept Increasing from 2013 to 2020.

All Launch Site Names

UNIQUE LAUNCH SITES

SQL QUERY

```
Select DISTINCT Launch_Site from tblSpaceX
```

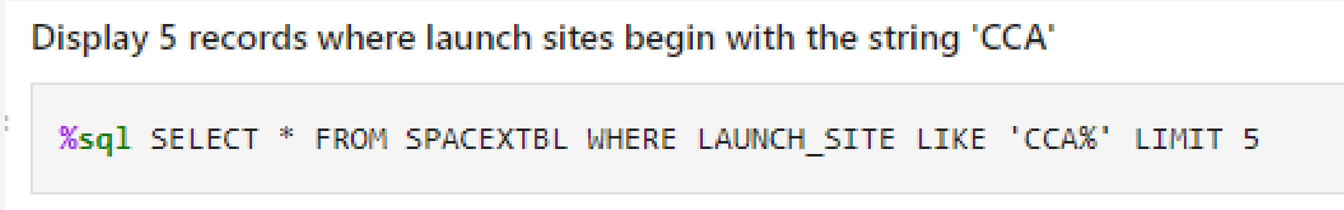


```
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E
```

Explanation: When the word ‘DISTINCT’ is applied to a query, it will return only unique values. In this case, it returned unique values in the Launch_Site column from tblSpaceX

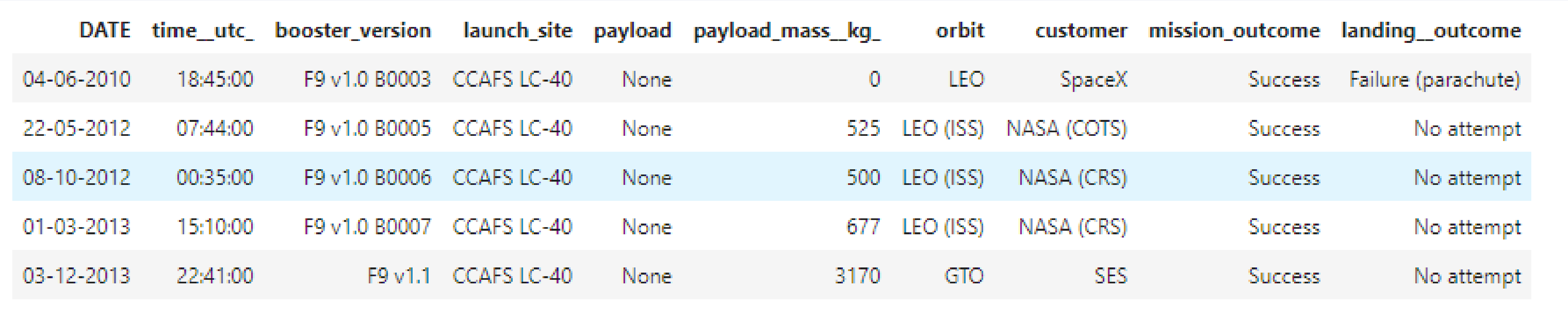
Launch Site Names Begin with 'CCA'

5 records where launch sites 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
```

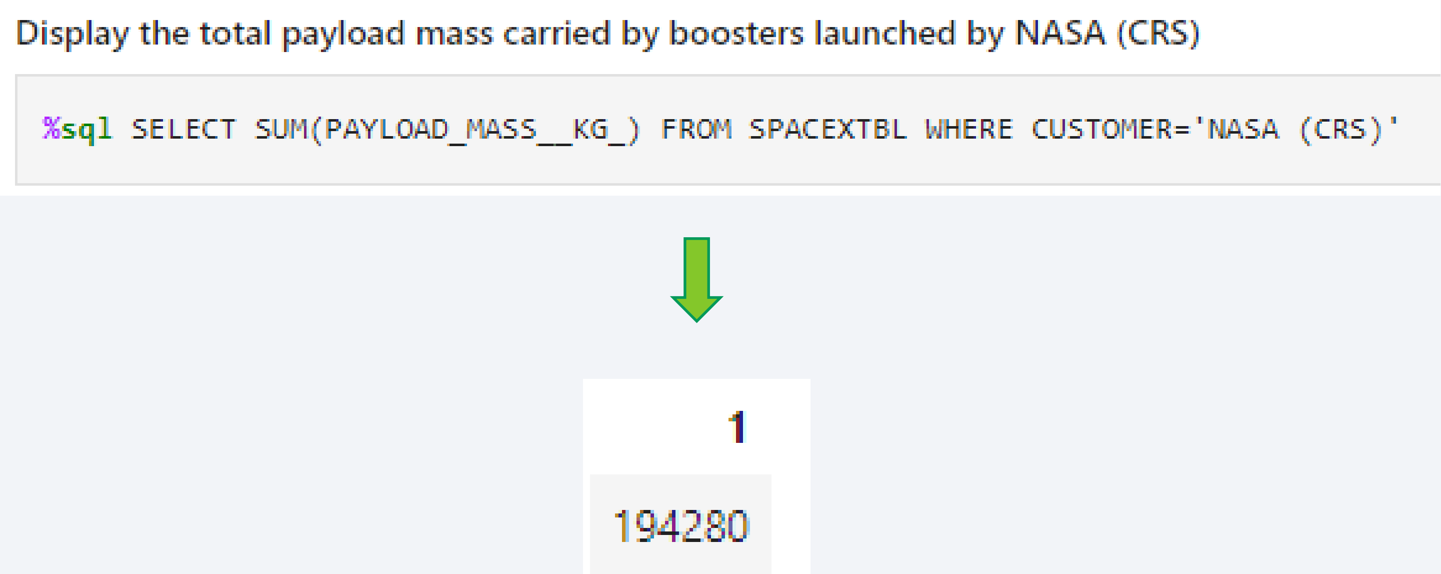


DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	None	0	LEO	SpaceX	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt
03-12-2013	22:41:00	F9 v1.1	CCAFS LC-40	None	3170	GTO	SES	Success	No attempt

Explaining the query: By using the words **‘LIMIT 5’** and **‘LIKE’**, it will return only 5 results according to this Keyword. The **‘%CCA’** (percentage) means that the Launch_Site must star begin with ‘CCA’.

Total Payload Mass

THE TOTAL PAYLOAD CARRIED BY BOOSTERS FROM NASA



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

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

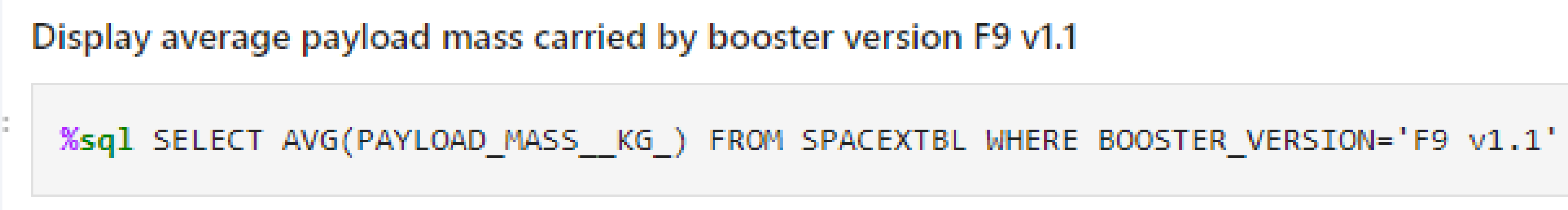
```
     1
194280
```

Explaining the query: By using the word **'SUM'**, it returns the total in the column PAYLOAD_MASS_KG_

The **'WHERE'** word clause filters the dataset to only perform calculations on Customer NASA (CRS)

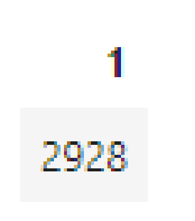
Average Payload Mass by F9 v1.1

CALCULATE THE AVERAGE PAYLOAD MASS CARRIED BY BOOSTER VERSION 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='F9 v1.1'
```

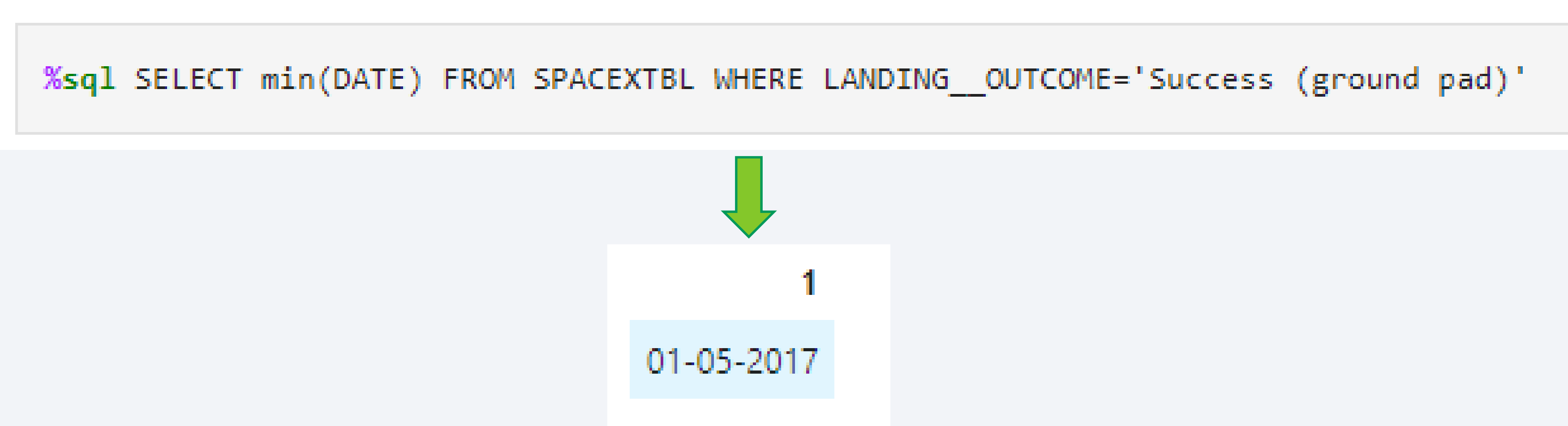


```
   1
2928
```

Explaining the query: The function ‘**AVG**’ works out the average in the column PAYLOAD_MASS_KG_. The word ‘**WHERE**’ filters the dataset to only perform calculations on Booster_version F9 v1.1

First Successful Ground Landing Date

The date of the first successful landing outcome on ground pad



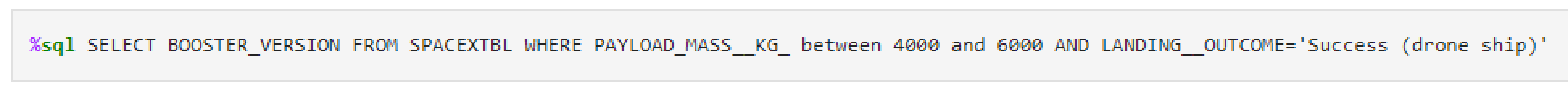
```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
         1
01-05-2017
```

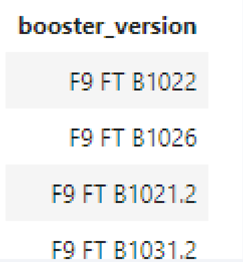
Explaining the query: The function **'MIN'** returns the minimum date in the column 'Date'
The word **'WHERE'** filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

Successful Drone Ship Landing with Payload between 4000 and 6000

THE NAMES OF BOOSTERS WHICH HAVE SUCCESSFULLY LANDED ON DRONE SHIP AND HAD PAYLOAD MASS GREATER THAN 4000 BUT LESS THAN 6000



```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

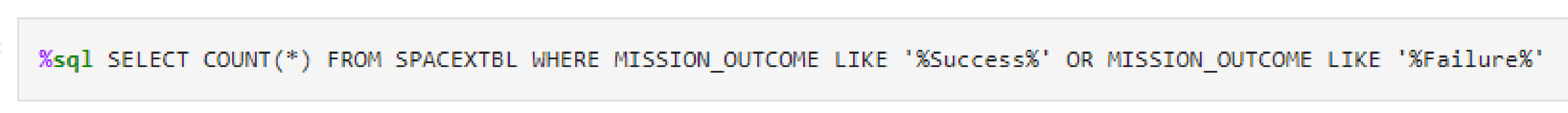


booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

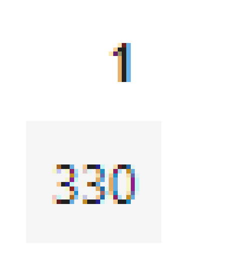
Explaining the query: ‘WHERE’ word filters the dataset to Landing_oucome = success (drone ship)
‘AND’ word specifies additional filter conditions like: PAYLOAD_MASS_KG_>400 AND PAYLOAD_MASS_KG_< 6000

Total Number of Successful and Failure Mission Outcomes

CALCULATE THE TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES



```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

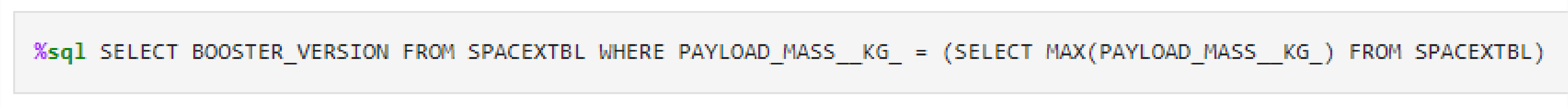


```
   1
 330
```

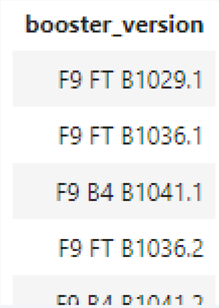
Explaining the query: ‘WHERE’ word filters the dataset to Missing_oucome = success (drone ship)
‘LIKE’ word specifies only ‘Success’ or only ‘Failure’ words.

Boosters Carried Maximum Payload

LIST THE NAMES OF THE BOOSTER WHICH HAVE CARRIED THE MAXIMUM PAYLOAD MASS



```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

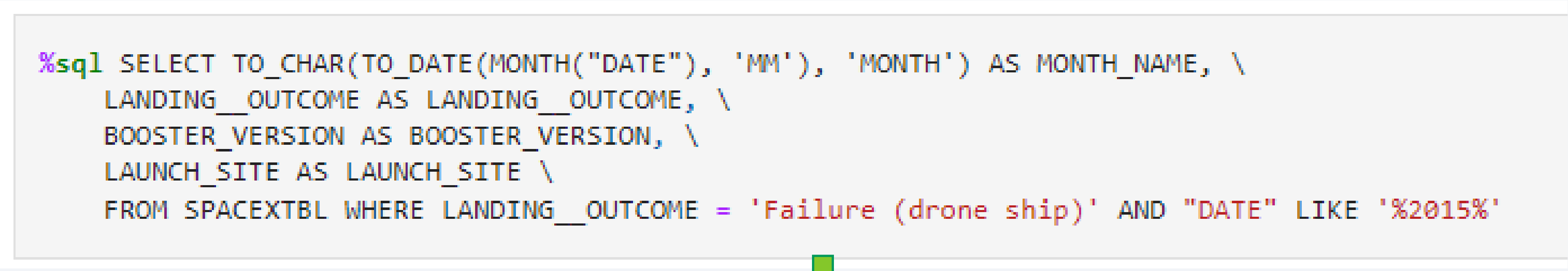


booster_version
F9 FT B1029.1
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1036.2
F9 B4 B1041.2

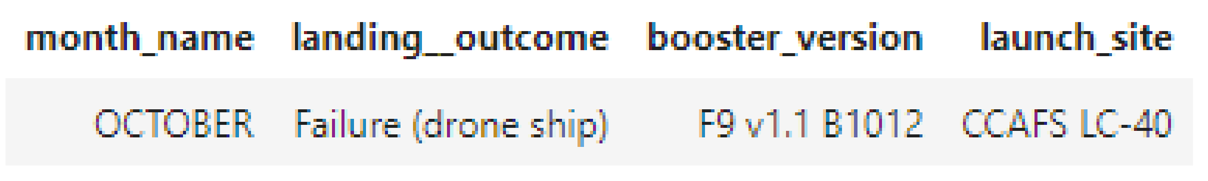
Explaining the query: The function **‘MAX’** returns the maximum date in the column ‘Date’
The word **‘WHERE’** filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

2015 Launch Records

THE FAILED LANDING_OUTCOMES IN DRONE SHIP, THEIR BOOSTER VERSIONS, AND LAUNCH SITE NAMES FOR IN YEAR 2015



```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

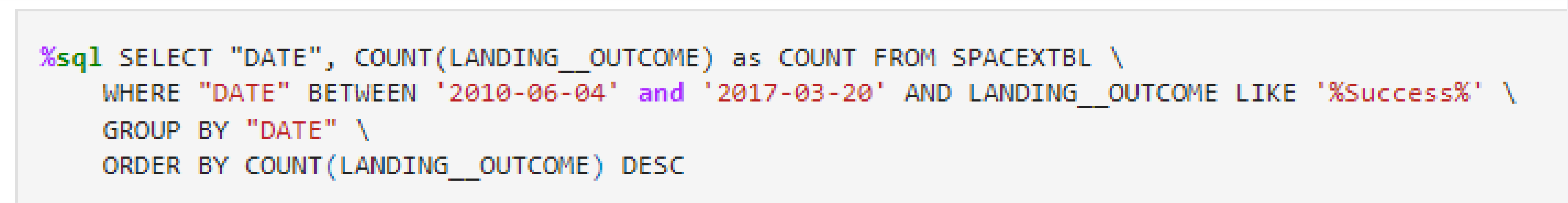


month_name	landing__outcome	booster_version	launch_site
OCTOBER	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

Explaining the query: ‘WHERE’ word filters the dataset to Missing_oucome = success (drone ship) **‘LIKE’** word specifies only ‘Success’ or only ‘Failure’ words.

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

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



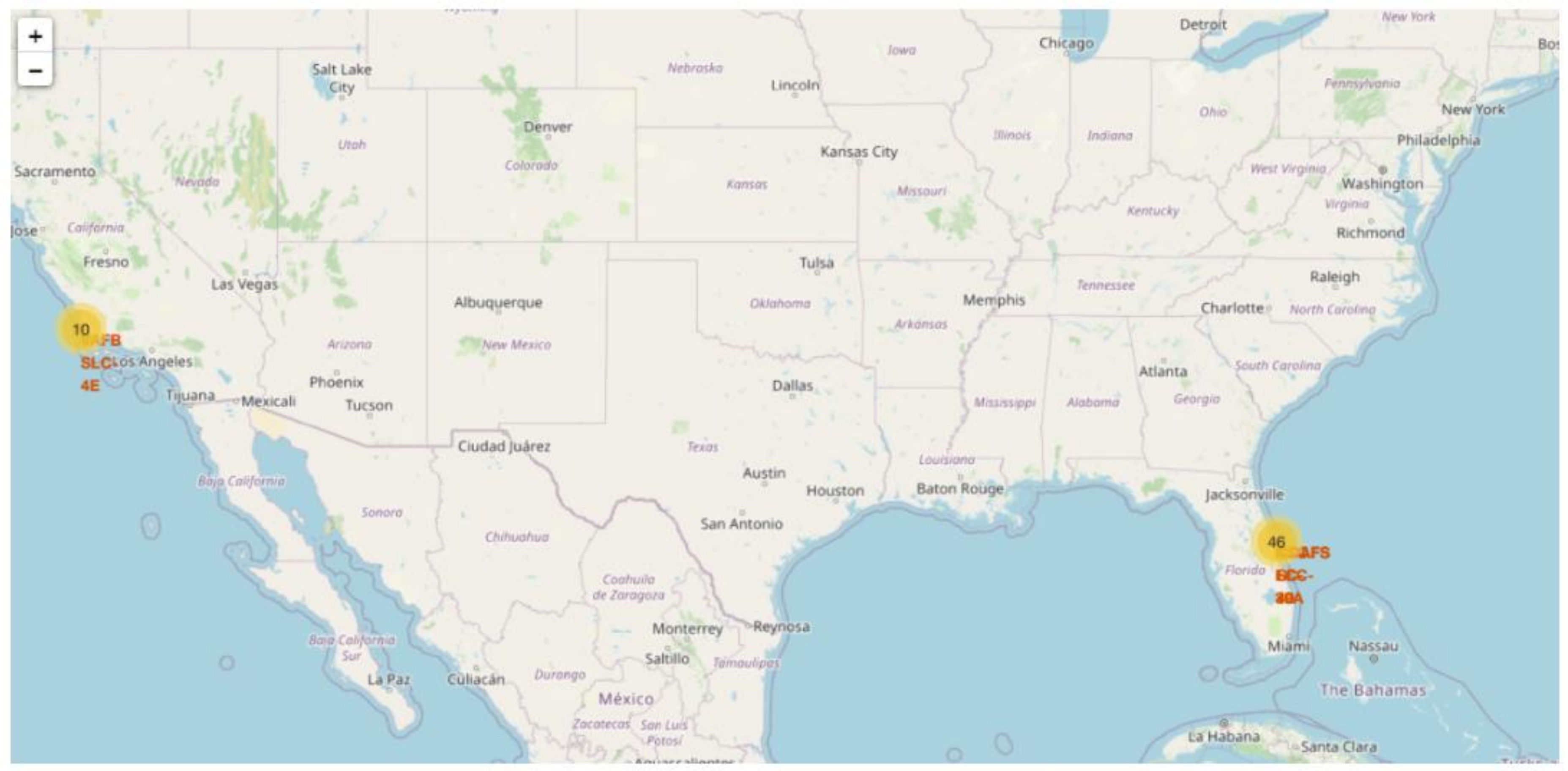
```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

Explaining the query:
'COUNT' counts records in a column;
'WHERE' word filters the dataset to 'Date';
'**BETWEEN**' filters a specific range;
'**LIKE**' word searches for a specific value
'GROUP' search in a row that have the same values into summary rows

Section 3

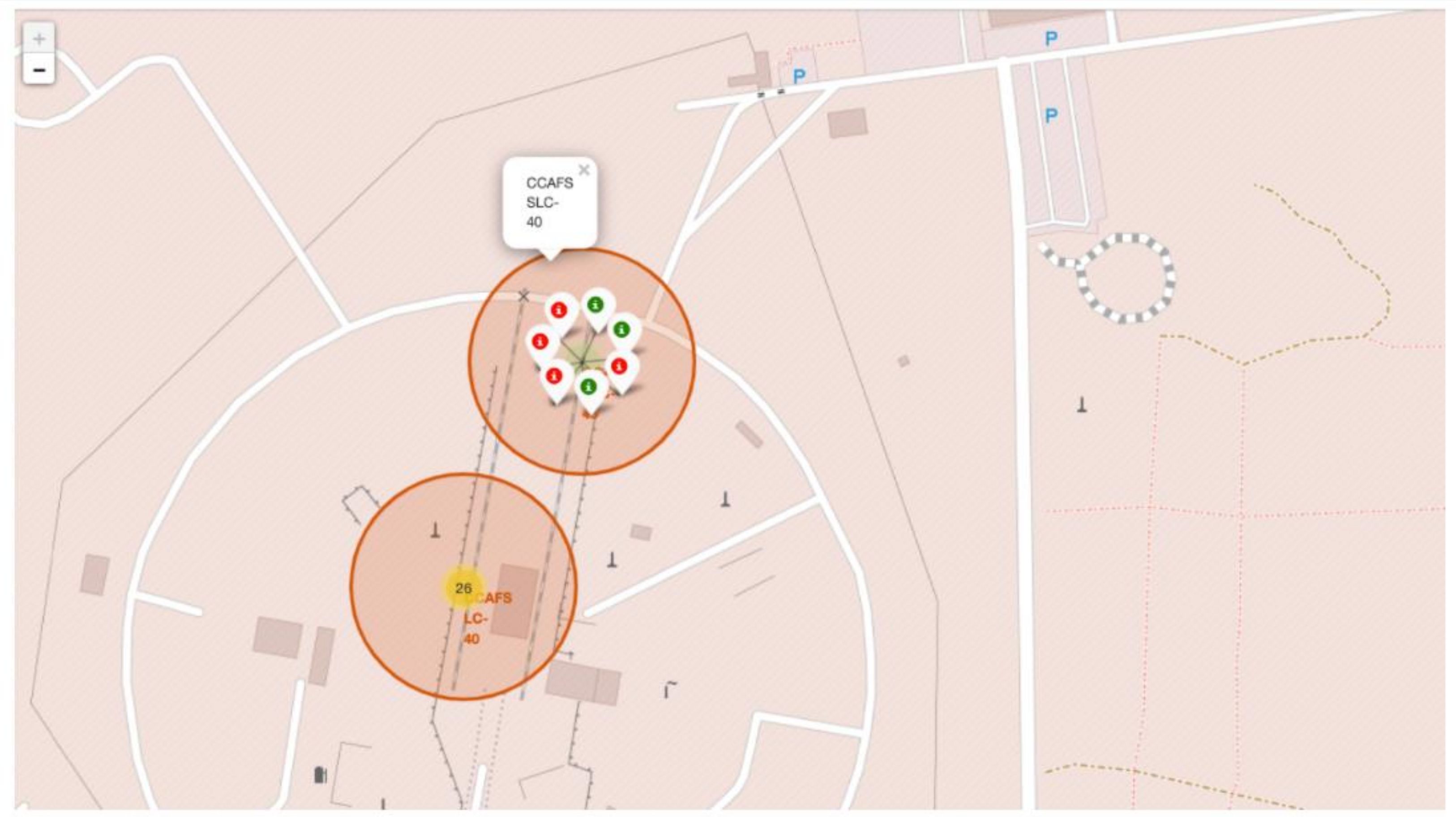
Launch Sites Proximities Analysis

LAUNCH SITES SPOTS



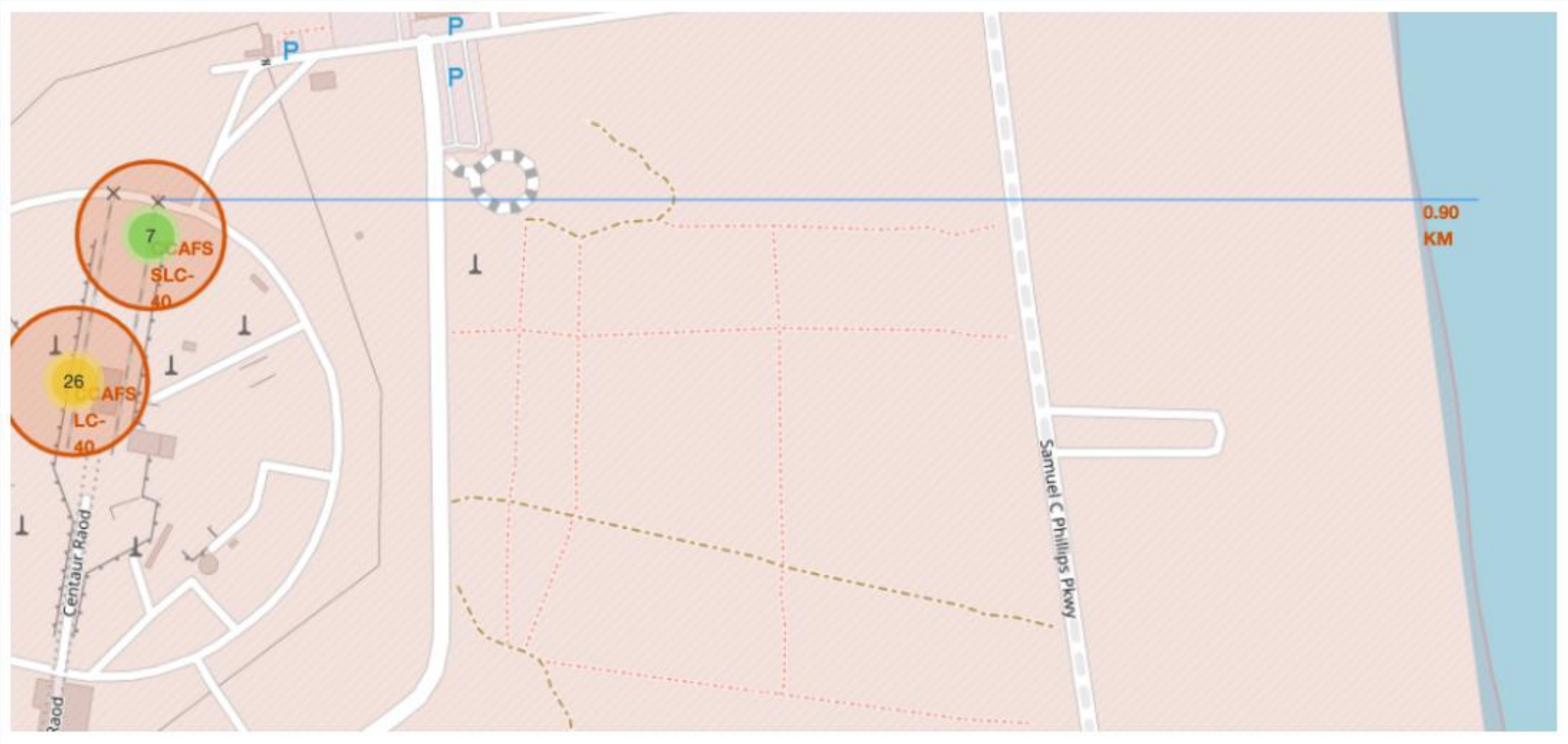
HIGHLIGHTS: Its possible to realize that the Space X launche spots are all located in the US coasts, speficially at Florida and California states.

SUCCESS OR FAILURE RATE - ROCKET LAUNCHES



HIGHLIGHTS: From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates. **RED** spots means 'failure' and **GREEN** spots means 'success'

LAUNCH SITES DISTANCE FROM LANDMARKS



- Are launch sites in close proximity to railways?

A: NO

- Are launch sites in close proximity to highways?

A: NO

- Are launch sites in close proximity to coastline?

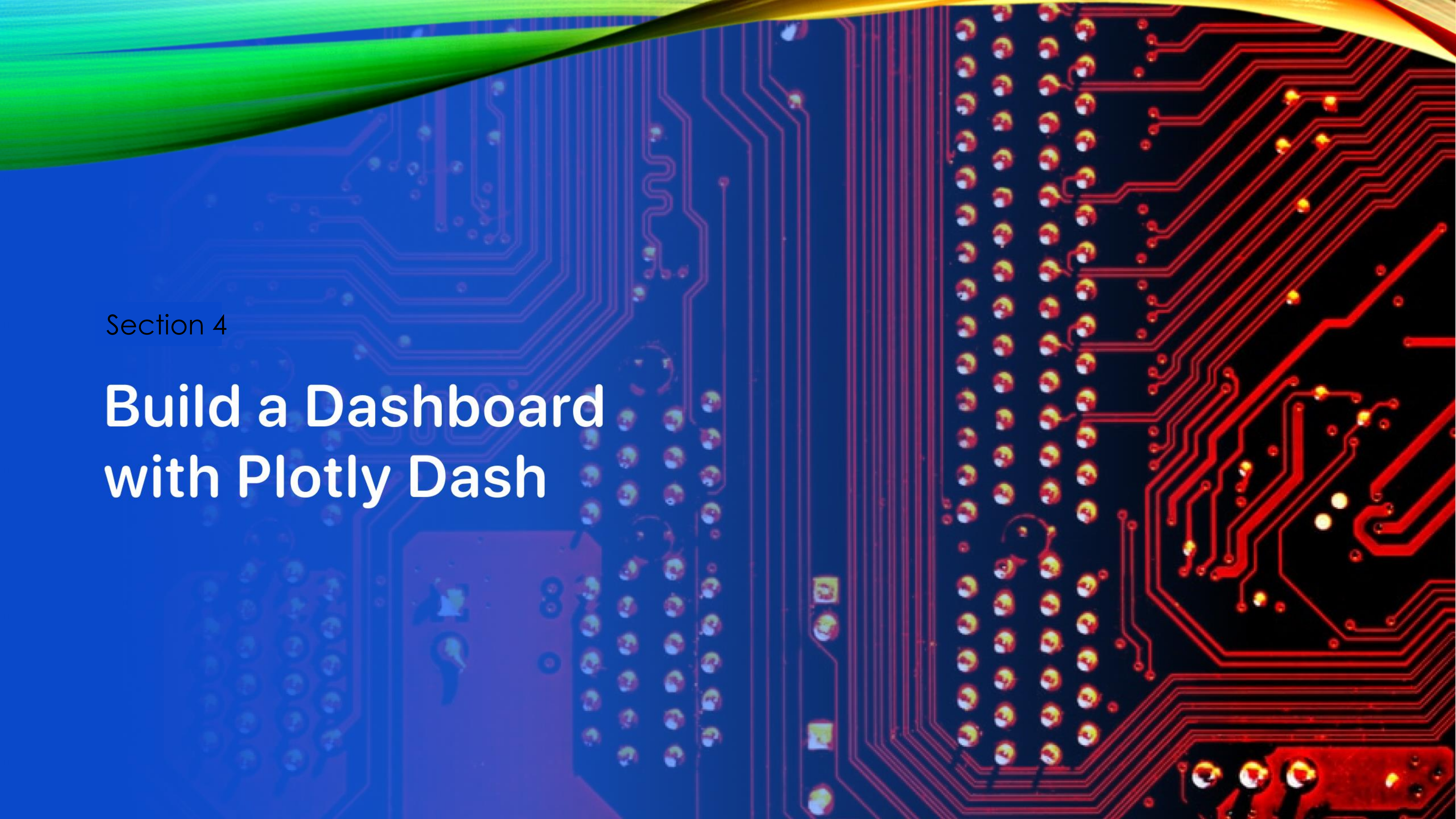
A: YES

- Do launch sites keep certain distance away from cities?

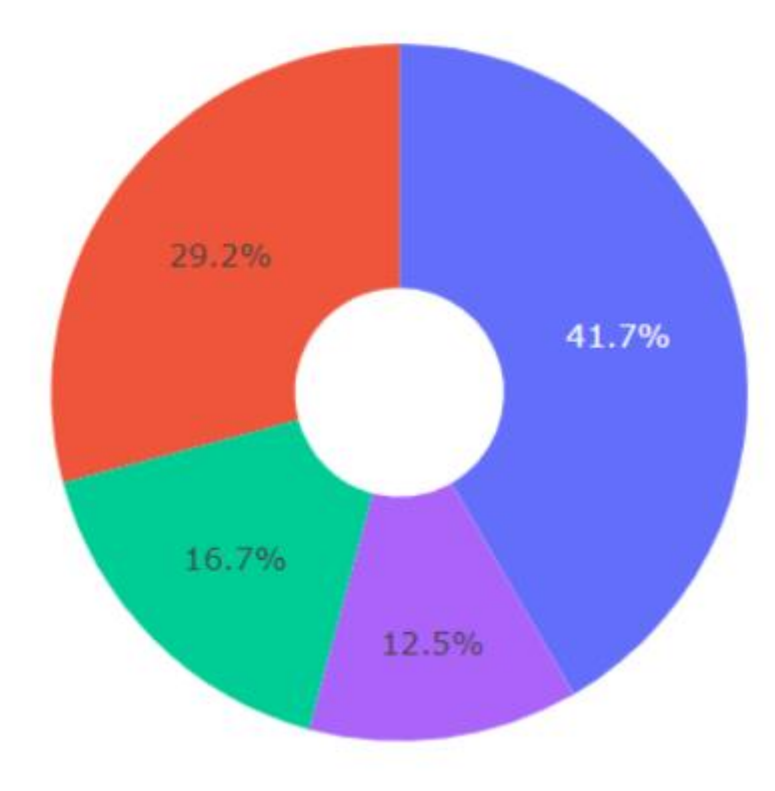
A: YES

Section 4

Build a Dashboard with Plotly Dash

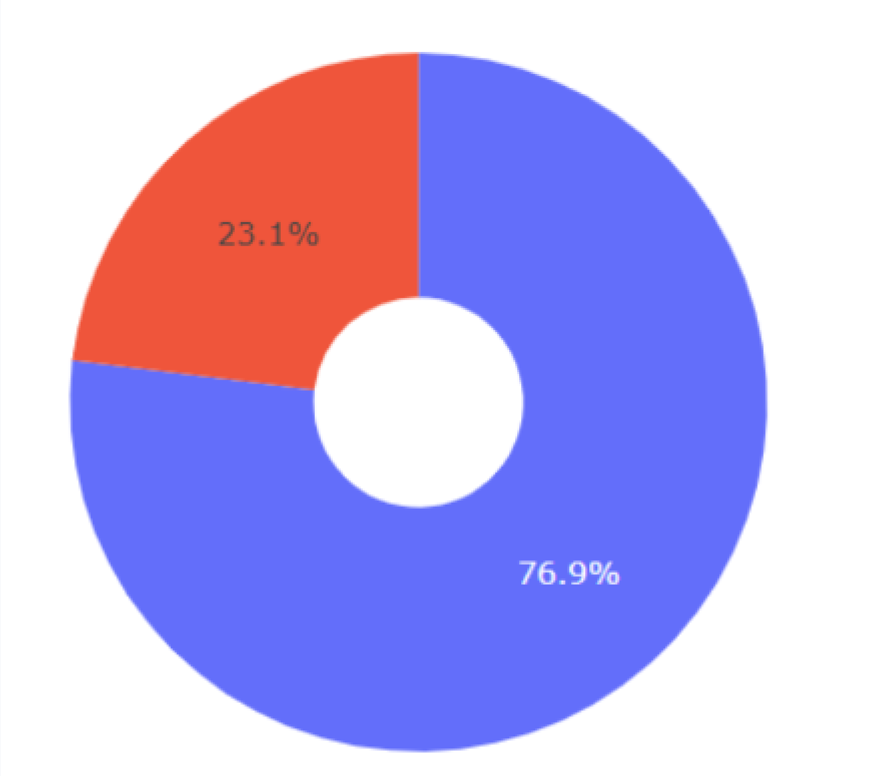


PIE CHART – SUCCESS RATE BY EACH LAUNCH SITE



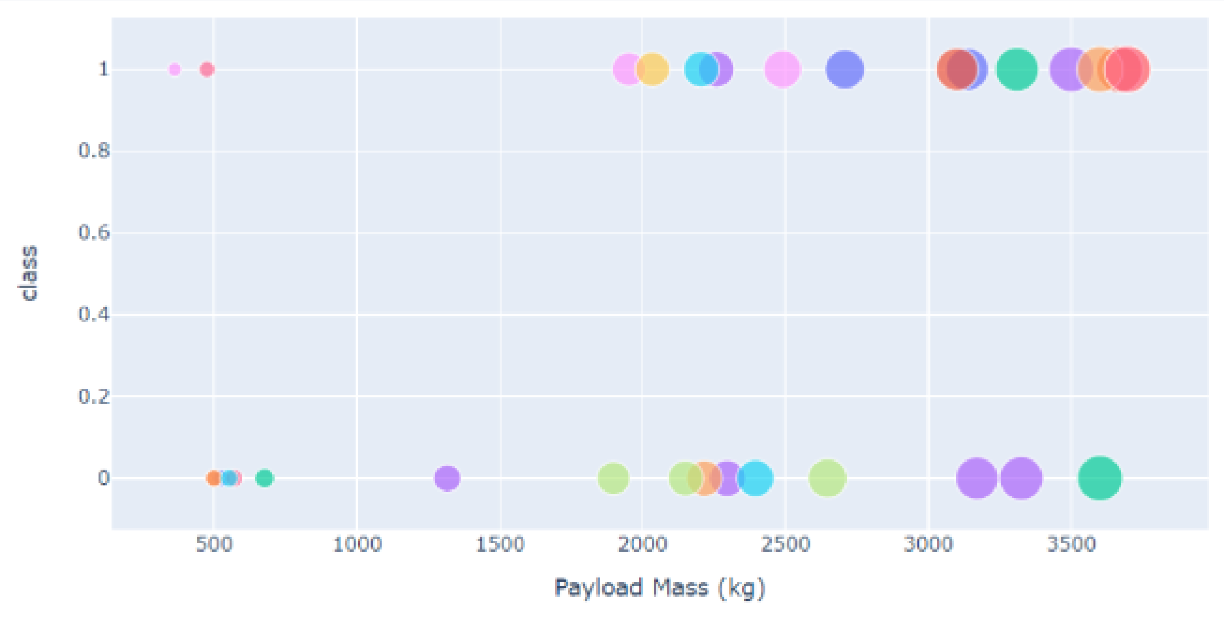
KSC LC-39 A representes the launch with the highest success rate, while **CCAFS SLC-40** representes the lowest

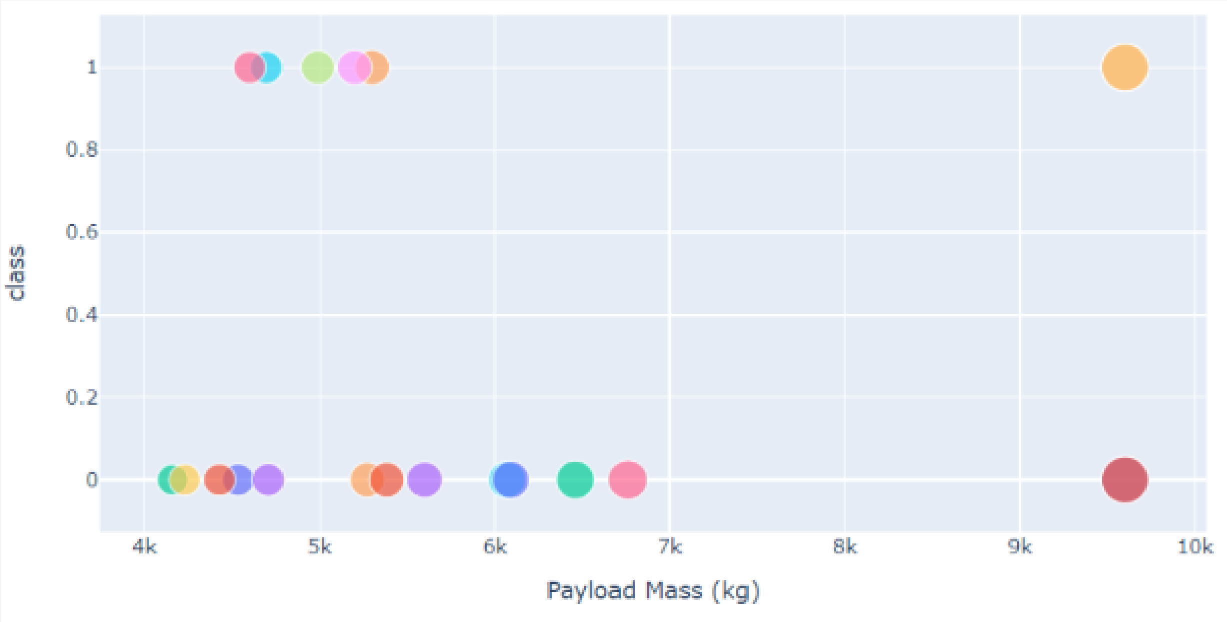
PIE CHART – REPRESENTATION OF THE SUCCESS AND FAILURE RATE OF THE LAUNCH KSC LC-39-A



KSC LC 39-A achieved a 76.9% success rate and a 23.1% of failure

PAYLOAD Vs. OUTCOME SCATTER PLOT



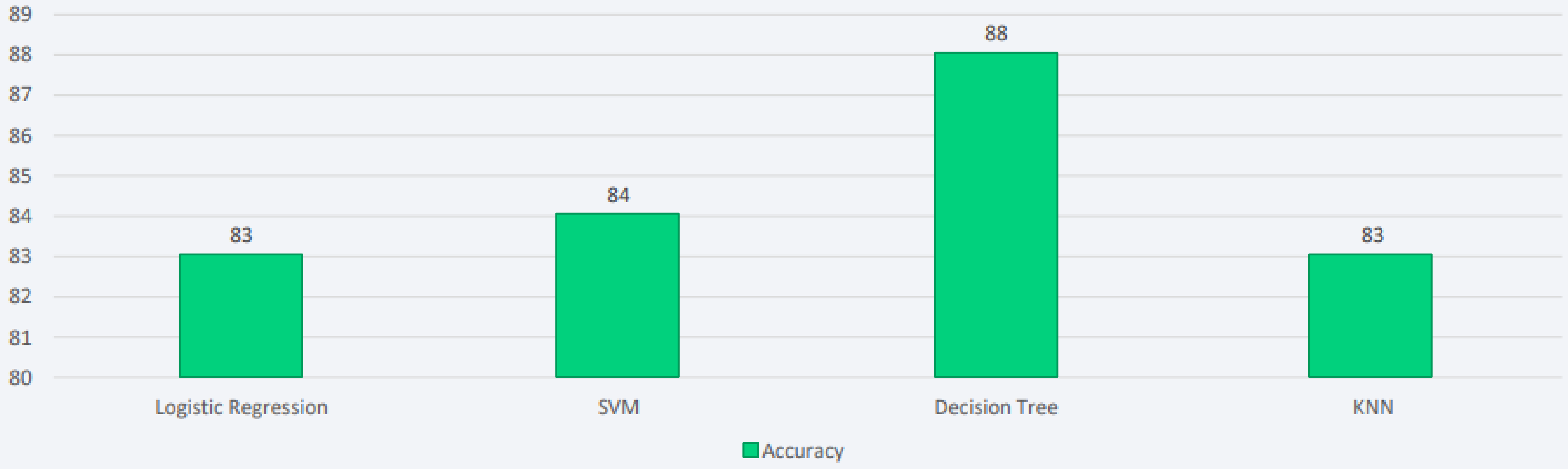


HIGHLIGHTS: Those scatter plot graphs representes launches for all sites, with different payloads Selected in a slider range. Its possible to realize that the success rates for low weighted payloads Is higher than the heavy weighted ones.

Section 5

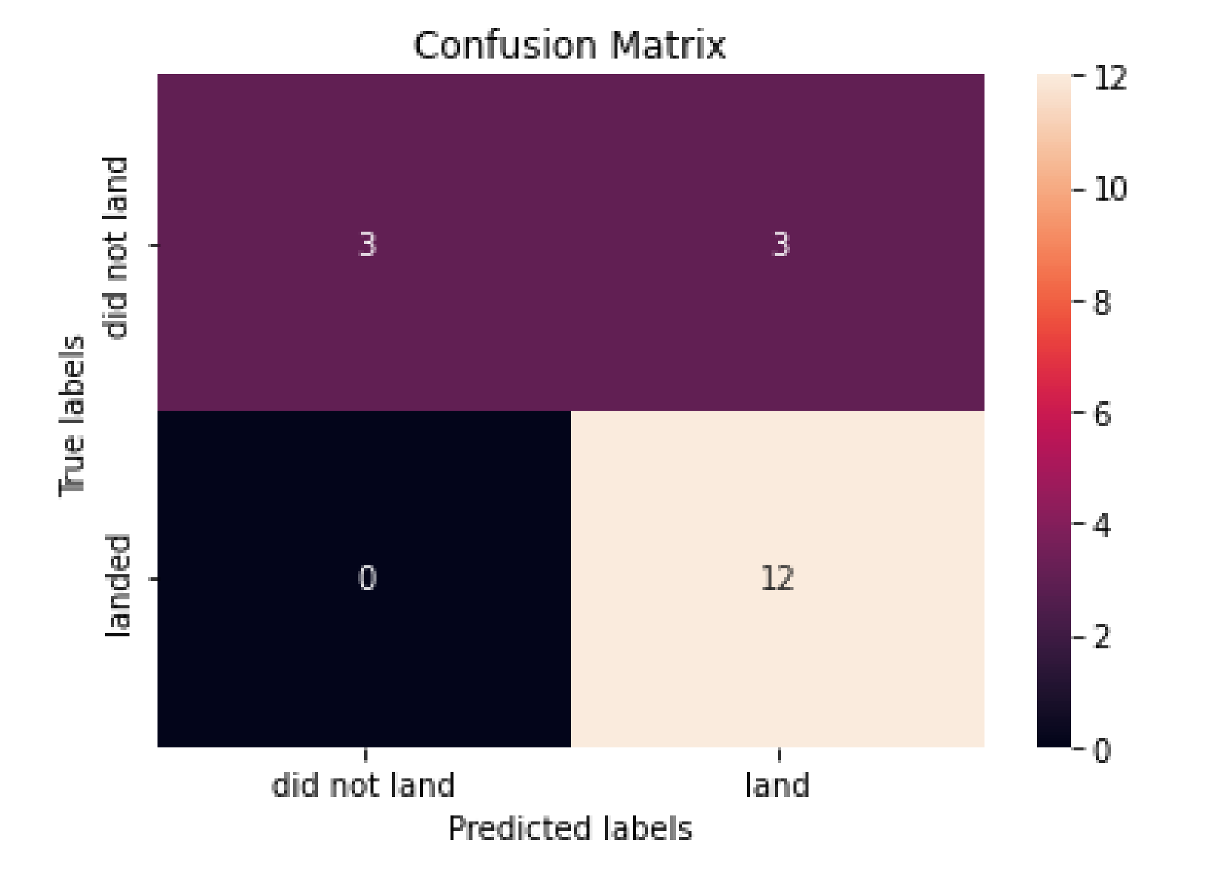
Predictive Analysis (Classification)

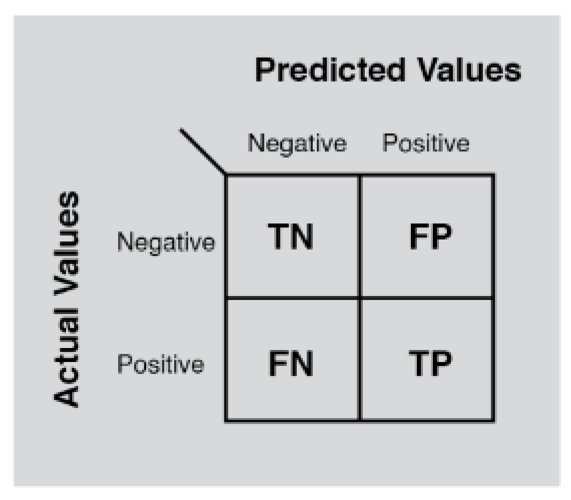
Classification Accuracy



HIGHLIGHTS: Decision Tree represents the best accuracy for this dataset

Confusion Matrix




Predicted Values

Actual Values

	Negative	Positive
Negative	**TN**	**FP**
Positive	**FN**	**TP**

True negatives and False positives have the some amount in this matrix

Conclusions

- ORBIT GEO, HEO, SSO, ES-L1 shows the higher success rate
- It is possible to realize that KSC LC-39A presents higher successful launches of all sites
- The success rate for SpaceX launches is directly proportional to the time. The longer (in years) of releases, the greater the chances of improvement.
- Lighter launches have highest chance of success
- The tree classififer algorithm is the best Machine Learning model for this dataset.

Appendix

PYTHON LIBRARIES

- Matplotblib
- TensorFlow
- Scikit-Learn
- Numpy
- Pandas

SQL SERVER MODULES

- ADG SQLSERVER

Thank you!