IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

JITTOU Asmaa

10/05/2023

Outline

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

Executive Summary

Methodologies

- Data collection Through API
- Data collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis using SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Summary of all results

- Data Analysis result
- Interactive Analytics result
- Predictive Analytics result of Machine Learning

Introduction

SpaceX is an American company specialized in the field of astronautics and space flight who has disrupt the space industry by offering a rocket launches, Flacon 9 specifically, as low as 62 million dollars. While other, each one providers cost upward of 165 million dollars. The price difference is explained by the fact that SpaceX can reuse the first stage.

By determining if the stage will land, as data scientist, we can determine the cost of a launch, gathering information about it, creating dashboards and train machine learning model using public information. The results is interesting for another company if it wants to compete with SpaceX for a rocket launch.

To solve the problem, the following questions need to be answered:

- What are the main characteristics of a successful or failed landing ?
- How can we determine if SpaceX will reuse the first stage ?
- What are the effects of each relationship of the rocket variables on the success or failure of a landing ?
- What are the conditions which will allow SpaceX to achieve the best landing success rate ?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling:
  - Chose necessary columns for data analysis
  - Apply One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models:
  - How to build, tune, evaluate classification models

Data Collection

Data collection, the most important phase in data science project, helps to collect and make sure the data is in the correct format. In this project, the data was collected from Rest SpaceX API and webscrapping Wikipedia.

REST API : Data collected are launches, rocket and payload information.

- Request and parse the SpaceX launch data using the GET request (URL: https://api.spacexdata.com/v4/{variable_name});
- Filter the dataframe (from JSON) to only include `Falcon 9` launches;
- Dealing with Missing Values and explore data;

Webscrapping : Data collected are Landing, launches and payload using BeautifulSoup.

- Request the Falcon9 Launch Wiki page from its URL;
- Extract all column/variable names from the HTML table header;
- Create a data frame by parsing the launch HTML tables;

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data (Json file) and did some basic data wrangling and formatting.

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

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

```python
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```python
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and rep
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the t
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Get Request using API

Convert json results to dataframe using json_normalize

Data cleaning

Data Collection - Scraping

- In this part, We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup. Then, parsed and converted the table into a pandas dataframe.

Request the flacon Launch Wikip page (url)

Create BeautifulSoup (HTML response)

Extracting attribute names needed from the HTML Table and Header

```python
data= requests.get(static_url).text
```

[4] Python

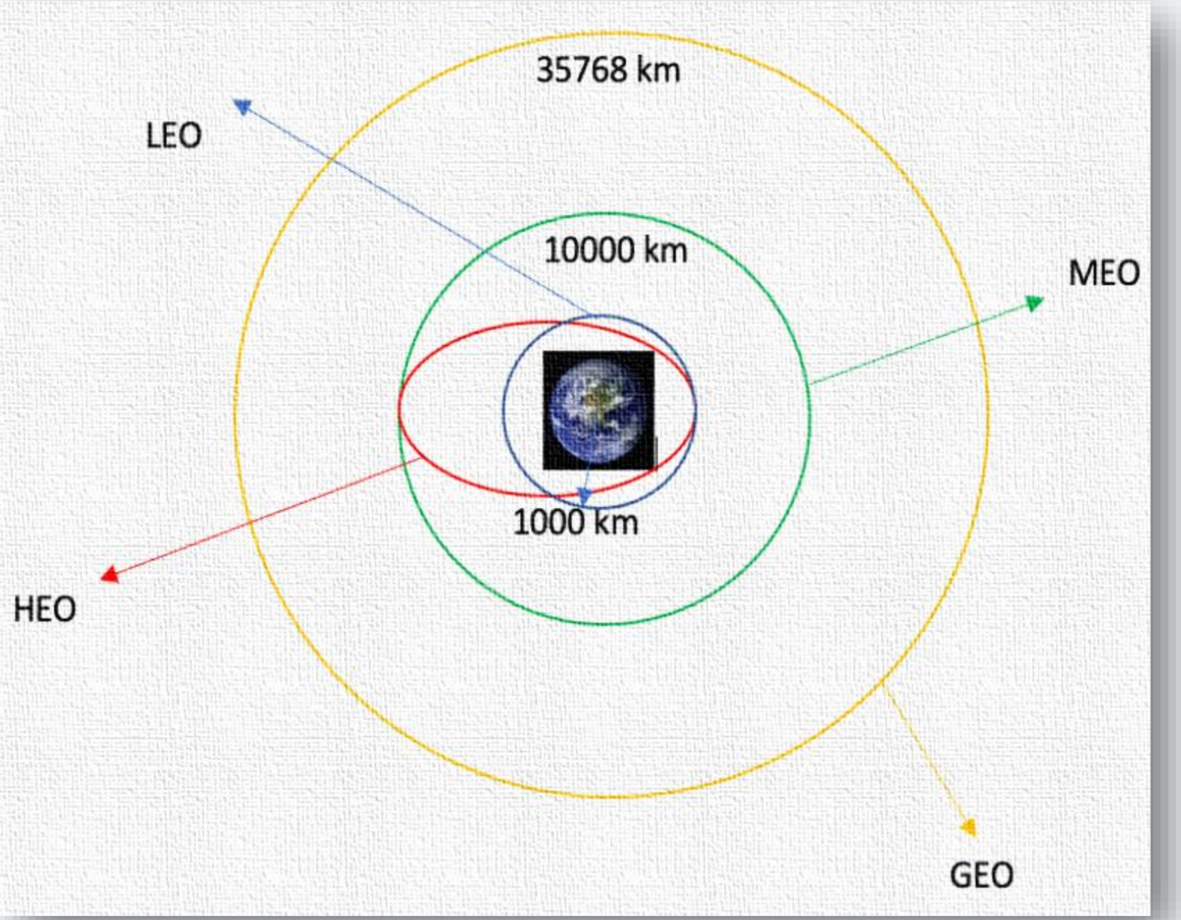
```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup =BeautifulSoup(data,'html.parser')
```

Python

```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```

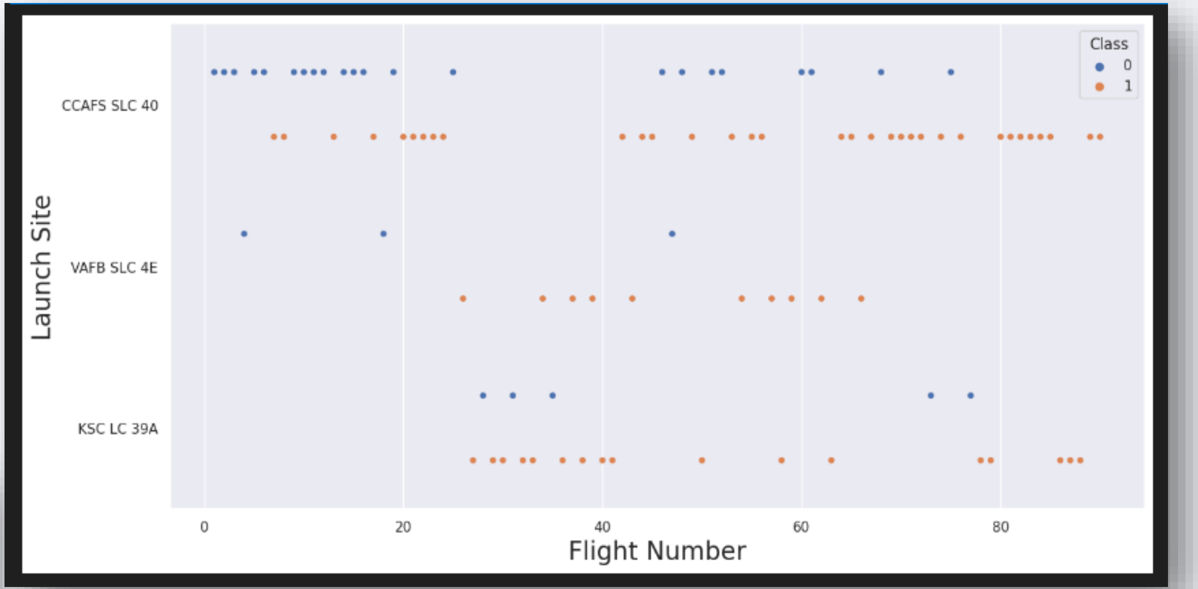
Data Wrangling

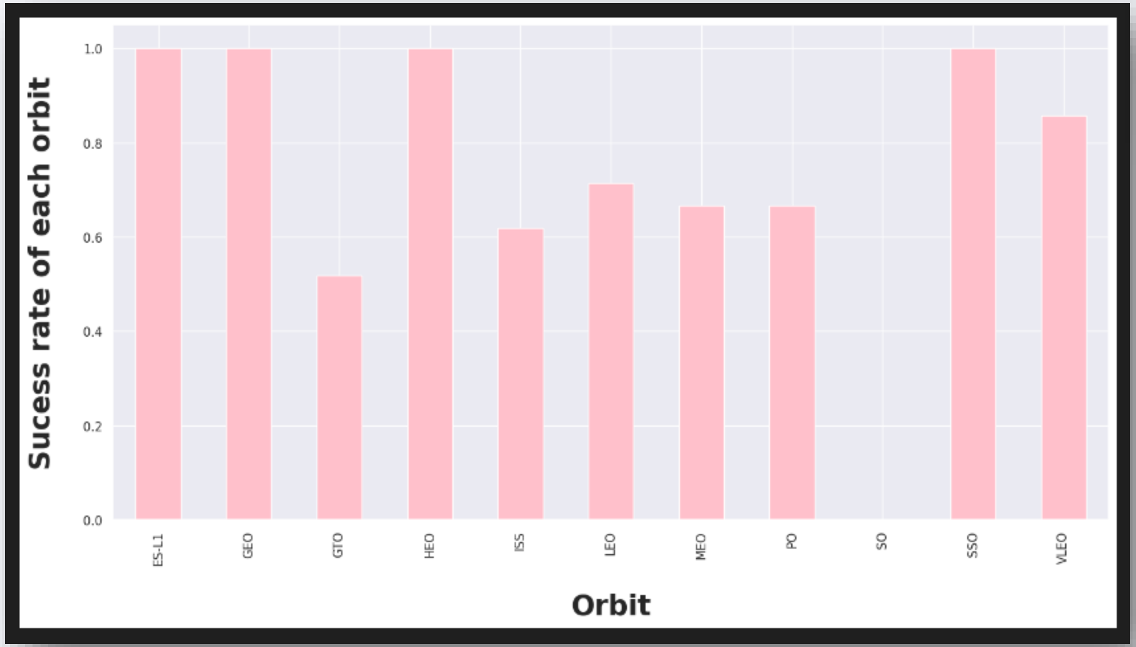
- When we talk about cleaning data, we talk also about Data Wrangling, is when we performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column, to simplfy our analysis and visualization, and finaly we exported the results to csv.



EDA with Data Visualization

- We started to find the relationship between variables using Scatter Graph. (in this example: Flight Number & and launch Site).

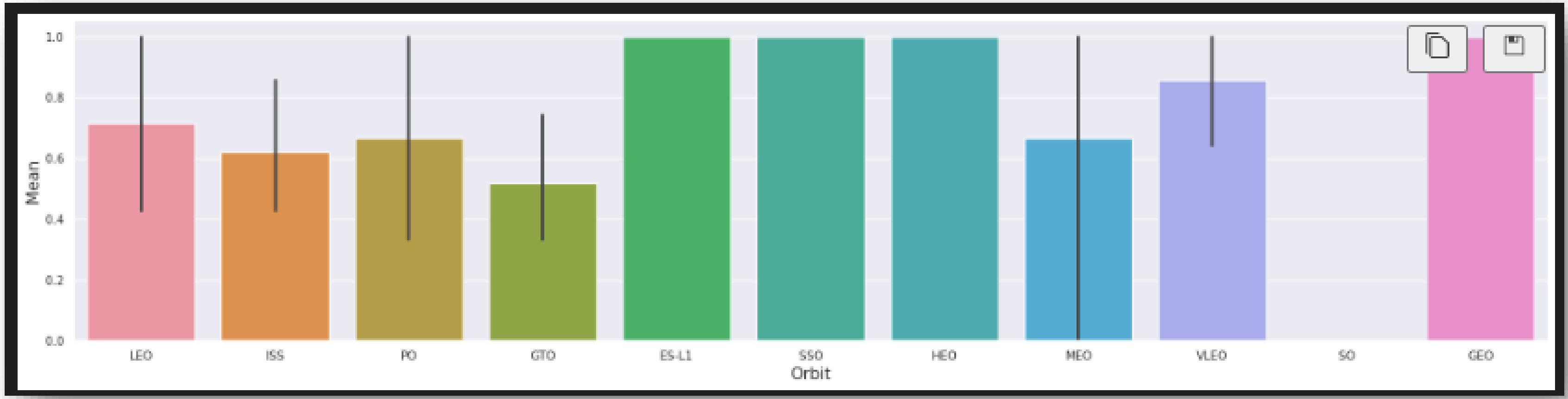




- There are different visualization tools (like bar graph, line plot ..) used for our analysis.

- In this bar graph, help us to determine which orbits have the success highest probability.

EDA with Data Visualization



- After Visualization, we obtained some preliminary insights about how each important variable would affect the success rate, we talk about Feature Engineering where we select the features that will be used in success prediction in the future module.

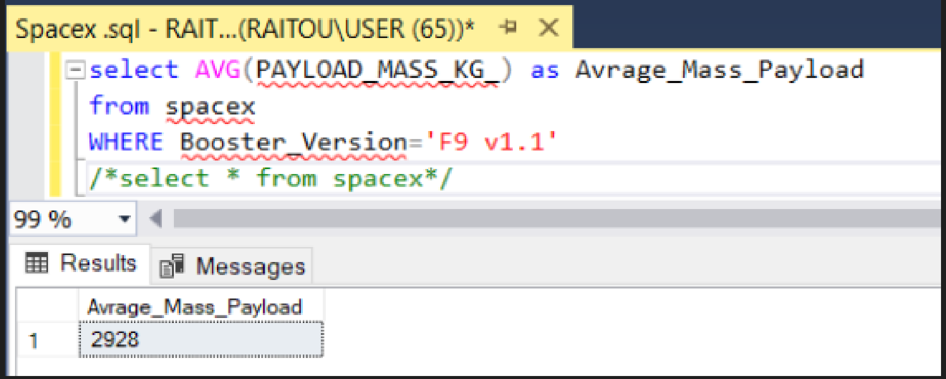
```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', …
```

EDA with SQL

- In our case, to better understand the dataset, we loaded the SpaceX (.sql) dataset into a SQL SERVER database (without jupyter notebook). We wrote some queries to get insights from data, like:
  - Displaying the names of unique launch sites.
  - Displaying 5 records where launch sites begin with ‘CCA’.
  - Displaying the total payload mass carried by boosters launched by NASA (CRS) ..

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

select AVG(PAYLOAD_MASS_KG_) as Avrage_Mass_Payload from spacex WHERE Booster_Version='F9 v1.1'



Build an Interactive Map with Folium

- This phase, we visualize the launch data into interactive map, added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes, a dataframe launch_outcomes, (failure or success) to class 0 and 1, where the color-labeled RED (class 0) marker for failure, and GREEN (class 1) marker for success.
- We then calculated the distances between a launch sites to different landmark to know:
  - How close the launch sites to near highways, railways and coastlines ?
  - How close the launch sites with other cities ?

Predictive Analysis (Classification)

Data preparation & building Model:

- Load the dataset using Numpy and Pandas libraries.
- Transform the data and split it into training and test datasets.
- Choose Machine Lerning types.
- Set algorithms toGridSearchCV and parameters.
- Fit the model

Evalueting the Model:

- Use accuracy metric to evaluate models.
- Get tuned hyperparams for each model/algorithm.
- For each algorithm, plot the confusion matrix.

Imporving & Find Best Model:

- Use Feature Engeering and algorithm Tuning.
- The best model's accuracy score is the best performing model.

Results

The results ll be categorized to :

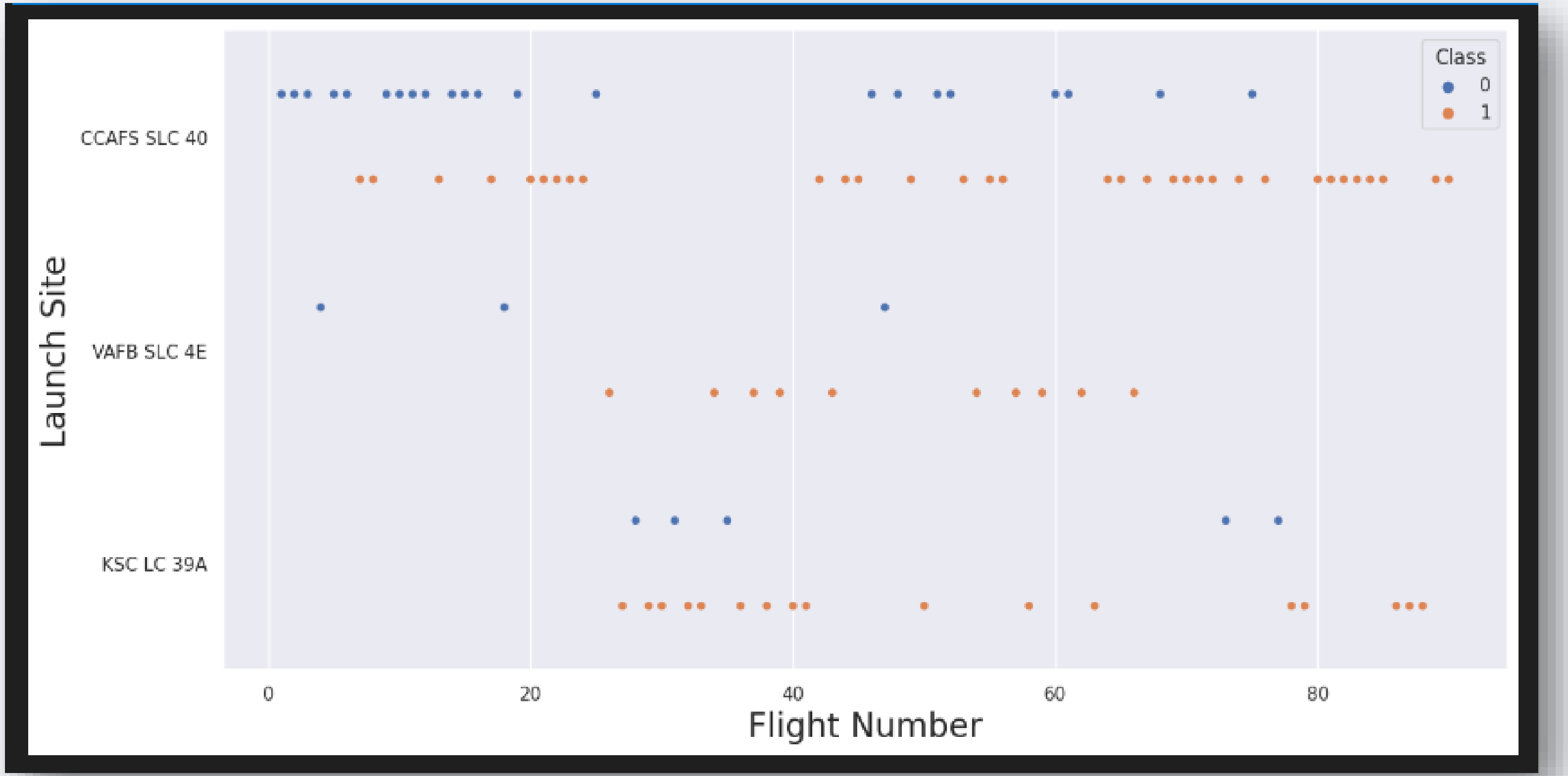
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

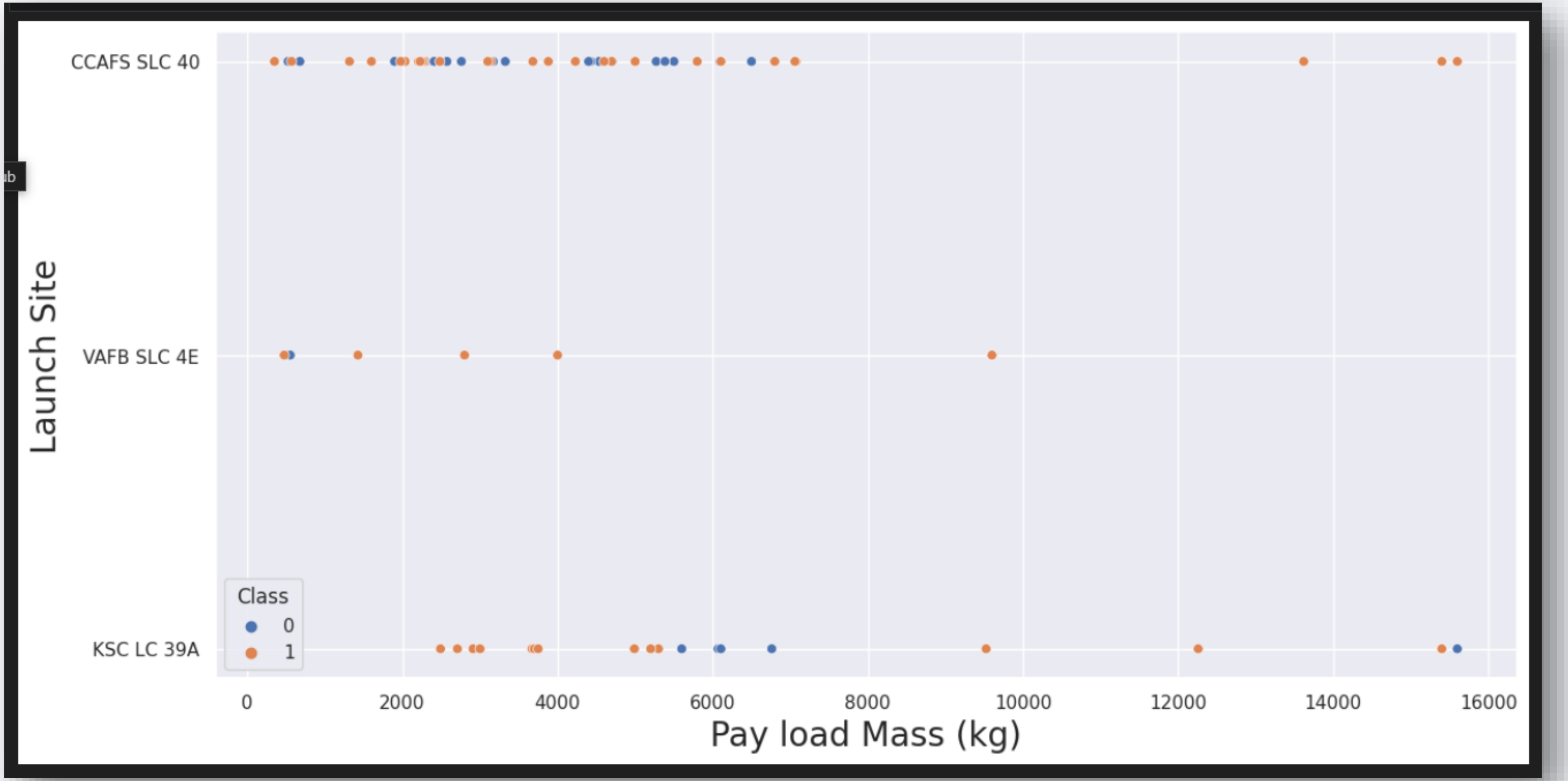
Flight Number vs. Launch Site

- In this scatter plot, the larger the flight amount at a launch site, the greater the success rate at a launch site. However, site CCAFS SLC40 shows the least pattern.



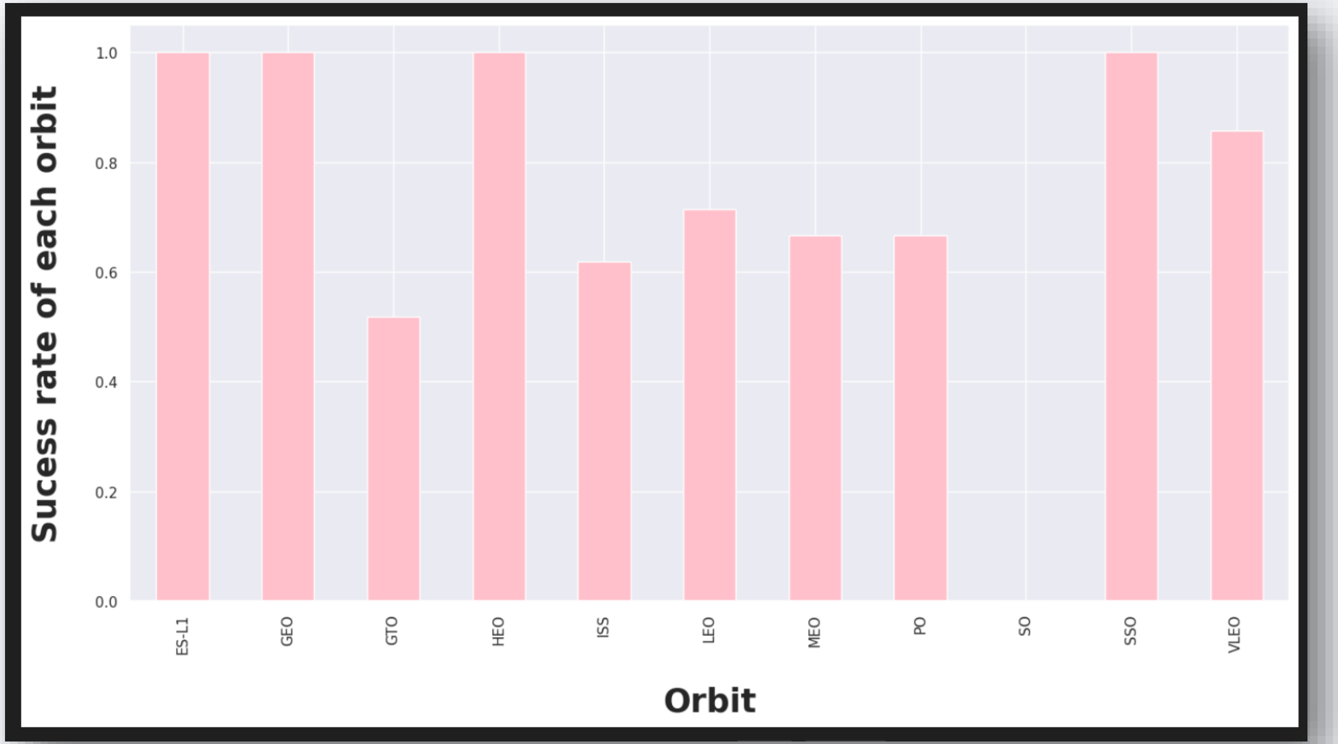
Payload vs. Launch Site

- The probability of the success rate will be highly increase when the pay load mass is greater than 7 000KG.



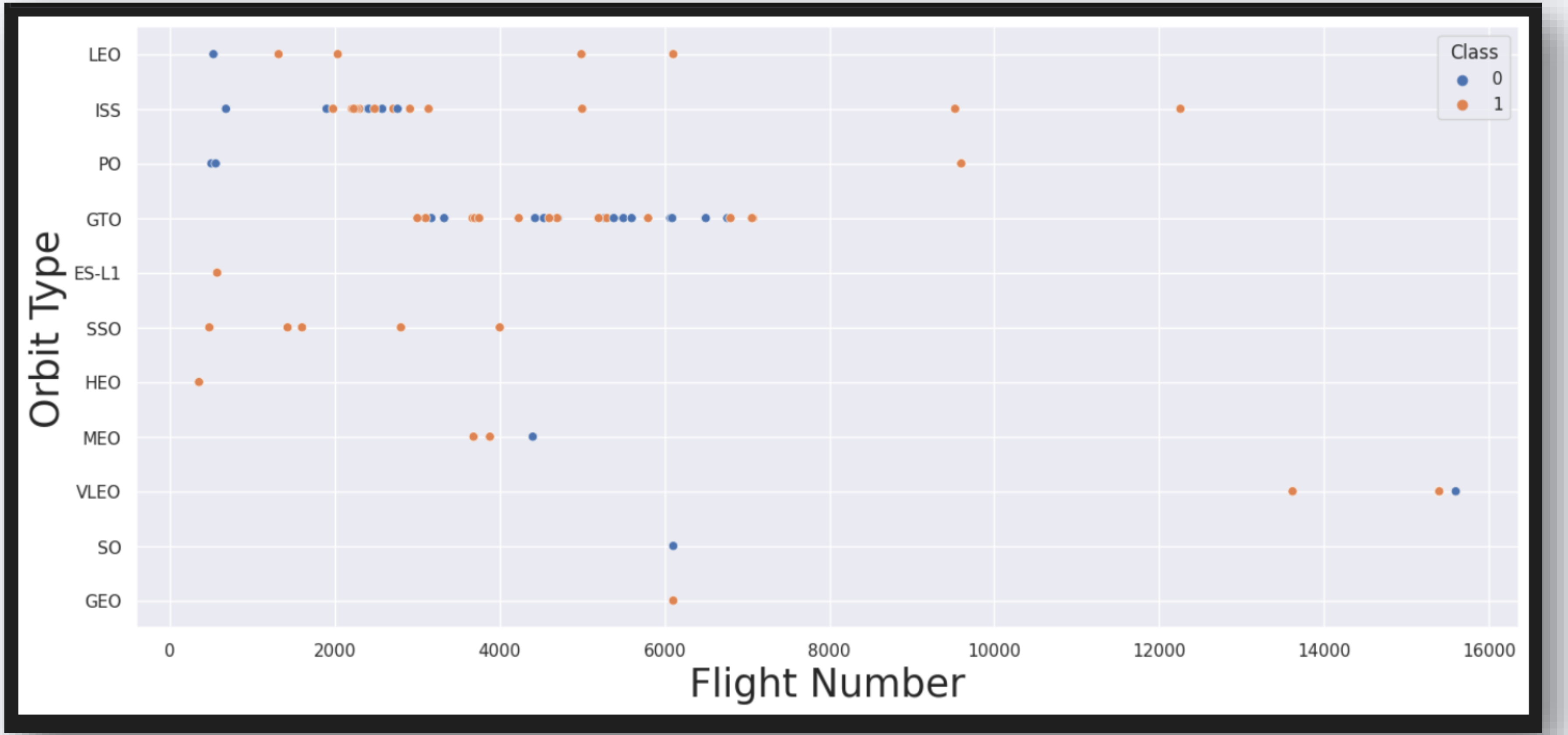
Success Rate vs. Orbit Type

- We can say that we need more data to see trends because, in this figure, some of orbits has only 1 occurrence (GEO, SO, HEO, ES-L01).
- In general, this figure show the orbits's probability to influences the landing oiutcomes, 100% for HEO, GEO, SSO and 0% success rate for orbit SO.



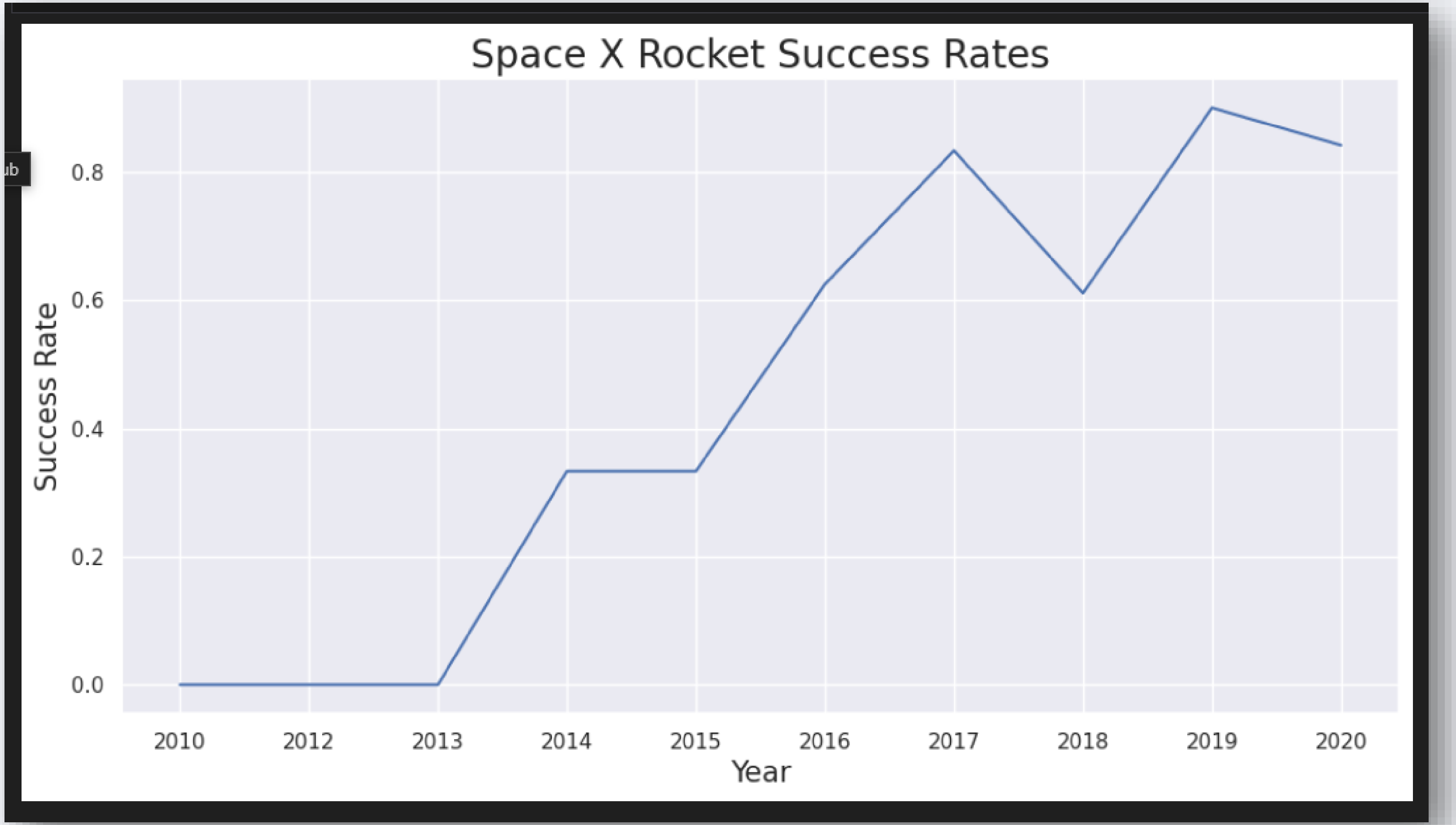
Flight Number vs. Orbit Type

- The same remark, we need more data (ES-L1).. But in general in this figure, we found that the larger the flight number on each orbits the greater the success rate (except GTO that didn't have any relationship).



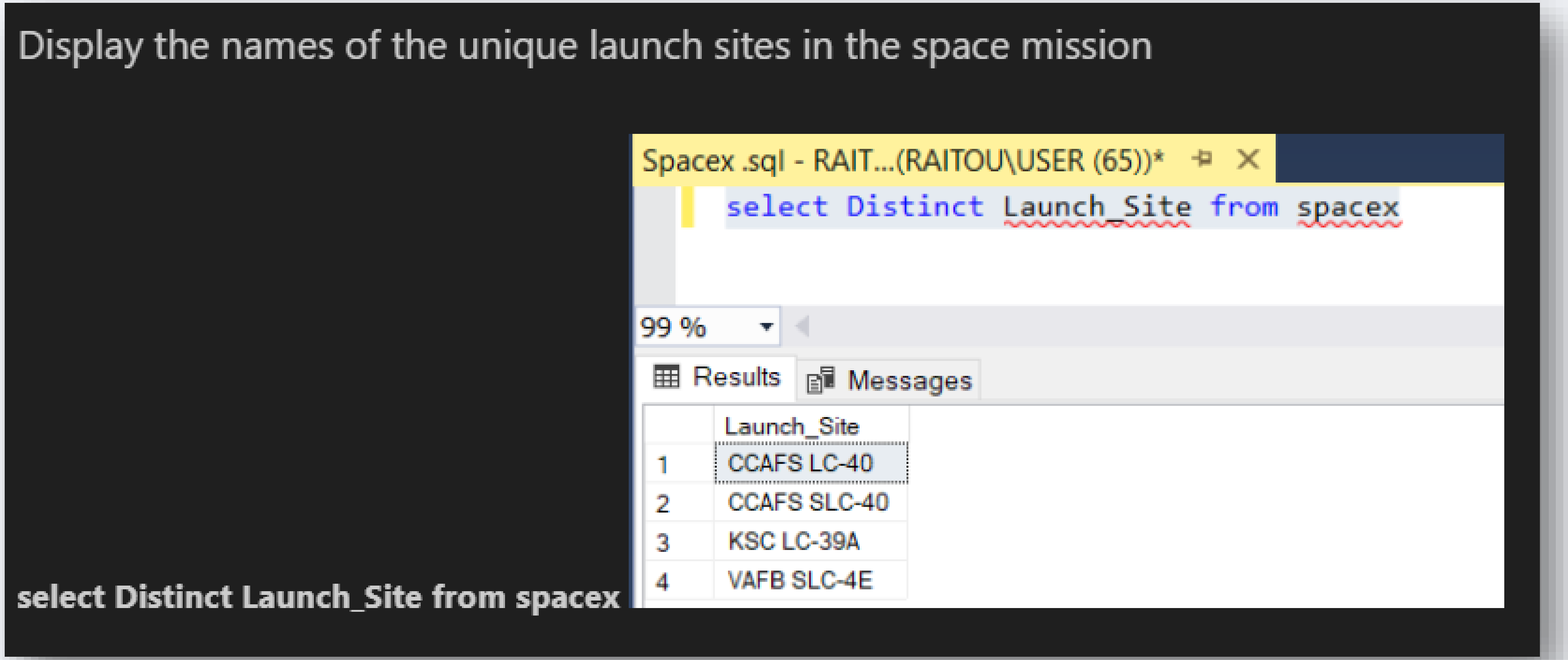
Launch Success Yearly Trend

- In this figure, we can say that success rate since 2013 kept on increasing until 2020.



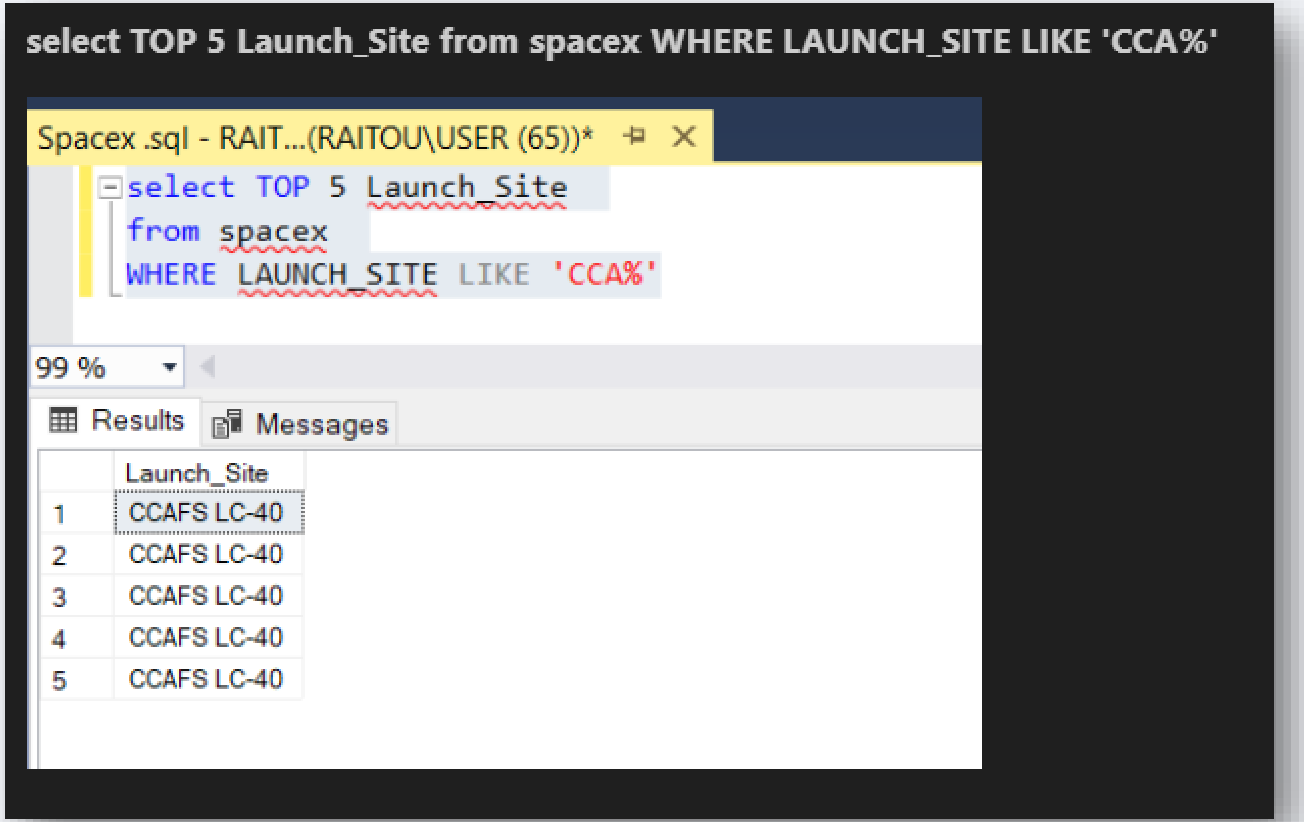
All Launch Site Names

- To show only unique launch sites from the dataset/or table SpaceX we use the sql key word **DINSTINT** in the cls select.



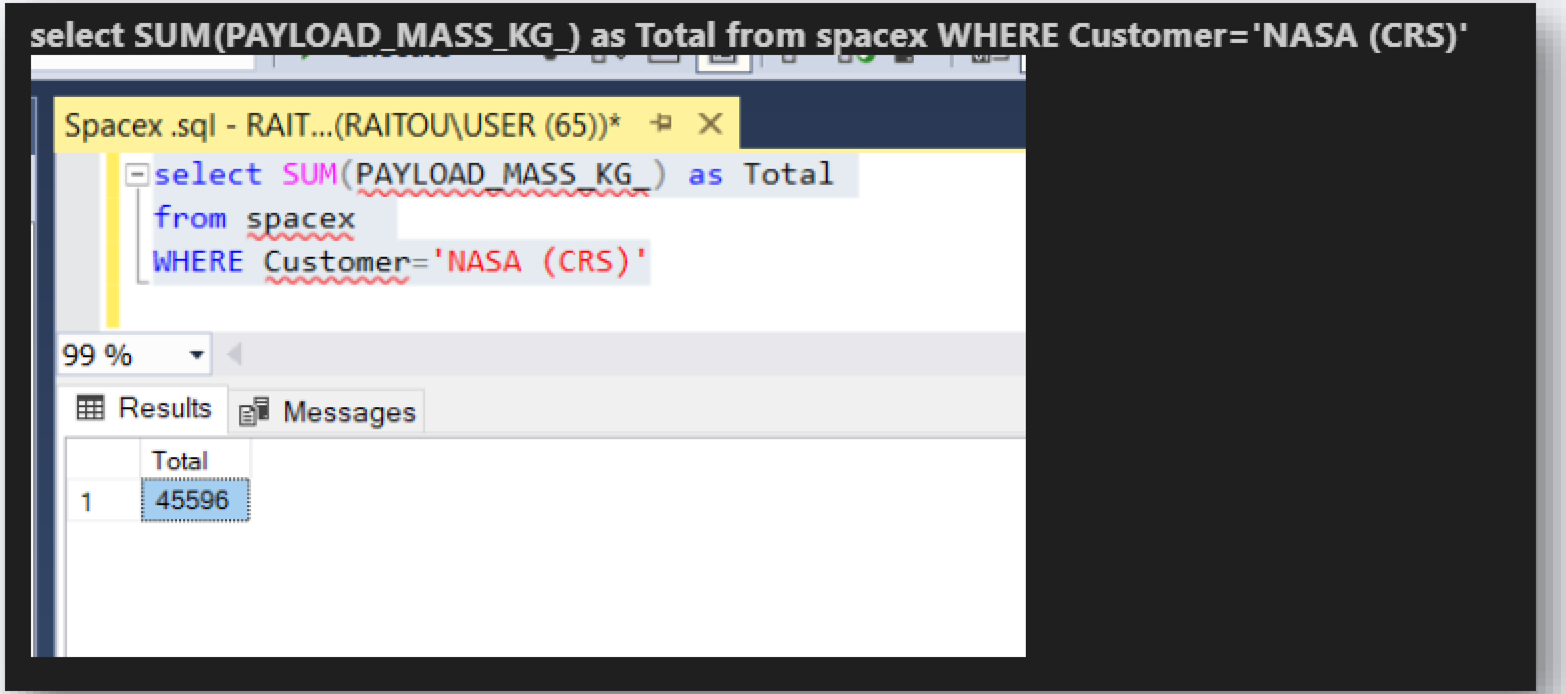
Launch Site Names Begin with 'CCA'

- To Find 5 records where launch sites begin with `CCA`, we use TOP (specify records number) and LIKE to find launch with CCA.



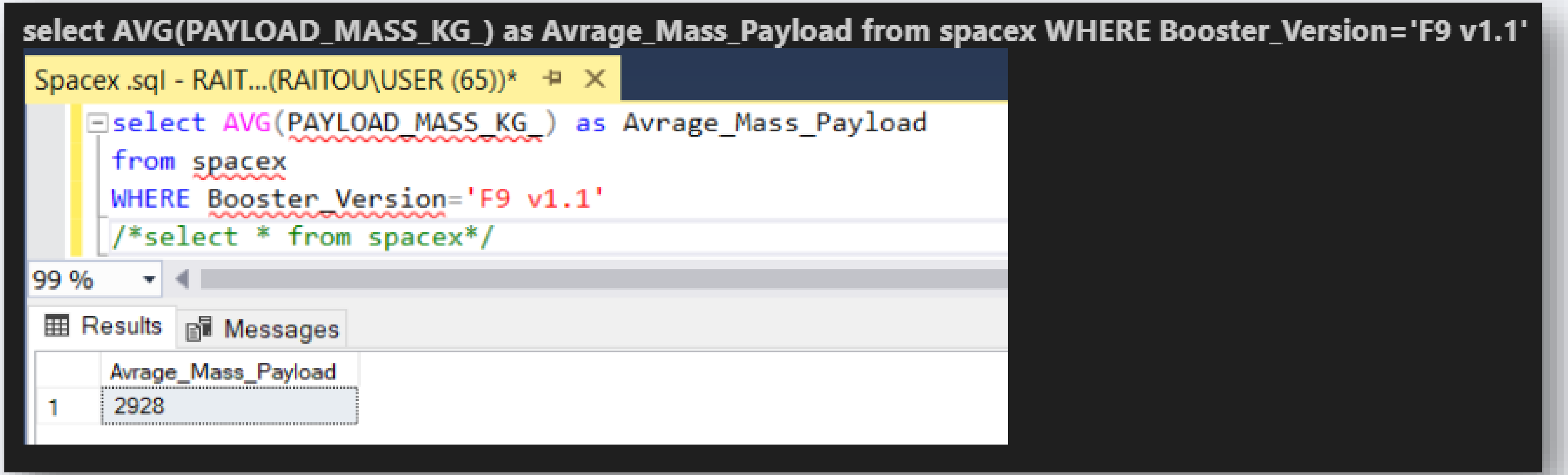
Total Payload Mass

- Using this query, we calculated the total payload carried by boosters from NASA and we get as result 45596.



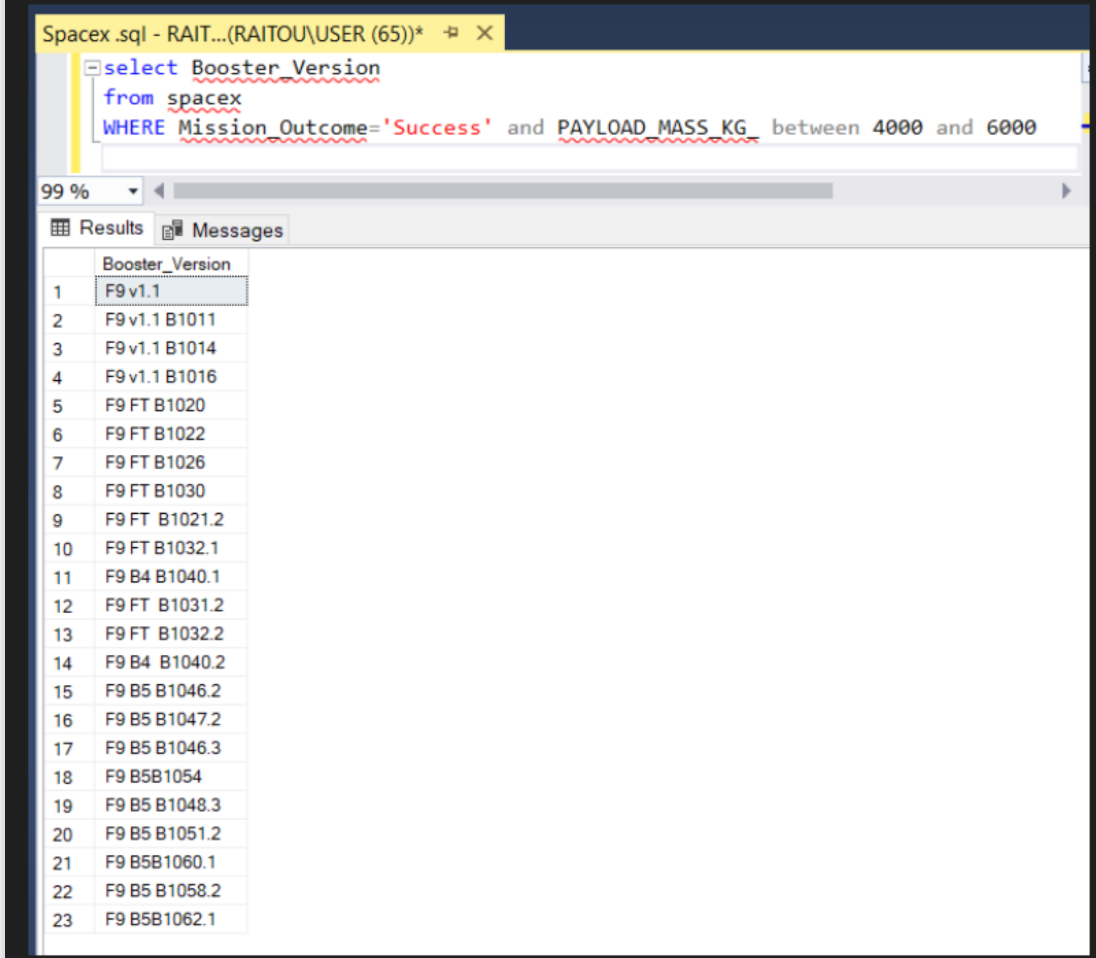
Average Payload Mass by F9 v1.1

- The Calculating result of the average payload mass carried by booster version F9 v1.1 is 2928 using the aggregation fuction **AVG()**.



Successful Drone Ship Landing with Payload between 4000 and 6000

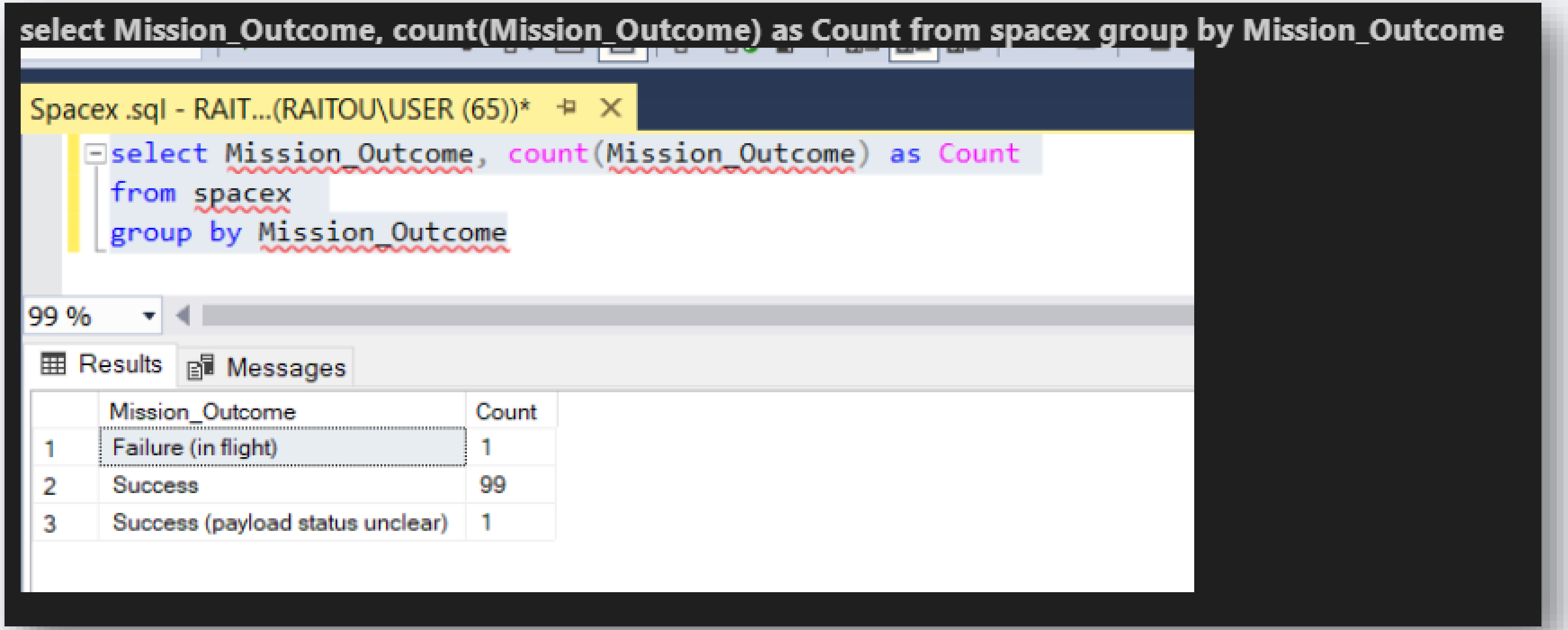
select Booster_Version from spacex WHERE Mission_Outcome='Success' and PAYLOAD_MASS_KG_ between 4000 and 6000



- We make condition and filtering in the clause **Where** to return a list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 and less than 6000.

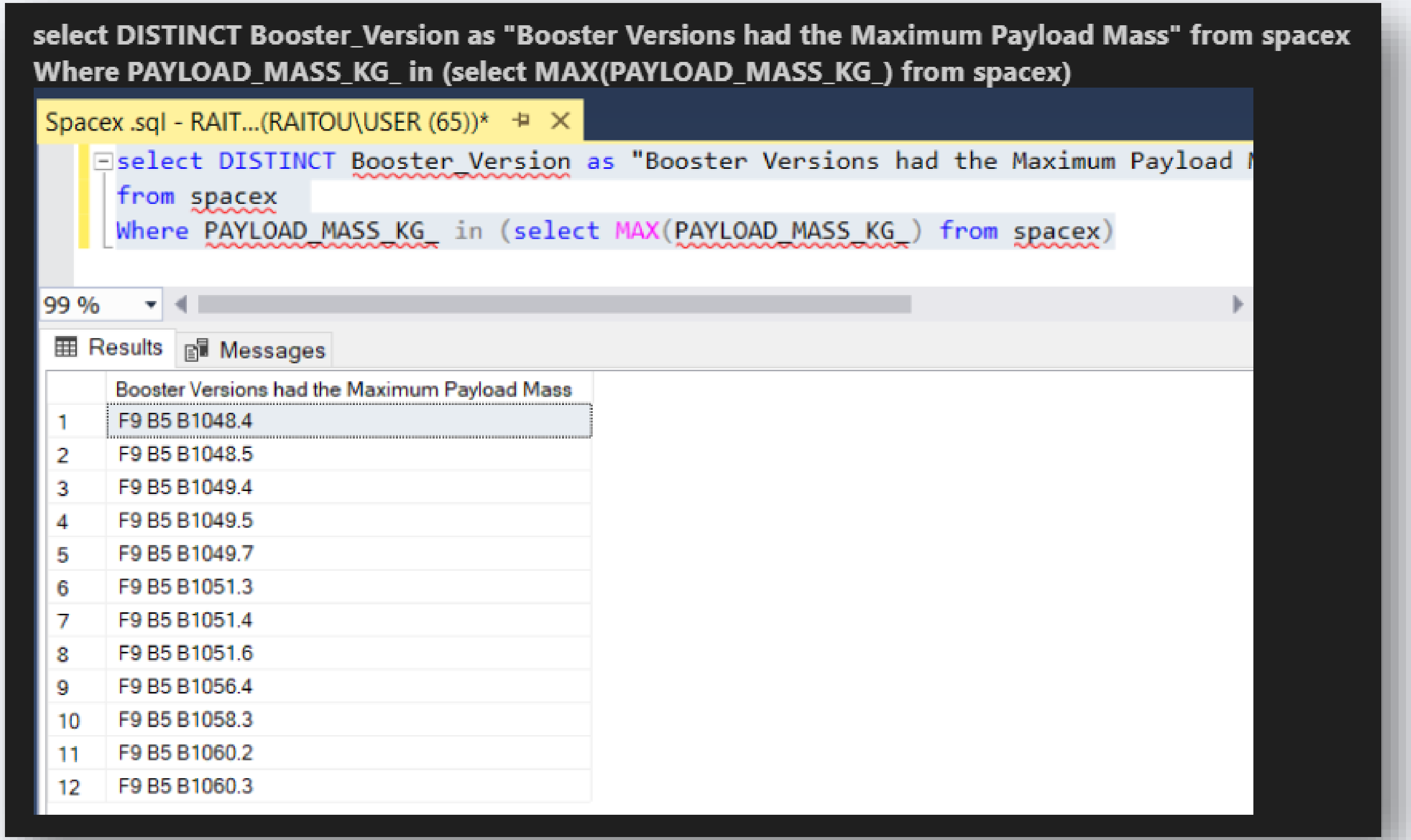
Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcome, we need to group records and use the aggregation function COUNT.



Boosters Carried Maximum Payload

- We use the Subquery in the clause **WHERE** and **MAX()** function to have the booster that have carried the maximum payload.



2015 Launch Records

- We add conditions in the clause WHERE to filter failed landing outcomes in drone ship List the failed landing_outcomes in drone ship, their booster versions, and launch site names for year 2015.

**select Landing_Outcome, Booster_Version , Launch_site from spacex as SpaceX Where Landing_Outcome='Failure (drone ship)' and Year(SpaceX.DATE)=2015 **

Spacex .sql - RAIT...(RAITOU\USER (65))*

```sql
select Landing_Outcome, Booster_Version , Launch_site
from spacex as SpaceX
Where Landing_Outcome='Failure (drone ship)' and Year(SpaceX.DATE)=2015
```

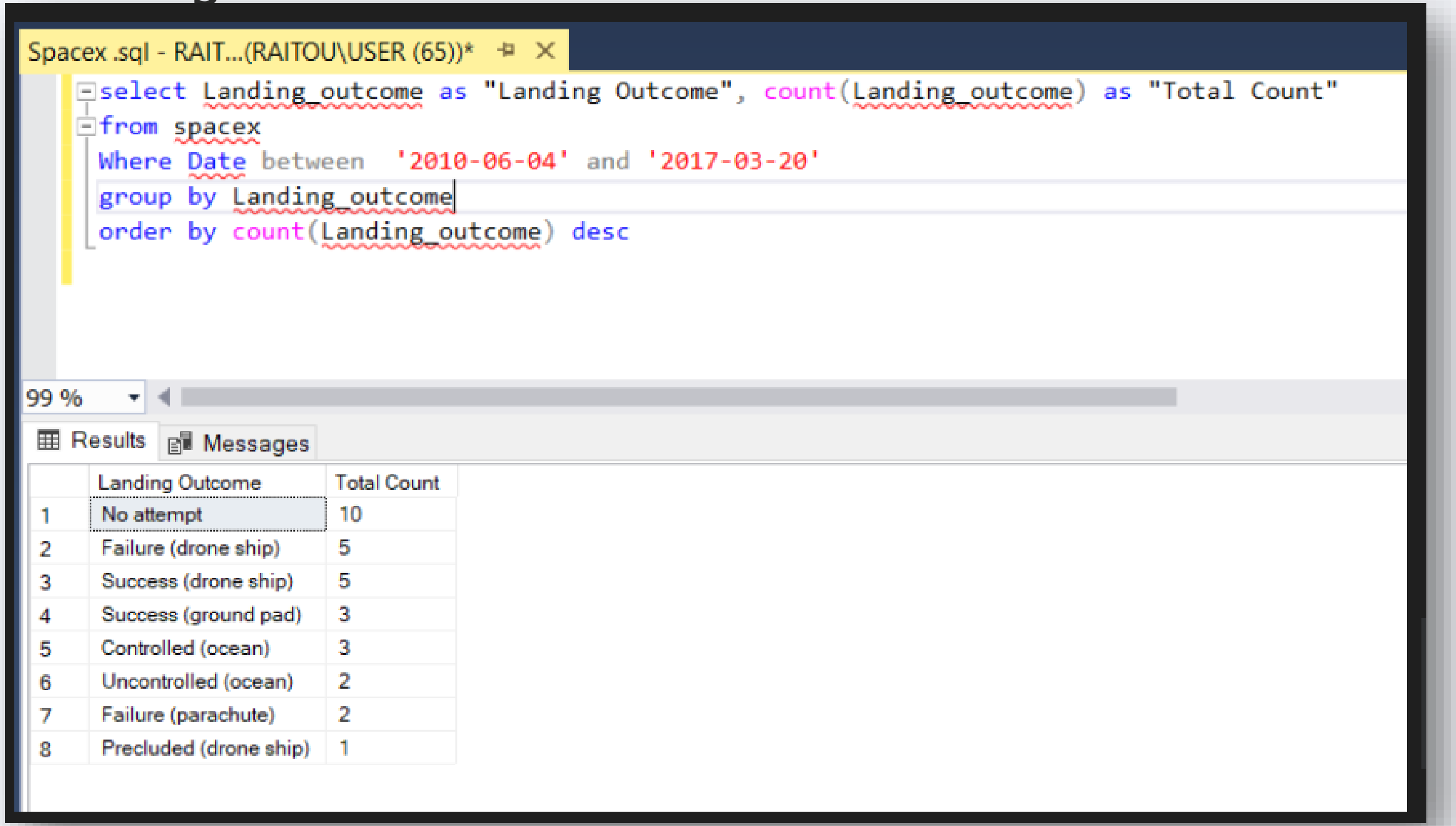
99 %

Results Messages

	Landing_Outcome	Booster_Version	Launch_site
1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- We use the aggregation finction COUNT to count landing outcomes and groupe them using **GroupBY** (such as Failure (drone ship) or Success (ground pad)), and the key BETWEEN to specify date range (2010-06-04 and 2017-03-20), and to order results's records in descending order we use ORDER BY DESC.




```sql
select Landing_outcome as "Landing Outcome", count(Landing_outcome) as "Total Count"
from spacex
Where Date between  '2010-06-04' and '2017-03-20'
group by Landing_outcome
order by count(Landing_outcome) desc
```

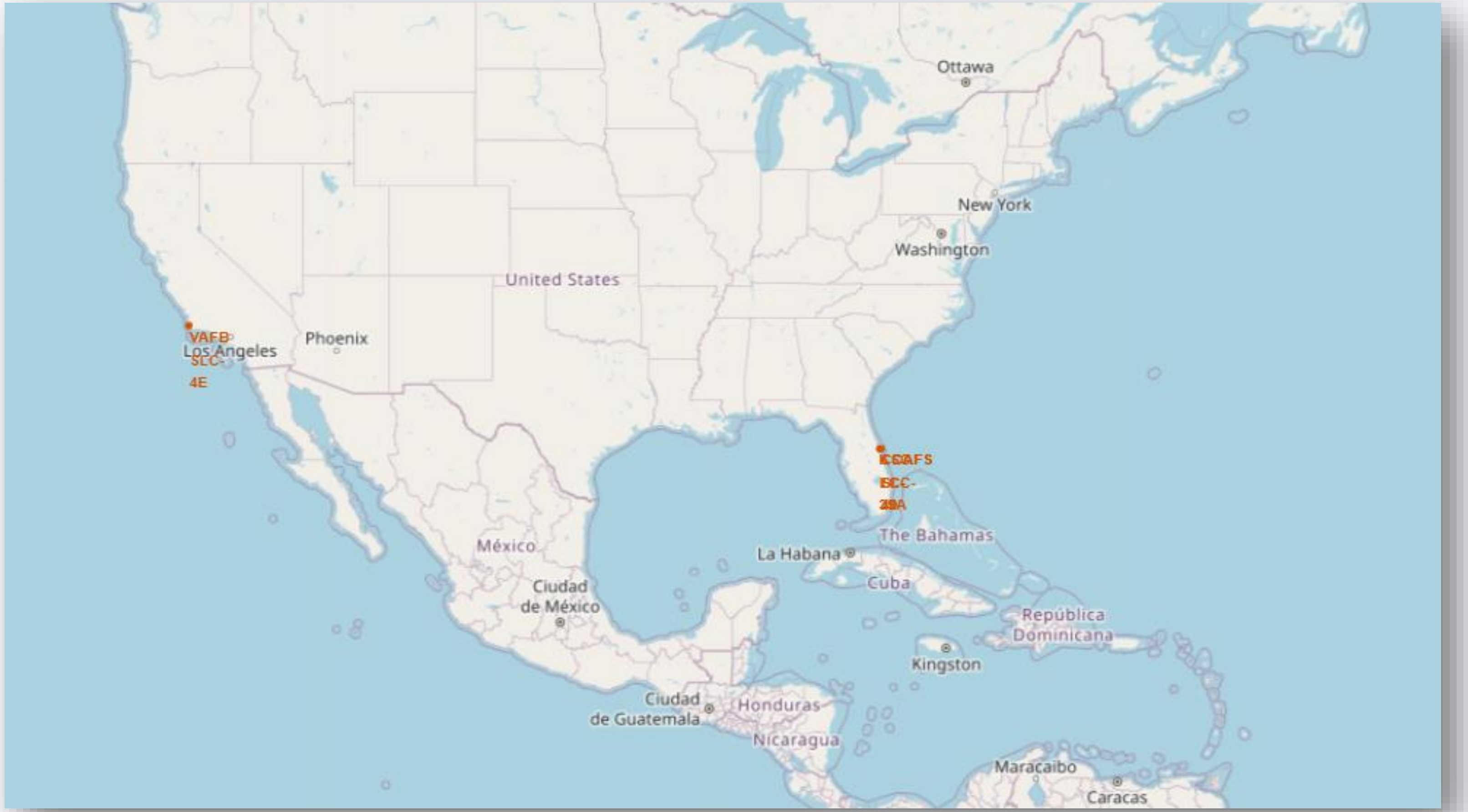
	Landing Outcome	Total Count
1	No attempt	10
2	Failure (drone ship)	5
3	Success (drone ship)	5
4	Success (ground pad)	3
5	Controlled (ocean)	3
6	Uncontrolled (ocean)	2
7	Failure (parachute)	2
8	Precluded (drone ship)	1

Section 3

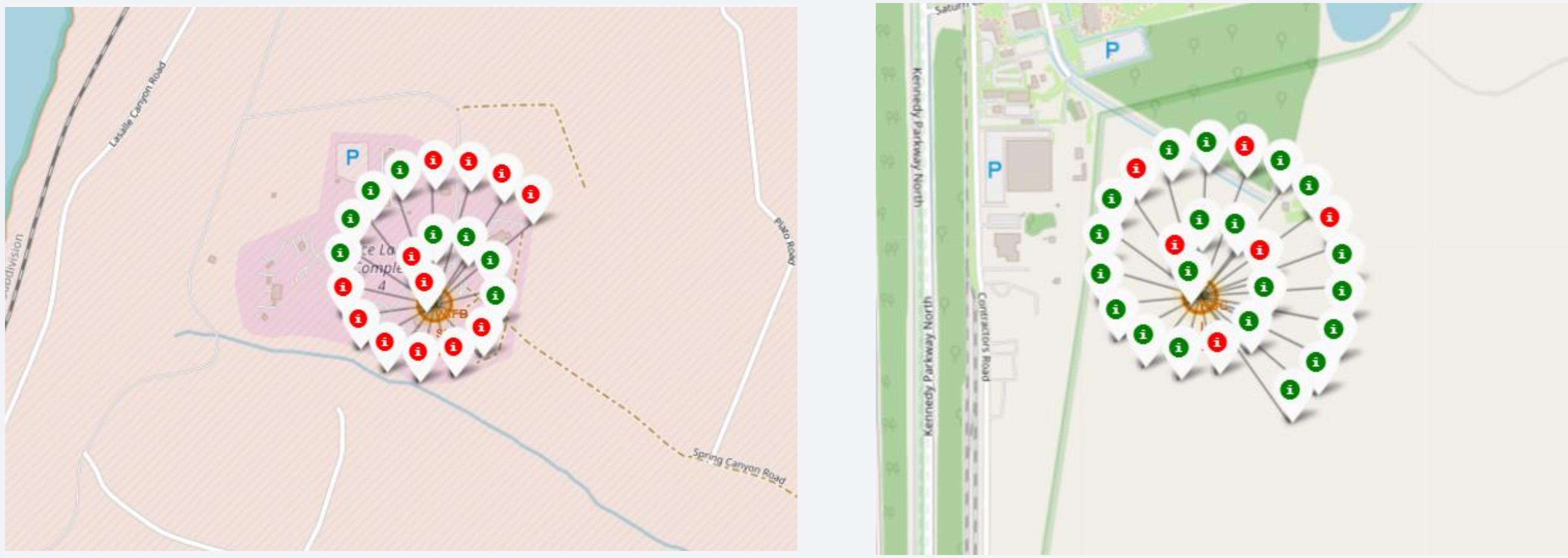
Launch Sites Proximities Analysis

All launch sites global map

- We can see that the SpaceX Launch sites are in Florida, California and United States of America Coasts.

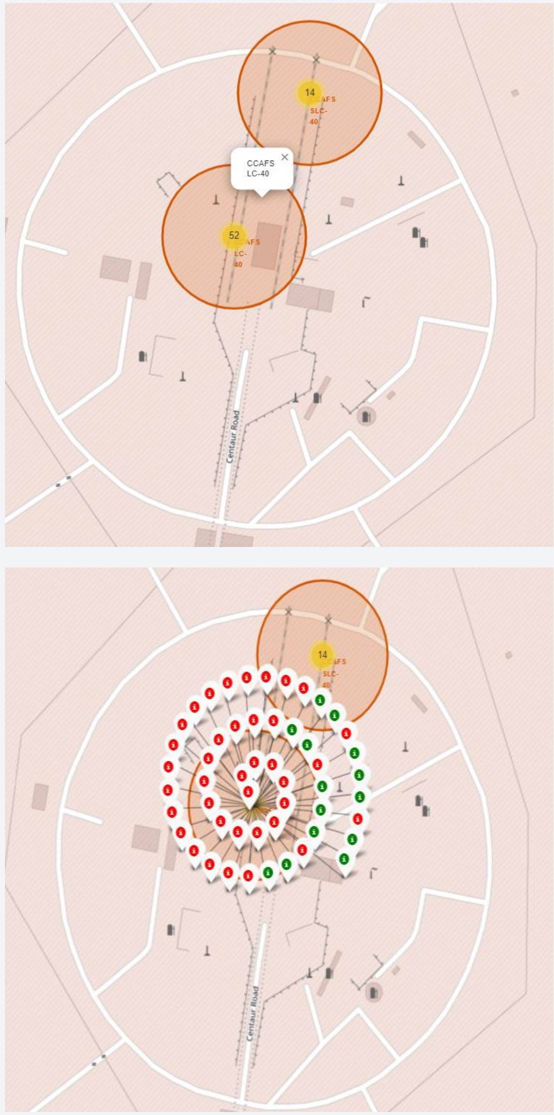


Markers launch sites labeled in the map



Florida Launch sites

- Green Marker shows successful Launches and the Red Marker shows Failures.



California Launch site

Section 4

Predictive Analysis (Classification)

Classification Accuracy

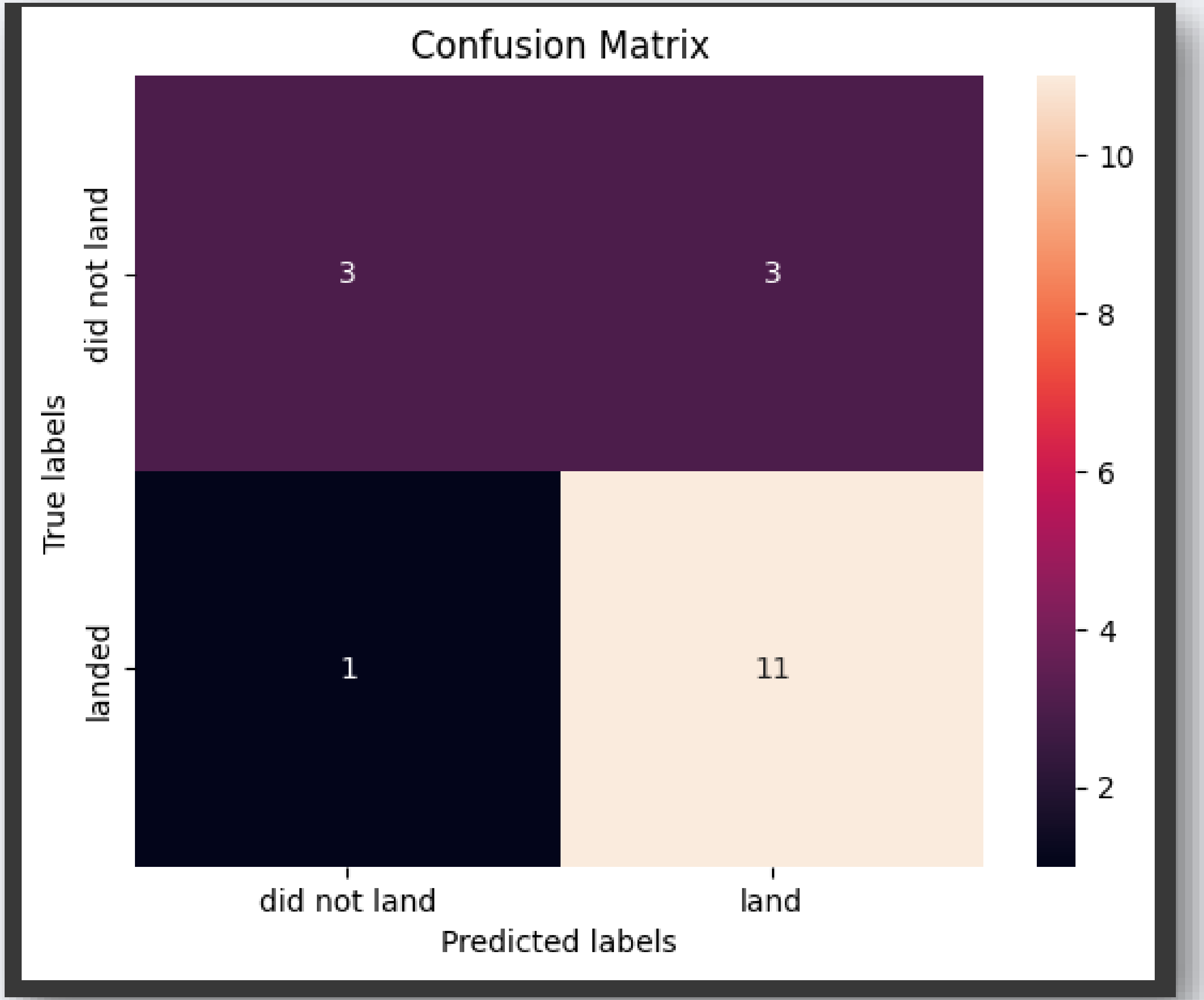
```python
algorithms = {'LogisticRegression':logreg_cv.best_score_ , 'SVM': svm_cv.best_score_ ,'Tree':tree_cv.best_score_ ,'KNN':knn_cv.best_score_,}
best_algorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',best_algorithm,',with score of:',algorithms[best_algorithm])
if best_algorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
elif best_algorithm == 'SVM':
    print('Best Params is :',svm_cv.best_params_)
elif best_algorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
elif best_algorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
```

Python

```
Best Algorithm is Tree ,with score of: 0.8767857142857143
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'best'}
```

- According results, we can identify the best algorithm :The decision tree classifier model which have the highest classification accuracy

Confusion Matrix



Conclusions

We can conclude that:

- The Decision tree classifier is the best machine learning algorithm for this dataset and task.
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Starting from 2013, Launch success rate started to increase until 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The low weighted payloads performed better than the heavy weighted ones.
- KSC LC-39A had the most successful launches of any sites (76.9%);

Thank you!