Data Science Capstone Project

PANJAB SINGH

11-10-2022

OUTLINE

EXECUTIVE SUMMARY

- **Summary of Methodologies:**
- Data collection through API
- Data collection with Web Scraping
- Data Wrangling
- EDA with SQL
- EDA using Pandas and Matplotlib
- Interactive Visual Analytics and Dashboard
- Predictive Analysis(Classification)

- **Summary of all Results:**
- EDA Results
- Interactive Visual Analytics and Dashboard(Screenshots)
- Predictive Analysis Results(ML)

INTRODUCTION

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. The majority of these savings can be attributed to SpaceX's brilliant idea to reuse the launch's first stage by re-landing the rocket for the following flight. The price will decrease considerably more if this method is repeated. The objective of my project, which I am working on as a data scientist for a firm that competes with SpaceX, is to build a machine learning pipeline to forecast how the first stage will land in the future. The success of this initiative will determine how much to offer SpaceX for a rocket launch.

Objective :

- Discover all factors that influence the landing outcome.
- The relationship between each variables and how it is affecting the target Variable.
- The best Possibility needed to increase the probability of successful landing.

METHODOLOGY

METHODOLOGY

- Executive Summary
  - Data collection methodology:
- Data was collected using SpaceX REST API and web scrapping from Wikipedia
  - Perform data wrangling :
- Data was processed using one-hot encoding for categorical features
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly
  - Perform predictive analysis using classification models :
- How to build, tune, evaluate classification models

DATA COLLECTION

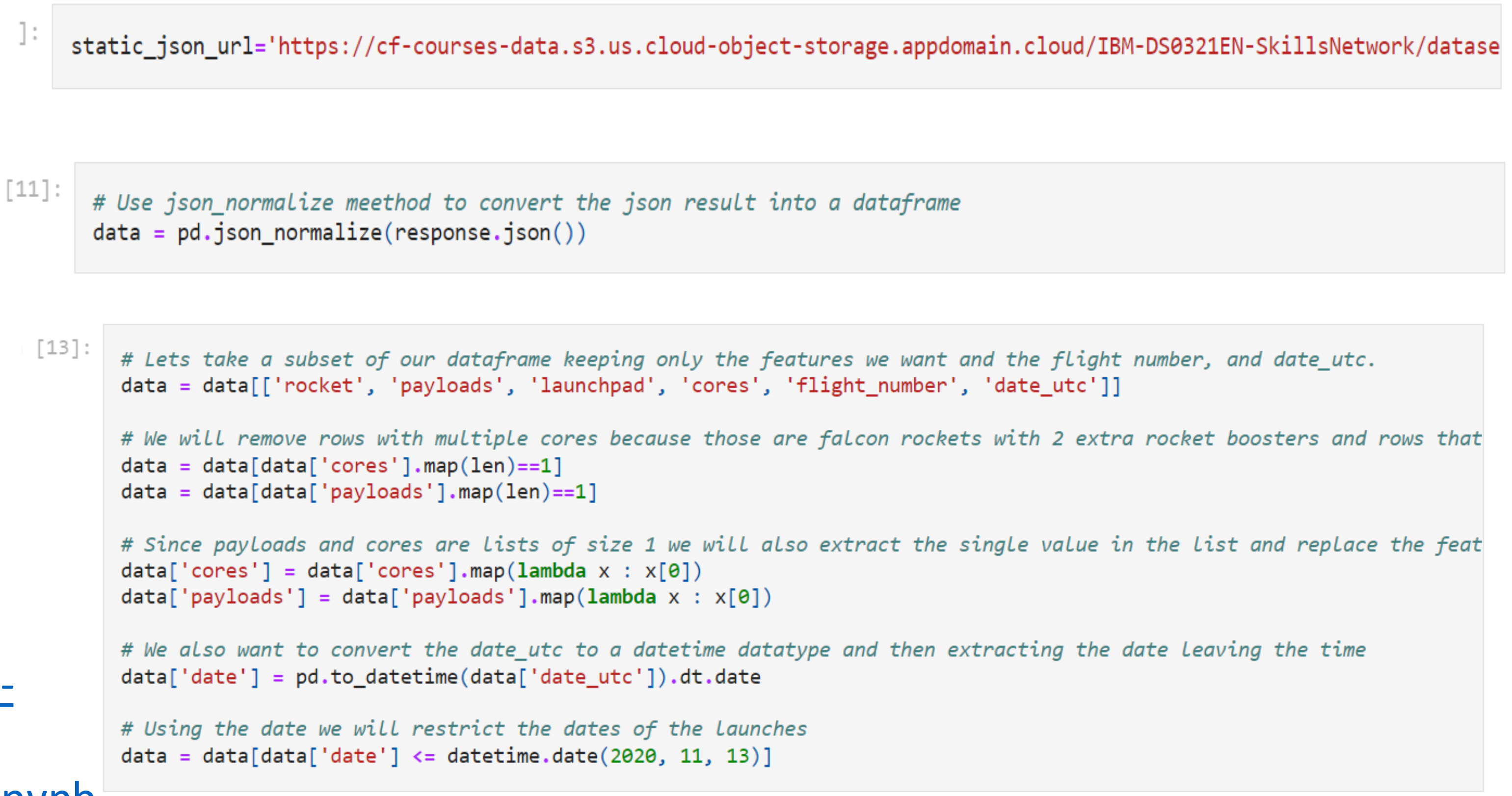
Data collection is the process of gathering and measuring information on specific variables in an established system, allowing one to answer pertinent questions and evaluate outcomes. As previously stated, the dataset was obtained from Wikipedia via REST API and web scraping.

For REST API, it begins with a get request. The response content was then decoded as Json and converted into a pandas dataframe using json normalize (). We then cleaned the data, looked for missing values, and filled in the gaps.

We will use BeautifulSoup for web scraping to extract the launch records as an HTML table, parse the table, and convert it to a pandas dataframe for further analysis.

Data Collection – SpaceX API

- Get request for rocket launch data using API
- Use json_normalize method to convert json result to dataframe
- Performed data cleaning and filling the missing value



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

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

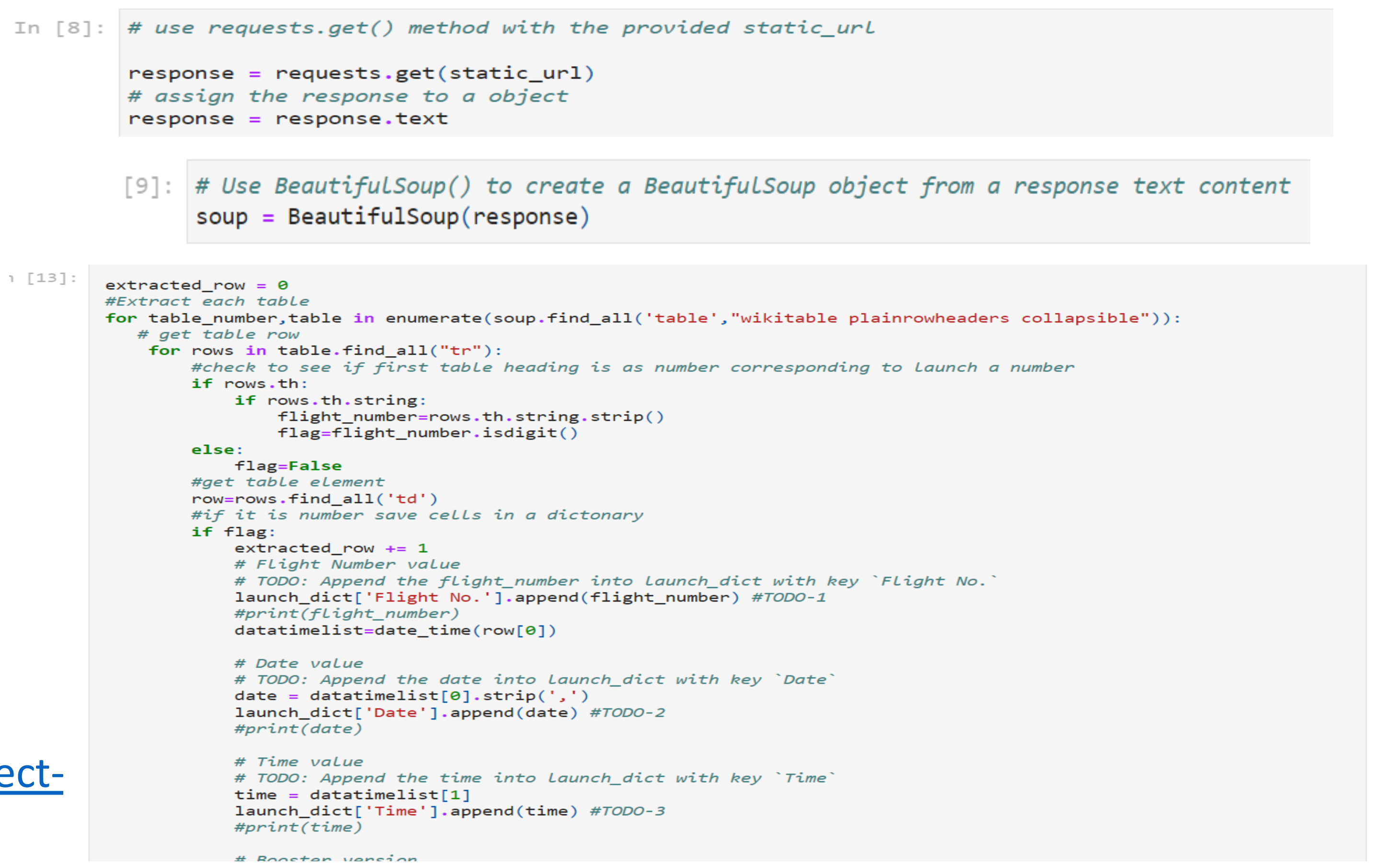
```
[13]:  # Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
       data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

       # We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that
       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 replace the feat
       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 time
       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)]
```

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/Data%20Collection%20API.ipynb

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from url
- Create a BeautifulSoup from the HTML response
- Extract all column/variable names from the HTML header

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/web%20scraping.ipynb



```
In [8]: # use requests.get() method with the provided static_url

        response = requests.get(static_url)
        # assign the response to a object
        response = response.text
```

```
[9]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     soup = BeautifulSoup(response)
```

```
[13]:
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')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number) #TODO-1
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date) #TODO-2
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            launch_dict['Time'].append(time) #TODO-3
            #print(time)

            # Booster version
```

DATA WRANGLING

- The process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis is known as data wrangling (EDA).
- We will first compute the number of launches on each site, followed by the number and frequency of mission outcomes per orbit type.
- The outcome column is then used to generate a landing outcome label. This will make further analysis, visualization, and machine learning easier.

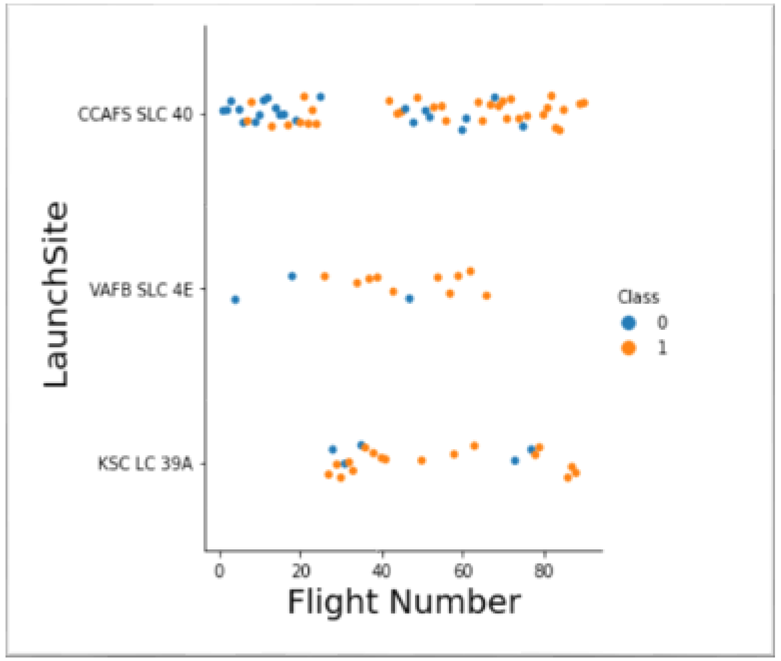
https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/Data%20wrangling%20.ipynb

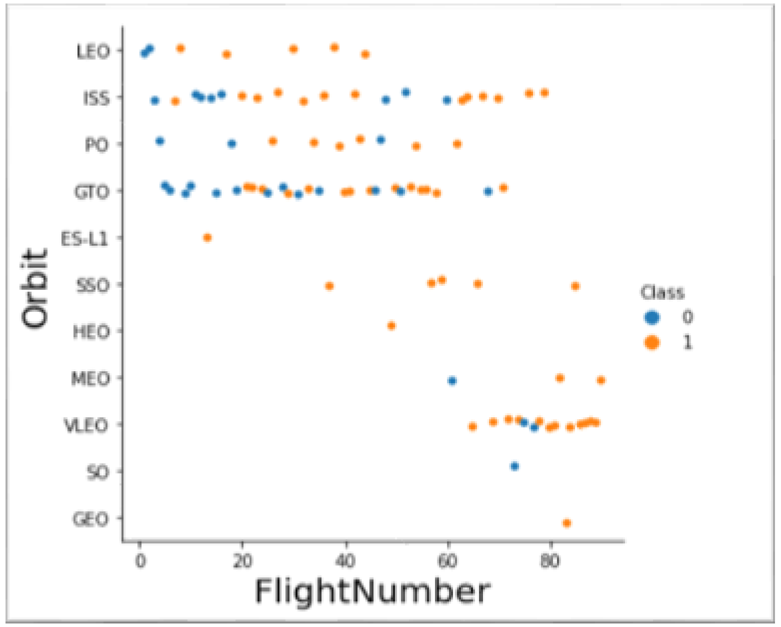
EDA with Data Visualization

- We first started by using scatter graph to find the relationship between the attributes such as between:
- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

- Satter plots demonstrate the interdependence of attributes. Once a pattern is discovered in the graphs. It is very clear which factors have the greatest influence on the success of landing outcomes.

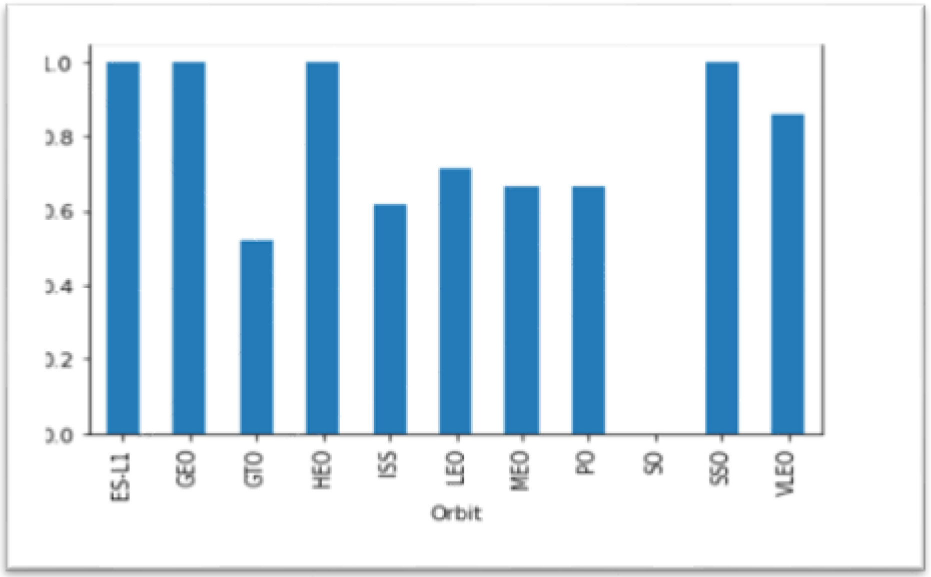
https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/EDA%20with%20Visualization.ipynb

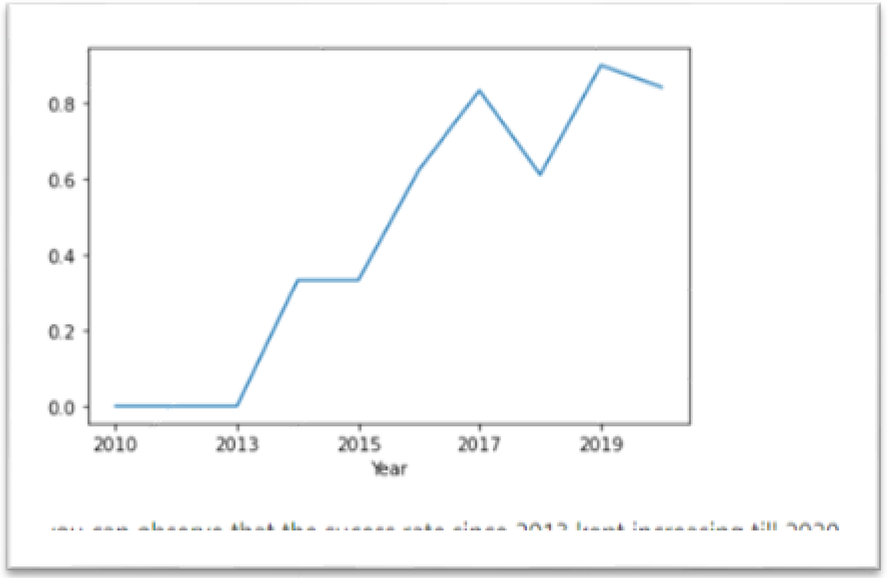




EDA with Data Visualization

- Using the scatter plot, we can get a sense of the relationships. We will then conduct additional analysis using additional visualization tools such as bar graphs and line plots graphs.
- One of the simplest ways to interpret the relationship between attributes is through bar graphs. In this case, the bar graph will be used to determine which orbits have the best chance of success.
- The line graph is then used to show trends or patterns of the attribute over time, which in this case is used to see the launch success yearly trend.
- We then use Feature Engineering to predict success in the future module by converting dummy variables to categorical columns.





https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/EDA%20with%20Visualization.ipynb

EDA with SQL

We ran many SQL queries to gain a better understanding of the dataset, for example:

- Displaying the names of the launch sites.

- Displaying 5 records where launch sites begin with the string 'CCA'.

- Displaying the total payload mass carried by booster launched by NASA (CRS).

- Displaying the average payload mass carried by booster version F9 v1.1.

- Listing the date when the first successful landing outcome in ground pad was achieved.

- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

- Listing the total number of successful and failure mission outcomes.

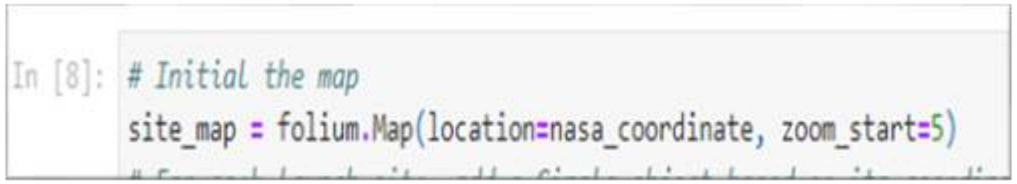
- Listing the names of the booster_versions which have carried the maximum payload mass.

- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.

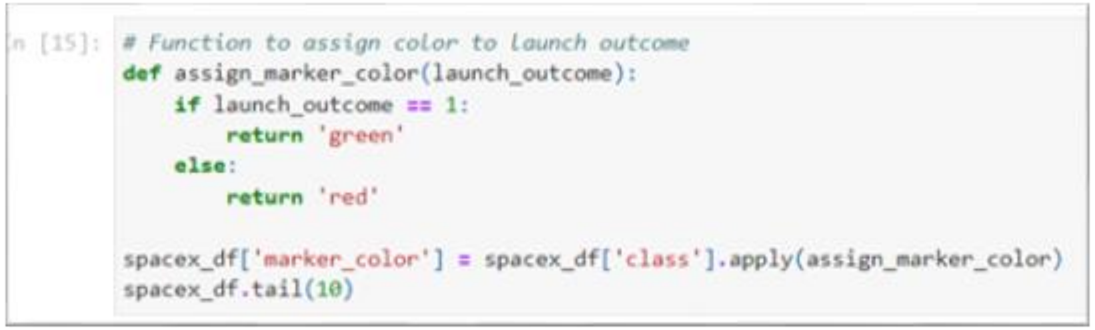
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/EDA%20with%20SQL.ipynb

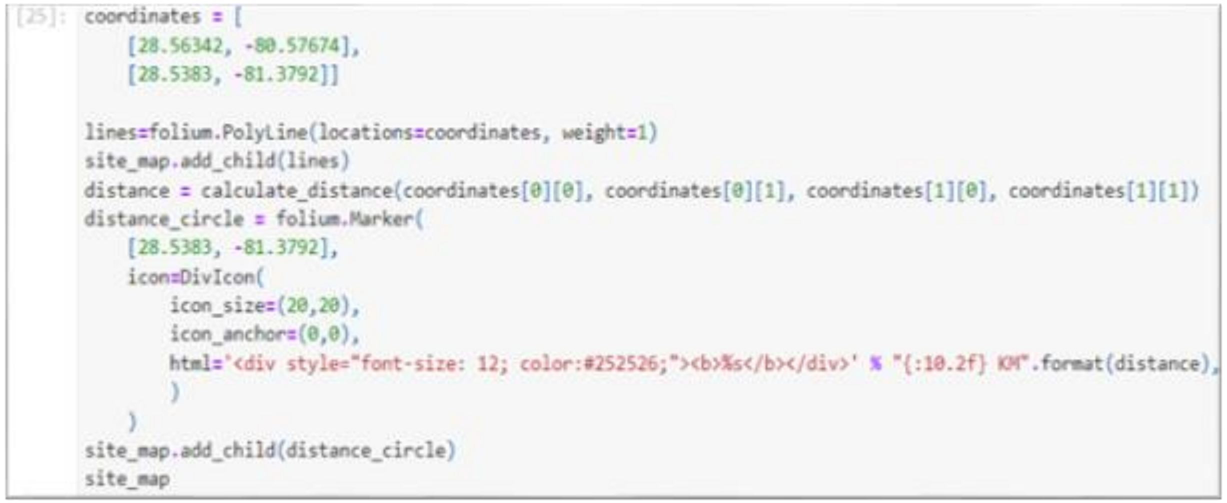
Visual Analytics & Dashboard



```
In [8]: # Initial the map
        site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
```



```
n [15]: # Function to assign color to launch outcome
        def assign_marker_color(launch_outcome):
            if launch_outcome == 1:
                return 'green'
            else:
                return 'red'

        spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
        spacex_df.tail(10)
```



```
[25]: coordinates = [
          [28.56342, -80.57674],
          [28.5383, -81.3792]]

      lines=folium.PolyLine(locations=coordinates, weight=1)
      site_map.add_child(lines)
      distance = calculate_distance(coordinates[0][0], coordinates[0][1], coordinates[1][0], coordinates[1][1])
      distance_circle = folium.Marker(
          [28.5383, -81.3792],
          icon=DivIcon(
              icon_size=(20,20),
              icon_anchor=(0,0),
              html='<div style="font-size: 12; color:#252526;"><b>%s</b></div>' % "{:10.2f} KM".format(distance),
              )
          )
      site_map.add_child(distance_circle)
      site_map
```

Launch Sites Locations

Analysis with Folium:

- Marked all launch sites on a map
- Marked the success/failed launches for each site on the map
- Calculated the distances between a launch site to its proximities

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.

We plotted pie charts showing the total launches by a certain sites.

We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/Dashboard%20plotly.ipynb

Predictive Analysis

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

The model with the best accuracy score will be the best performing model.

https://github.com/panjab1997/IBM-Capstone-Project---Spacex/blob/main/ML.ipynb

Results

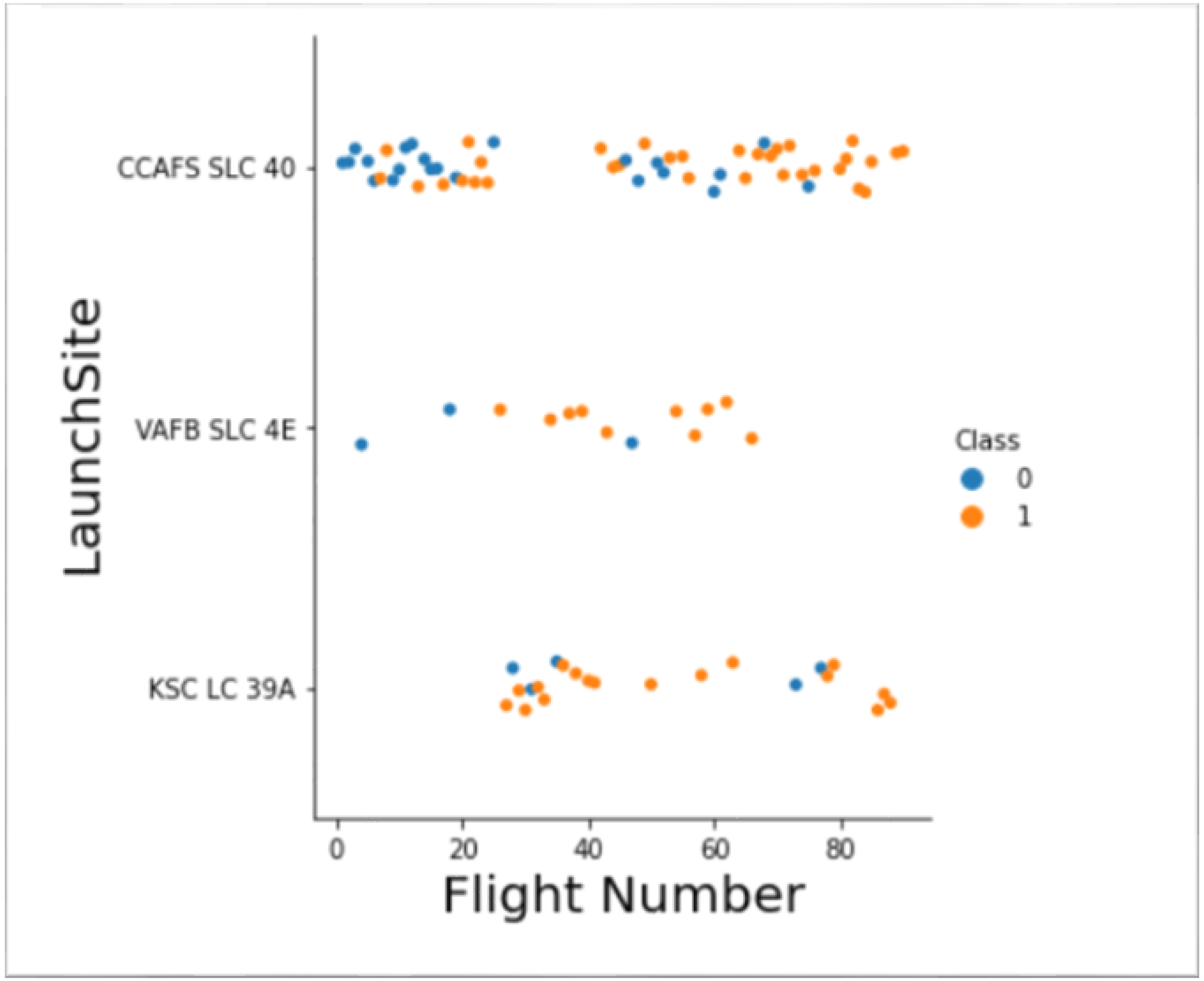
The results will be categorized to 3 main results which is:

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

INSIGHTS FROM EDA

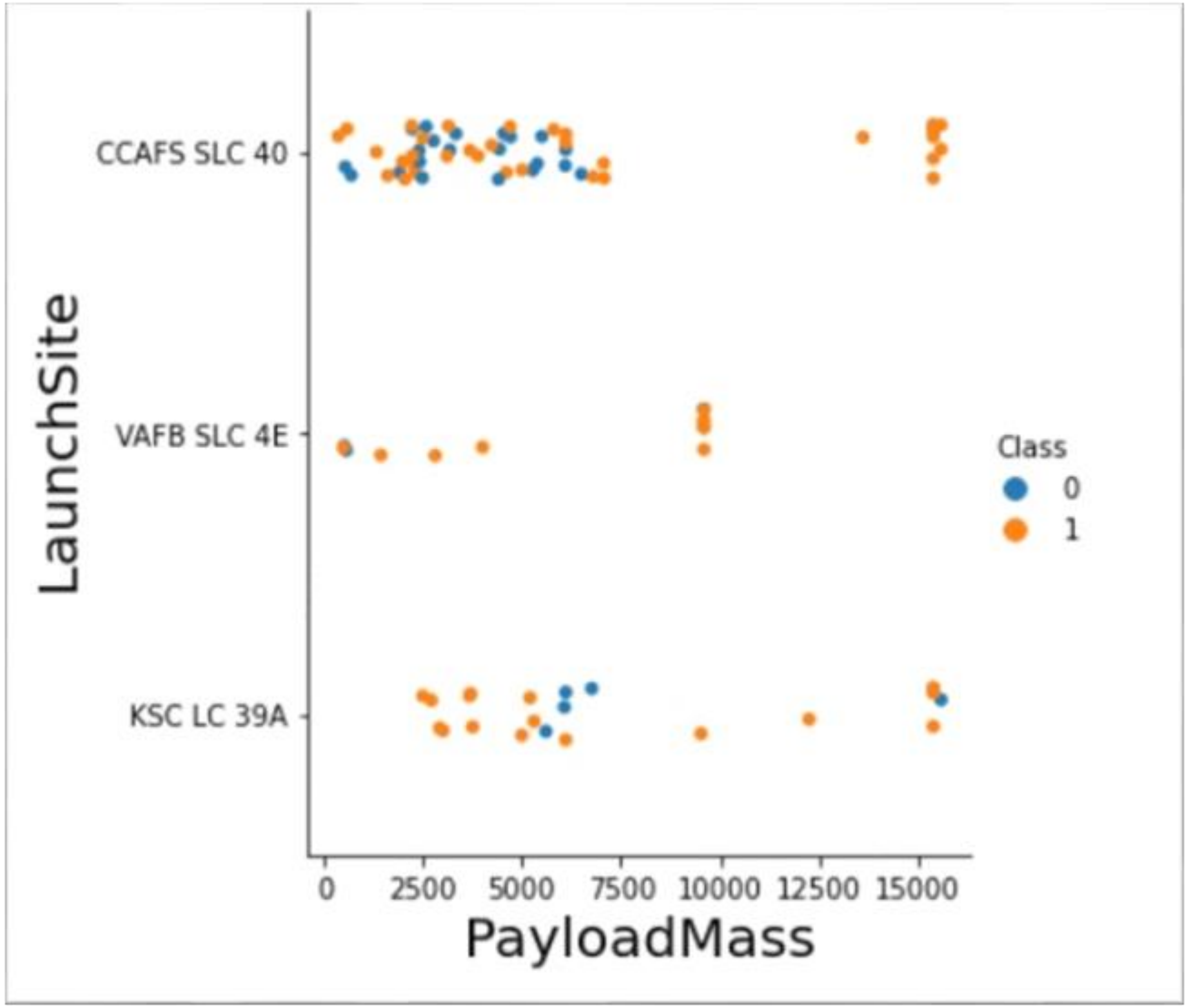
Flight Number vs. Launch Site

- This scatter plot shows that the greater the number of flights from the launch site, the higher the success rate.
- CAFS SLC40, on the other hand, exhibits the least pattern of this.



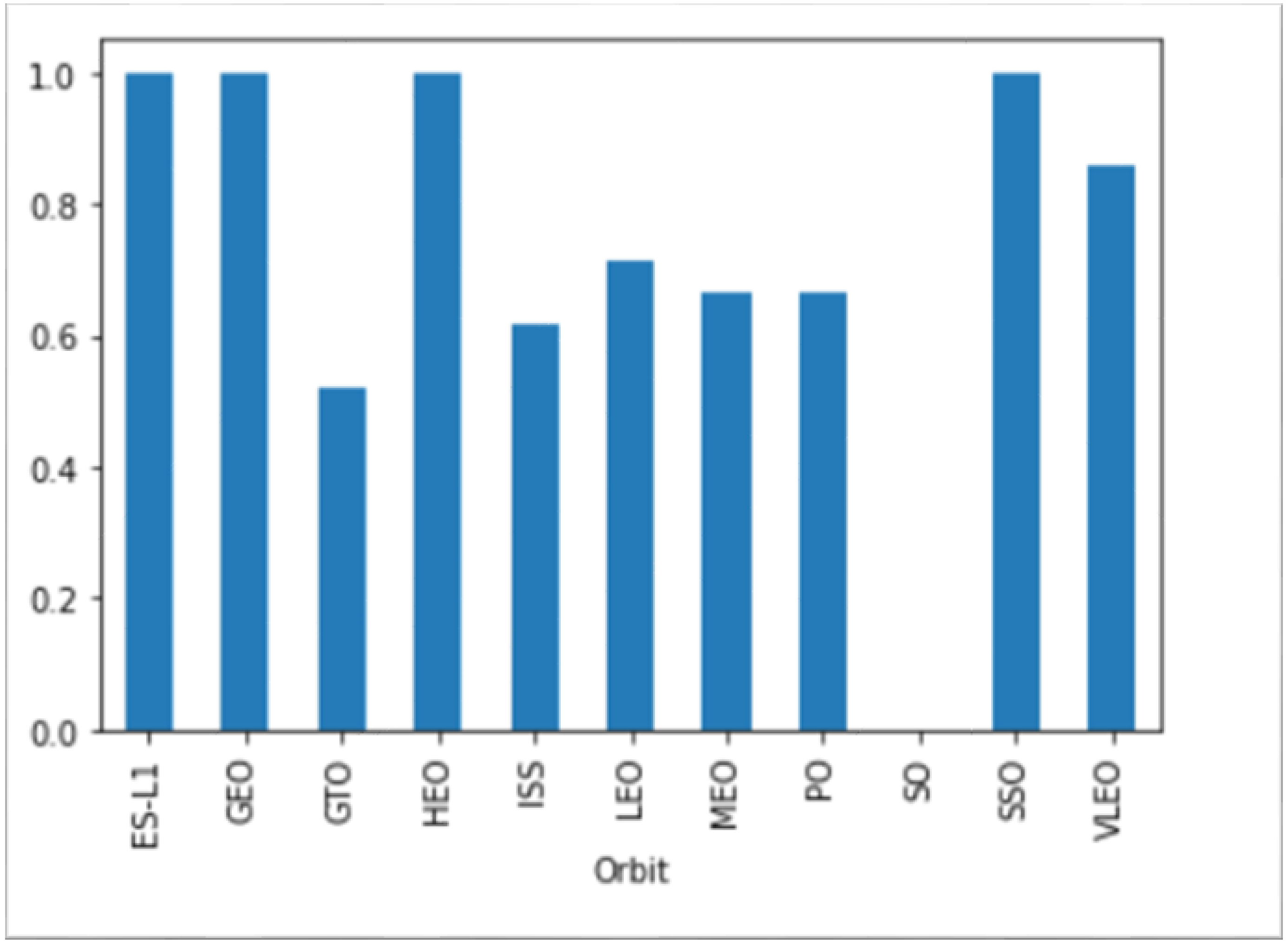
Payload vs. Launch Site

- This scatter plot shows that once the pay load mass exceeds 7000kg, the probability of success increases dramatically.
- However, there is no clear pattern indicating that the success rate of the launch site is dependent on the payload mass.



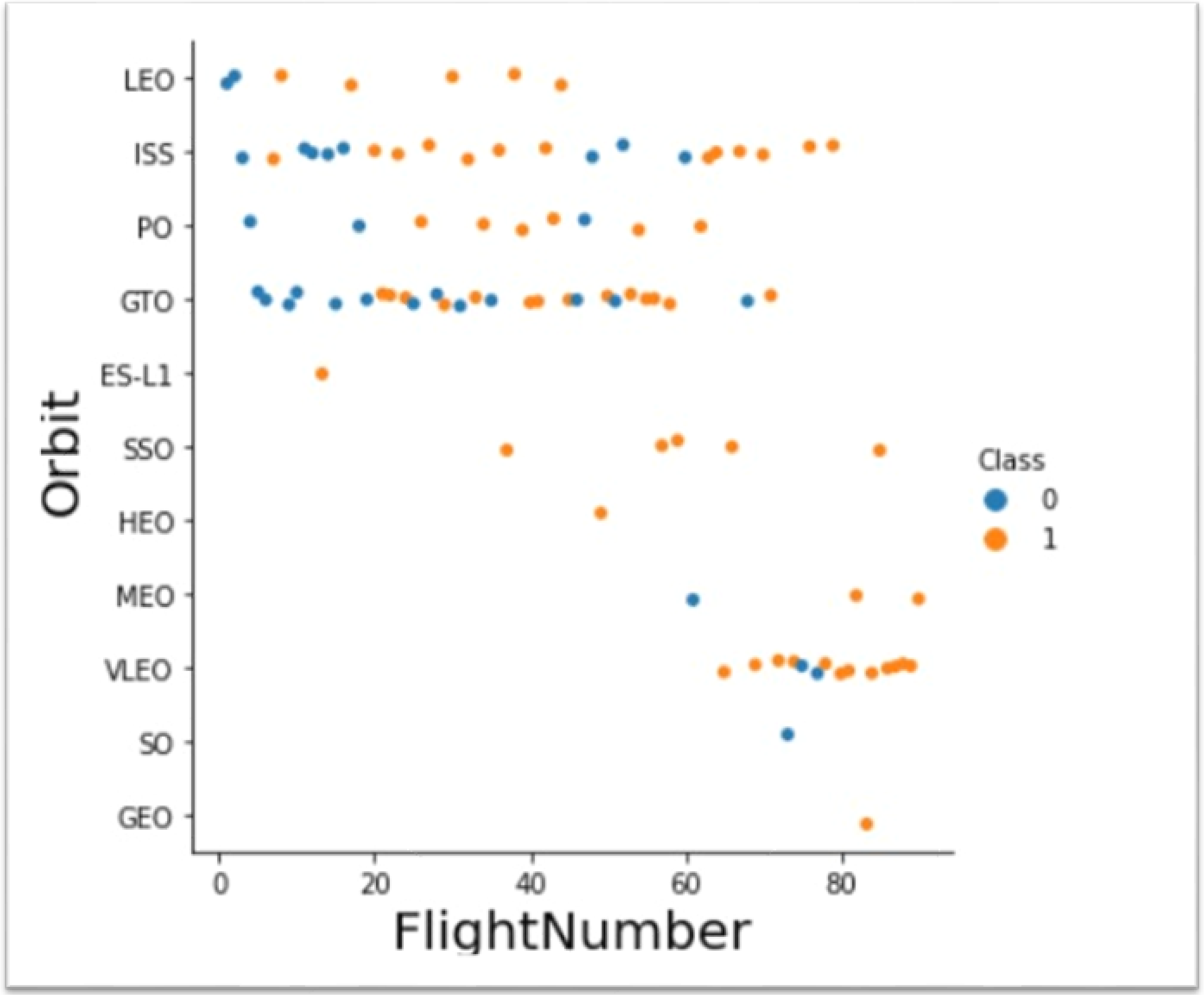
Success Rate vs. Orbit Type

- This bar chart depicted the possibility of orbits influencing landing outcomes, as some orbits have a 100% success rate, such as SSO, HEO, GEO, and ES-L1, while SO orbit has a 0% success rate.
- However, deeper analysis show that some of these orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



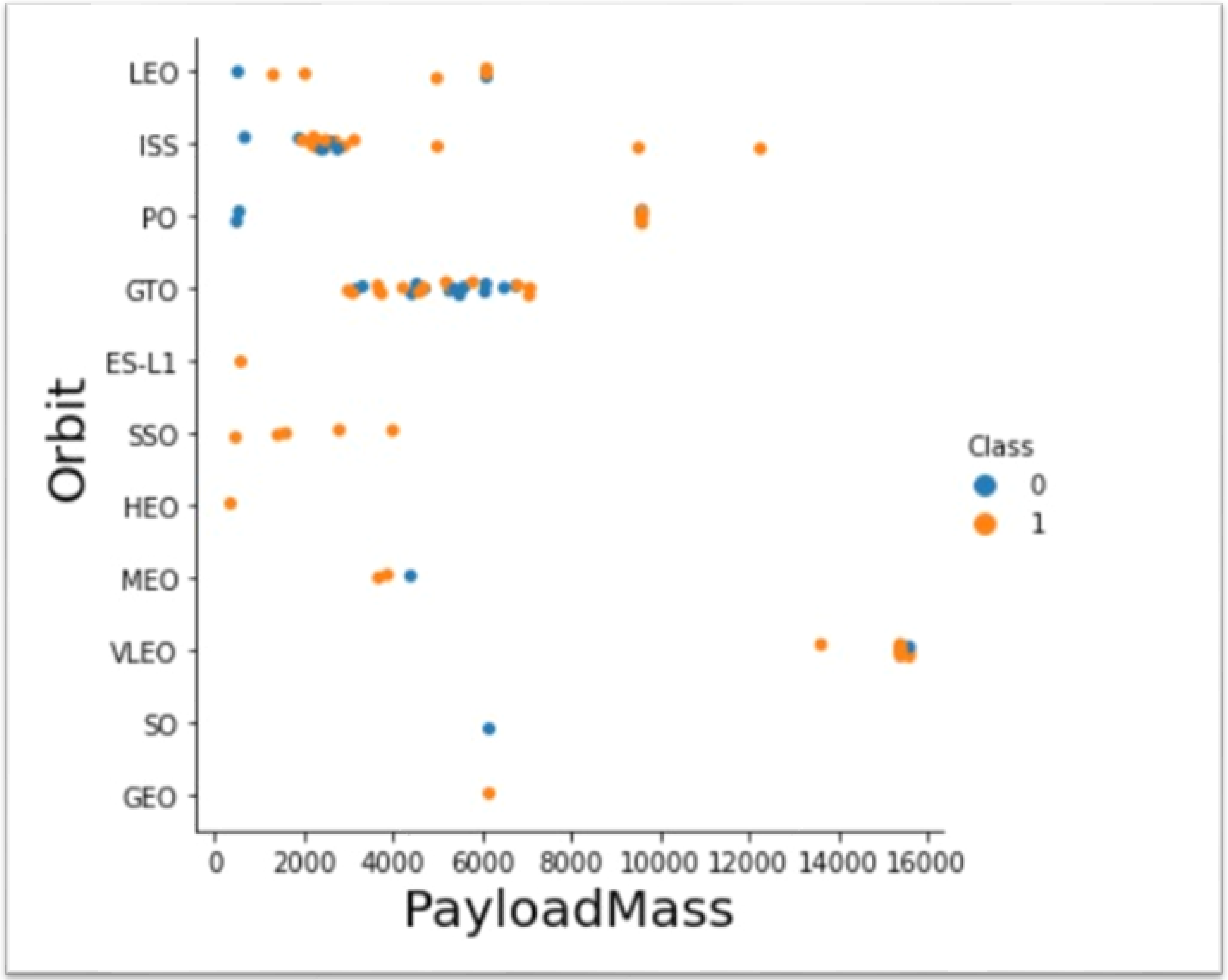
Flight Number vs. Orbit Type

- This scatter graph indicates that, in general, the greater the number of flights on each orbit, the higher the success rate (especially on LEO orbit), except for GTO orbit, which shows no relationship between both attributes.
- Orbits with only one occurrence should also be excluded from the preceding statement, as more data is required.



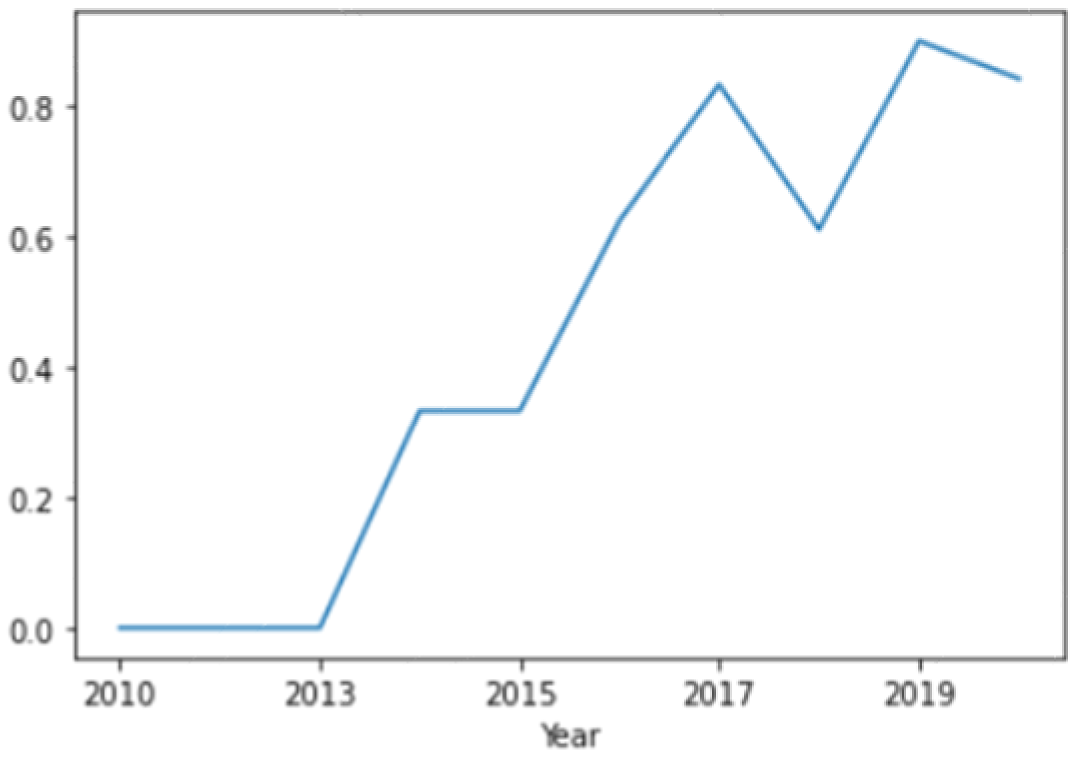
Payload vs. Orbit Type

- A heavier payload benefits LEO, ISS, and P0 orbit.
- It does, however, have a negative impact on MEO and VLEO orbit.
- The GTO orbit appears to show no relationship between the attributes.
- Meanwhile, the SO, GEO, and HEO orbits require more data to detect any pattern or trend.



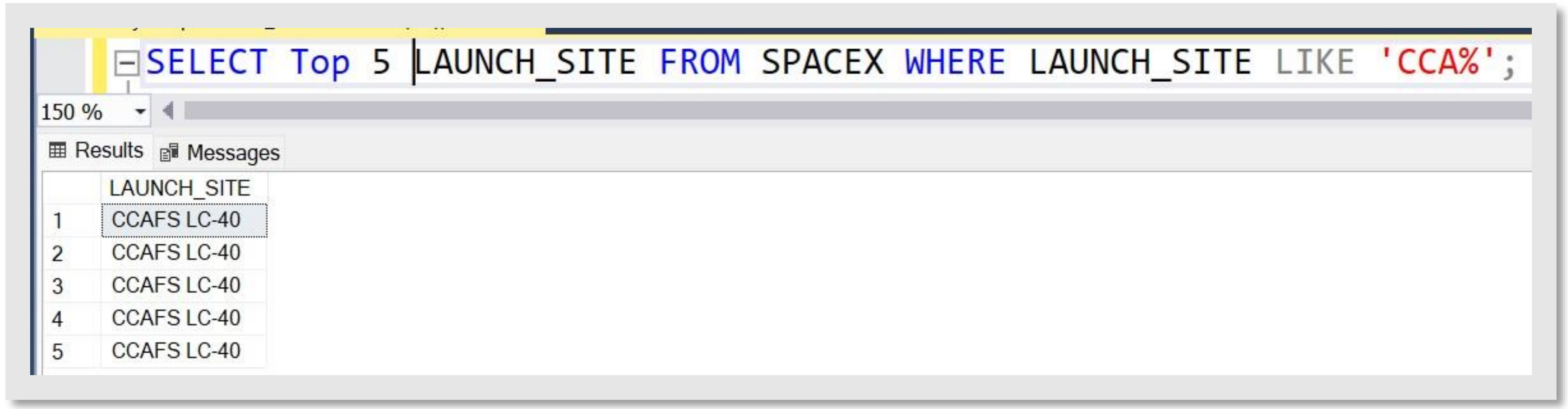
Launch Success(Yearly)

- If this trend continues in the coming year. The success rate will gradually increase until it reaches 100%.



you can observe that the sucess rate since 2013 kept increasing till 2020

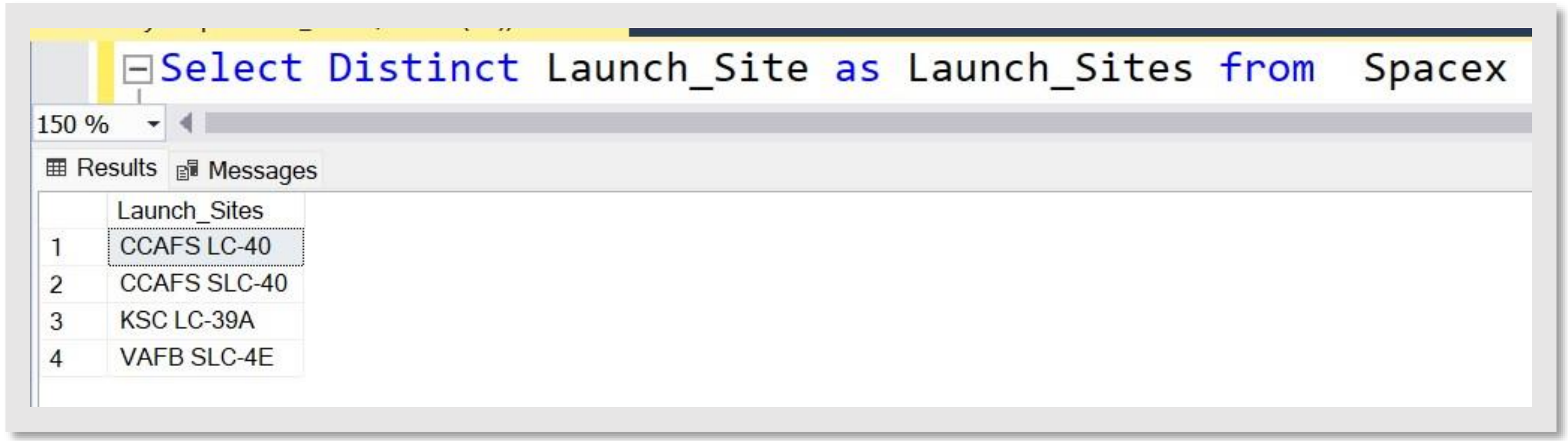
Launch Site Names Begin with 'CCA'



```
SELECT Top 5 LAUNCH_SITE FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%';
```

	LAUNCH_SITE
1	CCAFS LC-40
2	CCAFS LC-40
3	CCAFS LC-40
4	CCAFS LC-40
5	CCAFS LC-40

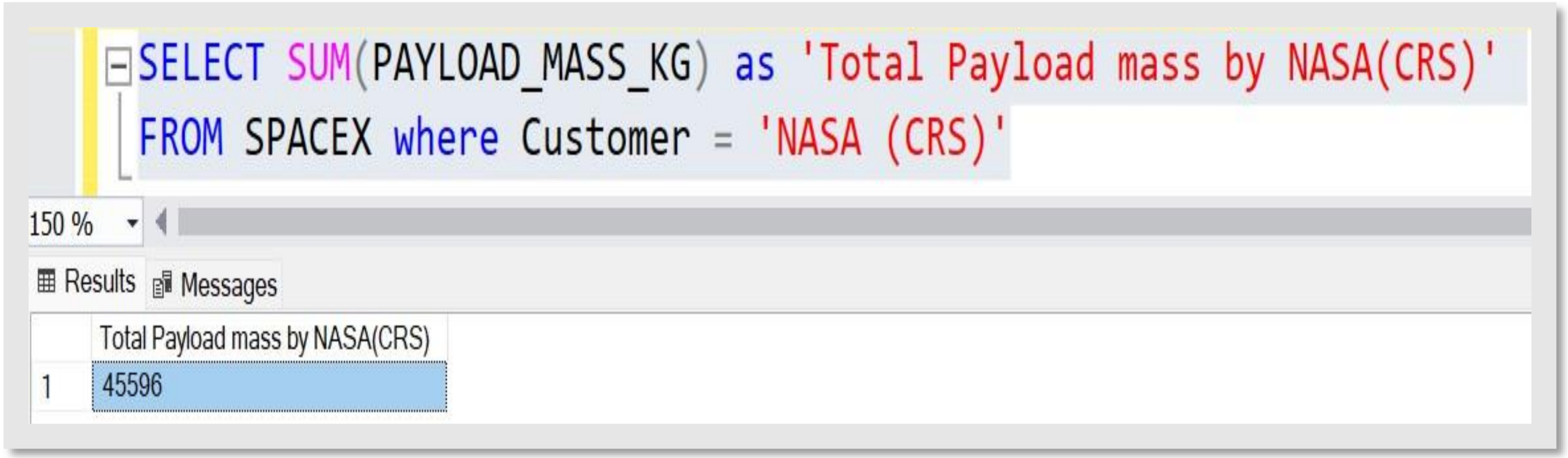
All Launch Site Names



```
Select Distinct Launch_Site as Launch_Sites from  Spacex
```

	Launch_Sites
1	CCAFS LC-40
2	CCAFS SLC-40
3	KSC LC-39A
4	VAFB SLC-4E

Total Payload Mass



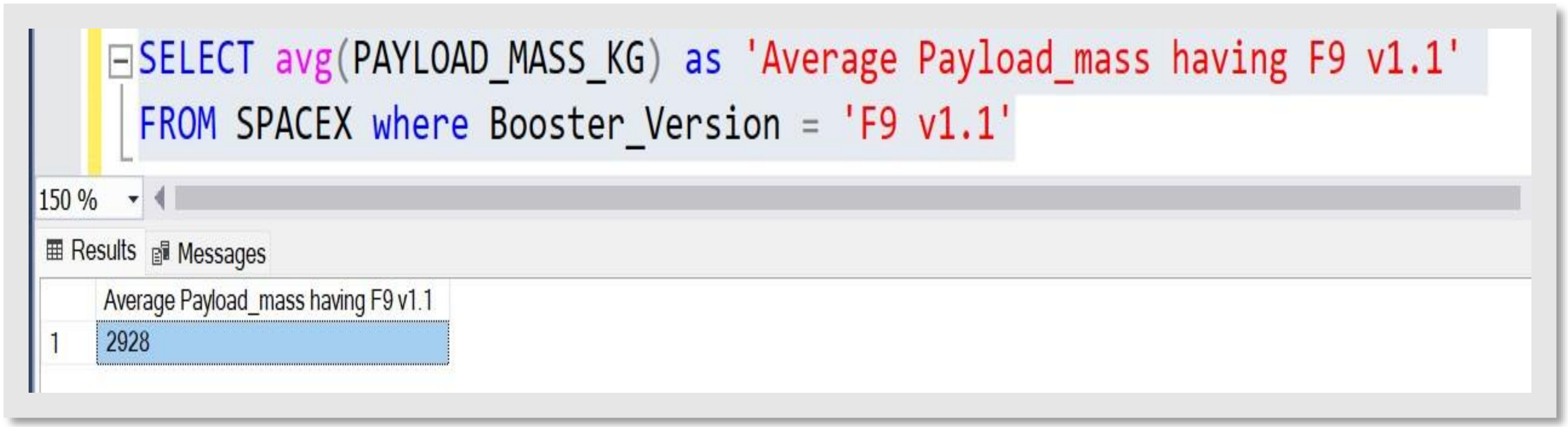
```
SELECT SUM(PAYLOAD_MASS_KG) as 'Total Payload mass by NASA(CRS)'
FROM SPACEX where Customer = 'NASA (CRS)'
```

150 %

Results | Messages

	Total Payload mass by NASA(CRS)
1	45596

Average Payload Mass by F9 v1.1



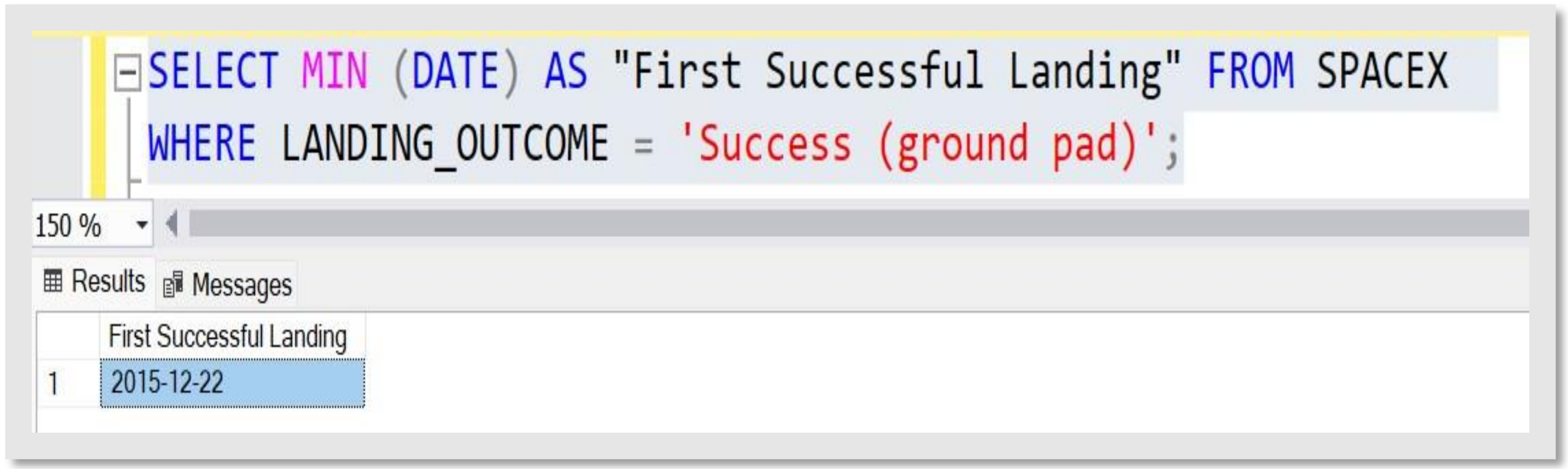
```
SELECT avg(PAYLOAD_MASS_KG) as 'Average Payload_mass having F9 v1.1'
FROM SPACEX where Booster_Version = 'F9 v1.1'
```

150 %

Results | Messages

	Average Payload_mass having F9 v1.1
1	2928

First Successful Ground Landing Date



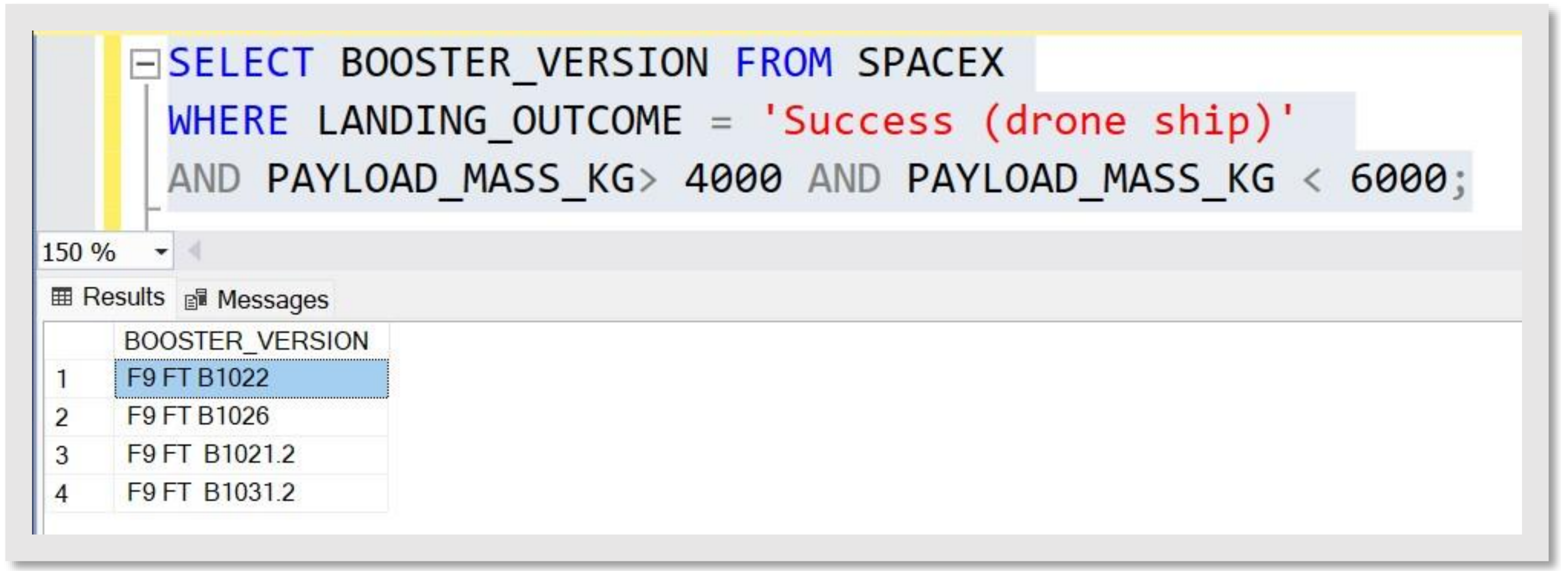
```
SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEX
WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

150 %

Results Messages

```
   First Successful Landing
1  2015-12-22
```

Successful Landing with Payload between 4000 and 6000



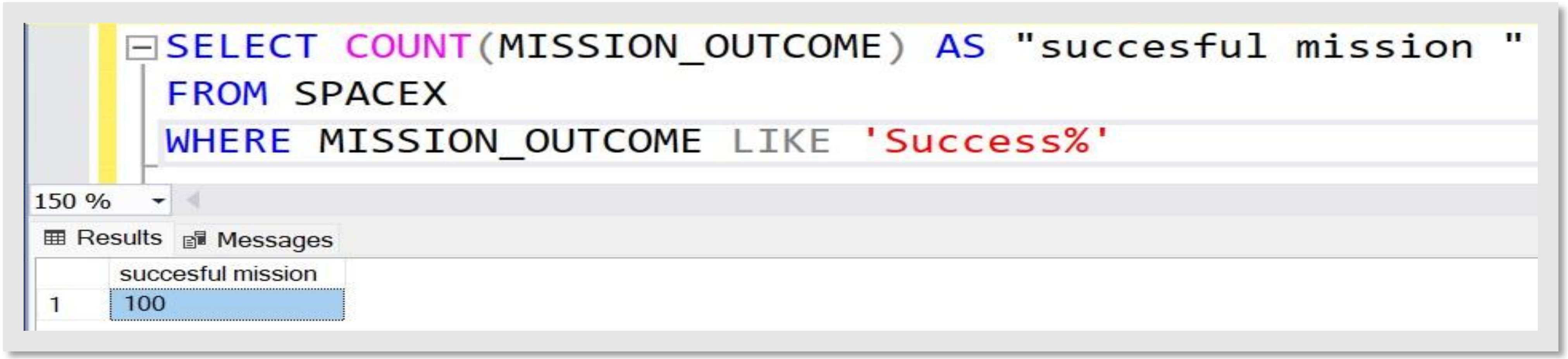
```
SELECT BOOSTER_VERSION FROM SPACEX
WHERE LANDING_OUTCOME = 'Success (drone ship)'
AND PAYLOAD_MASS_KG> 4000 AND PAYLOAD_MASS_KG < 6000;
```

150 %

Results Messages

```
   BOOSTER_VERSION
1  F9 FT B1022
2  F9 FT B1026
3  F9 FT  B1021.2
4  F9 FT  B1031.2
```

Total Number of Successful and Failure Mission Outcomes

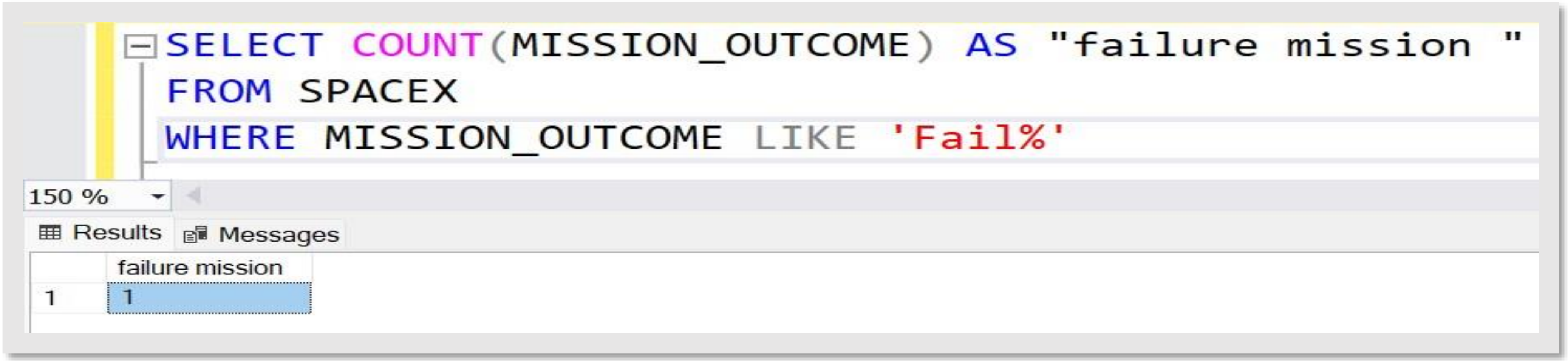


```
SELECT COUNT(MISSION_OUTCOME) AS "succesful mission "
FROM SPACEX
WHERE MISSION_OUTCOME LIKE 'Success%'
```

150 %

Results Messages

```
   succesful mission
1  100
```



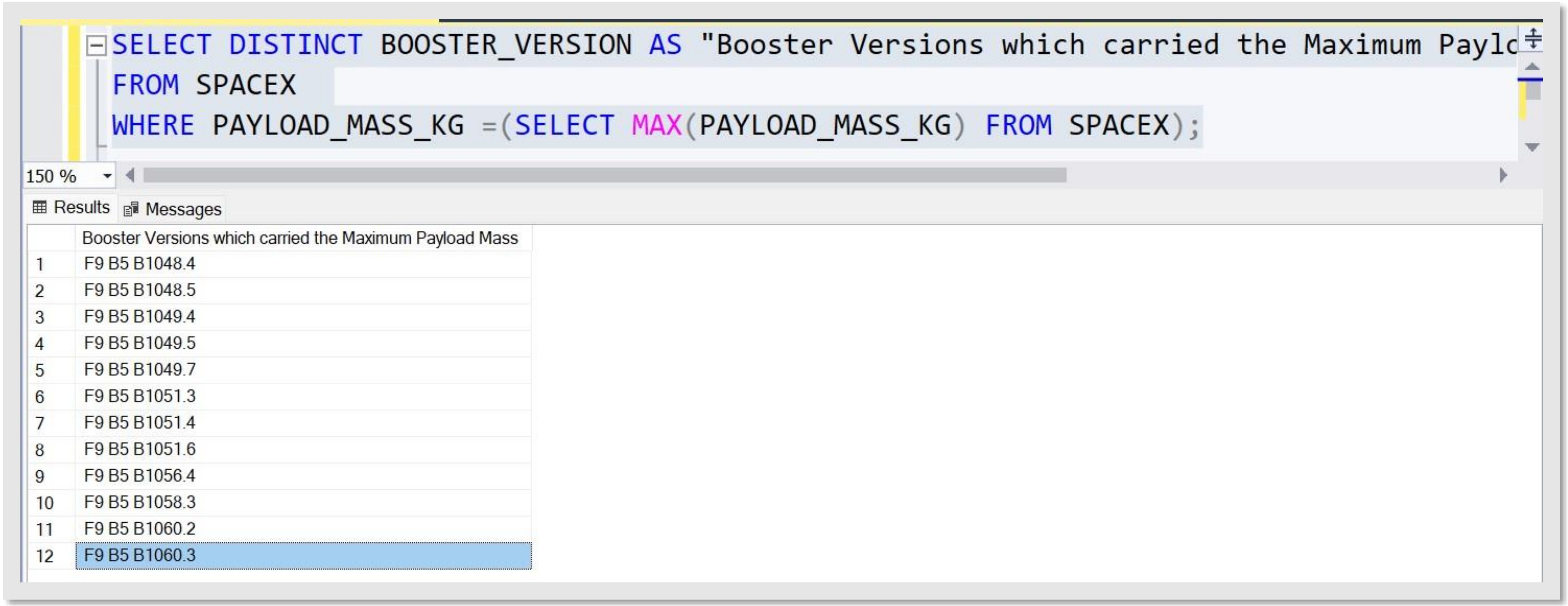
```
SELECT COUNT(MISSION_OUTCOME) AS "failure mission "
FROM SPACEX
WHERE MISSION_OUTCOME LIKE 'Fail%'
```

150 %

Results Messages

```
   failure mission
1  1
```

Boosters Carried Maximum Payload



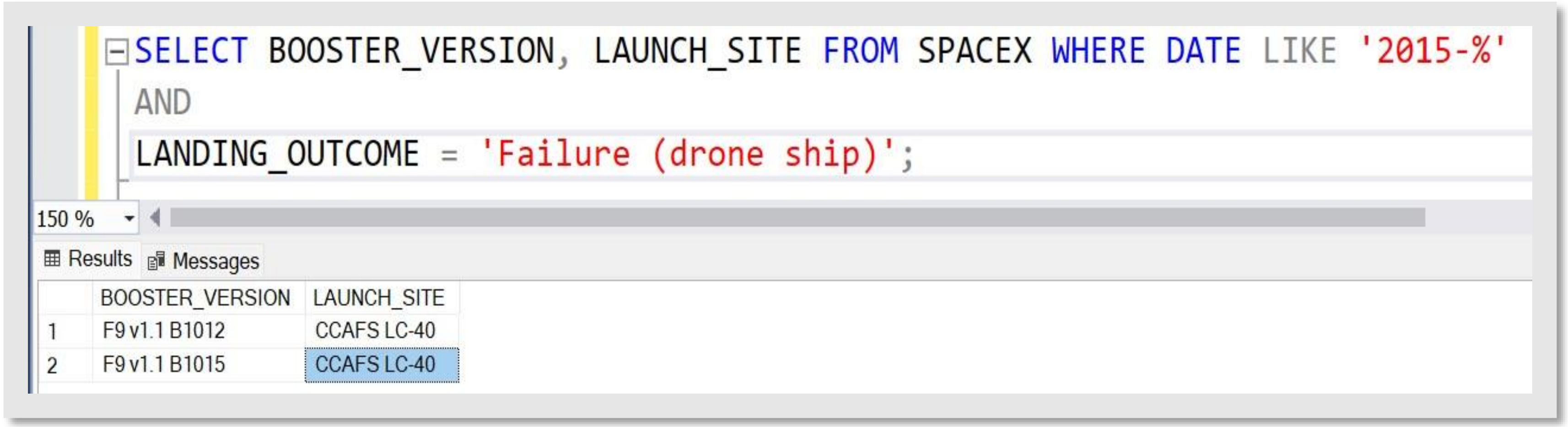
```
SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Paylc
FROM SPACEX
WHERE PAYLOAD_MASS_KG =(SELECT MAX(PAYLOAD_MASS_KG) FROM SPACEX);
```

150 %

Results | Messages

	Booster Versions which carried the Maximum Payload Mass
1	F9 B5 B1048.4
2	F9 B5 B1048.5
3	F9 B5 B1049.4
4	F9 B5 B1049.5
5	F9 B5 B1049.7
6	F9 B5 B1051.3
7	F9 B5 B1051.4
8	F9 B5 B1051.6
9	F9 B5 B1056.4
10	F9 B5 B1058.3
11	F9 B5 B1060.2
12	F9 B5 B1060.3

2015 Launch Records



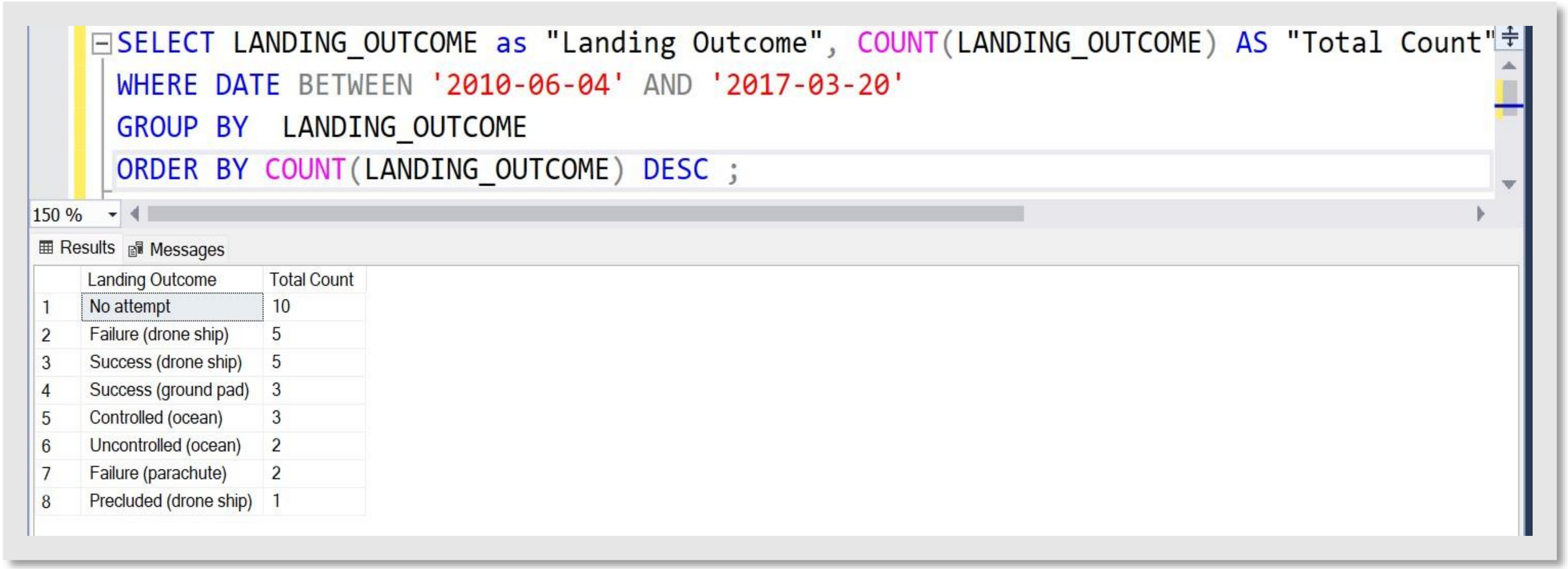
```
SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%'
AND
LANDING_OUTCOME = 'Failure (drone ship)';
```

150 %

Results | Messages

	BOOSTER_VERSION	LAUNCH_SITE
1	F9 v1.1 B1012	CCAFS LC-40
2	F9 v1.1 B1015	CCAFS LC-40

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



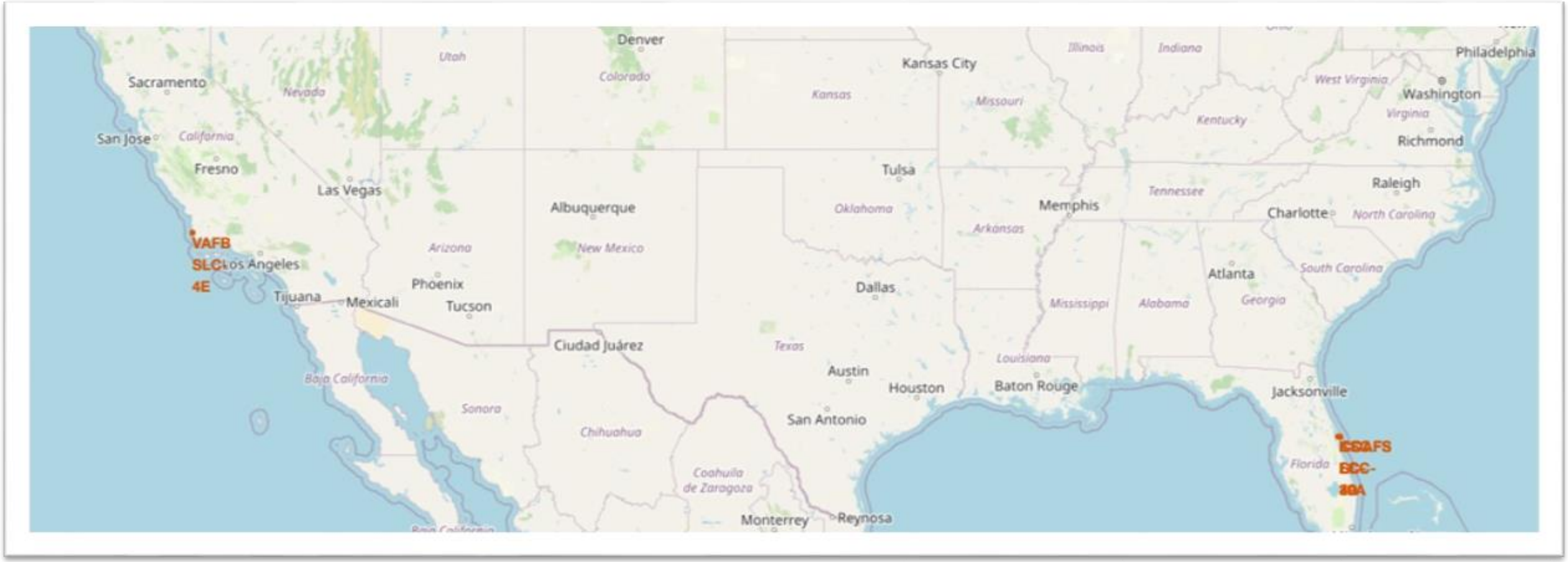
```
SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count"
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY  LANDING_OUTCOME
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

150 %

Results | Messages

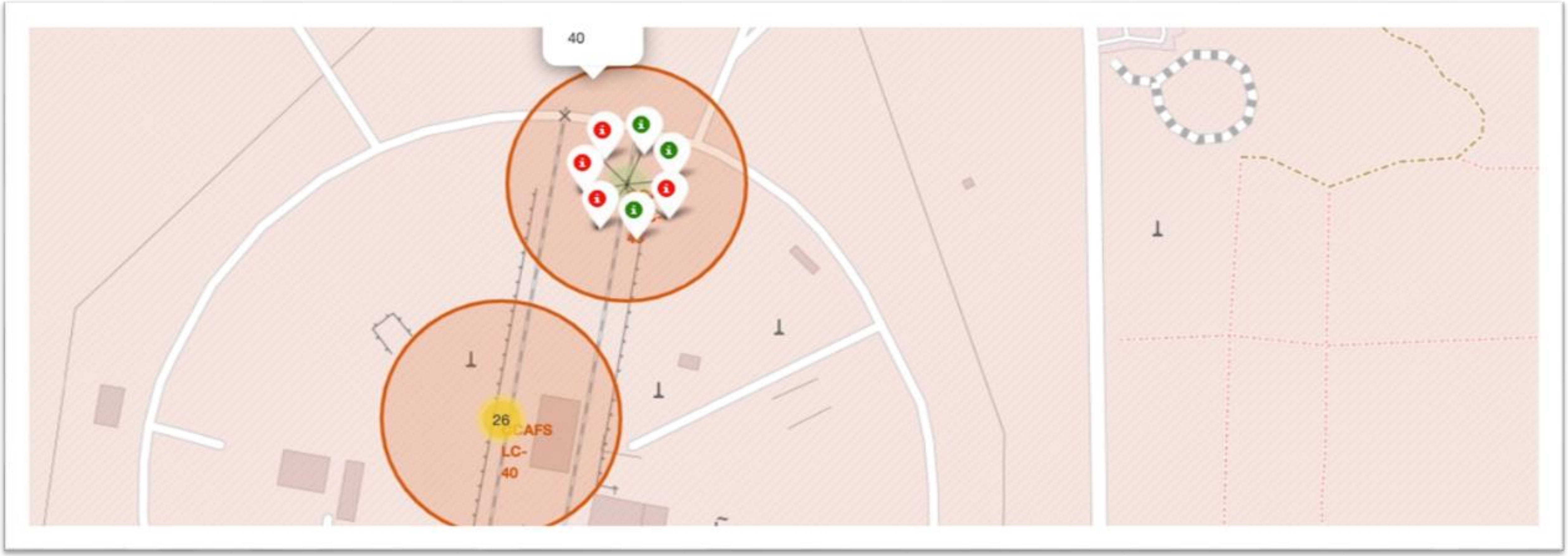
	Landing Outcome	Total Count
1	No attempt	10
2	Failure (drone ship)	5
3	Success (drone ship)	5
4	Success (ground pad)	3
5	Controlled (ocean)	3
6	Uncontrolled (ocean)	2
7	Failure (parachute)	2
8	Precluded (drone ship)	1

Interactive Map with Folium



Location of all the Launch Sites

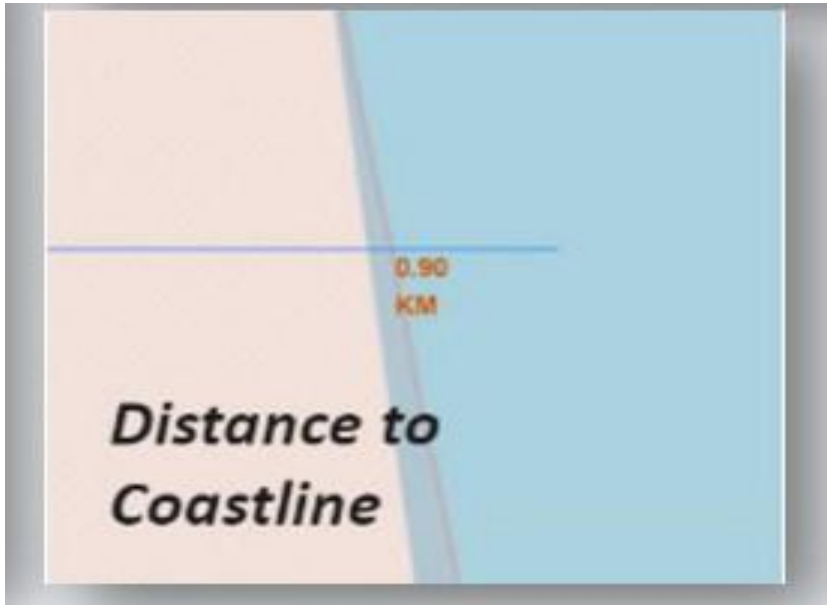
- We can see that all the SpaceX launch sites are located inside the United States

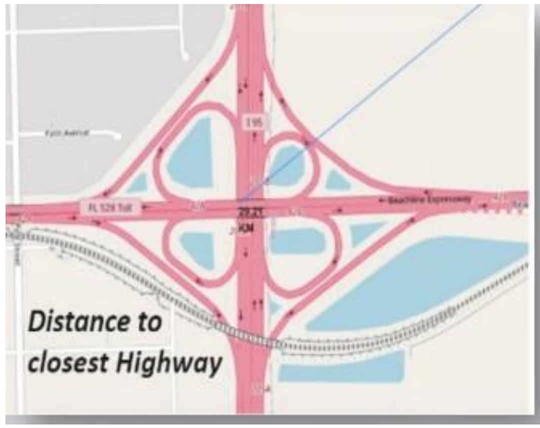


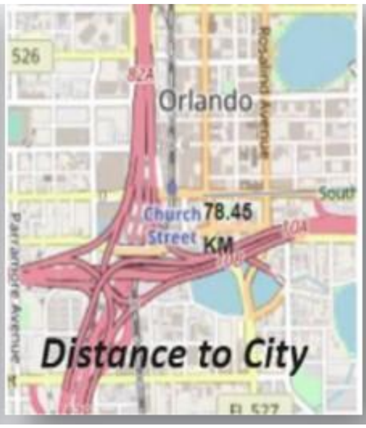
Markers showing Success & Failure

- Red color indicates failure and Green indicates Succes

Distance from Launch Sites to Landmarks







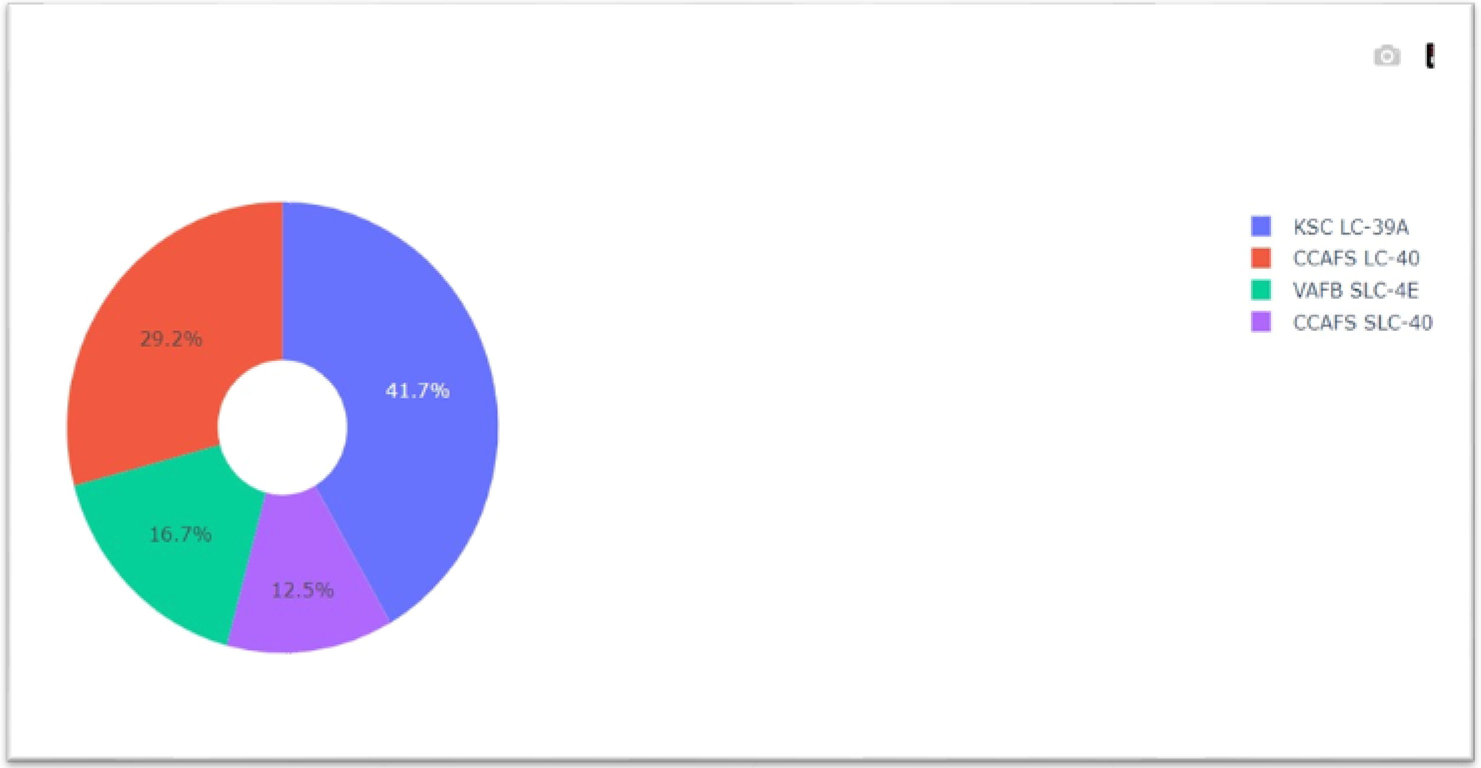
7
SLC-40
26
LC-40
Distance to coast

- Are launch sites in close proximity to railways? **NO**
- Are launch sites in close proximity to highways? **NO**
- Are launch sites in close proximity to coastline? **YES**
- Do launch sites keep certain distance away from cities? **YES**

Interactive Dashboard with Ploty Dash

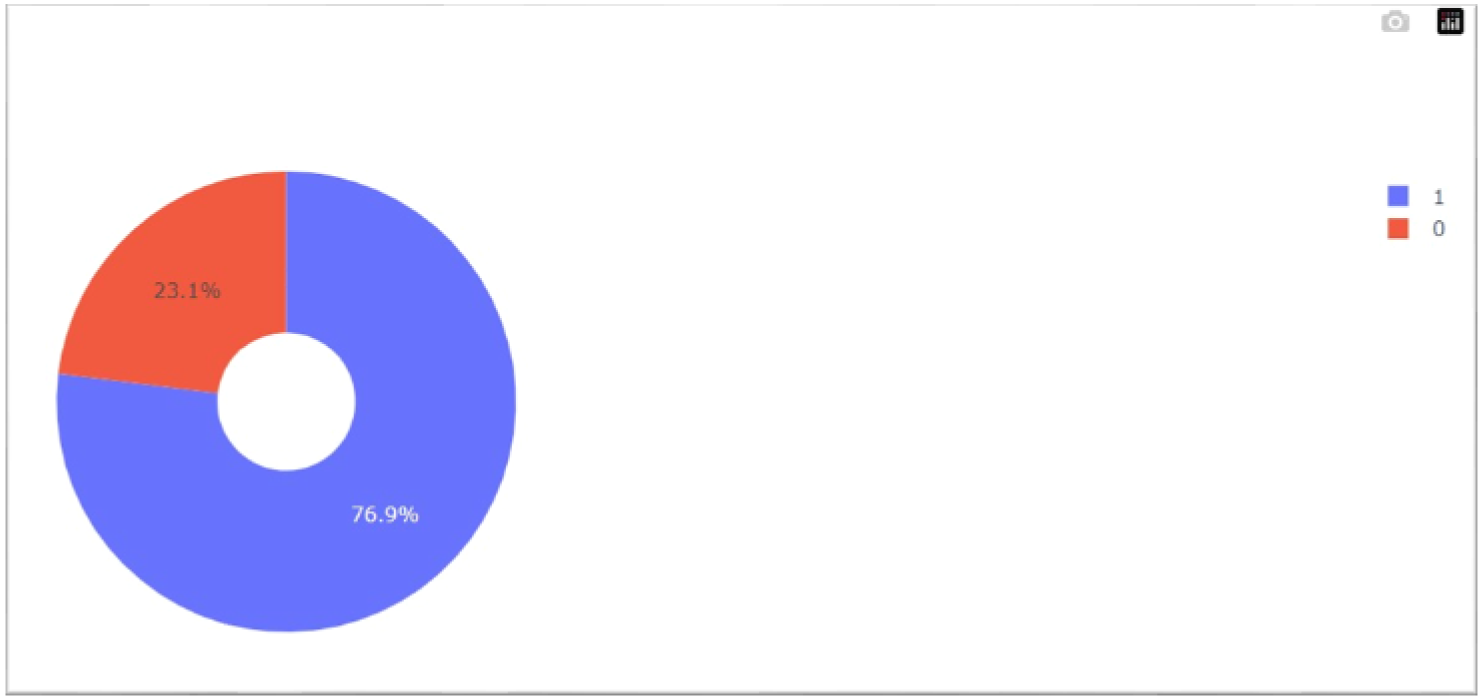
Total Successful Launch by all sites

As you can see KSC LC-39A has maximum number of lunches i.e. 41.7%



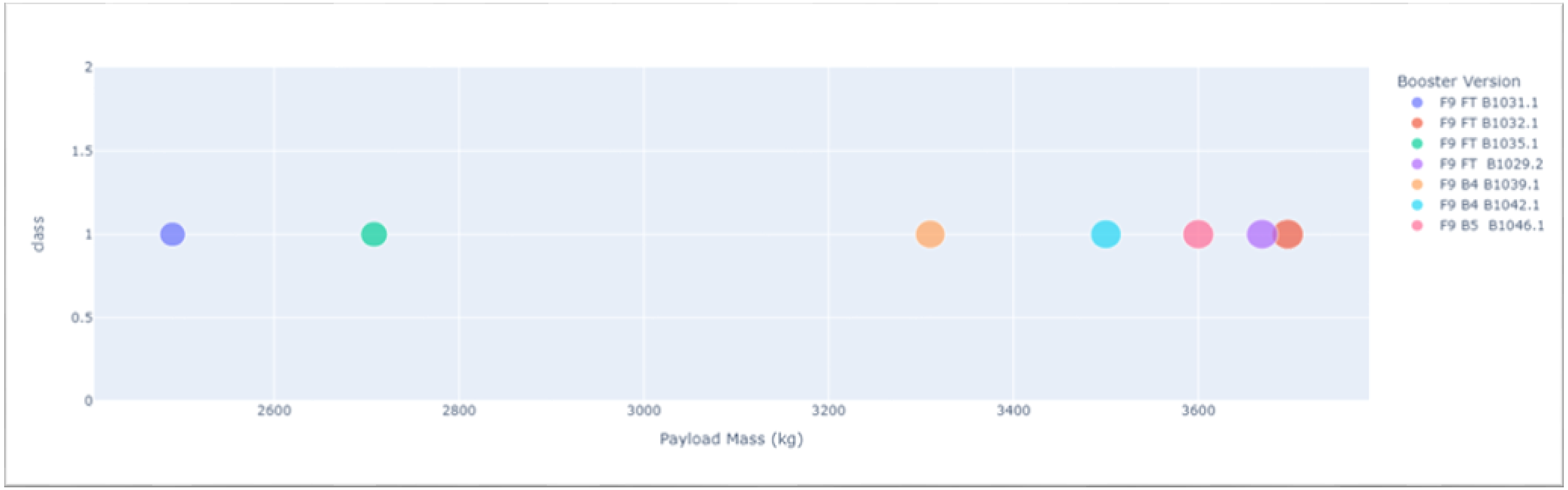
The highest launch-success ratio: KSC LC-39A

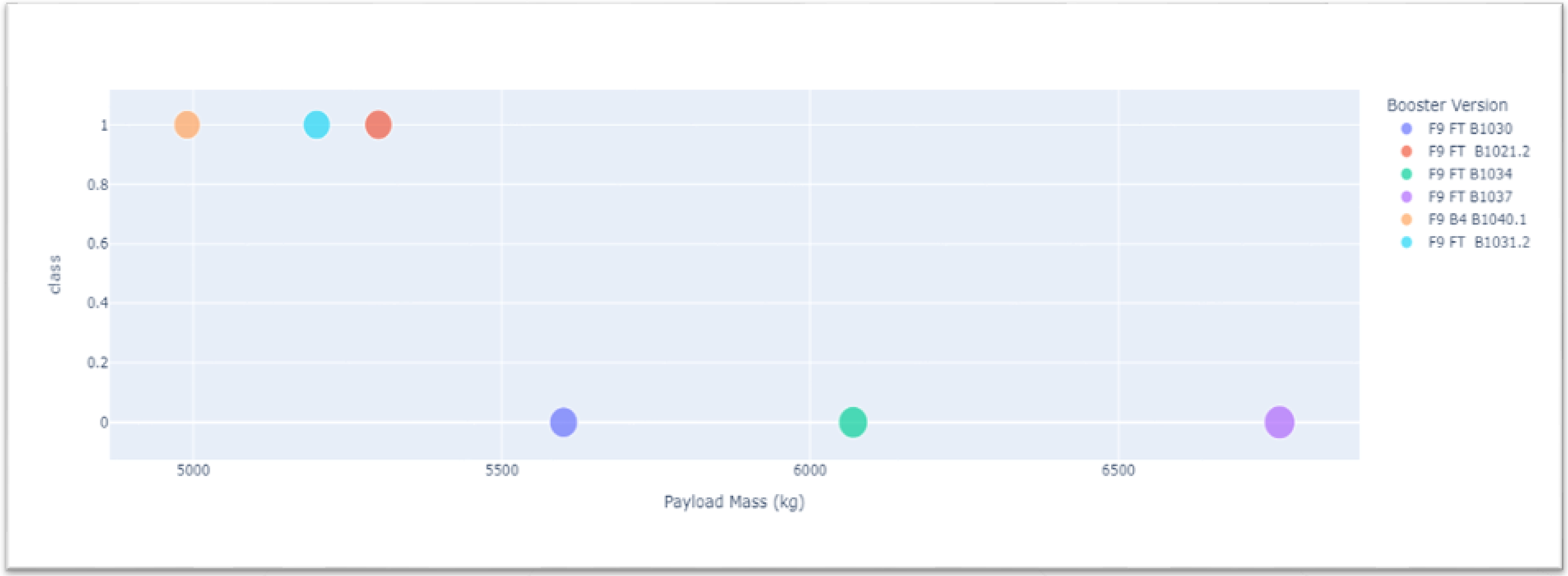
- KSC LC-39A has 76.9% success rate and 23.1% failure



Payload vs Launch Outcome

- Payload Mass in range 0 to 4000



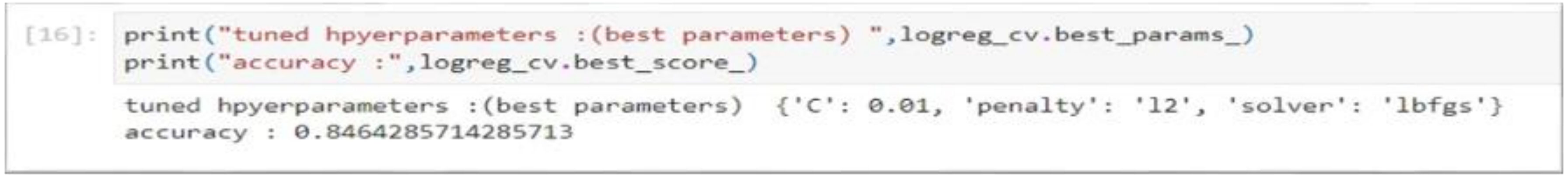


- **Payload Mass above 4000**

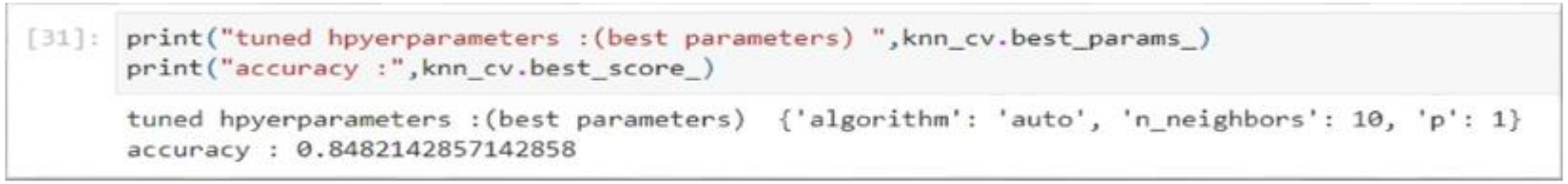
Payload vs Launch Outcome

PREDICTIVE ANALYSIS

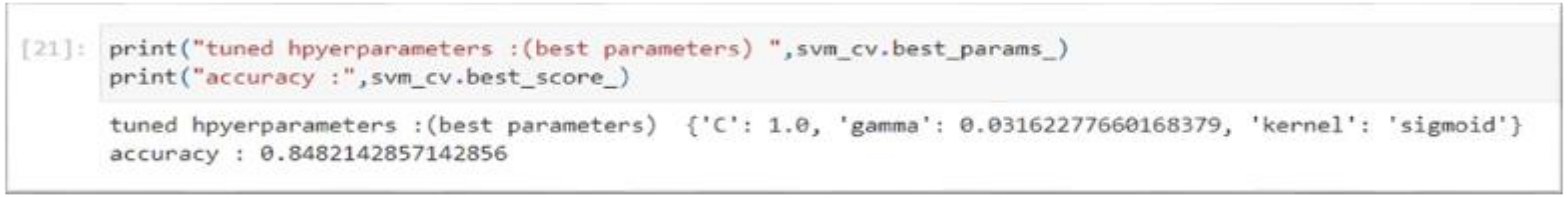
Classification Accuracy



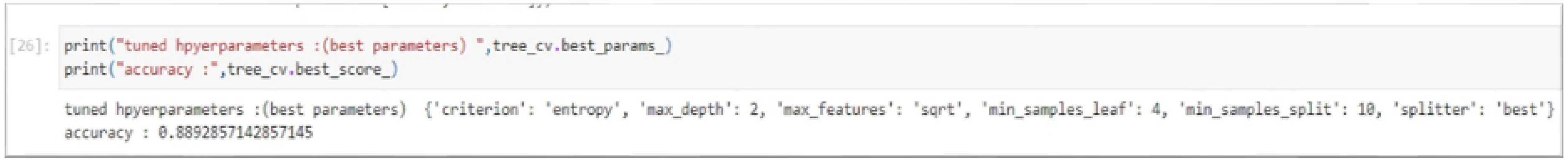
```
[16]: print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
      print("accuracy :",logreg_cv.best_score_)

      tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
      accuracy : 0.8464285714285713
```



```
[31]: print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
      print("accuracy :",knn_cv.best_score_)

      tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
      accuracy : 0.8482142857142858
```



```
[21]: print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
      print("accuracy :",svm_cv.best_score_)

      tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
      accuracy : 0.8482142857142856
```

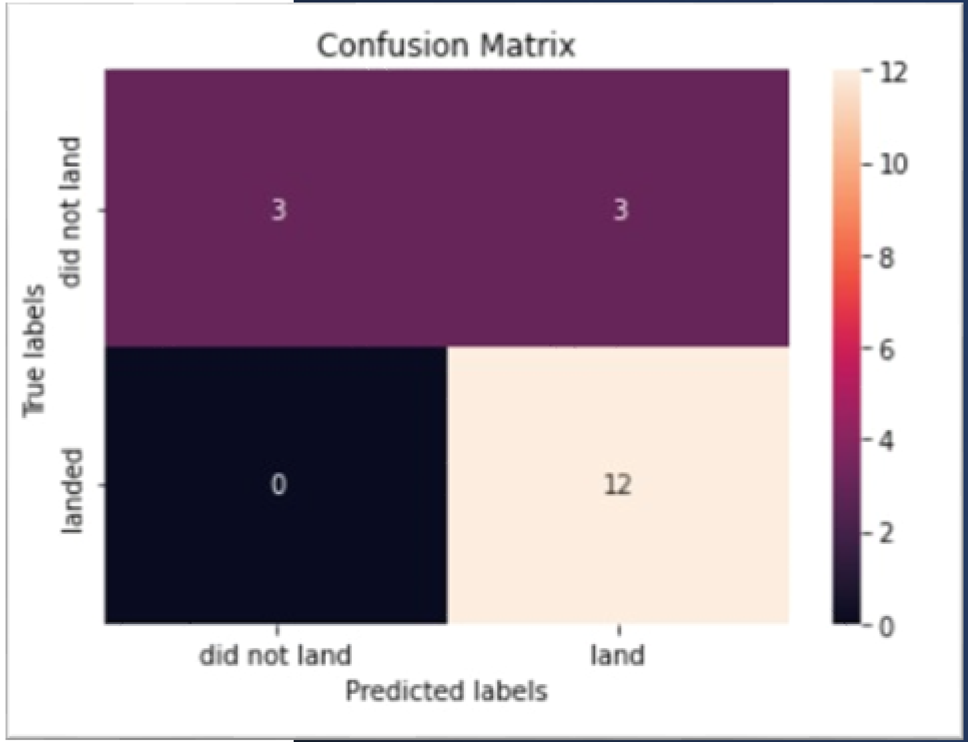


```
[26]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
      print("accuracy :",tree_cv.best_score_)

      tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'best'}
      accuracy : 0.8892857142857145
```

- As we can see, by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy

Confusion Matrix

- The confusion matrix for the decision tree classifier demonstrates that the classifier can differentiate between the various classes. The main issue is false positives. i.e., the classifier considers an unsuccessful landing to be a successful landing.



- For this dataset, the Tree Classifier Algorithm is the best Machine Learning approach.
- Low weighted payloads (defined as 4000kg and below) outperformed heavy weighted payloads.
- Since 2013, the success rate of SpaceX launches has increased in direct proportion to the time elapsed between 2013 and 2020, with the goal of eventually perfecting the launches in the future.
- KSC LC-39A has had the most successful launches of any site, with 76.9% success rate.
- he SSO orbit has the highest success rate; 100% with more than one occurrence.

Conclusions

THANK YOU!