Final Project

Prediction of SpaceX launch success

IBM Data Science Professional

Certification

Promise Emoghene

06/11/2023

Executive Summary

- The purpose of this project is to predict if the Falcon 9 first stage will land **successfully**. 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. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- I would use the get request python library to get the correct datasets directly from the spaceX API, Perform data wrangling using some some python lbrary like Pandas, Seaborn and Numpy and save the data sets to the sqllite database as Mydata1.
- I will use SQLlite DBMS for Python(sql magic) to query the Mydata1 data base and do major explorative data analysis.
- I would use some popular python visualization library such as matplotlib, Lunch Analysis with folium and Dash interactive dashbourds with plotly to check for trends in the data sets.
- The explorative data Analysis carried out on the data sets indicate that independence variables such as FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount and Serial have strong influence on the predictive variables
- I would utilize some popular sklearn tools such as preproccesing, GridsearchCV, DescisionTreeClassifier,LogisticRegression,KNeighboursClassifier ,Train_Test_Split as well as SupportVectorMachine for selecting the right machine learning algorithm for the prediction.
- I used the F1 test, Accuracy, Precision and confusion metrix to evaluate the model created and all model performed perfectly well with an average accuracy of 84% Success.

Introduction

The birth of space travel in october 4th 1957 opened a large door of innovations to companies who render services that is connected to space travel. The early stories of space travel Required millions of dollars and many businesses go into loss when there is a failed attempt and worse case may not even get licence from the licencing bodies in cases high casualties. These stories has been almost thesame in mordern times except that modern prediction techniques is now been implemented to ensure a greater percent of success during lunch attempt. Virgin Galactic is providing suborbital spaceflights, Rocket Lab is a small satelite provider and Blue Origin manufactures sub-orbital and orbital reusable rockets. The most successful and affordable of them all is SpaceX And their accomplishments include: Sending spacecraft to the International Space Station. The reason behind their success could be attributed to the fact that most of their launches are relatively cheap for example SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

The project would give an answer to some of these questions:

1. What is the price of each Launch?
2. Would spaceX reuse the first stage?
3. How does the feature columns for example payload mass, launch sites affect the launch success?
4. Which F9 Booster version has the highest launch success rate?

Methodology

- Utilize the python request library to get the correct datasets from the spaceX API and the spaceX wikipedia page
- Make use of some python helper functions such as get function which make use of the spaceX API to extract information using identification numbers in the launch data.
- Uttilize the isnull() function to know the sum of null value in the data set and decide on the best approach (replace the null value with mean of variables in the column containing the null values).
- Import the sklearn preprocessing and standardisation tools to process the datasets for machine learning algorithm.
- Utilize some of the popular python library such as pandas, seaborn and numpy as well as the SQLmagic functionality in python for explorative data analysis (EDA).
- Visualize the spaceX data set and use python Dash components to build an interactive dashboard.
- Utilize the various SKlearn tools to select and fit predictive models such as GridsearchCV DescisionTreeClassifier,LogisticRegression,KNeighboursClassifier ,Train_Test_Split as well as SupportVectorMachine for selecting the right machine learning algorithm for the prediction.

DATA COLLECTION OVERVIEW

- The data set where collected from spaceX public API and spaceX wikipedia page.
- The following columns: Flight number, date, booster version, payload mass, orbit, launch site, result, fights, GridFins, Reused, Legs, Landing Pad, Block, Reused Count, Series, Longitude and Latitude where collected from the SpaceX public API using the Python request object.
- The other columns such as Flight number, launch site, payload, payload mass, orbit, client, launch result, booster version, boosterlanding as well as date and time where webscraped from the SpaceX wikipedia page
- The next slide uses a flow chart to descrbe the steps of data collection

Data Collection – SpaceX Public API

1. Obtained and parsed the SpaceX launch data.

2. Normalize the JSON response into a data frame.

3. Extract only useful columns using auxiliary functions.

4. Create a new Pandas data frame from the dictionary.

1. Request and parse the SpaceX data using the Getrequest:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Normalize Json Response:

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

3. Extract usefull column using Auxiliary predefined functions:

getcoreData, getpayloadData, getlaunchsites, and getboosterversion

4. Create a Pandas data frame from the dictionary launch_dict:

```
launch_dict=pd.DataFrame.from_dict(launch_dict,
orient='index')
launch_dict=launch_dict.transpose()
```

Github Link:
https://github.com/promise32/myrep2/blob/8934c9cc6a54aab133530fc15c85c35721aa7f6c/spacex_data_collection_SpaceX_api.ipynb

Data collection-Webscraping From SpaceX wiki page

1. Obtain rocket launching data from the wikipedia page.

2. Request the Falcon9 Launch Wiki page from its URL and assign the resonse to a object.

3. Create a BeautifulSoup object from the HTML response

4. Find all tables on the SpaceX wikipedia page using the find_all function in the beutiful soup object to find all table columns and variable name.

5. Create a data frame by parsing the html table and save to a CSV format.

GitHub URL: https://github.com/promise32/myrep2/blob/5fc278a06b71a81f4faa95324e4b9ed13c1ed2f5/jupyter_labs_webscraping.ipynb

1. static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

2. Assign the response to a object:
static_url=requests.get(static_url).text

3. Create a BeautifulSoup object from the HTML response:
soup=BeautifulSoup(static_url, 'html5lib')

4. Use the find_all function in the BeautifulSoup object to find all tables:
Assign the result to a list called `html_tables`
html_tables=soup.find_all('table')

5. Extract column names one after the other
Create an empty dictionary with keys from the extracted columns:
launch_dict= dict.fromkeys(column_names)
Append the rows ‘tr’ and data in rows ‘td’ to the dictionary
Pass to dataframe and save to CSV
df.to_csv('spacex_web_scraped.csv', index=False)

SpaceX Data Wrangling

The Data wrangling stage of the SpaceX data was carried out as part of the data collection stage and these include:

- Using the isna().sum() python fuction to count the total number of missing values in each column.
- Used the .replace() python function and the numpy python library for mathimatical computation to Raplace all the missing variables with the mean of the column with the missing value.
- Save the dataframe to a CSV format for explorative data analysis.

Check for some of the rows with missing values in our dataset.

```
data.isna().sum()
```

↓

Dealing with Missing Values:

```
PayloadMass_mean=data['PayloadMass'].mean()
data["PayloadMass"].replace(np.nan, PayloadMass_mean, inplace=True)
```

↓

Export the Wrangled data to a CSV format

```
data.to_csv('dataset_part_1.csv')
```

Github Link:
https://github.com/promise32/myrep2/blob/8934c9cc6a54aab133530fc15c85c35721aa7f6c/spacex_data_collection_SpaceX_api.ipynb

SpaceX EDA with Data Visualization

- Exploratory data analysis (EDA) was performed using visualization tools such as Python's Matplotlib and Seaborn libraries
- Explorative data Analysis was done on some columns to check for their relationship and predictive variable dependences. The class column which is the outcome variable is either 0 for Failure or 1 for success and these visualization includes:
- A Scatter plot of Flight number versus payload mass with class as the lebel or legend.
- A Scatter plot of Flight number versus launch site with class as the label or legend
- A Scatter plot of Payload mass versus Launch Site with class as the label or legend
- A Bar plot of Orbit vs. Success Rate to check for thier Relationship
- A Scatter plot of Flight Number vs. Orbit type
- A Scatter plot of Payload vs. Orbit type
- A Line plot of Average Annual Success Trend

https://github.com/promise32/myrep2/blob/97b670bf5d660c6d9a1731ec23710ce7971c0ad5/jupyter_EDA_with_Data_Visualization.ipynb

SpaceX EDA with SQL

- Display the names of the unique launch sites in the space mission

```
%sql SELECT Distinct("Launch_Site")  FROM SPACEXTBL
```

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

```
%sql SELECT * from SPACEXTBL WHERE "Launch_Site" LIKE '%CCA%'  LIMIT 5
```

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

```
%sql select SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer="NASA (CRS)"
```

- 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'
```

- List the date when the first succesful landing outcome in ground pad was acheived

```
%sql select Landing_Outcome, Date from SPACEXTBL where Landing_Outcome= "Success (ground pad)"  LIMIT 5
```

https://github.com/promise32/myrep2/blob/a14ea47ddcae89301d4399ab16e7d9c69f09c4f3/final_project_jupyter_labs_eda_sql_coursera_sqlliteFINALANS.ipynb

Build An Interactive Map With Folium

- Folium map Marker and Circle objects would be used to create a visual location feature of all Launch sites on the map.
- The folium map Marker and Circle object was also utilized to show sites and their success/failure rate on the map.
- The folium map line object was used to super impose a line to show locations such as nearest coaslines and relaxation spots for example melbourne beach.
- Folium Map mouseposition object was created so as to get the correct coordinates of proximities such as railways in real time. this would assist in calculating different distances from differnt lacation of interest to the launch sites.
- I created the folium polyline object on the map to draw a line from one of the launch sites to three location namely the nearest coustline, melbourne beach and Samuel C philip parkway.

https://github.com/promise32/myrep2/blob/e2b51d1dcc0de63e886160c724b4c6f0ac9153ea/lab_jupyter_launch_site_locationPractice.ipynb

Build A Dashboard With Plotly Dash

- The Dash interactive application was use to create a dashboard to show pie charts and scatterplots with plotly.
- The Dashboard had five dropdown for the pie charts and a Range slider for the scatter plots.
- The five Dash dropdown show a Success/failure attempt pie plot of all the launch sites and and that of the individual four launch sites.
- The Dash Range slider was used to show how the different payload mass can affect affect the success rate for the booster version categories.

https://github.com/promise32/myrep2/blob/88c65fc9957e84478a7868505790650a3ceab969/Interactive_Dashboard_with_PlotlyspaceX.ipynb

SpaceX Predictive Analysis (Classification)

- The first stage in the machine learning and prediction phase is preparing the data for machine learning algorithm.
- The the class column which is the predictive variable(Binary variable) was changed to a numpy array using the numpy library object .to_numpy().
- The Sklearn preprocessing.StandardScaler() object was used to standardize the independent variables to account for outliers or data that would fall far from each other.
- The train_test_split function of the sklearn library was use to split the into training and testing set to account account for in sample error.
- The GridSearch model was implemented with other models such as LogisticRegression, DescisionTreeClassifier, SupportVectorMachine and KNearestNeibours to search for the best performing model.
- The model accuracy(for Test and Train set) was calculated for each model to evualuate the model to check for the best performing model.
- The confusion metrix was plotted to check for the percentage of false Negative(FP) and True positive(TP) in the prediction of the models in other to accertain the ones with an higher percentage.
- *https://github.com/promise32/myrep2/blob/ab2f328c8f3958078 9b4738 703e17f5cba2026ac /SpaceX_Machine_Learning_Prediction_jupyterlitePROJECT.ipynb*

Create a NumPy array from the column Class and assign it to the Y variable:

```
Y=data['Class'].to_numpy()
```

↓

Standardize the data in X then reassign it to the variable X:

```
transform = preprocessing.StandardScaler()
X = preprocessing.StandardScaler().fit(X).transform(X)
```

↓

Use the function train_test_split to split the data X and Y into training and test data:

```
X_train, X_test, Y_train, Y_test= train_test_split(X, Y, test_size=0.2,random_state=2)
```

↓

Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10:

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']
lr=LogisticRegression()
GSCV=GridSearchCV(lr, parameters, scoring='accuracy', cv=10)
logreg_cv=GSCV.fit(X_train, Y_train)
```

↓

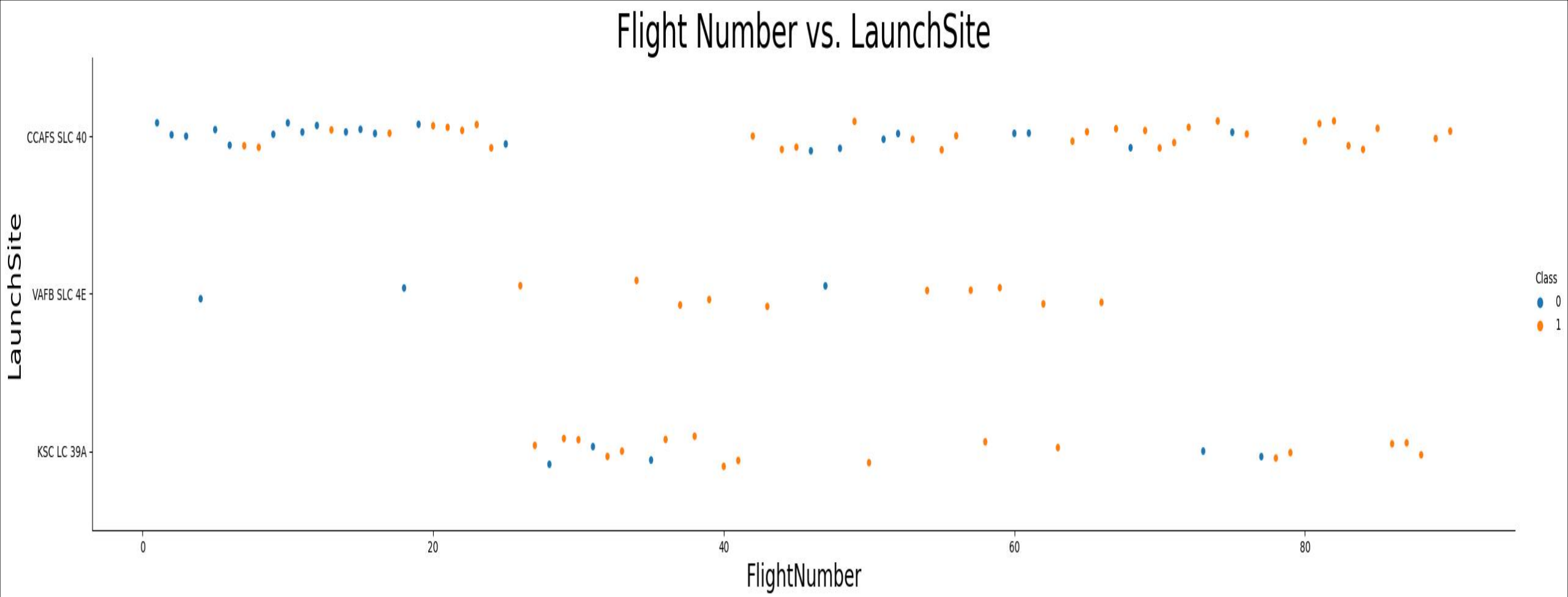
Display the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_:

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

SpaceX Success Results

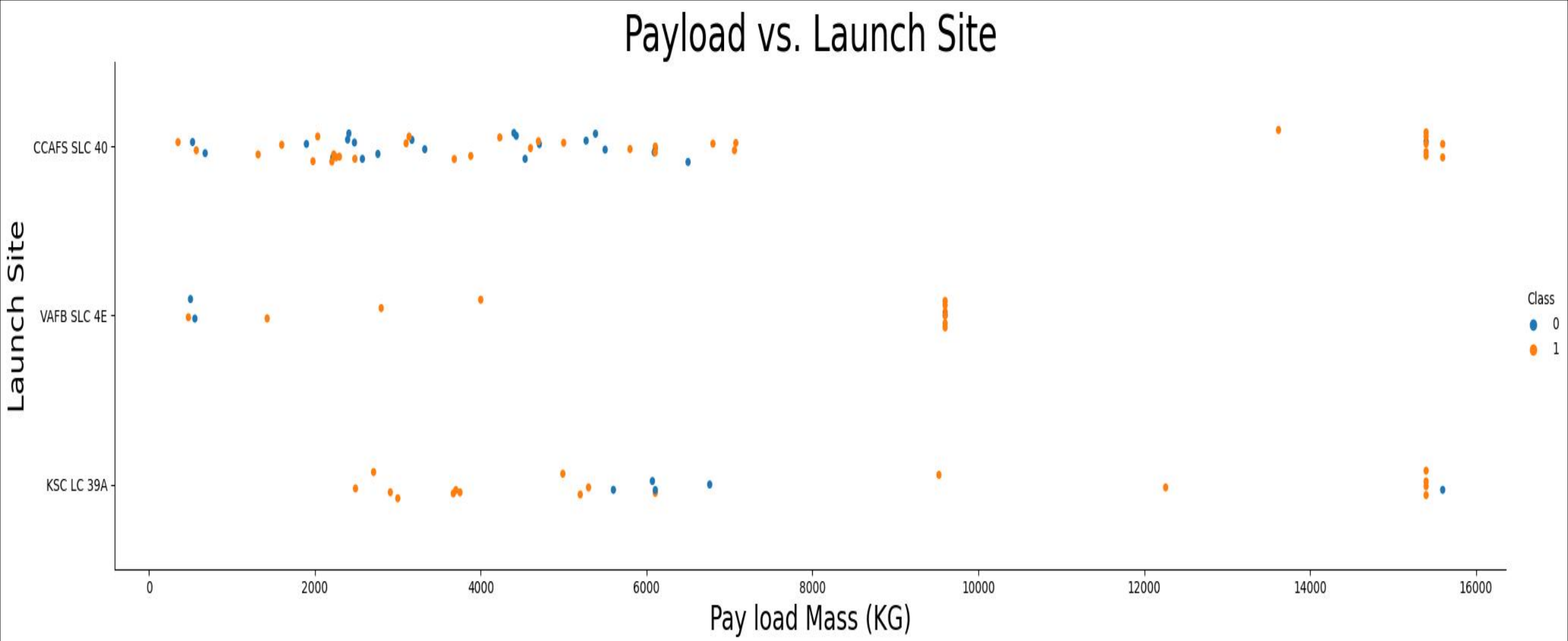
The following slides will show the results of:

- Explorative data analysis(EDA) with visualization
- Explorative data analysis(EDA) with SQL
- Interactive Map with Folium
- Interactive Dashboard with Dash
- Finally, the results of the models with approximately 83% accuracy.
- For visualization plots Blue = Failure
  Orange= Success



The Flight Number vs LaunchSIte scatter plot indicate that most of the launches where carried out at the CCAFS launch site, thus it has a success rate of about 65% where as sites like VAFB SLC 4E and KSC LC 39A has about 70% and 85% success rate respectively but very few launches for these sites compared to the CCAFS launch site.

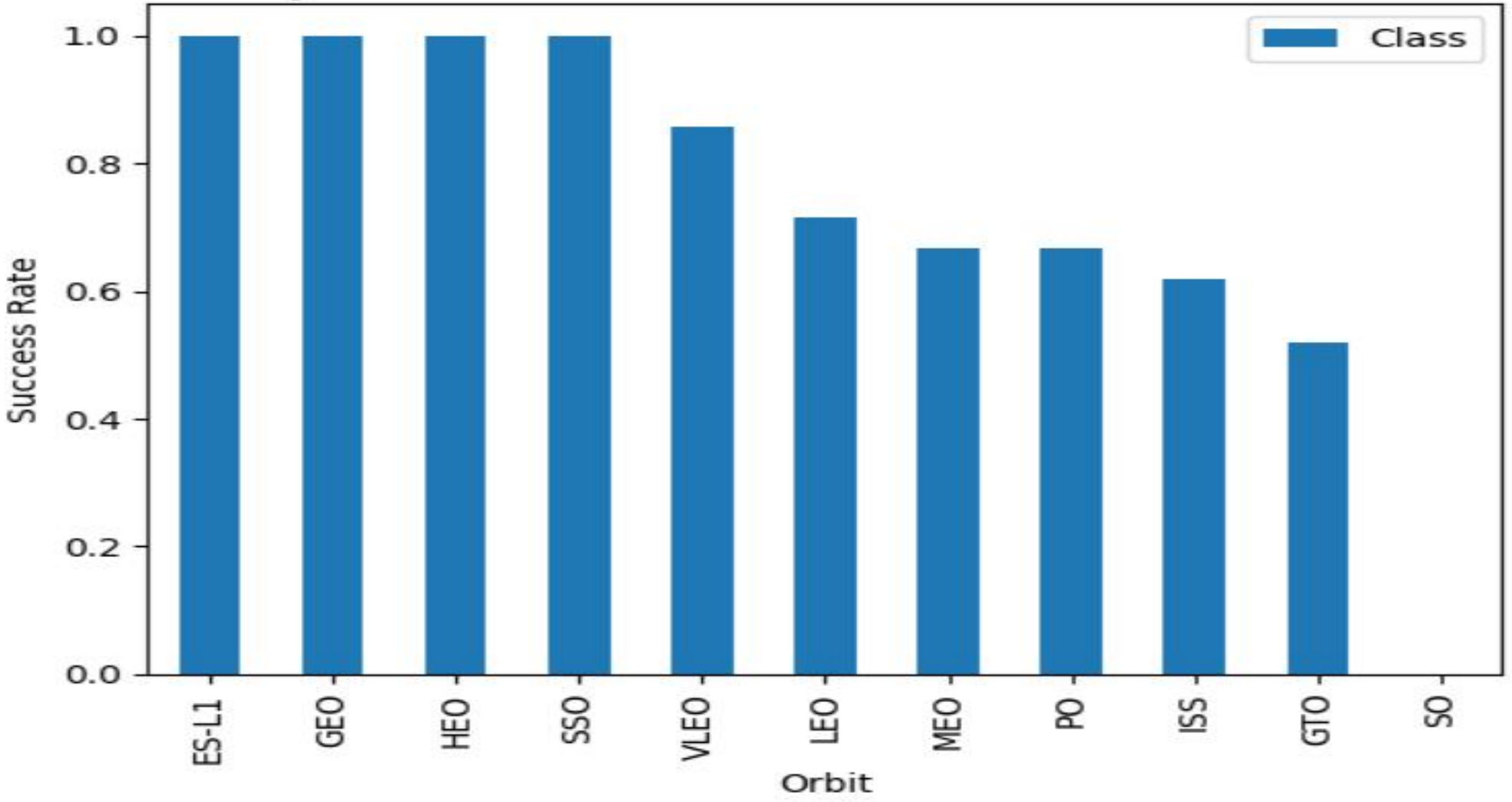
We can also notice a significant increase of the success rate as the flight number increases or as the attempt become much for all the launch sites.



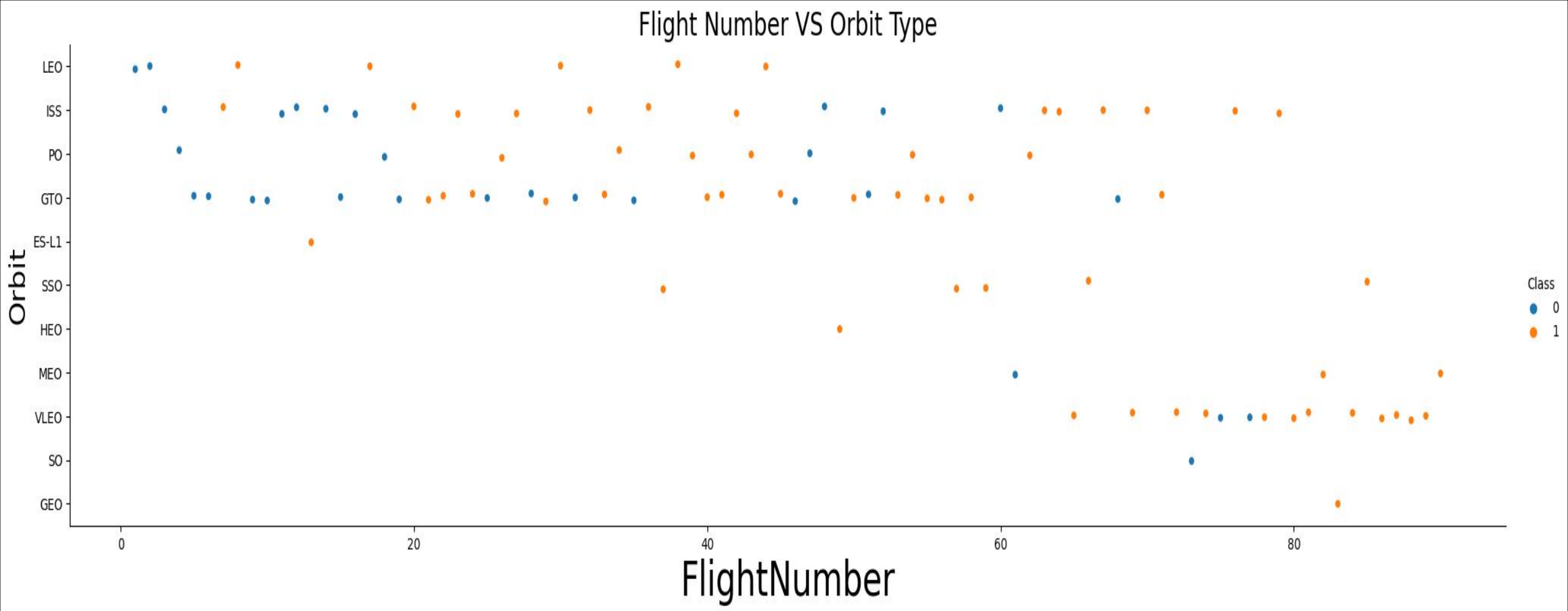
We see that there are absence of launches from about 9800kg payload for WAFB SLC 4E launch site and missing launches from about 12200kg to about 12900kg payload mass for the the rest of the sites.

what this surggest is that launches become fewer as the payload increases and this implies a weak correlation or incossitency in data which makes it difficult to draw a reasonable descision using only this metric.

Relationship between success Rate and Orbit Type

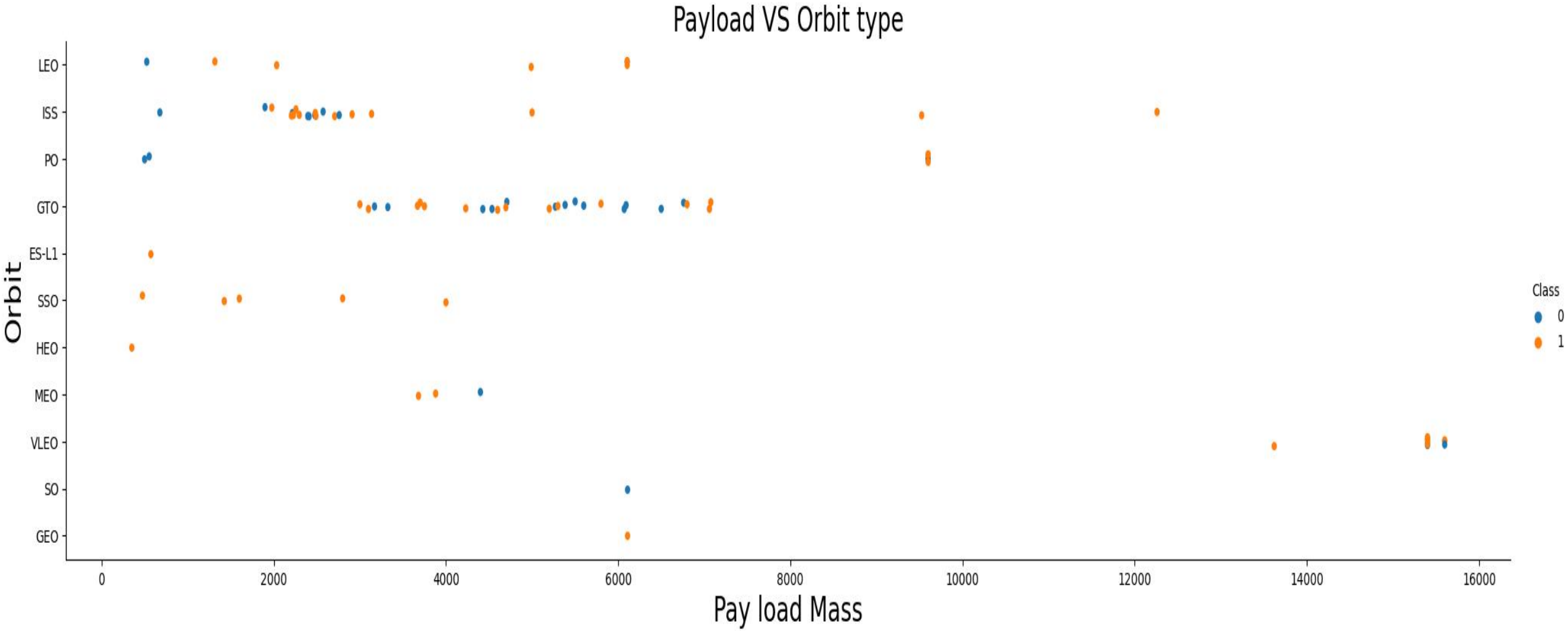


- The SSO, HEO, GEO and ES-L1 orbits have 100% success rates.
- SO orbit has no successful launches with a 0% success rate.
- It appears that the lower the orbit the more chances of successful launches it has.



We can tell alot of stories from the scatterplot for example:

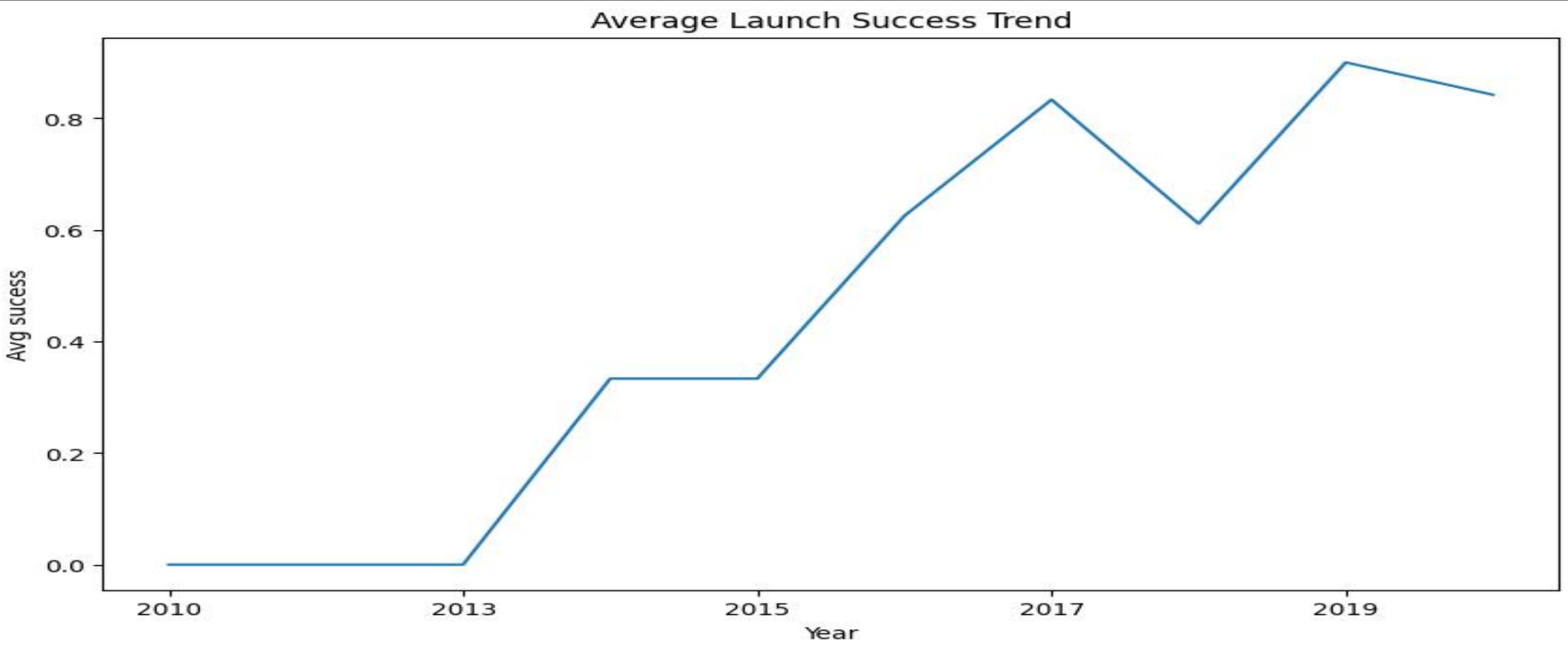
- SSO, HEO, GEO and ES-L1 orbits have 100% success rates.
- The SO orbit has no successful launches with a 0% success rate.
- VLEO orbit seems not to have significant launches from the first attempt to about 67th launch attempt where it record about 85% success.
- LEO orbit appears success related to the number of flights; on the other hand, there appears to be no relationship between the number of flight in the GTO orbit



We can see from the plot that With heavy payloads, the successful or positive landing rate is higher for Po, LEO and ISS However, for GTO we cannot distinguish this well, as both positive landing rate and negative landing rate are present here.

For VLEO a successful landing is also observed with heavy payload less than 16000.

It is observed that at lower payload some orbit types have higher probability of successful landing.



From the plot You can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2017 the success rate decreased a bit until 2018 it started increasing all the way up untill 2019.

We can conclude that there has been quite a fluctuation in success rate over this period.

All Launch Site Names

[ ]

Tasks

Now write and execute SQL queries to solve the assignment tasks.

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

Display the names of the unique launch sites in the space mission

```
[ ] %sql SELECT Distinct("Launch_Site") FROM SPACEXTBL
```

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

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

I used the sql query language select distinct(launchsite) from spacextbl. this table is already stored in a sqllite database i created so i used the sql code to query the data base so as to retrieve all launch sites from the table.

Launch Site Names That Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

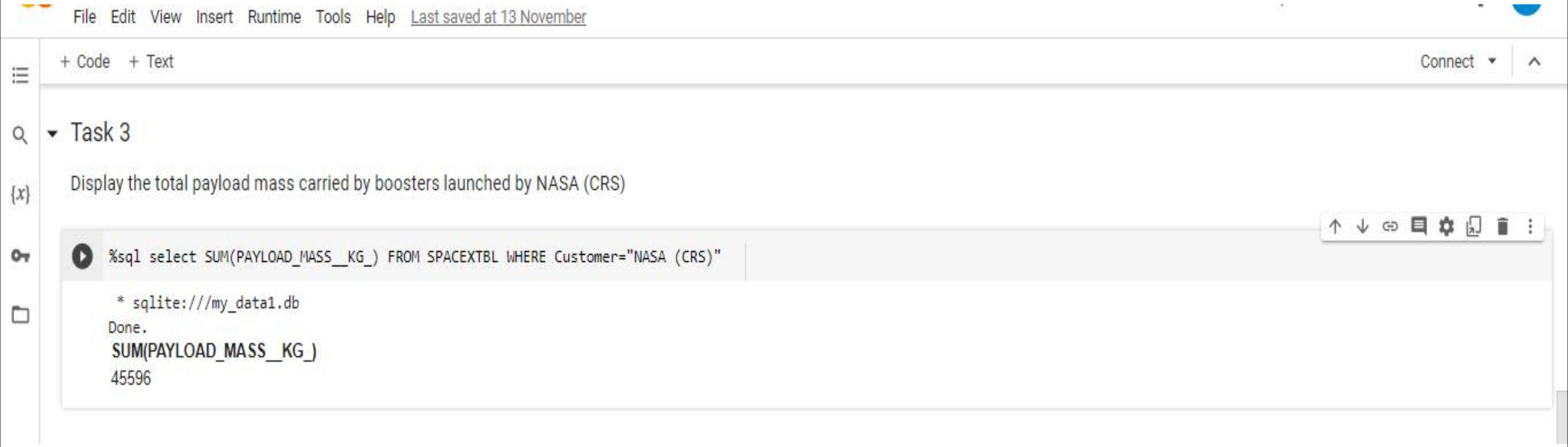
```
[ ] %sql SELECT * from SPACEXTBL WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
12/8/2010	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)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

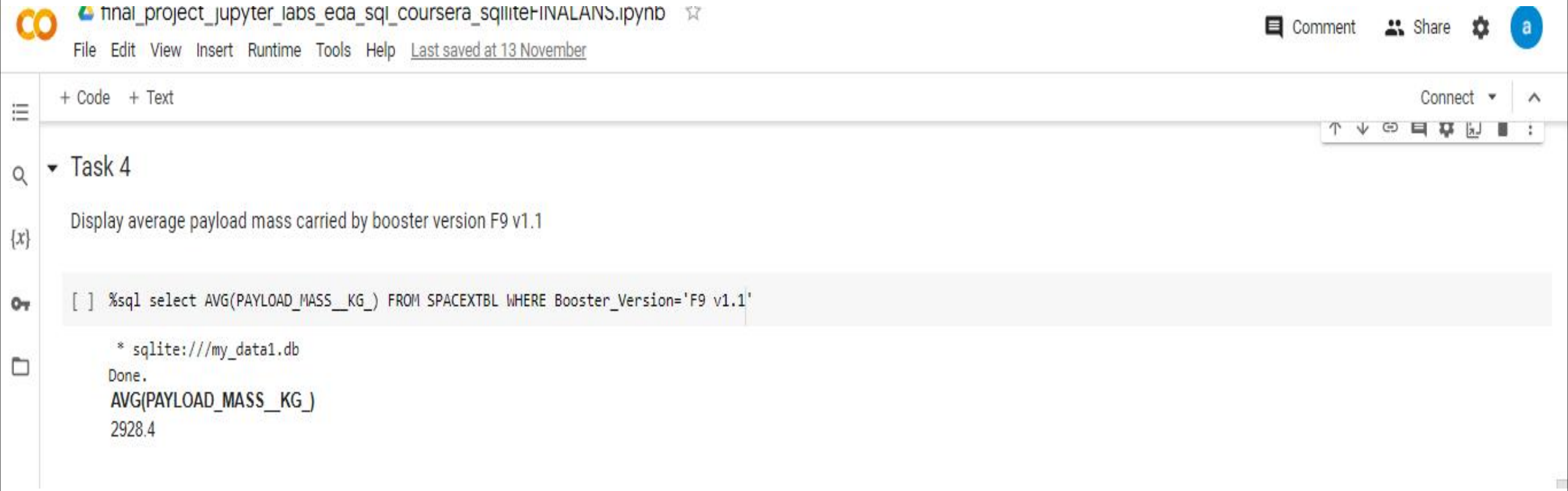
I performed a sql query to get the First five entries in database with Launchsite name beginning with CCA.

Total Payload Mass



This query sums the total payload mass in kg where NASA is the customer. NASA CRS (Commercial Resupply Services) is a series of supply missions to the International Space Station (ISS) carried out by SpaceX under contract with NASA. CRS mission payloads include a variety of scientic experiments, crew supplies, and spacelab hardware.

Average Payload Mass by F9 v1.1

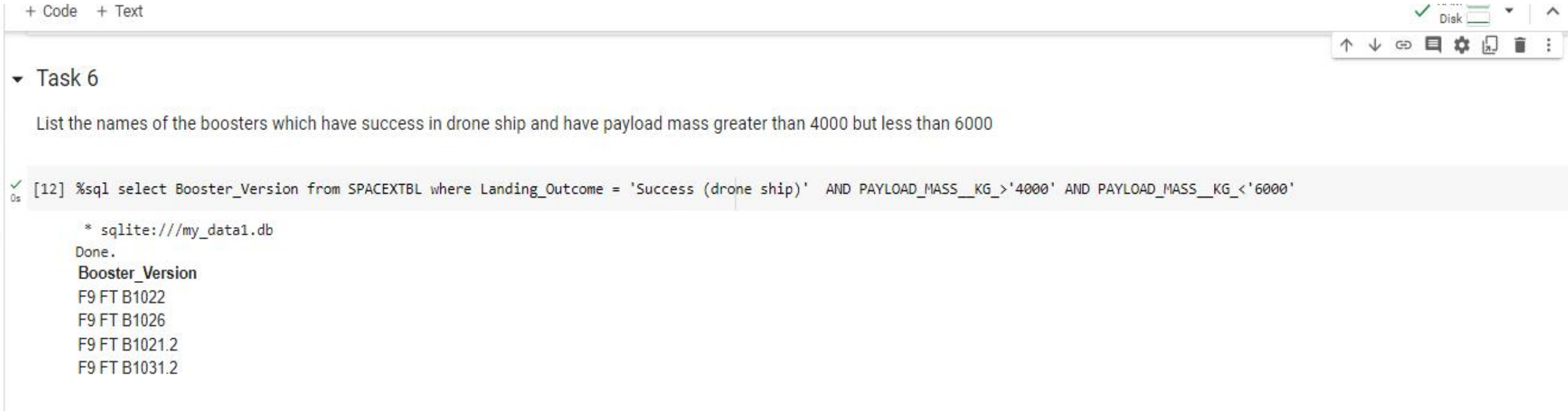


- This query calculates the average payload mass or launches that used the F9v1.1 booster version.
- The average payload mass of the F9 1.1 is at the lower end of the managed payload mass range.

First Successful Ground Landing Date

The first successfull landing on land was towards the end of 2015. Successful landings generally began to appear From 2014.

Successful Drone Ship Landing with Payload between 4000 and 6000



This query returns the four booster versions that had successful landing on drone ships and with a payload mass between 4000 and 6000, not inclusive.

Total Number of Successful and Failure Mission Outcomes

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Task 7

List the total number of successful and failure mission outcomes

```
[13] %sql SELECT mission_outcome, COUNT(*) AS total_count FROM SPACEXTBL GROUP BY mission_outcome
```

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

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

- This query returns a count of each of the mission results.
- The result of SpaceX missions is almost 99% successful.
- One successful launch is shown having an unclear payload status and another failed during night.

Boosters Carrying Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = ( select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

This query returns the booster versions, which carried the largest payload mass of 15,600 kilos.These booster versions are a variety of the F9 B5 B10xx.x boosters.

2015 Failed Drone Ship Landing Records

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql select strftime('%Y', Date) as year, strftime('%m', Date) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL
where Landing_Outcome= 'Failure (drone ship)' and year='2015'
```

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

year	month	Landing_Outcome	Booster_Version	Launch_Site
2015	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

This query returns the month, year, landing result, booster version, payload mass (kg), and launch location for 2015 launches in which stage 1 failed to land on a drone ship. .

There were two such launch events with landing on a drone ship in 2015 and the outcome was a failure.

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

Task 10

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

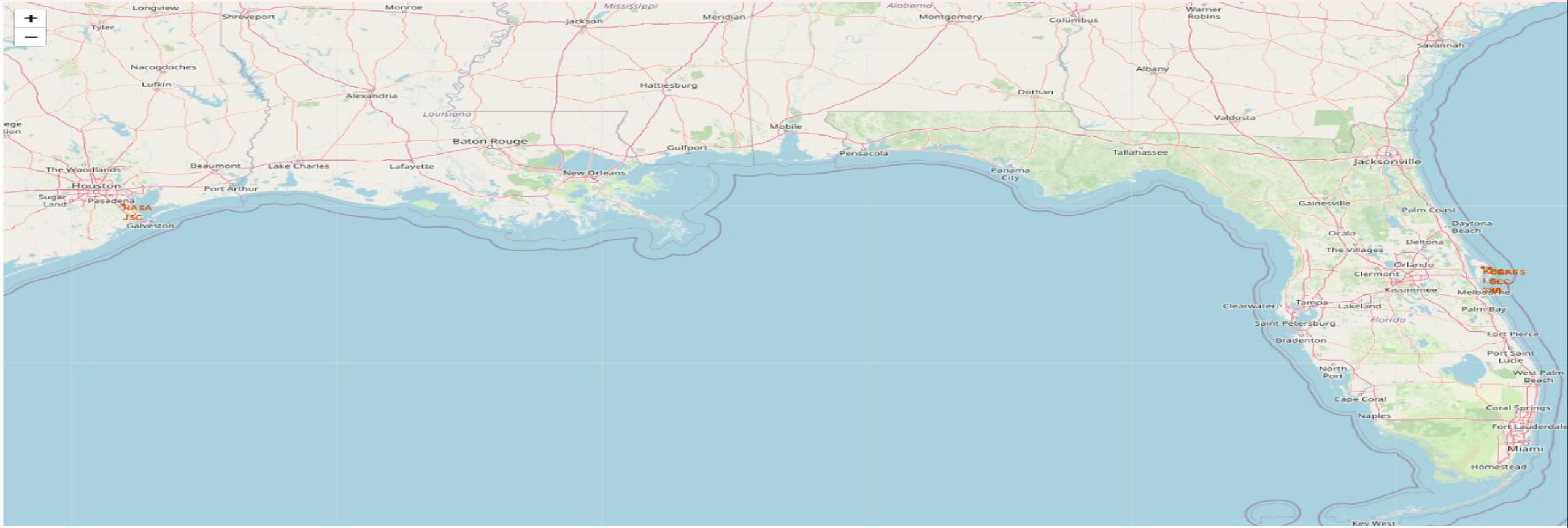
```
%%sql
SELECT Landing_Outcome,COUNT("Landing_Outcome") from SPACEXTBL where Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
and  Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by COUNT("Landing _Outcome") desc
```

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

Landing_Outcome	COUNT("Landing_Outcome")
Failure (drone ship)	5
Success (ground pad)	3

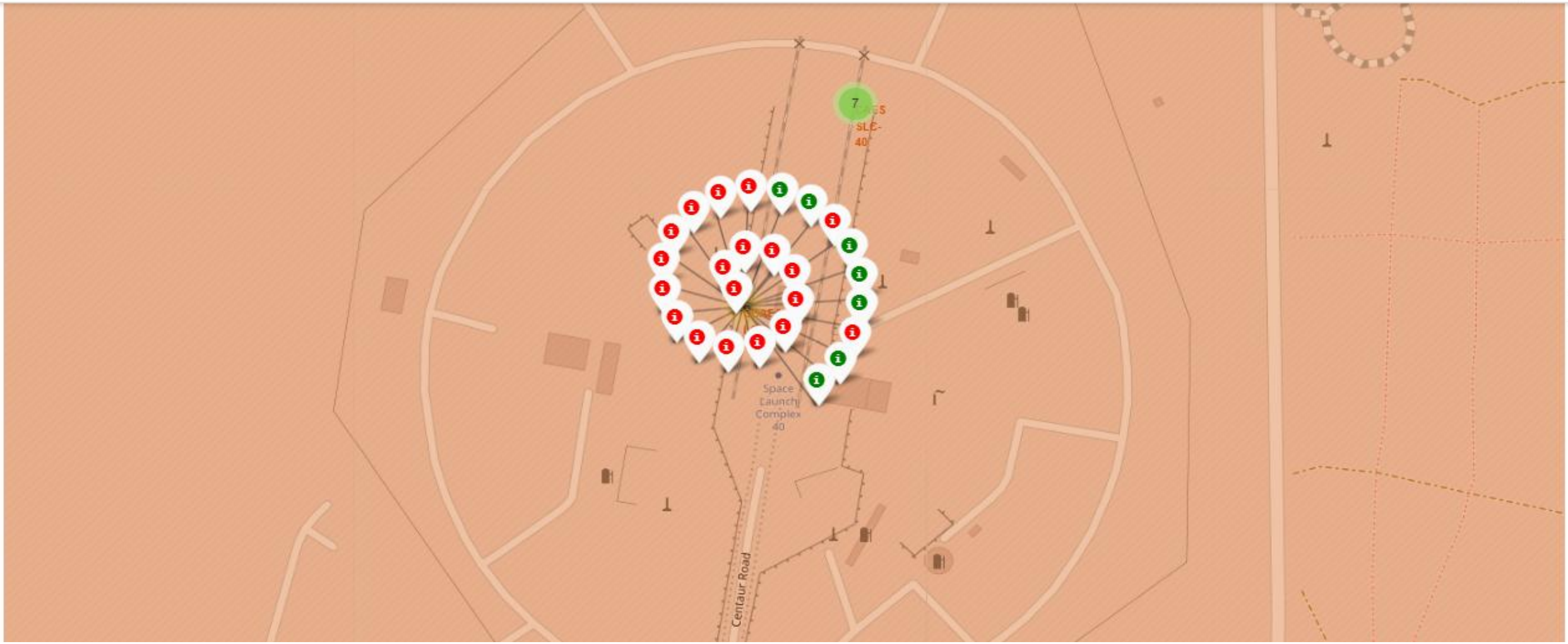
This sql query returns the total number of failure(drone ship) and that of the Success(ground pad).

Launch Locations on a folium Map



- The Red Cycles markers are indicators of where all SpaceX Launch and from the coordinate used to plot the map, it shows the sites are located in the US.
- The launch sites have been strategically placed near the coast. The proximity to the coast facilitates rocket launch operations, which require direct access to the sea for easy monitoring of the trajectory of rockets after launch and secondly the proximity to the equator means that experiences warm weather conditions throughout the year, ideal for rocket launch operations, which often require stable climatic conditions.

Launch Markers Differentiated by Color

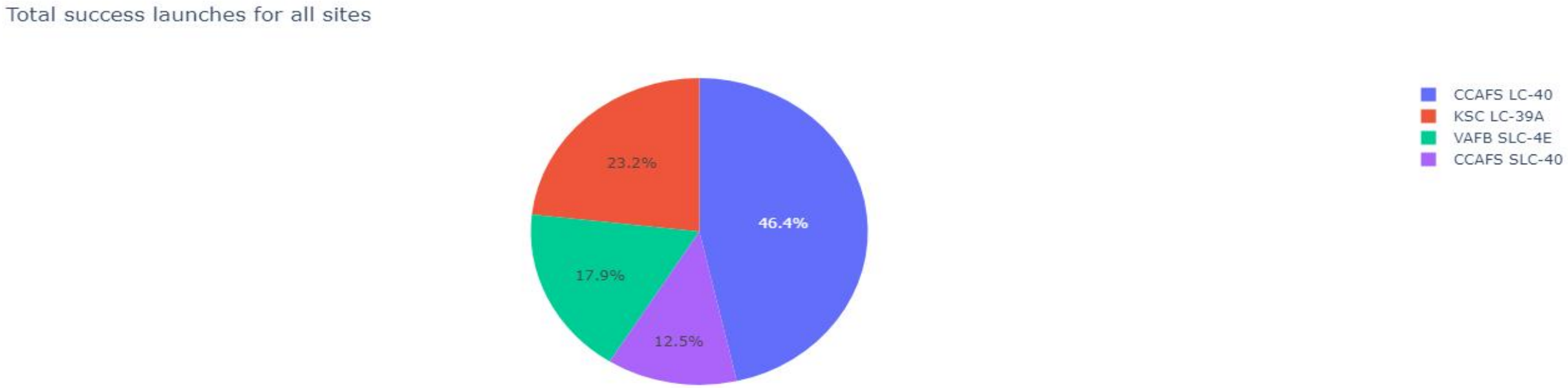


Clicking on groups on the Folium map shows each successful landing (green icon) and failed landing (red icon). In this example, CAFS SLC-40 shows 7 successful landings and 19 failed landings.

Proximities of Launch Sites

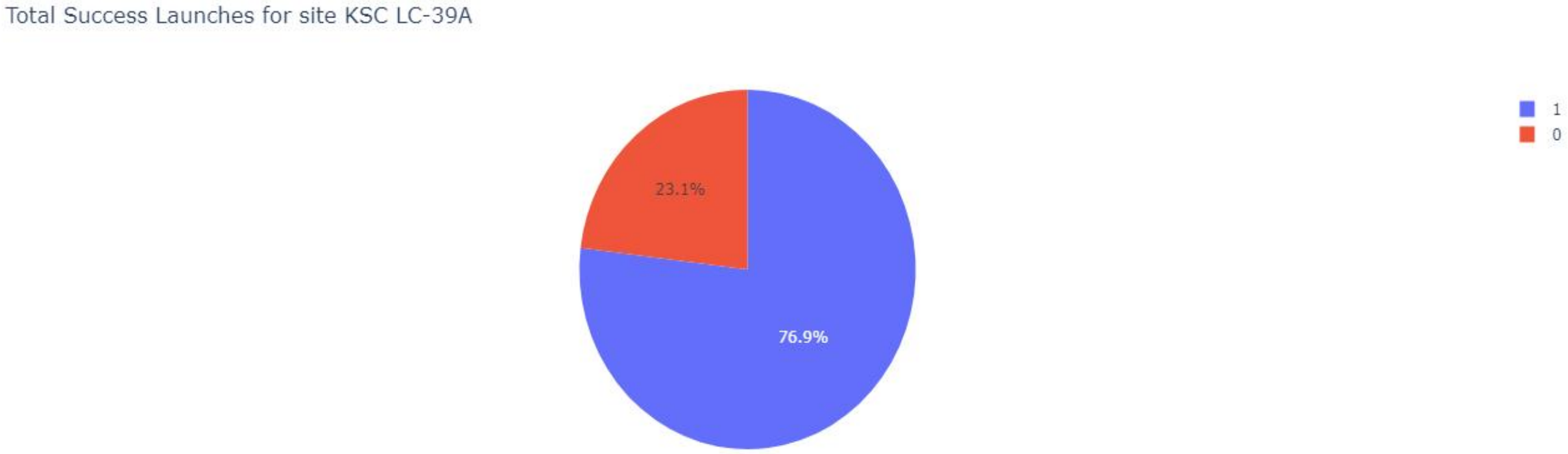
Referring to the CCAFS launch site SLC 40, it is close to the coast for launch failure testing. Launch sites are also kept some distance from cities. Furthermore, the location is also beneficial from a security point of view. Because it is far from major population areas, Rocket launches are less likely to interfere with human activities. However, they are close to some highways and trains for handling personnel and equipment.

Total Successful Launches By Site



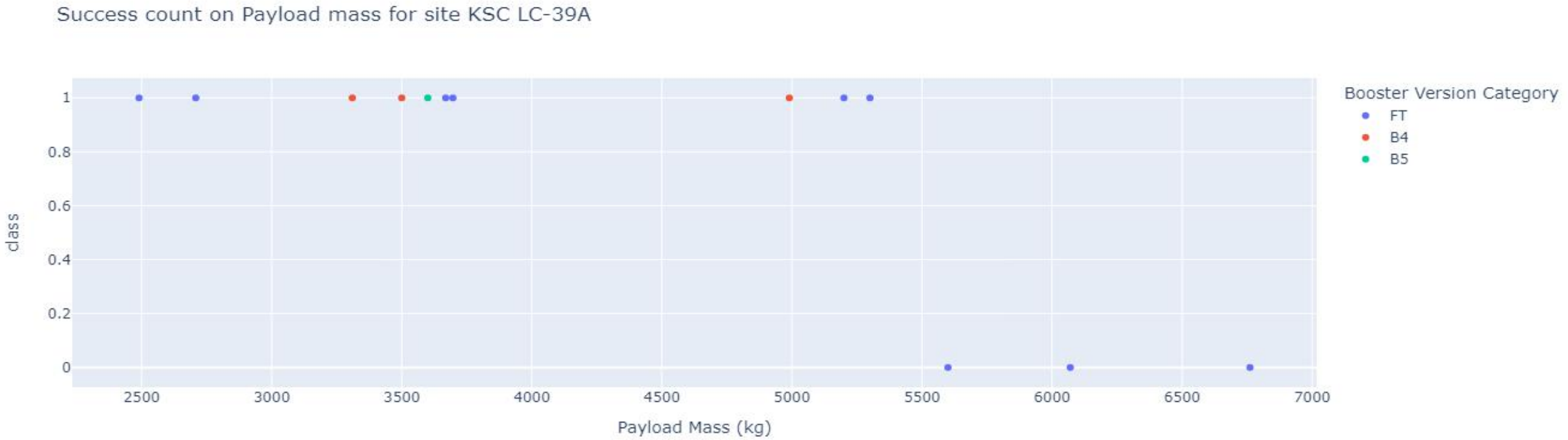
These are the percentages of successful landings at all launch sites. The one with the greatest apparent range is that of the KSC LC-39A. On the other hand CCAFS LC-40 have a similar name to CCAFS SLC-40, and refer to the same launch facility in Cape Canaveral, Florida, but SpaceX has used SLC-40 to launch the Falcon 9, so CCAFS and KSC they have the same number of successful landings, but most of the successful landings were made before the change to SCL. The VAFB has the lowest proportion of successful landings. This may be due to a smaller sample size.

Total Success Launches For Site KSC LC-39A



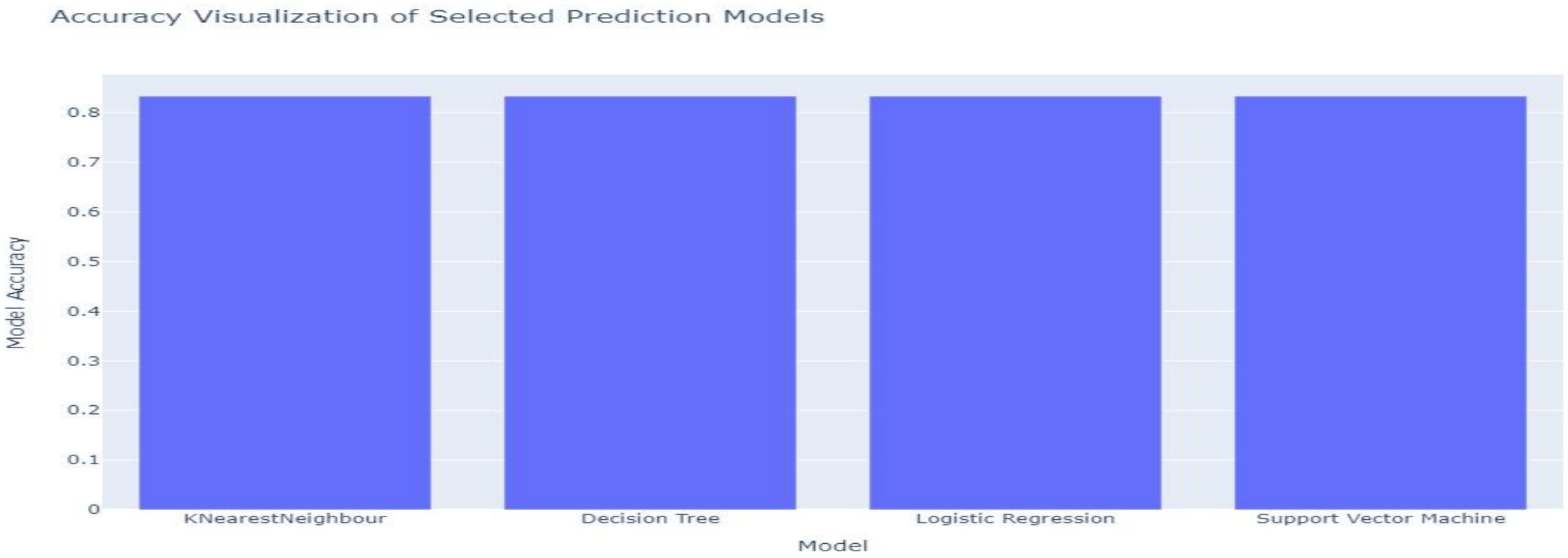
KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload vs. Launch Outcome For Site KSC LC-39A Scatter Plot



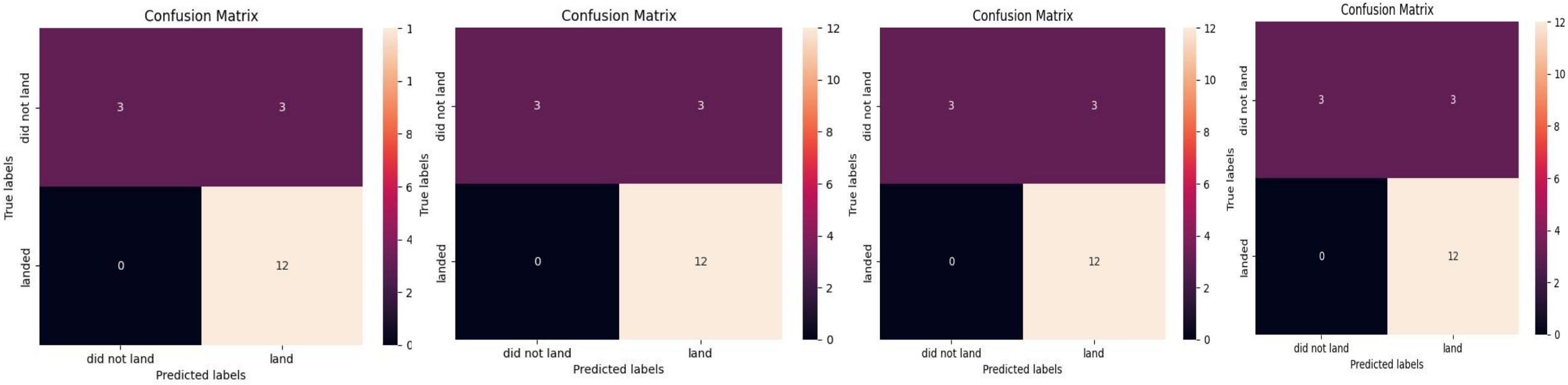
The payload range selector is set between 0 and 10000 instead of the maximum payload of 15600. The class indicates 1 for successful landing and 0 for failure.The highest success launch rate for all boosters where with payloads between 2500 and 5400kg but FT booster recorded failure from payload mass 5600kg to about 6300kg. The scatter plot also tries to explain how the size of boosters can affect the success rate of a Rocket launch..

Classification Accuracy



The logistic regression models, the support vector machine, the decision tree and the k Nearest Neighbors have the same Accuracy score, so they all perform with an accuracy equal to 83.3% and they are all good at predicting correctly positive cases, but it is worth noting that the trial data size is small, with a sample size of only 18. More data needs to be used to determine the best model.

Model Evaluation with Confusion Matrix



The models predicted 12 successful landings. When True tag succeeded (TruePositive) and 3 failed landings when True tag failed (True Negative). Since all models performed equally on the test set, the confusion matrix is the same across all models.

They also predicted 3 successful landings when the True label was on failed landing (False Positive); however, the models generally predicted successful outcomes and any of them are good at correctly predicting positive cases. Although the ideal result expected from the confusion metrix is for the model to give 0 false positives and 0 false negatives, that will not always be the case in real life as any model will not be 100% accurate in most cases. in the next slide we would use other metrics for the evaluation.

Model Evaluation with F1_score, Jaccard_Index and Log_Loss

Report2

	Model	JaccardIndex	F1_Score	Log_Loss
0	KNearestNeighour	0.7	0.814815	0.478667
1	Decision Tree	0.7	0.814815	NaN
2	Logistic Regression	0.7	0.814815	NaN
3	Support Vector Machine	0.7	0.814815	NaN

```
[49] %matplotlib inline

     import matplotlib as mpl
     import matplotlib.pyplot as plt

     mpl.style.use('ggplot') # optional: for ggplot like style
```

Further Evaluation of all the models show that the four model would perform very well. we notice thesame value for all the evaluation metrix namely Jaccardindex=0.7 accross all model, F1_score= 0.814815 and Log_Loss=0.478667.

Conclusions

- The analysis showed that there is a positive correlation between the number of flights and the success rate, the greater the number of flights the greater the success and this is should be reason why the average success rate is 66.6% and has improved with the passing of the years.
- There are certain orbits such as SSO, HEO, GEO and ES-L1 where the launches were more successful, because they are geosynchronous orbits i.e they can remain in a fixed position and are ideal for communications satellite operations, they can be closer to the equator.
- The success rate can be linked to the mass of the payload, as lighter payloads were generally more
- successful than heavier ones.
- The launch sites are strategically located near some roads and railways for transporting personnel and
- cargo, away from cities for security and safety reasons.
- SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful
- Stage 1 landing before launch to determine whether this launch should go ahead or not.
- If possible, more data should be collected to determine the best machine learning model and improve
- accuracy, because the data was insulficient, and it could happen that there was little variation in the
- classes, overtting, or homogeneous data.

Final Consideration

According to the results predicted by the classi cation models, there is an 83% probability that SpaceX will be able to reuse the rst stage of SpaceX and if the launches manage to land successfully, it can be predicted that each launch would cost 62 million dollars, even if it still costs a little more, it is a good business idea to launch rockets for various operations, especially for communications satellites, because according to EDA, the highest success rate occurs in orbits closest to the equator, which are geosynchronous and can remain in a xed site from the position of the earth.

Taking into account that the payload in each launch must be between 2000 and 4000 Kg. to ensure success, and locating the launch site at KSC LC 39A or the CCAF SLC 40, which although they have had some failed launches as As flights increase, conditions will improve, proving to have an average success rate of 66%.

In general terms, it is quite feasible to consider launching rockets in the coming years, following these guidelines to offer them at a very good price.

Appendix

Projects link:

https://www.coursera.org/professional-certicates/ibm-data-science

https://scikitlearn.org/stable/modules/generated/sklearn.tree.ExtraTreeClassifier.html

https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnNOcnVjdGlvbnNfdXJsIjoiaHROcHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWNOLXNOb3JhZ2UuYXBwZG9tYWluLmNsb3VkLOlCTURldmVsb3BlclNraWxsc05ldHdvcmstRFYwMTAxRU4tU2tpbGxzTmVOd29yay9sYWJzLO1vZHVsZSUyMDQvNC41XORhc2hfQmFzaWNzLm1kliwidG9vbF9OeXBlljoidGhlaWEiLCJhZG1pbiI6ZmFsc2UsImlhdCI6MTY4ODY1ODYzM30.giRGZ-OE7RdO7UeceVryDKjjJuRcSn3HxbqQG-lP1do

https://github.com/promise32/myrep2/blob/a14ea47ddcae89301d4399ab16e7d9c69f09c4f3/Interactive_Dashboard_with_PlotlyspaceX.ipynb

https://github.com/promise32/myrep2/blob/6732818416e053ef279aae0d82f3b99d2883250c/SpaceX_Machine_Learning_Prediction_jupyterlitePROJECT.ipynb

Appendix

https://github.com/promise32/jetty/blob/3dceabc167f15eb688e88cbc3fc3d750ae0c0405/lab_jupyter_launch_site_locationPractice.ipynb
https://github.com/promise32/myrep2/blob/6732818416e053ef279aae0d82f3b99d2883250c/jupyter_EDA_with_Data_Visualization.ipynb
https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstRFYwMTAxRU4tU2tpbGxzTmV0d29yay9sYWJzL3Y0L0RWMDEwMUVOLUNoZWF0U2hlZXQtMy5tZCIsInRvb2xfdHlwZSI6Imluc3RydWN0aW9uYWwtbGFiIiwiYWRtaW4iOmZhbHNlLCJpYXQiOjE2ODDcyNjY5MjJ9.7zZqH-bsK4UBWIRFy5SJQ-VpffE5SocG3379X8_oLGU
https://github.com/promise32/myrep2/blob/6732818416e053ef279aae0d82f3b99d2883250c/final_project_jupyter_labs_eda_sql_coursera_sqlliteFINALANS.ipynb
https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstREIwMjAxRU4tU2tpbGxzTmV0d29yay9sYWJzL015U1FML3dlZWszL211bHRpcGxldGFibGVzX3F1ZXJpZXMubWQiLCJ0b29sX3R5cGUiOiJ0aGVpYSIsImFkbWluIjpmYWxzZSwiaWF0IjoxNjgzODU2NjkzfQ.DljkrwF87ZNXRWs_252gD4TzDwgZvXQLWfMoqJ-8XGMD

Special thanks the very good instructors
https://www.coursera.org/professional-certificates/ibm-data-science