Space X : Falcon 9 Success Analysis

- Sayan Chakraborty

Outline

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

Executive Summary

❑ Summary of methodology

- Unstructured Data Collection
- Extracting Structured data from unstructured data
- Data Wrangling
- Exploratory data analysis with visualization
- Exploratory data analysis with SQL
- Building Interactive map of launch sites
- Predictive analysis

❑ Summary of results

- Exploratory data analysis result
- Predictive analysis result

Introduction

❑ Project Context:

Based on different information of Falcon 9 launch We tried to predict if the Falcon 9 first stage will land successfully. As per the advertisement in SpaceX website it takes 62million USD to launch Falcon 9 While other launching service providers will charge 165million. SpaceX is saving huge amount because of the reusable property of two-stage Falcon 9 rocket. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. If another firm wishes to compete with SpaceX for a rocket launch, this information can be used.

❑ Problem Statements to be solved:

- What are the features that influence if the rocket will land successfully?
- What are the effect of each features on the landing of Falcon 9?
- What conditions and parameters does SpaceX need to consider to improve its landing success rate?

METHODOLOGY

- DATA COLLECTION VIA SPACEX API
- DATA COLLECTION VIA WEB SCRAPING
- DATA WRANGLING
- EDA WITH DATA VISUALIZATION
- EDA WITH SQL
- PREDICTIVE ANALYSIS

Data collection via SpaceX API

Step 1: Get response from the SpaceX API

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

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

Step 2: Convert response to json data structure

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

Step 3: Extract Structured data from json data structure using Custom function

```
]: getBoosterVersion(data)
   getLaunchSite(data)
   getPayloadData(data)
   getCoreData(data)
```

Step 4: Assign extracted and cleaned data to the dictionary with feature names

```
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}
```

Step 5: Filtered the data for only Falcon 9 and saved as a CSV file

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

Data collection via Web Scraping

Step 1: Get response from the wikipidea page and convert into beautiful soup object

```
static_url = "https://en.wikipedia.org/w
response = requests.get(static_url)
response = response.text
soup = BeautifulSoup(response)
```

Step 2: finding all tables and extracting the columns names of the table

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

```
first_launch_table = html_tables[2]
```

```
column_names = []
for i in first_launch_table.find_all('th'):
    col_name = extract_column_from_header(i)
    if col_name!=None and len(col_name)>0 :
        column_names.append(col_name)
```

Step 3: Creation of dictionary

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Step 4: Appending values from table data of the web page into dictionary after cleaning with custom functions

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table'
    # get table row
     for rows in table.find_all("tr"):
         #check to see if first table heading is as number
```

Step 5: Converting the dictionary to dataframe and saving as CSV file

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- ***About the data set that we created from data collection process*:**

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

In data wrangling process –

- For continuous data we imputed the null values with the mean value for the feature.
- Based on the outcome data column we created the label for successful landing or unsuccessful landing. And the label is assigned to ‘Class’ column. This ‘Class’ column is our target class for predictive modeling.
- Finaly the new dataset is cleaned from missing value and a target class is added. The new data set is saved as csv to perform EDA and predictive analytics.

EDA with data Visualization

Categorical scatter plots–

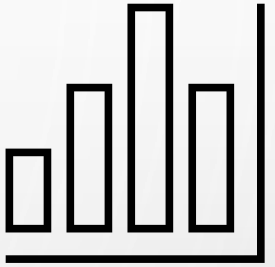
- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit Type
- Orbit VS. Payload Mass

Scatter plots depict how one variable influences another. Correlation is the term used to describe the relationship between two variables. Scatter plots are typically made up of a vast amount of data.

Bar plots–

- Mean VS. Orbit

A bar diagram allows you to quickly compare sets of data from different categories. On one axis, the graph shows categories, and on the other, a discrete value. The purpose is to demonstrate the connection between the two axis. Bar charts can also be used to depict significant changes in data over time.



Line plots–

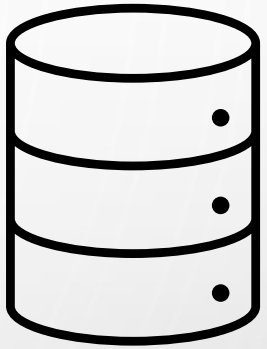
- Success Rate VS. Year

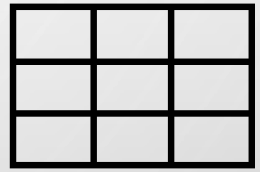
Line graphs are valuable because they clearly display data variables and patterns, and they can aid in making predictions about the outcomes of data that has not yet been recorded.

EDA with SQL

Performed SQL queries to gather information about the dataset.–

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015¶
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order





Interactive map of launch sites

- To make an interactive map out of the Launch Data. We used the Latitude and Longitude Coordinates for each launch site to create a Circle Marker with the name of the launch site labelled around it.
- With Green and Red markers on the map in a MarkerCluster(), we allocated the dataframe launch outcomes(failures, successes) to Classes 0 and 1.
- To identify distinct trends about what is happening in the area around the Launch Site to measure patterns on the map, lines are drawn to indicate the distance between landmarks.

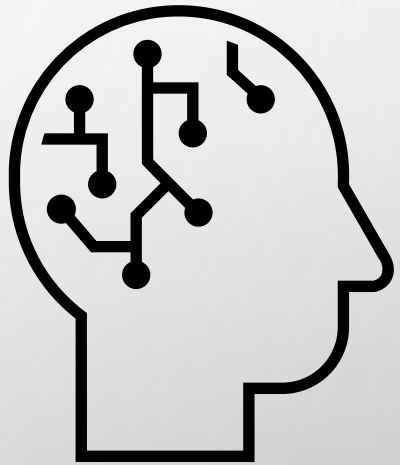
By plotting the distance line of proximities, we can answer following questions

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities.

Predictive Analysis

Model Building

- Transformed the data with standard normal transformation texgnique to scale the data.
- Data is split between training and testing set to measure the performance of the model
- Tested several classification model to check which model is peroformimng the best
- Fine tuned the models with grid search technique and found the best parameters for each model.



Evaluating the models:

- Checked accuracy and confusion matrix of each model on test data and train data.
- Used cross validation technique to perform out of sample performance.

Model outcome:

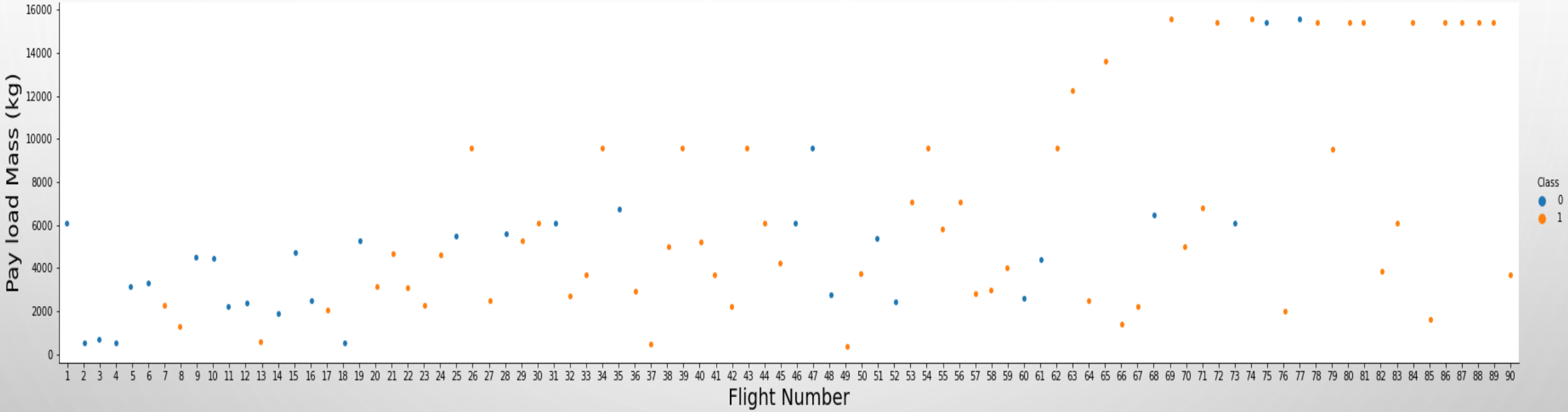
- Predicted test data with the best performing model
- Found the most influencing features from the data set

RESULTS

- EXPLORATORY DATA ANALYSIS RESULTS
- PREDICTIVE ANALYSIS RESULTS

EDA with Visualization

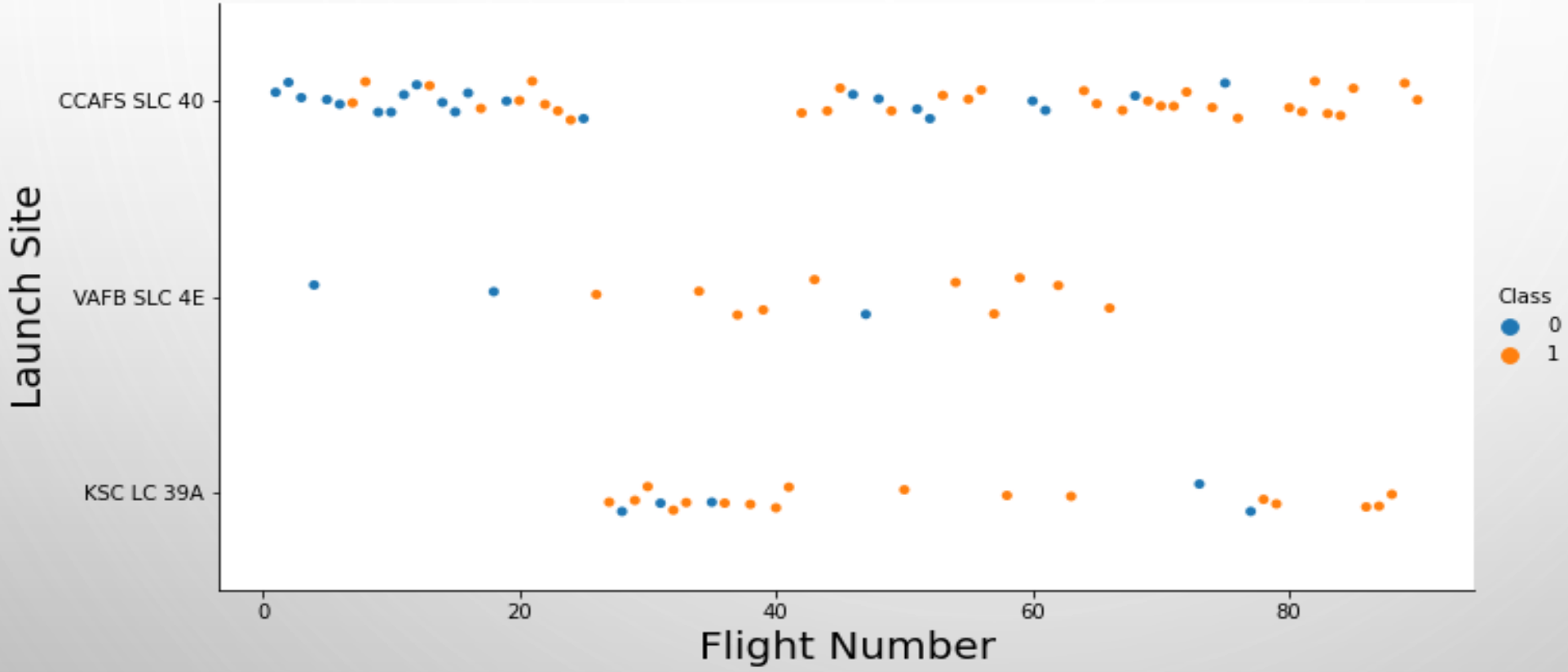
❑ Flight Number vs Payload mass



- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

EDA with Visualization

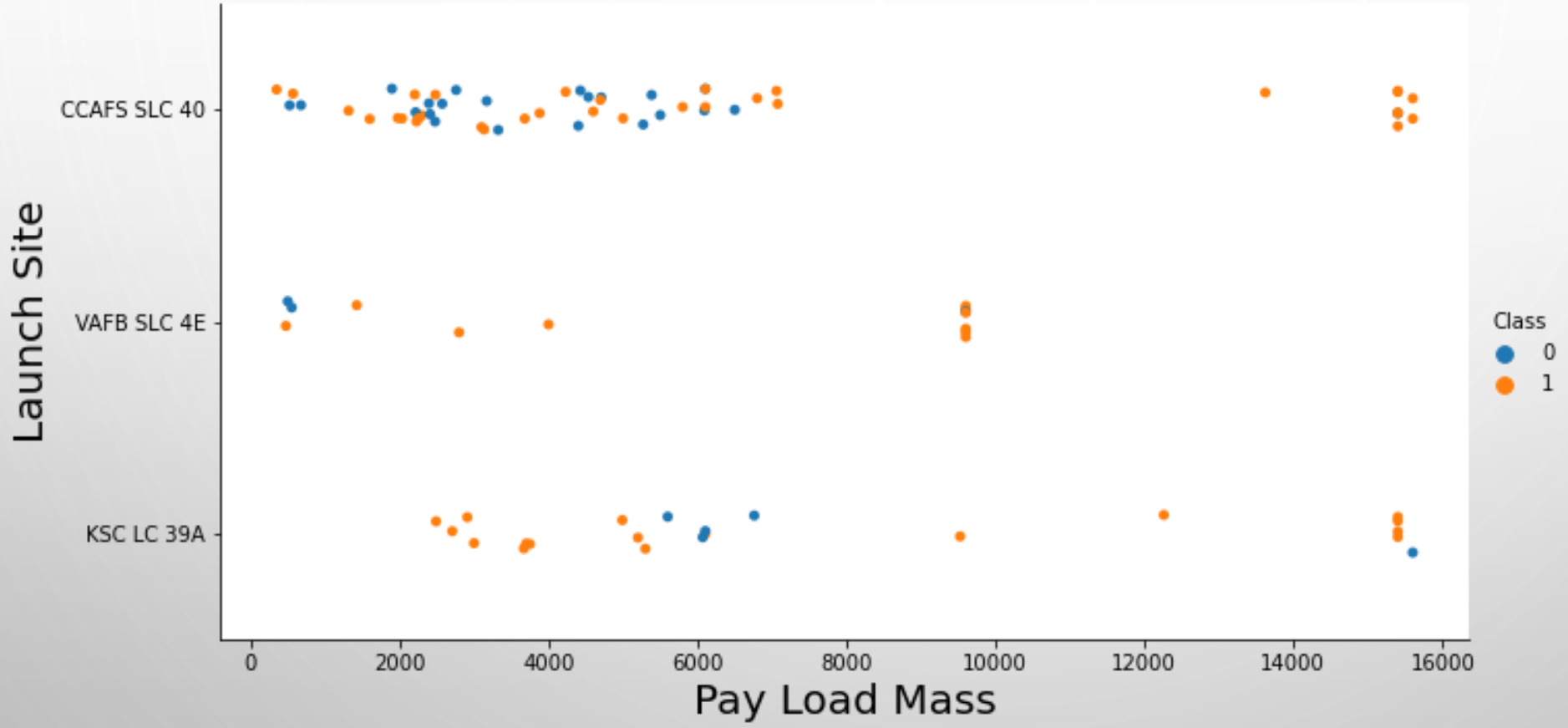
❑ Flight Number vs Launch Site



- Launch Site CCAF5 LC-40 has the highest number of launches. In starting of SpaceX project this launch site is used most frequently.

EDA with Visualization

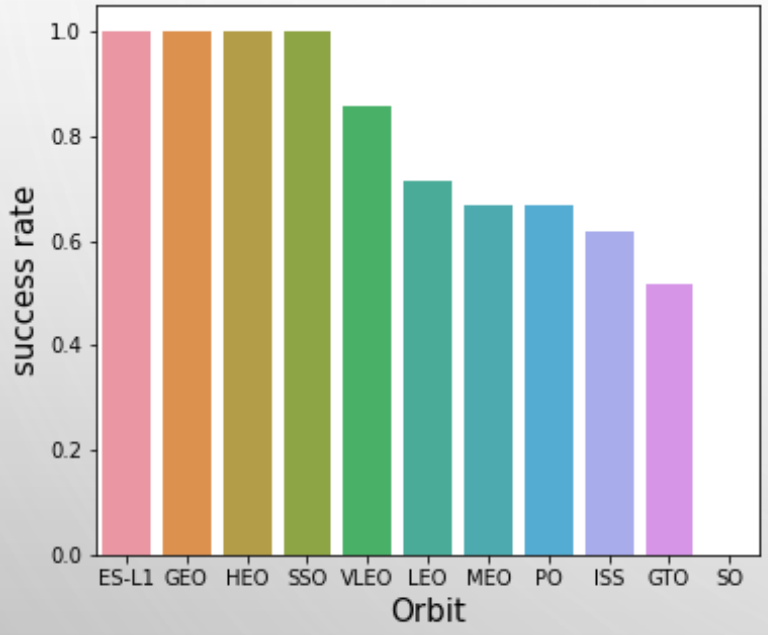
❑ Payload mass vs Launch Site

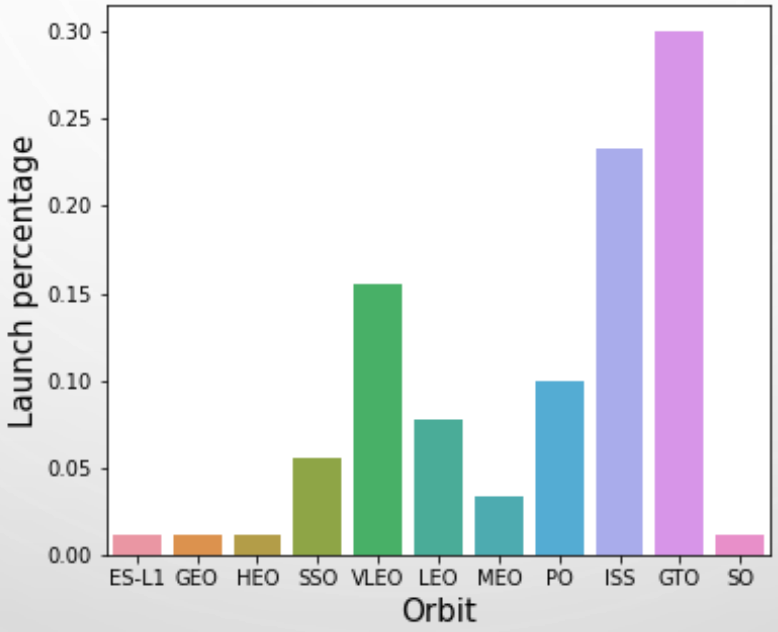


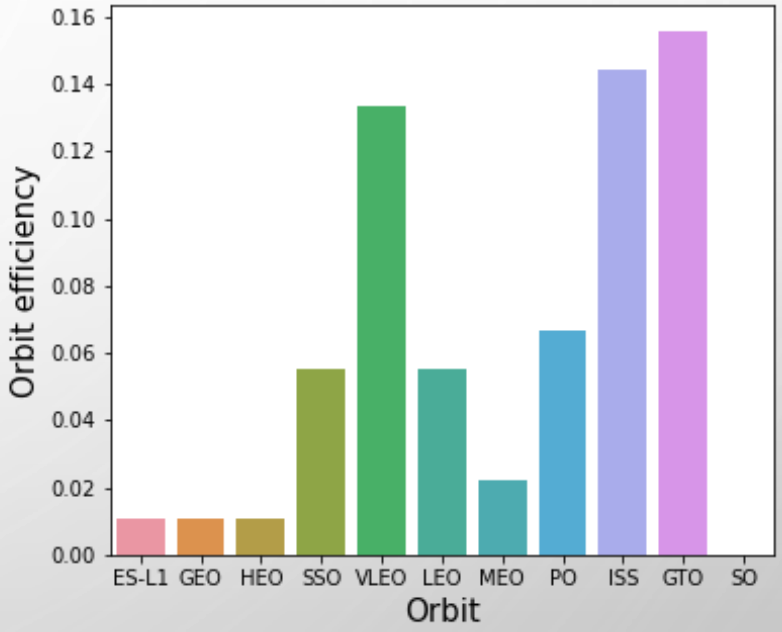
- The VAFB-SLC Launch Site there are no rockets launched for heavy payload mass(greater than 10000).

EDA with Visualization

❑ Relationship between success rate of each orbit type



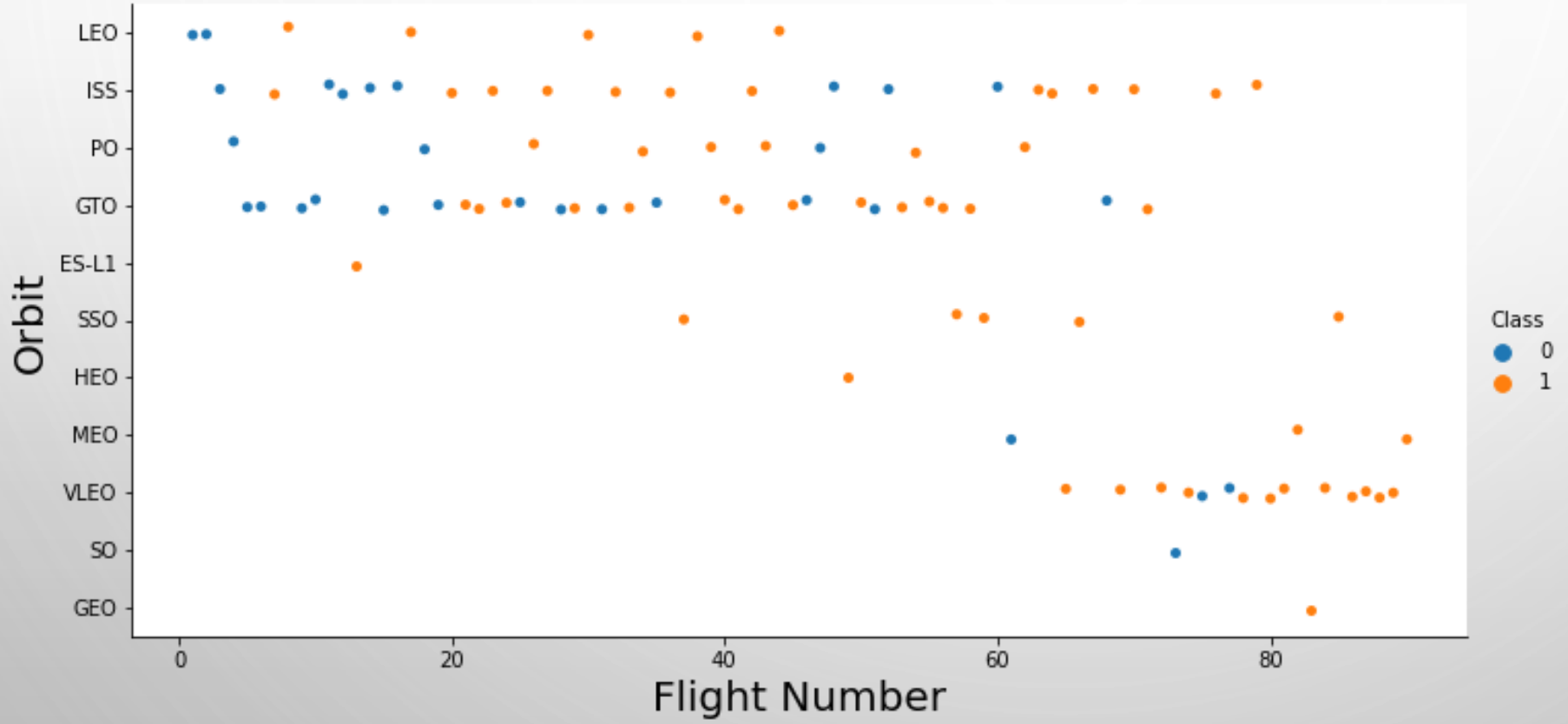




- ES-l1, GEO, HEO, SSO has highest success rate, but they have very few number of rocket launches. For orbit GTO the success rate is low but nuber of launches is highest.

EDA with Visualization

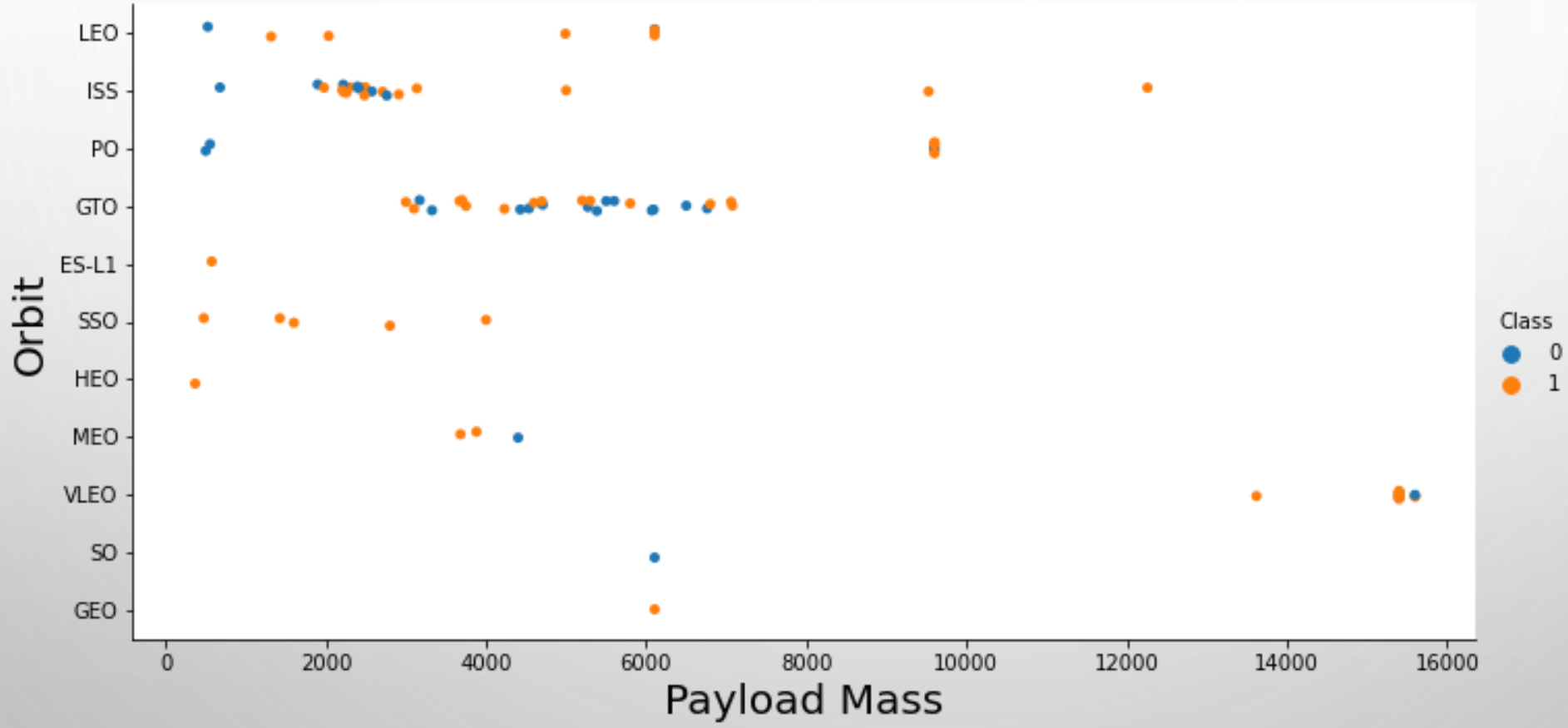
❑ Relationship between Flight Number and Orbit type



- The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

EDA with Visualization

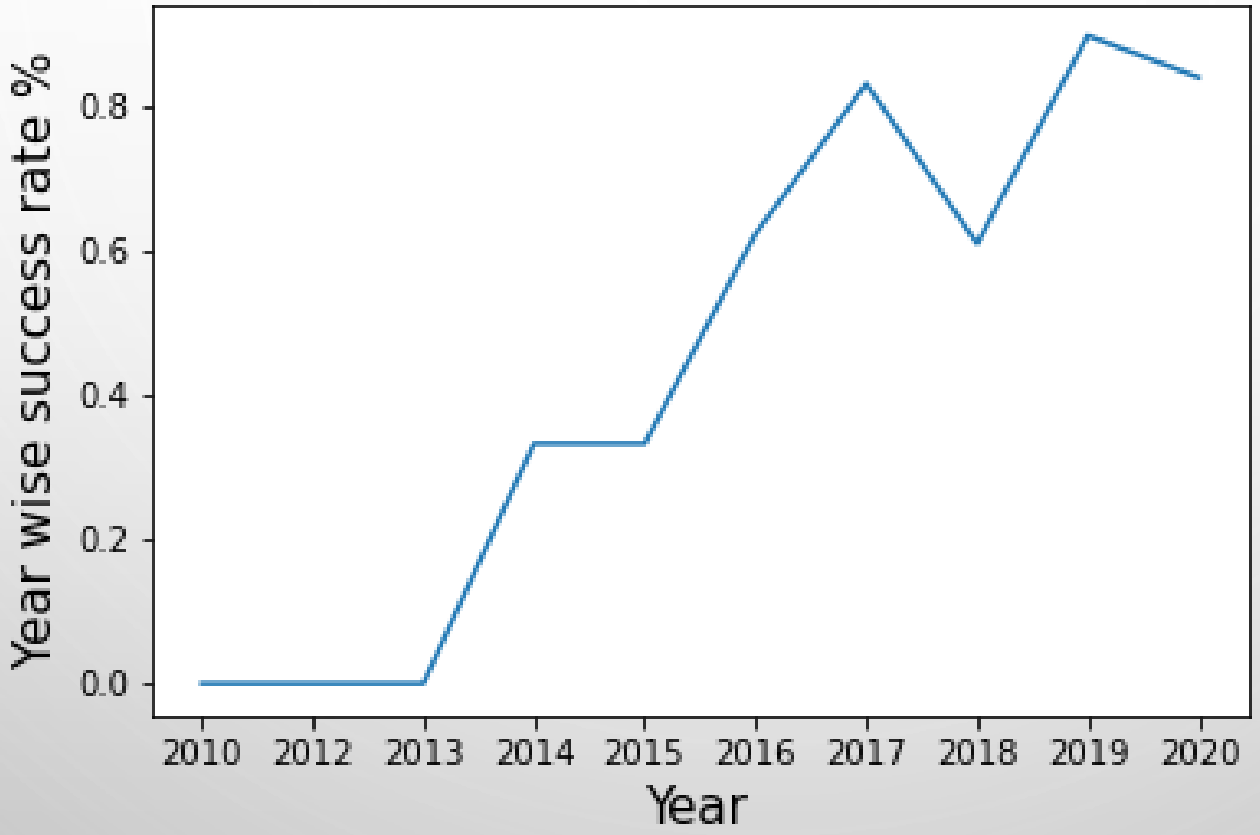
❑ Relationship between Payload and Orbit type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

EDA with Visualization

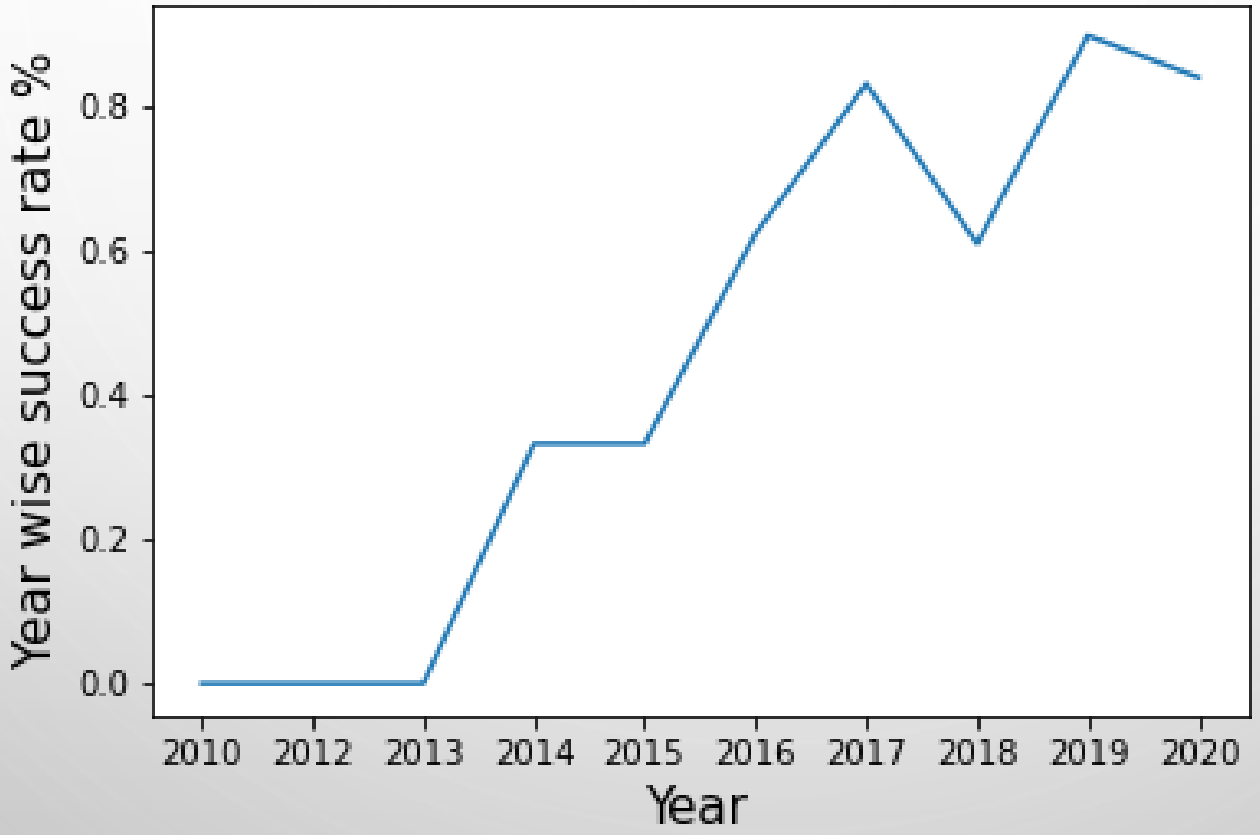
❑ Launch success yearly trend



- The success rate since 2013 kept increasing till 2020

EDA with Visualization

❑ Launch success yearly trend



- The success rate since 2013 kept increasing till 2020

EDA with SQL

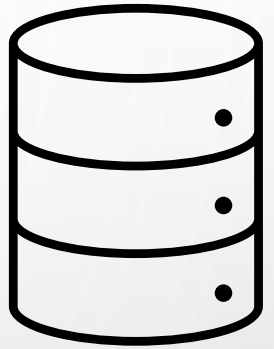
❑ Names of the unique launch sites in the space mission

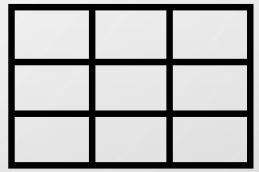
SQL Query

```
%sql select distinct launch_site from spacex
```

Query result

launch_site
CCAFS LC/40
VAFB SLC/4E
CCAFS SLC/40
KSC LC/39A

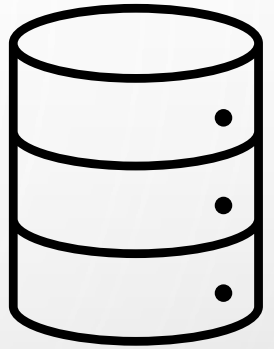




- The function DISTINCT in the query , will only show Unique values in the Launch Site column from table SpaceX

EDA with SQL

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

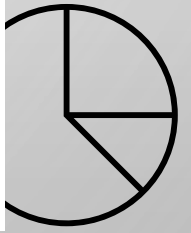


SQL Query

```
%sql select * from spacex\
where launch_site like 'CCA%'\
limit 5
```

Query result

date	time_utc	booster_version	launch_site	payload	payload_mass_kg	orbit	customer	mission_outcome	landing_outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC/40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC/40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC/40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC/40	SpaceX CRS/1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC/40	SpaceX CRS/2	677	LEO (ISS)	NASA (CRS)	Success	No attempt



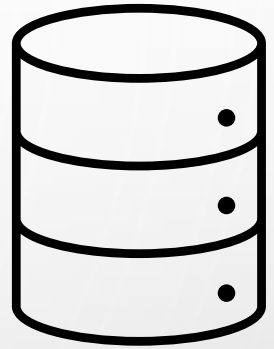
- Using the word **'limit 5'** in the query it will only show 5 records from spacex and **LIKE** keyword has a wild card with the words the percentage in the end suggests that the **Launch_Site** name must start with **KSC**.

EDA with SQL

❑ Total payload mass carried by boosters launched by NASA (CRS)

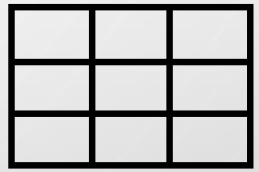
SQL Query

```
%sql select sum(payload_mass_kg) from spacex\
where customer = 'NASA (CRS)'
```



Query result

sum
45596



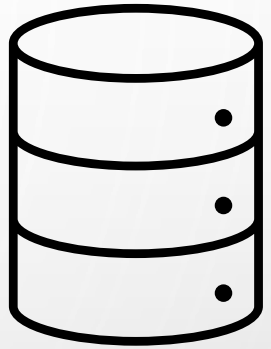
- Using the function **SUM** summates the total in the column **PAYLOAD_MASS_KG.** The **WHERE** clause filters the dataset to only perform calculations on Customer NASA (CRS)

EDA with SQL

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

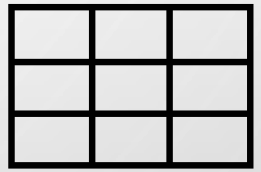
SQL Query

```
%sql select avg(payload_mass_kg) from spacex\
where booster_version = 'F9 v1.1'
```



Query result

avg
2928.4000000000000



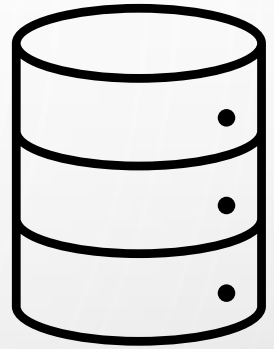
- Using the function **AVG** works out the average in the column **PAYLOAD_MASS_KG.** The **WHERE** clause filters the dataset to only perform calculations on **Booster_version** as **F9 v1.1**

EDA with SQL

- **Date when the first successful landing outcome in ground pad was acheived.**

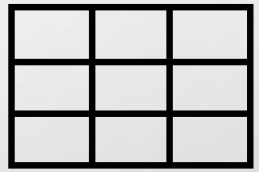
SQL Query

```
%sql SELECT MIN(DATE) FROM SPACEX \
WHERE LANDING_OUTCOME = 'Success (ground pad)';
```



Query result

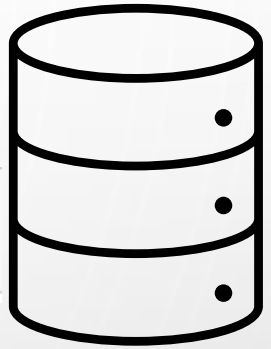
min
2015-12-22



- Using the function **MIN** works out the minimum date in the column **Date .**The **WHERE** clause filters the dataset to only perform calculations on **Landing_Outcome** as **Success (drone ship)**

EDA with SQL

- **Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**

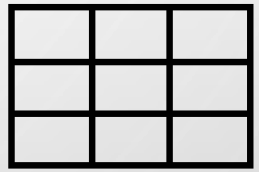


SQL Query

```
%sql select booster_version from spacex\
where landing_outcome = 'Success (drone ship)' and payload_mass_kg between 4000 and 6000
```

Query result

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



- The **WHERE** clause filters the dataset to **Landing_Outcome** as **Success (drone ship) .** The **AND** clause specifies additional filter conditions with **Between** keyword.

EDA with SQL

- Total number of successful and failure mission outcomes

SQL Query

```
%sql select count(*) from spacex\
where mission_outcome like 'Success%'
```

Query result

```
#success

 * postgresql://pos
1 rows affected.

count
100
```

SQL Query

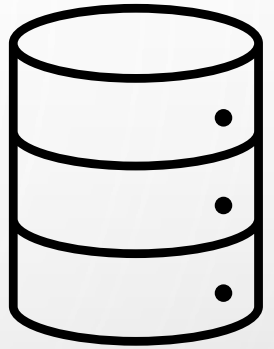
```
%sql select count(*) from spacex\
where mission_outcome not like 'Success%'
```

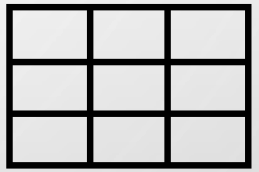
Query result

```
#failiure

 * postgresql:/
1 rows affected

count
1
```





EDA with SQL

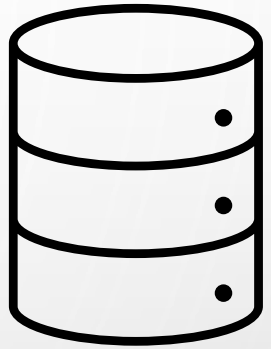
- names of the booster_versions which have carried the maximum payload mass

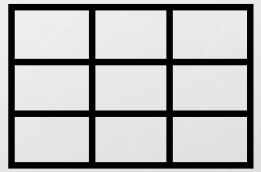
SQL Query

```
%sql select booster_version from spacex\
where payload_mass_kg = (select max(payload_mass_kg) from spacex)
```

Query result

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7





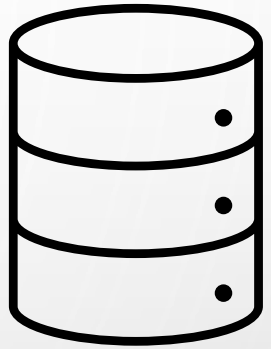
- Used subquery to find out the maximum of payload_mass_kg and then we used this value to filter out the booster with maximum payload mass.

EDA with SQL

- failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

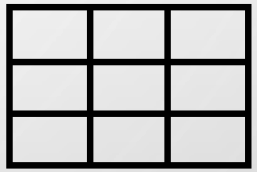
SQL Query

```
%sql SELECT  EXTRACT(MONTH FROM date) as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX\
WHERE EXTRACT(year FROM date) = '2015' AND LANDING_OUTCOME = 'Failure (drone ship)'
```



Query result

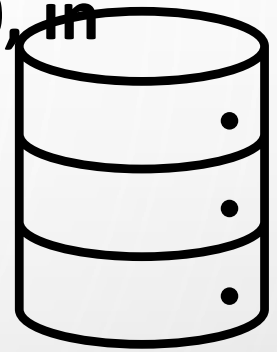
month	booster_version	launch_site
10	F9 v1.1 B1012	CCAFS LC/40
4	F9 v1.1 B1015	CCAFS LC/40



- Extracted the Month and Year from date column and the year is used to filter the result with year of 2015.

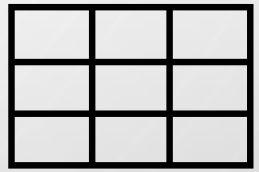
EDA with SQL

- 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 Query

```
%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

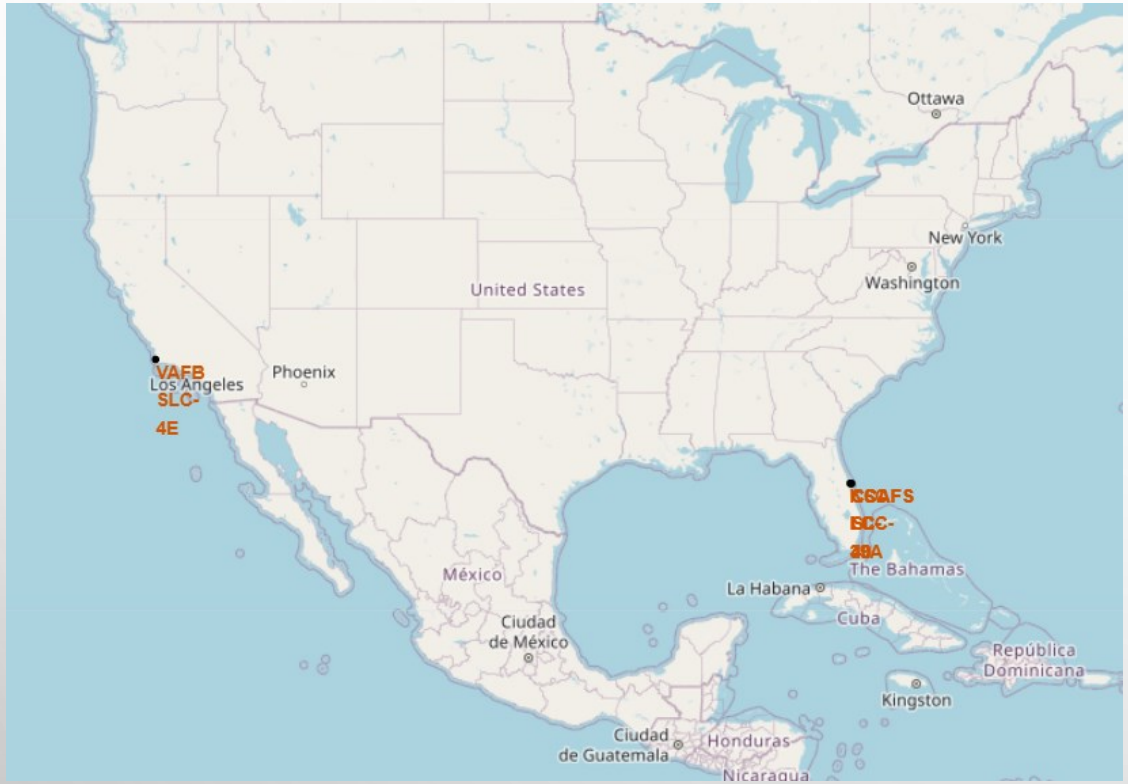


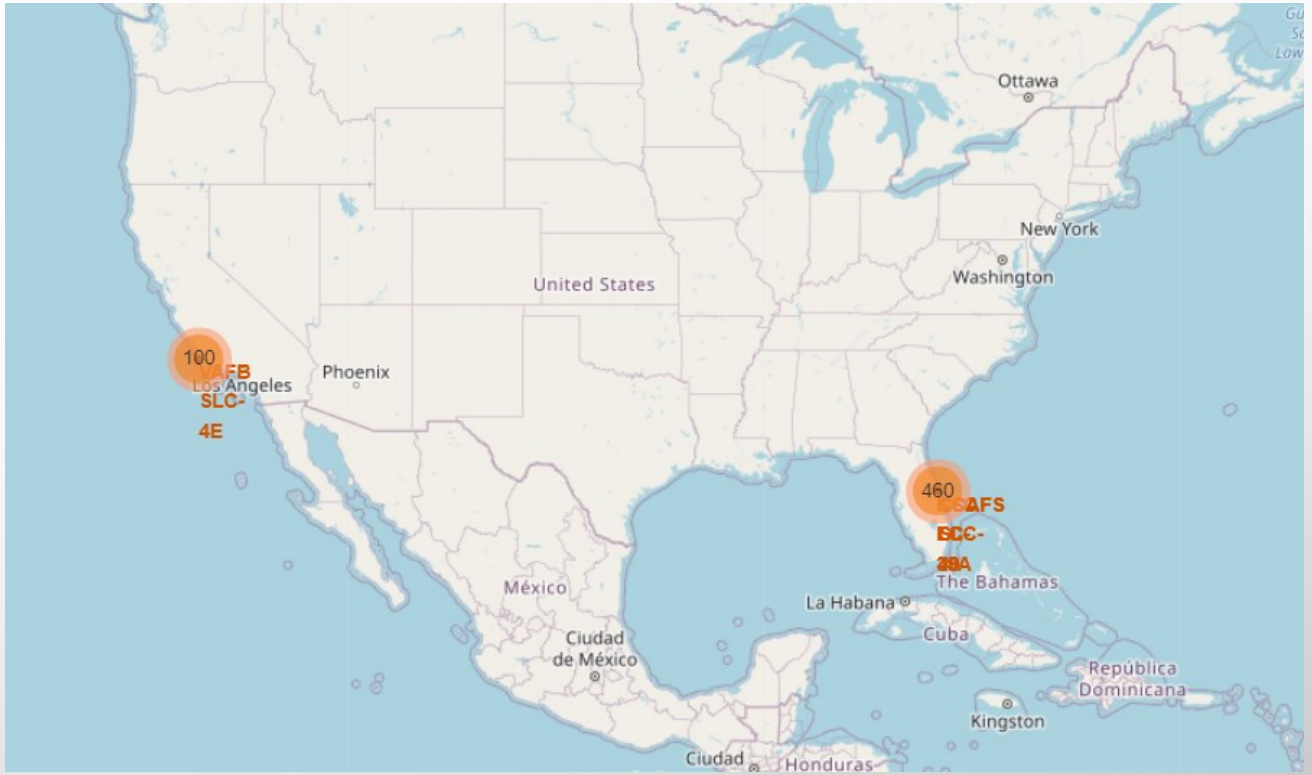
Query result

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

Interactive map with Folium

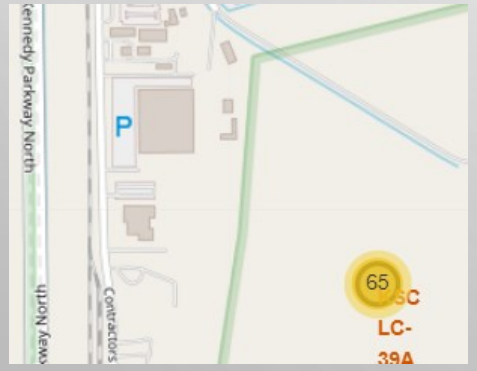
- Launch Site locations



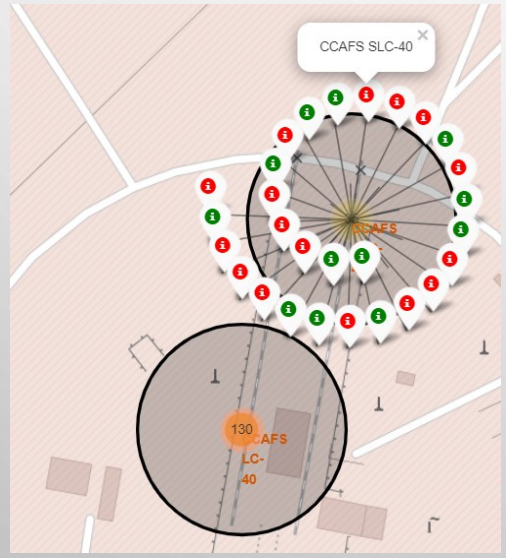


Interactive map with Folium

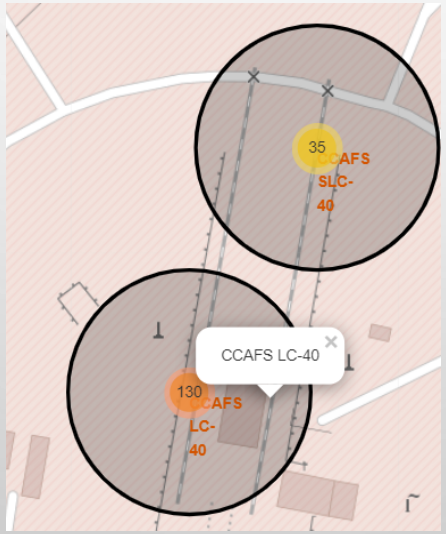
❑ Florida side Launch Sites



KSC LC-39A

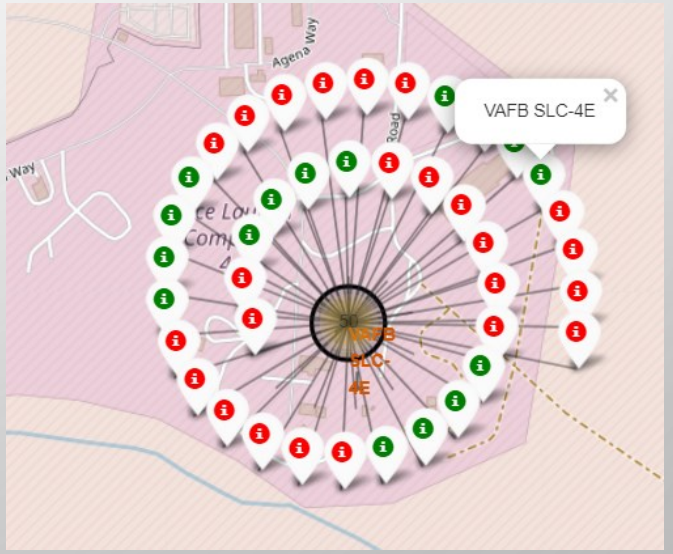


CCAFS SLC-40



CCAFS LC-40

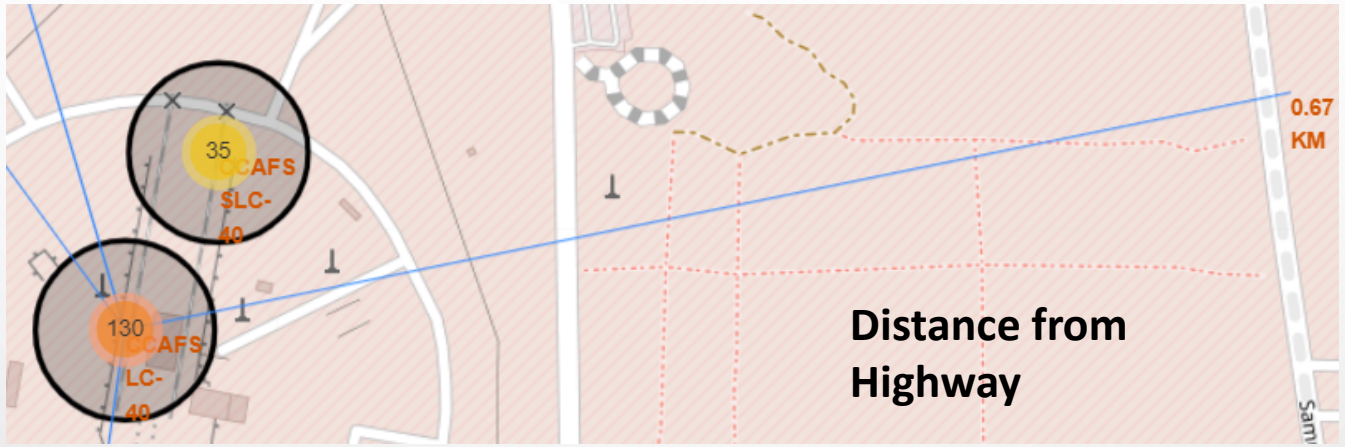
❑ California side Launch Sites

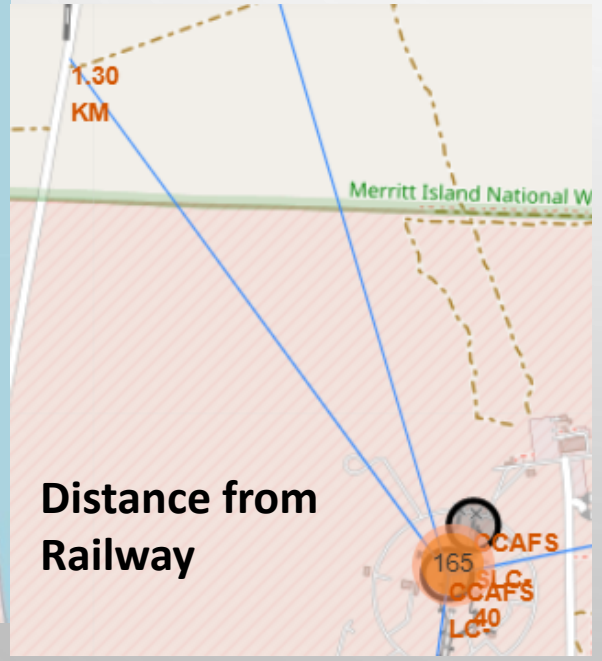


VAFB SLC-4E

Interactive map with Folium

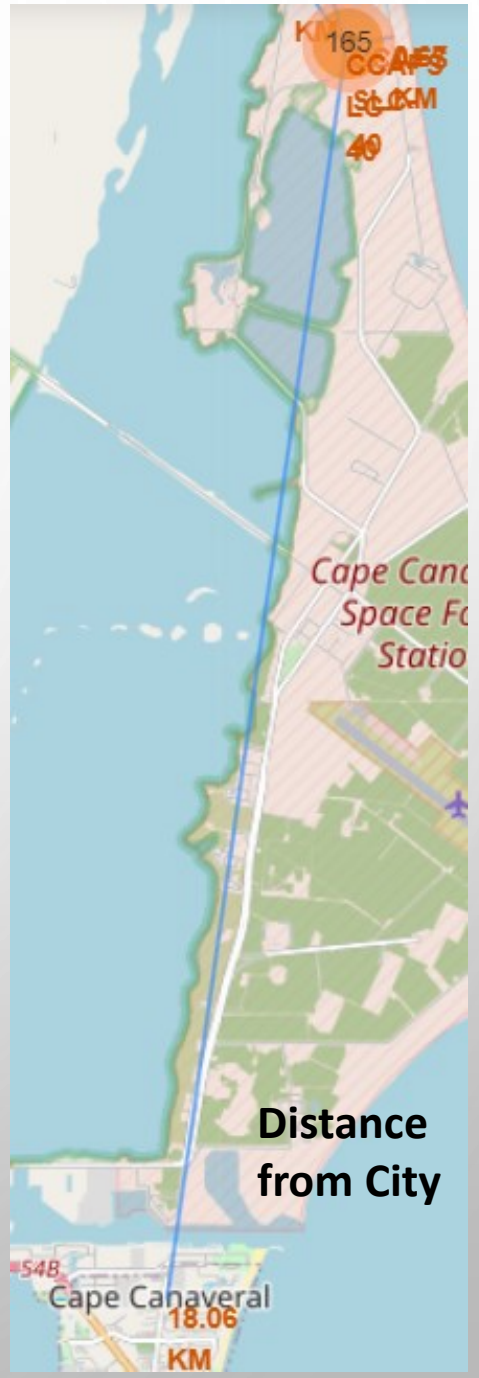
❑ Neighbor of Launch Sites CCAFS LC-40





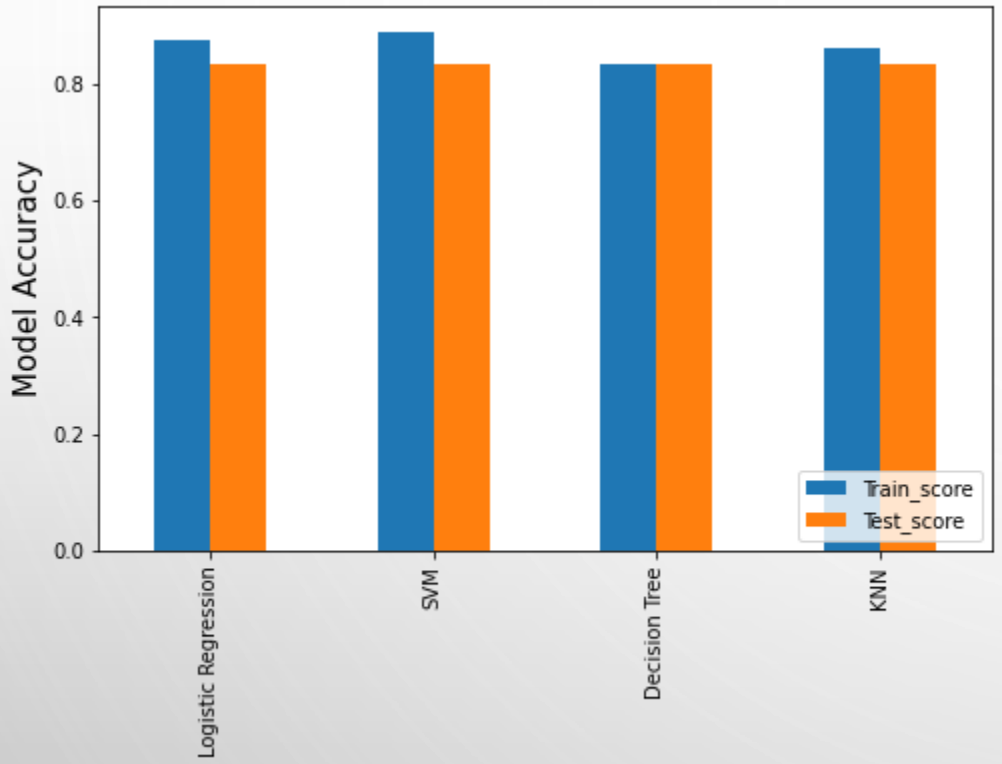
INFERENCE:

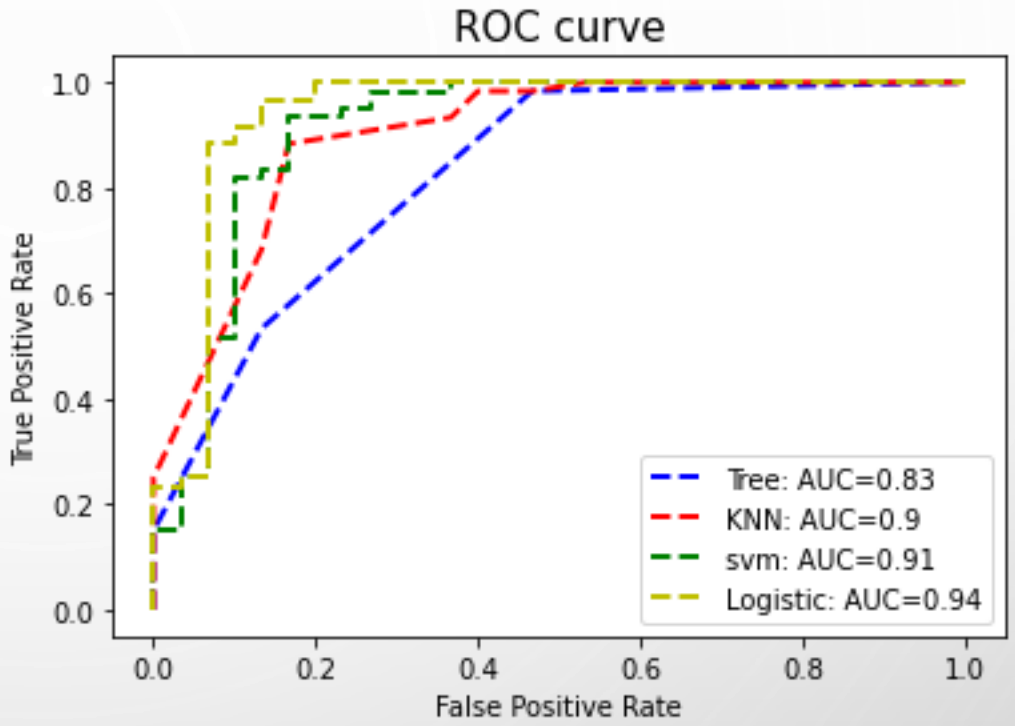
- So, the launch sites are close to coastal area. They are connected to highway, and railway. But the launch sites are way far from cities where human lives are inhabiting.



Predictive Analysis

Model Comparison





	index	Train_score	Test_score
0	Logistic Regression	0.875000	0.833333
1	SVM	0.888889	0.833333
2	Decision Tree	0.833333	0.833333
3	KNN	0.861111	0.833333

Analysis:

- Model scores on train and test data are close to each other.
- As we can see **the train score and the test score is highest for Logistic Regression** model, and the **area under receiving operating characteristics curve is highest** for Logistic Regression. Hence **Logistic Regression** model is the best performing model
- The Best model Accuracy on test data is 83.33%

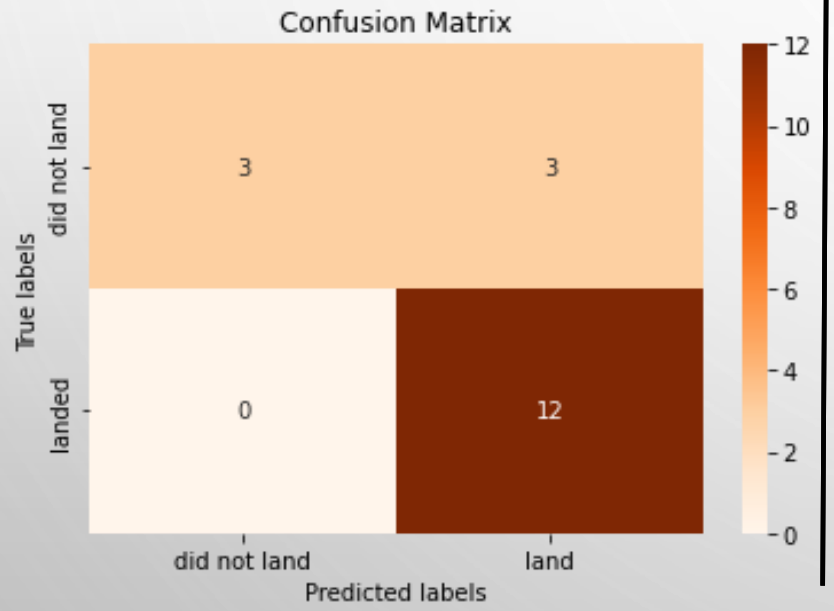
Predictive Analysis

Best Model Parameters

```
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
```

Tree model Results on Test Data



- The Logistic regression model is able to classify each class distinctly.

Classification Report

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Analysis:

- Recall and precision for success class in good, more than 80%.
- The Recall for the unsuccessful landing is 50% which is concerning. This means the false unsuccessful landing is predicted as successful. So, based on the model if we decide to invest on Rocket assuming that it will land successfully , there is a high chance that we will lose the money.

Feature Importance

	Coef
Legs_True	0.121337
GridFins_True	0.109195
ReusedCount	0.078297
Block	0.053276
LandingPad_5e9e3032383ecb267a34e7c7	0.053148
LandingPad_5e9e3032383ecb6bb234e7ca	0.051316
FlightNumber	0.047586
Serial_B1036	0.037581
Serial_B1006	0.036741
Serial_B1049	0.036299

Analysis:

- Feature coefficients are calculated based on normalized data. So we can compare the most influencing features that will decide the landing success of rocket

Conclusion

- The best model is Logistic Regression to predict the successful landing.
- Recall and precision for success class in good, more than 80%.
- The **Recall for the unsuccessful landing is 50%** which is concerning. This means the false unsuccessful landing is predicted as successful. So, based on the model if we decide to **invest on Rocket assuming that it will land successfully , there is a high chance that we will lose the money**. So the model need to be improved. If we can increase the data points the model can be improved
- If the rocket has legs, then there is high chance of landing. **Availability of leg** in the rocket is the most important factor.
- After leg feature if grid fans are available, chance of successful landing is high. **GridFan** feature is the second most important factor.
- If the rocket is already **reused**, then there is high chance it will successfully land again.
- It can be observed that launch site KSC LC-39A has the highest success rate.

in the end we can observe that SpaceX is improvising its launching year to year. In the next couple of years, the success rate will improve further.

APPENDIX

Description	Link
Data collection via SpaceX API	Click
Data collection via Web Scraping	Click
Data Wrangling	Click
EDA with data Visualization	Click
EDA with SQL	Click
Interactive map of launch sites	Click
Predictive Analysis	Click