IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shoukath Ali
13th May,2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Feature engineering
  - Exploratory Data Analysis (Sql and Visualization method)
  - Interactive Visualization and Dashboard
  - Predictive analysis

- Summary of all results -

-> Built an interactive dashboard for analysis.

-> Based on all the features/factors from a dataset provided, found an accurate and efficient machine learning predictive model. (Model – Decision Tree, Accuracy ~ 87%)

Introduction

- SpaceX rockets are Falcon 9 costs 62 million dollars, while others costs upward of 162 million dollars.
- All the above cost savings is due to its capability to reuse the rocket.
- Therefore, if we can determine whether the first stage will land successfully then we can determine the cost of the launch.
- So, the ultimate aim is to build a ML pipeline to predict the land(success/ failure)
- Insights to find -
- What are the important deciding / weighted features?
- Rate of dependence of the factors on the successful landing ?
- Adjustment of the factors for successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through API
- Perform data wrangling
  - Data was filtered and converted into suitable format for analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build the models from a library (sklearn) based on the different measuring parameters, tune model with random parameters and find the best model based on accuracy.

Data Collection

- Describe how data sets were collected.
- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Pull website using the request library(API) and collect through json format and convert into dataframe

- GitHub URL of the completed SpaceX API calls notebook - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/1.%20Data_collection_api.ipynb

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-stor
_spacex_api.json'
```

We should see that the request was successfull with the 200 status response cc

```
response.status_code
```

200

Now we decode the response content as a Json using `.json()` and turn it into a

```
# Use json_normalize meethod to convert the json result into a d
rec = requests.get(static_json_url)
rec = rec.json()

data = pd.json_normalize(rec)
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	tbd	net	window

Data Collection - Scraping

- Get the Website data and scrap using library and convert into dataframe

- GitHub URL of the completed web scraping notebook - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/1.1.Data_collection_webscrap.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URI

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response tex
soup = BeautifulSoup(response.content,'lxml')
```

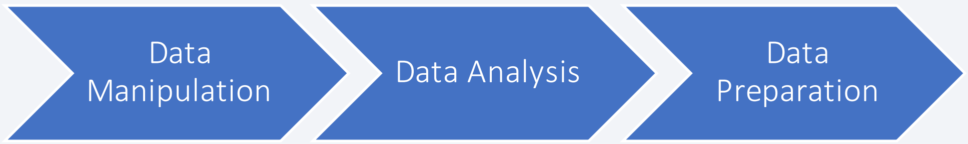
Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

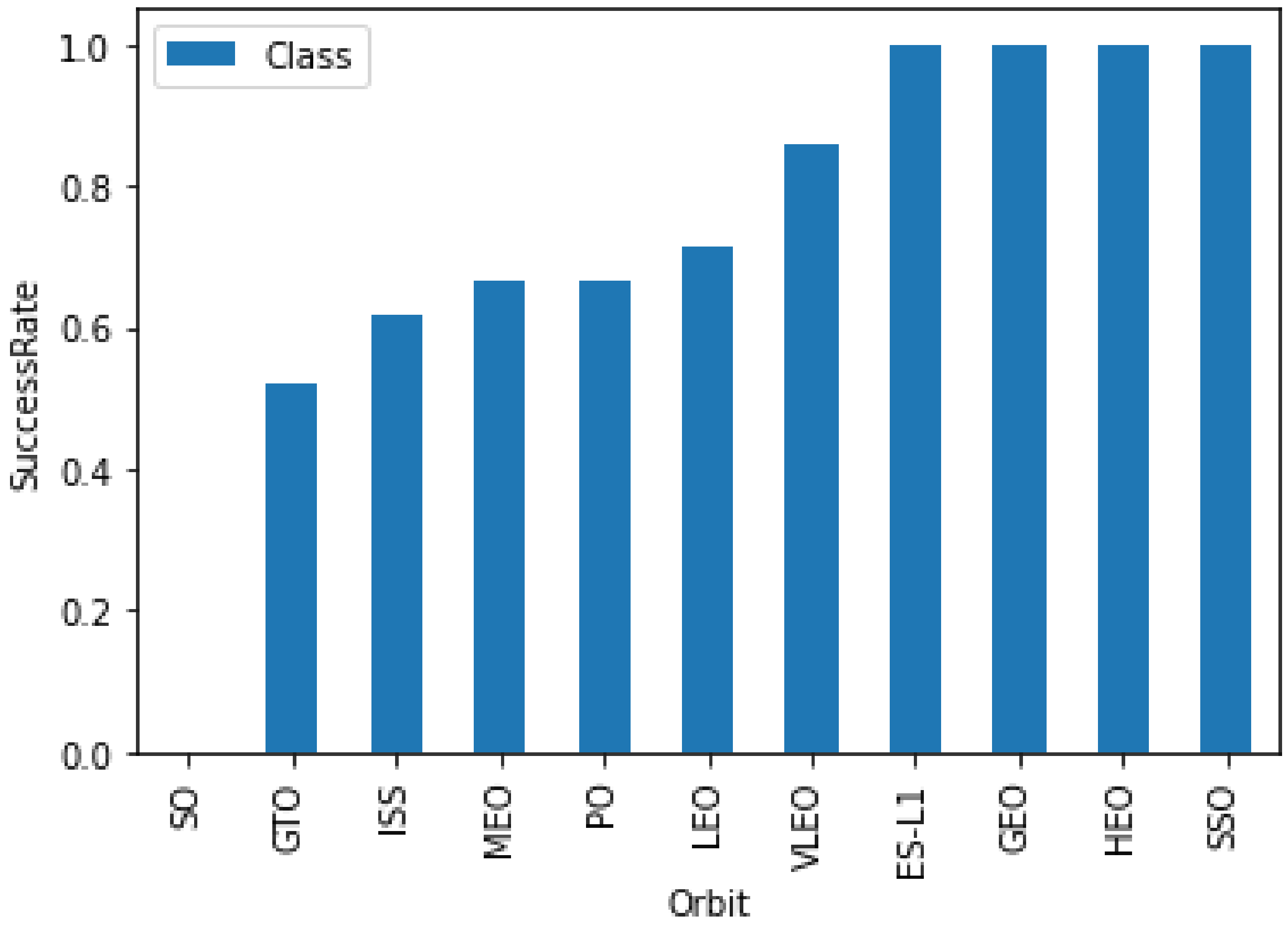
- Data was first filtered and converted into proper data type, check for null values and replace with some statistical methods, with analysis convert categorical value to a numerical .



- GitHub URL of your completed data wrangling - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/2.Data_wrangling.ipynb

EDA with Data Visualization

- EDA between SuccessRate vs features(Like Orbit) to determine important features to consider.
- GitHub URL - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/4.EDA_Viz.ipynb



EDA with SQL

- Used wildcard to simply search query, aggregate functions, rank function and some conditions to perform analysis
- GitHub URL - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/3.EDA_SQL.ipynb

```
%%sql

SELECT landing__outcome,COUNT(*) as Count,RA
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a
Done.
```

landing__outcome	COUNT	RANK
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

Build an Interactive Map with Folium

- Added markers, circle, lines in a folium map to get a brief about the locations of the launch site
- Marker is used to locate at exact point and denote the success/failure, Circle is used to cover the entire building/location radius and lines is used to calculate the distance between the points(like coast/ cities/ railway)
- GitHub URL - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/5.Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Interactions – Can select different Launch Sites , Can select a range of payload
- Figure – Pie chart , Scatter plot of Success Rate vs timeline with respect to launch site and payload mass.
- Pie chart - is used to show the percent of success rate with respect to launch site, in case of 'all' option the overall success rate of all launch site is calculated.
- Scatter plot – helps to identify the success rate vs date time with respect to launch site and payload range.
- GitHub URL - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/6.app.py

Predictive Analysis (Classification)

- With Sklearn module, we access a model and predefine a parameter with a list of hyperparameters for tuning and built a model
- Library GridCV is helpful for choosing the best parameter of model.
- Now with the help of the score the model with best parameters is selected.
- Among those models the best model is selected with the help of evaluation of score between tested and predicted outcomes.
- GitHub URL - https://github.com/shoukath-ali/Spacex_IBM_DS_Capstone/blob/main/7.Machine%20Learning%20Prediction.ipynb

Results

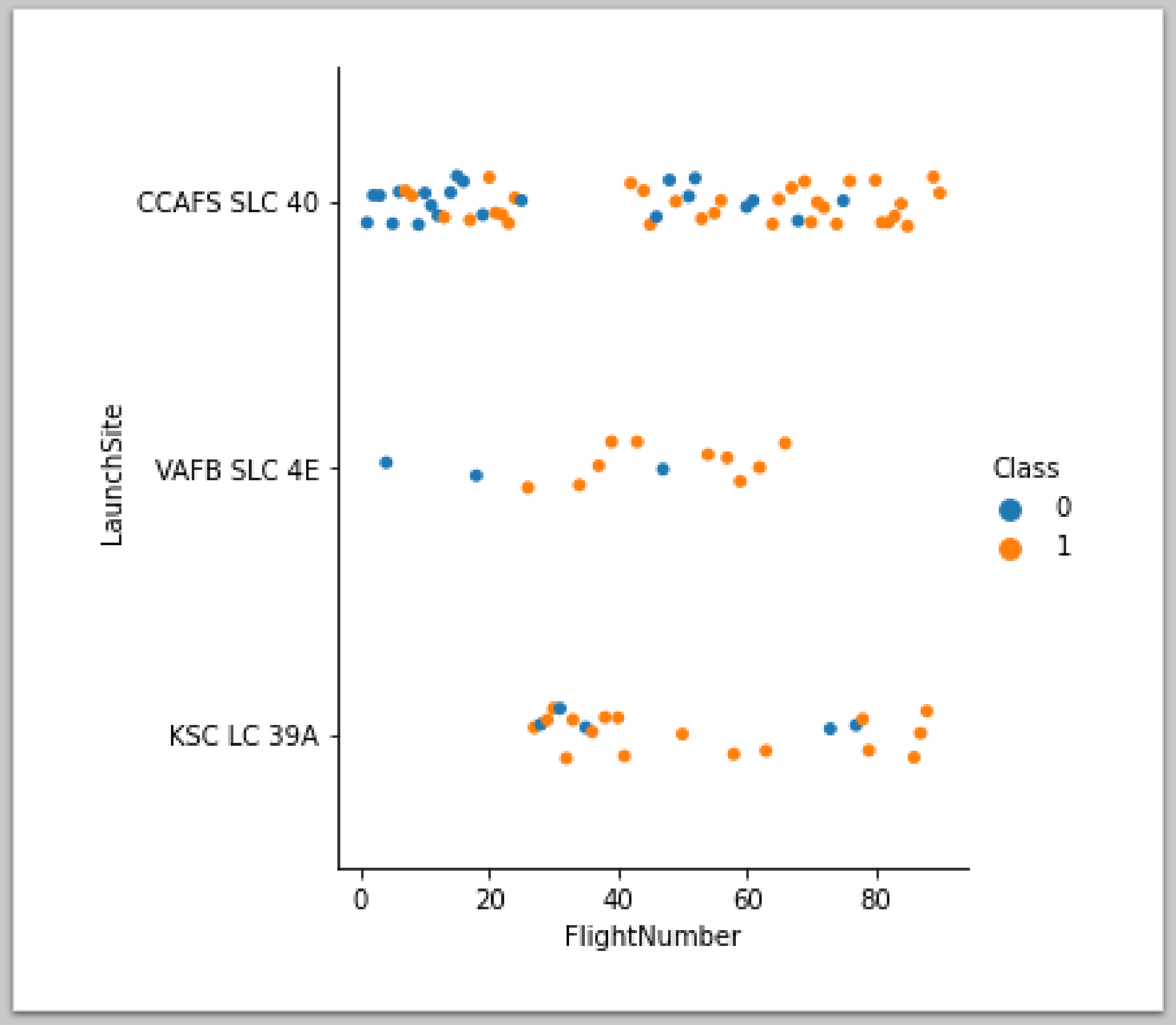
- EDA Result – Determined the important factors/labels to consider for analysis or build ML models.
- Interactive analytics demo in screenshots
- Predictive analysis results
  - Best Model – DECISION TREE
  - Model Score – 87%

Section 2

Insights drawn from EDA

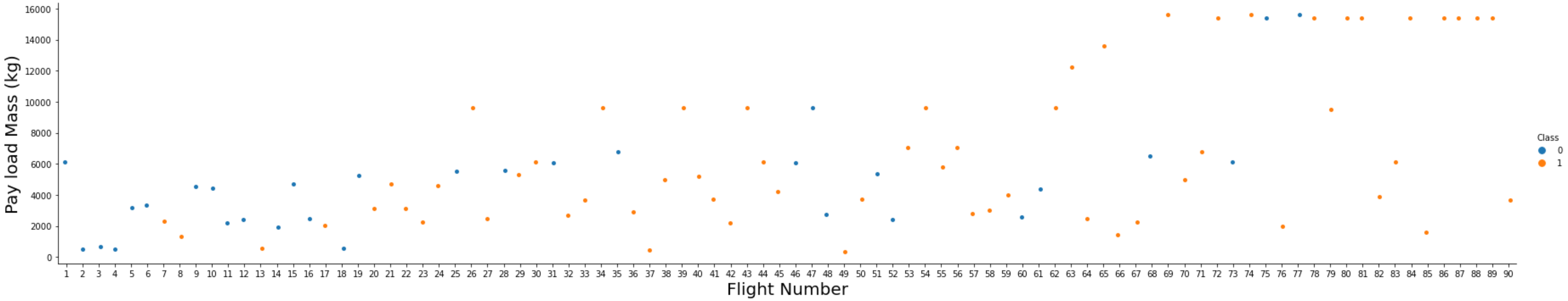
Flight Number vs. Launch Site

- Larger the Flight Number , High chances of Successful landing (class 1)



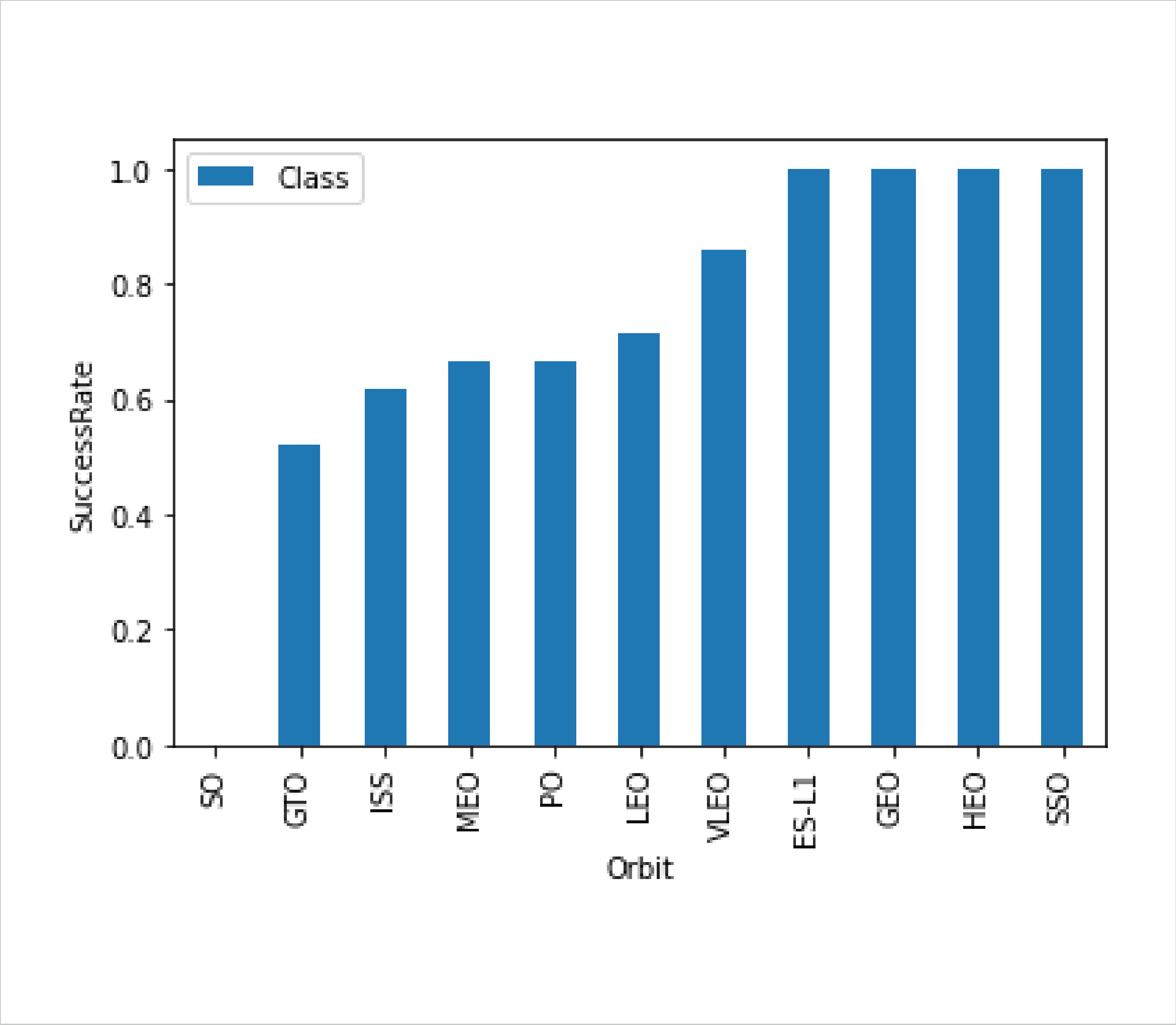
Payload vs. Flight Number

- Gradual Increment in Flight Number with increase in Payload, increases the chances of successful landing



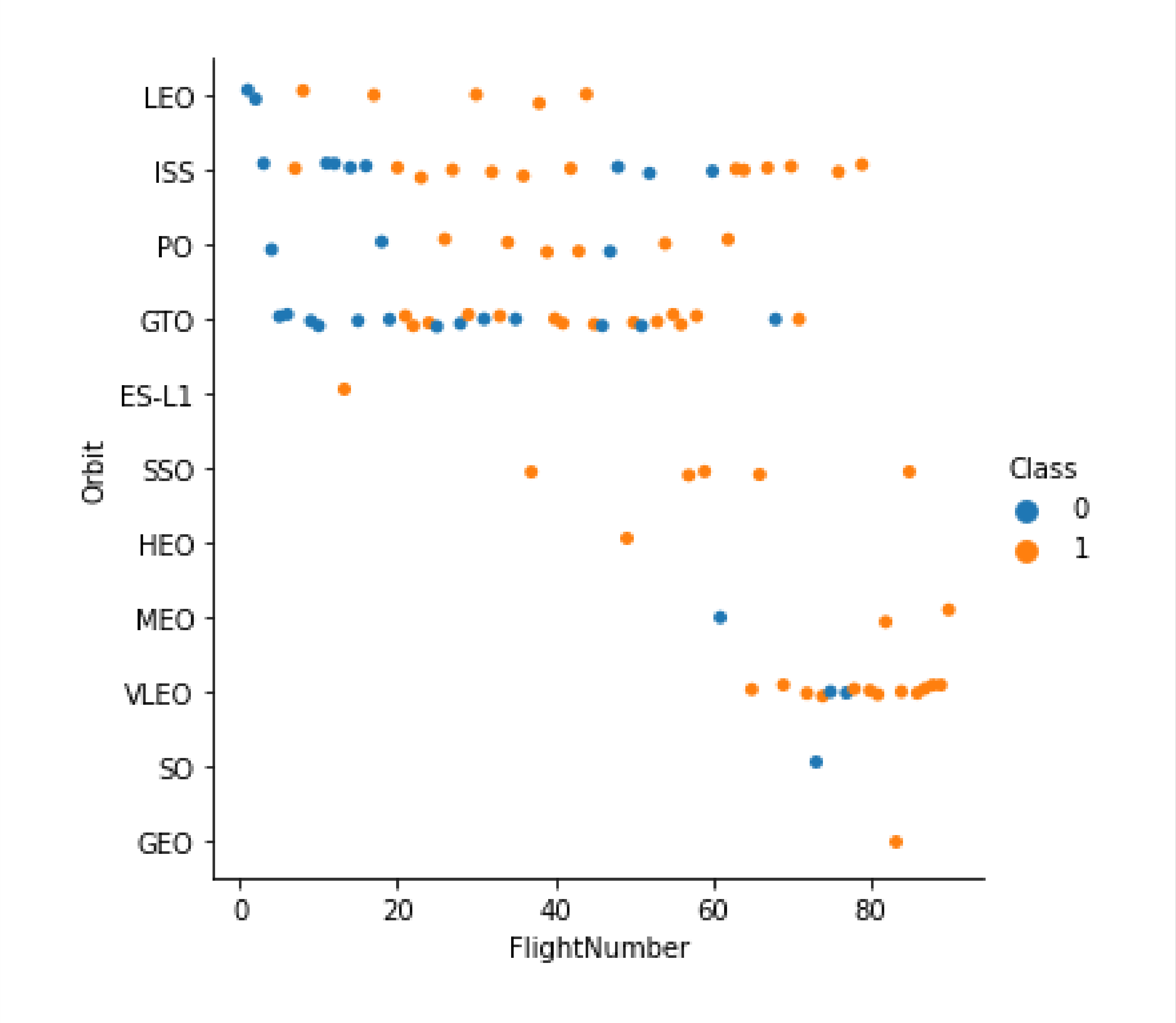
Success Rate vs. Orbit Type

- Orbit type – ES-L1 , GEO, HEO, SSO has a Success Rate of 1



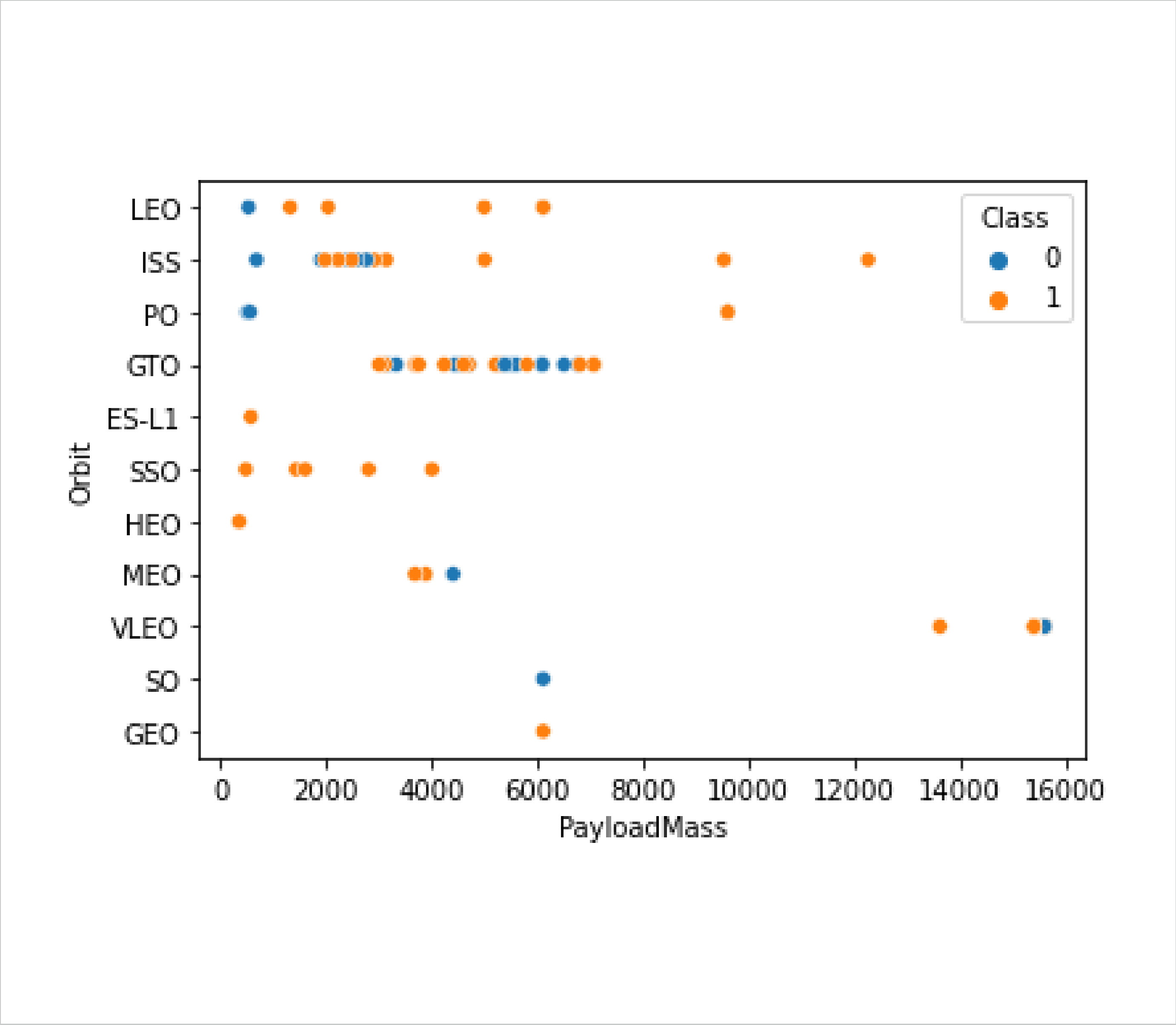
Flight Number vs. Orbit Type

- No, proper relationship between orbit vs Flight Number in case of GTO
- For remaining orbit, there is gradual improvement with respect to Flight Number



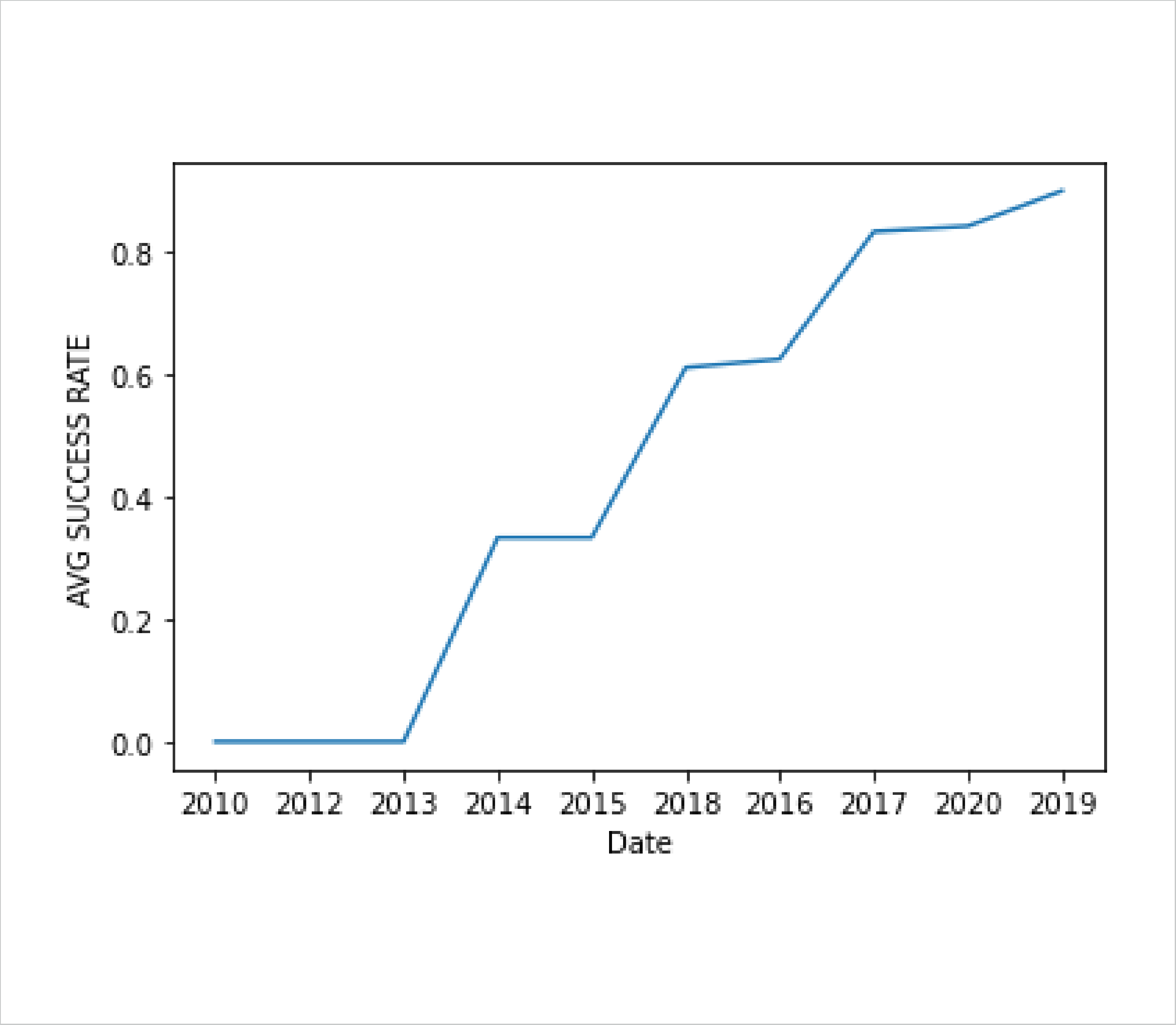
Payload vs. Orbit Type

- With heavy payloads the successful landing rate are more for Polar,LEO and ISS.
- GTO we cannot distinguish, as both positive and negative are seen



Launch Success Yearly Trend

- Launch Success rate since 2013 kept increasing



All Launch Site Names

- We use UNIQUE constraint and SELECT
- Statement to get all launch sites.

```
%%sql

SELECT UNIQUE launch_site FROM SPACEXDATASET
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-8
Done.
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`
- We use wild card to find the data beginning with 'CCA' and LIMIT to get top 5 data.

```
%%sql
SELECT * FROM SPACEXDATASET WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l
Done.
```

DATE	time_utc_	booster_version	launch_site	
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualificat
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSat of Brouere
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo f
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	Space
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	Space

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Aggregate function SUM is used to calculate total and WHERE condition to filter booster dataset

```
%%sql
SELECT SUM(payload_mass__kg_) as total_payload FROM SPACEXDATASET WHERE customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

total_payload
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- We use AVG function to calculate average with WHERE condition to filter booster

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

In [13]:

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- MIN function helps to find least date, according to data 22nd December 2015 was first successful landing outcome.

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCLAND FROM SPACEXDATASET WHERE landing__outcome ='Success (ground pad)';
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdoma
Done.
```

first_sucland
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- BETWEEN is used to simplify the task of filtering

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

```
%%sql
SELECT * FROM SPACEXDATASET WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- COUNT function helps to group the mission outcome

Here,

- 1 is Failure
- 99 is Successful
- 1 is unknown

```
%%sql
SELECT COUNT(mission_outcome) FROM SPACEXDATASET GROUP BY mission_outcome
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1c
Done.
```

1
1
99
1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Subquery is used to calculate the max payload mass and applied as a condition to a query.

```
%%sql
SELECT booster_version FROM SPACEXDATASET WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXDATASET)
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Wild card is used to filter Date and 'AND' is used to combine two conditions

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

```
%%sql
SELECT date,booster_version,launch_site,landing__outcome FROM SPACEXDATASET WHERE landing__outcome = 'Failure (drone ship)' AND date LIKE '2015%'
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- 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
- RANK function is used on COUNT which is calculated by grouping the landing outcomes and Conditions were applied.

```
%%sql

SELECT landing__outcome,COUNT(*) as Count,RANK() OVER(ORDER BY Count desc) as RANK FROM SPACEXDATASET WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
```

```
 * ibm_db_sa://yhc11747:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

landing__outcome	COUNT	RANK
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

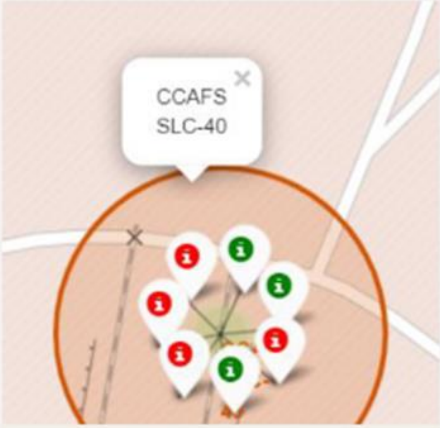
Section 3

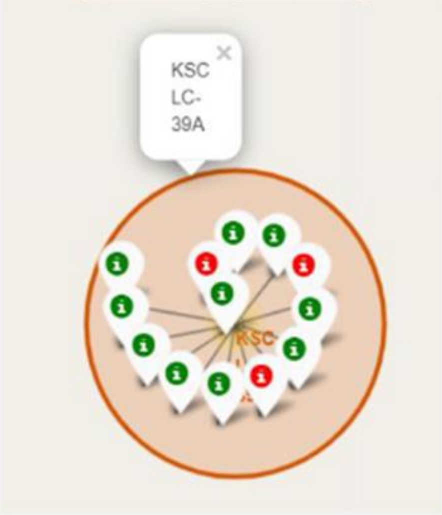
Launch Sites Proximities Analysis

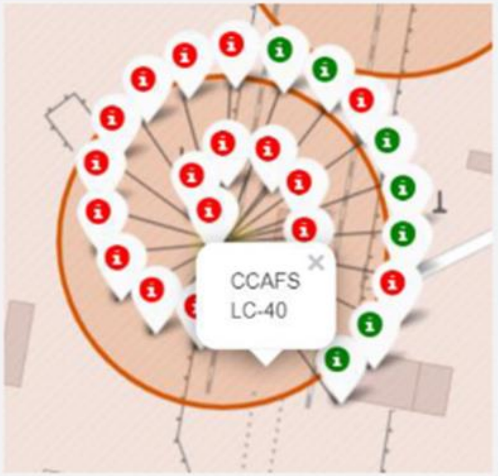
Launch Sites

Launch sites at coast line of Florida and California

Launch Sites with Markers







Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Distance between Launch Site and Landmarks

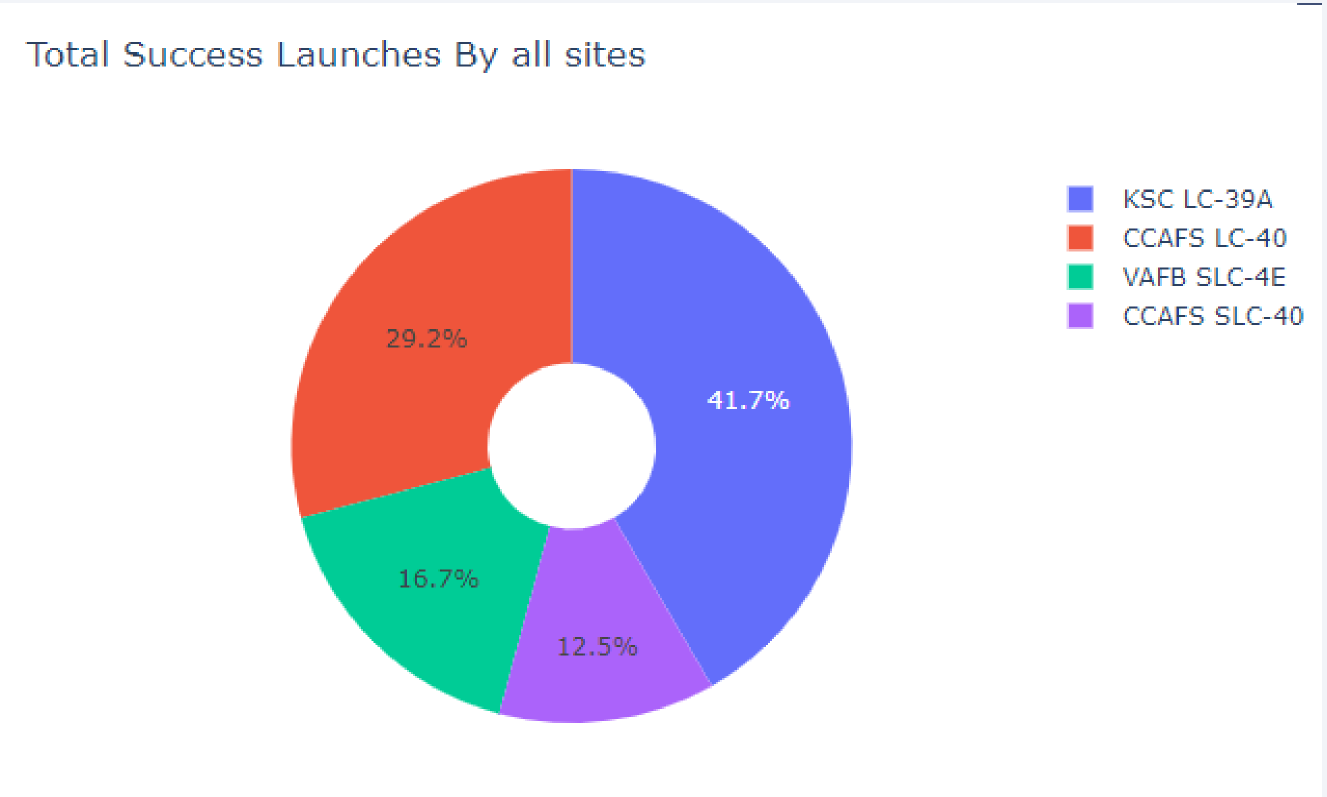
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash

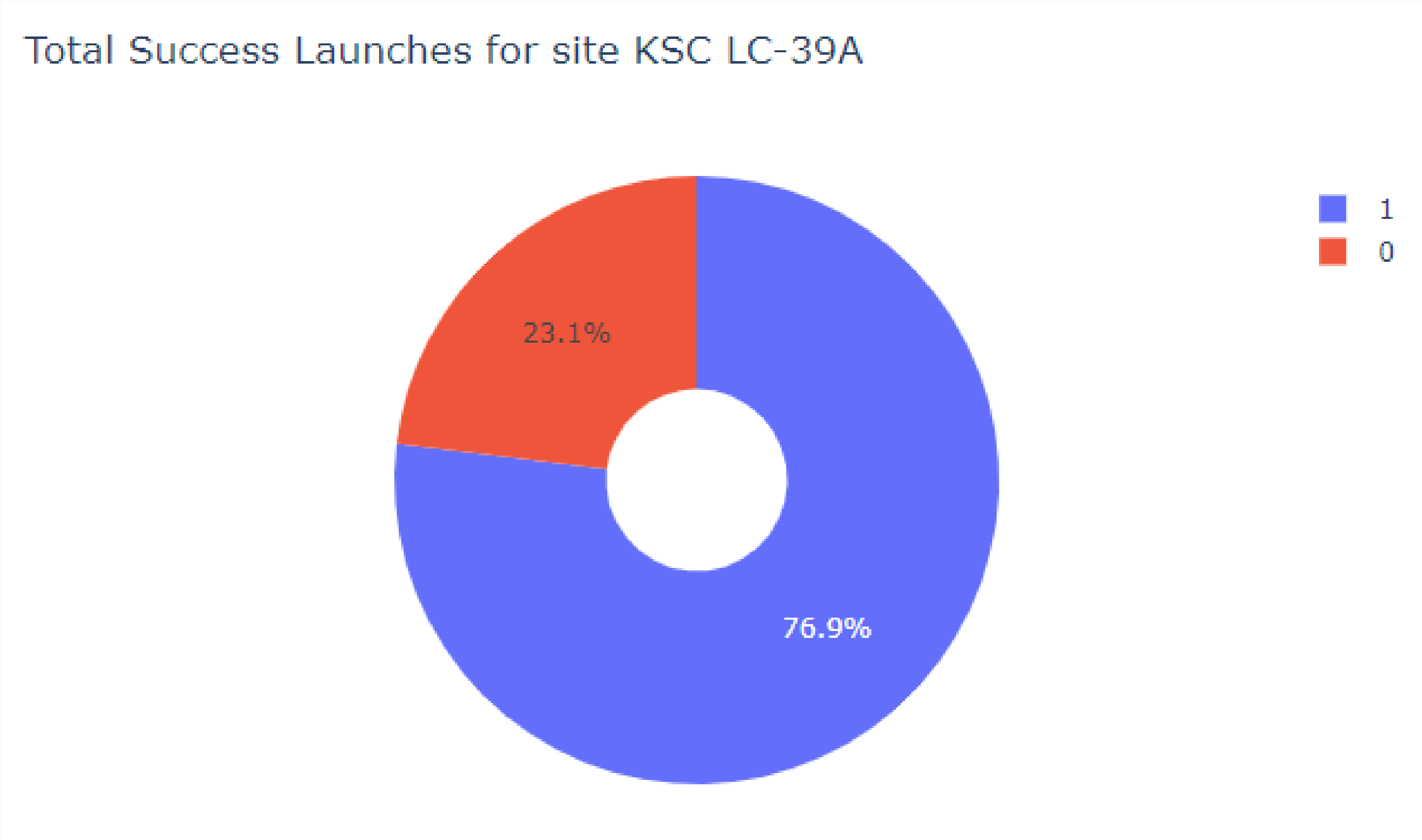
Launch success count for all sites

- KSC LC –39A has most successful launch count compare to others
- Least is CCAFS SLC-40



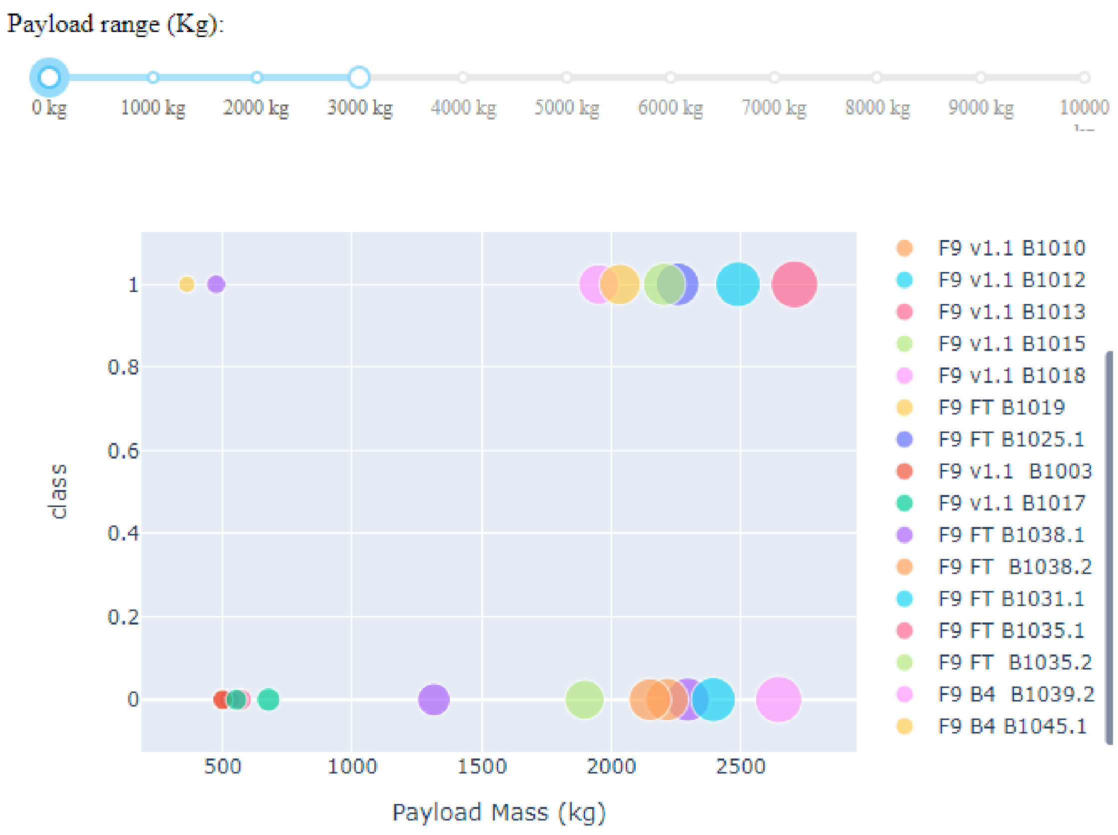
Launch Site with highest Success Rate

- Launch Site – KSC LC – 39A
- Has the highest success rate of 76.9%



Payload vs. Launch Outcome scatter plot for all sites

- Booster with high success at low payload mass(<3000) is F9 FT B1035.1
- Booster with high success at high payload mass(>3000) is F9 B4 B1041.1

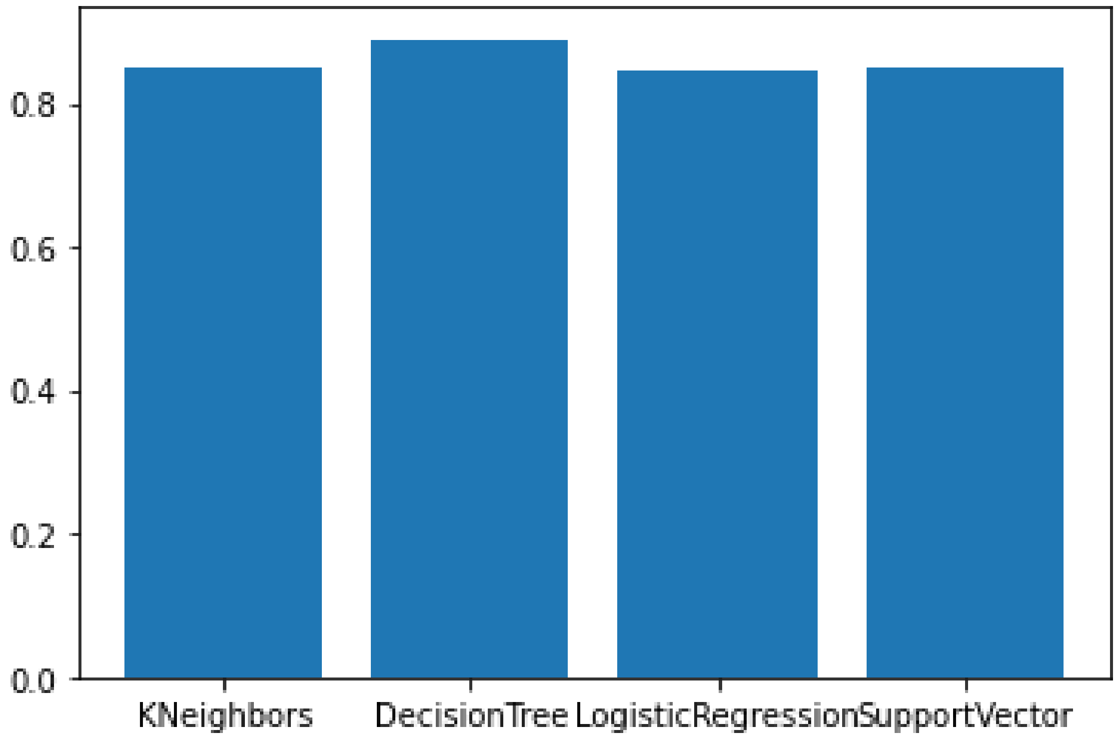


Section 5

Predictive Analysis (Classification)

Classification Accuracy

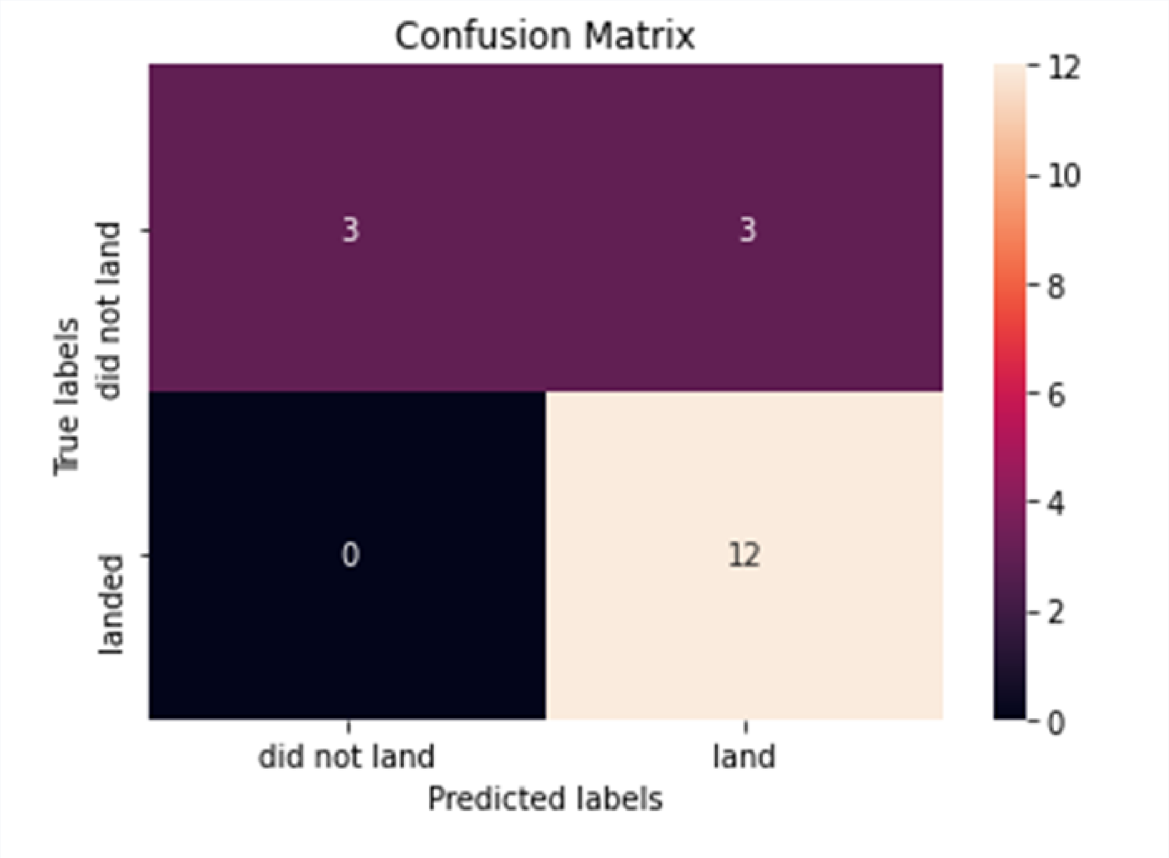
- Bar chart representation of different ML models with respective to scores
- Decision Tree model has the highest accuracy (around 87%)




```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- Confusion Matrix of Decision trees
- Here, 3 predicted outputs are shown wrong that is 'did not land ' is predicted as 'landed'.
- Remaining values are predicted accurately.
- This clearly shows the accuracy of 87%



Conclusions

- Launch success rate started to increase in 2013 till 2020.
- Gradual Increment in Flight Number, increases the chances of successful landing.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches comparatively.
- The Decision tree classifier is the best machine learning algorithm in this case

Appendix

- I have used Google colab, when there is a downtime in IBM Cloud. (Above ML bar chart is calculated here)

Related to this course, Here is the link -
https://colab.research.google.com/drive/1LAwULfQzXIx3vA-FdAEuSLuqy8KPId80?usp=sharing

Thank you!