IBM DATA SCIENCE PROJECT

SRAVAN

30/01/2022

Outline

- Executive Summary
- Methodology
- Results
- Conclusion

Executive Summary

SpaceX is the leading company in the US for rocket launches. It helps various companies and the government to send payload satellites into the space.

The reason for its position as the premier company in rocket launches is its capability of reusing the rocket for multiple launches (Falcon9).

This reduces the cost of launching rockets drastically.

The following project, using data science methodologies, helps in gaining valuable insights about the SpaceX rocket launches.

We try to find out how successful the landings have been and what the success rate was with varying payloads

Introduction

- The following is the order of steps used in this project.

Data Collection

Exploratory Data Analysis

Interactive Data Visualization

Predictive Analysis

METHODOLOGY

Data Collection

- There are two ways to collect the required data from SpaceX, they are: 1) data collection via API's

  2) data collection via web scrapping

- **Data collection via API's:** Data is collected from the SpaceX API and it is normalized into a Data Frame using Pandas.
- **Data collection via web scrapping**: Data is scrapped from the SpaceX Wikipedia page (Falcon rocket launches) and the required features are converted into a Data Frame using pandas pd.DataFrame() function.
- The Data Frames obtained can be converted into csv file using DataFrame.to_csv() function.

Data Wrangling

In the Data Frame check for null values

Create a new column in the data frame called 'class'

Class 1 means 'successful landing' and class 0 means 'unsuccessful landing'

```
df.isnull().sum()/df.count()*100
```

FlightNumber	0.000
Date	0.000
BoosterVersion	0.000
PayloadMass	0.000
Orbit	0.000
LaunchSite	0.000
Outcome	0.000
Flights	0.000
GridFins	0.000
Reused	0.000
Legs	0.000
LandingPad	40.625
Block	0.000
ReusedCount	0.000
Serial	0.000
Longitude	0.000
Latitude	0.000

dtype: float64

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Exploratory Data Analysis

- **EDA**: exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.
- **EDA with SQL**: using SQL we try to answer the following questions for the SpaceX 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

EDA Using Pandas and Matplotlib

- In this step, we perform the following data analysis on the SpaceX Data.

Visualize the relationship between the 'Flight Number' and 'Payload Mass'

Visualize the relationship between 'Flight Number' and 'Launch Site'

Visualize the relationship between 'Payload' and 'Launch Site'

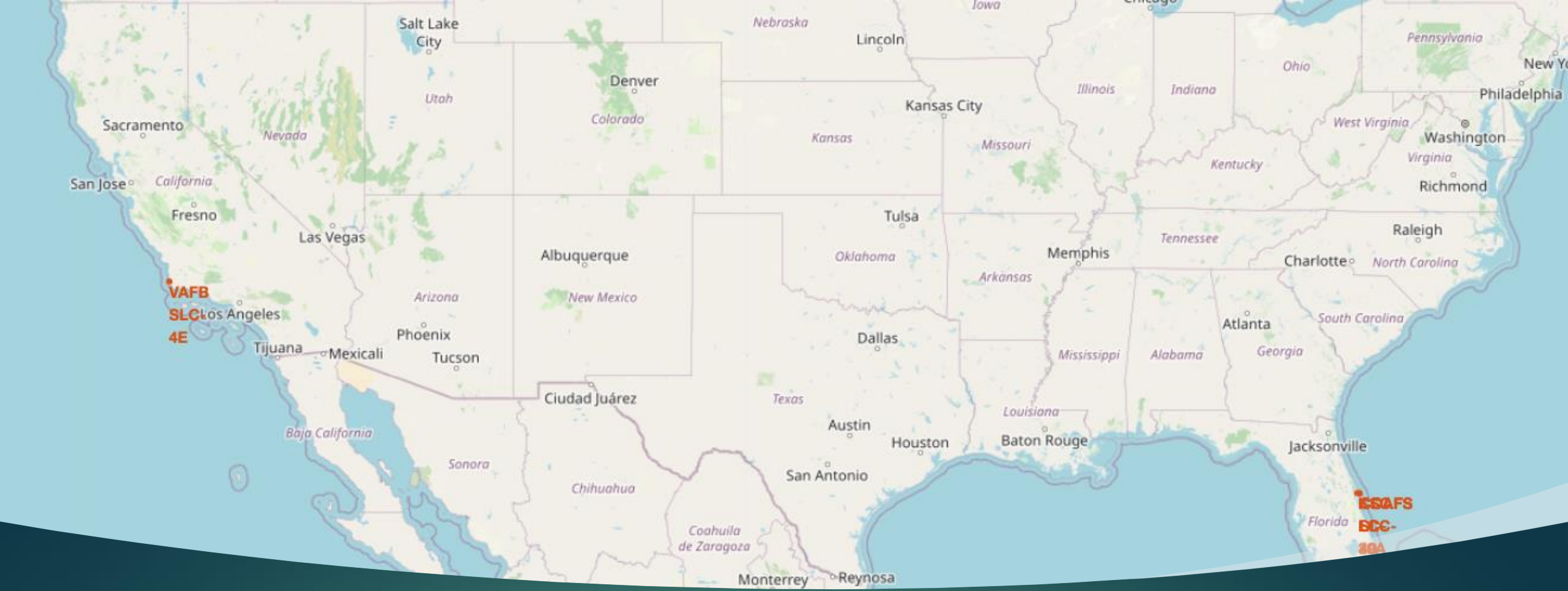
Visualize the relationship between 'success rate' of each 'orbit type'

Visualize the relationship between FlightNumber and Orbit type

Visualize the launch success yearly trend

Feature Engineering: Create dummy variables to categorical columns

Example: The following graph visualizes the launch success yearly trend



Launch Sites Locations Analysis with Folium

IN THIS STEP WE MAP THE LAUNCH SITE LOCATIONS ON A MAP USING FOLIUM AND ANALYSE DISTANCES TO PROXIMITIES(RAILWAYS, HIGHWAYS, COASTLINES).

Interactive Visual Analytics with Plotly Dash

- The following tasks are done to visualize which site has largest successful launches, highest launch success rate, payload ranges with high success rate.

1

TASK 1: Add a Launch Site Drop-down Input Component

2

TASK 2: Add a callback function to render `success-pie-chart` based on selected site dropdown

3

TASK 3: Add a Range Slider to Select Payload

4

TASK 4: Add a callback function to render the `success-payload-scatter-chart` scatter plot

Predictive Analysis Methodology

From the dataset, the data is divided into X(features) and y(labels).

The features in the scaled down using the StandardScaler() object

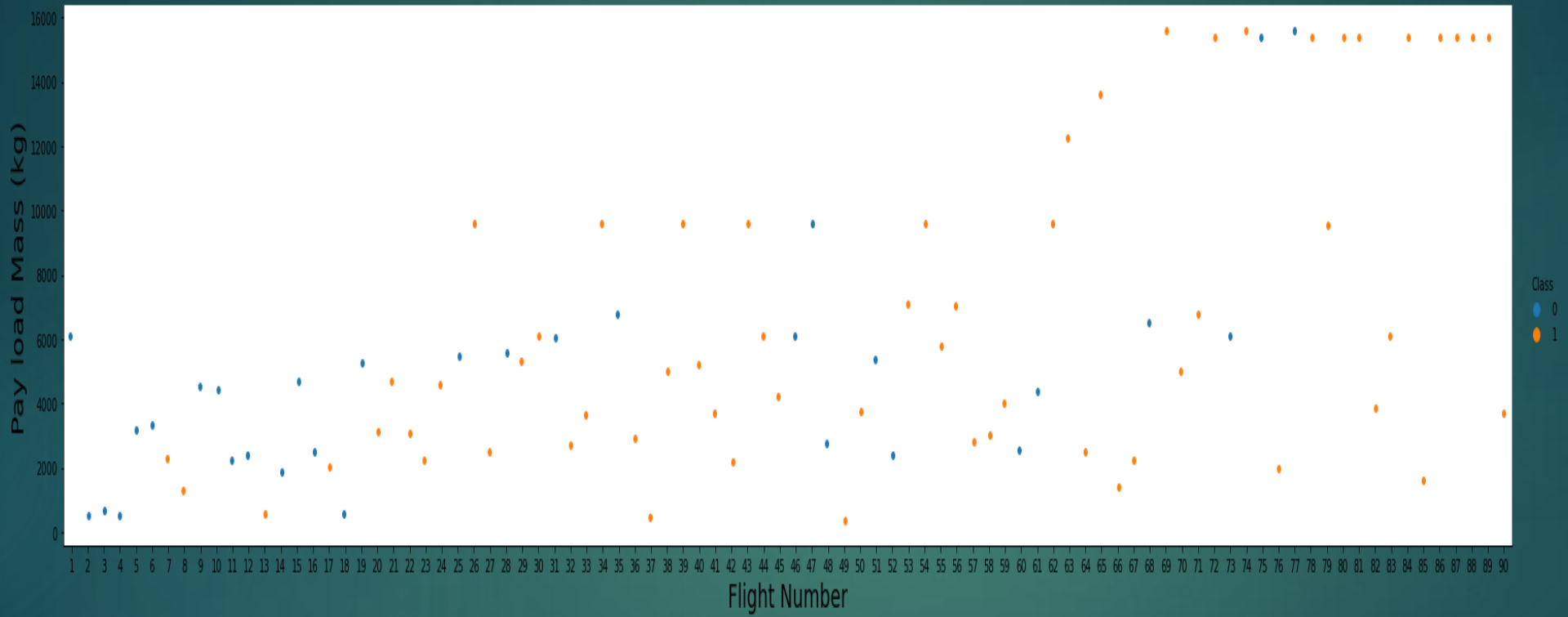
The X and y data is now split into training and testing datasets (X_train, Y_train, X_test, Y_test)

X_train and Y_train is used for training using different methods(logistic, SVM, decision tree, KNN)

We check for the predictions in each method and select the method which produces highest accuracy.

RESULTS

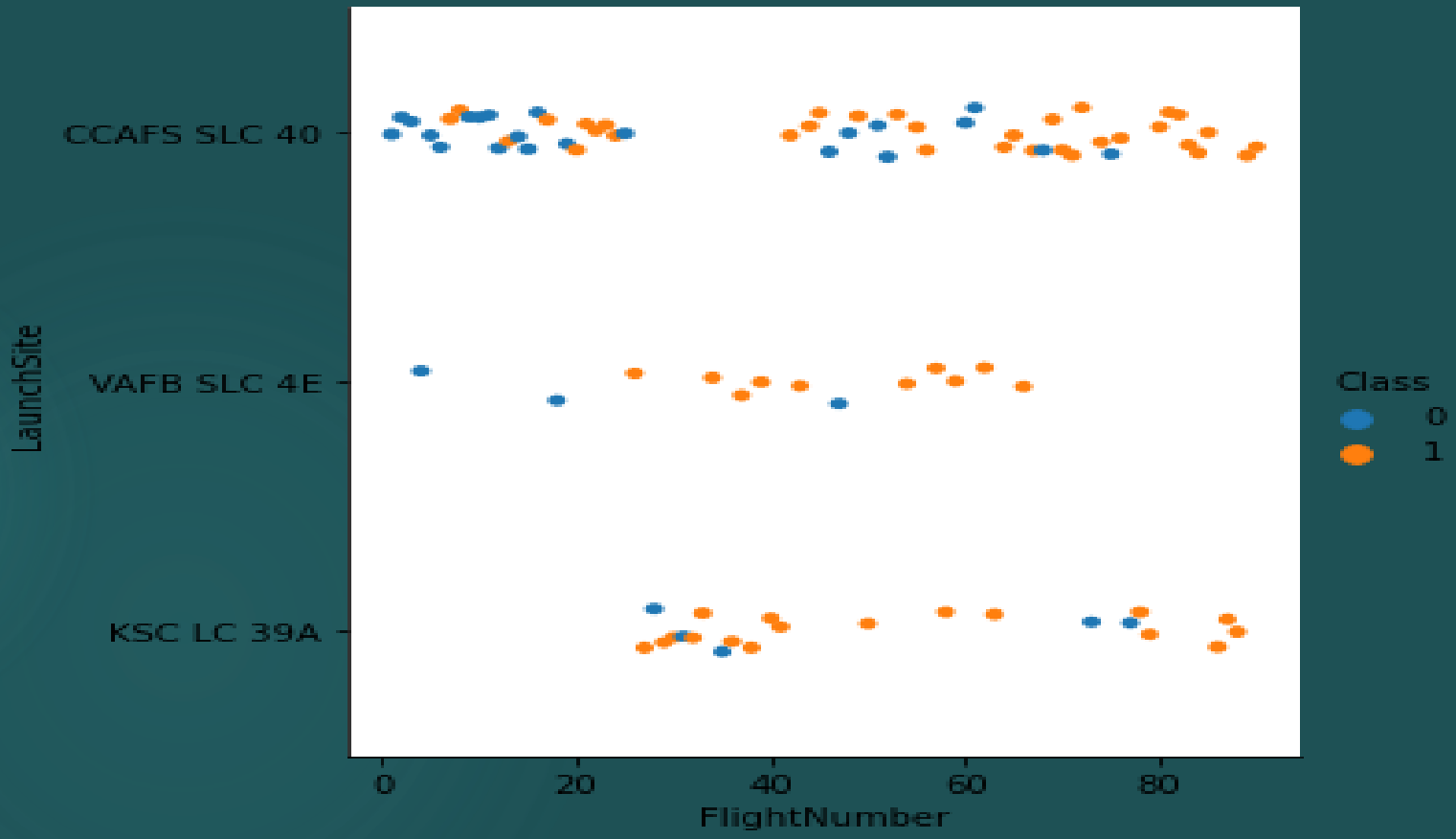
EDA with visualization results



Flight number vs payload mass

As flight number increases we there is an increase in success rate

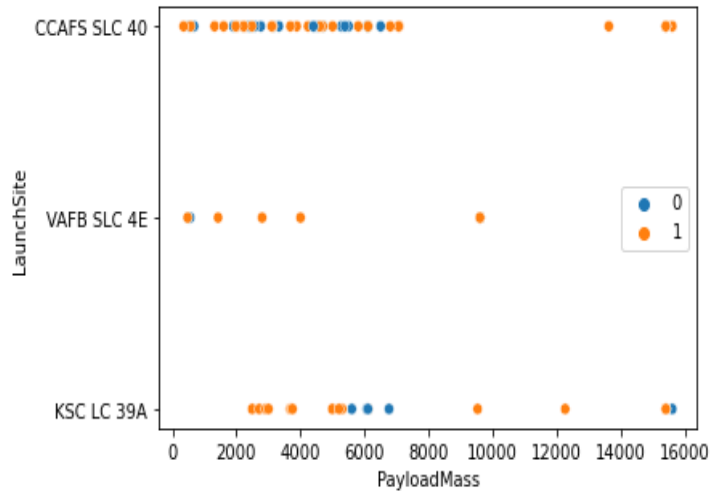
Flight number vs Launch Site



As the flight number increases CCAFS SLC 40 is used more and we move to higher than 70 flights, the success rate also increases.

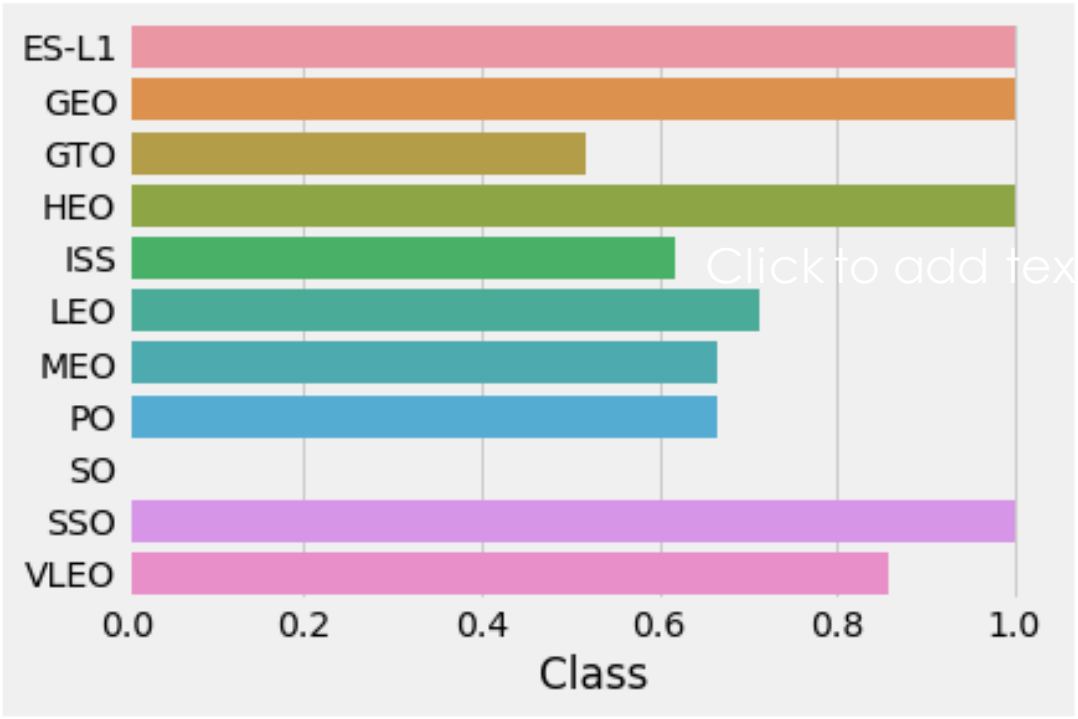
Payload vs Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class valu
e
sns.scatterplot(x='PayloadMass',y='LaunchSite',hue='Class',data=df)
plt.legend(loc='right')
plt.show()
```



Higher payloads have high success rate and launch site VAFB doesn't have payloads greater than 10000.

Visualize the relationship between success rate of each orbit type

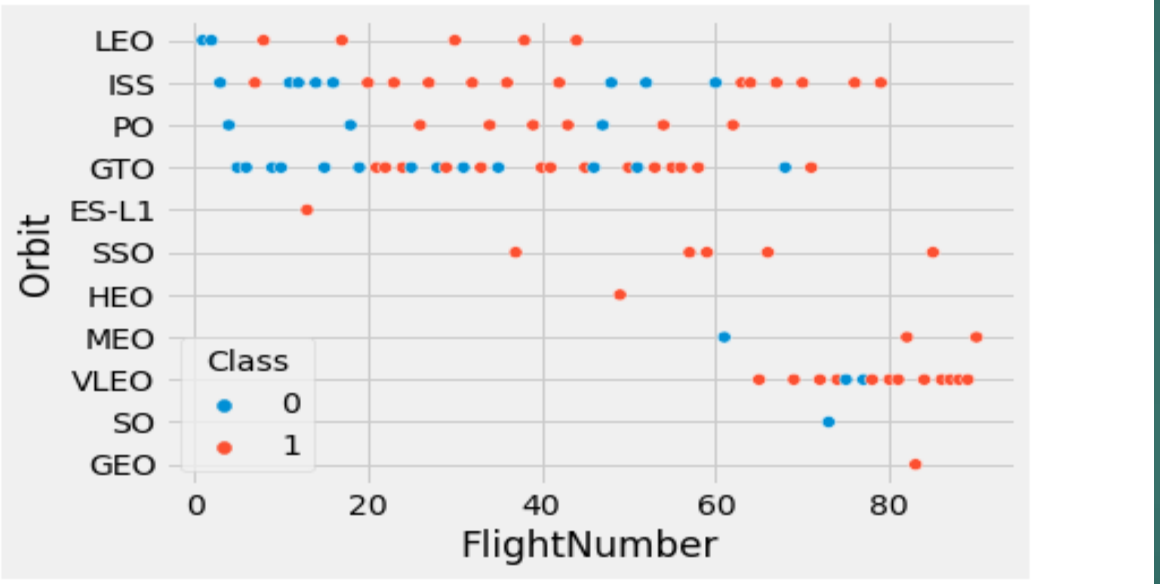


Orbits ES-L1, GEO, HEO, SSO have 100 percent success rate

Flight Number vs Orbit type

```
In [62]: # Plot a scatter point chart with x axis to be FlightNumber and y
         sns.scatterplot(x='FlightNumber', y='Orbit', hue='Class', data=df)
```

```
Out[62]: <AxesSubplot:xlabel='FlightNumber', ylabel='Orbit'>
```

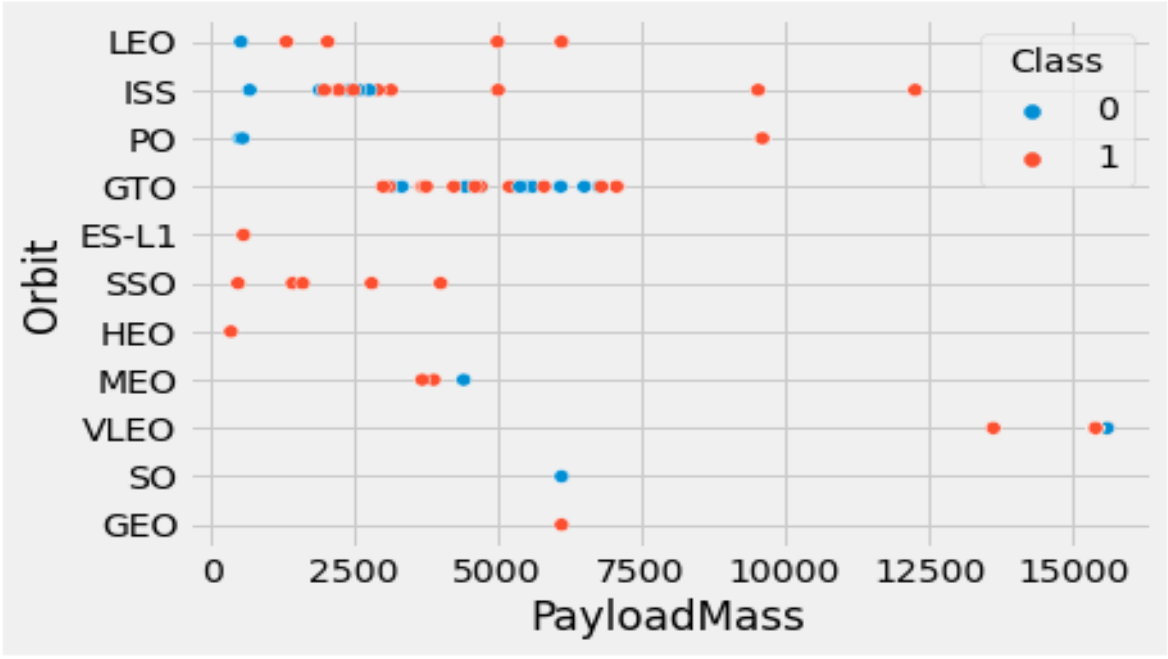


You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs Orbit type

```
# Plot a scatter point chart with x axis to be Payload and y axis to
sns.scatterplot(x='PayloadMass',y='Orbit', hue='Class', data=df)
```

```
<AxesSubplot:xlabel='PayloadMass', ylabel='Orbit'>
```



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

Launch success yearly trend

Generally the success rate of the rocket launches went up year except for the dip in 2018

EDA with SQL results

Task 1

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

In [16]:
```
%sql select DISTINCT Launch_Site from SPACEX;
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[16]:

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

Task 2

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

```
%sql select Launch_Site from SPACEX \
    where Launch_Site like 'CCA%'\
    limit 5;
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

2]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass_CRS\
      from SPACEX\
      where Customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

]:

total_payload_mass_crs
45596

Task 4

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

In [14]:
```
%sql select avg(PAYLOAD_MASS__KG_) as avg_payload_mass\
     from SPACEX\
     where Booster_Version= 'F9 v1.1'
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

Out[14]:

avg_payload_mass
2928

Task 5

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql select min(DATE) as first from SPACEX where LANDING__OUTCOME= 'Success (ground pad)';
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

]:

first
2015-12-22

Task 6

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 BOOSTER_VERSION from SPACEX where Landing__Outcome= 'Success (drone ship)' and  PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ <6000;
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

]:

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

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(Mission_Outcome) as Sucess from SPACEX  where Mission_Outcome = 'Success'
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

7]:

sucess
99

```
%sql select count(Mission_Outcome) as failure from SPACEX  where Mission_Outcome = 'Failure (in flight)'
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

3]:

failure
1

```
%sql select count(Mission_Outcome) as payload_status_unclear from SPACEX  where Mission_Outcome = 'Success (payload status unclear)'
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

9]:

payload_status_unclear
1

Task 8

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

```
%sql select Booster_Version from SPACEX where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX)
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

42]:

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

Task 9

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

```
%sql select LANDING__OUTCOME, Booster_Version, Launch_Site from SPACEX\
    where Landing__Outcome = 'Failure (drone ship)' and DATE like '2015%'
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

1]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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 as count from SPACEX \
    where DATE between '2010-06-04' and '2017-03-20'\
    group by Landing__Outcome\
    order by count DESC;
```

```
 * ibm_db_sa://zxz83611:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/bludb
Done.
```

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

Interactive map with Folium results slides

A data frame of the launch site and their respective latitudes and longitudes is created

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

Using the corresponding latitudes and longitudes a folium map is created which shows the Launch Sites

Folium Map with Launch Sites

Except for VAFB SLC-4E, the remaining three Launch sites are in close proximity in Florida

Calculate distance between the Launch site and its proximities(ex: CCAFS_LC_40)

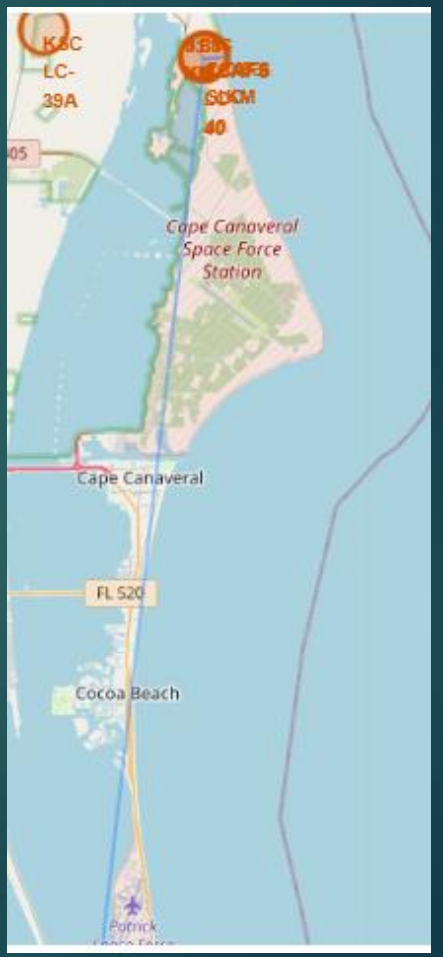
```
# Create a marker with distance to a closest city, railway, highway, etc.
# Draw a line between the marker to the launch site
coordinates = {
    'city' : [28.10537, -80.63674],
    'rail' : [28.57201, -80.58524],
    'high' : [28.56227, -80.57061],

}

for key in list(coordinates.keys()):
    coordinate = coordinates.get(key)
    dist = calculate_distance(launch_site_lat, launch_site_long, coordinate[0], coordinate[1])
    print(dist)
    marker = folium.Marker([28.57201, -80.58524], icon=DivIcon(icon_size=(20,20), icon_anchor=(0,0),
                                                               html='<div style="font-size: 12; color:#d35400;"><b>%s</b></d
iv>' % "{:10.2f} KM".format(dist),))
    lines = folium.PolyLine(locations=(coordinate, [launch_site_lat, launch_site_long]), weight=1)

    site_map.add_child(marker)
    site_map.add_child(lines)
```

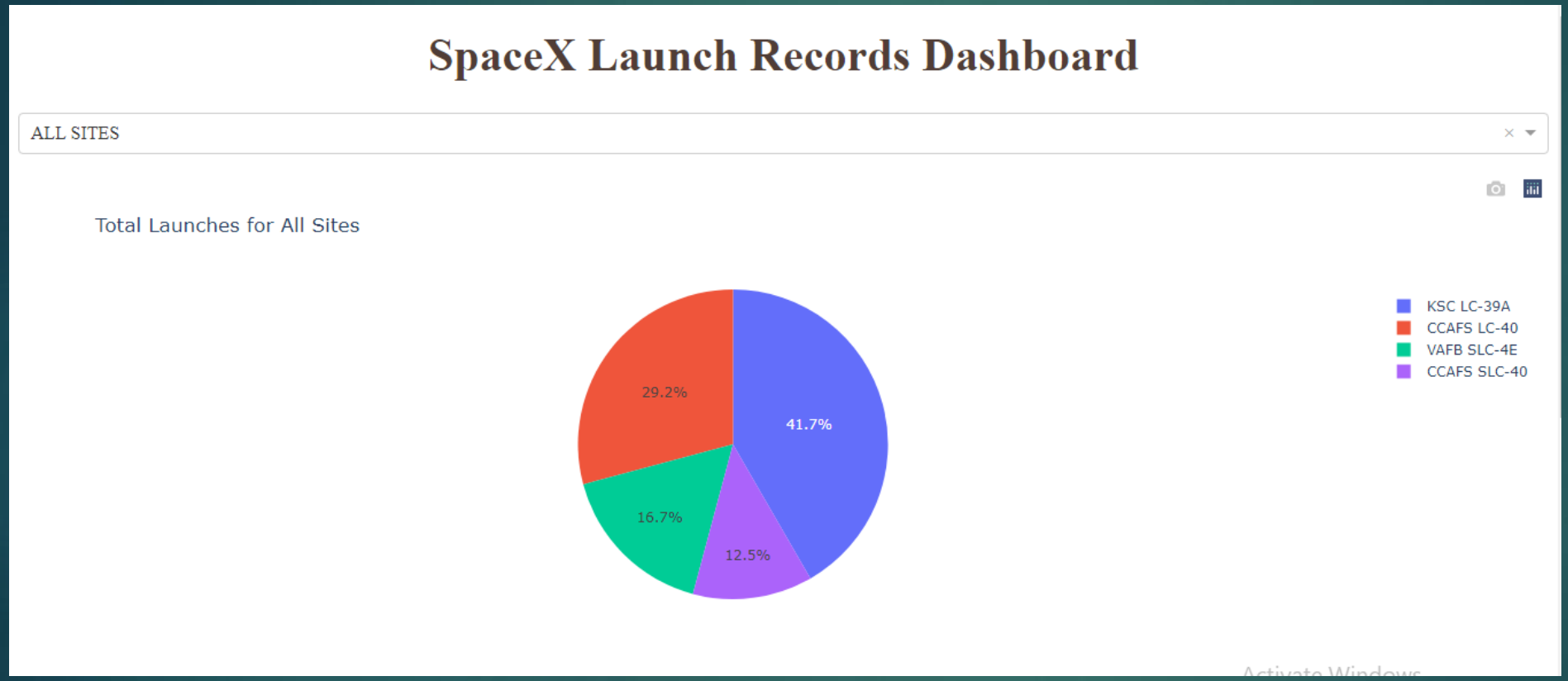
```
51.15591319665129
1.326316662331983
0.6590924814143757
```



All launch sites are close to highways and railways but bit far from the City

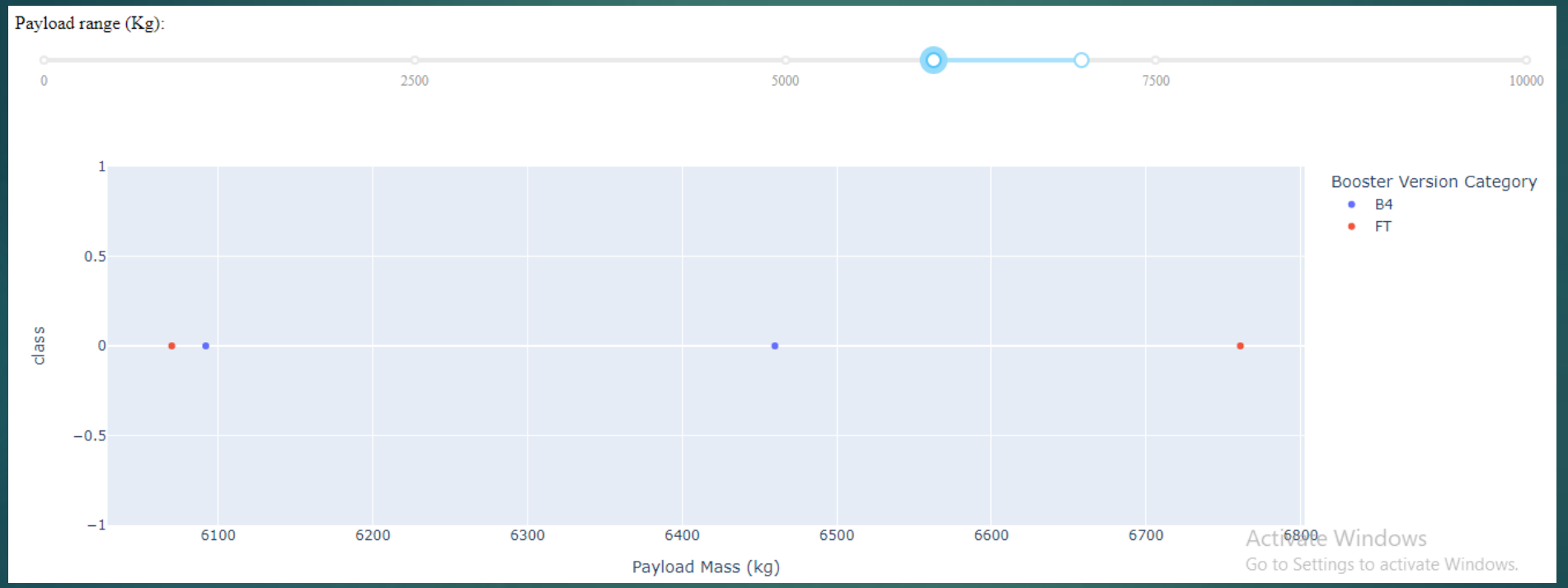
PLOTLY DASH DASHBOARD RESULTS SLIDES

Launch success for all sites



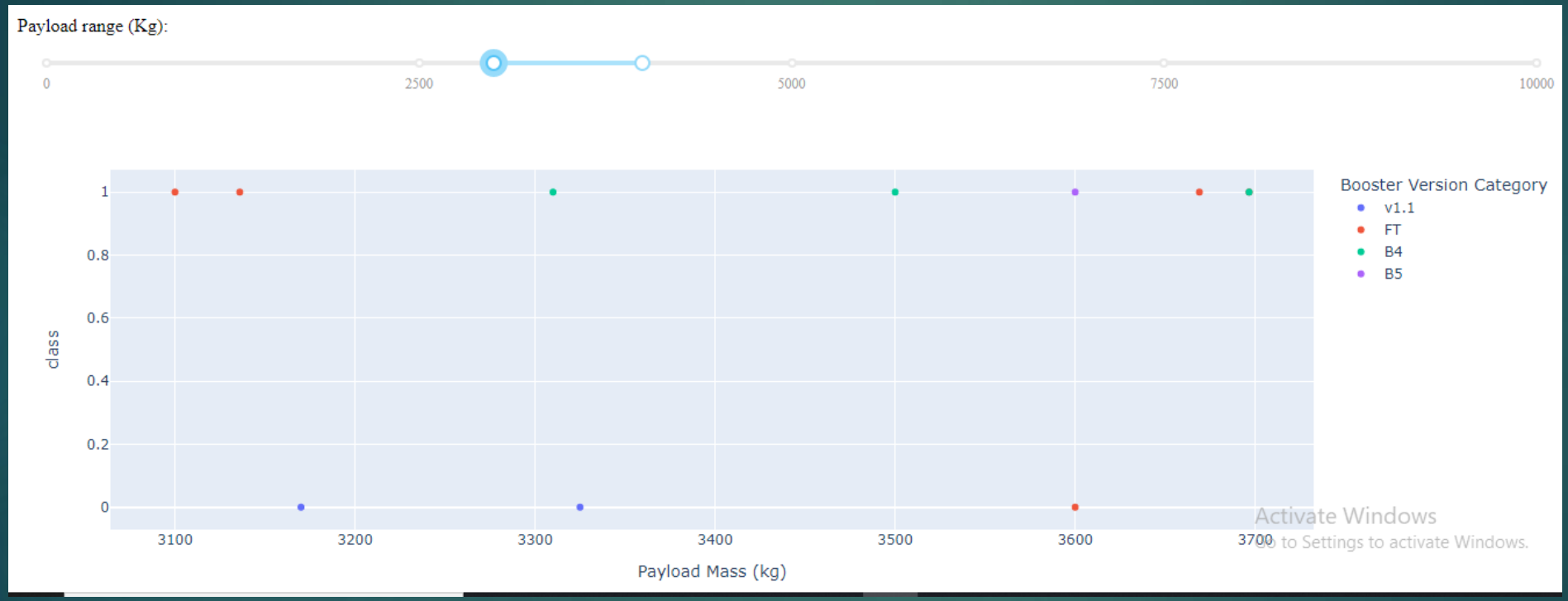
KSC LC 39A has the largest successful launches

Payload vs Launch outcome(lowest launch Success rate)



Range between 6000 and 6800 has the lowest success rate

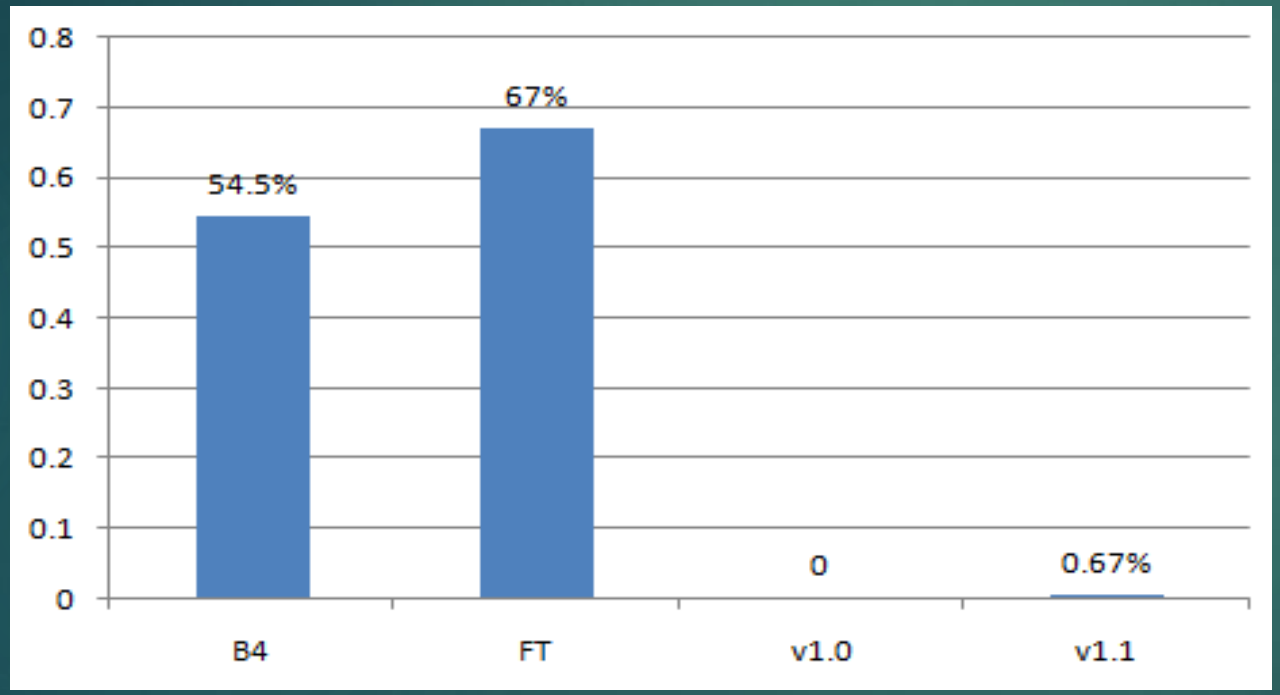
Payload vs Launch outcome(highest launch Success rate)



Range between 3000 and 4000 has the highest success rate

Booster version highest success rate

- Since B5 version has only one launch we are going to ignore it.



Booster version FT has highest success rate at 67%

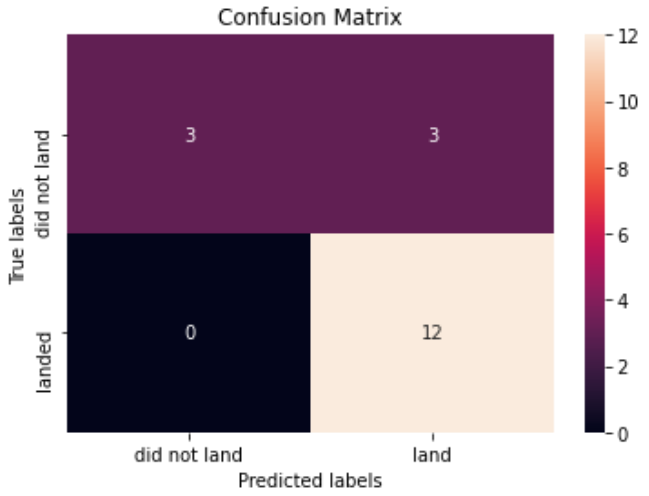
Predictive analysis

Calculating Accuracy :

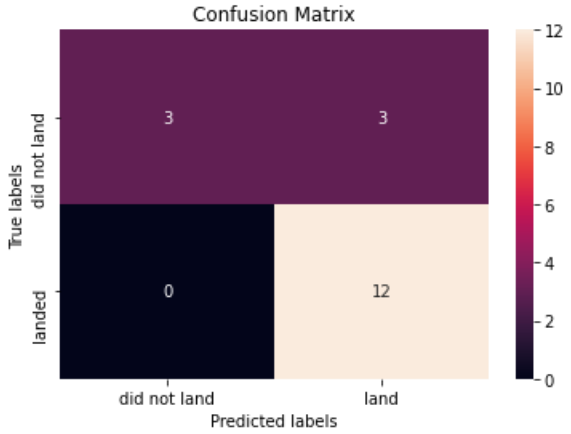
- Logistic: (TP+TN)/Total=(12+3)/18=0.833
- SVM: 15/18=0.833
- DecisionTree:(11+4)/18=0.833
- KNN: (12+3)/18=0.833

Confusion Matrix for all 4 methods:

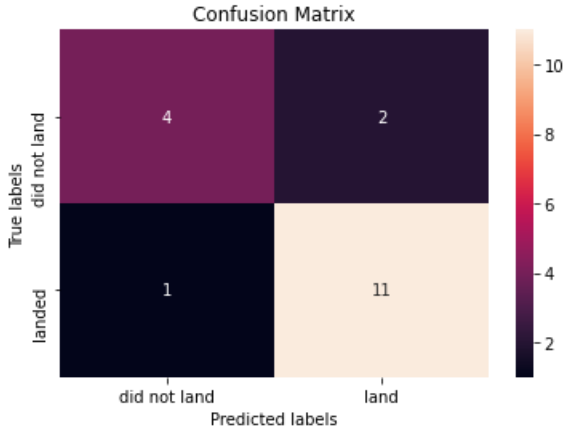
Logistic Regression



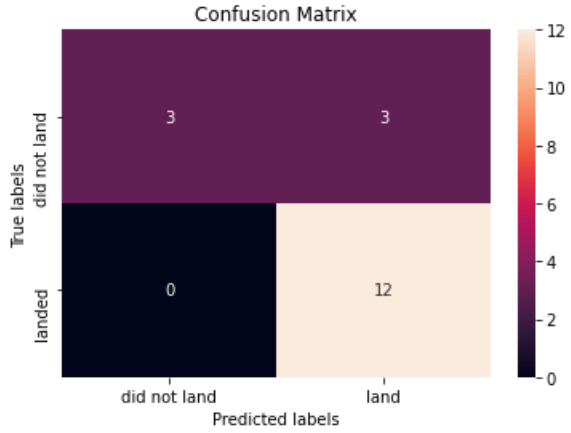
SVM



Decision_Tree



KNN



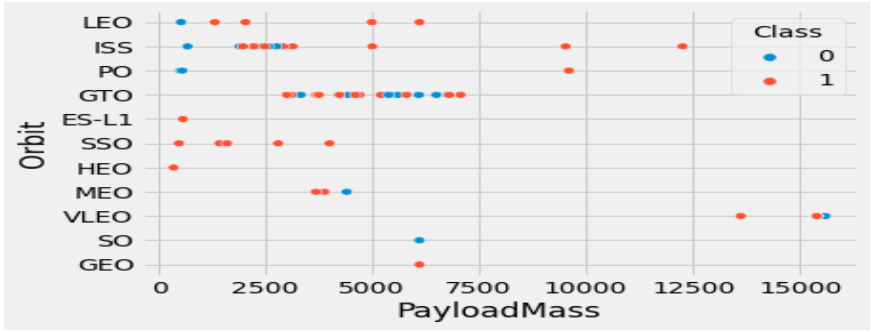
Comparing Accuracies to find best Model

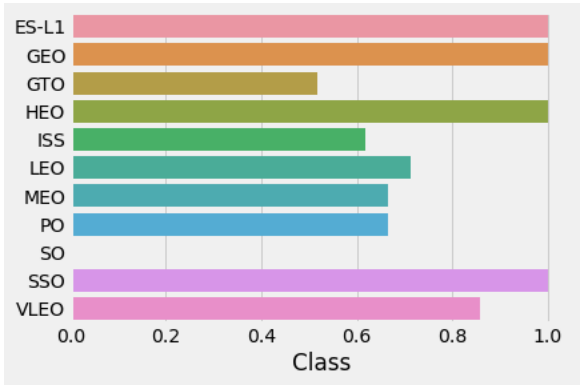
- Based on the best score and Accuracy, Decision Tree method seems to be the best machine learning at predicting the outcome for successful landing.

Method	Best_Score	Accuracy	Tuned Hyper parameters
Logistic Regression	0.846	0.833	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM	0.848	0.833	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
Decision Tree	0.875	0.833	{'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
KNN	0.848	0.833	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

Conclusion

- Success rate is high for payload mass greater than 7500
- Orbits ES-L1,GEO,HEO,SSO have highest success rate
- The rate of successful landing each year is going up with a slight dip in 2018
- KSC LC 39-A site has the most successful launches(67%)
- Decision Tree Classifier is the best machine learning model for predicting the successful landings.





Method	Best_Score	Accuracy
Logistic Regression	0.846	0.833
SVM	0.848	0.833
Decision Tree	0.875	0.833
KNN	0.848	0.833