IBM Applied Data Science Capstone Project

Mehrasa Modanlou Jouybari

April 10, 2023

https://github.com/MehrasaModanlou/Applied-Data-Science-Capstone

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion

EXECUTIVE SUMMARY

- Applied Data Science Capstone Project is a course I took to complete IBM professional certification in Data Science on Coursera.
- This capstone project enabled me to boost the skills I have learned in the previous courses to extract and clean real-world datasets.
  - Extracting: SQL, Web Scraping
  - Preprocessing: Fill or Remove none values, Feature Engineering, standardization
- It’s very interesting to explore and analyze data using visualization tools and machine-learning Python libraries to get insights from them.
  - Numpy, Pandas, Scikit-Learn, Matplotlib and Seaborn
- I have completed the certification today which I started the first month of the year and I’m thrilled to share a presentation about the capstone.
- In this capstone we will be analyzing the data ,from wiki extracted through web scrapping and spacex api to get insights and predict booster landing to drone ship safely.

INTRODUCTION

- SpaceX is a aerospace manufacturer, space transportation services and communications corporation which is a disruptive just like Tesla both founded by Elon Musk. Despite being less than 20 years old SpaceX has managed to reduce the cost by more than 50% compared to other company and said to reduce by 99% when the Starship project will be completed.
- This is because spaceX has developed Technology to land the first stage booster which is 70% the cost of the rocket. By landing it safely they are able to reuse the booster and the cost of the launches. Using their reused boosters cost 50% less of their their cost to use a new booster and this has made spaceX the company of that dominate the market. SpaceX has gained worldwide attention for a series of historic milestones.
- It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars wheras other providers cost upward of 165 million dollars each, much of the savings is because Space X 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.

METHODOLOGY

- **Data Collection**
  - Data collection for Falcon 9 launch records through Web scraping using python BeautifulSoup package is available on my github here
  - SpaceX api data collection about launches is available on my github here
- **Data Wrangling:**
- After collecting the data we check the missing data and data types click here, and do one of the following to clean the data:
  - *Deal with Null values: Replace the missing data with one- Using mean, median or so.*
  - *Change data type of the data.*
  - *Represent categorical data using integer or float dummy numbers – one hot encoding*
  - *split data into train and test set*

METHODOLOGY

- **Exploratory Data Analysis(EDA):**
- After data cleaning we can proceed to Analyzing the data using visualization to get some insights of the launches. EDA is the first step of any data science project.
  - SQL: click here to see notebook
  - Visualization here: using Pandas and Matplotlib

- **interactive visual analytics and dashboards**
  - explore and manipulate data in an interactive and real-time way.
  - find visual patterns faster and more effectively.
  - more appealing story
    - Folium: click here
    - Plotly Dash: click here

METHODOLOGY

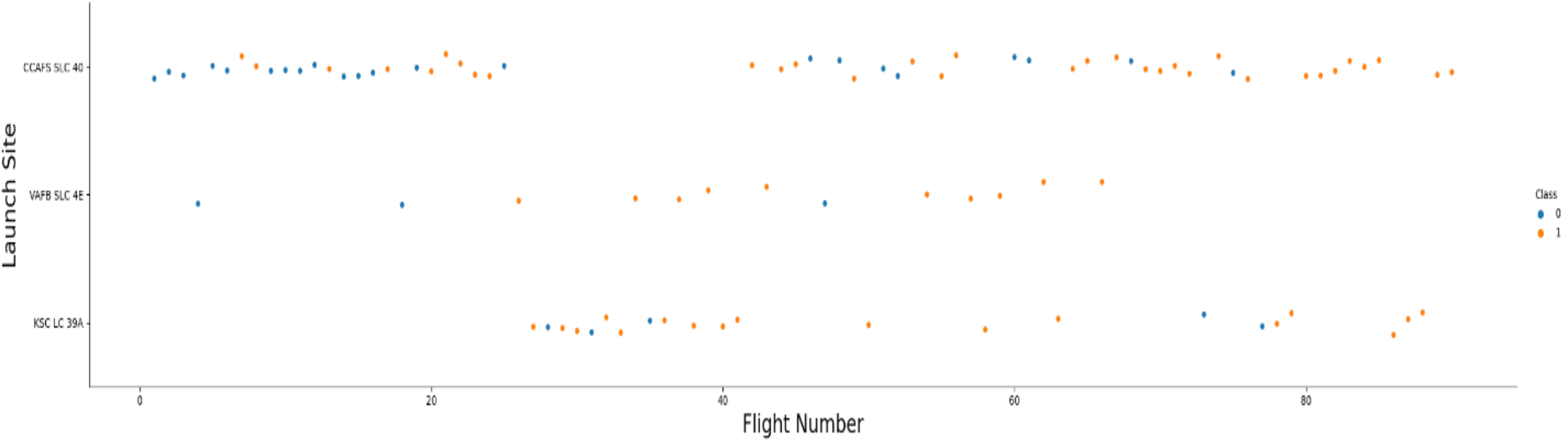
- **Predictive Analysis using Machine Learning**

  Some data attributes can be used to determine if the first stage can be reused. We can use these features with machine learning algorithms to automatically predict if the first stage can land successfully or not. here

  - Logistic Regression
  - Decision Tree Classifier
  - Support Vector Machine
  - K-Neighbors Classifier

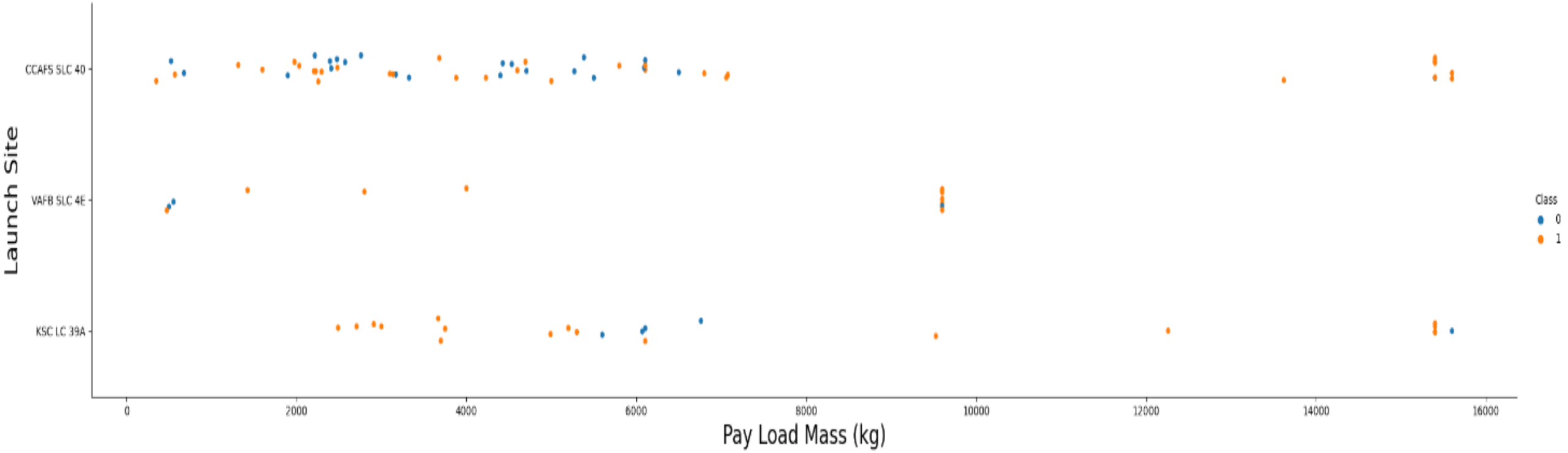
RESULTS

- EDA with visualization results
  - **Launch Site Vs. Flight Number**



RESULTS

- Launch Site Vs. Pay Load Mass(Kg)



RESULTS

- EDA with visualization results

From the relation between Launch Site with Flight Number and Pay Load Mass we can conclude that:

Launch Site Vs. Flight Number

i. Earlier flights launch were from CCAFS-SLC-40 site, Followed by KSC-LC-39A.

ii. Most Launches are Launched from CCAFS-SLC-40.

iii. Fewer Launches from VAFB SLC 4E site.

Launch Site Vs. Pay Load Mass(Kg)

i. VAFB SLC 4E has Low Payload launches.

ii. CCAFS SLC 40 has more Higher Payload Launches and Low Payload Lauches.

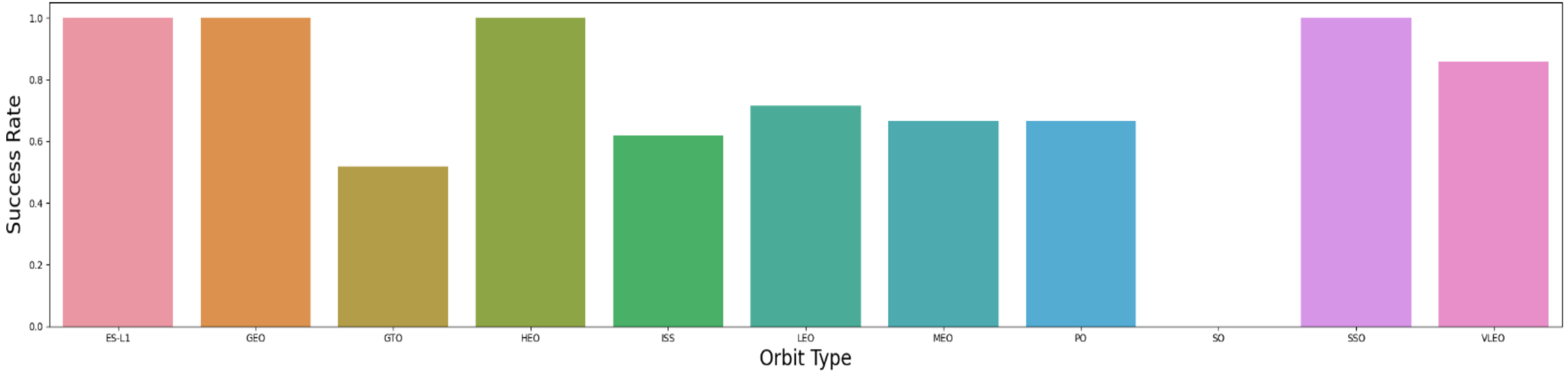
iii. for the VAFB-SLC 4E there are no rockets launched for heavypayload mass(greater than 10000 Kg).

iv. for the KSC LC 39A there are no rockets launched for light payload mass(less than 2000 Kg).

RESULTS

- EDA with visualization results

  From the bar chart visualization, we can conclude that:
  - ES-L1, GEO, HEO, & SSO have high success rate.
  - SO success rate is almost zero.

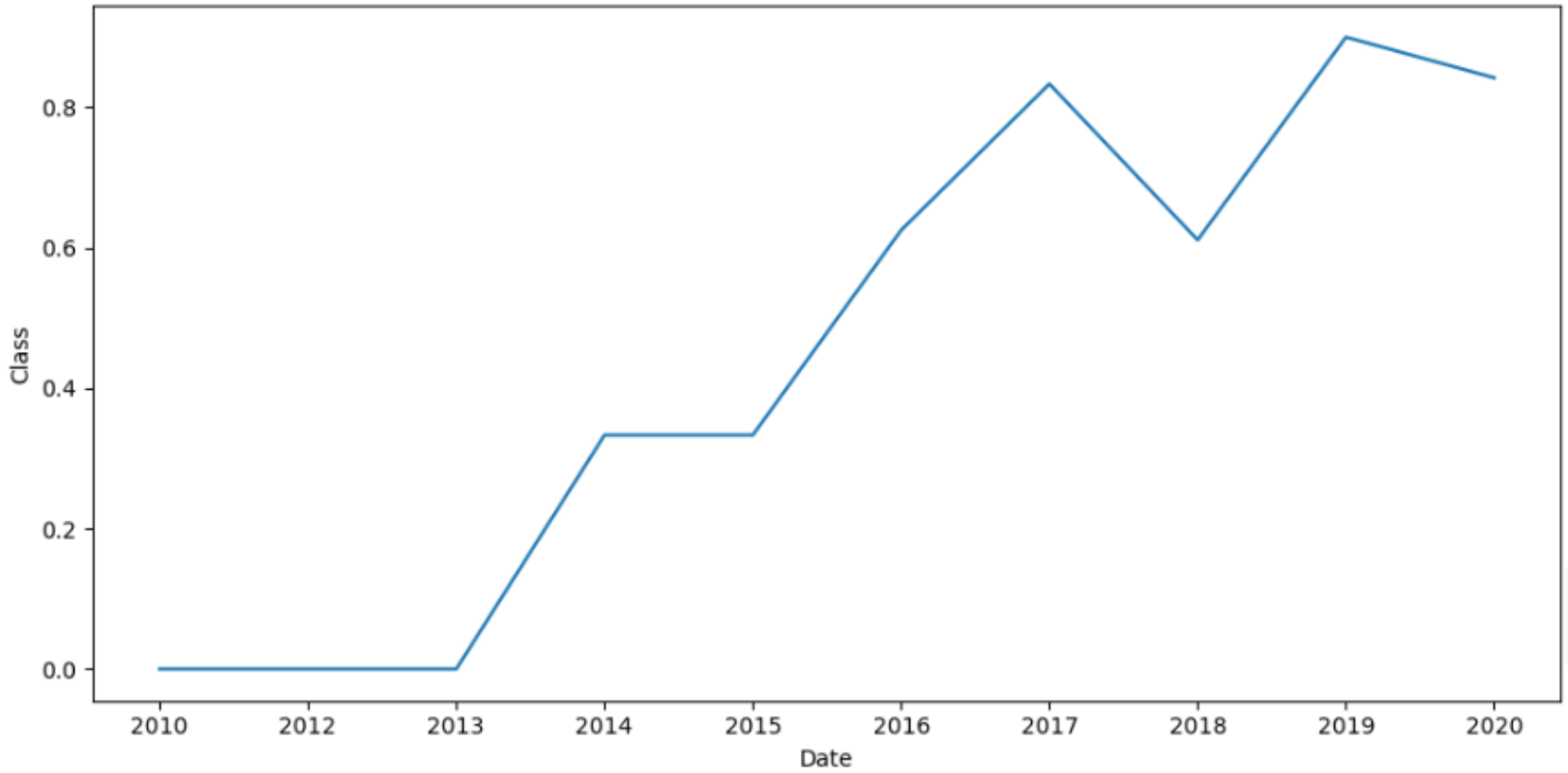


RESULTS

- EDA with visualization results

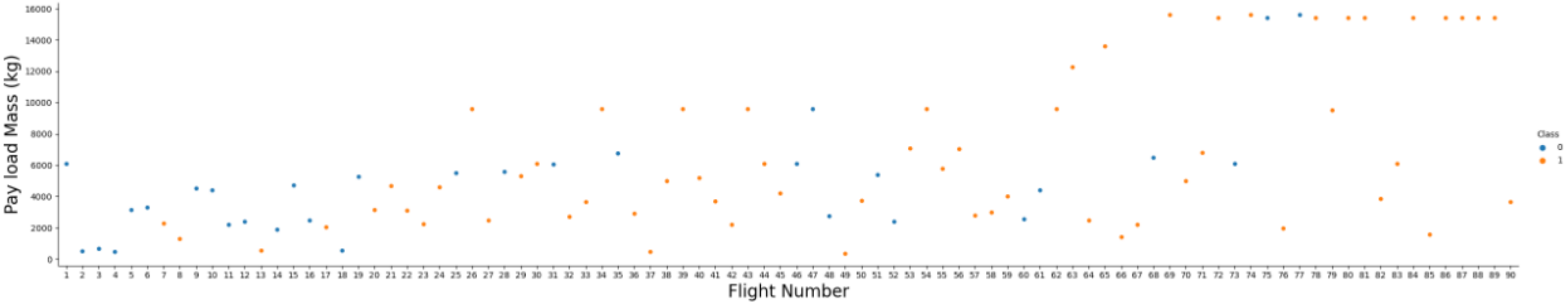
  From the line plot visualization, we can conclude that:

  - the sucess rate since 2013 kept increasing till 2020



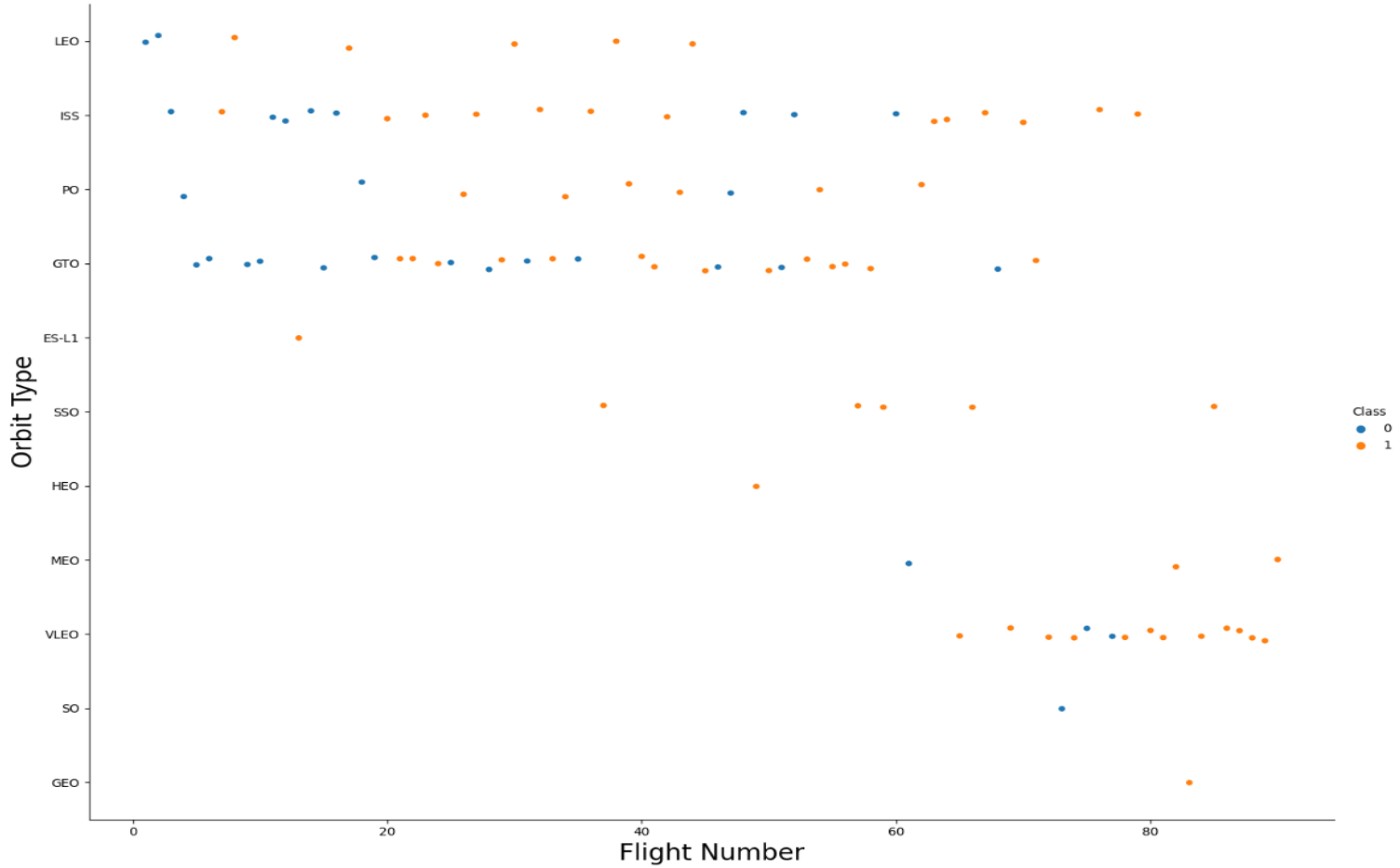
RESULTS

- Flight Number Vs. Pay Load Mass(Kg)



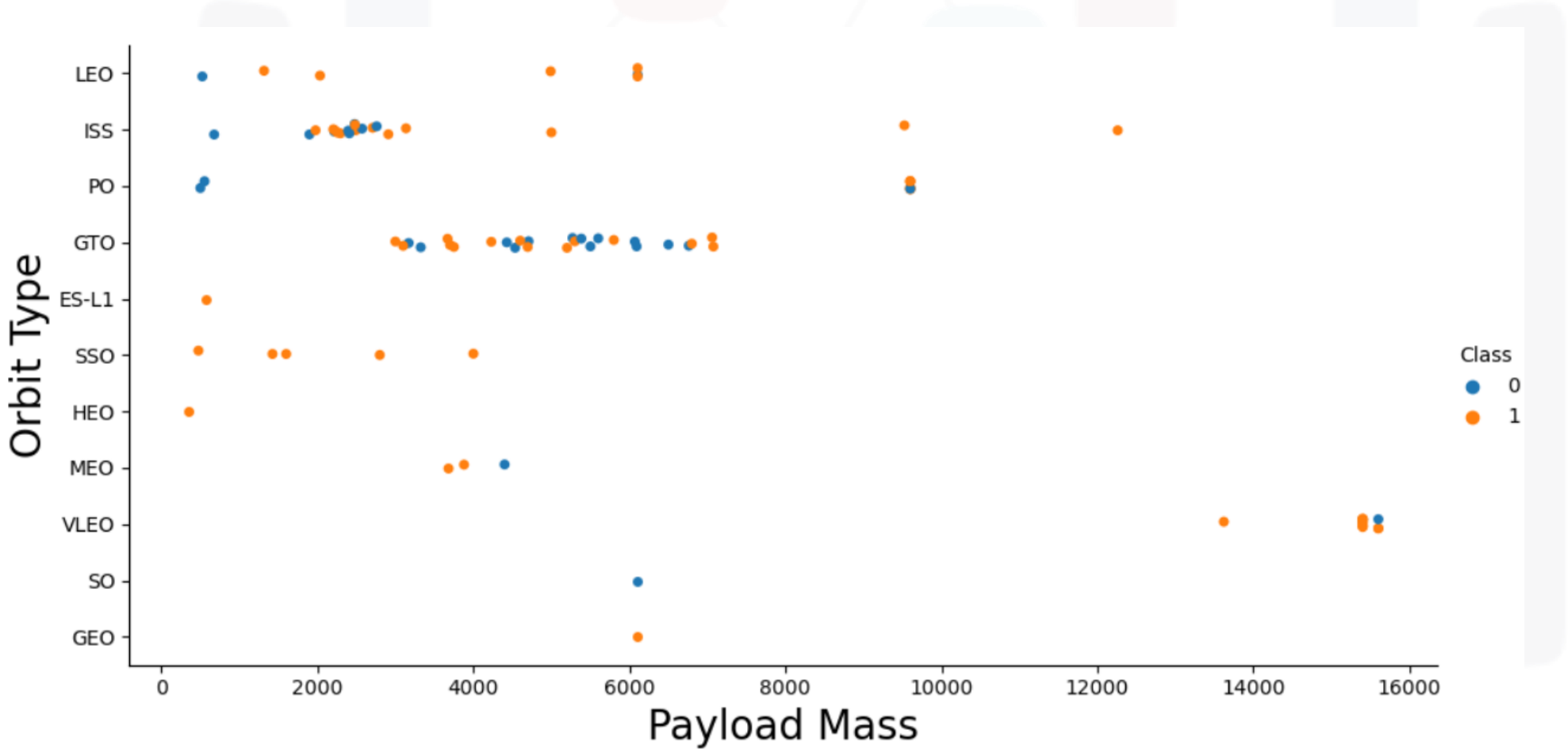
RESULTS

- **Flight Number Vs. Orbit Type**



RESULTS

- Orbit Type Vs. Pay Load Mass(Kg)



RESULTS

- **Create dummy Variables**

```
# HINT: use astype function
features_one_hot = features_one_hot.astype('float64')
features_one_hot.dtypes
```

```
FlightNumber    float64
PayloadMass     float64
Flights         float64
GridFins        float64
Reused          float64
                 ...
Serial_B1056    float64
Serial_B1058    float64
Serial_B1059    float64
Serial_B1060    float64
Serial_B1062    float64
Length: 80, dtype: object
```

RESULTS

- **EDA with SQL results**
- **Display the names of the unique launch sites in the space mission**

From the SQL visualization, we can conclude that:

- Failure in outcomes for the months in year 2015 is regarding CCAFS LC-40 launch site with “F9 v1.1 B1012” and “F9 v1.1 B1015” booster versions.

```
%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

RESULTS

- **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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

RESULTS

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS 'TOTAL_PAYLOAD_MASS'
FROM SPACEXTBL
WHERE "Customer" = "NASA (CRS)"
```

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

TOTAL_PAYLOAD_MASS
45596

RESULTS

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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS 'AVG_PM_F9_v1.1'
FROM SPACEXTBL
WHERE "Booster_Version" = "F9 v1.1"
```

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

AVG_PM_F9_v1.1
2928.4

RESULTS

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

```
q = pd.read_sql("select min(Date) as First_Success_Date from spacexdata where Landing__Outcome='Success (ground pad)'", conn)
q
```

	First_Success_Date
0	2015-12-22 00:00:00

RESULTS

- 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 DISTINCT("Booster_Version"), "Landing _Outcome", PAYLOAD_MASS__KG_ FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

RESULTS

- List the total number of successful and failure mission outcomes

```
q = pd.read_sql("select substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from spacexdata  group by 1", conn)
q
```

	Mission_Outcome	count(*)
0	Failure	1
1	Success	100

RESULTS

- **Using a subquery, List the names of the booster_versions which have carried the maximum payload mass.**

```
%%sql
SELECT DISTINCT("Booster_Version"), PAYLOAD_MASS__KG_ AS MAX_PAYLOAD_MASS
FROM SPACEXTBL
WHERE MAX_PAYLOAD_MASS = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Booster_Version	MAX_PAYLOAD_MASS
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

RESULTS

- 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.

```
%%sql
SELECT substr(Date,7,4) AS Year,
substr(Date, 4, 2) AS Month,
"Date",
"Landing _Outcome",
"Booster_Version",
"Launch_Site"
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "%Failure (drone ship)%" AND Year='2015'
```

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

Year	Month	Date	Landing _Outcome	Booster_Version	Launch_Site
2015	01	10-01-2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	14-04-2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

RESULTS

- **Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.**

```
q = pd.read_sql("select Landing__Outcome, Date, count(*) as Count from spacexdata where Landing__Outcome like '%Success%' and Date between '2010-06-04
q
```

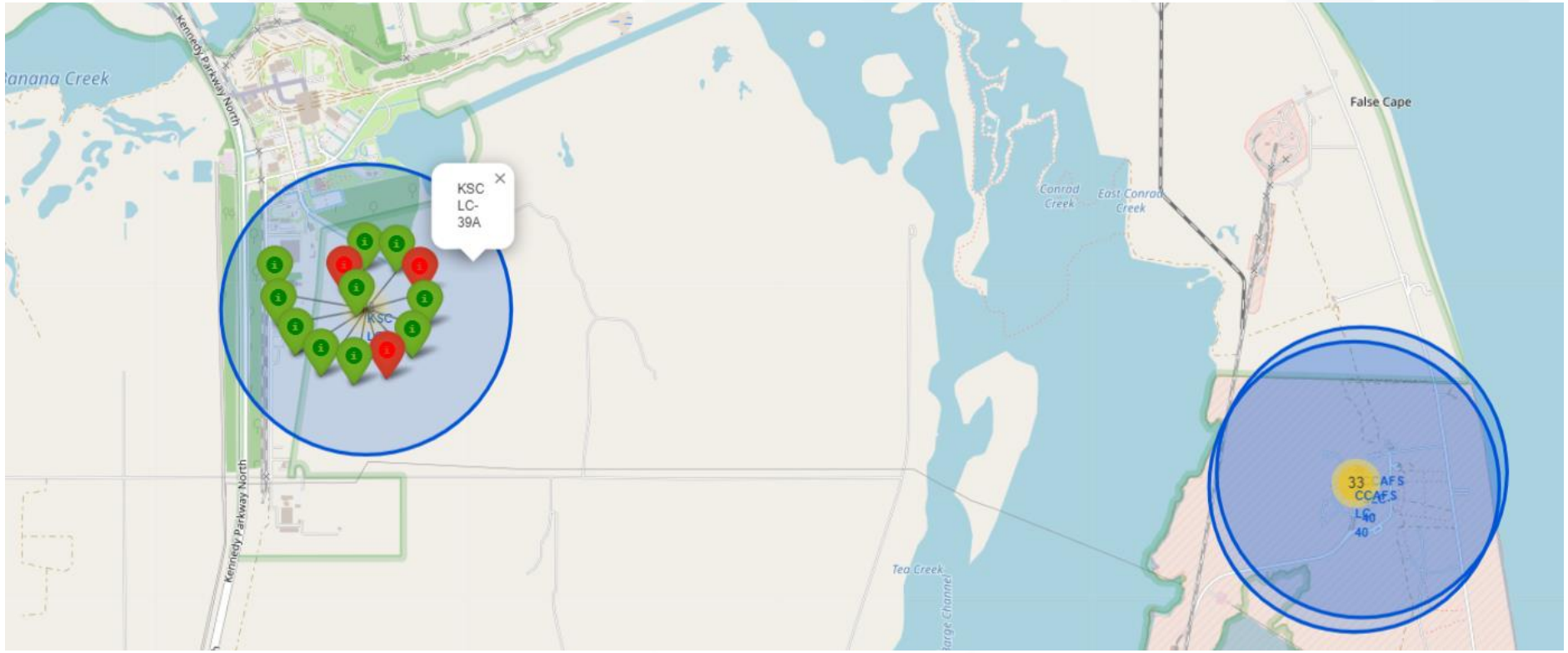
	Landing__Outcome	Date	Count
0	Success (drone ship)	2016-04-08 00:00:00	5
1	Success (ground pad)	2015-12-22 00:00:00	3

RESULTS

- Interactive map with Folium results
- Since spaceX launches come from different launch sites I displayed the information of successful and failed launches as a cluster of points on the map. Through zooming in and out you can observe the clusters of successful launches and failed launches.
  - CCAFS SLC-40: 3 success and 4 failure
  - CCAFS LC-40: 7 success and 19 failure
  - KSC LC-39A: 10 success and 3 failure

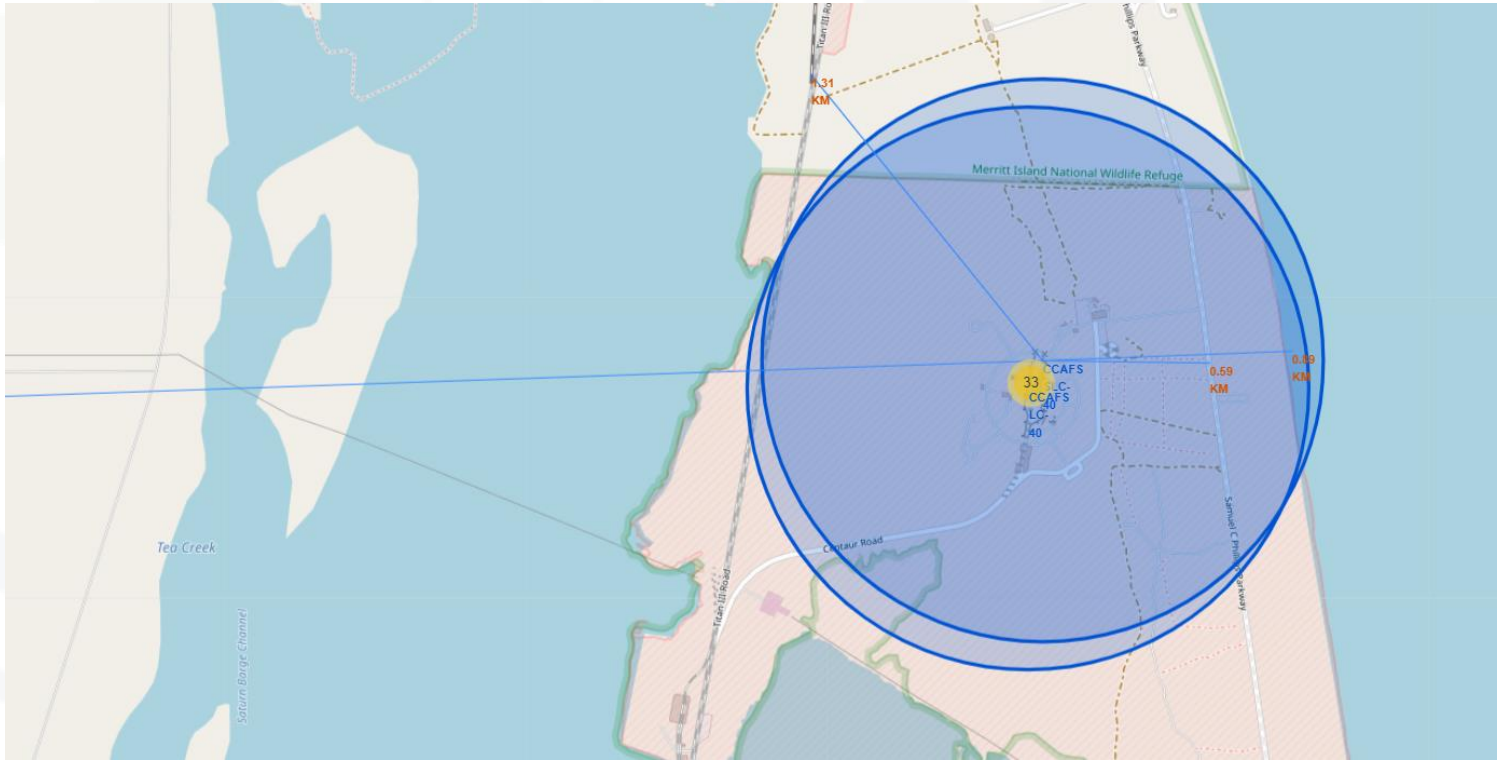
RESULTS

- Interactive map with Folium results
  - KSC LC-39A has the most success rate (76.9%)



RESULTS

- Interactive map with Folium results
- I drew a line from the CCAFS SLC-40 launch site to its closest proximities and calculated the distances between them using their coordinates.
  - Distance from coastline: 0.89 KM
  - Distance from railway: 1.31 KM
  - Distance from highway: 0.59 KM
  - Distance from Orlando: 78.63 KM



DASHBOARD

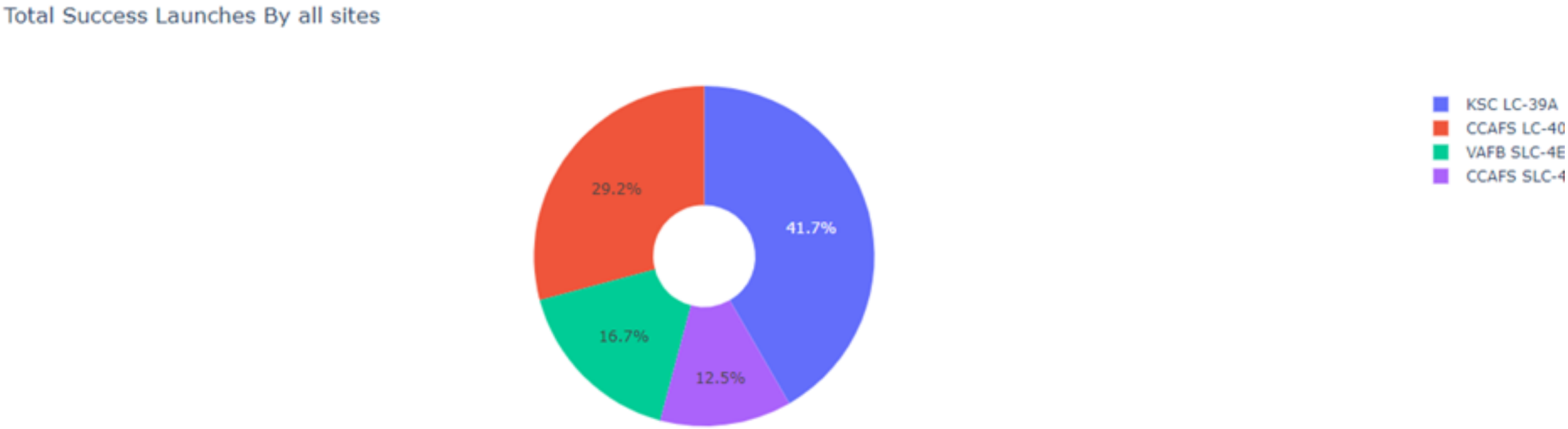
- Plotly Dash is a Python library that makes it easier to create a dashboard for us as Data scientists. With a simple interactive dashboard, one can change the inputs to see a representation of values in graphs.

- Dashbord are available in my GitHub

DASHBOARD TAB 1

Plotly Dash dashboard tab 1 results

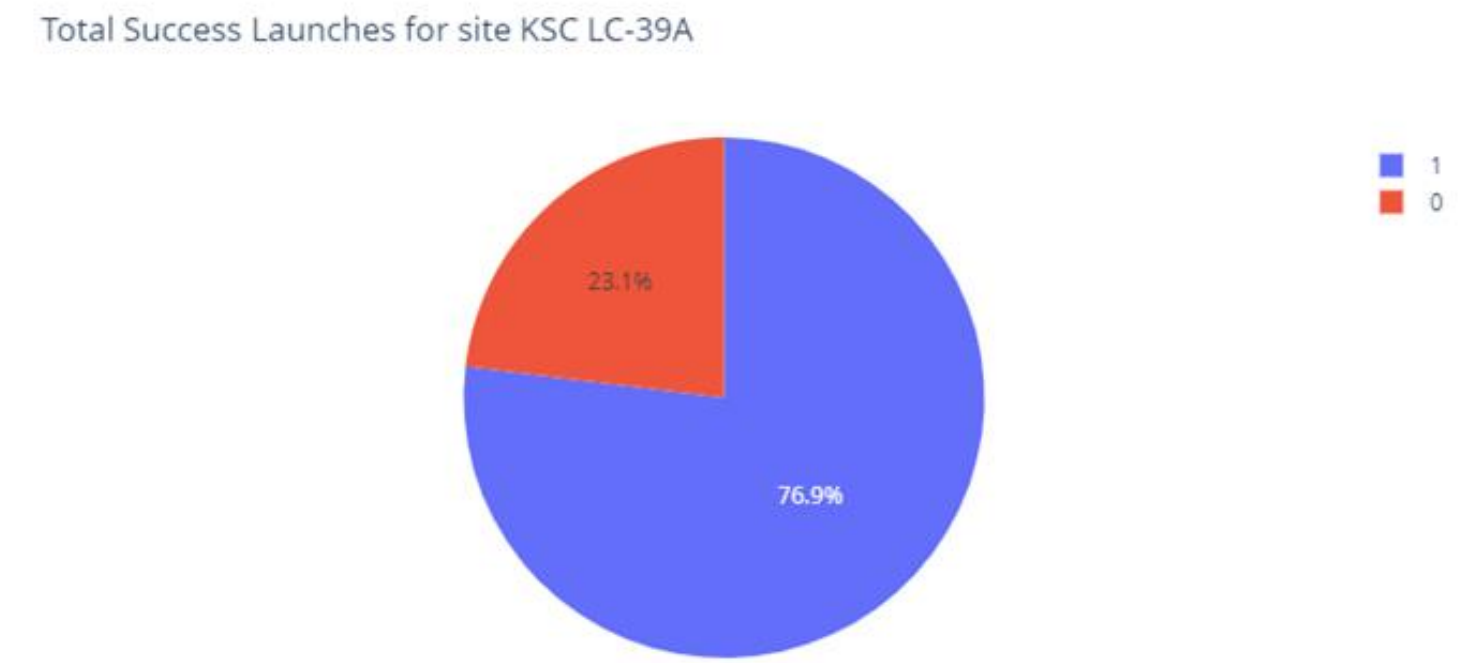
- KSC LC-39A has the largest successful launches



DASHBOARD TAB 2

Plotly Dash dashboard tab 2 results

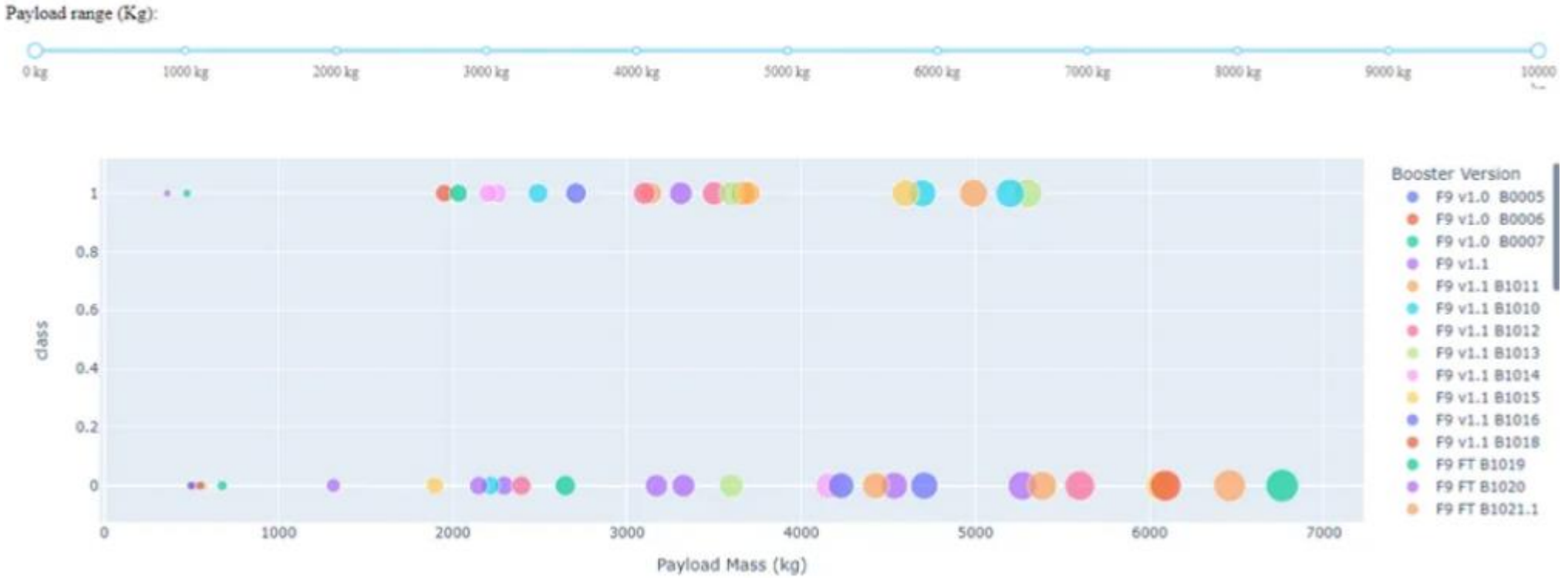
- KSC LC-39A has the highest launch success rate



DASHBOARD TAB 3

Plotly Dash dashboard tab 3 results

- Among F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.), FT has the highest launch success rate with 15 successes and 8 failures



RESULTS

- predictive analysis (classification) results
  - I trained the ML algorithms using the SpaceX past data and tuned the model parameters to achieve the best model accuracy. I evaluated different models performance and compared them through test data in terms of confusion-matrix & score.

```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```

```
knn_cv = GridSearchCV(KNN, parameters, cv=10)
knn_cv.fit(X, Y)
knn_cv.best_estimator_
```

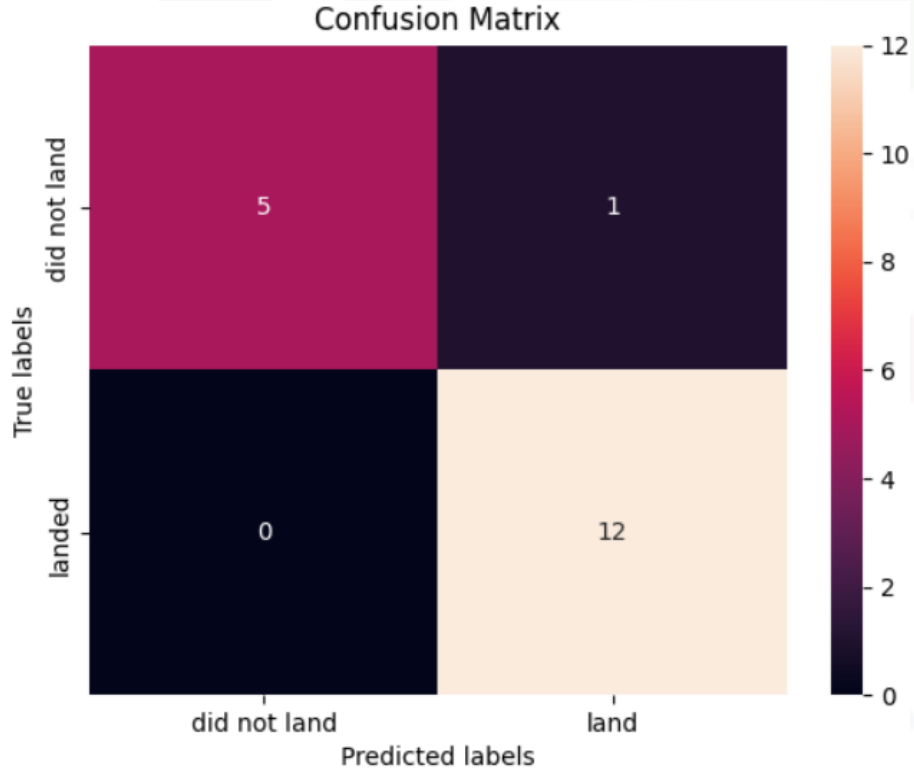
```
KNeighborsClassifier(p=1)
```

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

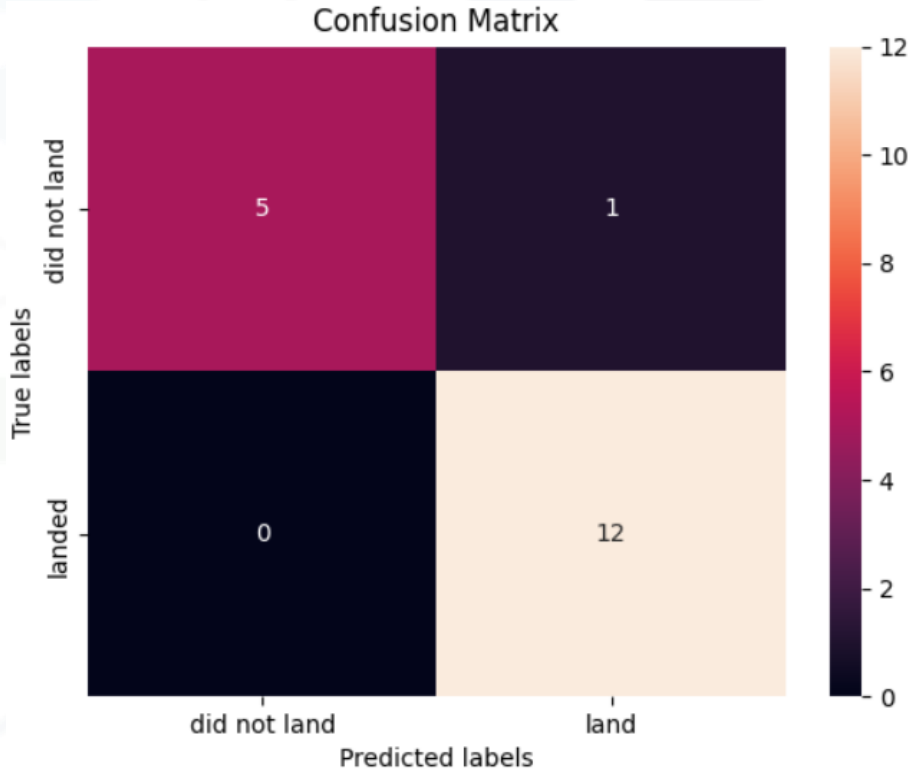
```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 5, 'p': 1}
accuracy : 0.8444444444444444
```

RESULTS

- predictive analysis (classification) results
  - Logistic Regression

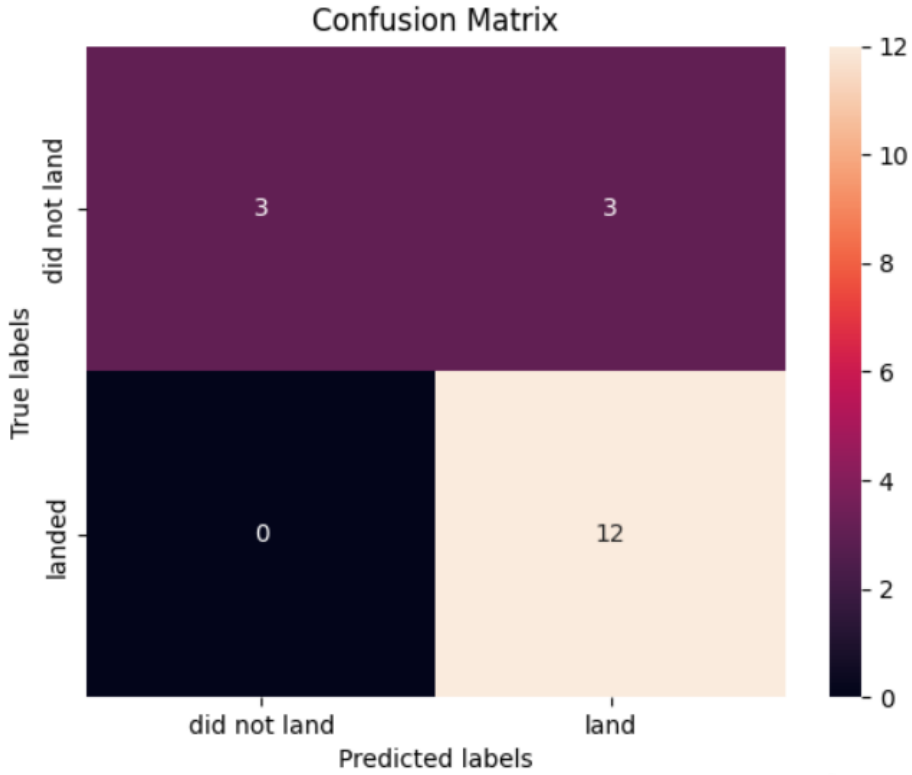


  - Support Vector Machine

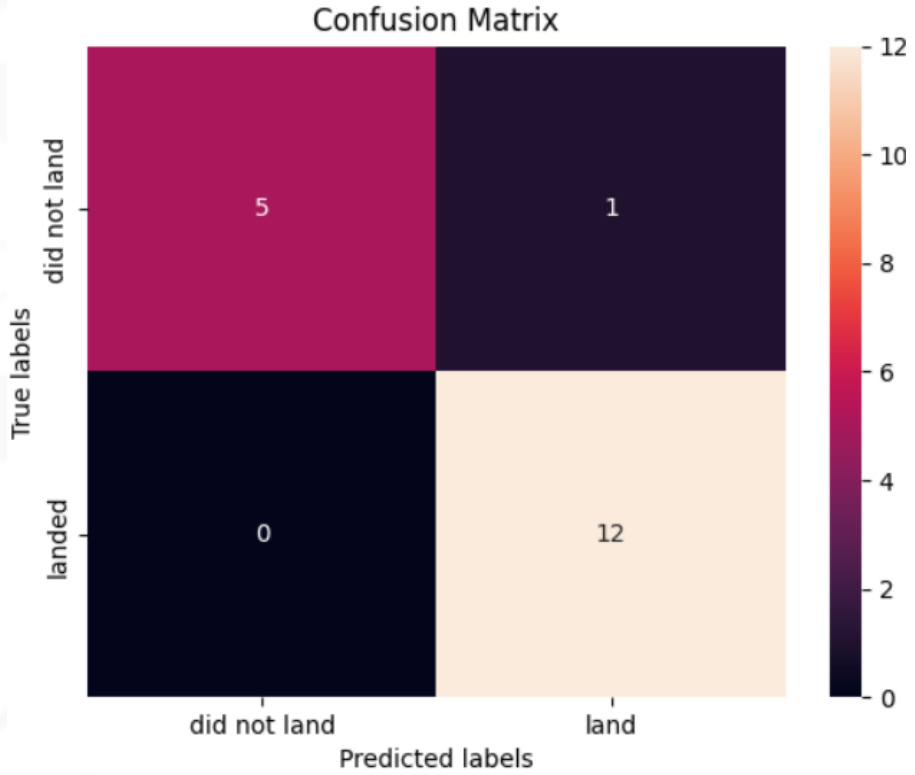


RESULTS

- predictive analysis (classification) results
  - Decision Tree Classifier



  - K-Nearest Neighbors



RESULTS

- predictive analysis (classification) results
  - I understand KNN, Logistic Regression and SVM obtained superior results against Decision Tree.
  - The best accuracy score was about 0.94%

Find the method performs best:

```
# We can dermine the best method/classifier by the highest performance measure results on test(unseen)data

print("score on GridSearchCv with Logistic Regression: ", logreg_cv.score(x_test, y_test))
print("score on GridSearchCv with Support Vector Machine: ", svm_cv.score(x_test, y_test))
print("score on GridSearchCv with  Decision Tree Classifier: ", tree_cv.score(x_test, y_test))
print("score on GridSearchCv with K-Nearest Neighbors: ", knn_cv.score(x_test, y_test))
```

```
score on GridSearchCv with Logistic Regression:  0.9444444444444444
score on GridSearchCv with Support Vector Machine:  0.9444444444444444
score on GridSearchCv with  Decision Tree Classifier:  0.8333333333333334
score on GridSearchCv with K-Nearest Neighbors:  0.9444444444444444
```

DISCUSSION

Findings & Implications:

- My insights from the analysis are that most successful launches were from Kennedy Space Center Launch Complex 39A (KSC LC-39A). The reason behind this includes: It's near Spacex production factory.
- Most failure launches were from Cape Canaveral Space Launch Complex 40 (CCAFS SLC-40).
- The total payload mass carried by boosters launched by NASA (CRS) was 45596.
- ES-L1, GEO, HEO, & SSO orbit types have the highest success rate.
- Falcon Heavy launches mostly to the full payload to maximize the use of the falcon payload capacity.
- Probability of booster landing increases over time by use of the data collected from failing.
- The first successful landing outcome in the ground pad was achieved on 2015-12-22.

CONCLUSION

Using Existing Data and Analyzing it, SpaceX and other rocket companies can be able to get the best ways to reduce the cost of launches, by using appropriate Orbit Type, Launch Site, Pay Load Mass, and other features to evolve before their traditional costly launches lead to their absoluteness and losing their clients.