IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thariq Irza

Outline

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

Executive Summary

- Summary of methodologies

    1. Data Collection using web scraping and SpaceX API.
    2. Exploratory Data Analysis (EDA), including Data Wrangling, Data Visualization, and Interactive Visual Analytics using Folium
    3. Machine Learning Algorithms for Prediction

- Summary of all results

    - Exploratory Data Analysis result
    - Interactive Analytics result
    - Predictive Analytics result from Machine Learning Algorithms

Introduction

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

- Data collection methodology:
  - The data is collected using SpaceX API and Web Scraping
- Perform data wrangling
  - Dealing with missing values
  - One Hot Encoding for categorical data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models
  - Standardize data, split data into training and testing, machine learning models, select the best model

Data Collection

The data were collected using two methods:

1. Using SpaceX API
    - Data was request using get requests to the SpaceX API
    - Decode the response as JSON using .json_normalize()
    - Cleaning the data
2. Using Web Scraping
    - Scrap the SpaceX data from Wikipedia
    - Parse and extract the data using BeautifulSoup

Data Collection – SpaceX API

Source: Data Collection Using SpaceX API

Request the data to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and save the data into a CSV format

Get request for rocket launch data using SpaceX API

↓

Use .json_normalize() to convert json result to dataframe

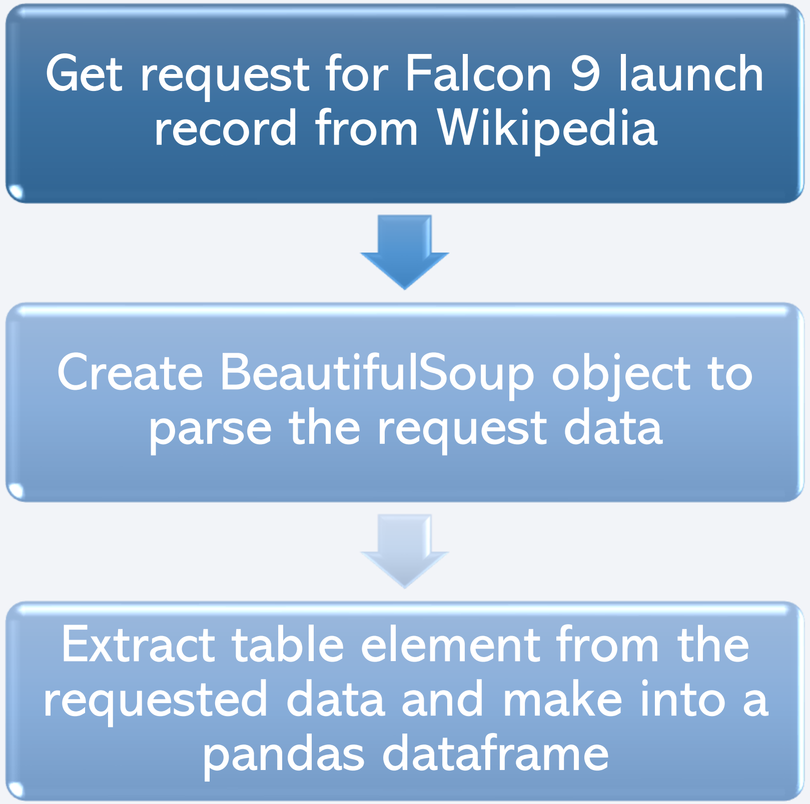
↓

Performed basic data wrangling and save the data into CSV

Data Collection – Scraping

Source: [Data Collection with Web Scraping]()

- Request the Falcon 9 data launch records from Wikipedia
- Parse the data using BeautifulSoup and Extract the table and convert into a pandas dataframe



Data Wrangling

Source: [Data Wrangling]

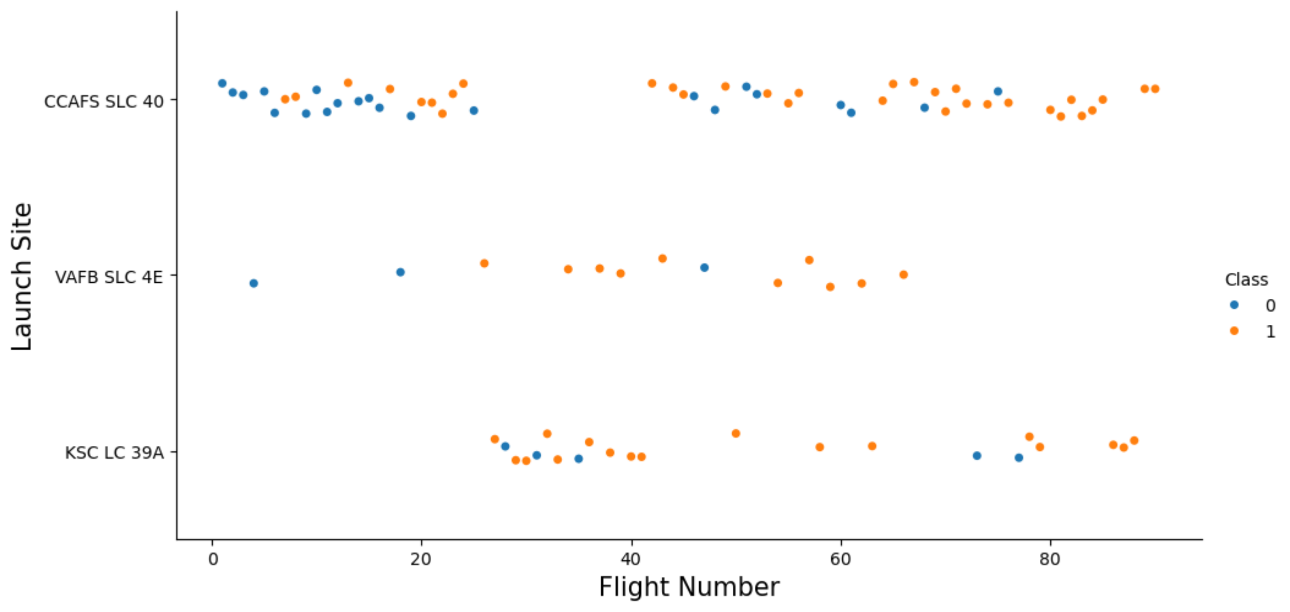
- Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then the summaries launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
- Finally, the landing outcome label was created from Outcome column.

EDA with Data Visualization

Source: [EDA Data Visualization]

The data was explored using scatterplot, bar chart, and line chart to visualize the data. Scatterplot show the relationship between a pair of feature. The scatterplot for this project is used for these several pair of features:

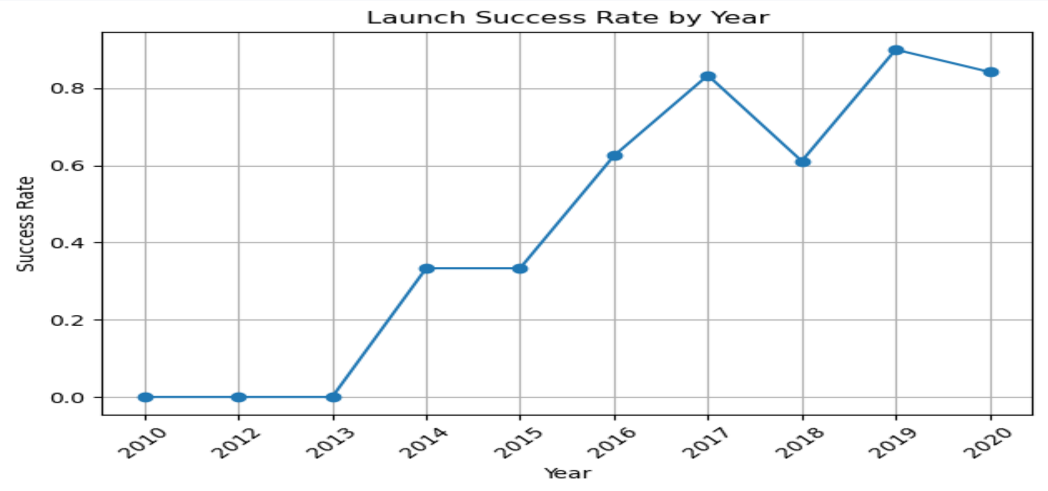
- Payload vs Flight Number
- Flight Number vs Launch Site
- Payload vs Launch Site
- Flight Number vs Orbit Type
- Payload vs Orbit Type

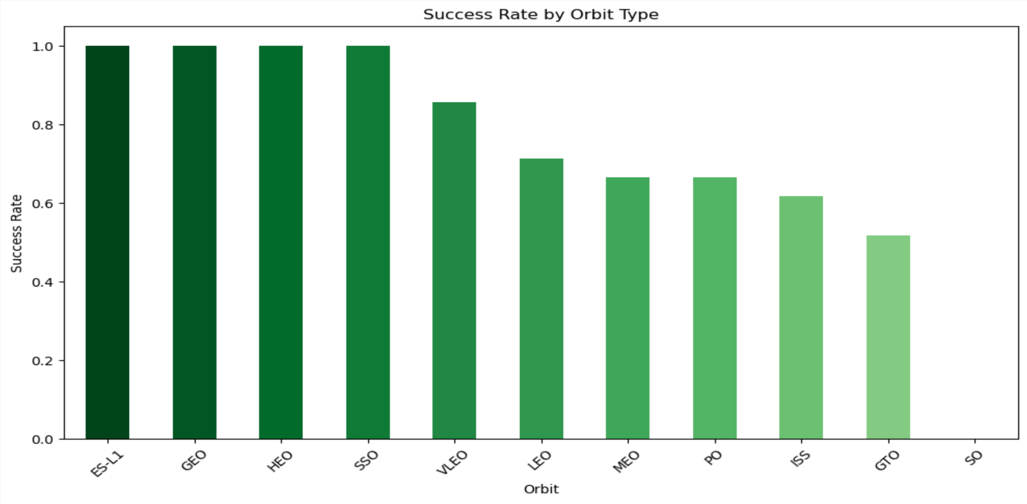


EDA with Data Visualization

Source: [EDA Data Visualization]()

- For this project, bar chart is used to visualize the success rate for the orbit type. Line chart is used to show the trend of the feature over time which in this project, is used to see the success launch for the past 10 years. Also, feature engineering is applied to select the features to be used in success prediction.





EDA with SQL

Source: EDA with SQL

- We loaded the SpaceX dataset into a SQLite database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

Build an Interactive Map with Folium

Source: [Interactive Map with Folium]

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success and assigned a green color to the success value and red the failure value
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

Source: Dashboard using Plotly Dash

- Create
  - Dropdown list with Launch Site values
- Pie chart
  - To show the percentage of the average success launch
- Slider of Payload Mass
  - To filter the range of payload
- Scatter Chart
  - To see the correlation between payload and success rate

Predictive Analysis (Classification)

Source: Machine Learning Prediction

- **Create** NumPy array from the Class column
- **Standardize** the data with StandardScaler. Fit and transform the data.
- **Split** the data using train_test_split
- **Create** a GridSearchCV object with cv=10 for parameter optimization
- **Apply** GridSearchCV on different algorithms:
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree
    - K-Nearest Neighbor
- **Calculate** accuracy on the train and test data using .score() for all models
- **Assess** the confusion matrix for all models
- **Identify** the best model using Accuracy

Results

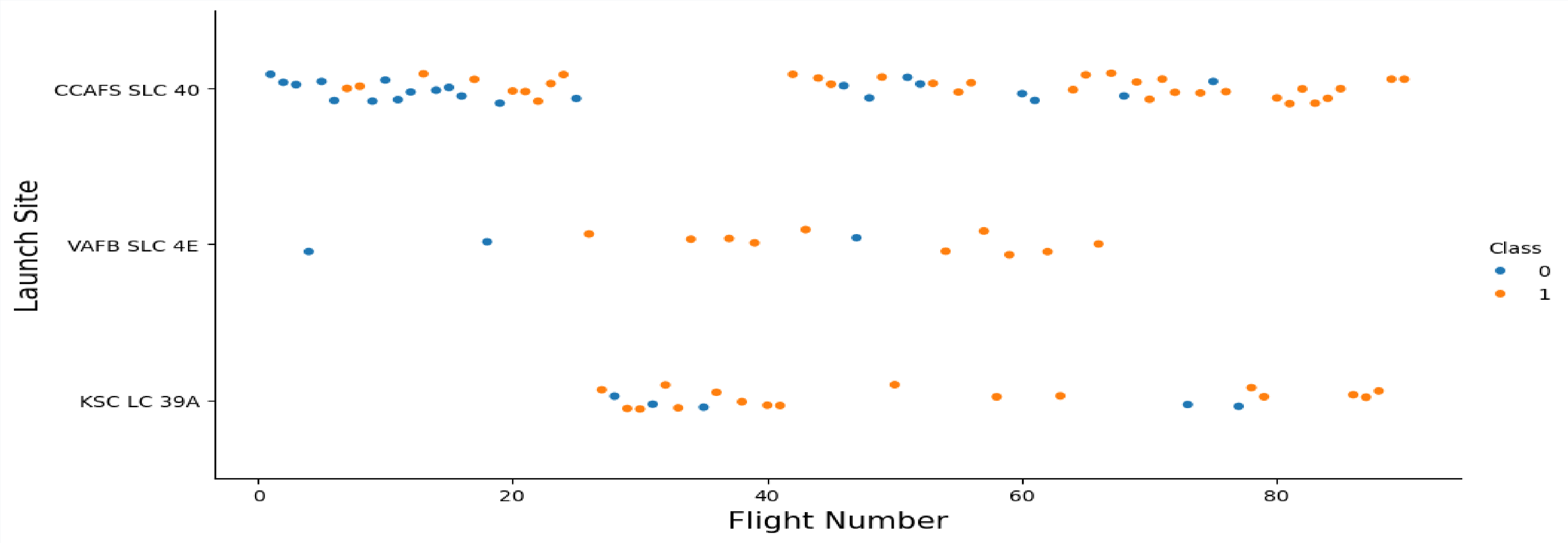
- Exploratory Data Analysis
  - Launch success has improved over time
  - KSC LC-39A has the highest success rate among landing sites
  - Orbits ES-L1, GEO, HEO and SSO have a 100% success rate
- Visual Analytics
  - Most launch sites are near the equator, and all are close to the coast
  - Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities
- Predictive Analytics
  - Decision Tree model is the best predictive model for the dataset

Section 2

Insights drawn from EDA

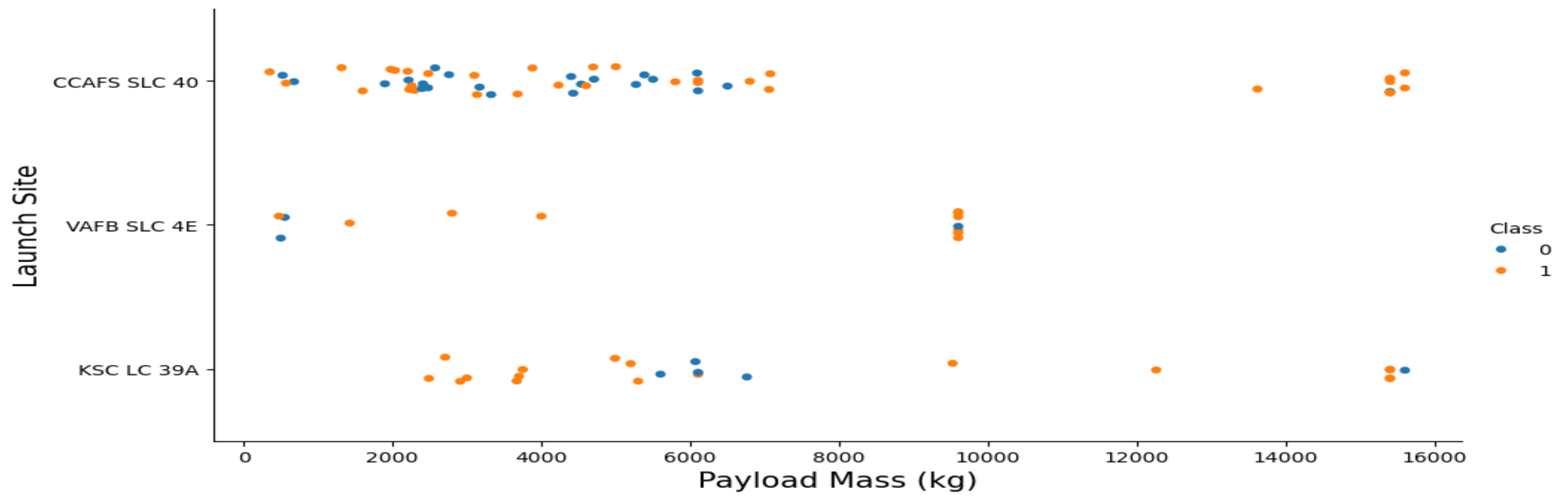
Flight Number vs. Launch Site

- **Earlier flights** has a lower success rate in **Landing Outcome**
- **Later flights** has a higher success rate in **Landing Outcome**. Based on this information, we can infer that, the new launches have a higher success rate in the landing outcome



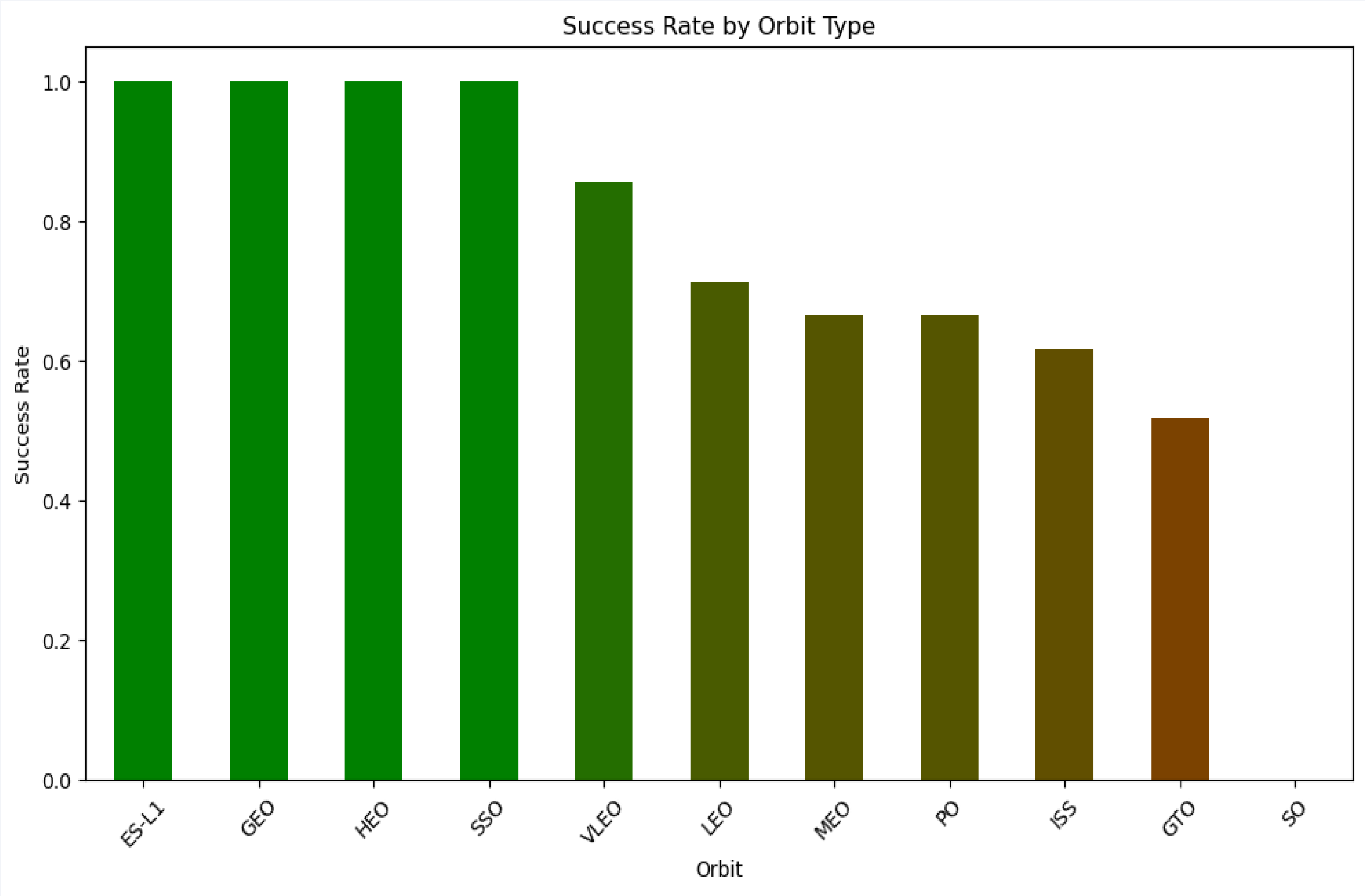
Payload vs. Launch Site

- The **higher** the payload, the **higher** the **success rate** in **Landing Outcome**
- Most launches with a payload **greater than 7000 kg** were successful
- **KSC LC 39A** has a 100% success rate for launches with payload less than 5,500 kg
- **VAFB SLC 4E** has not launches with a payload greater than approximately 10,000 kg



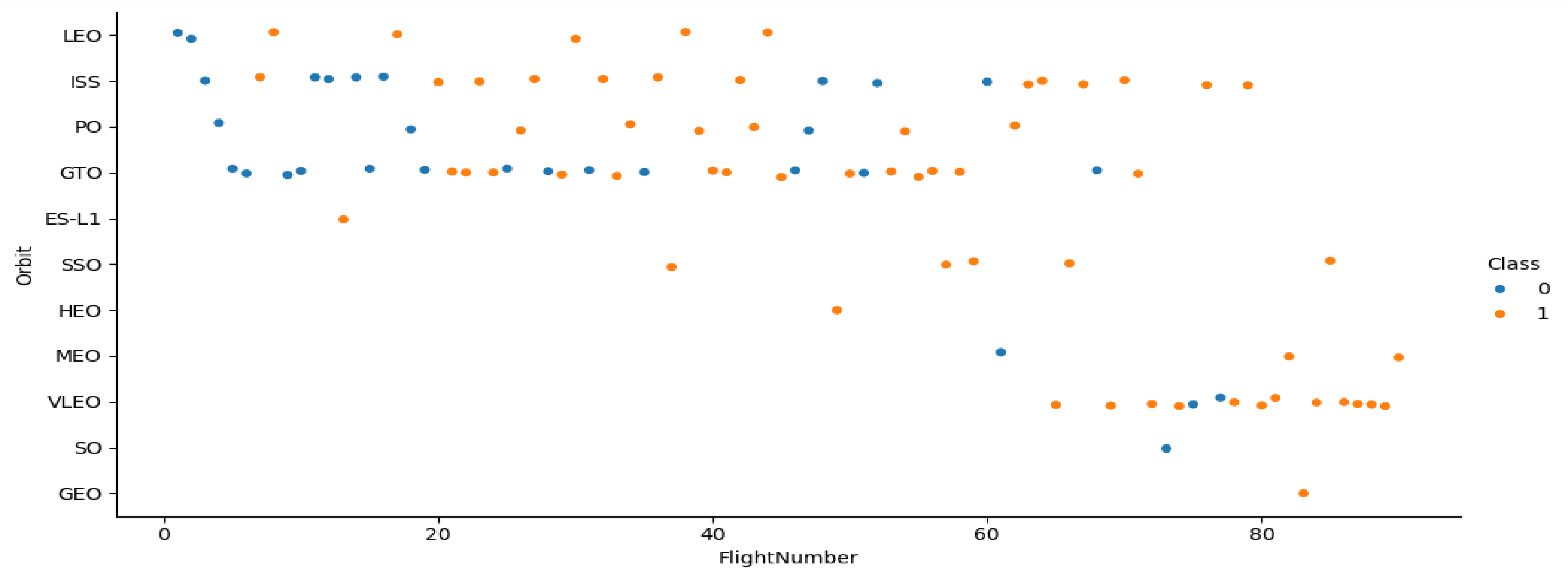
Success Rate vs. Orbit Type

- 90% -100% Success rate: ES-L1, GEO, HEO, SSO, VLEO
- 50% - 80% Success rate: LEO, MEO, PO, ISS, GTO
- <50% Success rate: SO



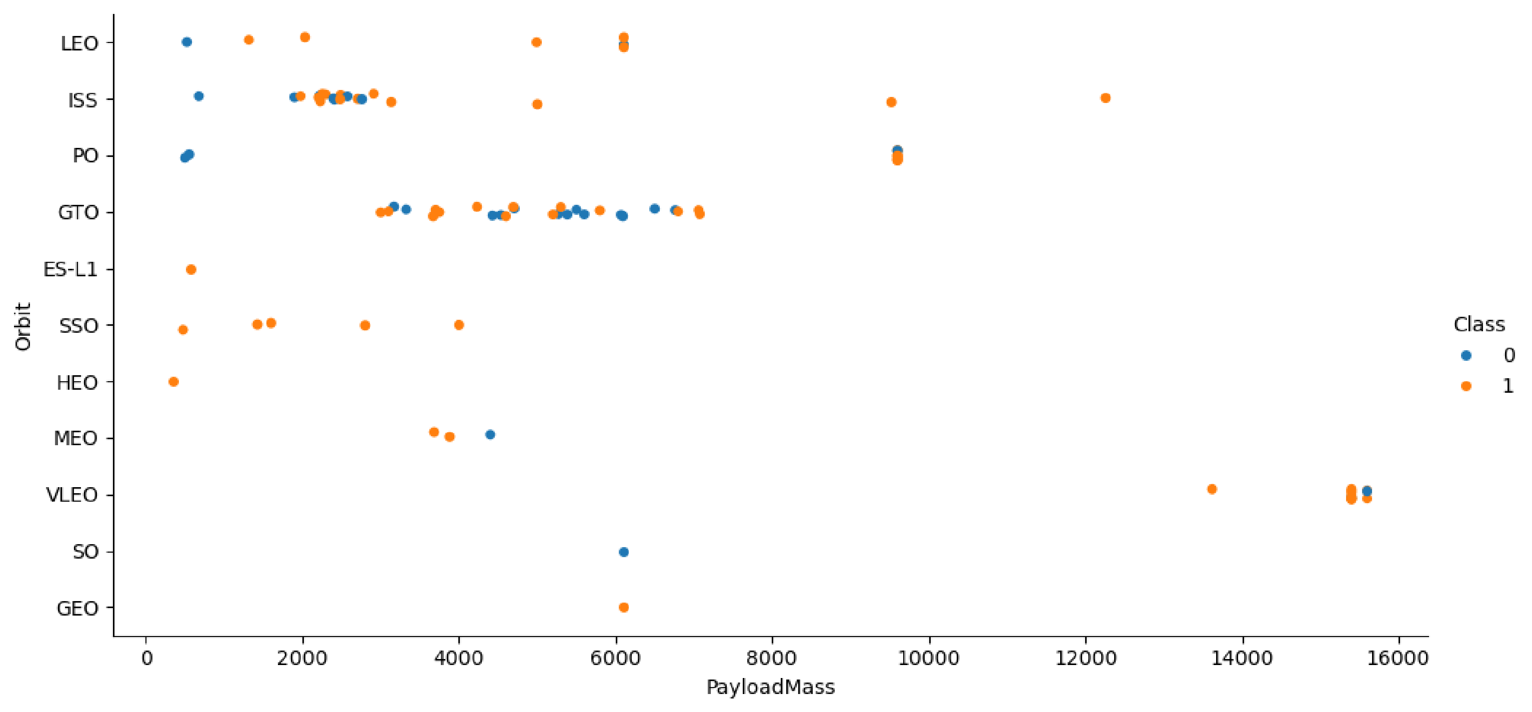
Flight Number vs. Orbit Type

- The **success rate** typically **increases** with **the number of flights** for each orbit
- This relationship is highly apparent for the LEO orbit
- The GTO orbit, however, does not follow this trend



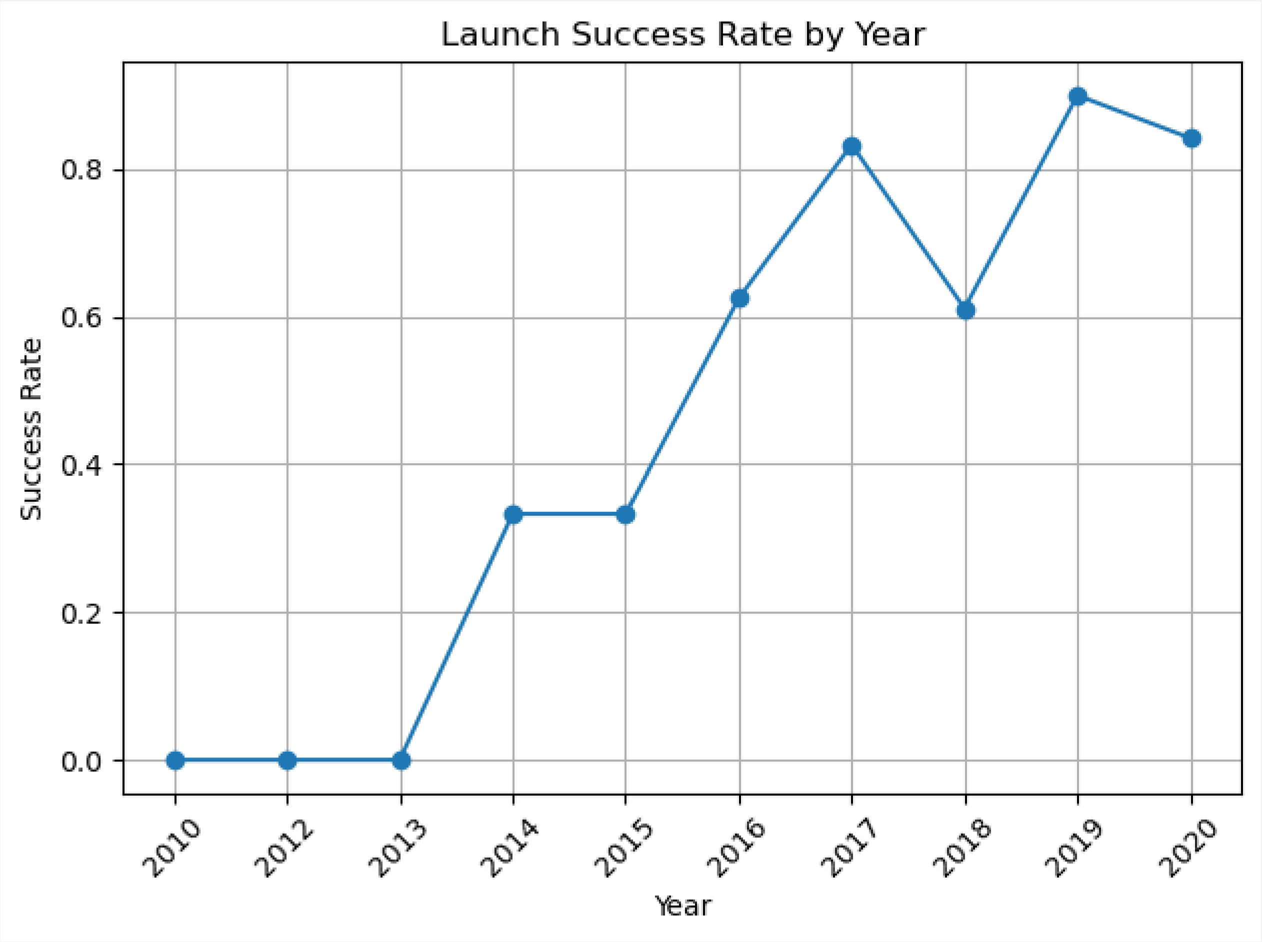
Payload vs. Orbit Type

- **Heavy payloads** are better LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads



Launch Success Yearly Trend

The pattern show that, overall for the past 10 years, the rocket success rate is improved starting from 2013.



All Launch Site Names

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

```
%%sql SELECT DISTINCT Launch_Site
FROM SPACEXTABLE
```

✓ 0.0s Python

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

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

Launch Site Names Begin with 'CCA'

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

```
%%sql SELECT *
FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5
```

✓ 0.0s Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) as 'Total Payload Mass'
FROM SPACEXTABLE
```

[55] ✓ 0.0s Python

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

Total Payload Mass
619967

Average Payload Mass by F9 v1.1

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

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1'
```

[56] ✓ 0.0s Python

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

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

```
%%sql SELECT MIN(Date) as 'First Successfull Landing'
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
```

✓ 0.0s Python

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

First Successfull Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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 SPACEXTABLE
WHERE Landing_Outcome ='Success (drone ship)'
AND PAYLOAD_MASS__KG_ >4000
AND PAYLOAD_MASS__KG_ < 6000
```

✓ 0.0s Python

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

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql SELECT COUNT(Mission_Outcome) AS 'Success Outcome'
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success%'
```

[59] ✓ 0.0s Python

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

Success Outcome
100

```
%%sql SELECT COUNT(Mission_Outcome) AS 'Failure Outcome'
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Failure%'
```

[60] ✓ 0.0s Python

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

Failure Outcome
1

Boosters Carried Maximum Payload

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

```
%%sql SELECT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
FROM SPACEXTABLE)
```

[61] ✓ 0.0s Python

* sqlite:///my_data1.db
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
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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

```
%%sql SELECT
    SUBSTR(Date, 6, 2) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM
    SPACEXTABLE
WHERE
    SUBSTR(Date, 0, 5) = '2015' AND
    Landing_Outcome LIKE '%Failure (drone ship)%'
```

[62] ✓ 0.0s Python

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

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

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.

```
%%sql SELECT
    Landing_Outcome,
    COUNT(*) AS OutcomeCount
FROM SPACEXTABLE
WHERE
    Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY OutcomeCount DESC;
```

[63] ✓ 0.0s Python

* sqlite:///my_data1.db
Done.

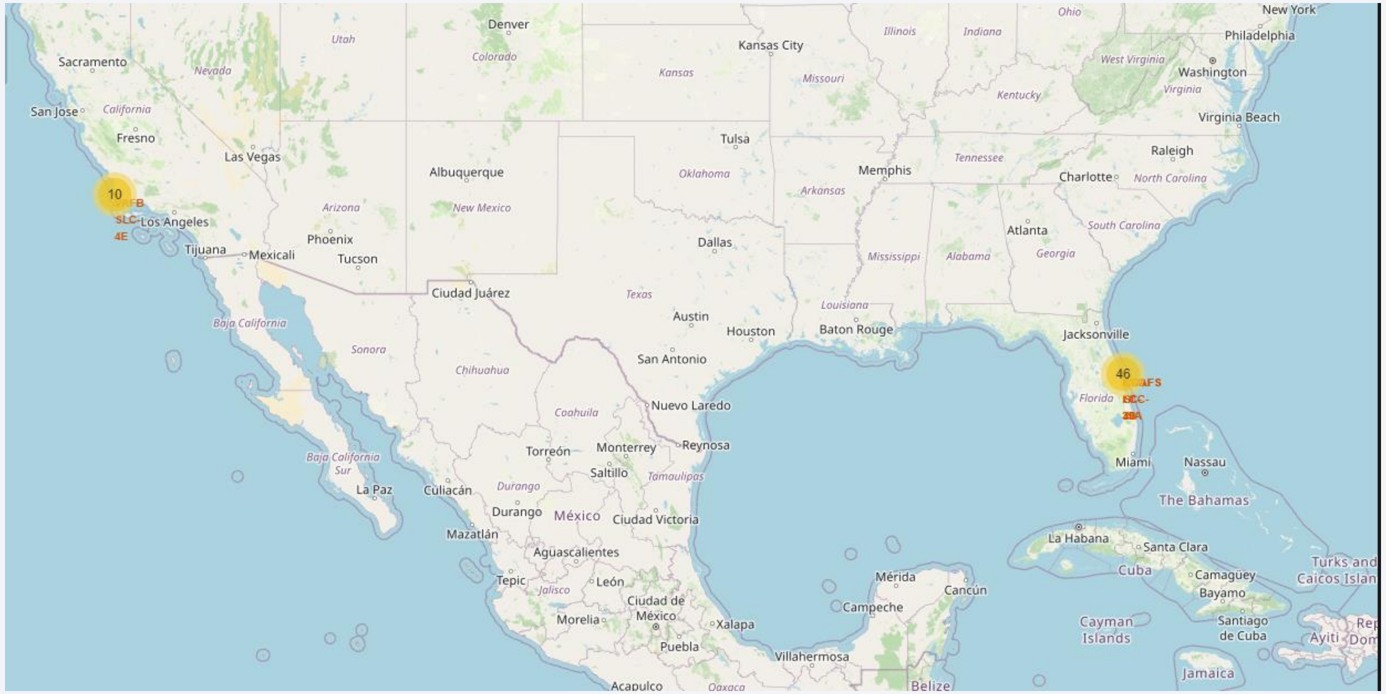
Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

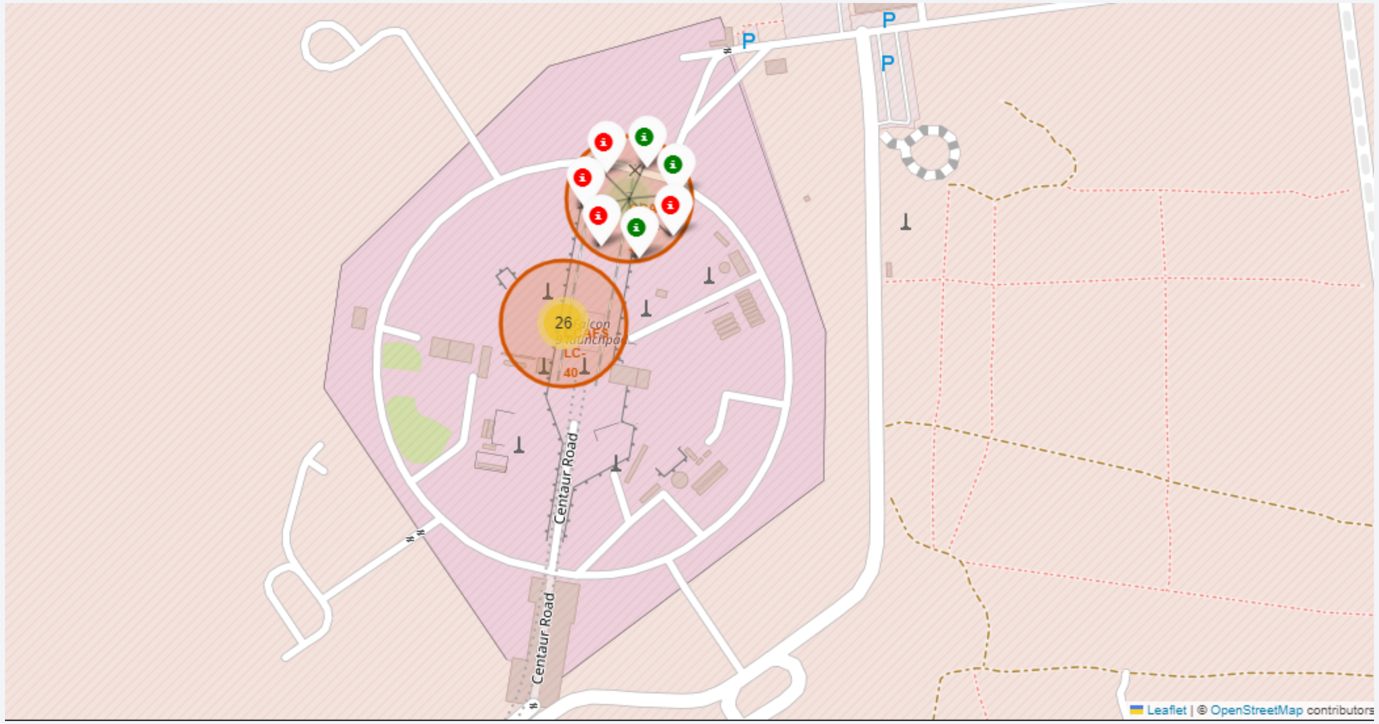
Launch Site

Based on the map, not all launch sites are located close to the Equator line. Although the distances to the equator line are not substantial, the advantage of an additional velocity boost from Earth's rotational speed is less significant for launch sites farther from the Equator. However, all the launch sites are very close to coastal areas, which provide a safer launch environment by minimizing risks in case of failures, offering wide areas for rocket trajectories, and facilitating easier transportation of rocket parts.



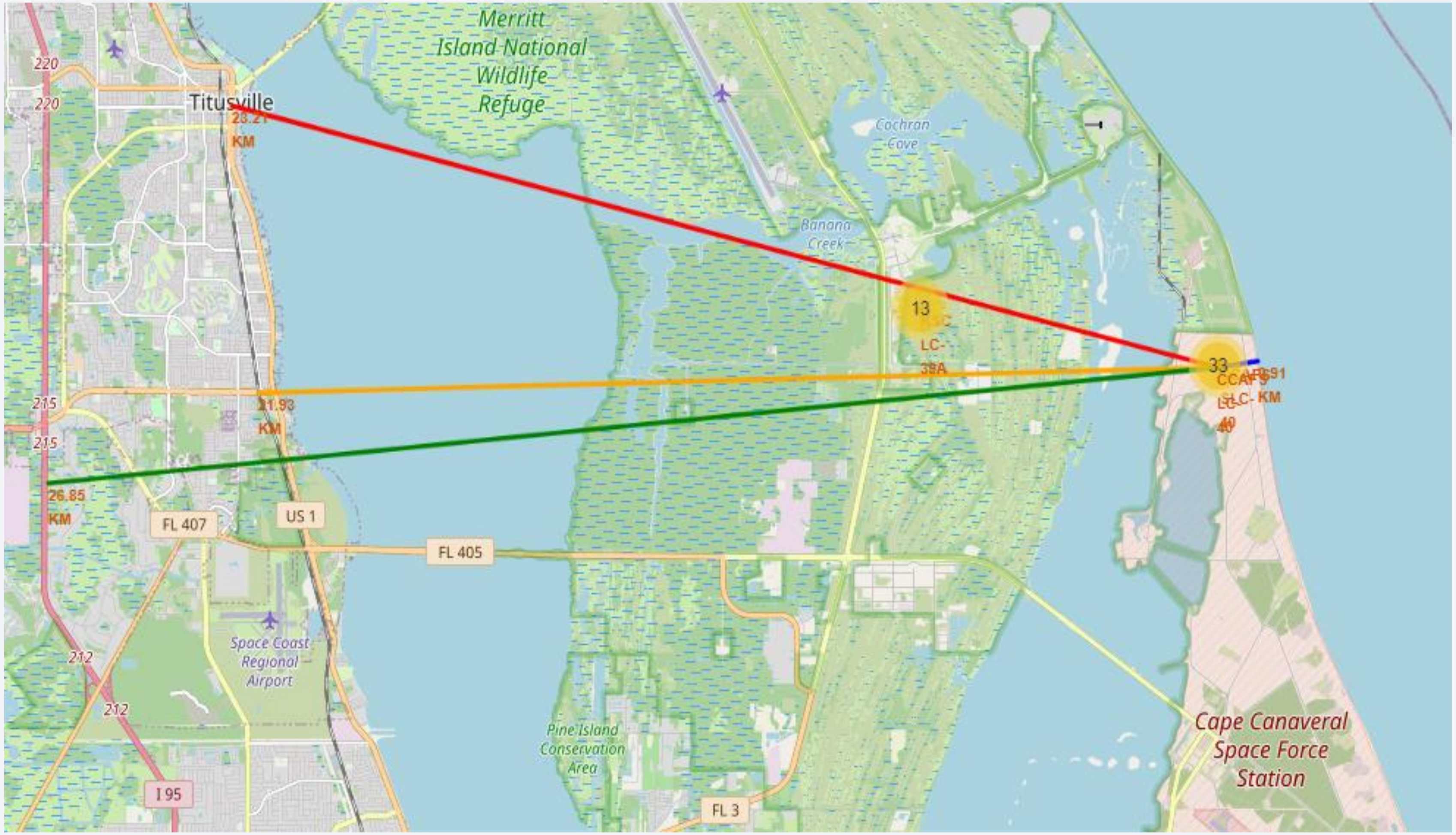
Launch Outcomes

- Green markers for successful launches
- Red markers for unsuccessful launches
- Launch site **CCAFS SLC-40** has 42.9% success rate in the launch outcome



Distance to Proximities

- **.91 km** from the nearest coastline
- **21.93 km** from the nearest railway
- **23.21 km** from the nearest city
- **26.85 km** from the nearest highway

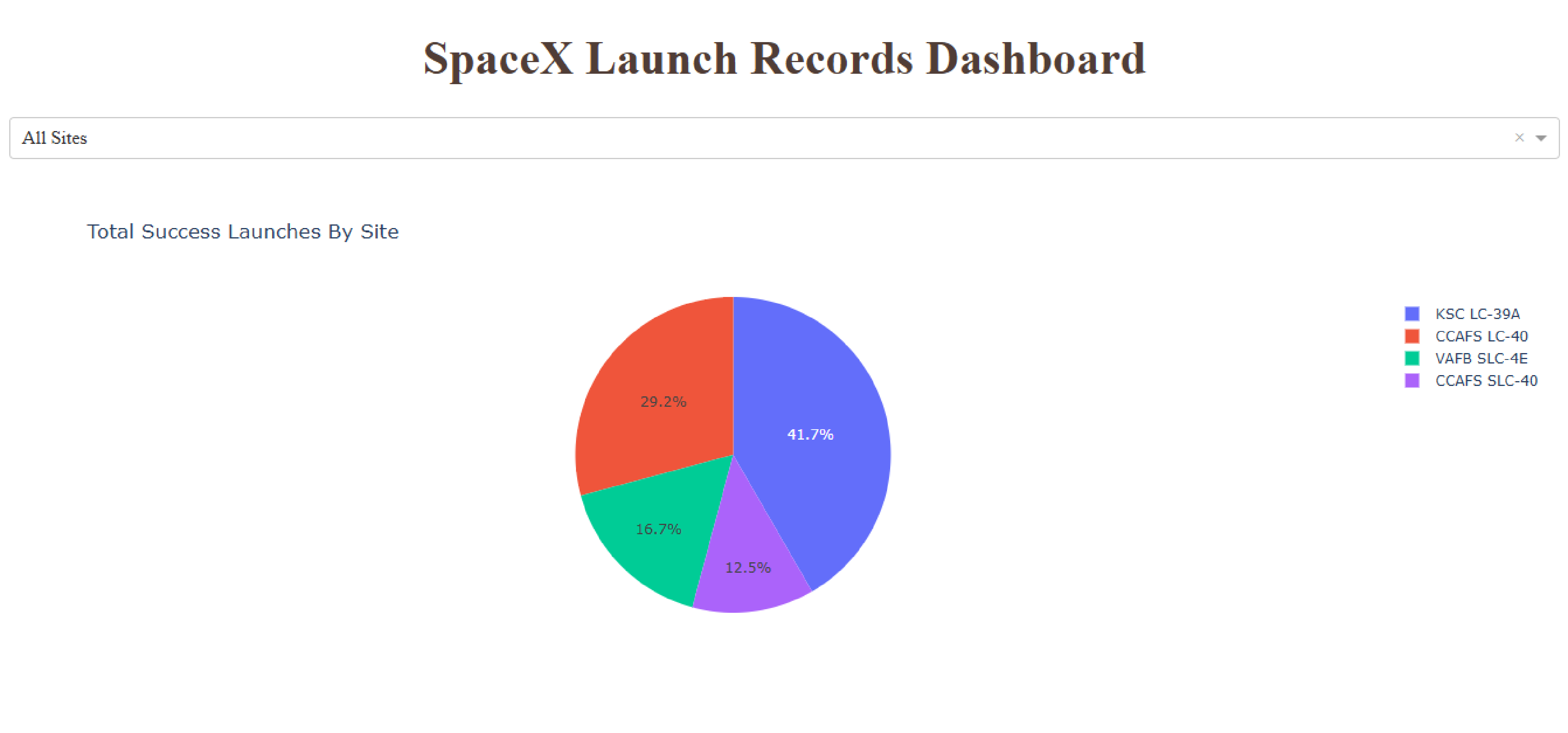


Section 4

Build a Dashboard with Plotly Dash

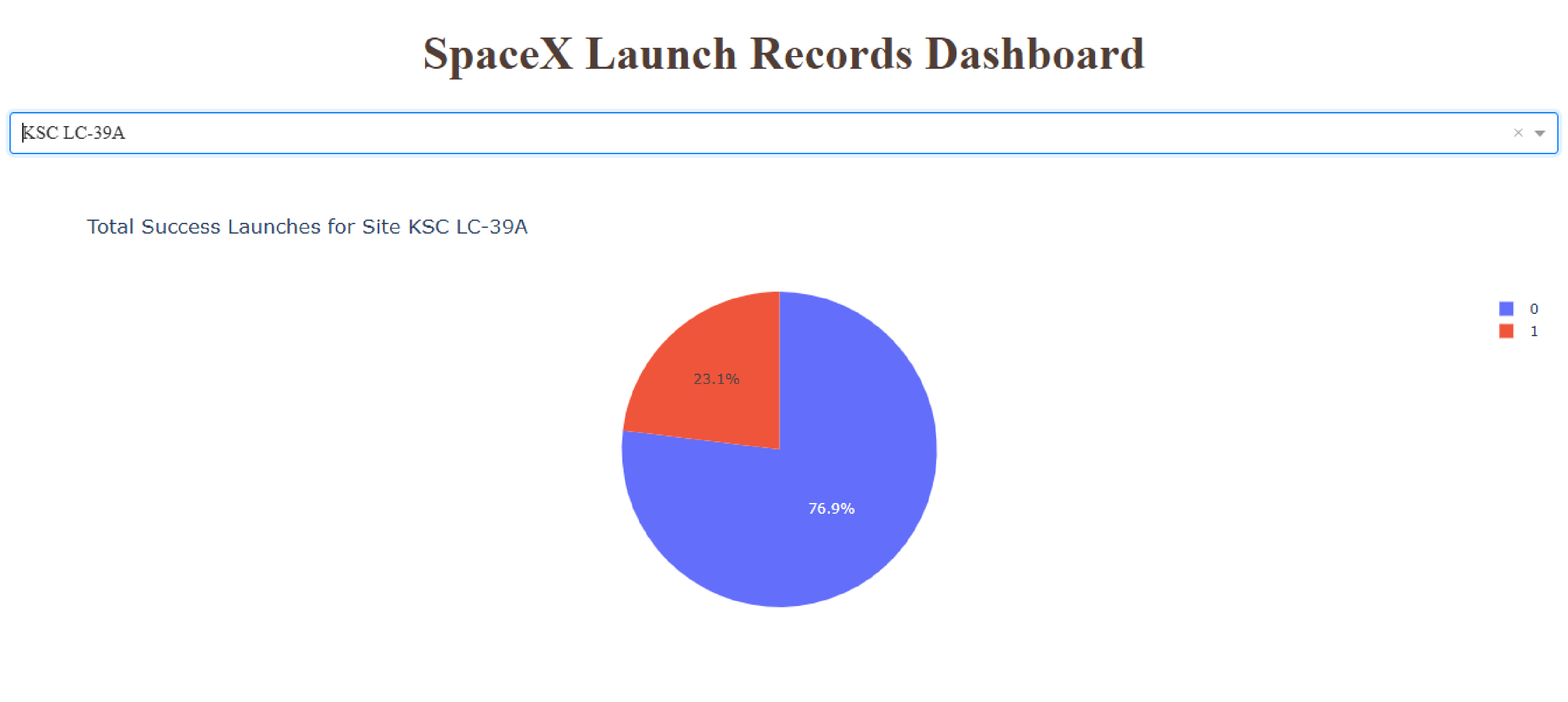
Pie Chart Launch Site

KSC LC-39A has the most successful launch with a percentage of **41.7%**



Pie Chart Success (KSC LC-39A)

KSC LC-39A has the highest percentage of success rate from the other launch sites, with the success percentage of **76.9%**



Payload Mass and Success

- Payload between **2000 kg** and **below 6000 kg** has the highest success rate in launch outcome.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All the model performed the same level and has the **same score** for the test accuracy. But for the train accuracy, Decision Tree has a slightly **better score** amongst the models
- Because of the same test accuracy, we create a method to search for the best model using the max() function for the model test accuracy. The result show that KNN is the best algorithm for this project.

	Models	Training Accuracy Score	Testing Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

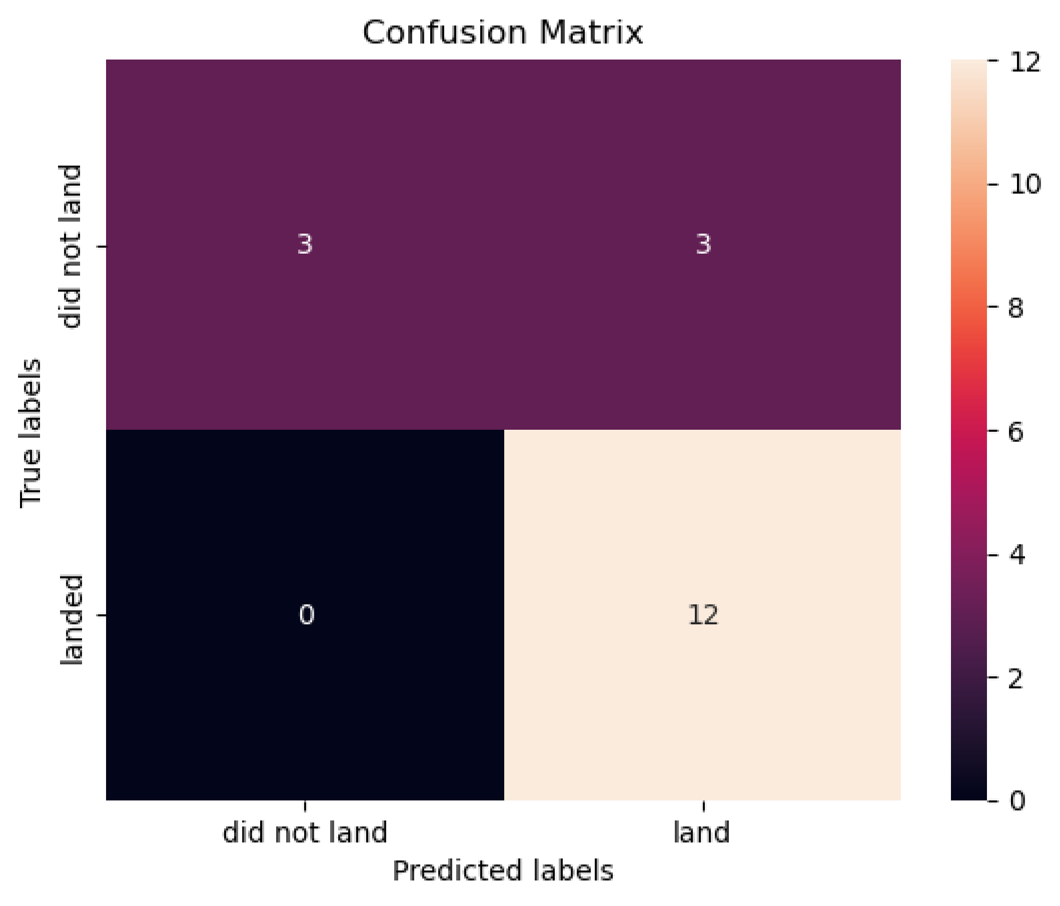
```
models = {'KNeighbors':knn_cv_test_accuracy,
              'DecisionTree':tree_cv_test_accuracy,
              'LogisticRegression':logreg_cv_test_accuracy,
              'SupportVector': svm_cv_test_accuracy}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

✓ 0.0s

```
Best model is KNeighbors with a score of 0.8333333333333334
Best params is : {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
```

Confusion Matrix

- The **confusion matrix** for the **KNN** shows that the classifier can distinguish between the different classes. The major problem is the **false positives** .i.e., **unsuccessful** landing marked as **successful** landing by the classifier.



Conclusions

- The **larger** the flight number at a launch site, the **greater** the **success rate** at a launch site.
- Launch success rate started to increase in 2013.
- Orbits **ES-L1, GEO, HEO, SSO, VLEO** had the most success rate.
- **KSC LC-39A** had the most successful launches of any sites.
- **The K-Nearest Neighbor** is the best machine learning algorithm for this task.

Thank you!