Applied Data Science Capstone Project

Predicting Falcon 9 First Stage Landing Success Using Machine Learning

Ali Hosein Ghanemi

Skills Network

IBM

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

The objective of this research is to analyze SpaceX Falcon 9 data gathered from multiple sources and use machine learning models to predict the success of the first stage landing. This will help other space agencies determine whether they should bid against SpaceX.

- Summary of Methodologies

The following methods and concepts were used for data collection, analysis, model building, and prediction:

- Data collection via API and web scraping
- Data transformation through data wrangling
- Exploratory data analysis with SQL and data visualizations
- Creation of an interactive map with Folium to analyze the proximity of launch sites
- Finally, a predictive model was built to forecast the success of Falcon 9's first stage landing.
- Summary of Results
  - Data analysis results
  - Predictive model analysis results

Introduction

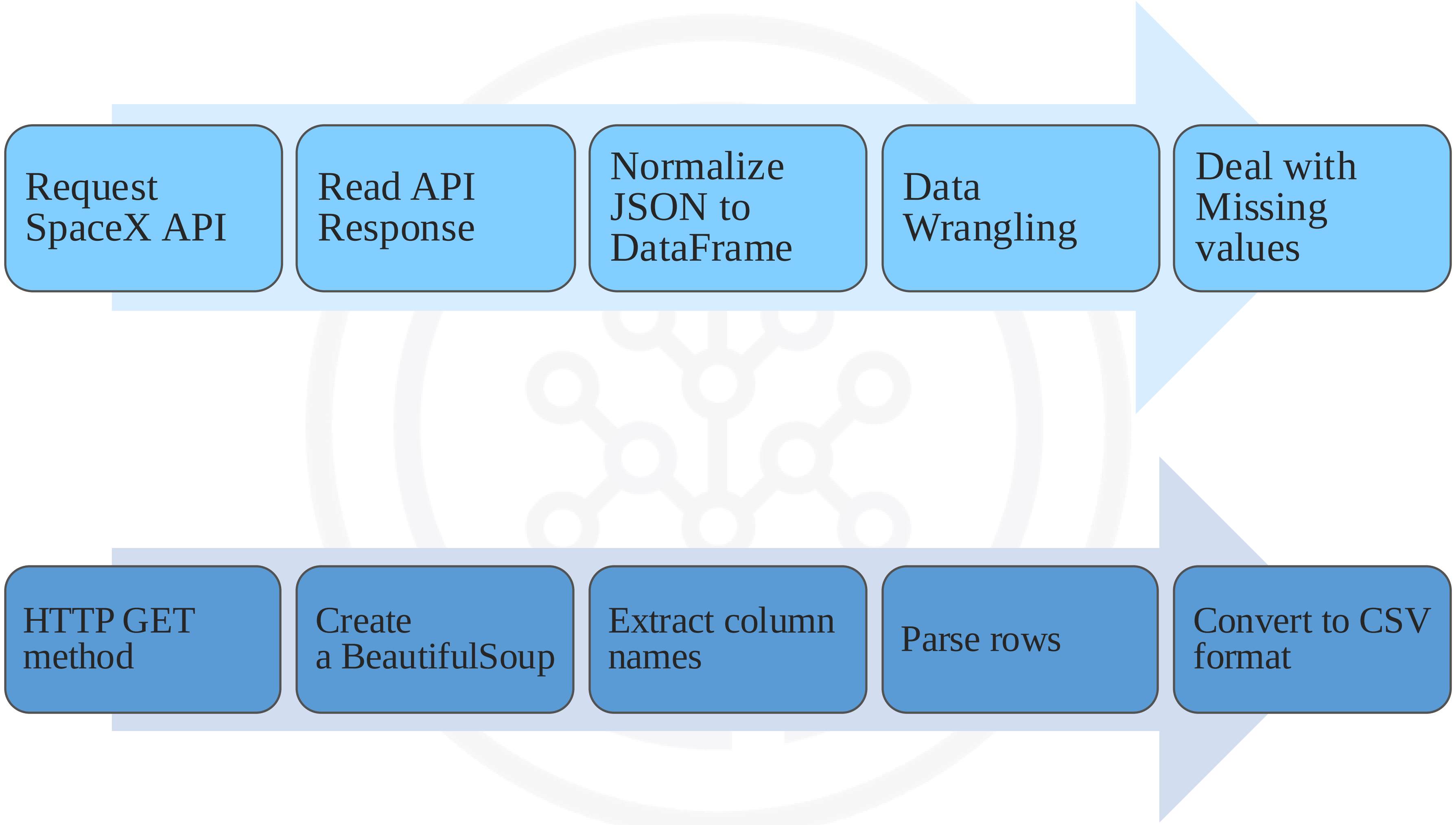
SpaceX is the leading company in the commercial space industry, making space travel more affordable. They list the cost of a Falcon 9 rocket launch on their website at 62 million dollars, while other providers charge more than 165 million dollars. The savings come mainly because SpaceX can reuse the first stage of the rocket. So, by predicting if the first stage will land, we can estimate the launch cost. Using public data and machine learning models, we aim to predict whether SpaceX will reuse the first stage.

Methodology

Methodology

- Data collection methodology:
  - SpaceX Rest API
  - Web scrap Falcon 9 and Falcon Heavy launch records from Wikipedia
- Performed data wrangling:
  - Filtering the data
  - Dealing with missing values
  - Using One Hot Encoding to prepare the data to a binary classification
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium
- Performed predictive analysis using classification models - Building, tuning and evaluation of classification models to ensure the best results

Data Collection



Request SpaceX API

Read API Response

Normalize JSON to DataFrame

Data Wrangling

Deal with Missing values

HTTP GET method

Create a BeautifulSoup

Extract column names

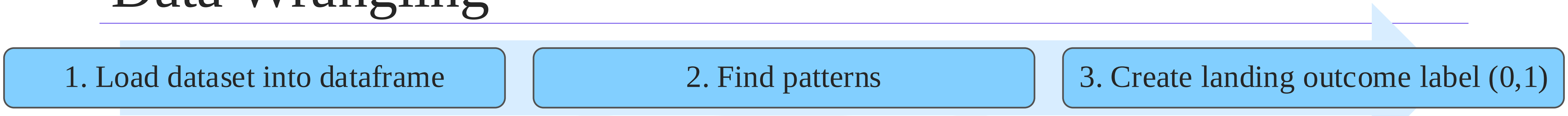
Parse rows

Convert to CSV format

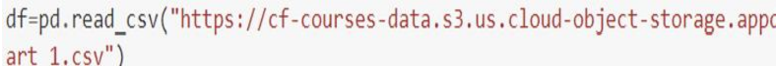
Data Wrangling

- The dataset has information about different mission outcomes, which are turned into labels for training the machine learning model. A Label 1 means the booster landed successfully, while a Label 0 means the booster didn't land successfully. The landing scenarios considered to create these labels are as follows:
- True Ocean: The booster successfully landed in a specific region of the ocean.
- False Ocean: The booster unsuccessfully landed in a specific region of the ocean.
- True RTLS: The booster successfully landed on a ground pad.
- False RTLS: The booster unsuccessfully landed on a ground pad.
- True ASDS: The booster successfully landed on a drone ship.
- False ASDS: The booster unsuccessfully landed on a drone ship.
- None ASDS and None None: These represent failures to land.

Data Wrangling


1. Load dataset into dataframe
2. Find patterns
3. Create landing outcome label (0,1)

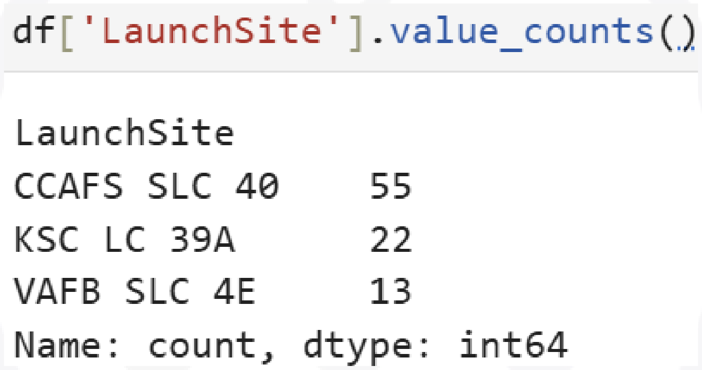
1. Load SpaceX dataset into a csv Dataframe



```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appc
art 1.csv")
```

2. Evaluate the data to get a better understanding and find pattern

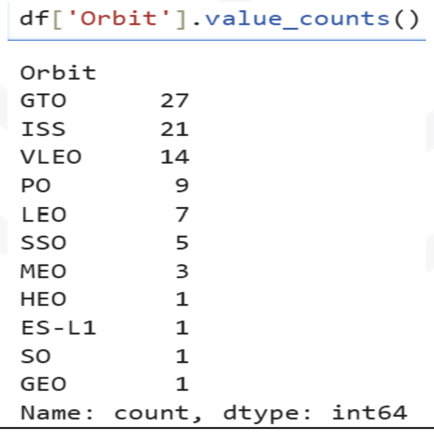
- Calculate the number of launches on each site



```
df['LaunchSite'].value_counts()

LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

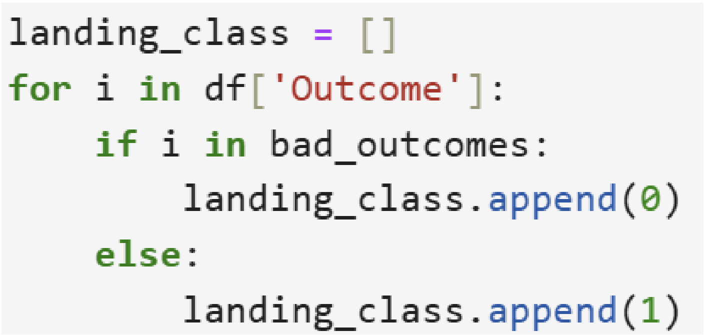
- Calculate the number and occurrence of each orbit



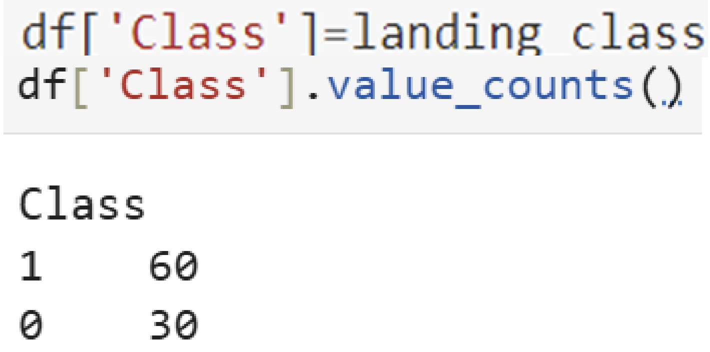
```
df['Orbit'].value_counts()

Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

3. Create landing feature as the prediction column of the dataframe



```
landing_class = []
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```



```
df['Class']=landing_class
df['Class'].value_counts()

Class
1     60
0     30
```

EDA with SQL

- To better understand the SpaceX dataset, the following SQL queries and operations were carried out in the IBM virtual lab environment, Skills Network Labs (SN Labs):
  1. Display the names of the unique launch sites in the space mission
  2. Display 5 records where launch sites begin with the string 'CCA'
  3. Display the total payload mass carried by boosters launched by NASA (CRS)
  4. Display average payload mass carried by booster version F9 v1.1
  5. List the date when the first succesful landing outcome in ground pad was achieved
  6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  7. List the total number of successful and failure mission outcomes
  8. List the names of the booster_versions which have carried the maximum payload mass(Use a subquery)
  9. 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.
  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.

Perform EDA and Feature Engineering using Pandas and Matplotlib

During the Exploratory Data Analysis (EDA), the following charts were made to better understand the dataset:

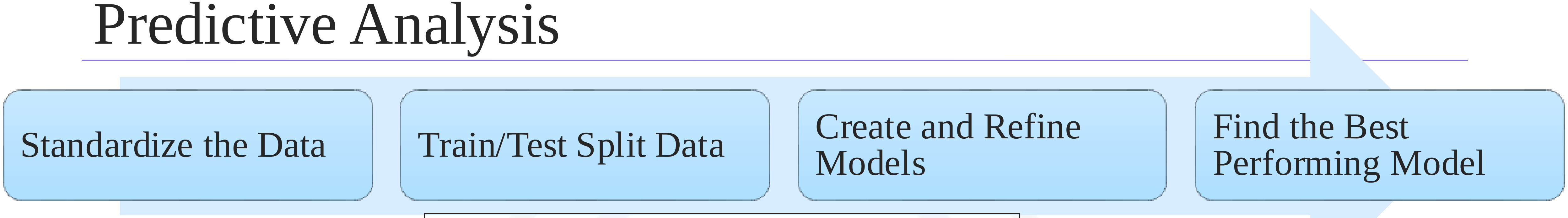
1. **Scatter Plot:**
   - Shows the relationship between two variables, making patterns easier to see, as shown in the following charts:
     - Relationship between Flight Number and Launch Site
     - Relationship between Payload and Launch Site
     - Relationship between Flight Number and Orbit Type
     - Relationship between Payload and Orbit Type
2. **Line Chart:**
   - Line charts are commonly used to track changes and show trends over time. The following line chart was plotted to observe this:
     - Average launch success yearly trend
3. **Bar Chart:**
   - Bar charts are used to compare values at a specific time, showing which groups are the highest and how they compare. The length of each bar shows its value. The following bar chart was created to visualize this:
     - Relationship between success rate of each orbit type

- In the Exploratory Data Analysis (EDA) process, feature engineering was performed by creating dummy variables for categorical columns.

Build an Interactive Map with Folium

- The Folium interactive map helps analyze geospatial data, making it easier to visualize and understand factors like the location and distance of launch sites that influence the success rate of launches.
- Map Features Created:
  - Marked all launch sites on the map, visually showing their locations.
  - Used ‘folium.Circle’ and ‘folium.Marker’ to highlight the area with a text label over each launch site.
  - Added ‘MarkerCluster()’ to show launch success (green) and failure (red) markers for each site.
  - Calculated distances between launch sites and nearby features (e.g., coastline, railway, highway, city).
  - Integrated ‘MousePosition()’ to get coordinates when hovering over the map.
  - Added ‘folium.Marker()’ to display the distance (in km) to the point of interest (e.g., coastline, railway, highway, city).
  - Used ‘folium.Polyline()’ to draw lines between the launch sites and their proximities.
  - Repeated these steps to add markers and lines for each proximity (coastline, railway, highway, city).
- Key Findings:
  - Are launch sites near railways? YES
  - Are launch sites near highways? YES
  - Are launch sites near coastlines? YES
  - Do launch sites maintain a safe distance from cities? YES

Predictive Analysis



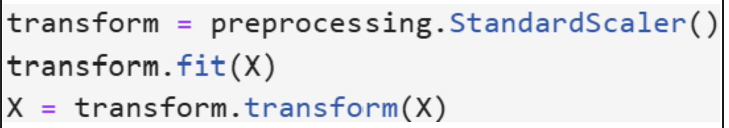
Standardize the Data

Train/Test Split Data

Create and Refine Models

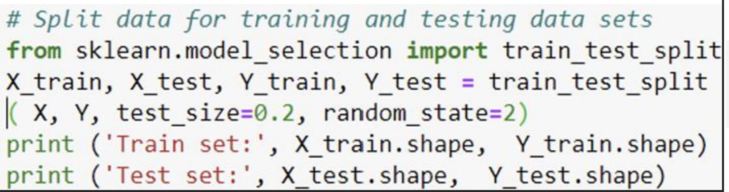
Find the Best Performing Model

1. Define a standard scaler and standardize the Data



```
transform = preprocessing.StandardScaler()
transform.fit(X)
X = transform.transform(X)
```

2. Using train test function separate data to train and test data

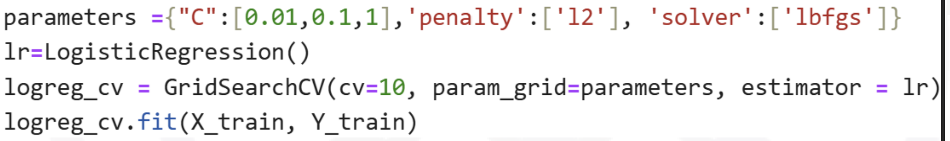


```
# Split data for training and testing data sets
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split
( X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

3. Create and refine classification models :

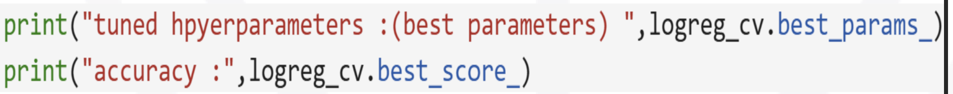
i. Create a logistic regression model with a search space

ii. Fit the data using GridSearchCV object



```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
lr=LogisticRegression()
logreg_cv = GridSearchCV(cv=10, param_grid=parameters, estimator = lr)
logreg_cv.fit(X_train, Y_train)
```

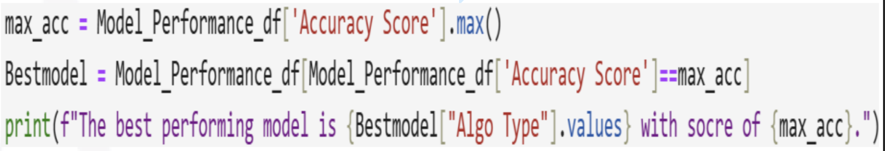
iii. Find the best combination of hyperparameters with highest accuracy



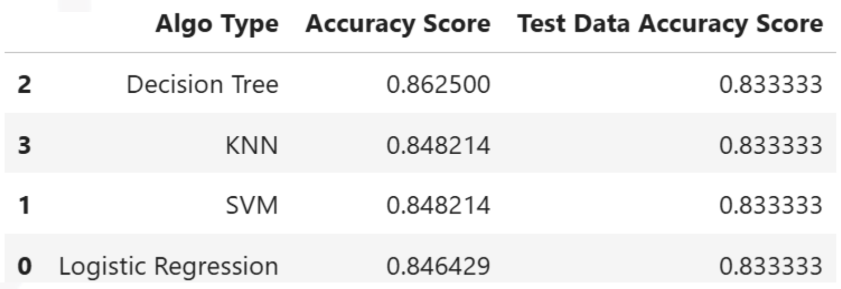
```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

iv. Repeat above steps other classification models

4. Find the best performing model



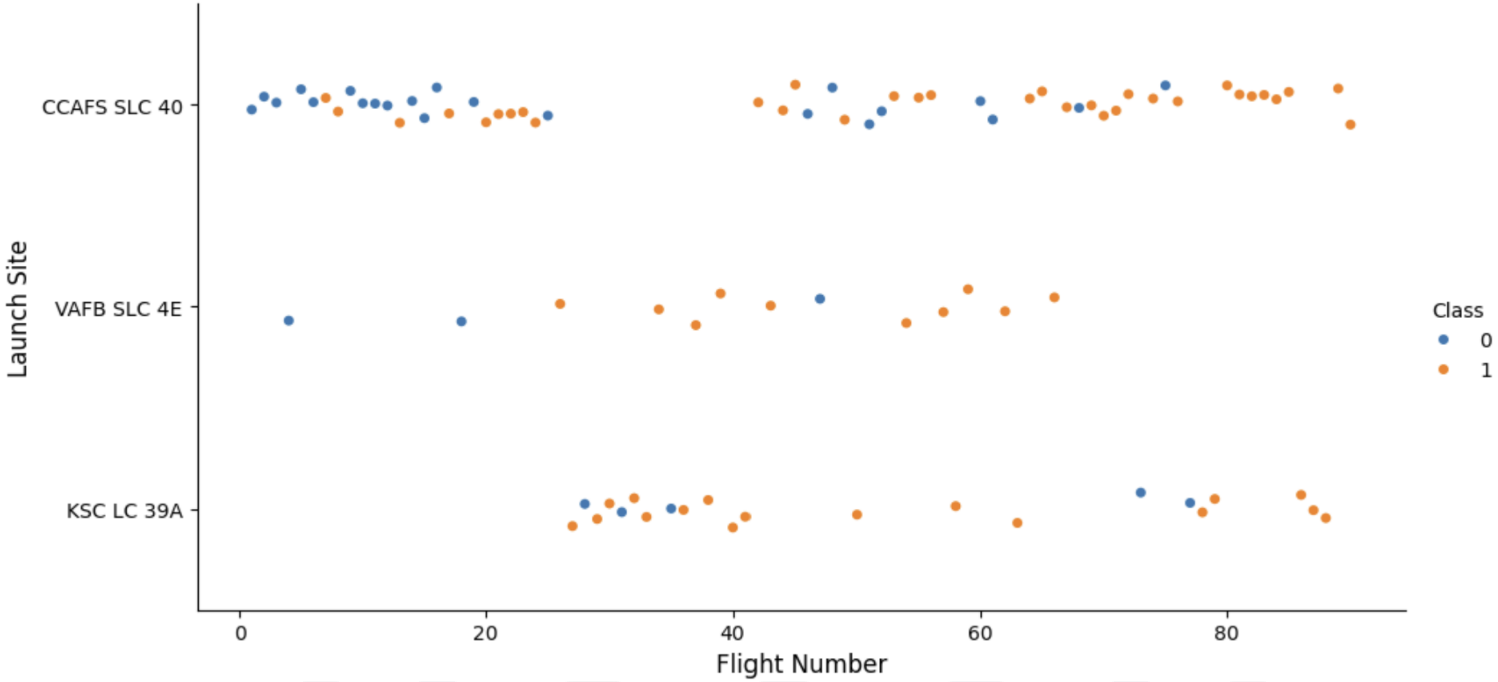
```
max_acc = Model_Performance_df['Accuracy Score'].max()
Bestmodel = Model_Performance_df[Model_Performance_df['Accuracy Score']==max_acc]
print(f"The best performing model is {Bestmodel["Algo Type"].values} with socre of {max_acc}.")
```



	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.862500	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

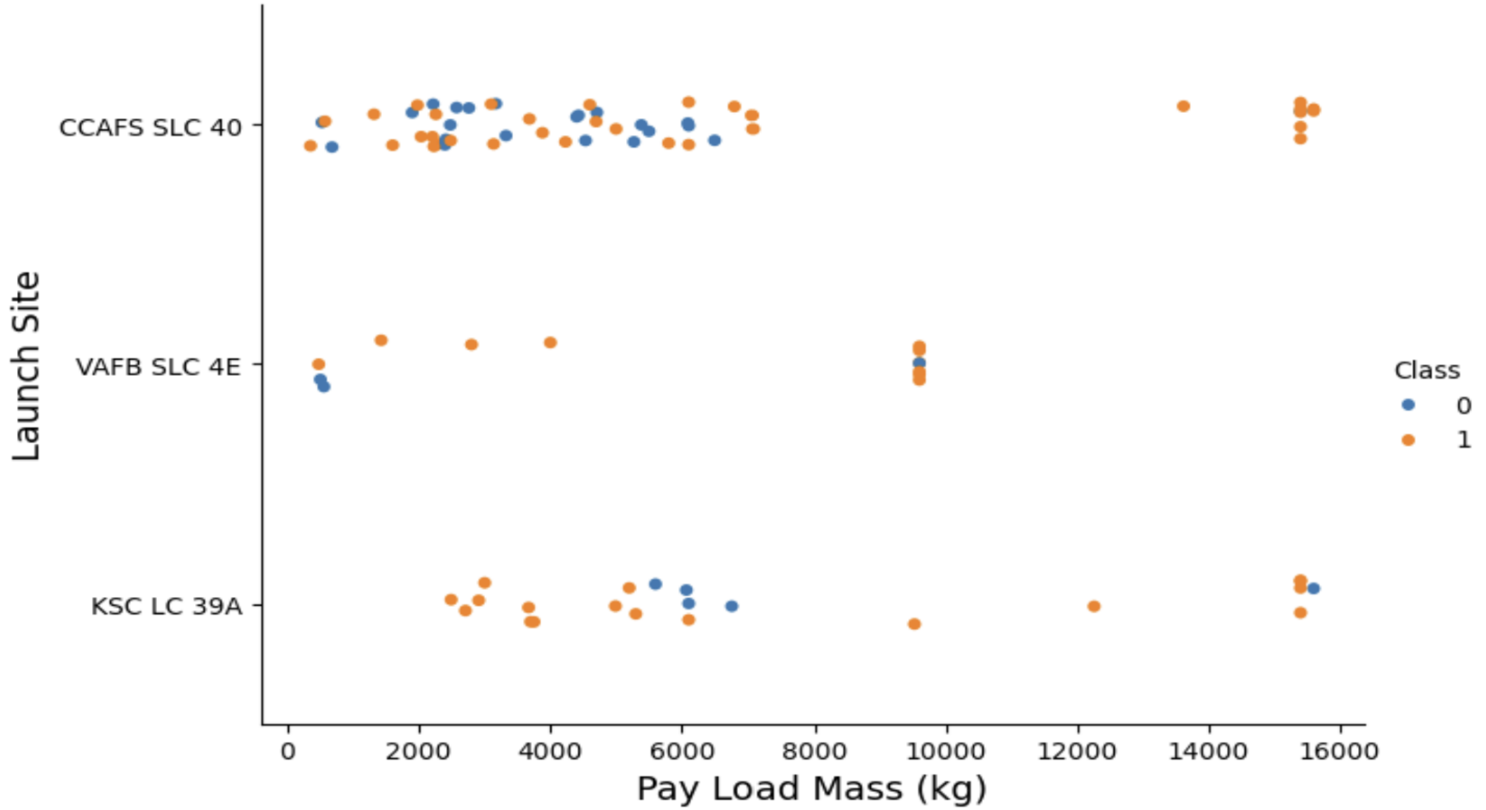
Insights from Exploratory Data Analysis

Flight Number vs. Launch Site



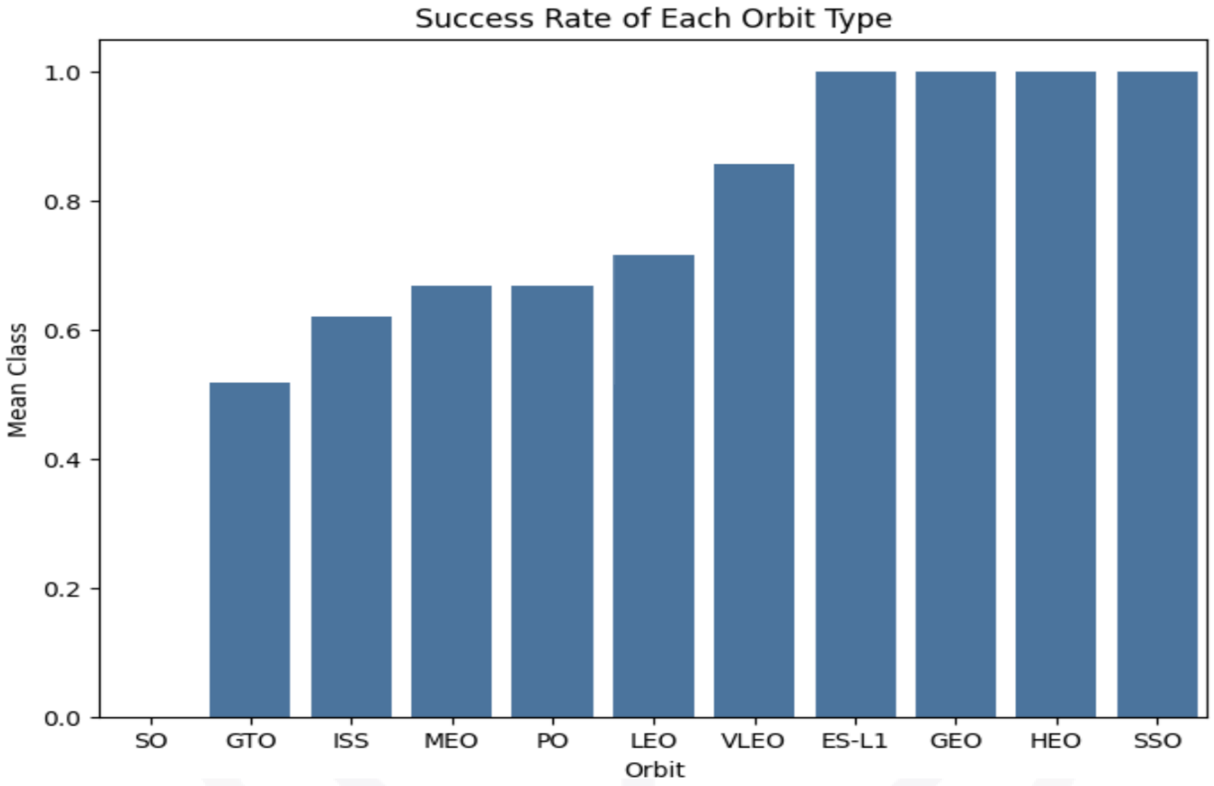
- As flight numbers rise, the proportion of successful launches increases, reflecting improvements in launch processes.
- CCAFS SLC 40 experienced more unsuccessful launches early on, while KSC LC 39A and VAFB SLC 4E showed a more balanced success rate, with increasing success over time.

Payload vs. Launch Site



- •Larger payloads are more likely to have successful launches (Class 1).
- •Smaller payloads are more likely to result in unsuccessful launches (Class 0), especially at sites like VAFB SLC 4E.

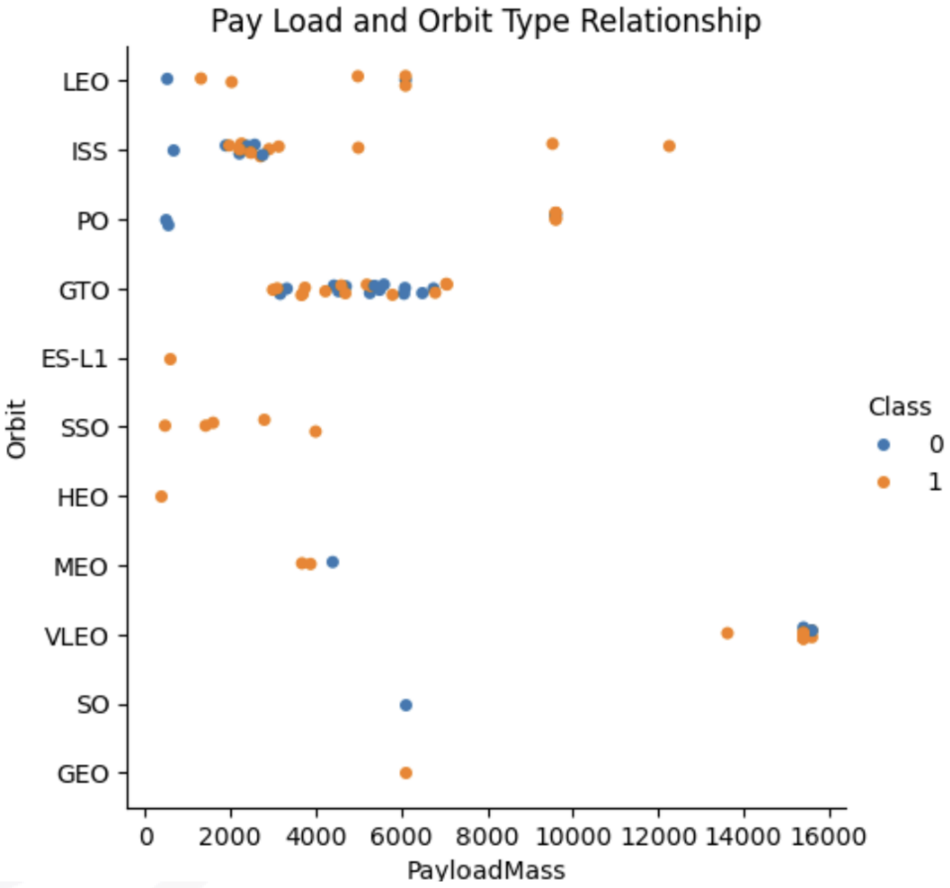
Success Rate vs. Orbit Type



- Orbit types like SSO, HEO, and GEO have the highest success rates.
- Orbit SO has the lowest success rates in all orbit types.

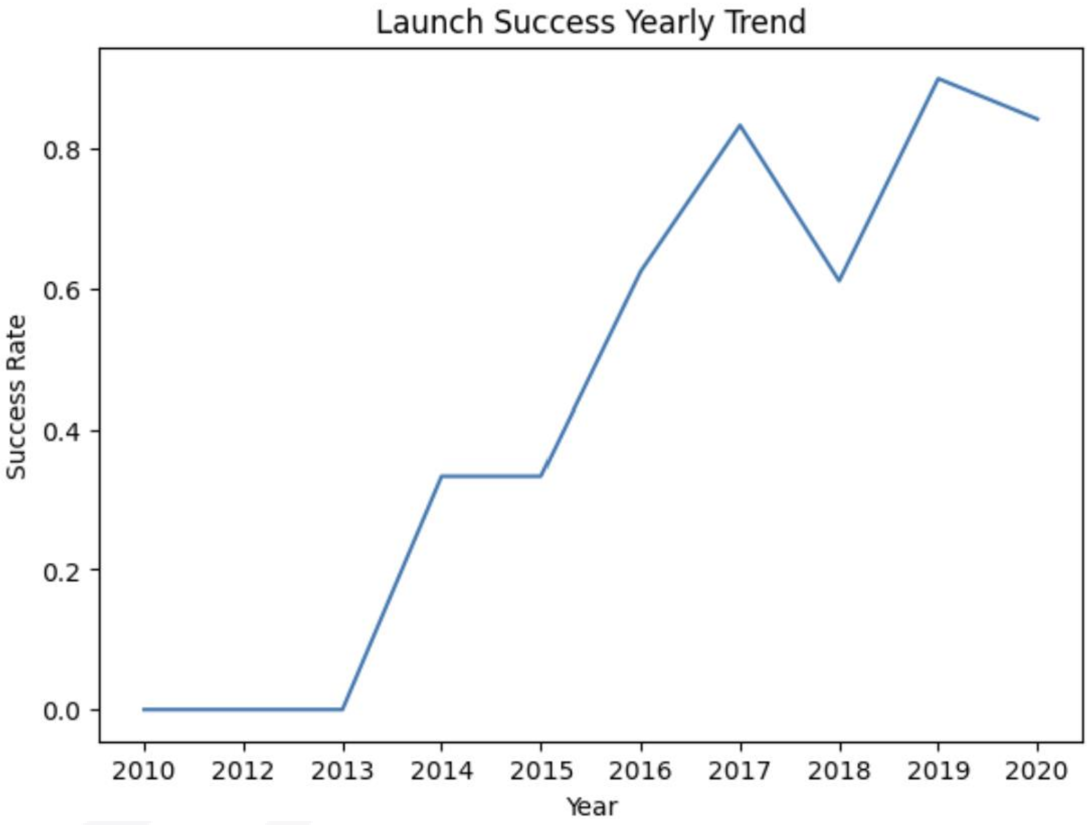
Payload vs. Orbit Type

- SSO has 100% success with small payloads.
- Heavier payloads lead to more successful launches for PO, LEO, and ISS orbits.
- GTO shows both successful and unsuccessful launches, with no clear pattern based on payload size.



Launch Success Yearly Trend

- The success rate (Class=1) grew by approximately 80% from 2013 to 2020.
- The success rates stayed constant from 2010 to 2013 and from 2014 to 2015.
- There was a decline in success rates between 2017 and 2018, as well as between 2019 and 2020.



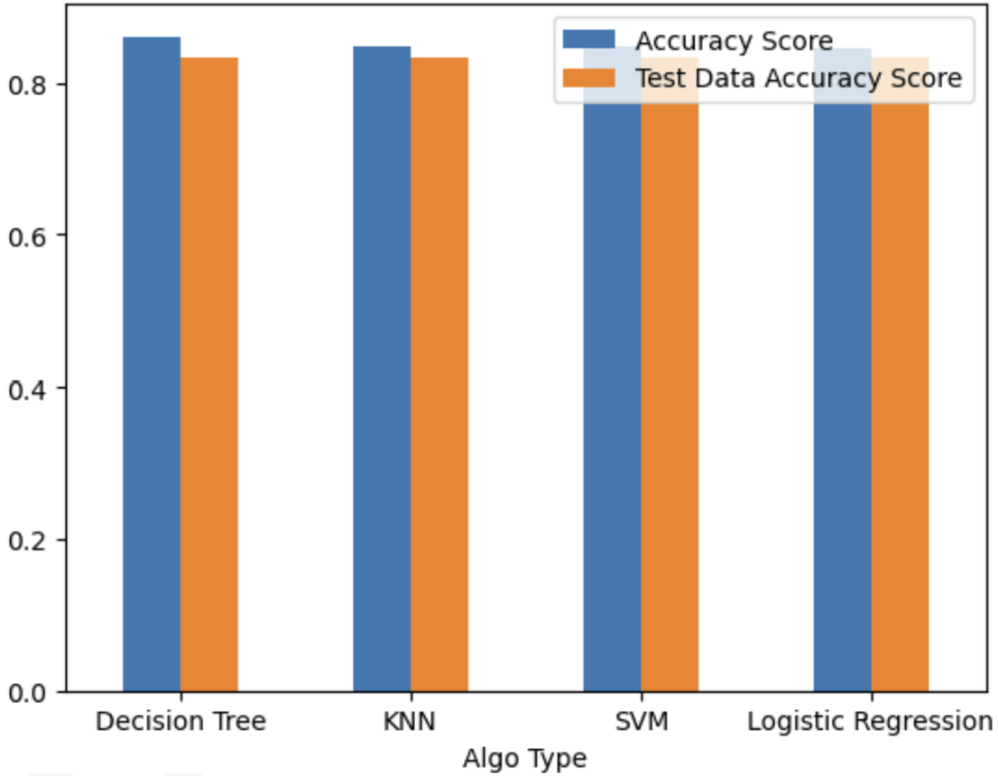
Classification Result

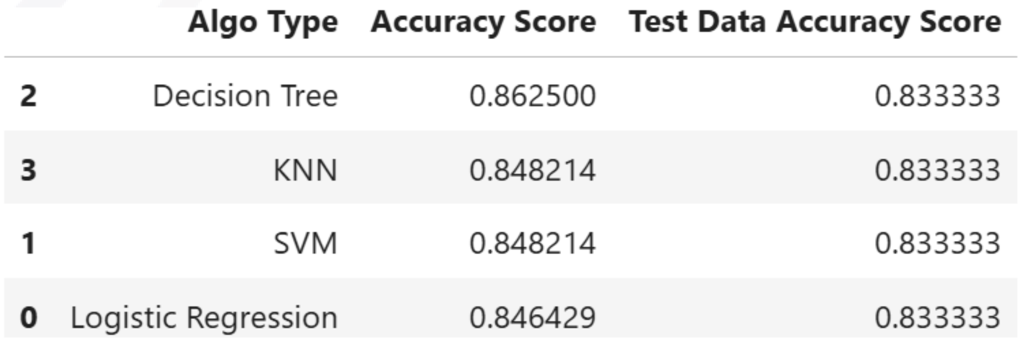
Classification Accuracy

- The Decision Tree algorithm performs the best with an accuracy score of 0.8750, showing its superior classification ability.

- All models, including KNN, SVM, and Logistic Regression, have the same test accuracy of 0.8333, indicating similar performance.

- Since the models show similar results, a larger dataset could help improve performance and allow for further tuning.

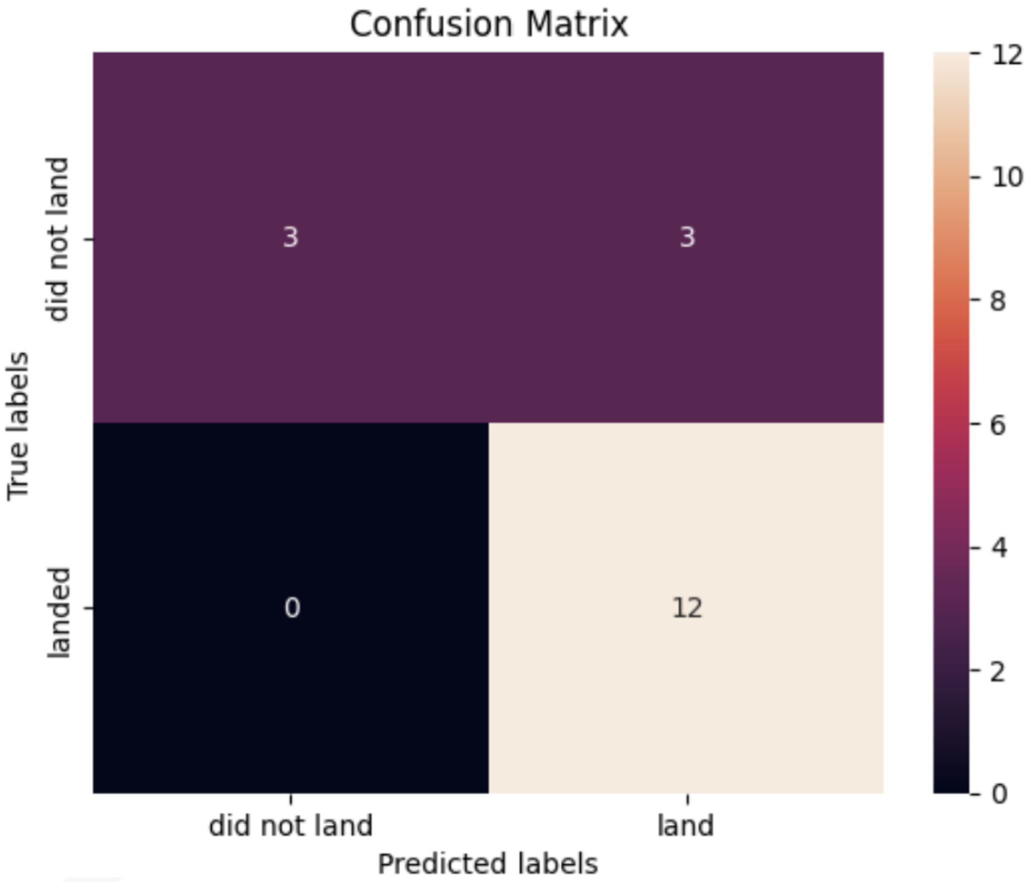




	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.862500	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

Confusion Matrix

- The confusion matrix is the same for all models (Logistic Regression, SVM, Decision Tree, and KNN).
- Based on the confusion matrix, the classifier made 18 predictions in total.
- 12 scenarios were predicted as "Yes" for landing, and they did land successfully (True positives).
- 3 scenarios (top left) were predicted as "No" for landing, and they did not land (True negatives).
- 3 scenarios (top right) were predicted as "Yes" for landing, but they did not land successfully (False positives).
- Overall, the classifier is correct about 83% of the time, with a misclassification rate of 16.5% (calculated as (False positives + False negatives) / Total).



Conclusion

- Payload mass shows an upward trend in success rates, but there is no clear correlation between payload size and success.
- Launch success rates increased by about 80% from 2013 to 2020, indicating improvements over time.
- KSC LC-39A has the highest launch success rate, while CCAFS SLC 40 has the lowest.
- As flight numbers increase, the proportion of successful launches rises, reflecting improvements in launch processes.
- CCAFS SLC 40 had more unsuccessful launches early on, while KSC LC 39A and VAFB SLC 4E showed a more balanced success rate, with increasing success over time.
- Orbits SSO, HEO, and GEO have the highest success rates, while Orbit SO has the lowest.
- The Decision Tree model performs best with an accuracy of 86.25%, but all models showed similar test accuracy (83.3%), suggesting that more data is needed for further improvement.