IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Narendra Kandel
31/03/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

Project Context

SpaceX is the most successful commercial company by making space travel affordable for everyone. One of the main reasons that SpaceX can do this is the rocket launches are relatively inexpensive because SpaceX can reuse the first.Unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage in the most of the cases. Spaces X’s Falcon 9 launches like regular rockets. We will predict if the Falcon 9 first stage will land successfully. Stage two, or the second stage, helps bring the payload to orbit, but most of the work is done by the first stage.

Questions that need to be addressed:

What factors are useful in determining if a rocket will successfully land?

Which correlation can be obtained from these factors to influence these successes?

What operating conditions needs to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from SpaceX API and Wikipedia by using python web scraping.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was gathered using get request to the SpaceX API.
- Web-scraping for Falcon 9 launch records was done by using of BeautifulSoup API.
- Data was processed as a json and data frame through utilizing .json() function and .json_normalize().
- Data was cleaned, verified for missing variables, and NULL values were dealth with inside the PayloadMass by replacing with the calculated mean of the PayloadMass data.
- The NULL values of LandingPad is dealt with using one hot encoding.

Data Collection – SpaceX API

- Get request for rocket launch data using API
- Use json_normalize method to convert json result to dataframe
- Perform data cleansing and fill missing value

https://github.com/narendrakandel/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from URL
- Create a BeautifulSoup object from the HTML response
- Extract all columns and variable names from the HTML reader

https://github.com/narendrakandel/Applied-Data-Science-Capstone/commit/87ef66d43b2c235a1adf9f77b8b0c128d7122a66

Data Wrangling

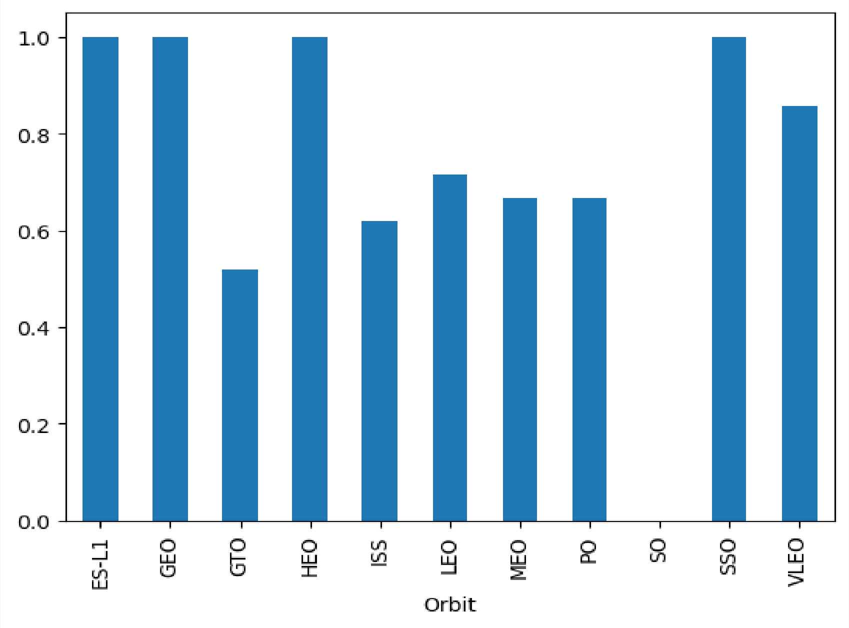
- Calculate the mean for the PayloadMass using the .means()
- Use the mean and the .replace() function to replace np.nan values in the data with the mean

https://github.com/narendrakandel/Applied-Data-Science-Capstone/blob/main/labs-jupyterspacexData%20wrangling.ipynb

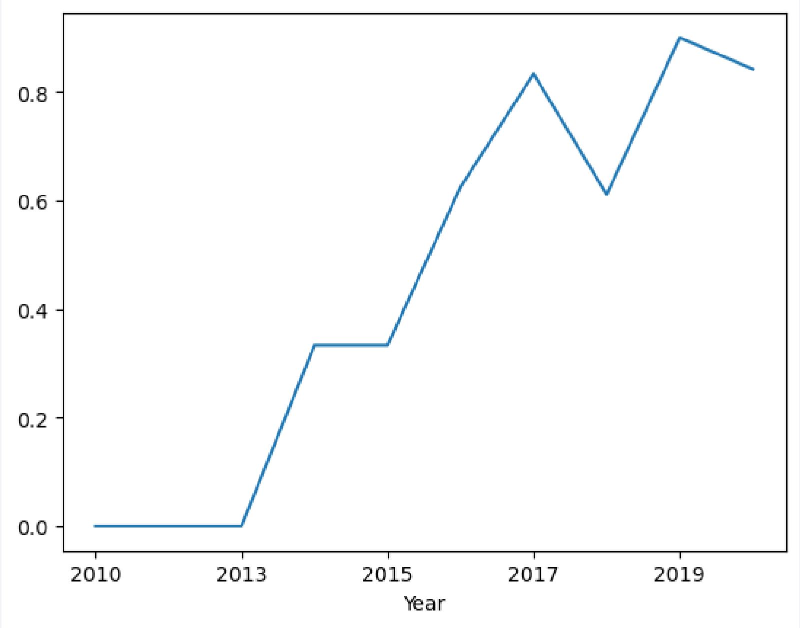
EDA with Data Visualization

Exploration of the data showing the relationship between flight number/ launch Site, payload/launch site, success rate of each orbit type, flight number/orbit type, the launch successes per year.

Success Rates by Orbit Types



Yearly Success Trends



EDA with SQL

The following queries were executed for EDA with SQL

- The names of all distinct launch sites in the space mission.
- The total payload mass carried by boosters launched by NASA
- The average payload mass carried by the specified booster model: F9 v1.1
- The breakdown of the successful and failed mission outcomes
- Failed landings on drone ship; along with their booster version and location of launch.

Build an Interactive Map with Folium

To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.

https://github.com/narendrakandel/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location%20(2).ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload
- Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

- We build a machine learning pipeline to predict the first stage of the Falcon 9, which include: Preprocessing, allowing us to standardize our data, and Train_test_split and allowing us to split our data into training and testing data.
- We then train the model and find the hyperparameters that allow a given algorithm to perform best.
- With the best hyperparameter values, we determine the model with the best accuracy using the training data.
- Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors tests are carried out and finally, the confusion matrix are produced to find the best model with highest accuracy score.

https://github.com/narendrakandel/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

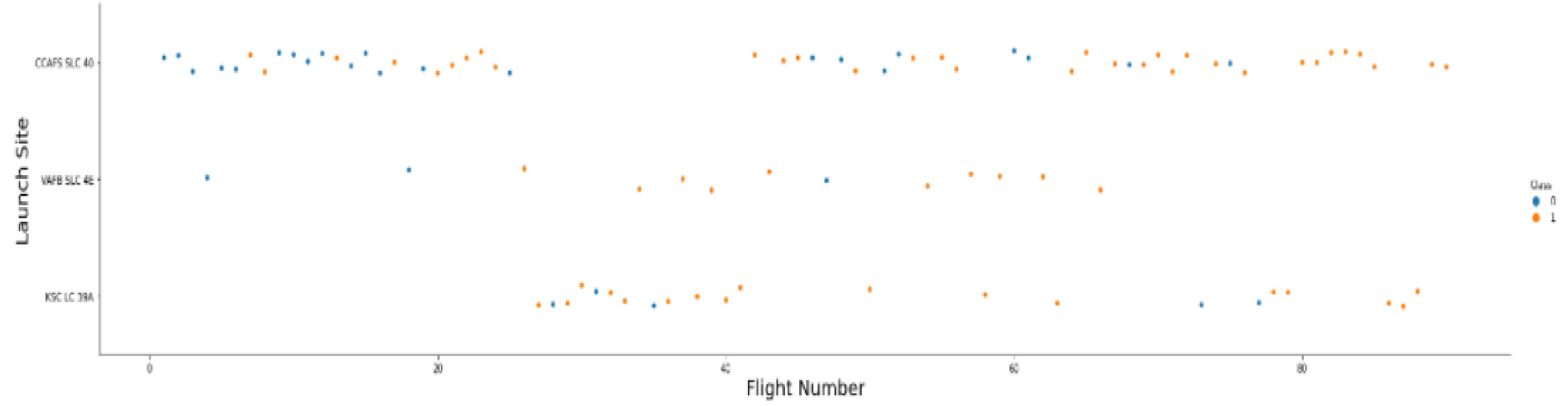
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

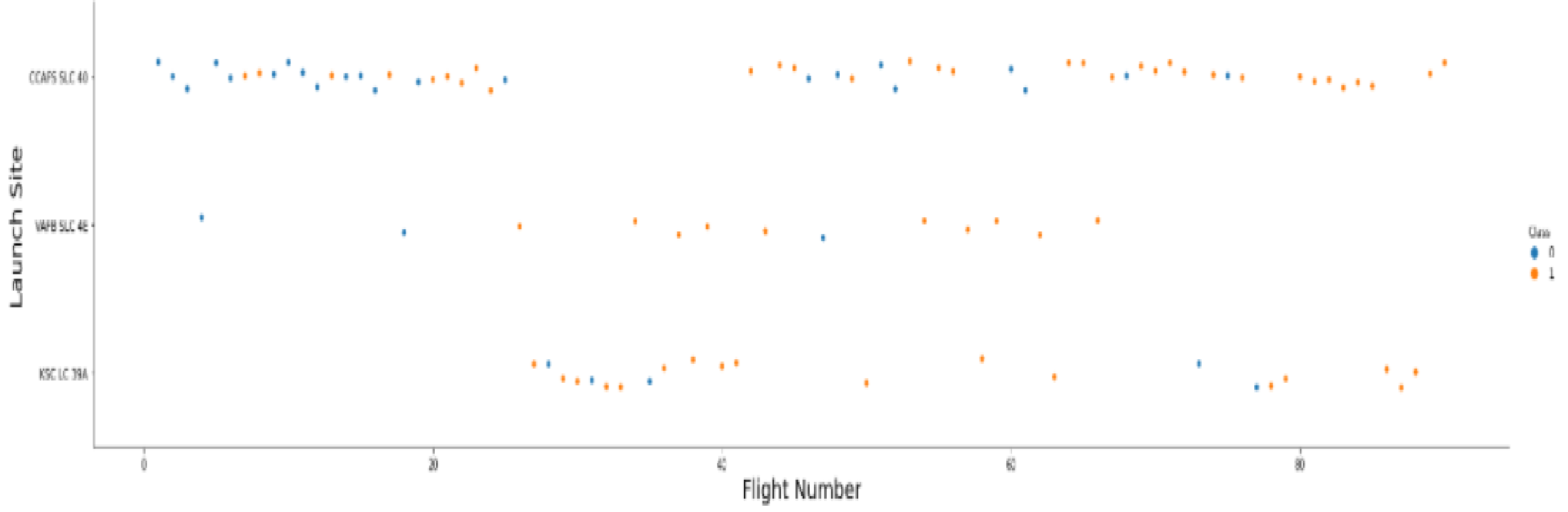
Insights drawn from EDA

Flight Number vs. Launch Site



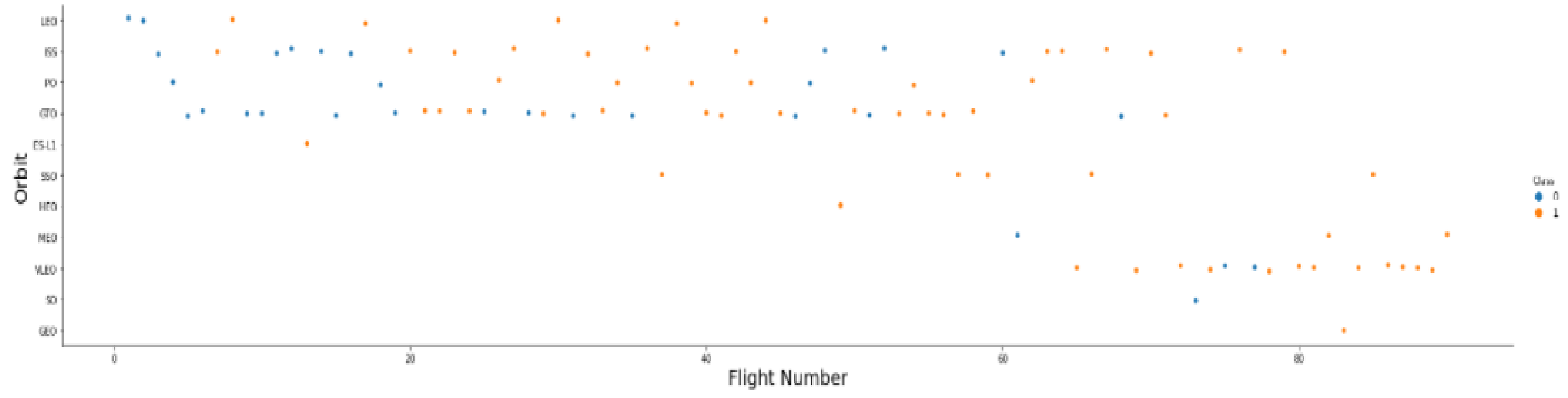
The findings is that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site



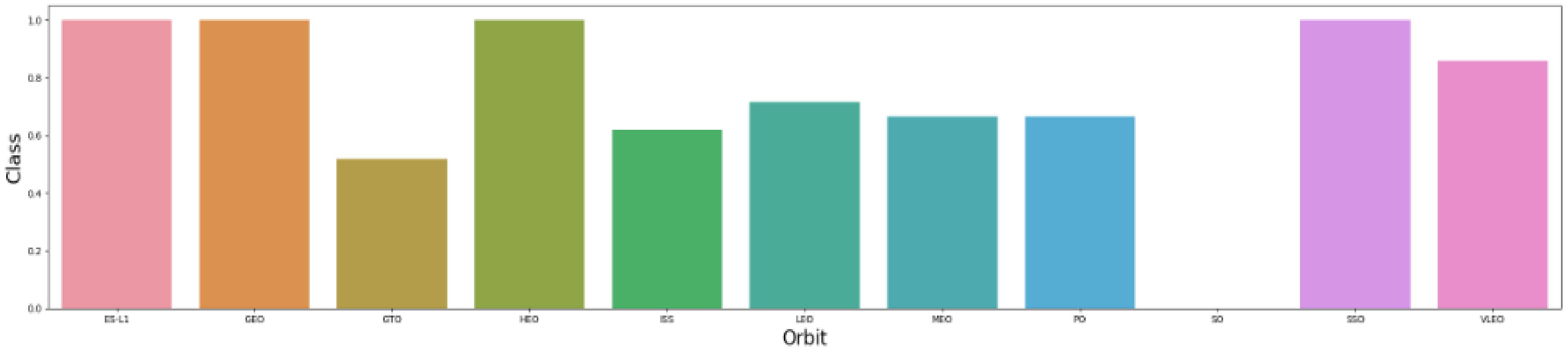
The findings is that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Flight Number vs. Orbit Type



The findings is that success is related to the number of flights in the LEO orbit. However, there is no relationship between flight number and orbit for the GTO.

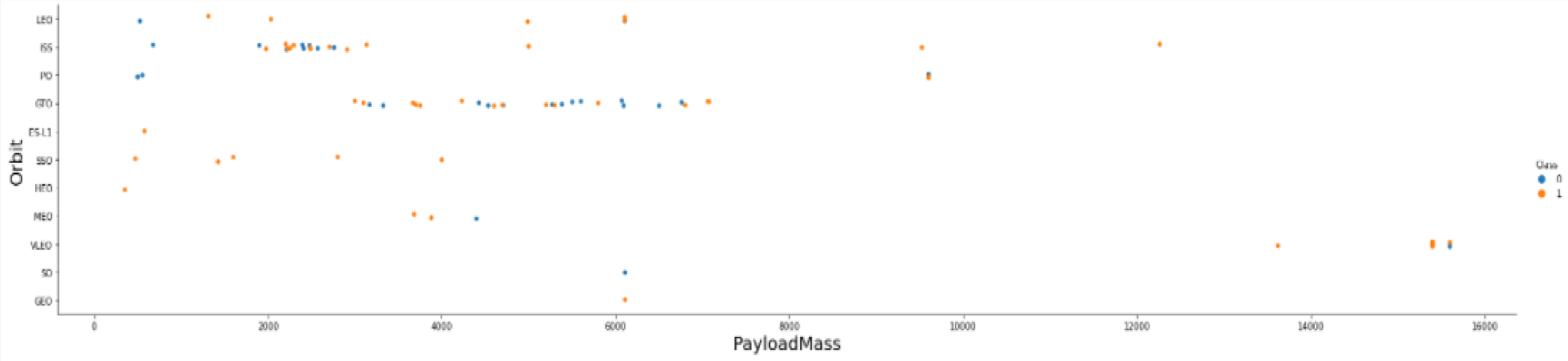
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

The findings is that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

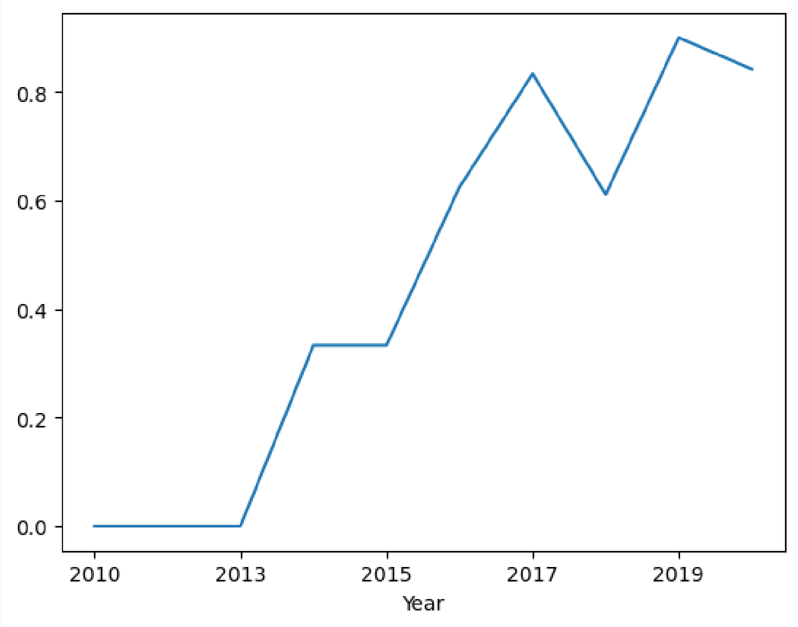
Payload vs. Orbit Type



As the payloads increase, the successful landing are more likely for PO, LEO and ISS orbits

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



All Launch Site Names

- Find the names of the unique launch sites

  %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' returns first 5 unique row only

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

%sql select min(date) from SPACEXTBL where Landing _Outcome = 'Success (ground pad)'

	firstsuccessfull_landing_date
0	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

%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME ='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

%sql Select Mission_Outcome, count(Mission_Outcome) as count from SPACEXTBL group by Mission_Outcome

List the total number of successful and failure mission outcomes

```
%sql Select MISSION_OUTCOME,count(MISSION_OUTCOME) as count from SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

Mission_Outcome	count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

```
%sql select distinct BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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 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 SPACEXTBL where YEAR(DATE) = '2015' and Landing _Outcome = 'Failure (drone ship)‘

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	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

landingoutcome	count
No attempt	10
Success (drone ship)	6
Failure (drone ship)	5
Success (ground pad)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

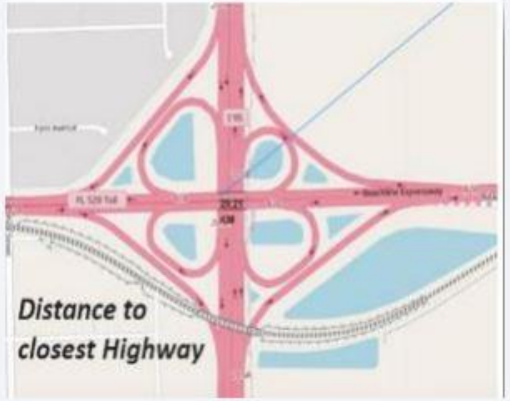
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

<Folium Map Screenshot 2>

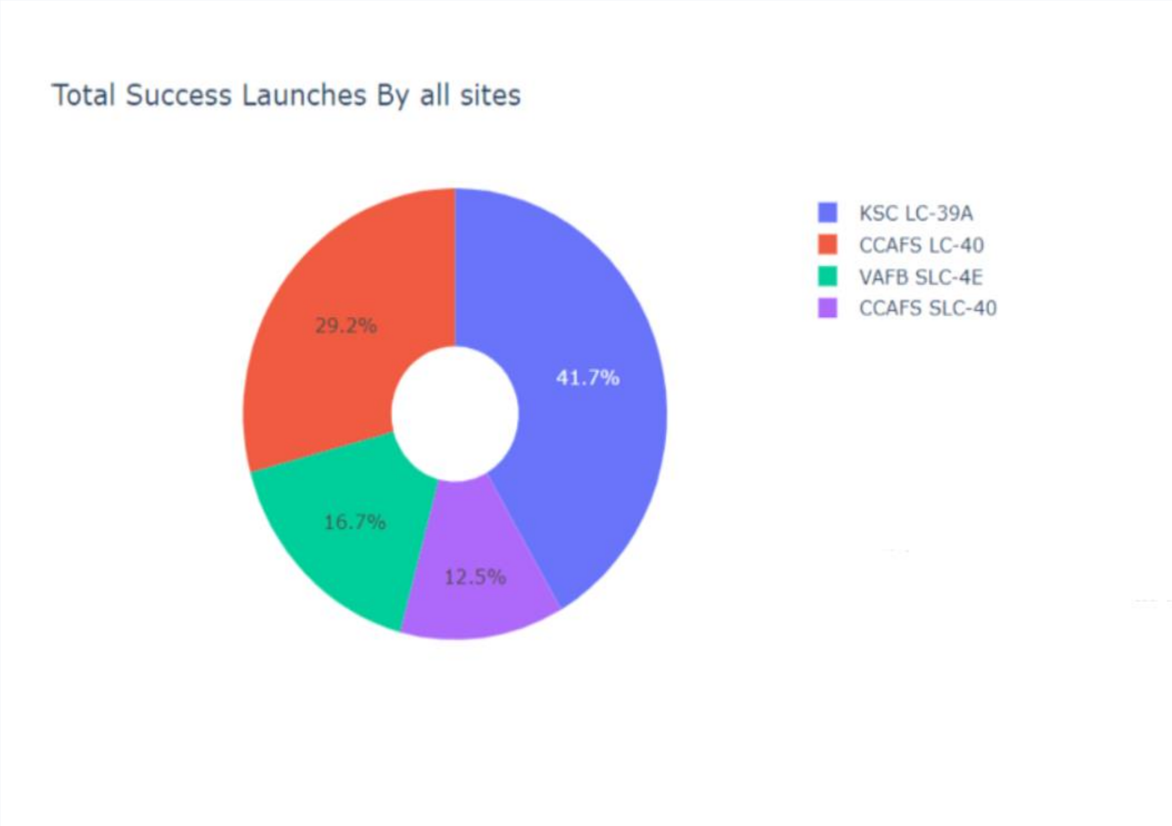
<Folium Map Screenshot 3>



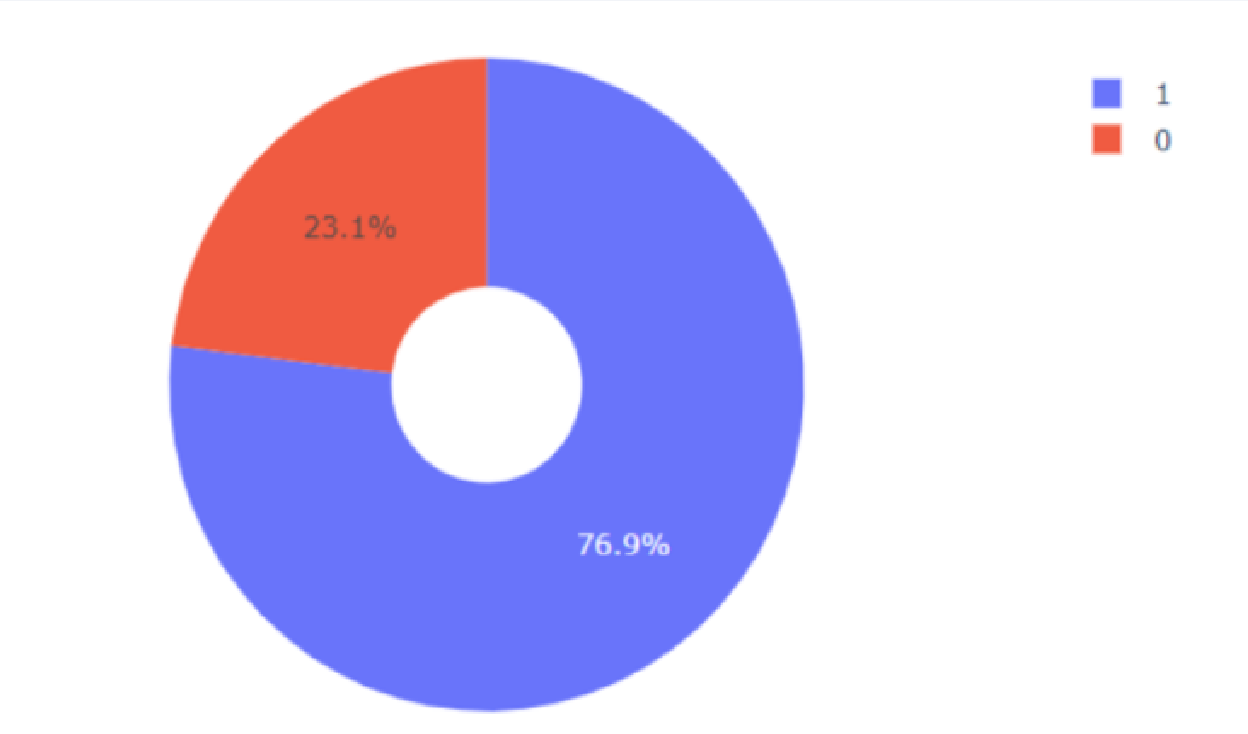
Section 4

Build a Dashboard with Plotly Dash

The success percentage by each sites

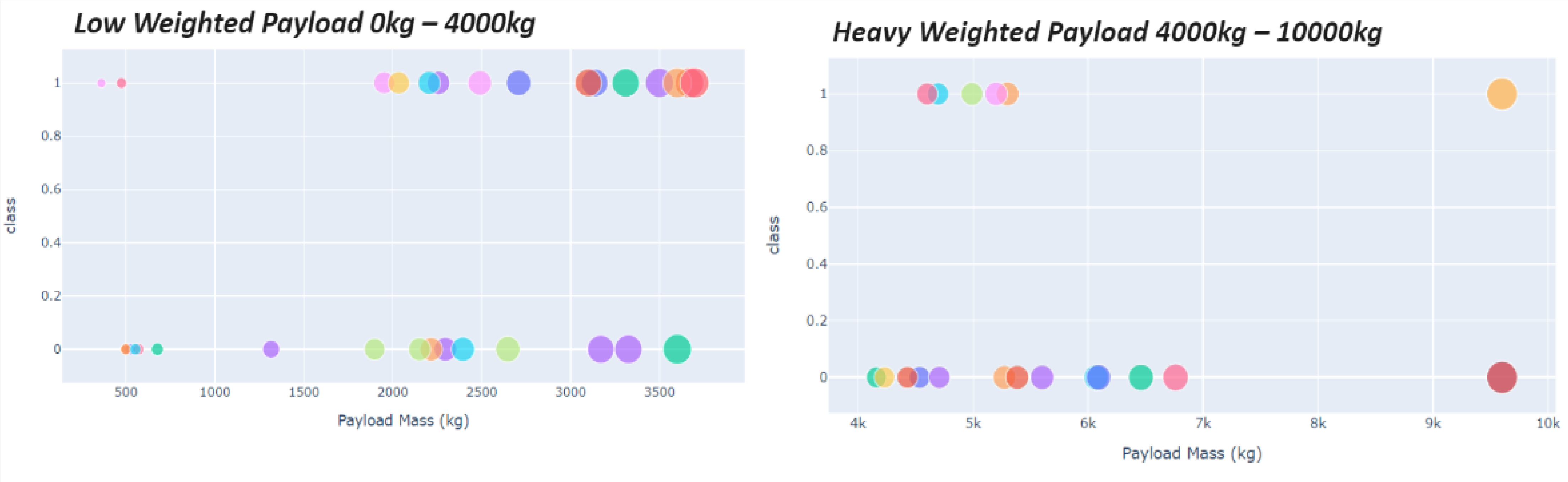


The highest launch-success ratio: KSC LC-39A



Payload and Success Rate

Heavier the payloads, largest success rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

the best algorithm to be the Tree Algorithm which have the highest classification accuracy

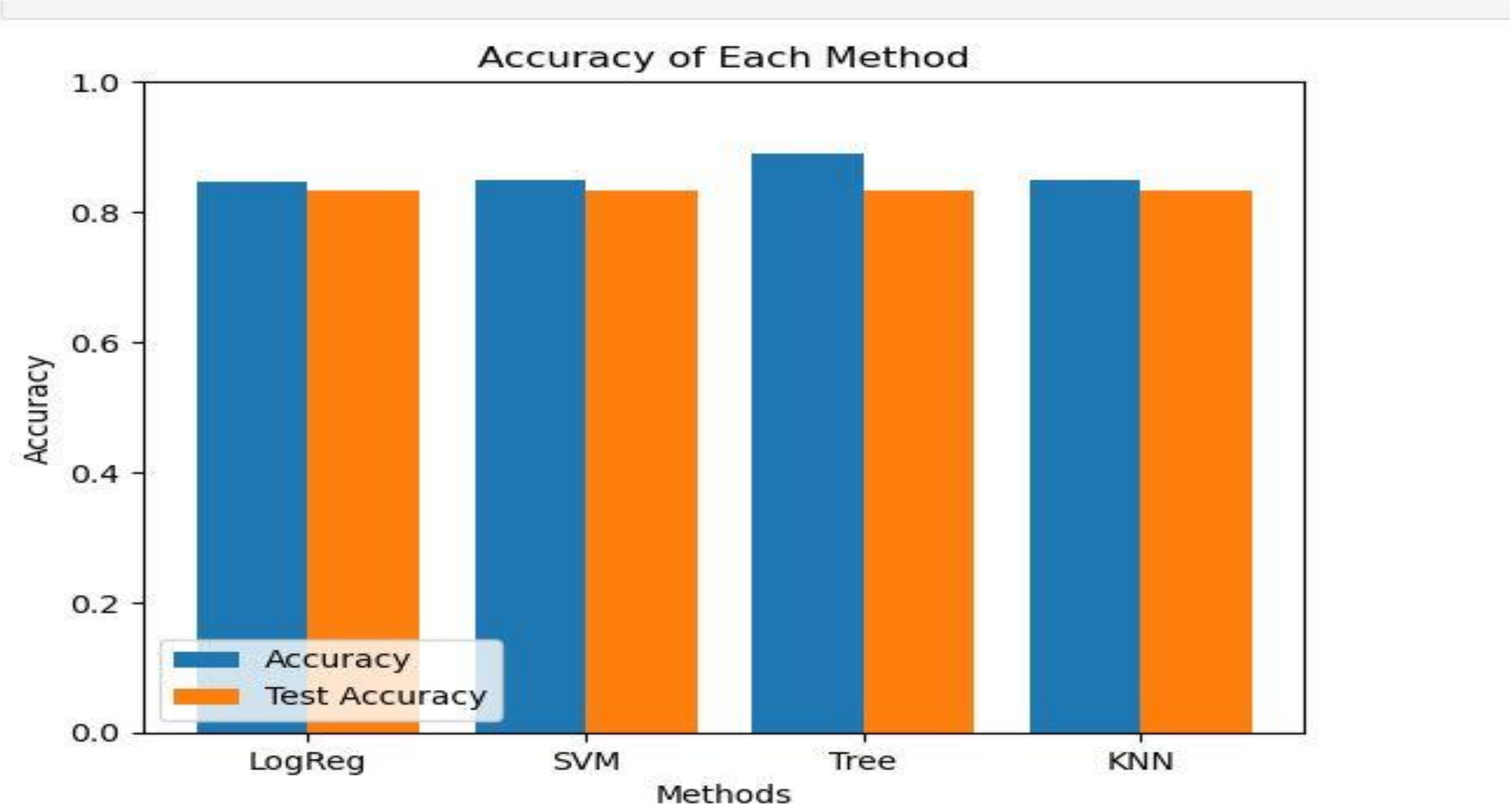
Find the method performs best:

```
print("Model\t\tAccuracy\tTestAccuracy")#,logreg_cv.best_score_)
print("LogReg\t\t{}\t\t{}".format((logreg_cv.best_score_).round(5), logreg_cv.score(X_test, Y_test).round(5)))
print("SVM\t\t{}\t\t{}".format((svm_cv.best_score_).round(5), svm_cv.score(X_test, Y_test).round(5)))
print("Tree\t\t{}\t\t{}".format((tree_cv.best_score_).round(5), tree_cv.score(X_test, Y_test).round(5)))
print("KNN\t\t{}\t\t{}".format((knn_cv.best_score_).round(5), knn_cv.score(X_test, Y_test).round(5)))

comparison = {}

comparison['LogReg'] = {'Accuracy': logreg_cv.best_score_.round(5), 'TestAccuracy': logreg_cv.score(X_test, Y_test).round(5)}
comparison['SVM'] = {'Accuracy': svm_cv.best_score_.round(5), 'TestAccuracy': svm_cv.score(X_test, Y_test).round(5)}
comparison['Tree'] = {'Accuracy': tree_cv.best_score_.round(5), 'TestAccuracy': tree_cv.score(X_test, Y_test).round(5)}
comparison['KNN'] = {'Accuracy': knn_cv.best_score_.round(5), 'TestAccuracy': knn_cv.score(X_test, Y_test).round(5)}
```

```
Model           Accuracy        TestAccuracy
LogReg          0.84643         0.83333
SVM             0.84821         0.83333
Tree            0.88929         0.83333
KNN             0.84821         0.83333
```

Classification Accuracy

the best algorithm to be the Tree Algorithm which have the highest classification accuracy



Confusion Matrix

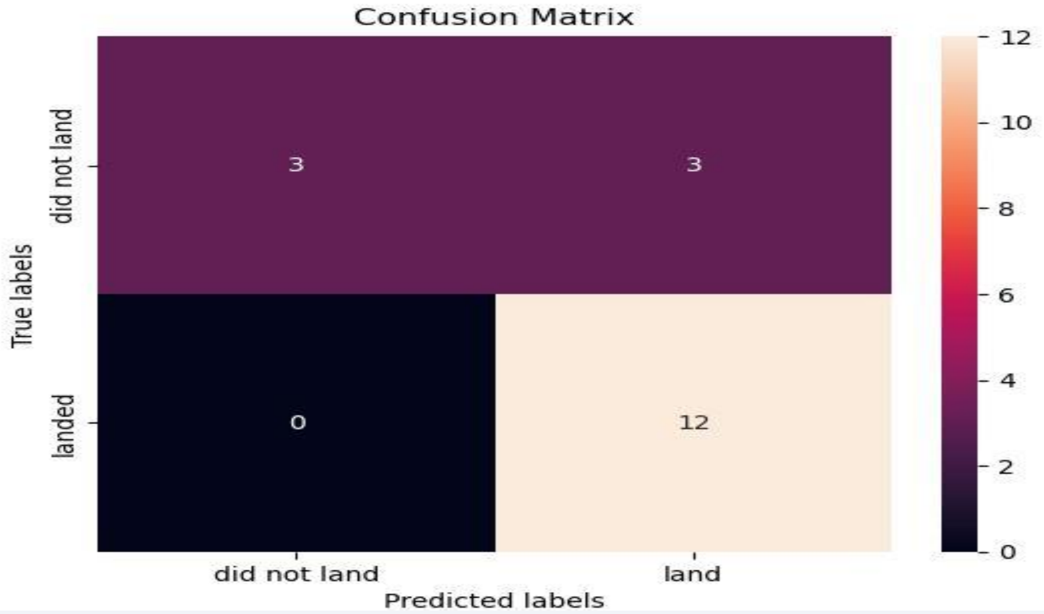
The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

```
[27]: print("accuracy :", tree_cv.score(X_test, Y_test))
```

```
accuracy : 0.8333333333333334
```

We can plot the confusion matrix

```
[28]: yhat = svm_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.

The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.

Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional to time in years to 2020, which it will eventually perfect the launches in the future.

KSC LC-39A have the most successful launches of any sites; 76.9%

SSO orbit have the most success rate; 100% and more than 1 occurrence.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!