Winning Space Race with Data Science

Prachi Choughule
15/06/2022

Outline

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

Executive Summary

- Collected data from public SpaceX API and SpaceX Wikipedia page. Created labels column 'class' which classifies successful landings. Explored data using SQL, visualization, folium maps, and dashboards. Gathered relevant columns to be used as features. Changed all categorical variables to binary using one hot encoding. Standardized data and used GridSearchCV to find best parameters for machine learning models. Visualize accuracy score of all models.
- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with accuracy rate of about 83.33%. All models over predicted successful landings. More data is needed for better model determination and accuracy.

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.

Space Y wants to compete with Space X. We will be using the information about Space X and create dashboards for the team.

We will train a machine learning model and use public information to find if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page.
- Perform data wrangling
  - Use one-hot encoding for categorical data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Model tuning using GridSearchCV

Data Collection

Data collection process involved a combination of API requests from SpaceX API and web scrapping data from the Wikipedia page.

SpaceX API

- Request (SpaceX API)
- Normalize json results into dataframe.
- Filter data to only include Falcon 9 launches.
- Impute missing data by calculating mean.

- Web Scraping
  - Request html page
  - Using BeautifulSoup parser parse html table.
  - Extract data by iterating on table elements.

Data Collection - SpaceX API

- https://github.com/prachisc/AppliedDSCapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- https://github.com/prachisc/AppliedDSCapstone/blob/main/Module10-Webscrapping.ipynb

- Perform HTTP Get request Wikipage to get html response object.
- Parse the reponse by creating BeautifulSoup object.
- Extract HTML header and table data.
- Create dictionary items from extracted data.
- Create dataframe from dictionary and then export it to be used along with SpaceX API

Data Wrangling

Create training label with landing outcomes where succuessful = 1 and failure = 0.

Outcome has two components ‘Mission Outcome’ and ‘Landing Location’

New training label column **class** with value of 1 if Mission Outcome is True and 0 otherwise.

Value Mapping:

True ASDS, True RTLS, & True Ocean - set to -> 1

None None, False ASDS, None ASDS, False Ocean, False RTLS - set to -> 0

https://github.com/prachisc/AppliedDSCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly.

This exploration was to find the correlated features that can then be used to create our machine learning model.

https://github.com/prachisc/AppliedDSCapstone/blob/main/EDA%20With%20Data%20Visualization.ipynb

Build an Interactive Map with Folium

Folium maps mark launch sites, successful and unsuccessful landings and proximity to key locations. E.g. Railways, Highways, Cost and city.

This allows us to understand why launch sites may be located where they are. Also visualize successful landings relative to location.

https://github.com/prachisc/AppliedDSCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The interactive dashboard created with Plotly Dash were pie charts aand scatter plot.
- Pie char shows the total launches and their status from all or particular launch site.
- Scatter plot helps us to see how launch success varies across launch sites, payload mass and booster version category.

https://github.com/prachisc/AppliedDSCapstone/blob/main/ploty_app.py

Predictive Analysis (Classification)

Various classification models were used to perform predictive analysis. The dataset was split into train and test set using sklearn library. The source data was standardized and split into training and test data. Same data was then used across various classification models and the accuracy was compared to find the best performing classification model for the source data.

https://github.com/prachisc/AppliedDSCapstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

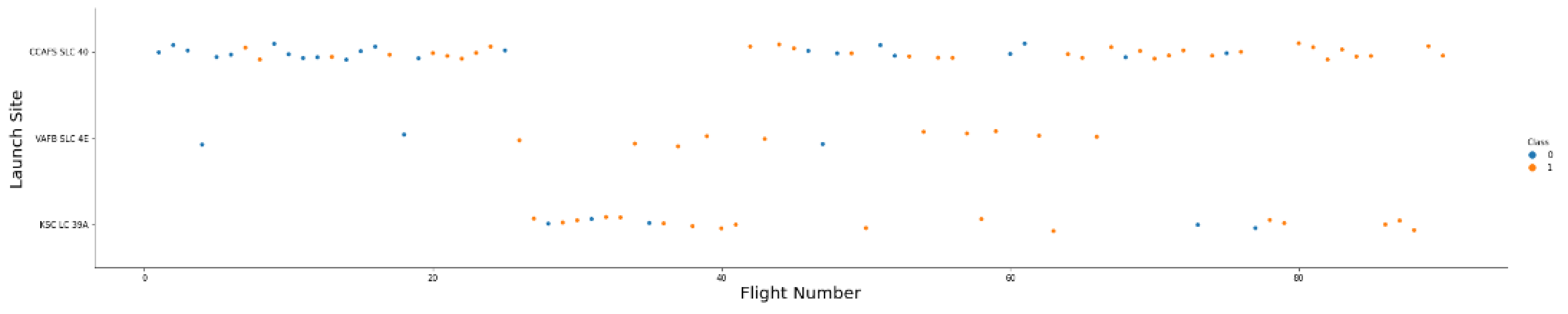
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

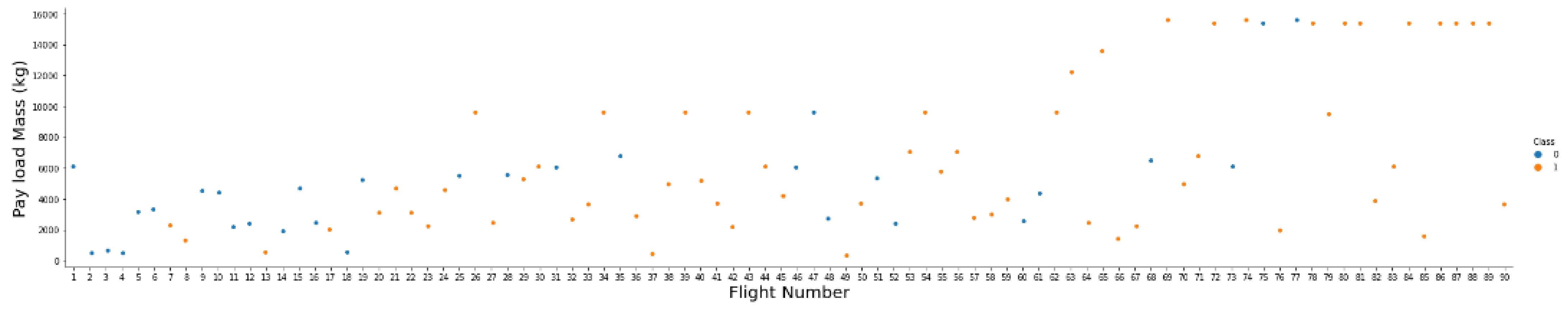
The scatter plot shows increase in success rate over time. After flight 20 success rate increases significantly.

CCAFS is the main launch site as there is increased volume from that launch site.

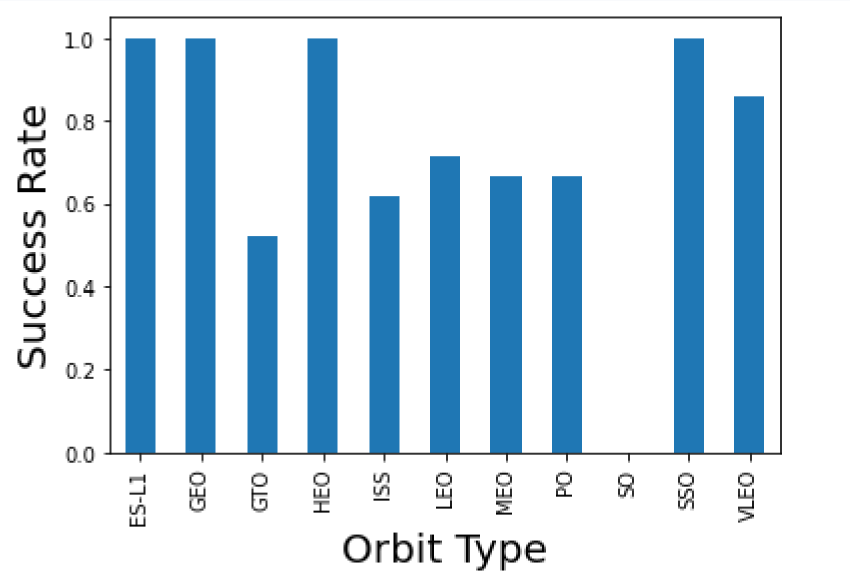


Payload vs. Launch Site

Similar to previous observations launch success rate increased over a time (after flight 20) across different paloads. Higher payload the success rate was much higher.



Success Rate vs. Orbit Type

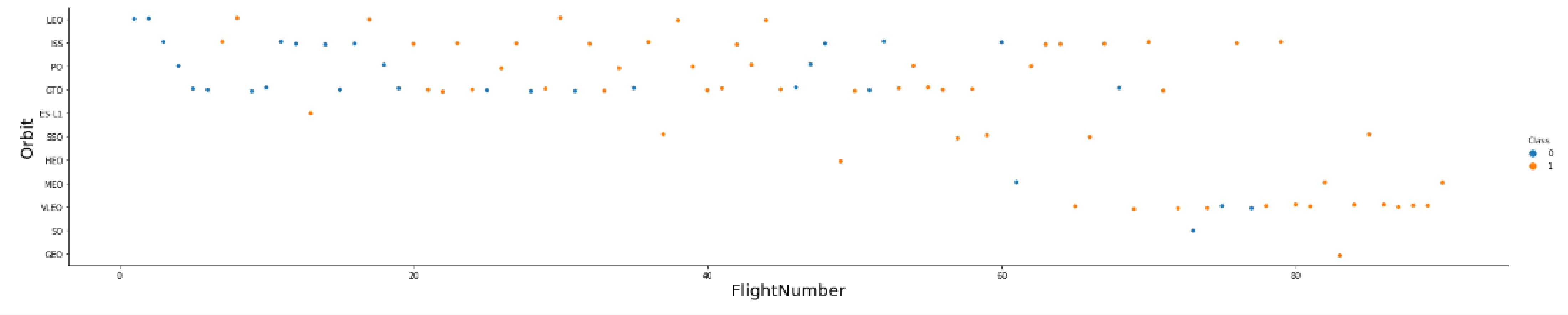


- ES-L1, GEO, HEO and SSO have 100% success rate
- VELO around 80%
- No details found for SO

Flight Number vs. Orbit Type

Success rate better after flight 20 for the same orbit types. LEO, ISS, PO, GTO.

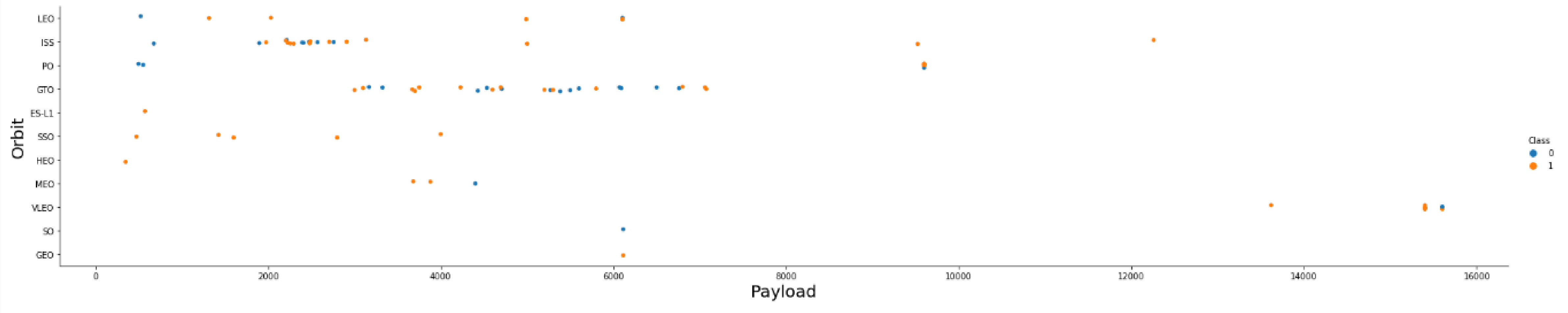
Can see more success in VELO orbit type when the launches in that orbit started.



Payload vs. Orbit Type

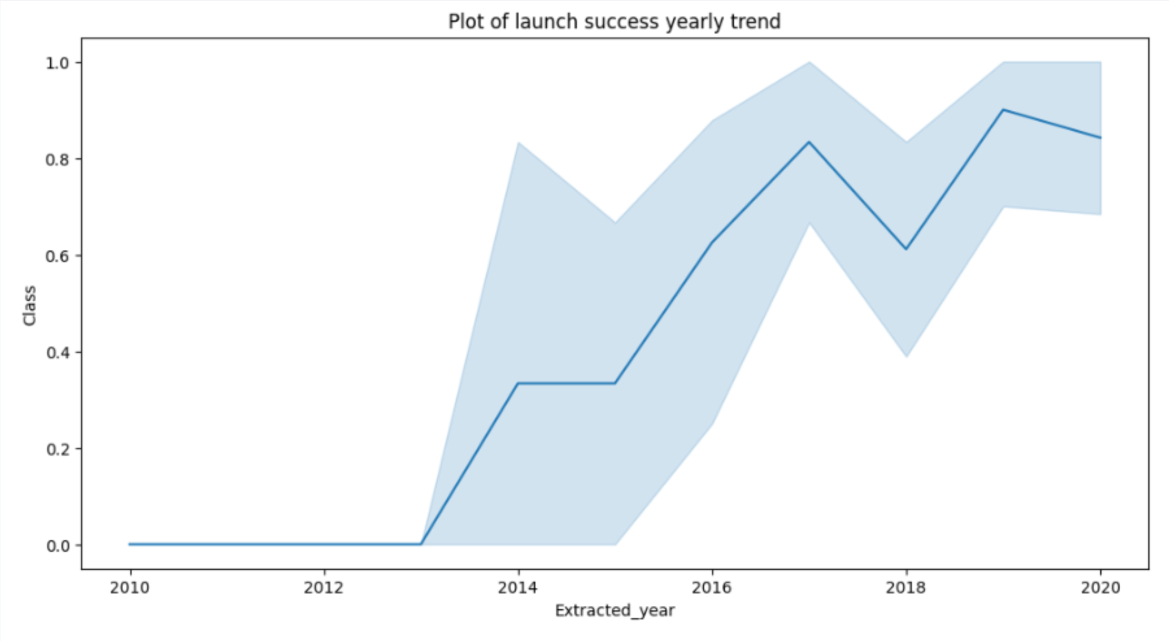
Most orbits had low mass payload under 6000

VELO orbit had launches with only higher payloads and were mostly successful.



Launch Success Yearly Trend

- Success rates increases after 2013



All Launch Site Names

```
In [4]: %%sql
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;

 * ibm_db_sa://ftb12020:***@0c77d6f
Done.
```

Out[4]:

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

The unique list of launch_sites are as found in the results.

Launch Site Names Begin with 'CCA'

In [5]:
```
%%sql
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[5]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Entries in database with launch site name beginning with CCA

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

- The query uses the SQL function SUM to add the payload_mass from the database for all rows where the customer was NASA (CRS)

Average Payload Mass by F9 v1.1

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

In [13]:

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

	firstsuccessfull_landing_date
0	2015-12-22

Use the min sql function to get the first date for successful landing

Successful Drone Ship Landing with Payload between 4000 and 6000

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

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

- Using AND condition to get the correct set of records.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

- Using like query and % to get the results.

Boosters Carried Maximum Payload

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

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600

2015 Launch Records

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

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

	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))

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

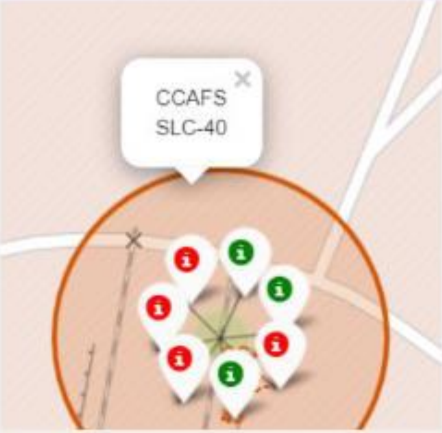
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

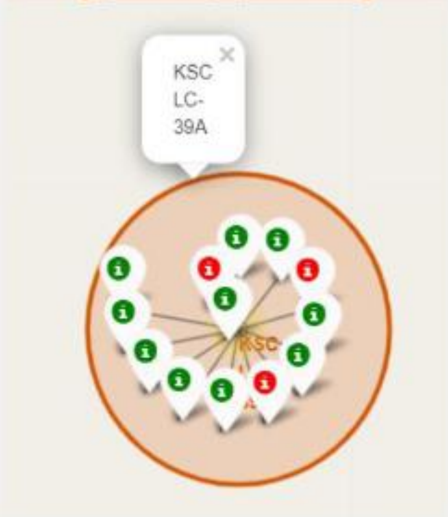
Section 3

Launch Sites Proximities Analysis

All launch sites

Launch sites with markers





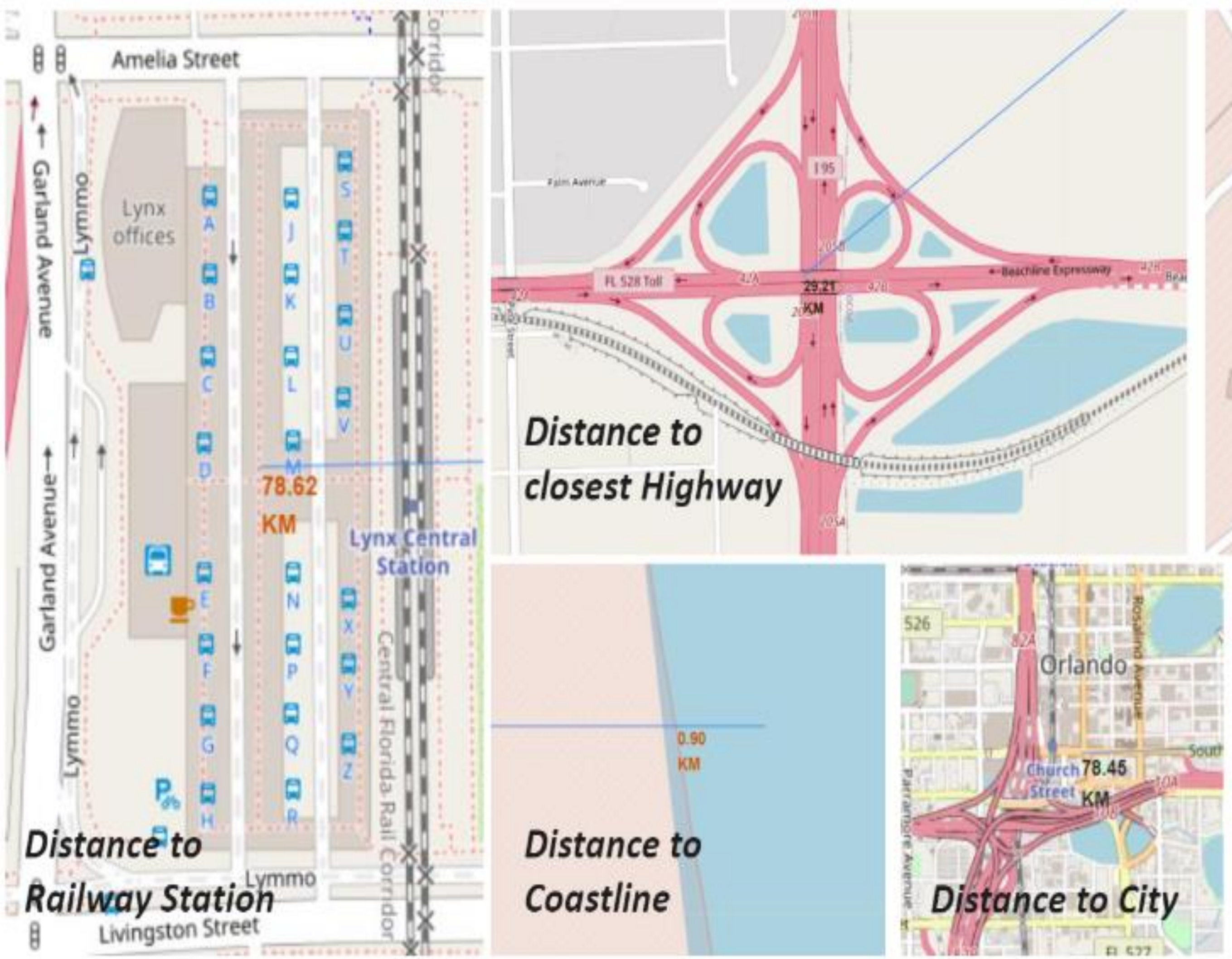
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch sites distance from landmarks

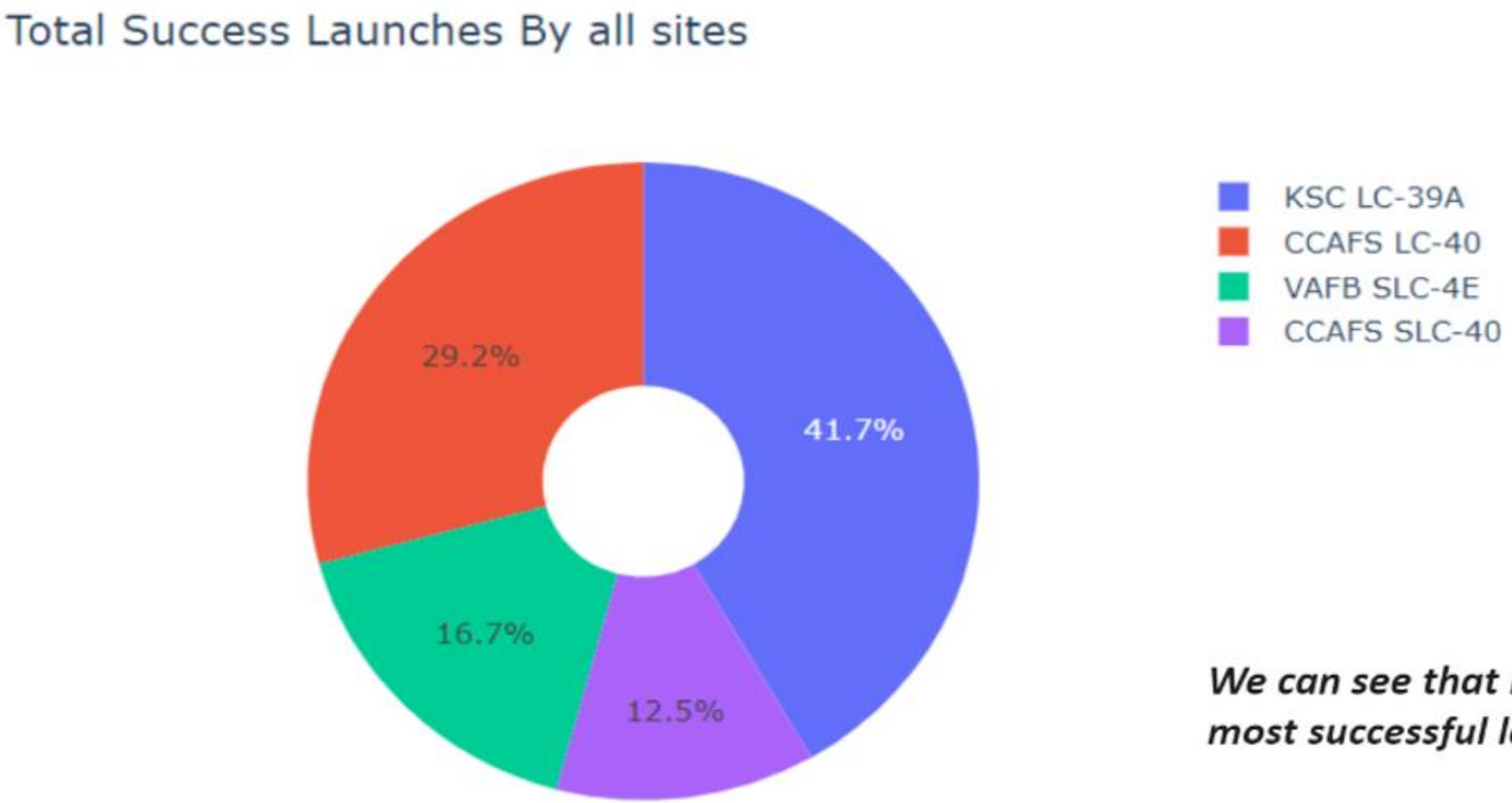


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

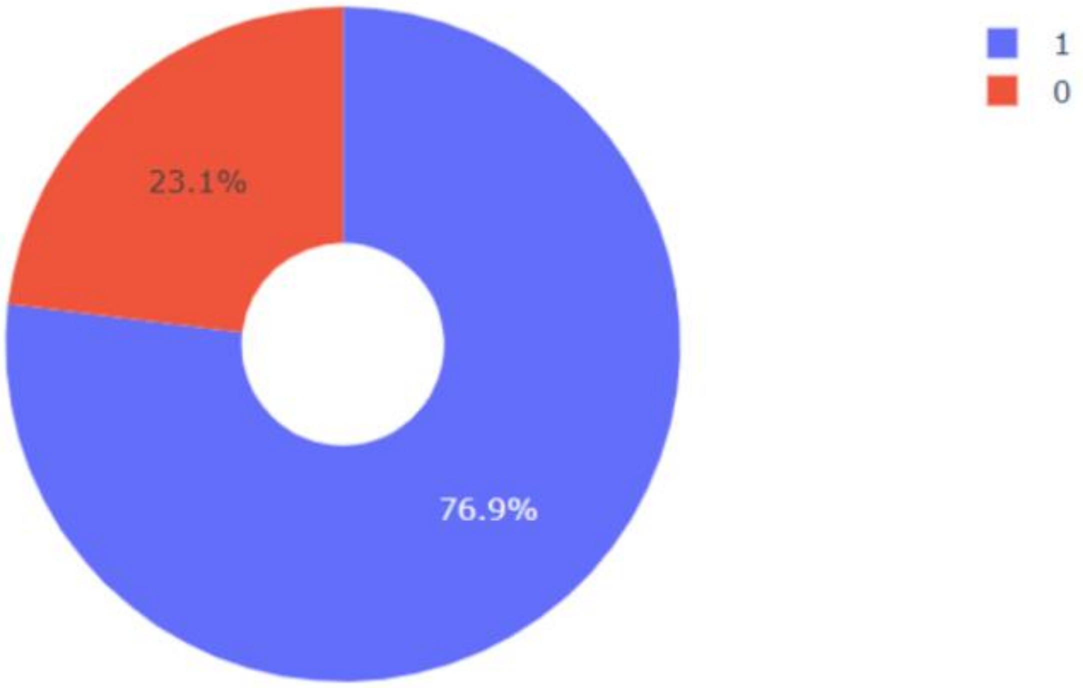
Build a Dashboard with Plotly Dash

Pie chart showing success % by launch site



We can see that KSC LC-39A had the most successful launches from all the sites

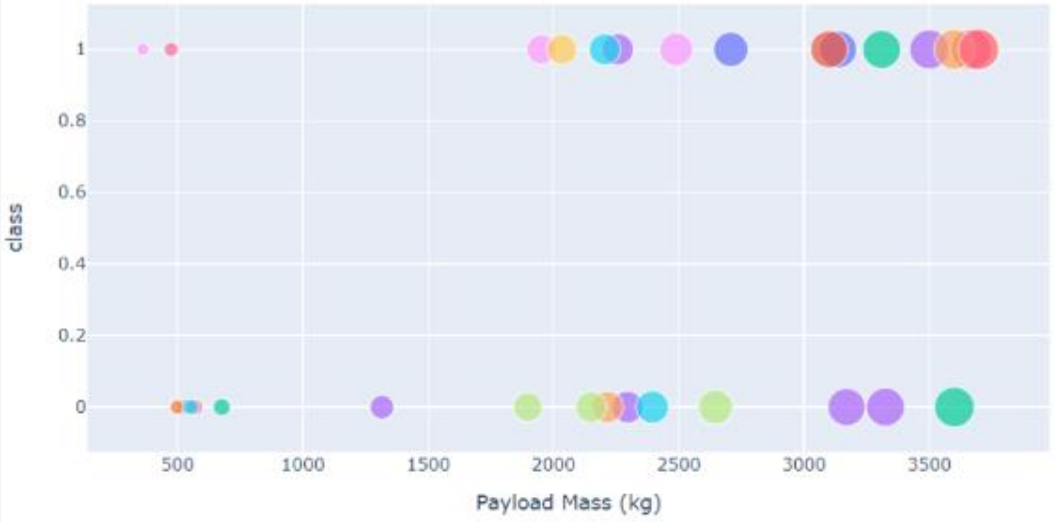
Pie chart with highest launch success ratio



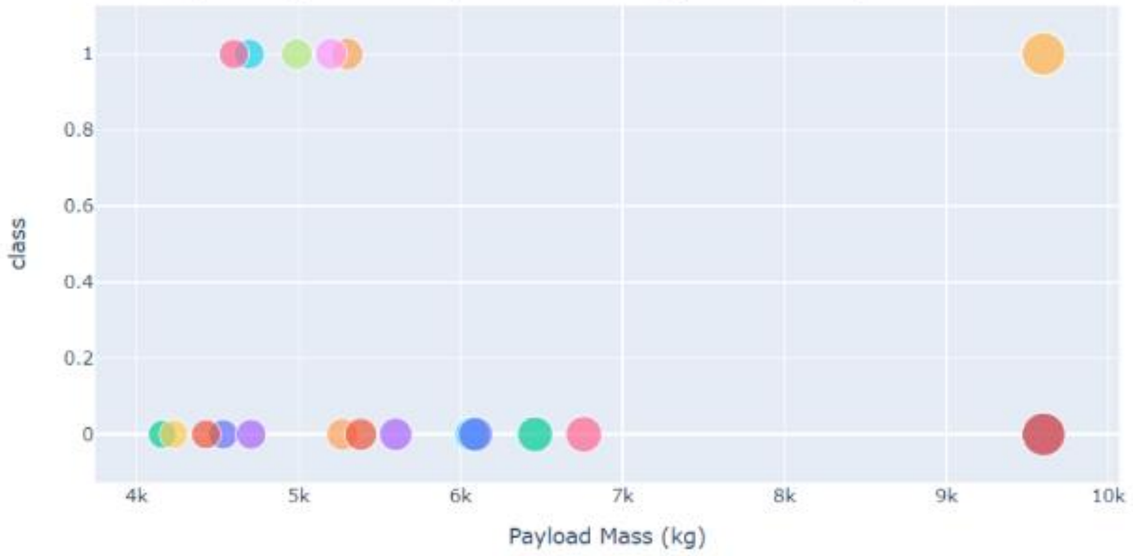
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot Payload vs Launch Outcome

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

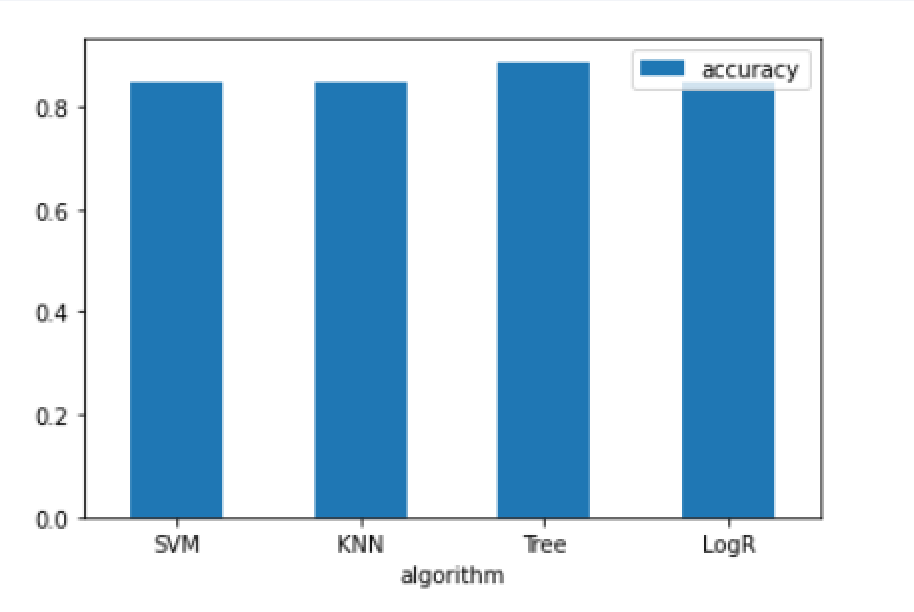
Section 5

Predictive Analysis (Classification)

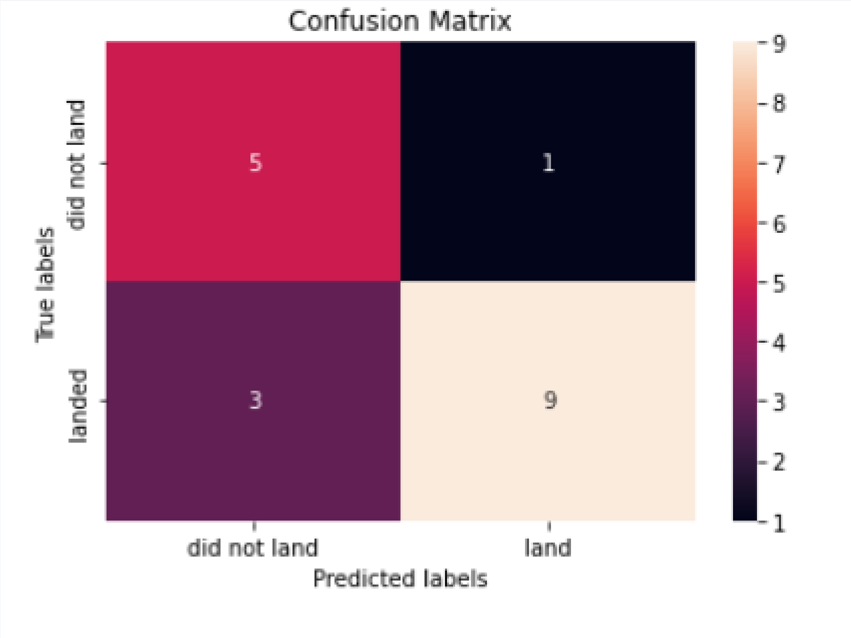
Classification Accuracy

```
algorithms = {'SVM':svm_cv.best_score_,'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_sc
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
if bestalgorithm == 'SVM':
    print('Best Params is :',svm_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8875
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2,
'splitter': 'random'}
```



Confusion Matrix



Conclusions

- Launch success rates started to increase after 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 Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- GitHub Repository URL
  - https://github.com/prachisc/AppliedDSCapstone

Thank you!