IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Vidit Sharma
26th July 2022

Outline

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- EDA with SQL
- EDA with Data Visualization
- Interactive Visual Analytics with Folium lab
- Machine Learning Prediction

Summary of all results

- Data Collection and EDA result
- Machine Learning Prediction provided the best model to predict the outcome

Introduction

Project background and context

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- What are the best places to make a launch?

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data was collected using SpaceX API and from web scraping from Wikipedia.

Perform data wrangling

- perform some Exploratory Data Analysis (EDA) to find some patterns in the 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

- Training Data is used to build, tune, evaluate classification models on the Testing Data.

Data Collection

- Data was collected using two methods:
  - Using SpaceX API
  - https://api.spacexdata.com/v4/rockets/
  - Web Scraping from Wikipedia

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/1.%20Data%20Collection%20API.ipynb

```python
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```python
response = requests.get(spacex_url)
```

```python
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion'] != 'Falcon 1']
data_falcon9.describe()
```

Data Collection - Scraping

- We applied web scrapping to web-scrap Falcon 9 launch records with Beautiful Soup.
- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/2.%20Data%20Collection%20with%20Web%20Scraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup= BeautifulSoup(html_data.text, 'html.parser')
```

Data Wrangling

- We perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
landing_class = []

for key, value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/3.%20Data%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 trend.

- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/5.%20EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- 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.
- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/4.%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

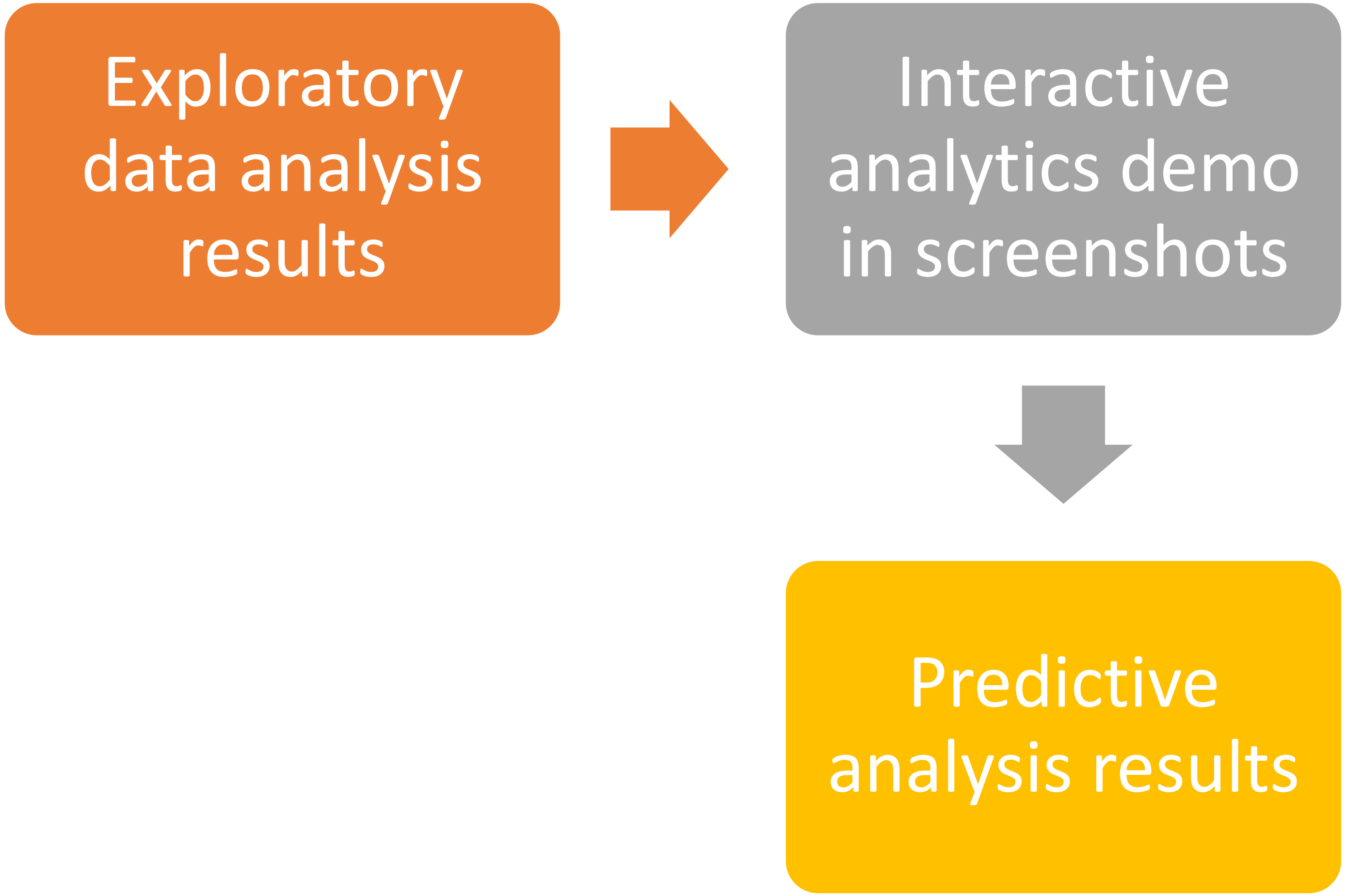
- Markers, circles, lines and marker clusters were used with Folium Maps
  - Markers indicate points like launch sites.
  - Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center.
  - Marker clusters indicates groups of events in each coordinate, like launches in a launch site
  - Lines are used to indicate distances between two coordinates.

- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/6.%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- GitHub URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone/blob/master/7.%20Machine%20Learning%20Prediction.ipynb

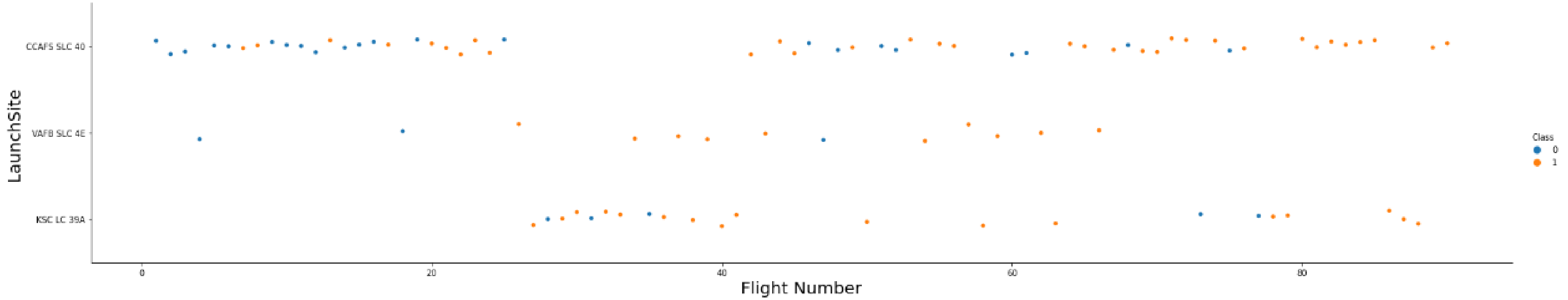
Results


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

Section 2

Insights drawn from EDA

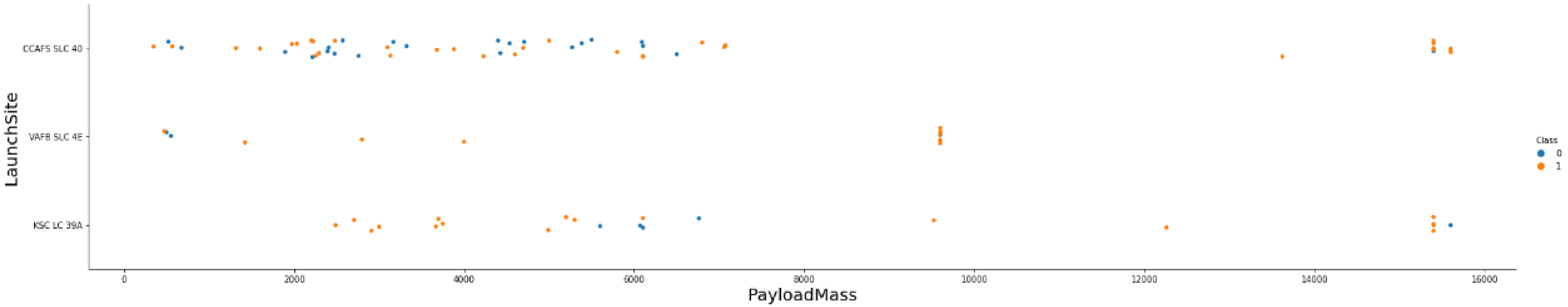
```python
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Flight Number vs. Launch Site

- Flight Number vs. Launch Site

```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

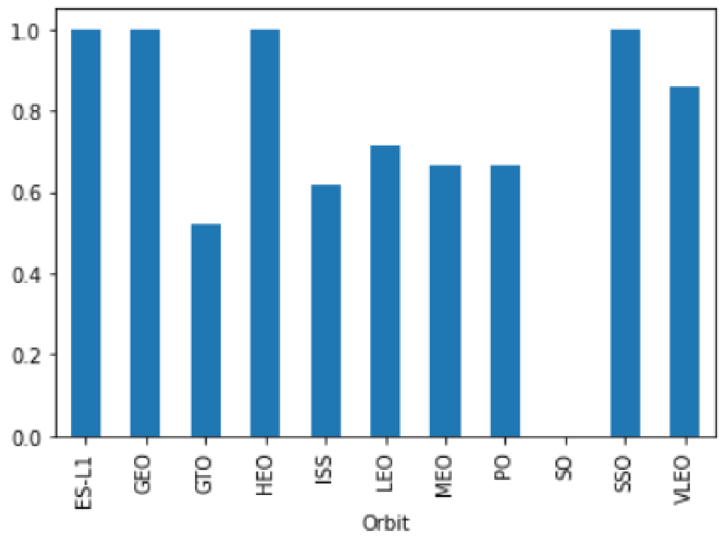


Payload vs. Launch Site

- Payload vs. Launch Site

```
df.groupby('Orbit')['Class'].mean().plot.bar()
```

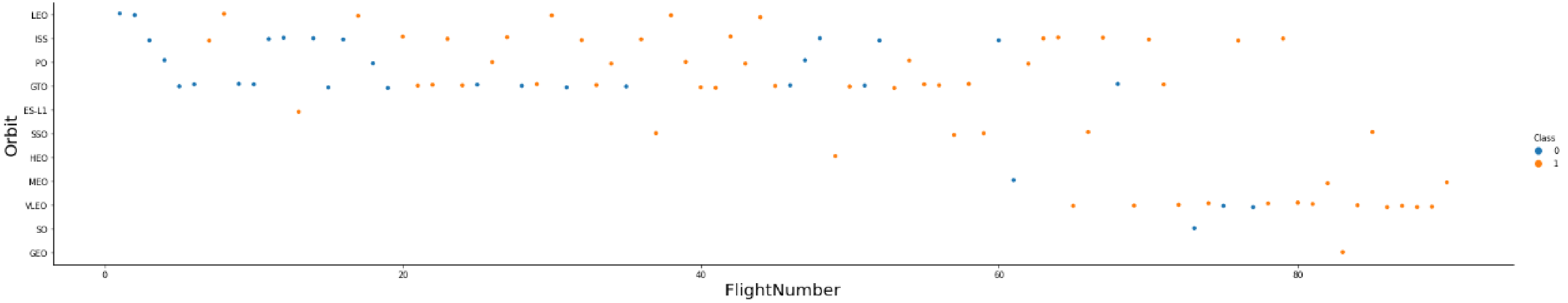
<AxesSubplot:xlabel='Orbit'>



Success Rate vs. Orbit Type

- Success rate of each orbit type

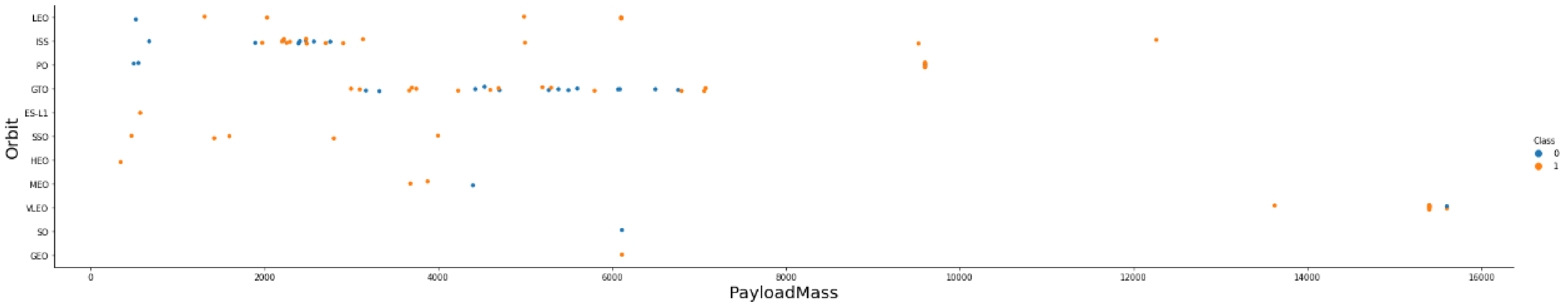
```python
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

- Flight number vs. Orbit type

```python
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

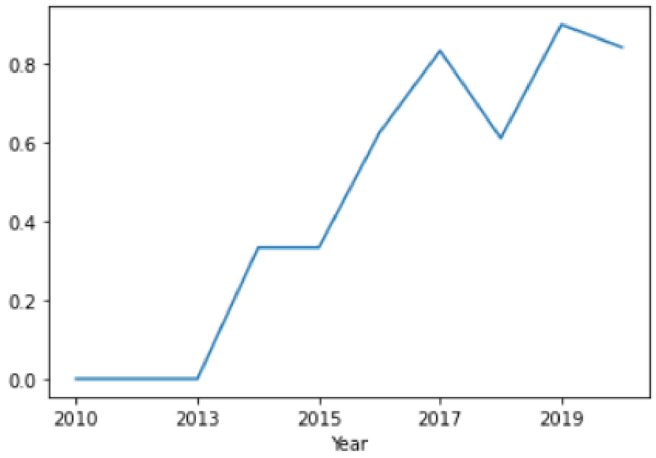


Payload vs. Orbit Type

- payload vs. orbit type

```python
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
temp_df = df.copy()
temp_df['Year'] = year
temp_df.groupby('Year')['Class'].mean().plot()
```

<AxesSubplot:xlabel='Year'>



Launch Success Yearly Trend

- Yearly average success rate

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1
Done.
```

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

All Launch Site Names

```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
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	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

Launch Site Names Begin with 'CCA'

```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

total_payload
111268

Total Payload Mass

```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

avg_payload
2928

Average Payload Mass by F9 v1.1

```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

first_success_gp
2015-12-22

First Successful Ground Landing Date

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

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

Successful Drone Ship Landing with Payload between 4000 and 6000

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

mission_outcome	qty
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Total Number of Successful and Failure Mission Outcomes

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDE
R BY BOOSTER_VERSION;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4

Boosters Carried Maximum Payload

```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DAT
E) = 2015;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

2015 Launch Records

```
sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__O
UTCOME ORDER BY QTY DESC;
```

```
 * ibm_db_sa://hyb94666:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

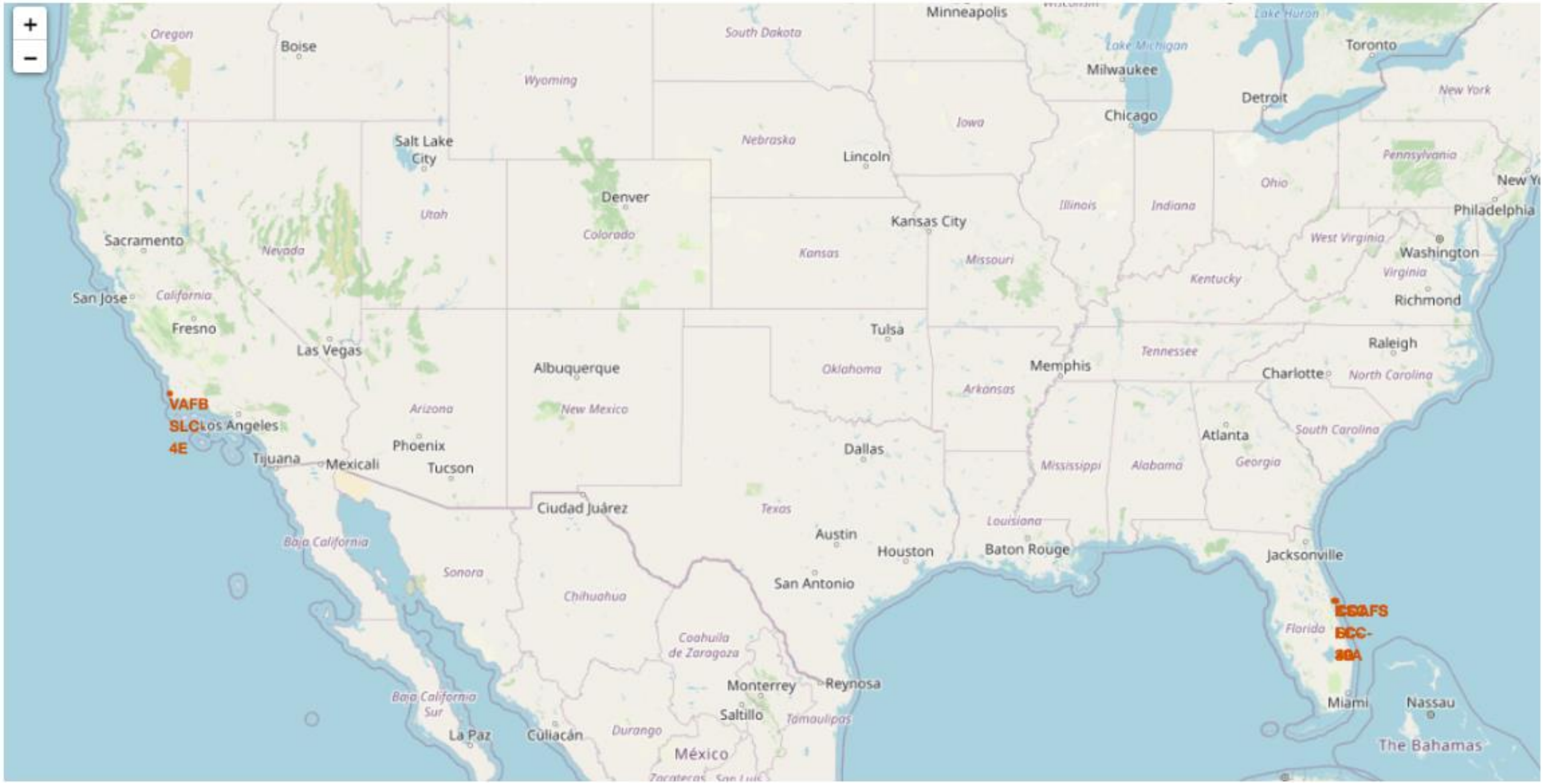
landing__outcome	qty
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Section 3

Launch Sites Proximities Analysis

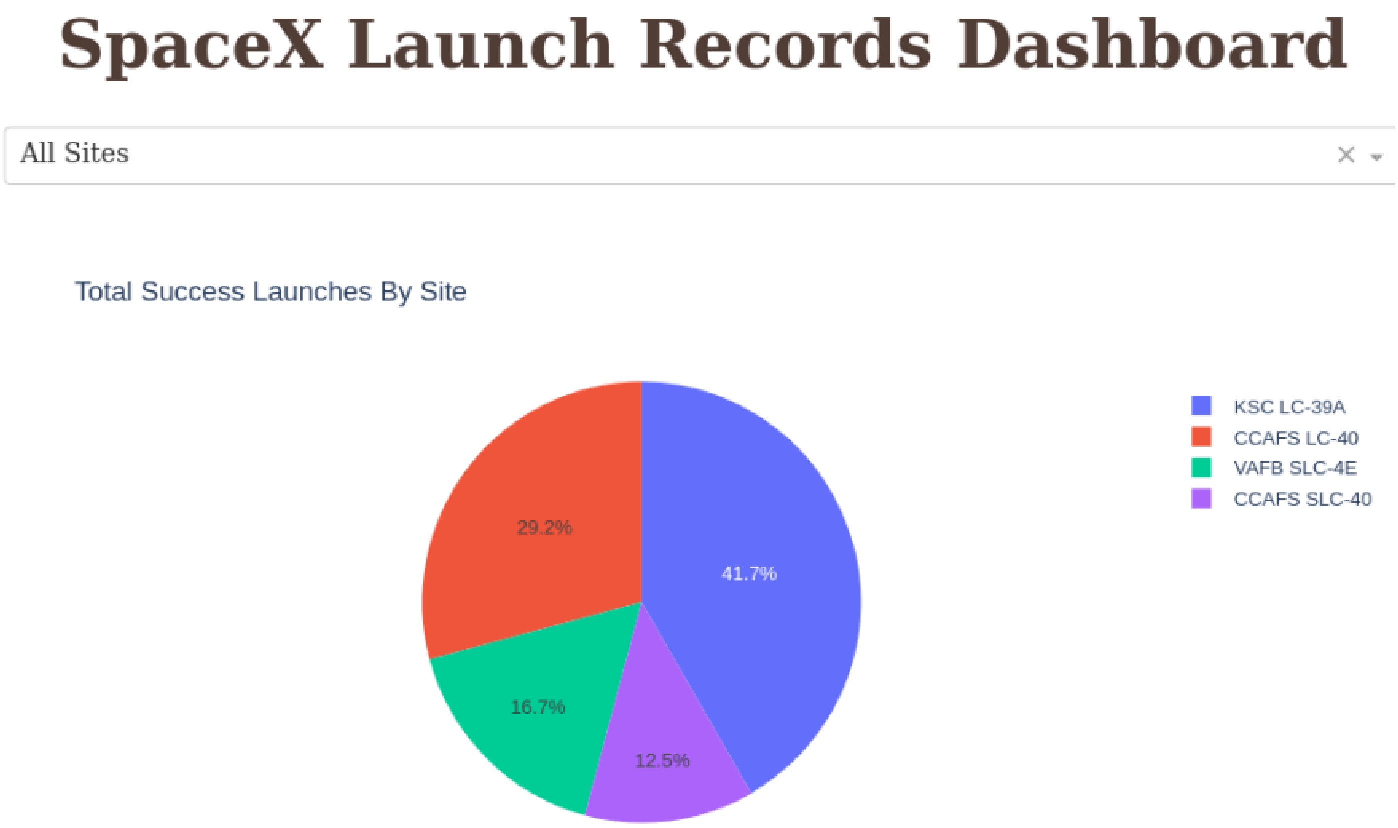
All launch sites US map markers



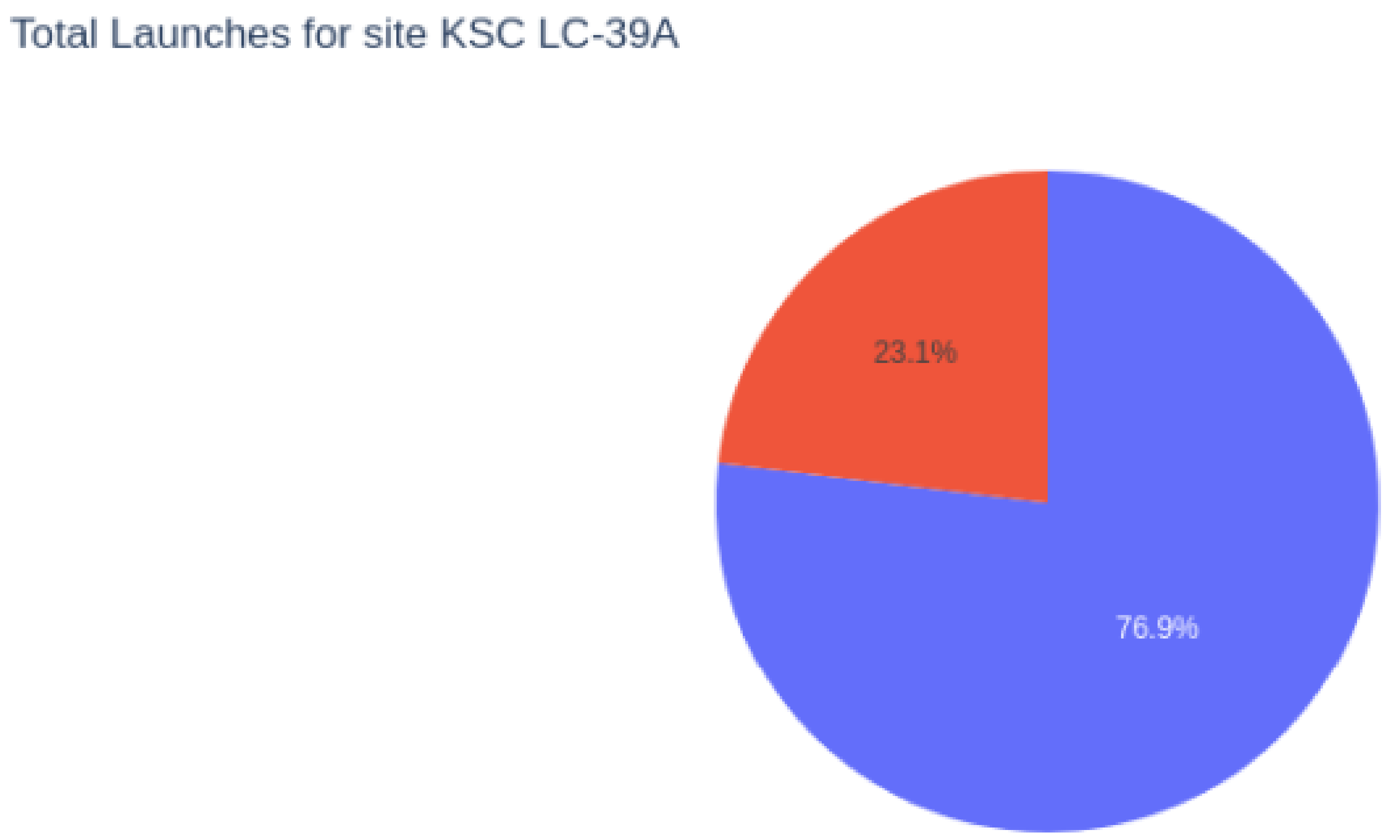
Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



Launch Success Ratio for KSC LC-39A



Section 5

Predictive Analysis (Classification)

Classification Accuracy

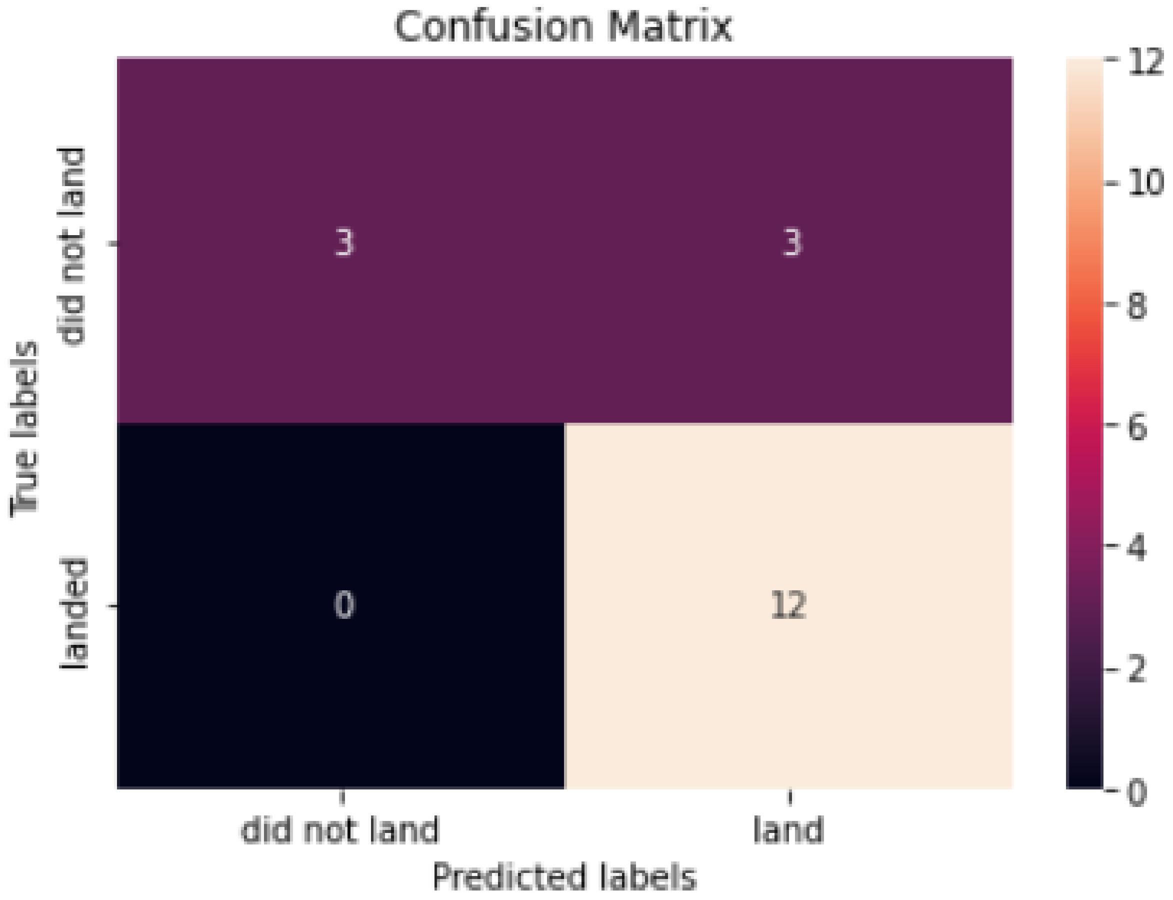
```
print("Model\t\tAccuracy\tTestAccuracy")#,logreg_cv.best_score_)
print("LogReg\t\t{}\t\t{}".format((logreg_cv.best_score_).round(5), logreg_cv.score(X_test, Y_test).round(5)))
print("SVM\t\t{}\t\t{}".format((svm_cv.best_score_).round(5), svm_cv.score(X_test, Y_test).round(5)))
print("Tree\t\t{}\t\t{}".format((tree_cv.best_score_).round(5), 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		0.87679		0.88889
KNN		0.84821		0.83333
```

Confusion Matrix



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- 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 URL: https://github.com/vidit-sharma/IBM-Data-Science-Capstone.git

Thank you!