Applied Data Science Capstone

Cody Oneil Harry

2023-08-24

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Applying data science skills as a Data Scientist for a private space launch company which includes recognizing the significance of gathering as much pertinent data as feasible, collecting data from diverse sources enhancing its quality through data wrangling becomes a focal point and delving into the processed data.

. Guided by expertise, the exploration of captivating real-world datasets occurs collaboratively with the team. The opportunity arises to practice SQL skills for data querying and gaining valuable insights. Further comprehension of the data unfolds while employing rudimentary statistical analysis and data visualization methods. These tools allow for direct observation of relationships among variables. Data segmentation into groups characterized by categorical variables or factors reveals more intricate insights within the dataset, unlocking potential for even more riveting discoveries.

INTRODUCTION

In this capstone project, the goal is to predict the successful landing of the Falcon 9 first stage. SpaceX prominently features Falcon 9 rocket launches on its website, pricing them at 62 million dollars. In contrast, other providers charge upwards of 165 million dollars per launch. SpaceX's cost efficiency is largely due to its ability to reuse the first stage of the rocket. Therefore, by accurately predicting the first stage landing outcome, we can effectively estimate the launch cost. This predictive insight holds particular value when considering competitive bids against SpaceX for rocket launches.

Within this lab, the primary task involves gathering data from an API and ensuring its conformity to the correct format. The following excerpt illustrates both a successful launch and its respective outcome.

METHODOLOGY

Firstly, our focus will be on conducting Exploratory Data Analysis (EDA) to uncover discernible patterns within the data. Additionally, we will ascertain the most appropriate label for training supervised models.

SpaceX launch data was gathered from the SpaceX Rest API and Wikipedia page. This provided information about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

Contained within the dataset are multiple instances where the booster's landing was not successful. These cases encompass a range of scenarios. Sometimes, an attempted landing resulted in failure due to unforeseen events. For instance, when the outcome is labeled as "True Ocean," it signifies that the mission concluded with a successful landing within a specific region of the ocean. Conversely, when labeled as "False Ocean," it indicates an unsuccessful landing within a designated ocean region.

METHODOLOGY

Data Collection and Wrangling

```python
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```python
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```python
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

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

200

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

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

Print the page title to verify if the `BeautifulSoup` object was created properly

```python
# Use soup.title attribute
soup.title
```

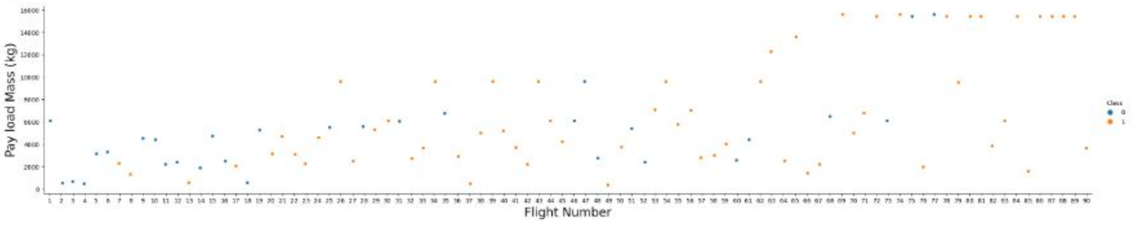
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

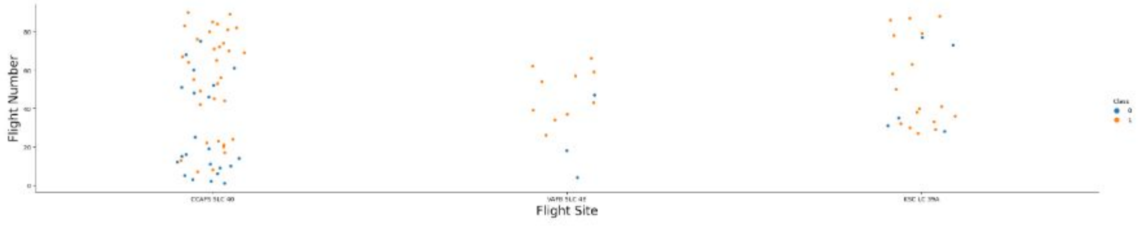
METHODOLOGY

EDA and Interactive Visual Analytics

```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```




```
sns.catplot(y="FlightNumber", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight SiteFlight Number",fontsize=20)
plt.ylabel("",fontsize=20)
plt.show()
```



METHODOLOGY

Predictive Analysis

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
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_)
```

```
Best Algorithm is Tree with a score of 0.9
Best Params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_spli
t': 10, 'splitter': 'best'}
```

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

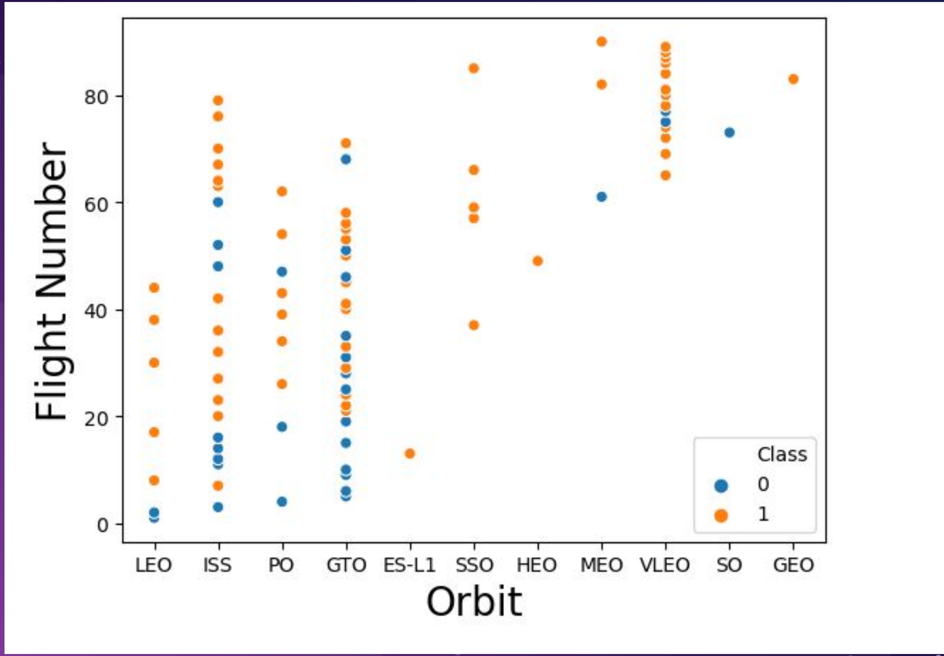
```
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X,Y)
```

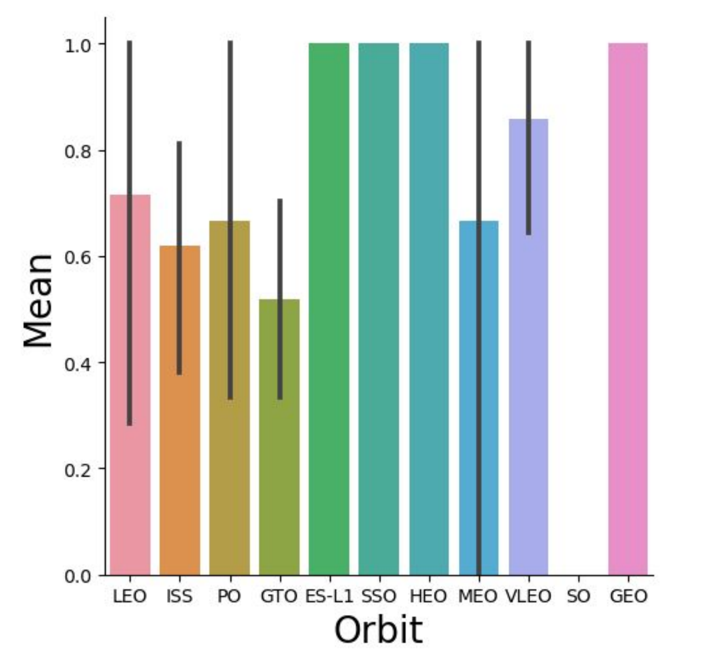
```
import warnings
warnings.filterwarnings(action='ignore')

parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr = LogisticRegression()

logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X,Y)
```

RESULTS

EDA with Visualization





RESULTS

EDA with SQL

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

* sqlite:///my_data1.db
Done.

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

```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

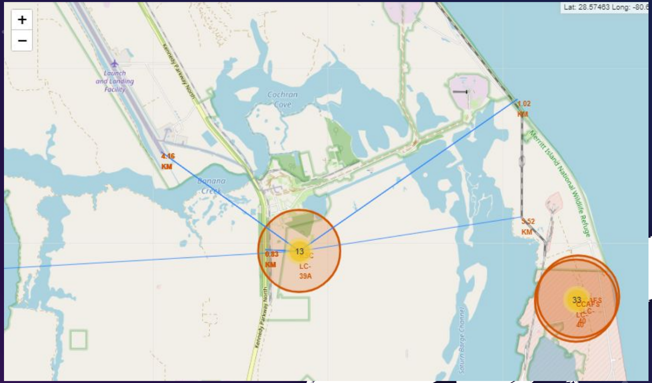
* sqlite:///my_data1.db
Done.

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

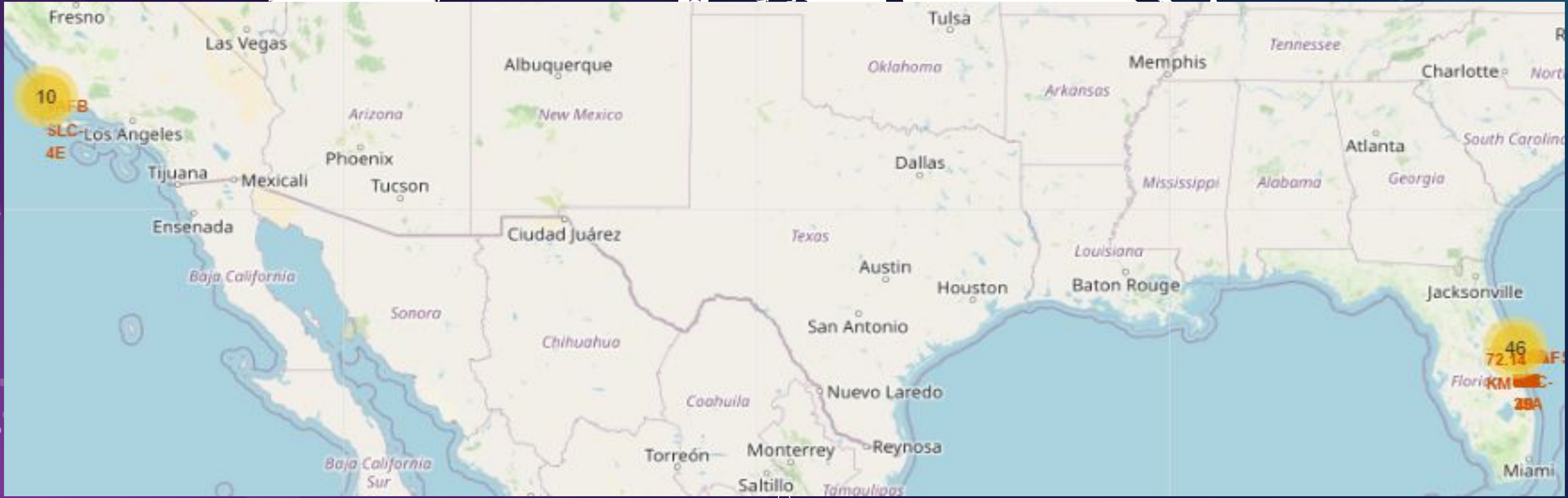
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING_OUTCOME='Success (
```

* sqlite:///my_data1.db
Done.

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

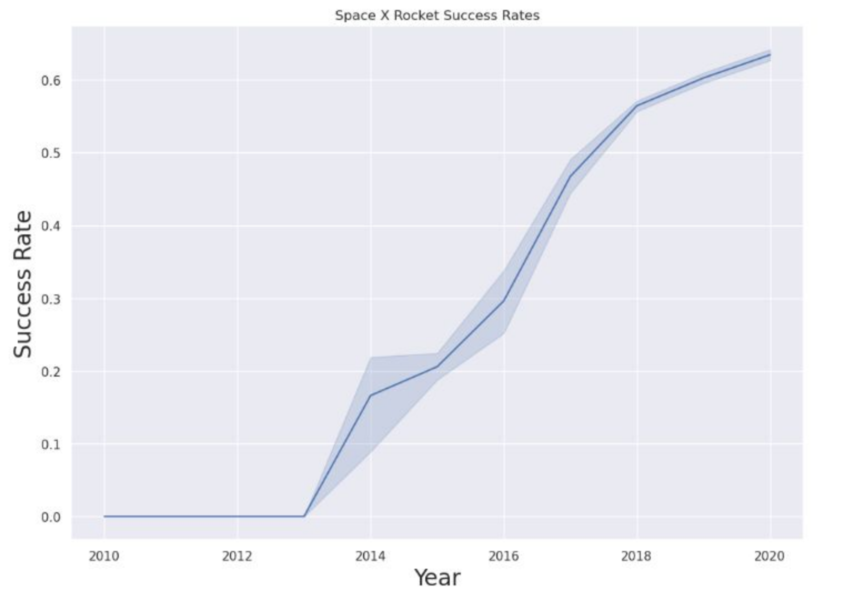


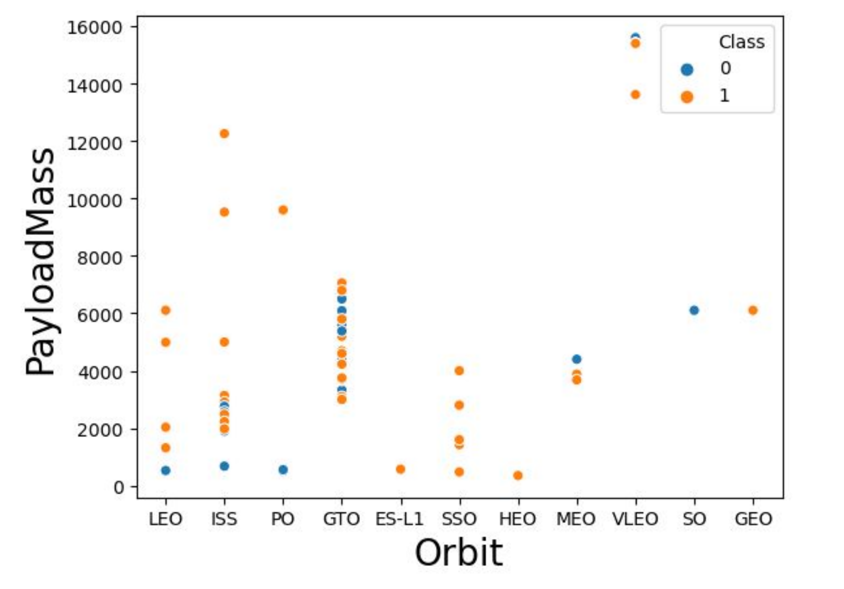
INTERACTIVE MAPS



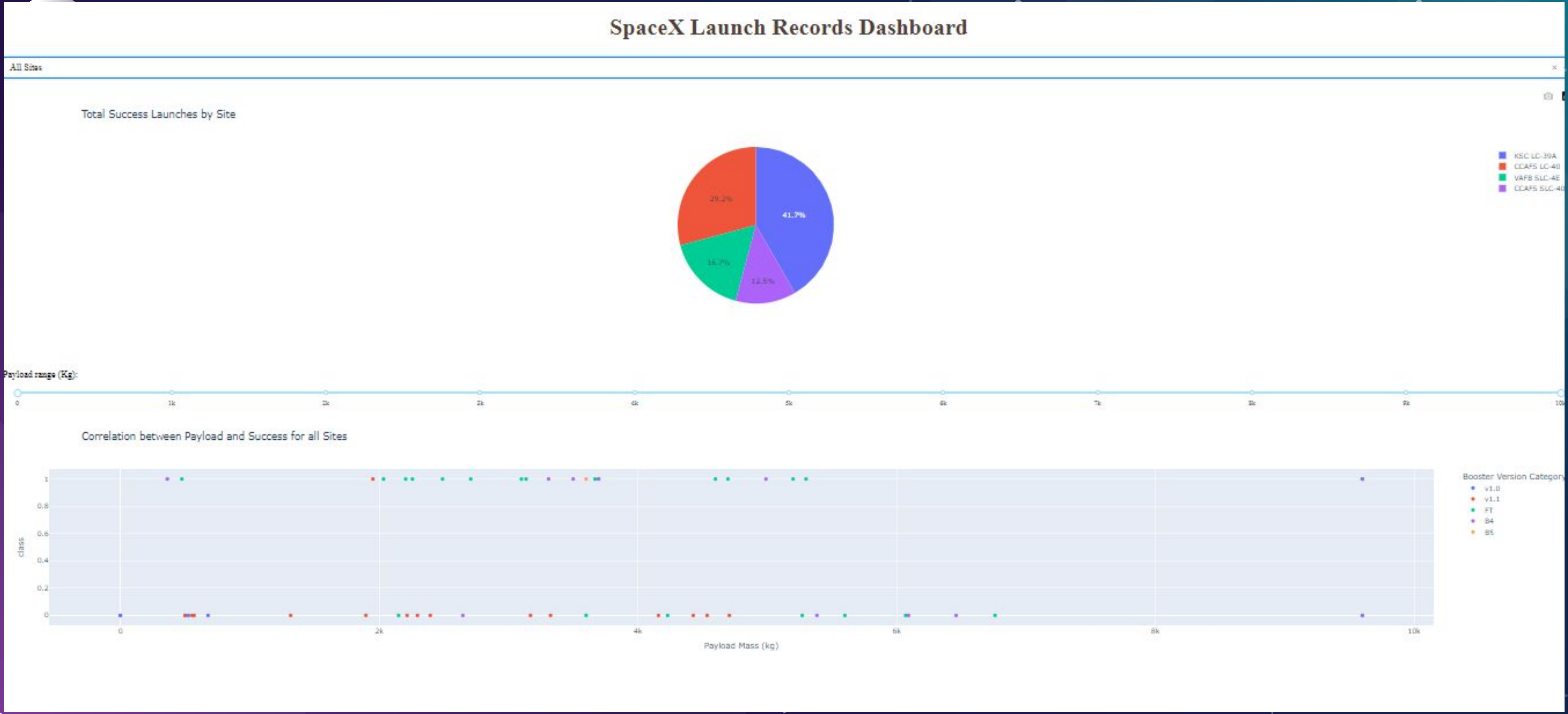
RESULTS

Predictive Analysis

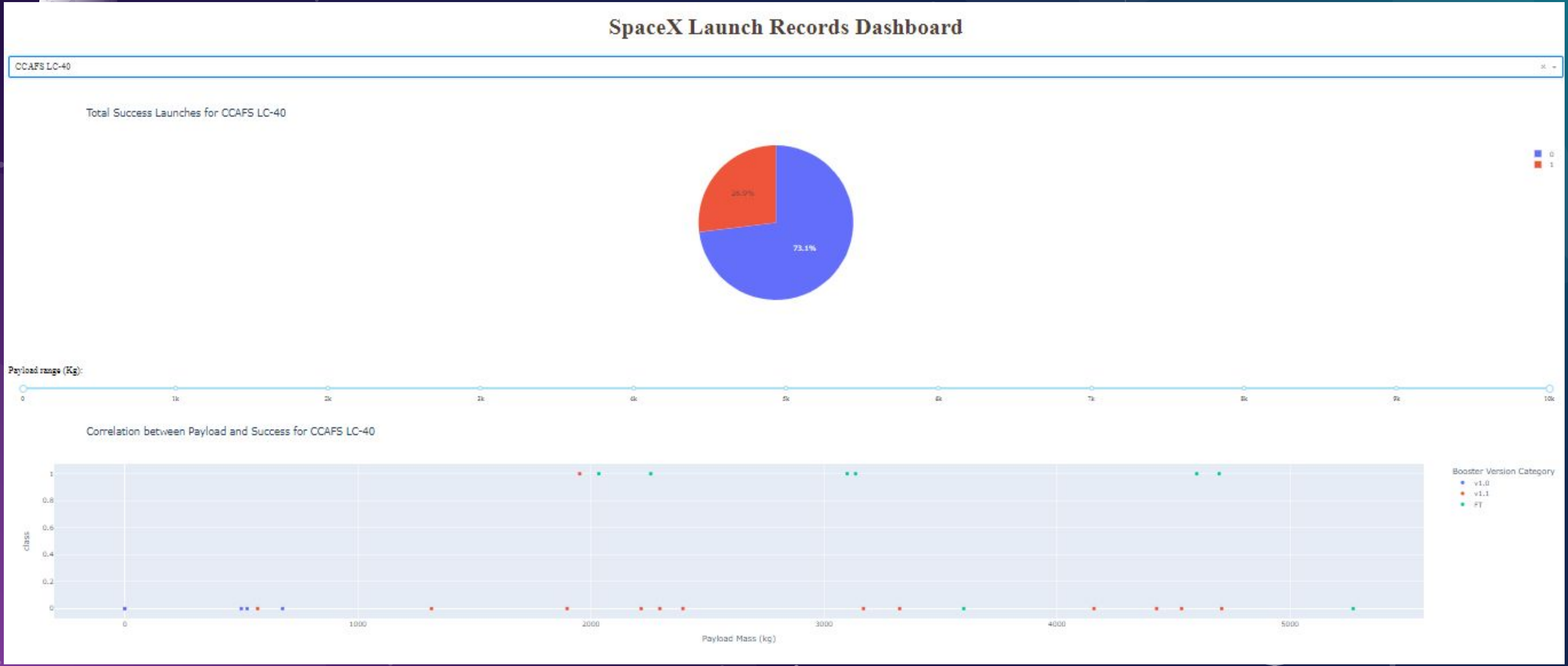




DASHBOARDS - Plotly Dash



DASHBOARDS - Plotly Dash



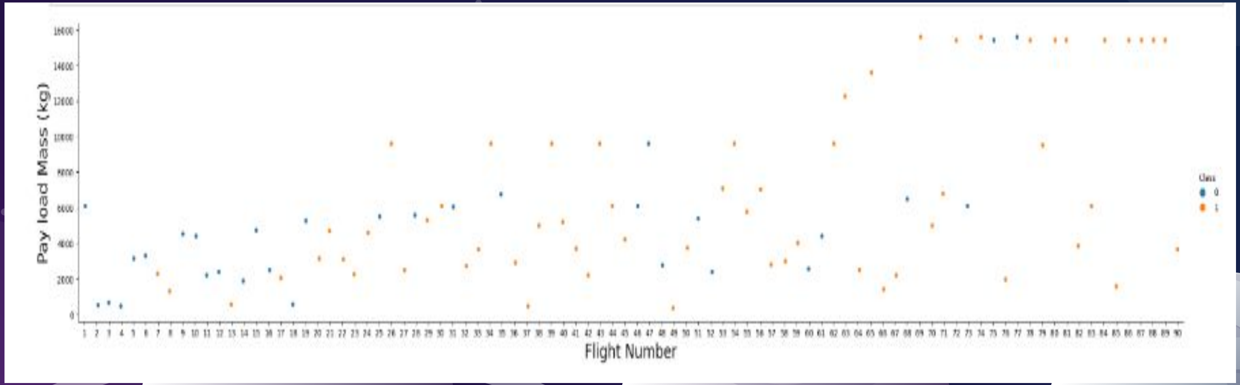
DISCUSSION

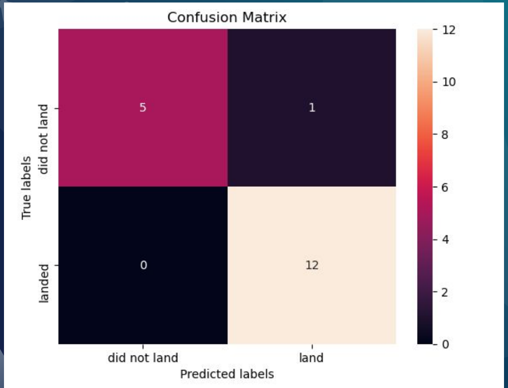
The success rate has increased significantly over the years.
The first successful landing outcome on ground pad occurred on December 22, 2015.
CCAFS SLC-40 was the launch site that had the highest Falcon 9 first stage landing success rate (42.9%).
The payload range from about 2,000 kg to 5,000 kg has the largest success rate.
All models performed equally well except for the Decision Tree model which performed poorly relative to the other models.
ES-L1, SSO, HEO and GEO orbits have no failed first stage landings.

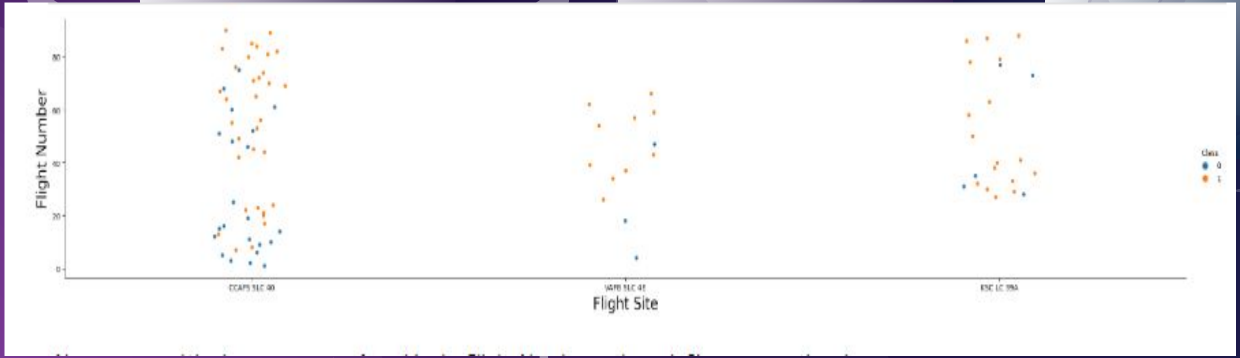
CONCLUSION

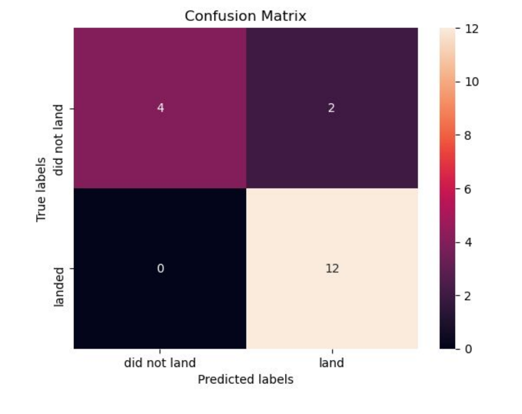
> The Tree Classifier Algorithm is the best for Machine Learning for this dataset
> Low weighted payloads perform better than the heavier payloads
> The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
> We can see that KSC LC-39A had the most successful launches from all the sites
> Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

APPENDIX


Pay load Mass (kg)
Flight Number
Class


Confusion Matrix
True labels
did not land
landed
5
1
0
12
did not land
land
Predicted labels


Flight Number
Flight Site
Class


Confusion Matrix
True labels
did not land
landed
4
2
0
12
did not land
land
Predicted labels

Thanks!