IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

RICHA GUPTA

Outline

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

Executive Summary

Space Y is the new company would like to compete with SpaceX founded by Billionaire industrialist Allon Musk. Hence our team need to determine the price of each launch. To Know the price of launch we can compare it with Space and can gather information about Space X

First we should determine if SpaceX will reuse the first stage. instead of using rocket science to determine if the first stage will land successfully, we can use machine learning model and use public information to predict if SpaceX will reuse the first stage.

By comparing the features of Space X with Space Y, Space Y can works better .For getting better results we can compare different ML models and find the accuracy and precision and can use appropriate evaluation Metrics to get better result

Introduction

- Space X is the most successful satellite provider founder by Allon Musk
- For creating a satellite provider company which performs far better from space X is the current mission
- Space Y is the new satellite provider which should work better
- To make Space Y best, we can gather required information from Space X and train with different ML model and find different characters
- We can find the estimate cost for launch, we can predict whether it will reuse it first stage, it will attempt to land a rocket or not.
- Different visualization techniques can be used to visualize data and can create dashboard.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collecting the Data with an API
  - API will give data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Perform data wrangling

  Data split into dependent variable and independent variable, dependent variable as label (1,0) based on whether it crash on landing

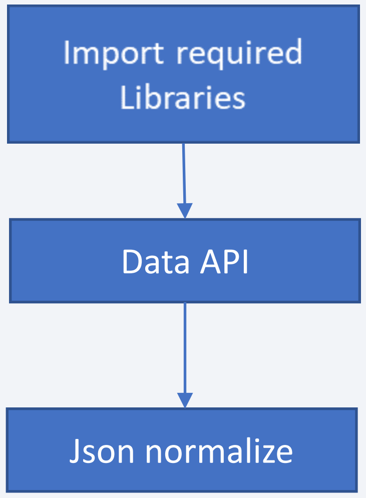
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

  After Splitting of data as test and train test perform different ML modeling

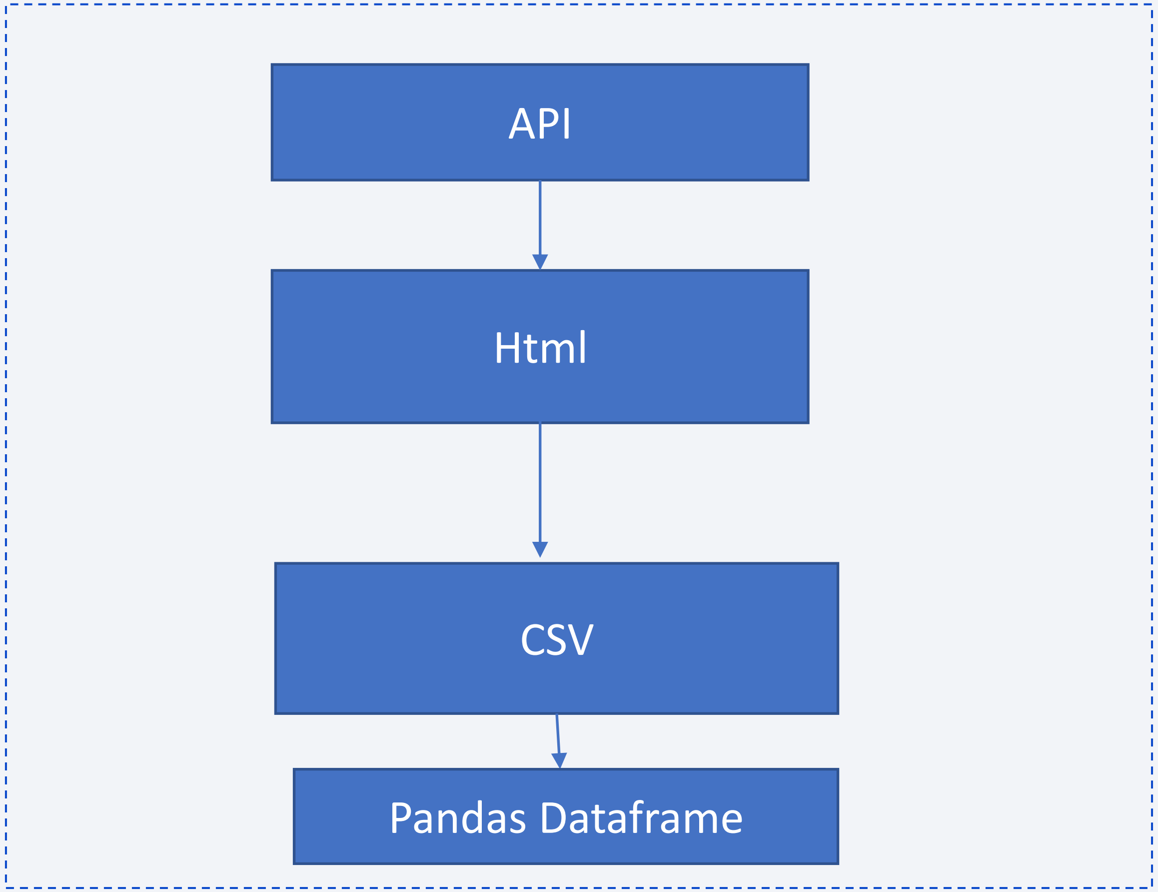
Data Collection

- Collecting the Data with an API
- API will give data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

Data Collection – SpaceX API



Data Collection - Scraping



Data Wrangling

- Checked Null Values
- Replace null value with mean
- Checking Outlier ,removing Outlier
- Find outcome as label

EDA with Data Visualization

- Visualize flight number and Launch site using Scatter plot
- Visualize payload and Launch site using Scatter Plot
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between Flight Number and Orbit type
- Visualize the relationship between Payload and Orbit type
- Visualize the launch success yearly trend using line chart
- Using feature engineering create features
- Create dummy variable to all categorical value
- Change all numeric column to float

EDA with SQL

- Displayed the names of the unique launch sites in the space mission
- Displayed 5 records where launch sites begin with the string 'CCA’
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

Build an Interactive Map with Folium

- Mark all launch sites on a map
- Add circle inside map
- Mark the success/failed launches for each site on the map
- While adding this inside map we can find the proximity of the launch site to different points
- Launch site is near to Highway for easy transportation of goods
- Launch site is near to railway to carry heavy goods
- Launch site is faraway from city to avoid dense population and accidents

Predictive Analysis (Classification)

- Import all required python libraries including Scikit Learn libraries
- Change class into numpy Array
- Preprocessing with standard scalar
- Split train and test set
- Create a logistic regression object then create a GridSearchCV object.
- Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2.

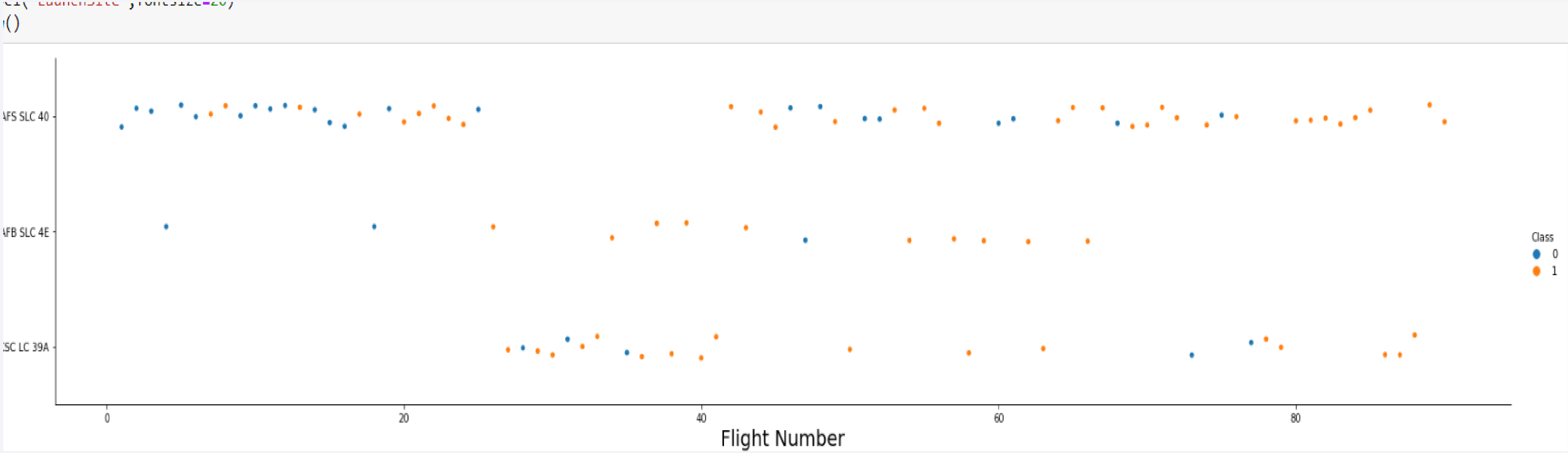
Results

- Launch sites are in closer to coastline so they can fly over the ocean during launch
- Launch sites are in closer to railways, for transporting heavy cargo.
- Launch sites are not in close to cities, which reduce accident
- Launch sites are in closer to highways, which allows for easily transport

Section 2

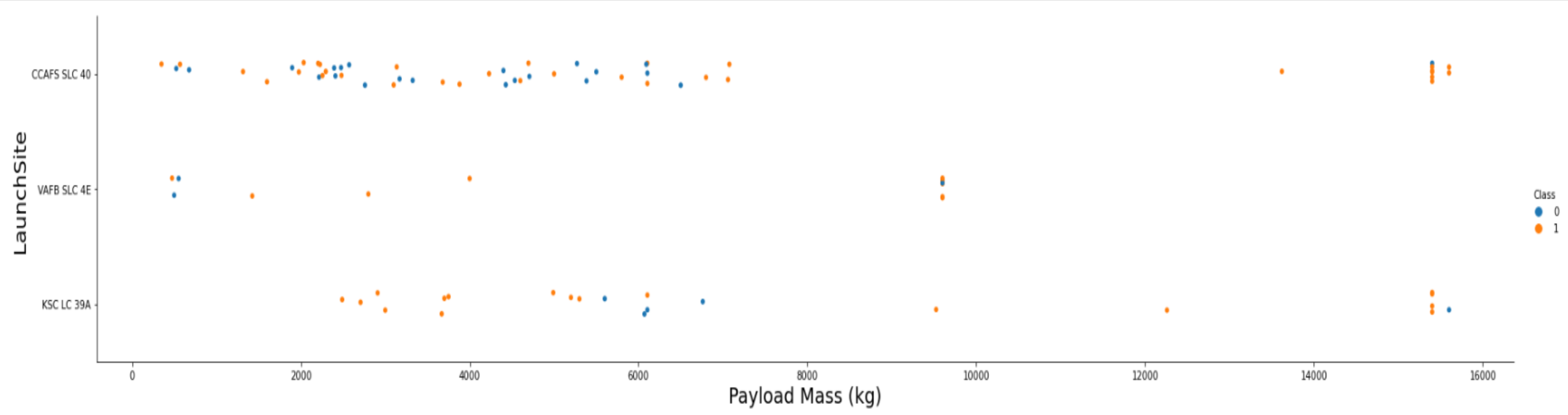
Insights drawn from EDA

Flight Number vs. Launch Site

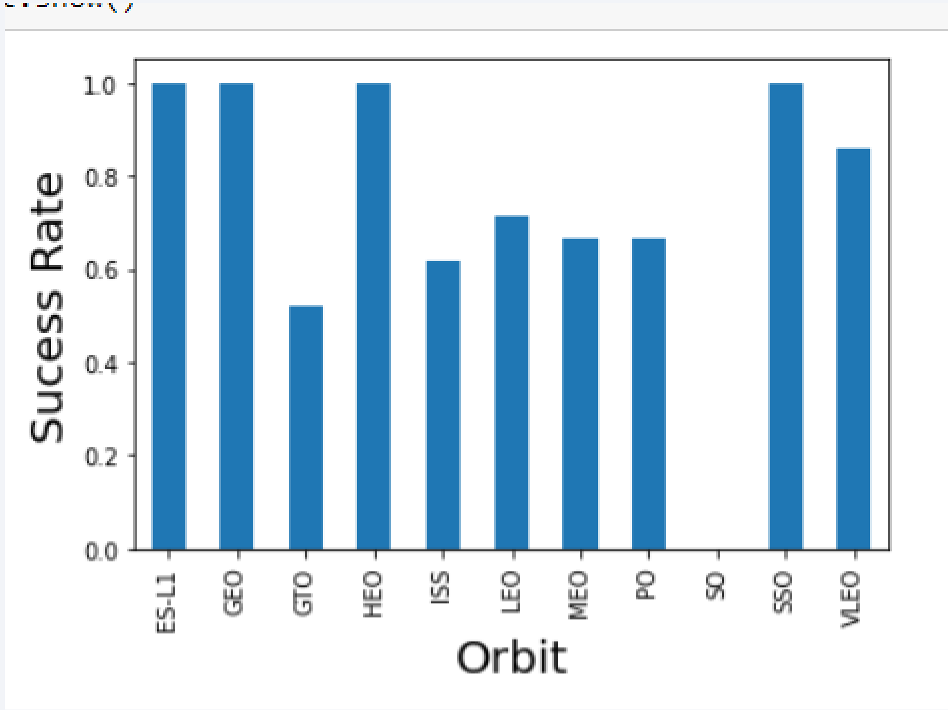


Payload vs. Launch Site

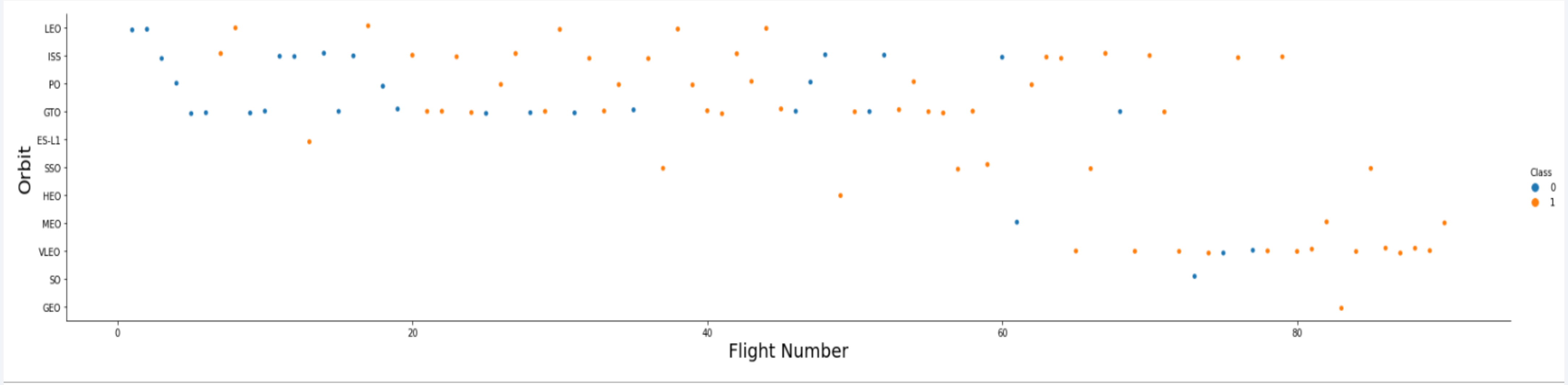
```
t.ylabel("LaunchSite",fontsize=20)
t.show()
```



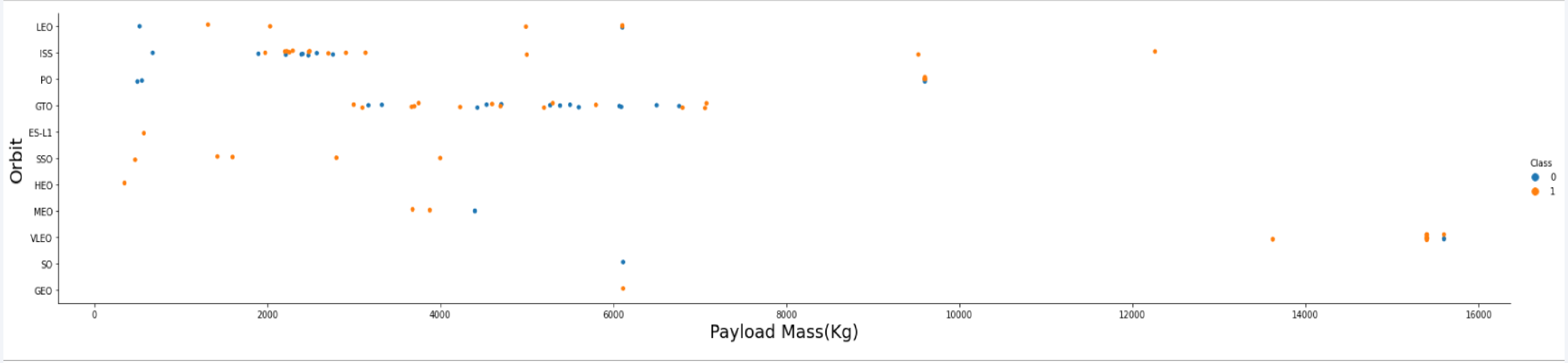
Success Rate vs. Orbit Type



Flight Number vs. Orbit Type

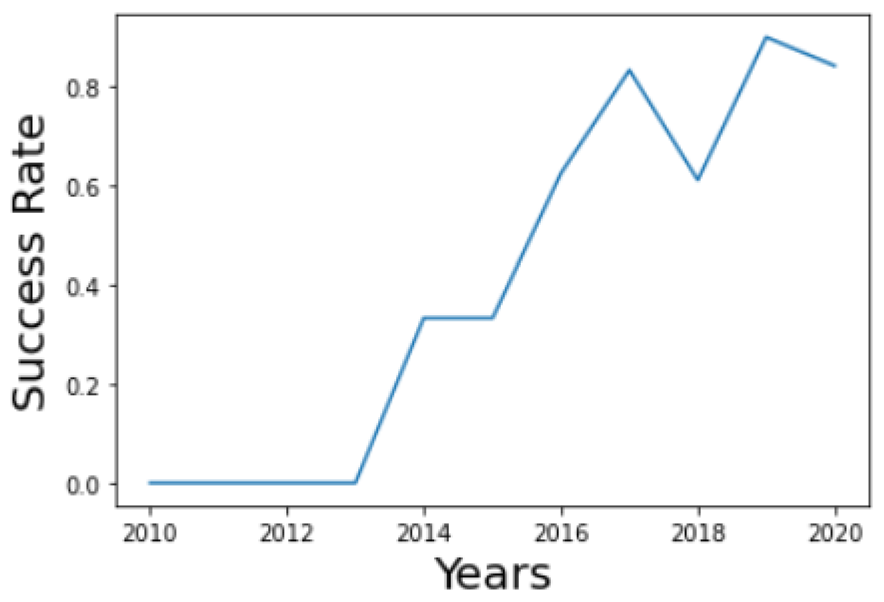


Payload vs. Orbit Type



Launch Success Yearly Trend

```
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;

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

Launch Site Names Begin with 'CCA'

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

Total Payload Mass

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

1
45596

Average Payload Mass by F9 v1.1

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

1
2534

First Successful Ground Landing Date

%%sql SELECT MIN() FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT COUNTA(LANDING__OUTCOME) AS SUCCESSFUL_MISSIONS FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success%';
```

successful_missions
61

Boosters Carried Maximum Payload

%%sql SELECT DISTINCT(BOOSTER_VERSION), PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600

2015 Launch Records

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

- Present your query result with a short explanation here

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

%%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, YEAR(
) AS DATE_YEAR FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(
) = '2015'

landing__outcome	booster_version	launch_site	date_year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

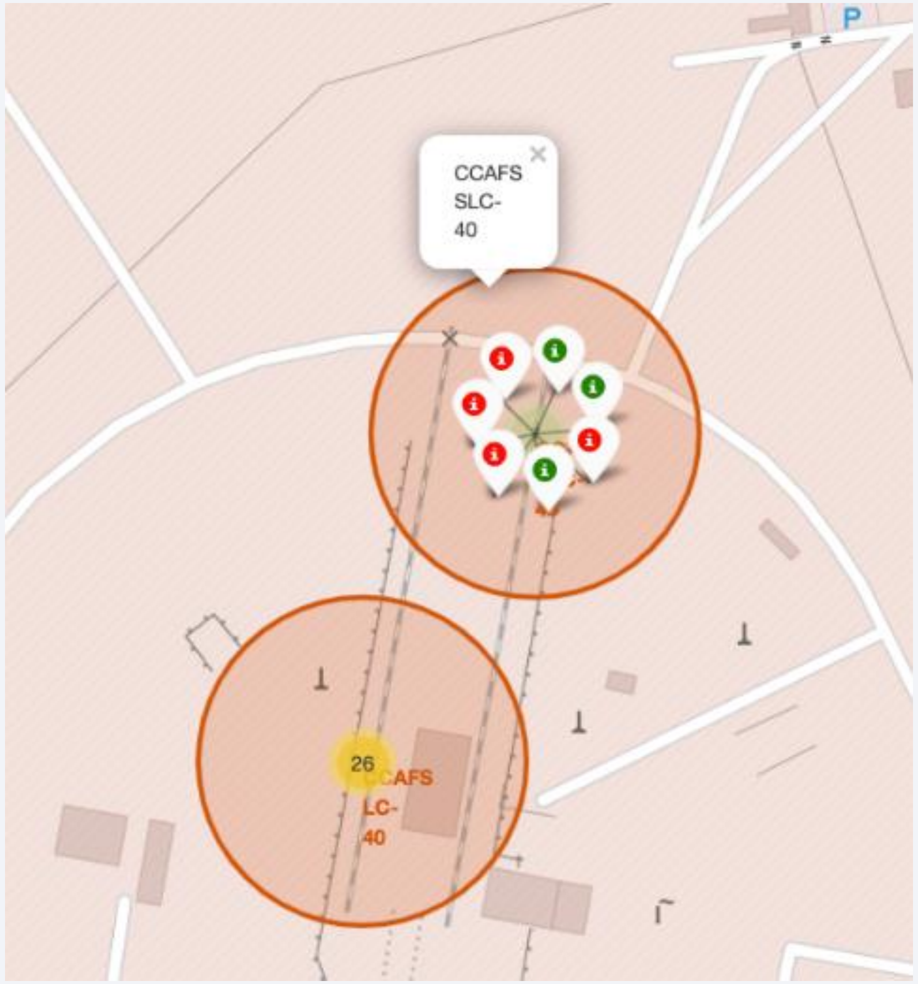
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

<Folium Map Screenshot 2>

<Folium Map Screenshot 3>



Section 5

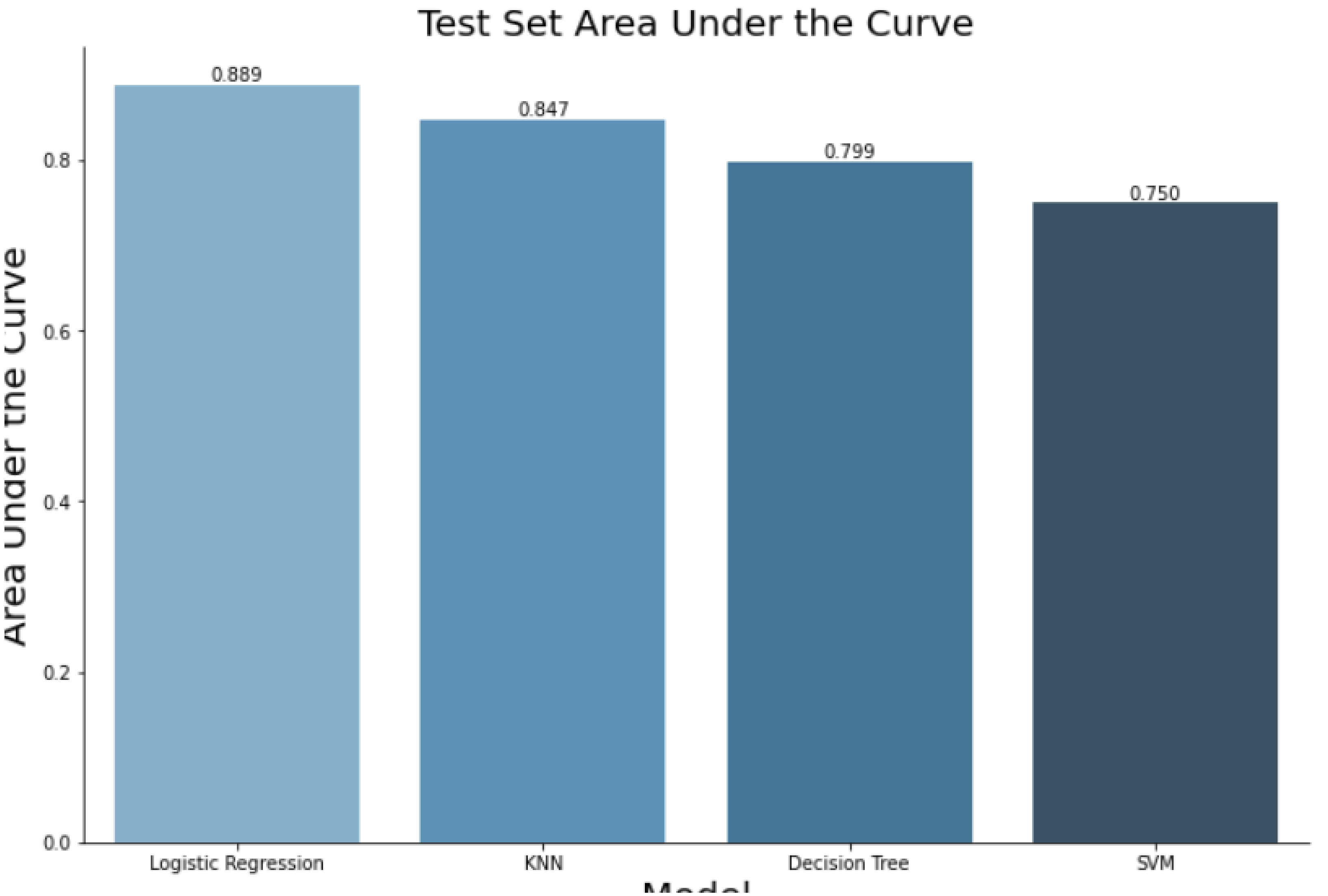
Predictive Analysis (Classification)

Classification Accuracy

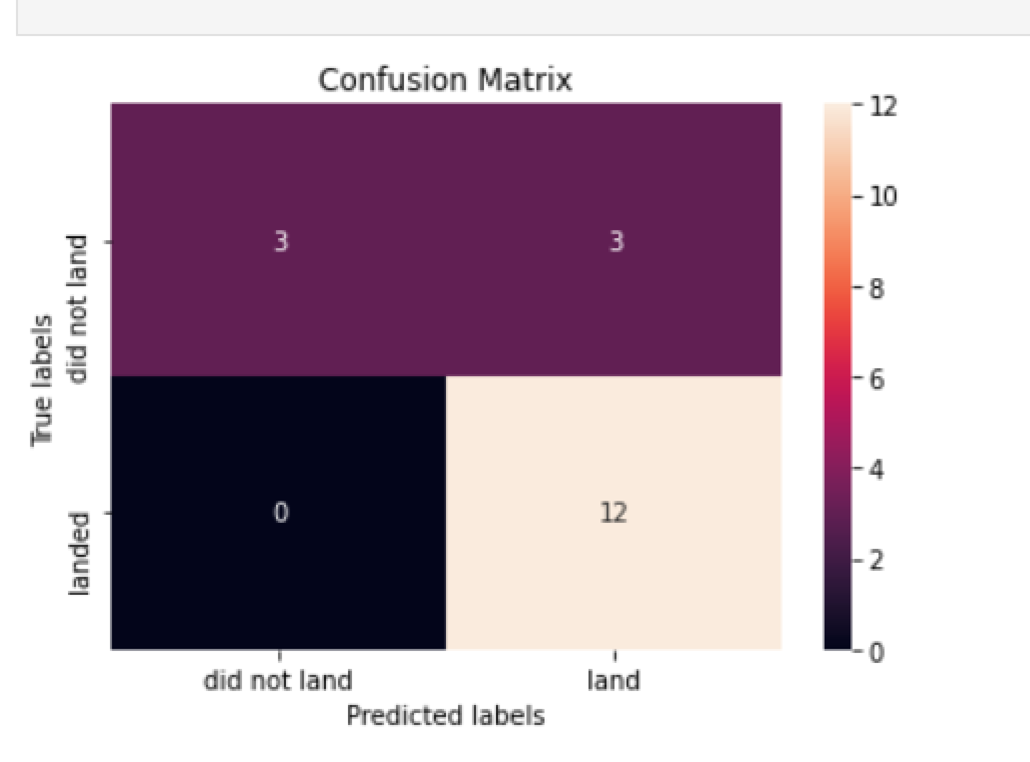
```
1]: data = {'AUC': [glm_auc, svm_auc, tree_auc, knn_auc], 'F1-Score': [glm_f1, svm_f1, tree_f1, knn_f1],
            'Precision': [glm_prec, svm_prec, tree_prec, knn_prec], 'Recall': [glm_rec, svm_rec, tree_rec, knn_rec],
            'Accuracy': [glm_acc, svm_acc, tree_acc, knn_acc]}
    res = pd.DataFrame(data, index=['Logistic Regression', 'SVM', 'Decision Tree', 'KNN']).sort_values(by=['AUC'], ascending=False)
    res.round(3)
```

t[61]:

	AUC	F1-Score	Precision	Recall	Accuracy
Logistic Regression	0.889	0.889	0.8	1.0	0.833
KNN	0.847	0.889	0.8	1.0	0.833
Decision Tree	0.799	0.889	0.8	1.0	0.833
SVM	0.750	0.889	0.8	1.0	0.833


Test Set Area Under the Curve
Area Under the Curve
0.0
0.2
0.4
0.6
0.8
0.889
0.847
0.799
0.750
Logistic Regression
KNN
Decision Tree
SVM
Model

Confusion Matrix



Conclusions

- Accuracy equal for every model
- Auc Best for logistic regression

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!