IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Molefi Molise
18th January 2022

Outline

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

Executive Summary

- Summary of methodologies
    - Data Collection through Web Scrapping and API
    - Data Preparation
    - EDA with Visualization and SQL
    - Data Wrangling
    - Map Visualizations with Folium
    - Machine Learning Predictions
- Summary of all results
    - EDA Results
    - Predictive Analytics Results

Introduction

- Project Background and context
  - 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 due to the fact that SpaceX can reuse the first stage. It's becomes obvious from this statement that to reduce costs, successful landing of first stage is crucial since they can be reused in next missions. In this project, the aim is to predict if the Falcon 9 first stage will land successfully.
- Problems you want to find answers
  - What are the determining factors for a successful landing of the first stage?
  - Which conditions guarantee a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Two techniques were used for data collection

  1. Web Scrapping: Data used in this project was scrapped from Wikipedia website. BeautifulSoup was used to extract rows of data from a table on the website.

  2. Data Collection through API: SpaceX API was used to retrieve data about different rocket launches. The data was received in a form of json and loaded into pandas dataframe using .json_normalize() function.

```
In [7]: # use requests.get() method with the provided static_url
        response = requests.get(static_url).text
        # assign the response to a object
```

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

```
In [8]: # Use BeautifulSoup() to create a BeautifulSoup object fr
        soup = BeautifulSoup(response, 'html.parser')
```

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

Data Collection – SpaceX API

Link to Notebook

Get data using spaceX URL

Decode contents into json & load into dataframe

Filter dataframe leaving falcon 9 data only

Clean data

Getting data from API

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS032
```

```
response = requests.get(static_json_url)
```

Decode contents into Json

```
# Use json_normalize meethod to convert the json result in
json_data = response.json()
data = pd.json_normalize(json_data)
```

Filter leaving falcon 9 data

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data["BoosterVersion"]=="Falcon 9"]
```

Data Collection - Scraping

Link to Notebook

Get html data using wikipedia URL

Convert response to BeautifulSoup object

Create dataframe from extracted table rows

Clean data

Getting HTML Data from URL

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

Create BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup objec
soup = BeautifulSoup(response, 'html.parser')
```

Create dataframe

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

[Link to Notebook]

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, `True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad. `True ASDS` means the mission outcome was successfully landed on a drone ship `False ASDS` means the mission outcome was unsuccessfully landed on a drone ship. This outcome variable will be used to generate a target label where 1 means a successful landing and 0 means unsuccessful landing.

Click to add text

```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = df.Outcome.map(lambda x: 0 if x in bad_outcomes else 1)
landing_class
```

```python
df['Class']=landing_class
df[['Class']].head(8)
```

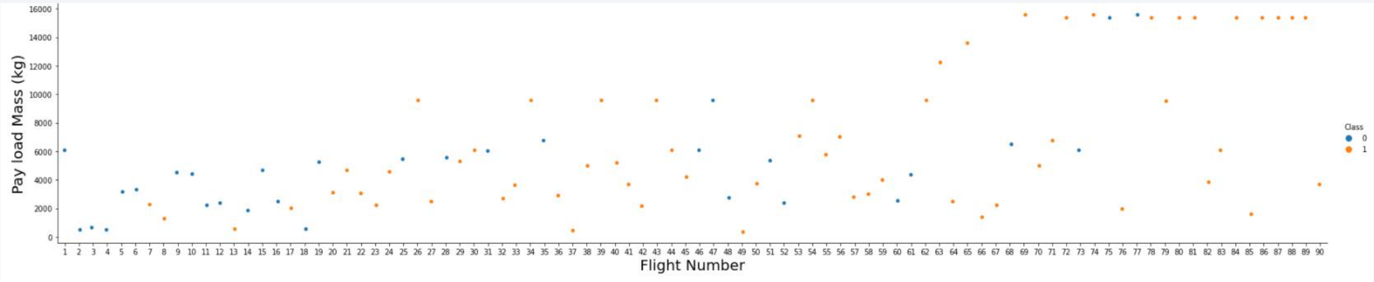
]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

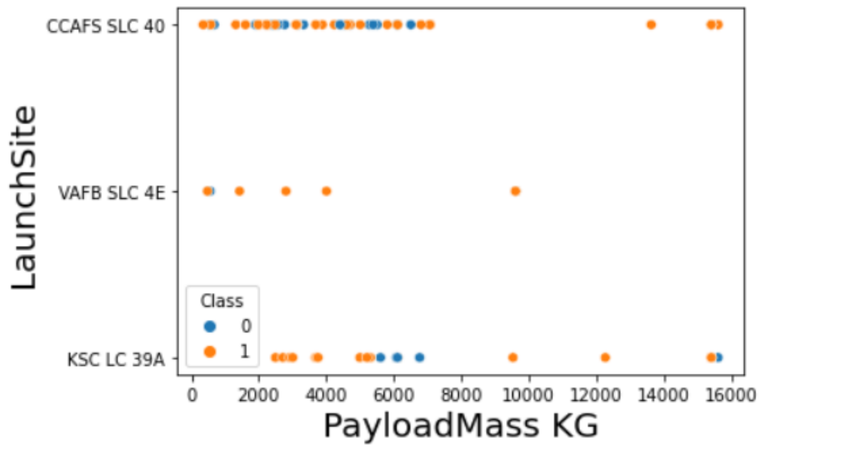
EDA with Data Visualization

Link to Notebook

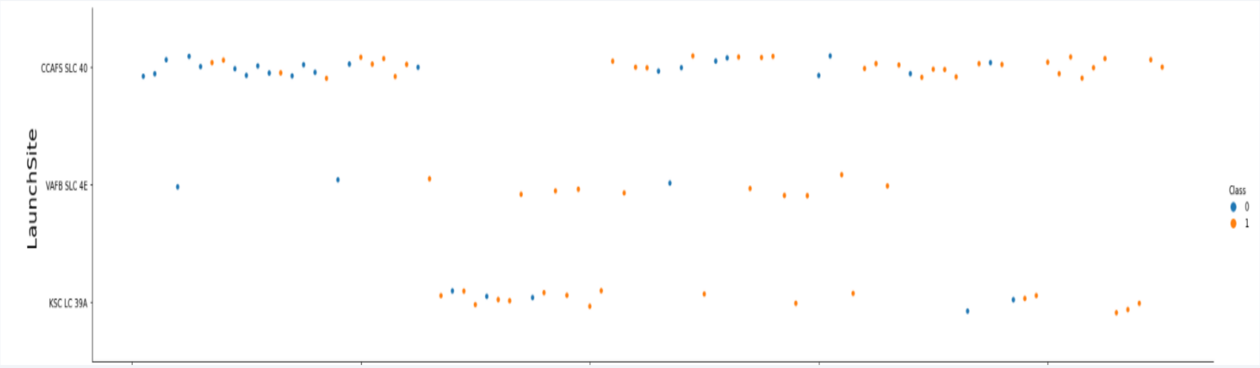
Different visualizations were used to determine how different variables influence the launch outcome.



`FlightNumber` vs. `PayloadMass`



Payload and Launch Site



Flight Number and Launch Site

EDA with SQL

Link to Notebook

- In this section SQL queries were used to gather basic info about spacex data store in IBM DB2.
- The SQL was used to find:
  - Names of unique launch sites
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - 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

Build an Interactive Map with Folium

- To visually understand where first stage landed Folium maps were used to show different launch sites and also use red and green markers to indicate successful and unsuccessful landing of different missions.
- To build a folium map, coordinates of the of the map are provided together with the zoom level.
- A map can have markers and circles which can be used to pin locations on the map.

Build a Dashboard with Plotly Dash

An interactive dashboard was built with Flask and Dash

The dashboard is comprised of HTML components and Plotly graphs

Graphs

- Pie chart showing the total number of launches per site
- Scatter plots showing relationship between outcome and payload mass (KG) for different Booster versions.

Predictive Analysis (Classification)

- Model building
  - Dataset was divided into train and test sets
  - Different models were build e.g SVM, KNN, Decision Trees,logistic regression

- Model Enhancement
  - GridSearch CV was used to model parameters that produce the highest score
  - Accuracy score and Classification matrix was used to evaluate models

- Model Selection
  - All four models were compared and the best model was a decision tree.

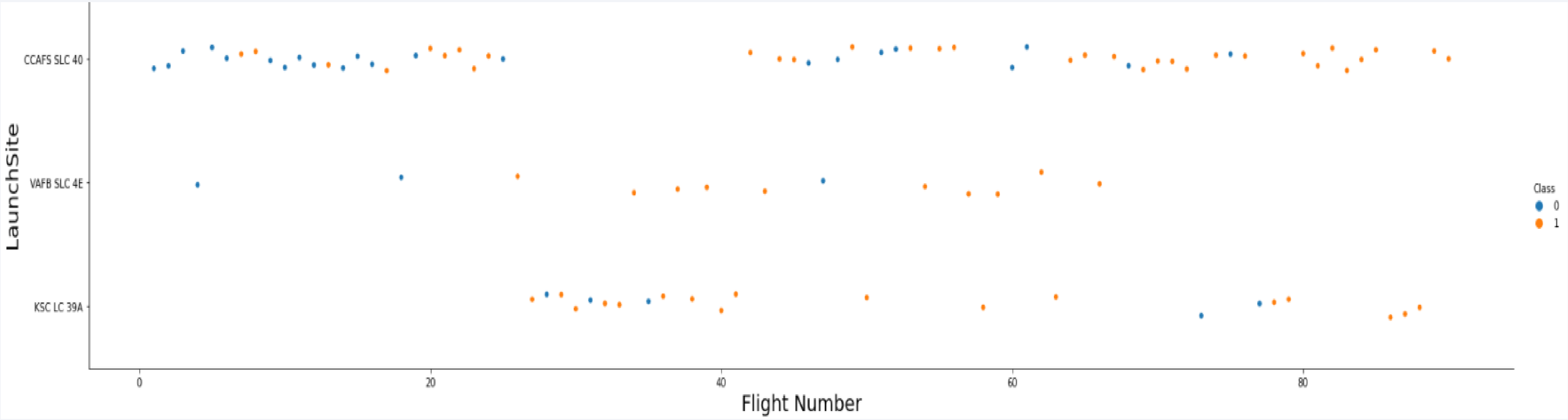
Results

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

Section 2

Insights drawn from EDA

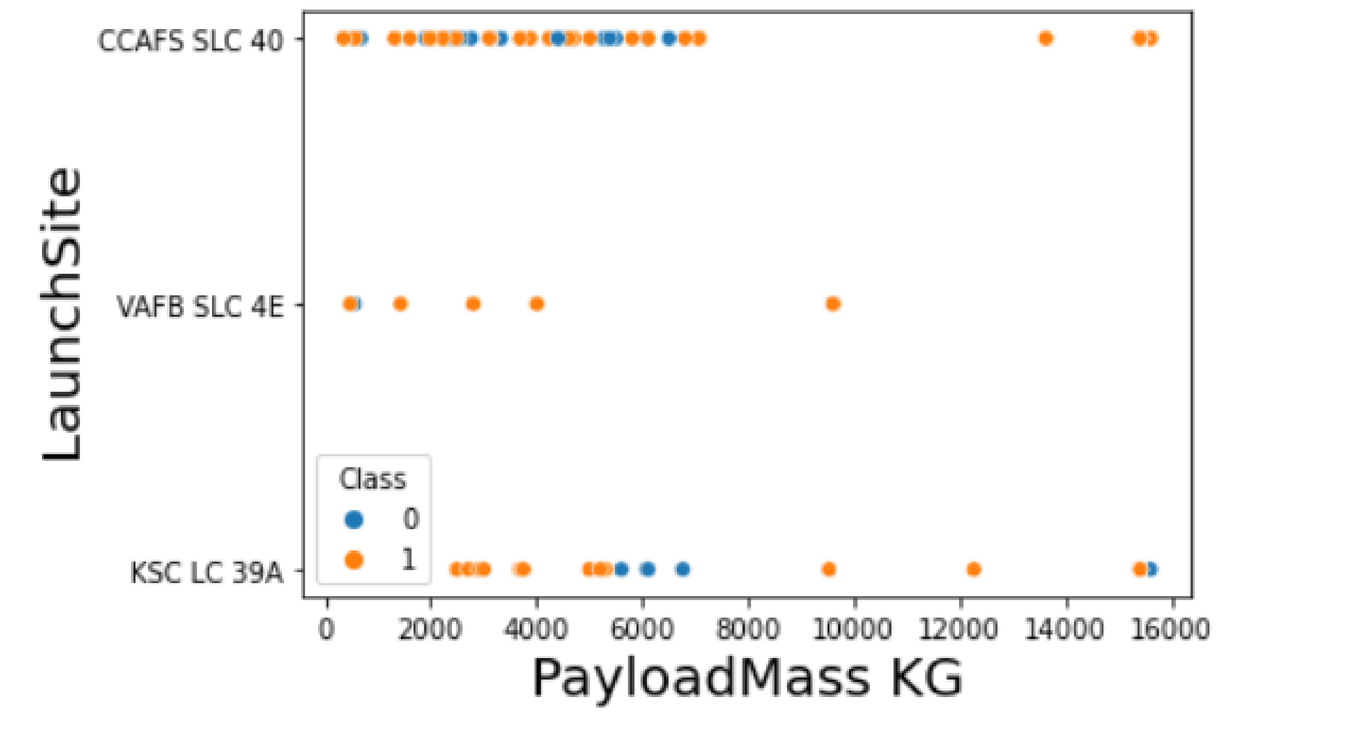
Flight Number vs. Launch Site



CCAFS SLC 40 had most failed launches

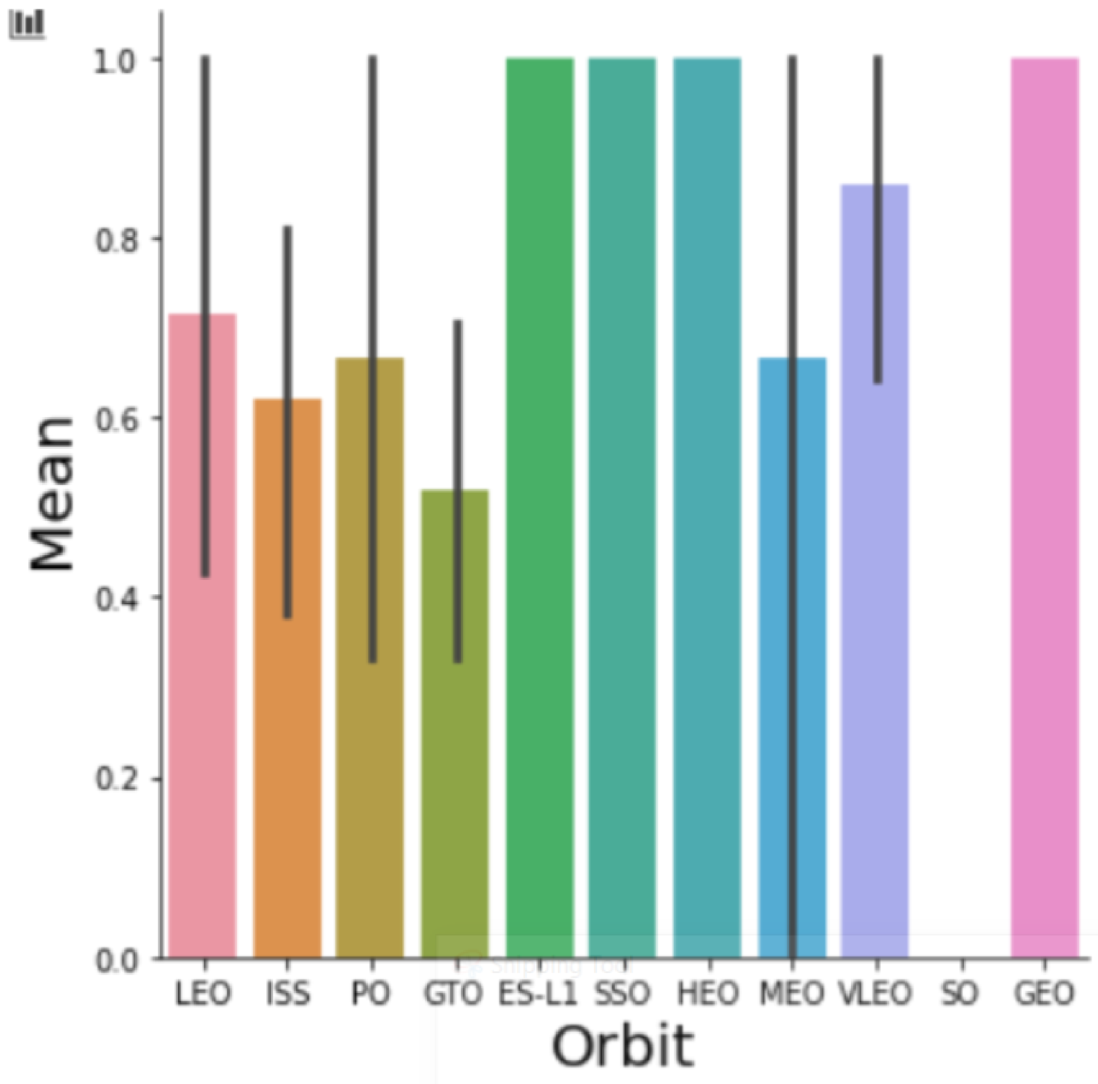
Payload vs. Launch Site

- Launches that have high PayloadMass have a higher chance of being successful.



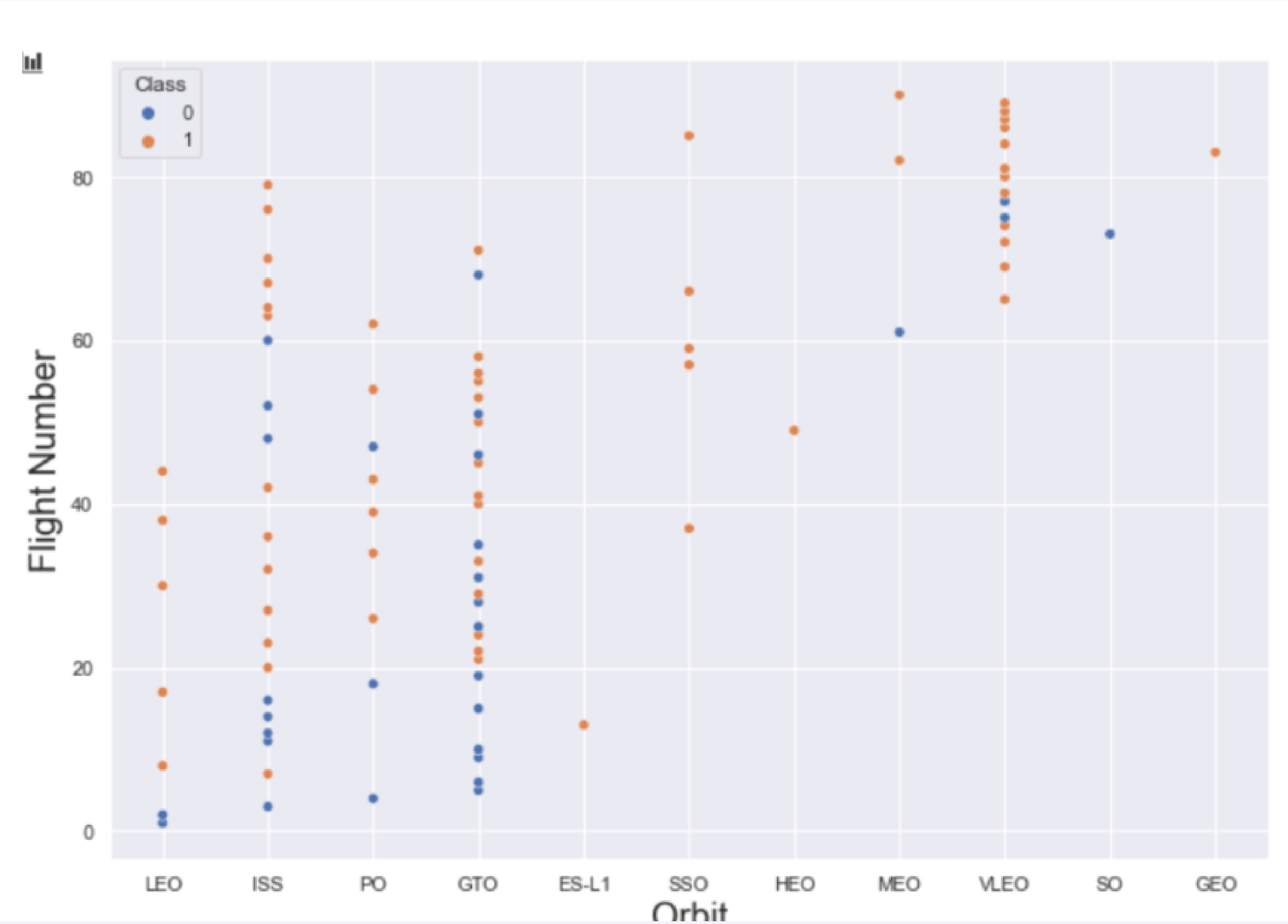
Success Rate vs. Orbit Type

- ES-L1,SSO,HEO,GEO have the highest success rate.



Flight Number vs. Orbit Type

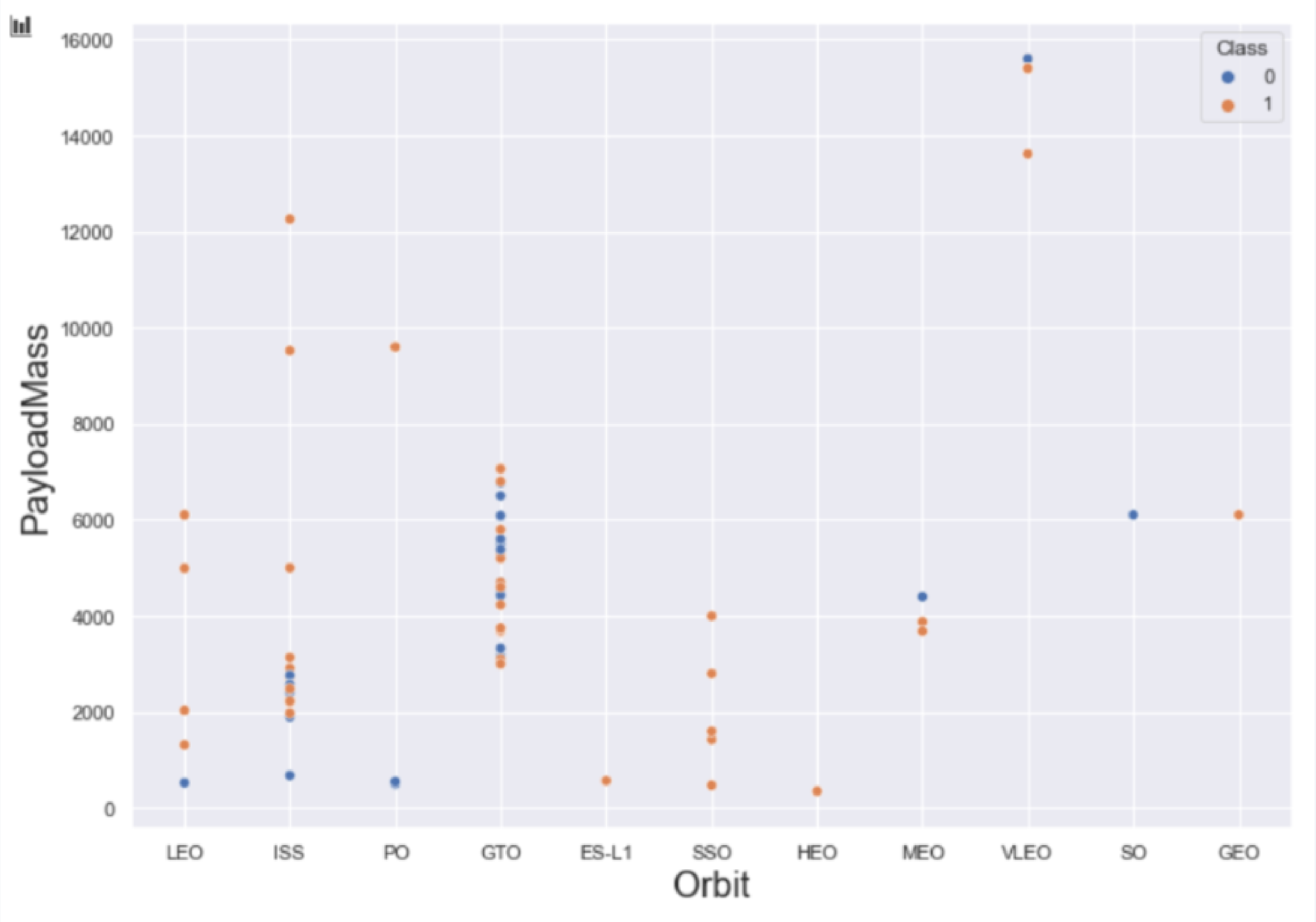
- Flight numbers over 40 are mostly successful. Also SSO,HEO,MEO,VLEO,SO,GEO have high flight numbers



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

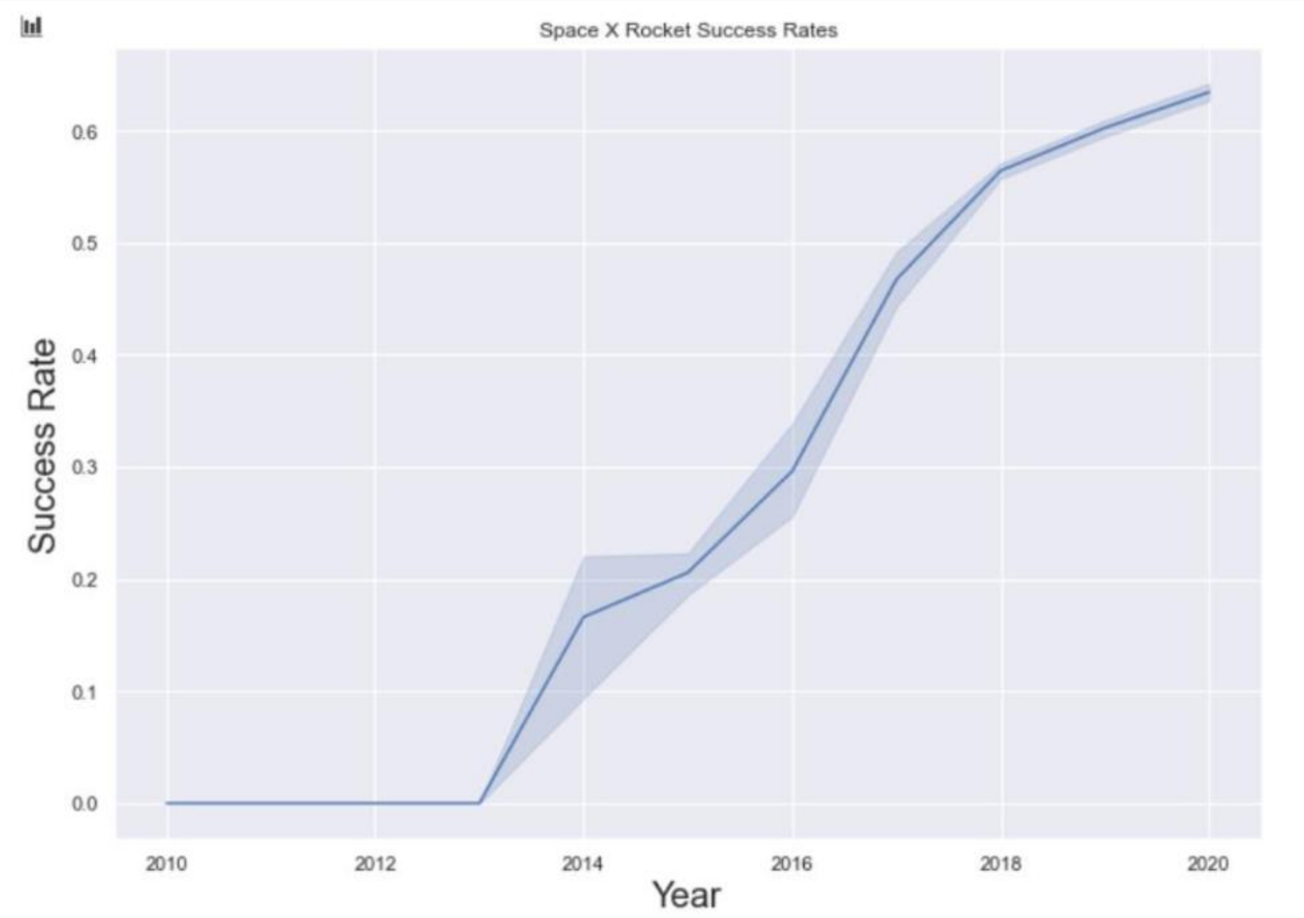
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- SpaceX Dataset consisted of 4 launch sites shown on the right

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

Launch Site Names Begin with 'CCA'

- % records that have launch site names that begin with 'CCA'

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

- NASA carried a total payload of 45596KG

total	customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

Average Payload mass by F1 v1.1 is 2928

total	booster_version
2928	F9 v1.1

First Successful Ground Landing Date

- First successful landing date is 2010-06-04

1
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Landing _Outcome
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

	Successful_Mission_Outcomes	Failure_Mission_Outcomes
0	100	1

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

	Booster_Version	Maximum Payload Mass
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
...	...	...
92	F9 v1.1 B1003	500
93	F9 FT B1038.1	475
94	F9 B4 B1045.1	362
95	F9 v1.0 B0003	0
96	F9 v1.0 B0004	0

97 rows × 2 columns

2015 Launch Records

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

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 FT B1029.1	VAFB SLC-4E	Success (drone ship)
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
March	F9 FT B1021.2	KSC LC-39A	Success (drone ship)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1029.2	KSC LC-39A	Success (drone ship)
June	F9 FT B1036.1	VAFB SLC-4E	Success (drone ship)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
August	F9 FT B1038.1	VAFB SLC-4E	Success (drone ship)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
October	F9 B4 B1041.1	VAFB SLC-4E	Success (drone ship)
October	F9 FT B1031.2	KSC LC-39A	Success (drone ship)
October	F9 B4 B1042.1	KSC LC-39A	Success (drone ship)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

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

- 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

	Successful Landing Outcomes Between 2010-06-04 and 2017-03-20
0	34

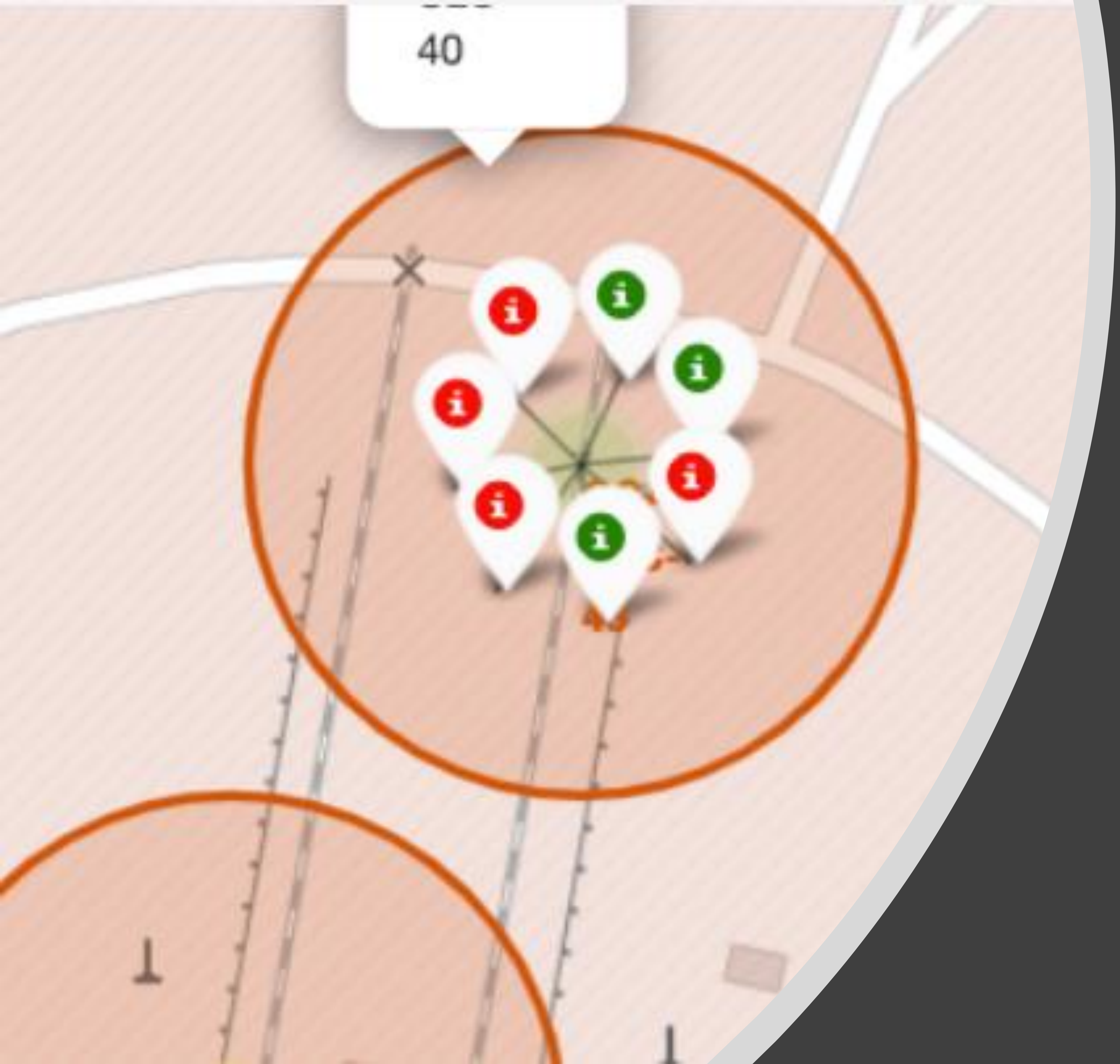
Section 4

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

Some Launch sites are in California while others are situated in Florida

<Folium Map Screenshot 2>

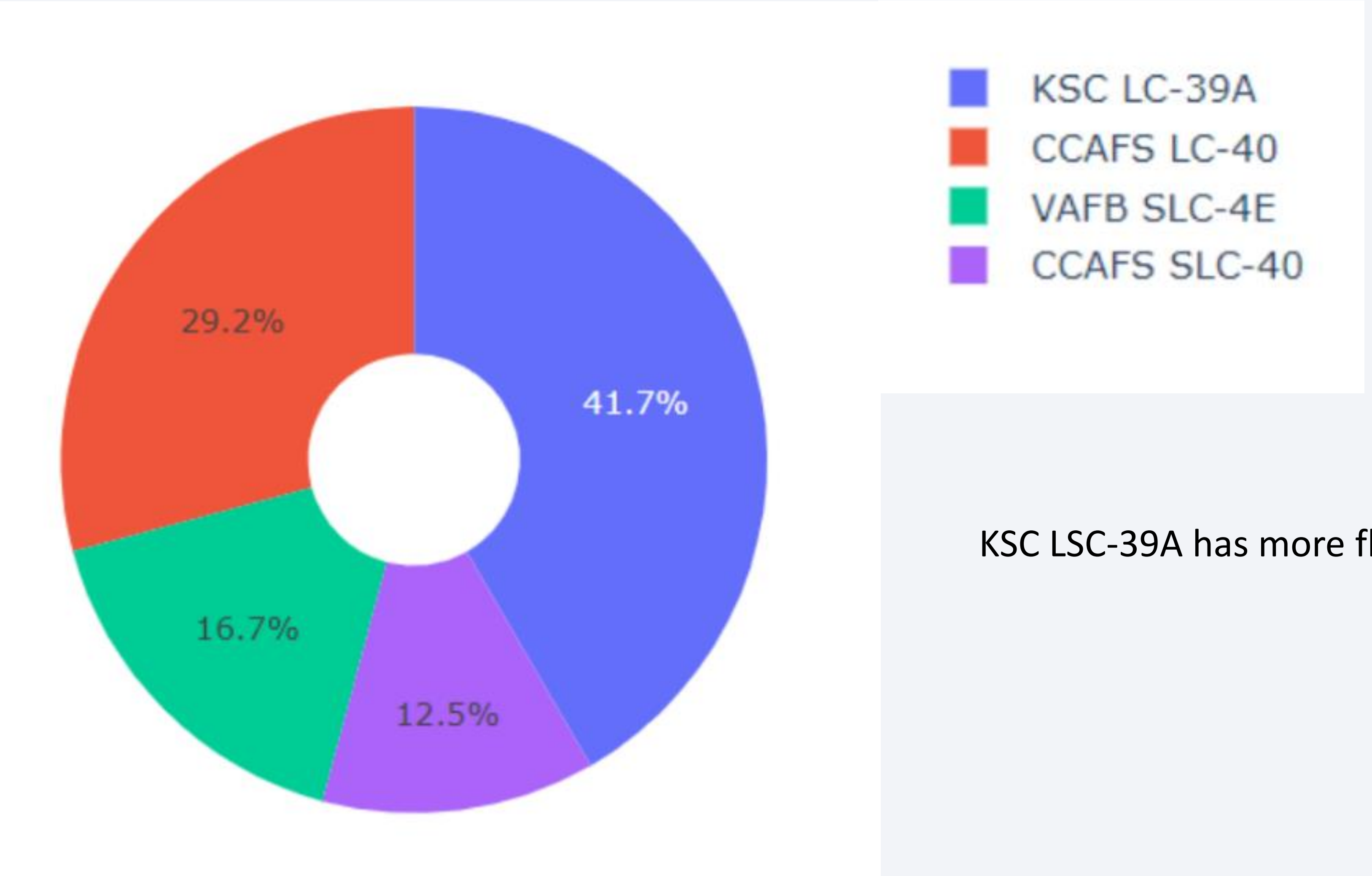


<Folium Map Screenshot 3>

Section 5

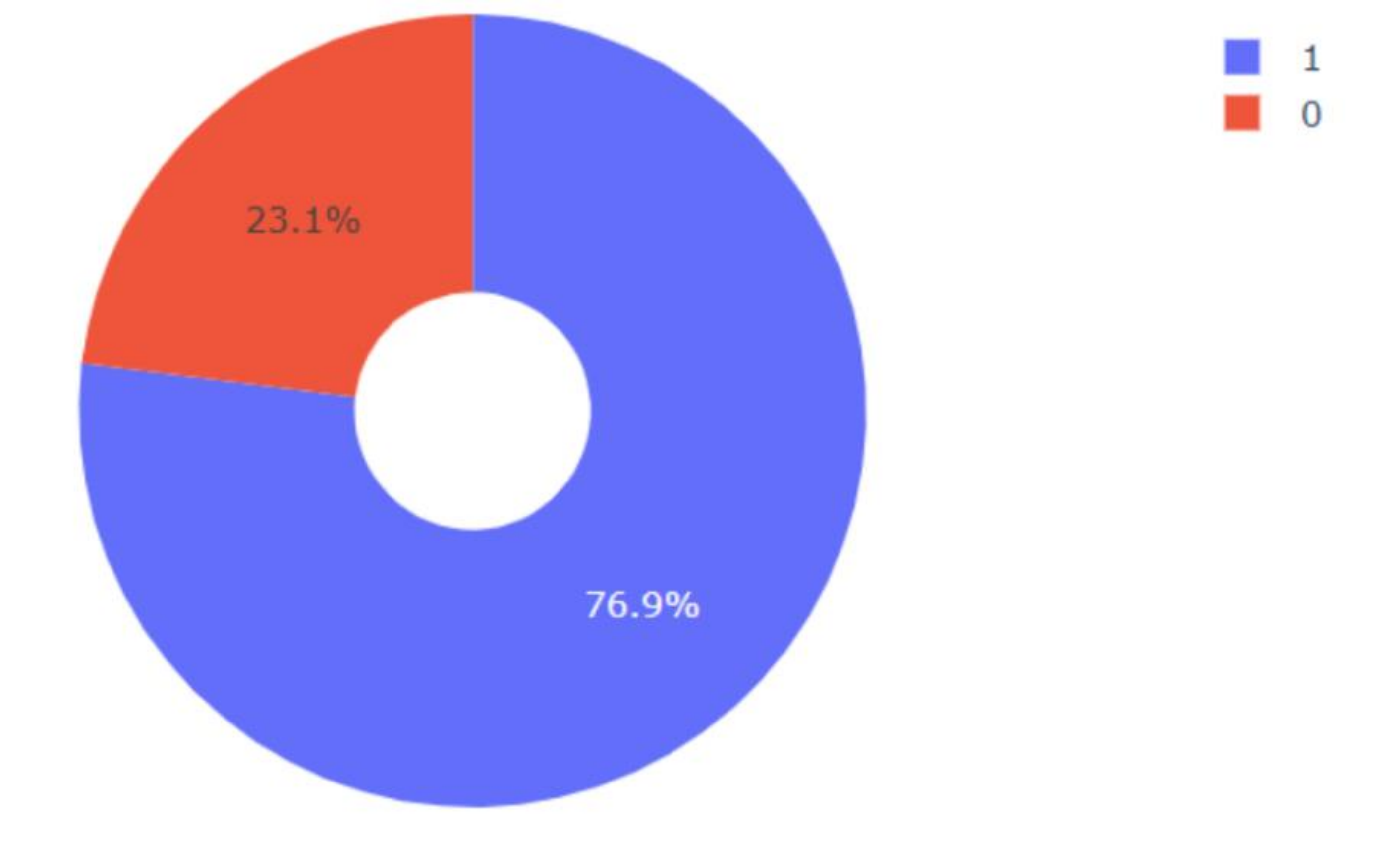
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



KSC LSC-39A has more flight launches

<Dashboard Screenshot 2>

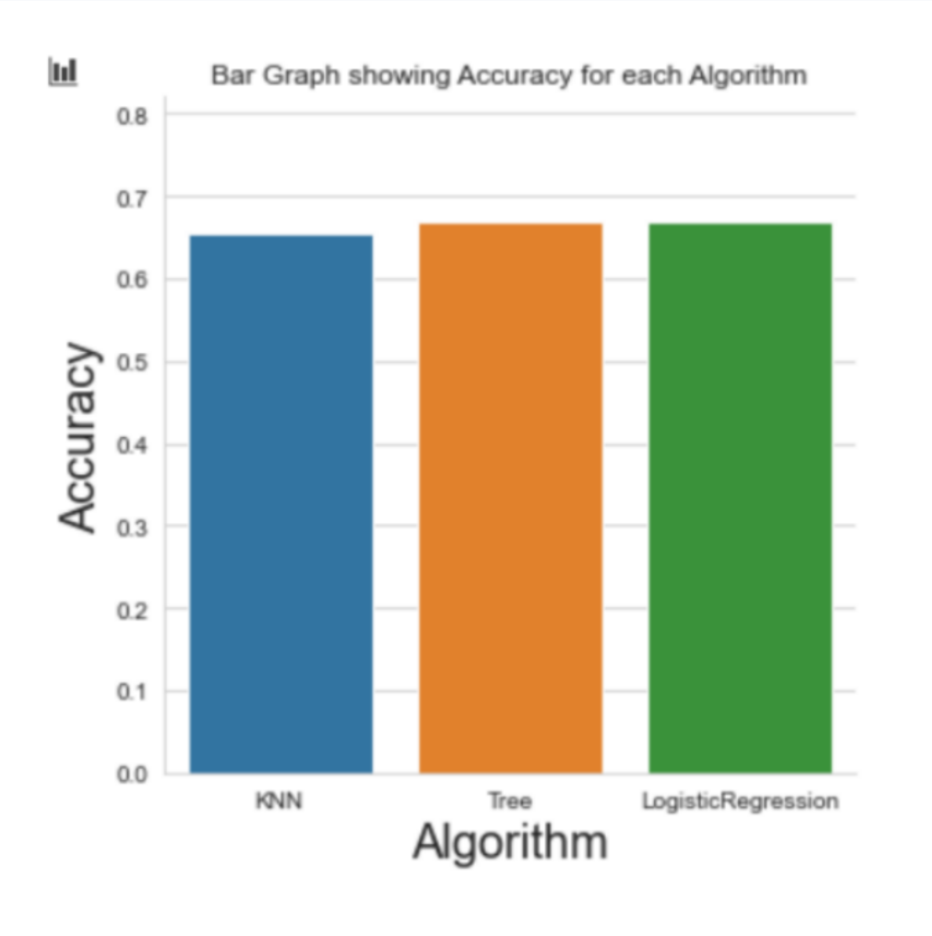


KSC LSC-39A has 76.9& successful launches

Section 6

Predictive Analysis (Classification)

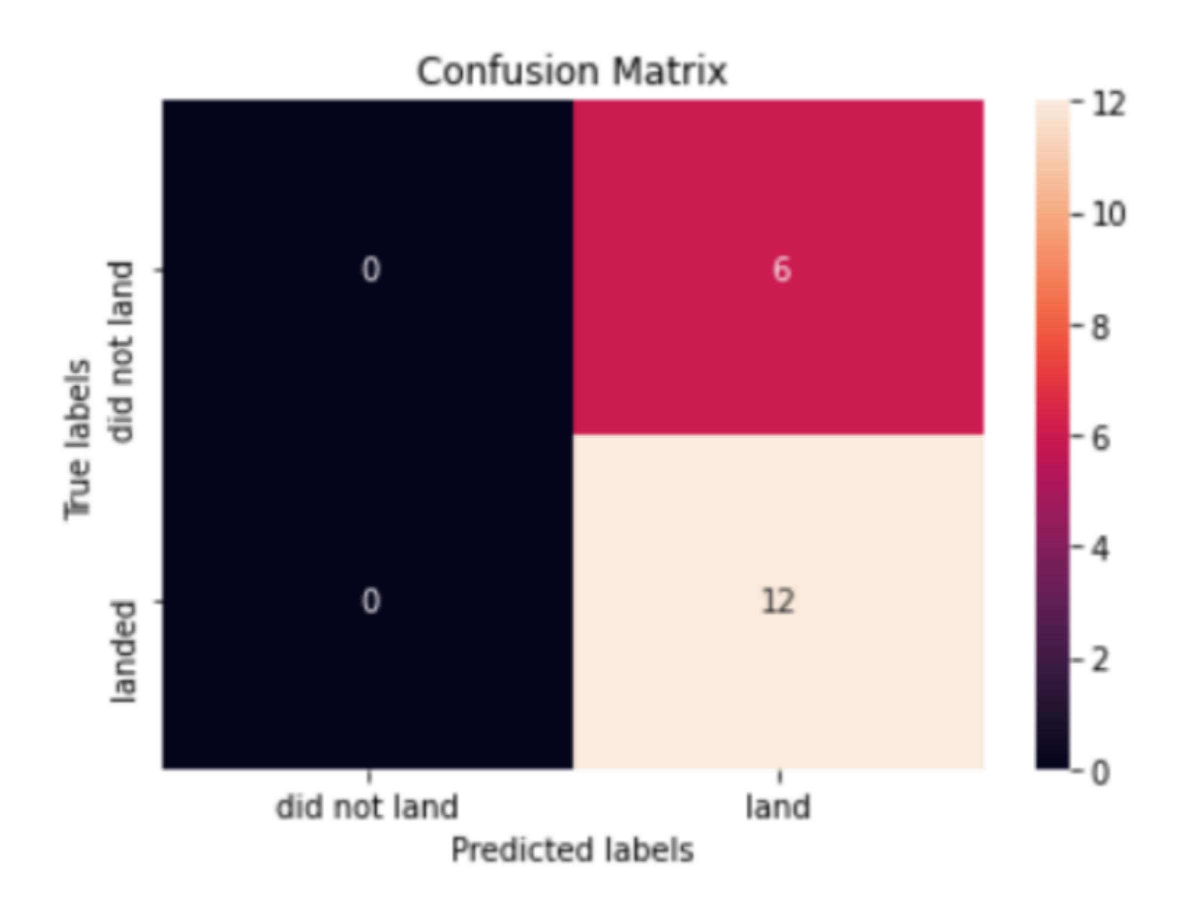
Classification Accuracy



	Accuracy	Algorithm
0	0.653571	KNN
1	0.667857	Tree
2	0.667857	LogisticRegression

From the table we can see that Logistic Regression is the best algorithm for this problem

Confusion Matrix



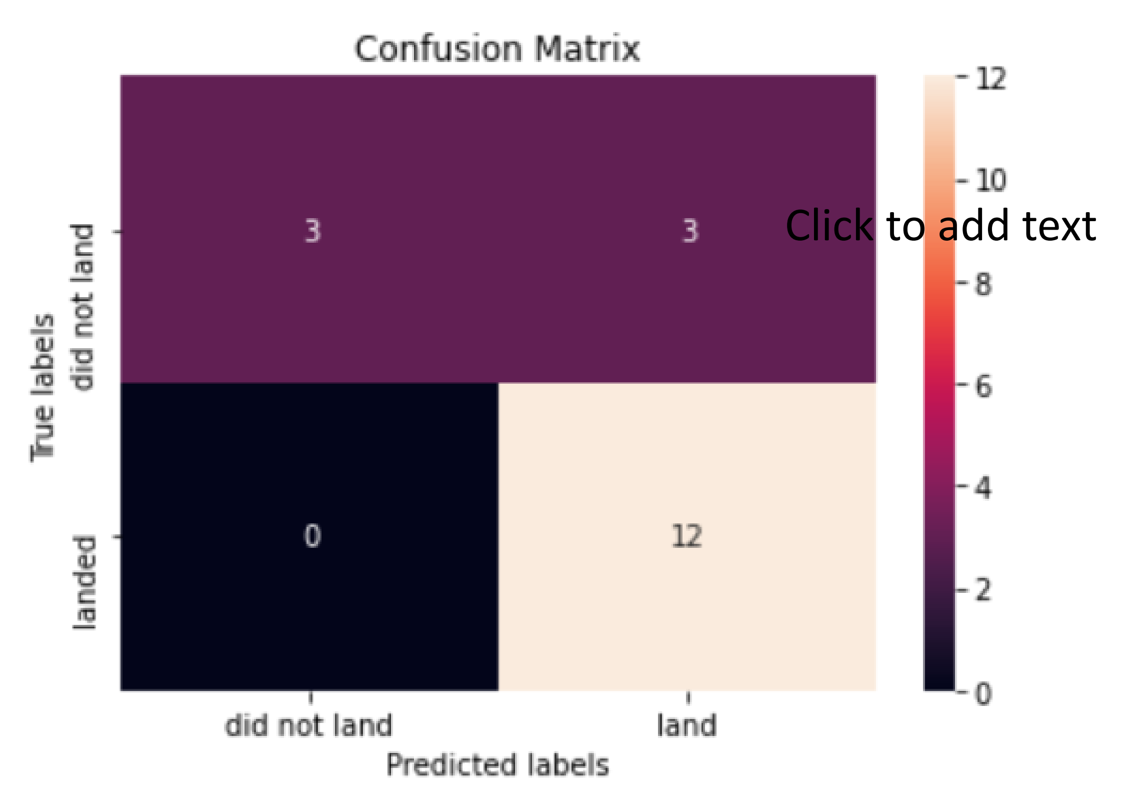
Confusion Matrix for Logistic Regression

Conclusions

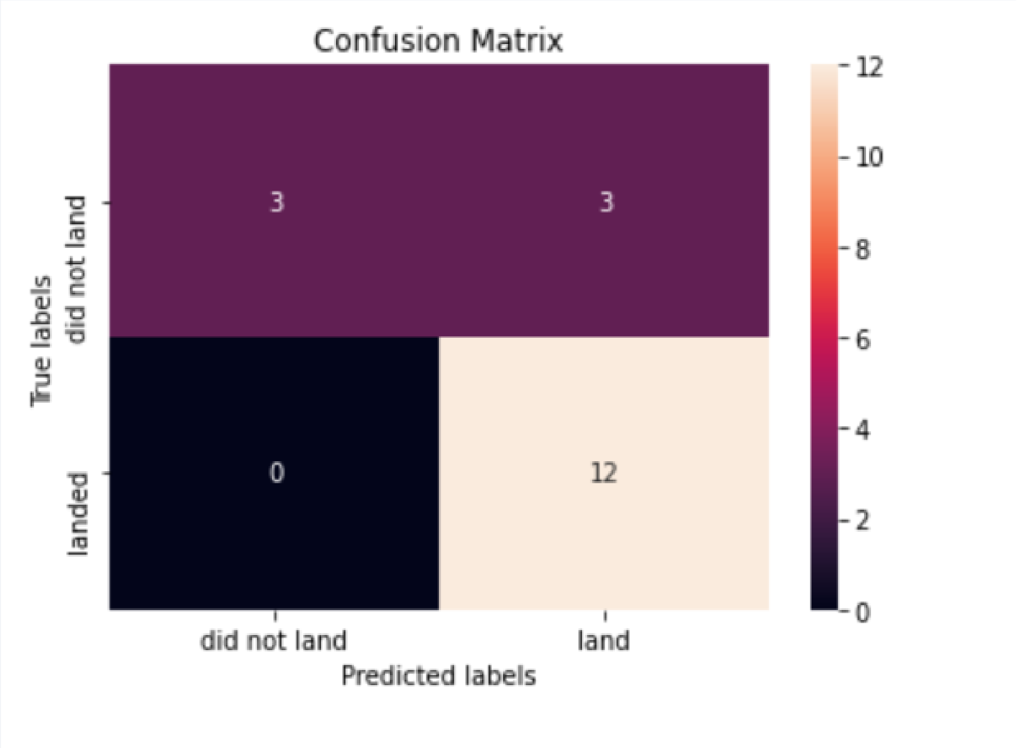
- According to my analysis Logistic Regression is the best model for predicting landing outcome
- KSC LSC-39A is the best launch site

Appendix

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Click to add text



Thank you!