Data Science Capstone Project

Julienne Manalo
December 20, 2021

Outline

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

Executive Summary

Executive Summary

- The following is the collected data from the public SpaceX API and the SpaceX Wikipedia pages.

- The data is explored by using tools such as SQL, visualization, folium maps, and dashboards.

- The four machine learning models that were produced for this: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All have given similar results with accuracy rate of approximately 83.33%. All models over-predicted successful landings, which means that more data is needed for better accuracy.

Introduction

Introduction

Background:

- Commercial Space Age is Here
- Space X has best pricing ($62 million vs. $165 million USD)
- Largely due to ability to recover part of rocket (Stage 1)
- Space Y wants to compete with Space X

Problem:

- Space Y tasks us to train a machine learning model to predict a successful Stage 1 recovery

Methodology

Methodology

Executive Summary

- The overview of Data Collection, Wrangling, Dashboard Visualization, and Model Methods
- 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

It involves a combination of API requests from both the Space X public API and web scraping data from a table in Space X's Wikipedia entry.

The Data Columns taken from SpaceX API were the following:

- 'FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit', 'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude'

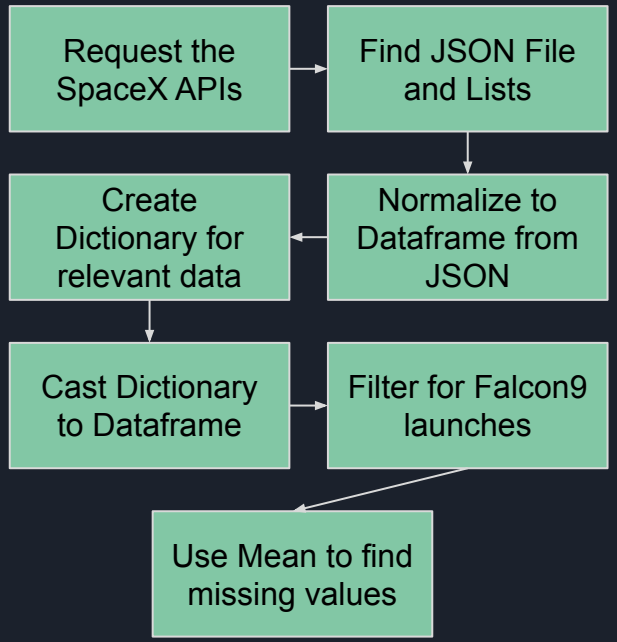
And here are the Webscraping Data Columns taken from its Wikipedia page:

- 'FlightNum', 'LaunchSite', 'Payload', 'PayloadMass', 'Orbit', 'Customer', 'LaunchOutcome', 'VersionBooster', 'BoosterLanding', 'Date', 'Time'

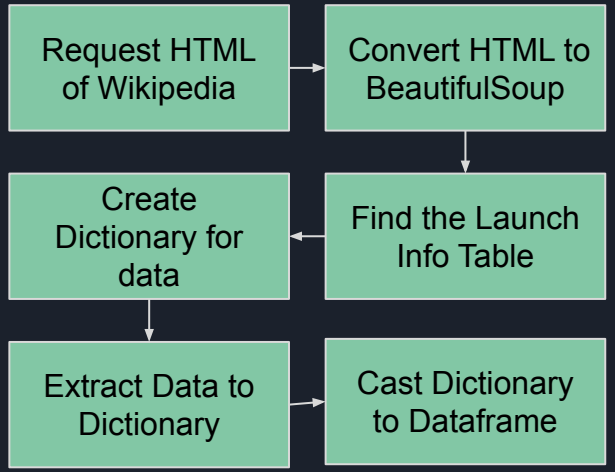
Flow charts located in the next slide.

Data Collection

API



Webscraping



GitHub Links:

- API
- Webscraping

Data Wrangling

- Created a training label with landing outcomes:
  - Successful = 1 | Failure = 0.
- Outcome column has two components: ‘Mission Outcome’, and ‘Landing Location’
- New training label column class with a value of 1 if ‘Mission Outcome’ is True and 0 if False.
- Value Mapping:
  - True ASDS, True RTLS, & True Ocean – set to -> 1
  - None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

GitHub Link

EDA with Data Visualization

- It's the Exploratory Data Analysis that performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
- Plots Used:
  - 'FlightNumber vs. PayloadMass', 'FlightNumber vs. LaunchSite', 'PayloadMass vs. LaunchSite', 'Orbit vs. SuccessRate', 'FlightNumber vs. Orbit', 'Payload vs Orbit', and 'SuccessYearlyTrend'
- Scatter plots, line charts, and bar plots were used to compare relationships between variables in order to decide if a relationship exists. It's so that we know if they could be used in training the machine learning model.

GitHub Link

EDA with SQL

- Data set is loaded into an IBM DB2 Database.
- It's then queried by using the SQL Python integration so we can get a better understanding of the dataset.
- The information that's queried are launch site names, mission outcomes, various pay load sizes of customers and booster versions, landing outcomes. etc.

GitHub Link

Build an Interactive Map with Folium

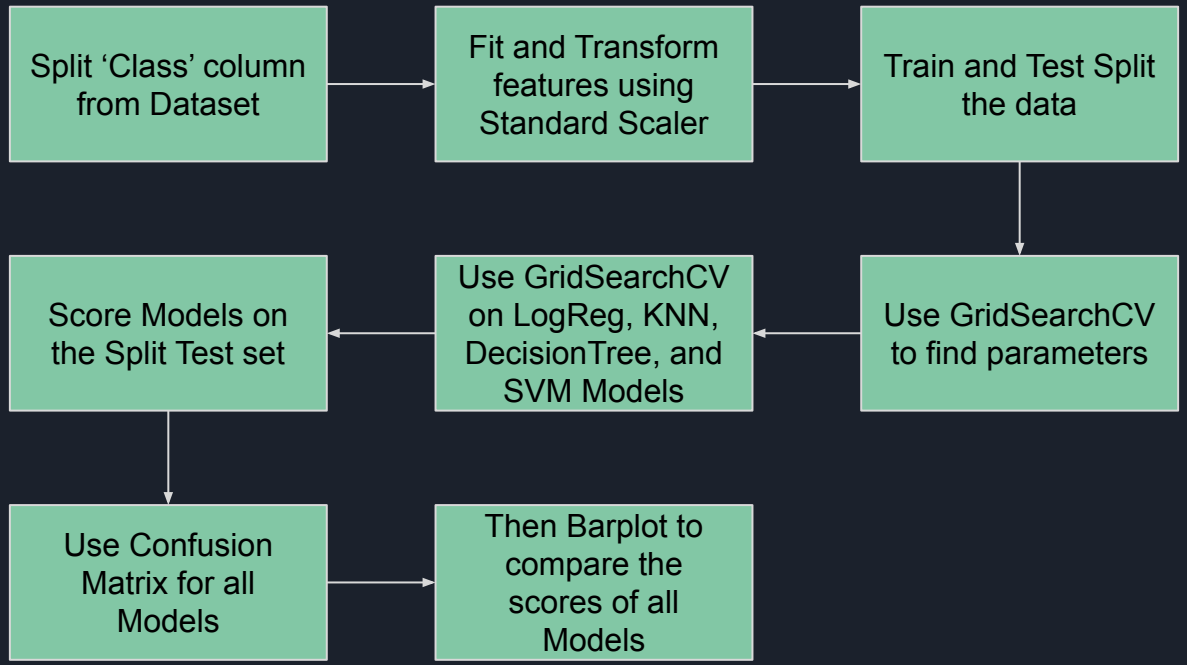
- The purpose of Folium maps is to mark Launch Sites, successful and unsuccessful landings, as well as a proximity examples to several key locations like the Railway, Highway, Coast, and City.
- This is so it would allow us to understand why launch sites may be located where they are. It also visualizes the successful landings that are relative to the location.

GitHub Link

Build a Dashboard with Plotly Dash

- The Dashboard includes both a pie chart and a scatter plot.
- The Pie chart can be selected to visualize the distribution of successful landings across all launch sites and can also be selected to show individual launch site success rates.
- The Scatter plot takes two inputs: all sites or an individual site, and then a payload mass with a slider inbetween. It can help us see how success rates varies across the different launch sites, payload mass, and booster version category.

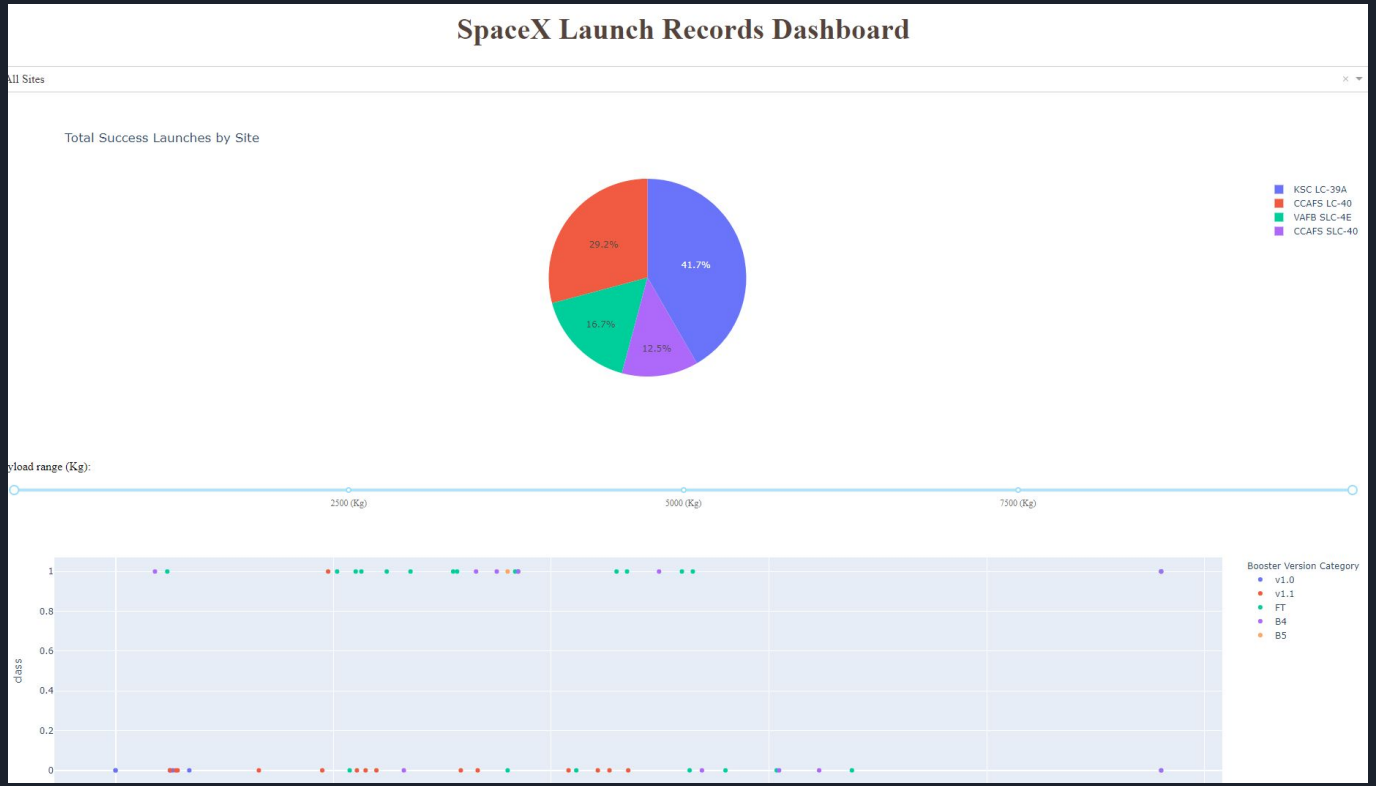
Predictive Analysis (Classification)



GitHub Link

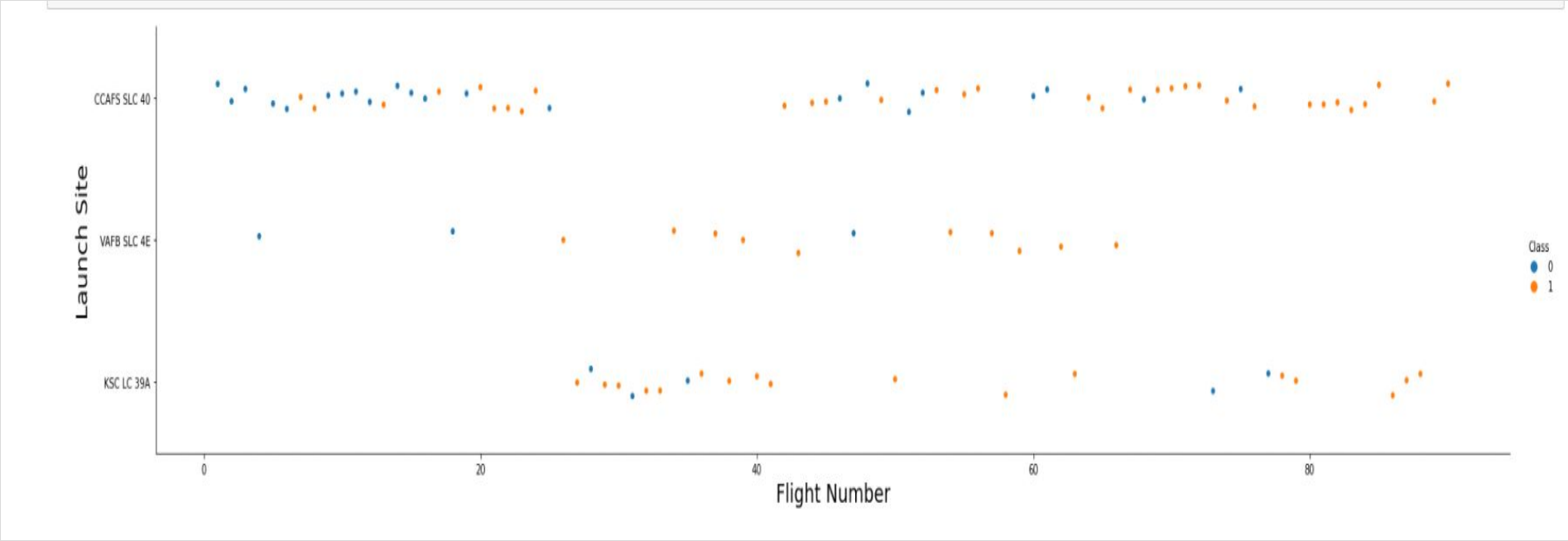
Results

Results

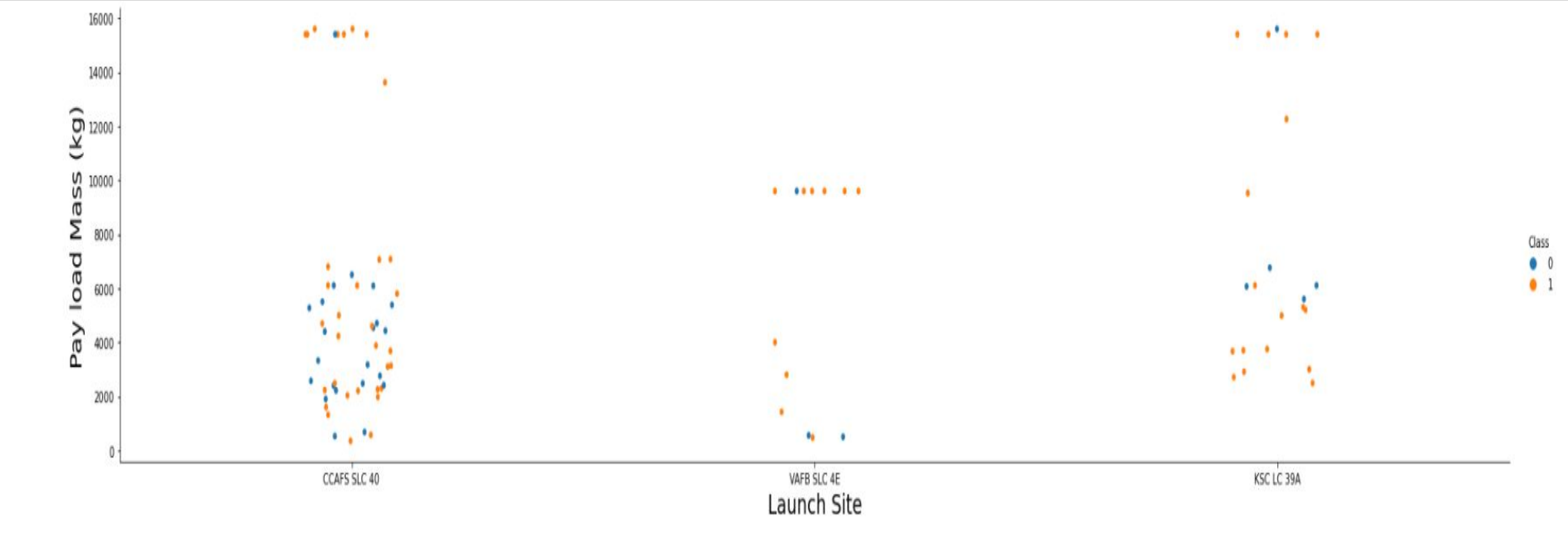

SpaceX Launch Records Dashboard
All Sites
Total Success Launches by Site
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
29.2%
41.7%
16.7%
12.5%
yload range (Kg):
2500 (Kg)
5000 (Kg)
7500 (Kg)
class
1
0.8
0.6
0.4
0.2
0
Booster Version Category
v1.0
v1.1
FT
B4
B5

Insights Drawn from EDA

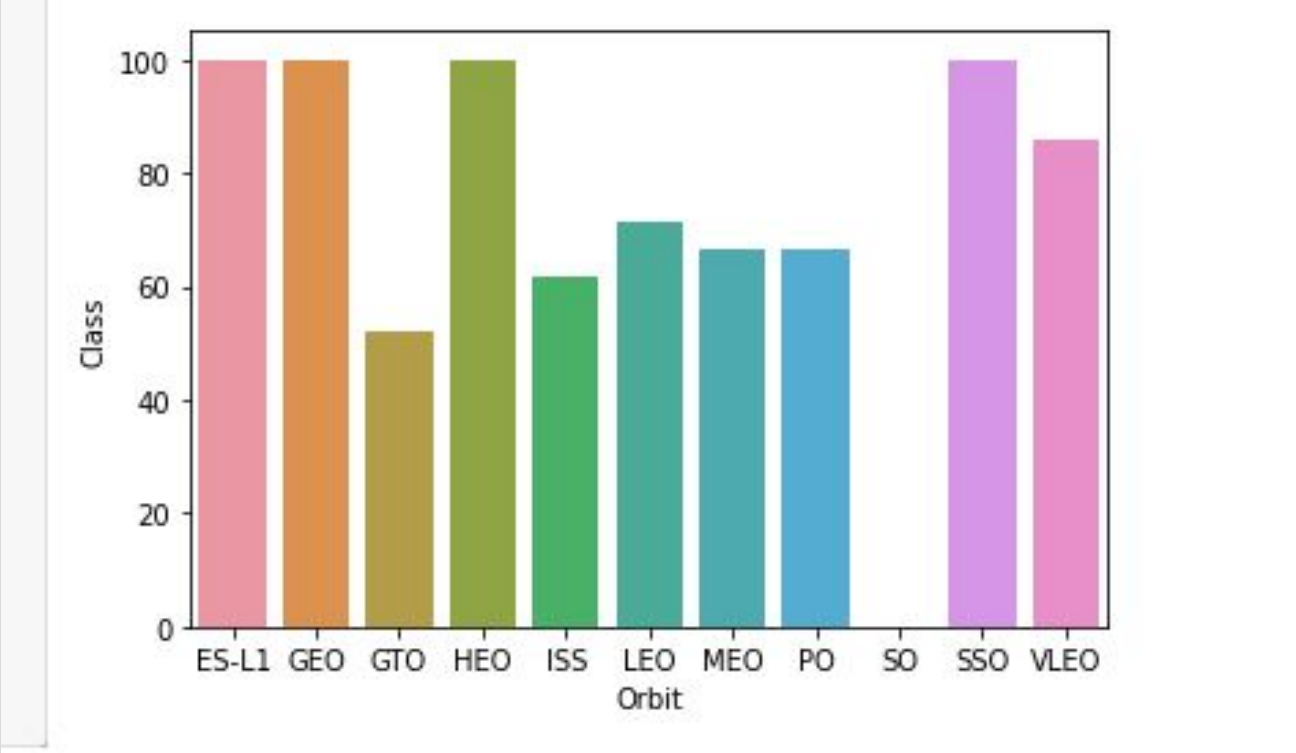
Flight Number vs. Launch Site



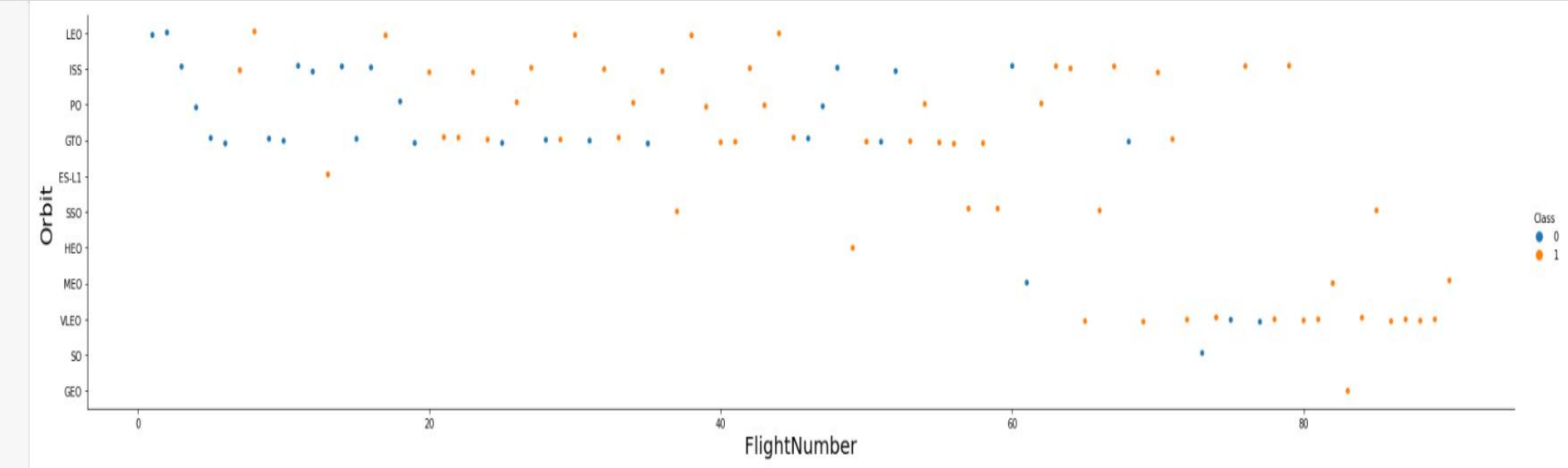
Payload vs. Launch Site



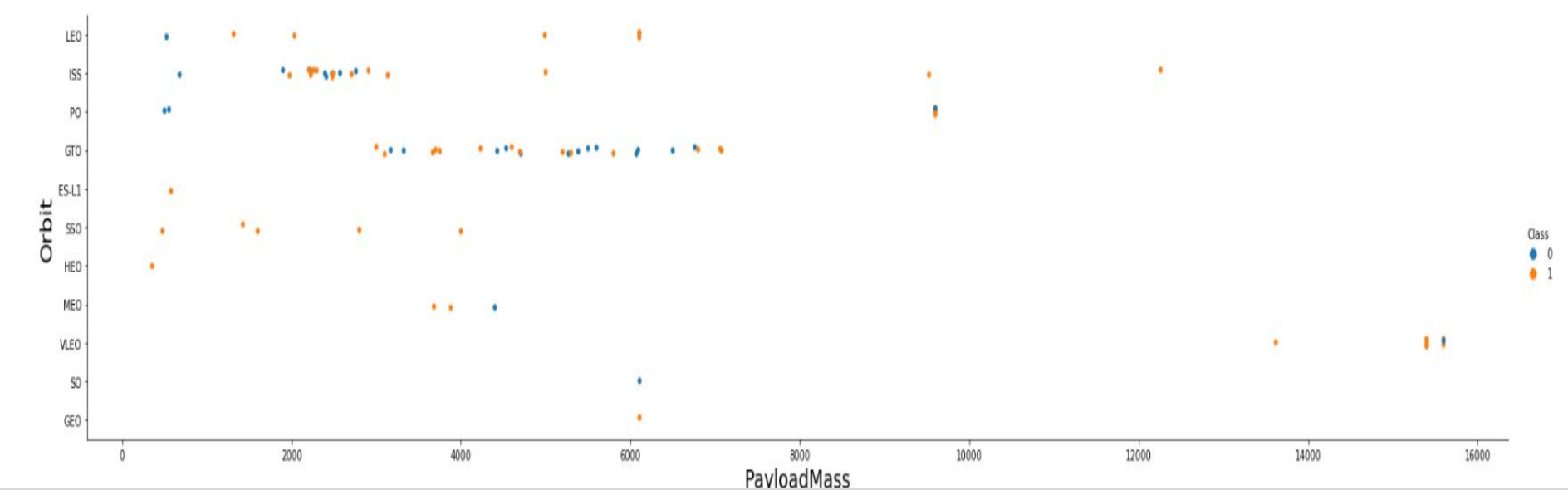
Success Rate vs. Orbit Type



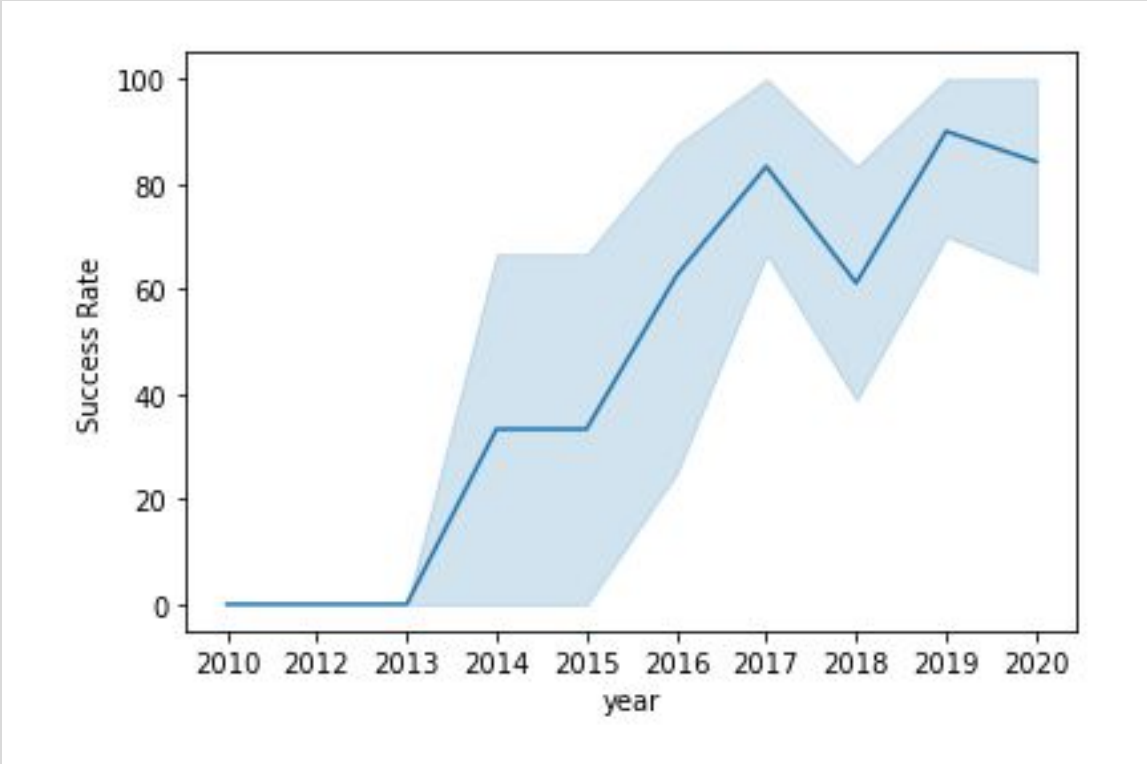
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Display the names of the unique launch sites in the space mission

```
In [11]: %sql select DISTINCT LAUNCH_SITE from SPACEXDATASET
 * ib[redacted]baa7
Done.
```

Out[11]:

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

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
[10]: %sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
```

[redacted]

Done.

Out[10]:

DATE	time__utc_	booster_version	launch_site	payload	INTEGER	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

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(INTEGER) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

[redacted]

Done.

SUM
45596

Average Payload Mass by F9 v1.1

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

```
In [21]: %sql select avg(INTEGER) as Average from SPACEXDATASET where booster_version like 'F9 v1.1%'
 * [redacted]datab
Done.
```

Out[21]:

average
2534

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
In [13]: %sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

[redacted]

```
Done.
```

Out[13]:

DATE
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[18]: %sql Select booster_version from SPACEXDATASET where LANDING__OUTCOME ='Success (drone ship)' and INTEGER BETWEEN 4000 AND 6000
```

```
 * ibm_db [redacted]375/bludb
Done.
```

Out[18]:

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
[15]: %sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm[redacted]loud:30
Done.
```

Out[15]:

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

Boosters Carried Maximum Payload

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

In [21]:
```
%sql select distinct BOOSTER_VERSION from SPACEXDATASET where INTEGER = (select MAX(INTEGER) FROM SPACEXDATASET)
```

[redacted]ln.cloud

Done.

Out[21]:

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
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

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

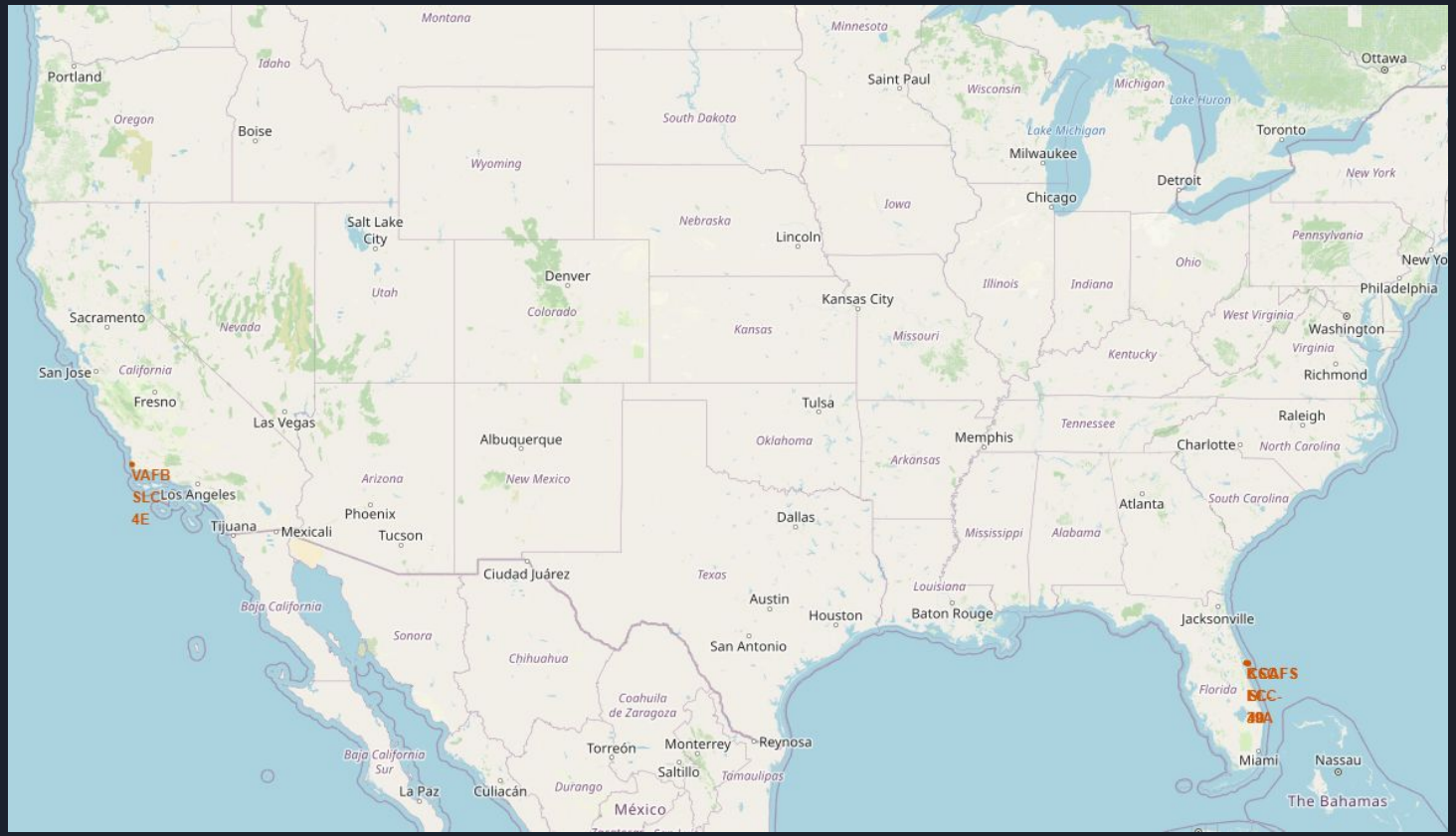
```
2]: %sql select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATASET where YEAR(DATE) = '2015' and LANDING__OUTCOME = 'Failure (drone ship)'
 * ibm_db[redacted]
Done.
```

[22]:

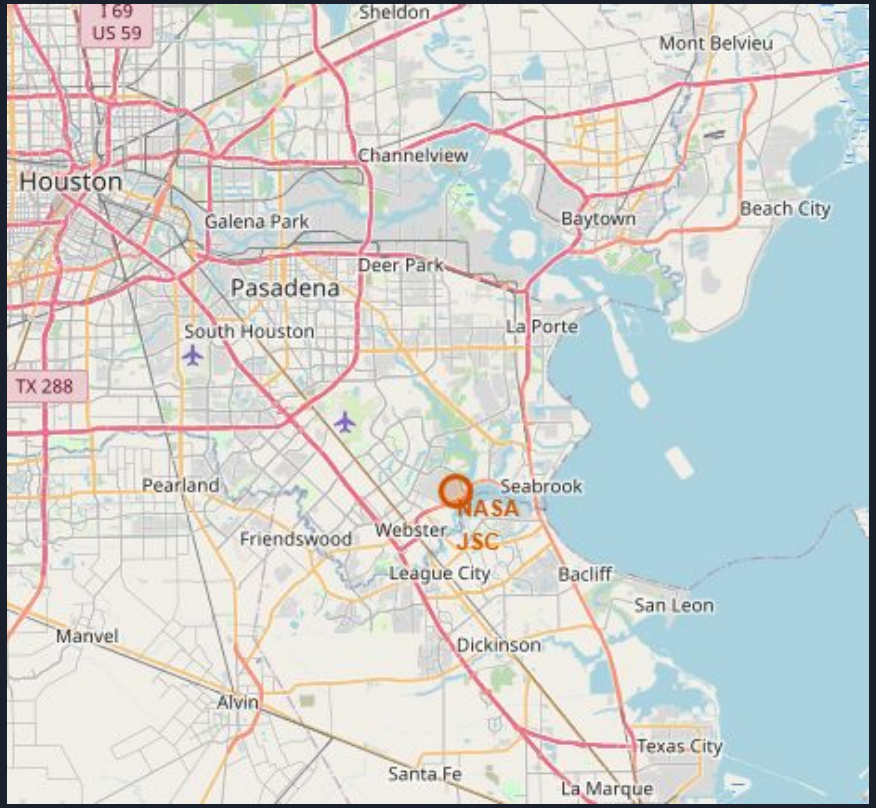
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



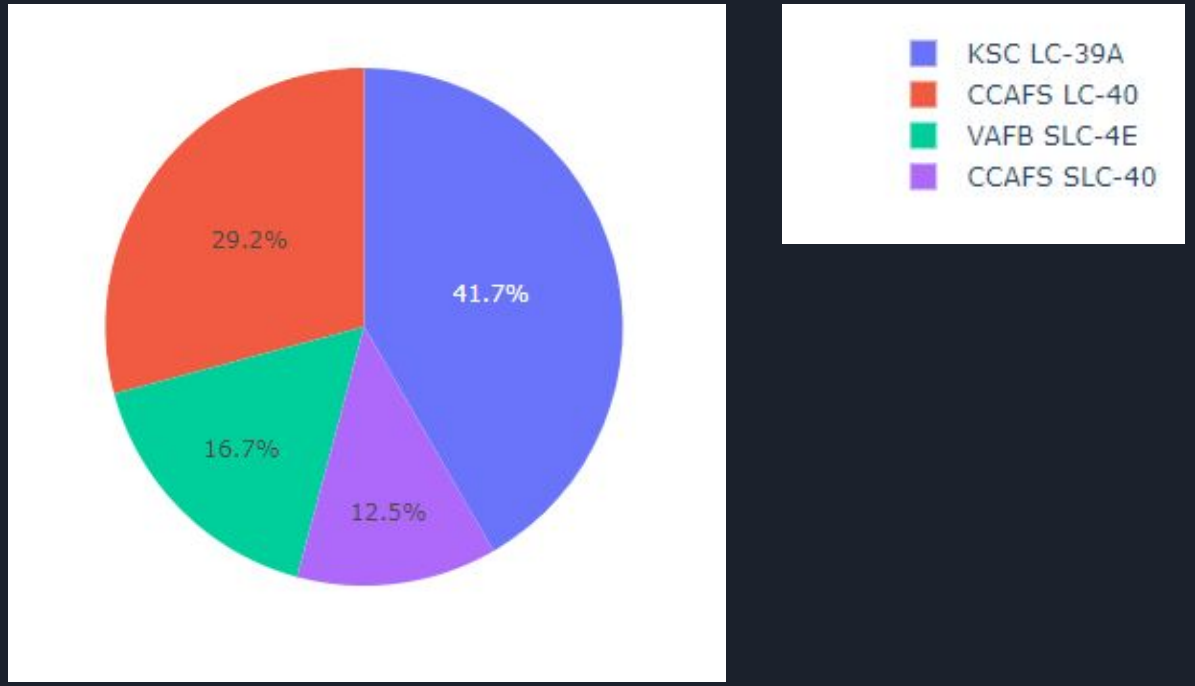
<Folium Map Screenshot 2>



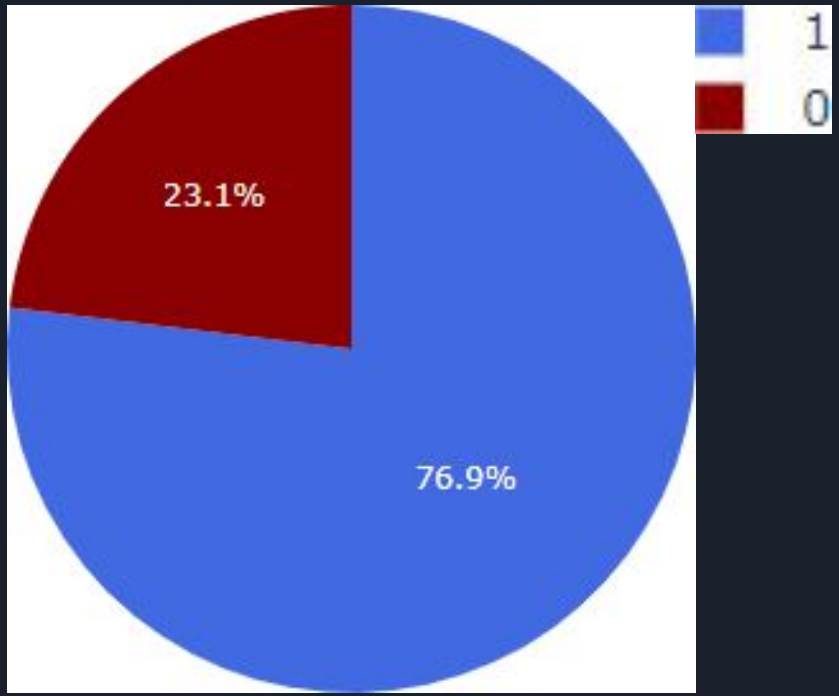
<Folium Map Screenshot 3>

Build a Dashboard with PlotlyDash

<Dashboard Screenshot 1>



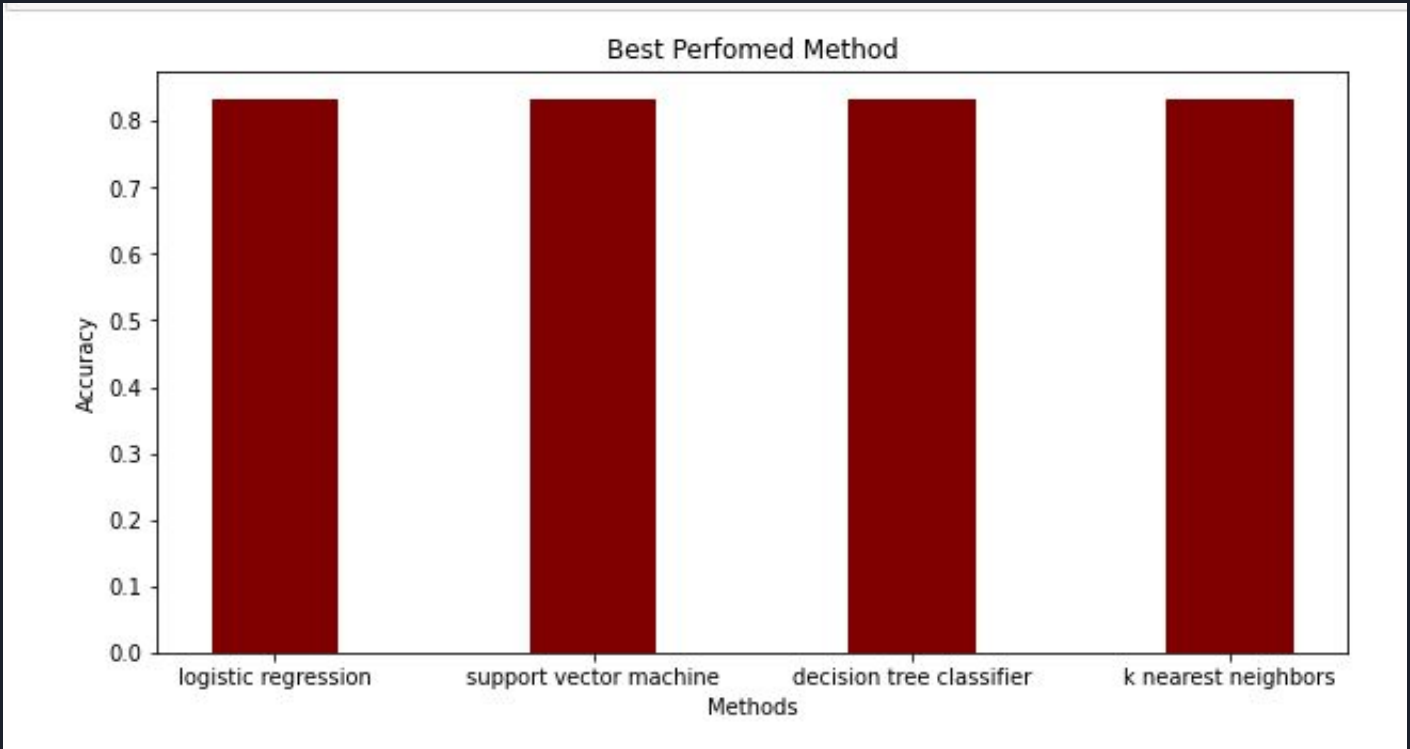
<Dashboard Screenshot 2>



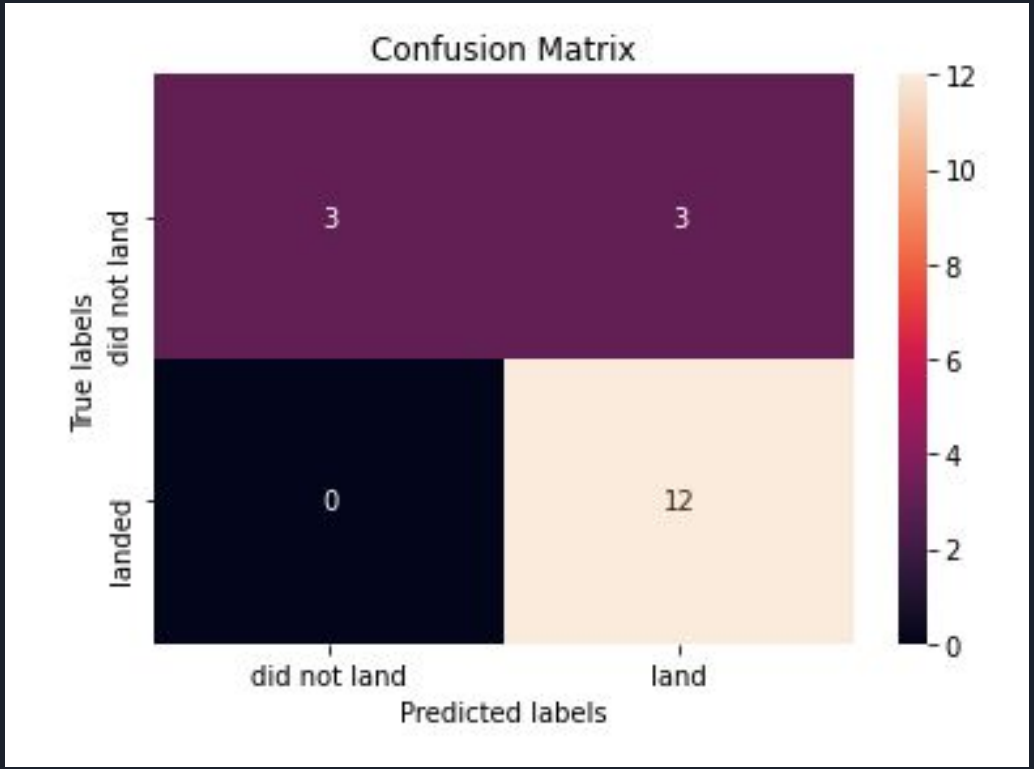
<Dashboard Screenshot 3>

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

Conclusions

- The task in hand: To develop a machine learning model for Space Y who wants to bid against SpaceX
- The goal of model is to predict if and when Stage 1 will successfully land to save ~$100 million USD
- The data that's used came from a public SpaceX API and web scraping SpaceX Wikipedia page
- We created a machine learning model with an accuracy of 83.3333%
    - Allon Mask of SpaceY can use this model to predict with relatively high accuracy if a launch will have a successful Stage 1 landing in order to determine whether or not the launch should be made
- More data should be collected to better determine the best machine learning model and improve accuracy if possible.

Appendix

Appendix

GitHub Repository URL

Thank You!

Julienne Manalo
December 20, 2021