IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

James L Glueck

December 19, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection and Manipulation
  - Data Wrangling
  - Exploratory Data Analysis
  - Site Location Analysis and Data Dashboarding
  - Machine Learning Based Predictive Analysis
- Summary of all results

Introduction

- The purpose of this applied data science project is to predict whether or to what degree the Falcon 9 first stage will land successfully.
- A key value proposition of the SpaceX program is cost minimization. The key to cost minimization is the extent to which launch components are redeployable for other missions.
- In particular retention of the first stage for subsequent missions is a key factor in SpaceX pricing missions significantly lower than competitors currently.
- Determining whether the Falcon 9 first stage will land successfully thus will allow us to compete more effectively against SpaceX based on overall price for rocket launch missions.

Section 1

Methodology

Methodology

Executive Summary

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

Data Collection

- API to extract information using identification numbers in the launch data.
- spacex_url=https://api.spacexdata.com/v4/launches/past
- static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'

Data Collection – SpaceX API

- GitHub URL of completed SpaceX API calls notebook:
  - https://github.com/jameslglueck/capstone/blob/main/001_jupyter-labs-spacex-data-collection-api-v2.ipynb

In [17]:
```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, a
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all collumns of a dataframe
pd.set_option('display.max_columns', None)
# Setting this option will print all of the data in a feature
pd.set_option('display.max_colwidth', None)
```

Below we will define a series of helper functions that will help us use the API to extract information using identification numbers in the launch data.

From the `rocket` column we would like to learn the booster name.

In [19]:
```
# 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.

In [21]:
```
# 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'])
```

Data Collection - Scraping

- GitHub URL of completed web scraping notebook:
    - https://github.com/jameslglueck/capstone/blob/main/002_jupyter-labs-webscraping.ipynb

```python
In [5]:
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

def booster_version(table_cells):
    """
    This function returns the booster version from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if i%2==0][0:-1])
    return out

def landing_status(table_cells):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=[i for i in table_cells.strings][0]
    return out


def get_mass(table_cells):
    mass=unicodedata.normalize("NFKD", table_cells.text).strip()
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find("kg")+2]
    else:
        new_mass=0
    return new_mass


def extract_column_from_header(row):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.contents)

    # Filter the digit and empty names
    if not(colunm_name.strip().isdigit()):
        colunm_name = colunm_name.strip()
        return colunm_name
```

Data Wrangling

- Data Wrangling Process
  - Data Analysis
    - Calculate the number of launches on each site
    - Calculate the number and occurrence of each orbit
    - Calculate the number and occurrence of mission outcome of the orbits
    - Create a landing outcome label from Outcome column
- GitHub URL of completed data wrangling related notebook:
  - https://github.com/jameslglueck/capstone/blob/main/003_labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA with Data Visualization

- Plotted Charts
  - Scatterplot to visualize the relationship between Flight Number and Launch Site
  - Scatterplot to visualize the relationship between Payload and Launch Site
  - Scatterplot to visualize the relationship between success rate of each orbit type
  - Scatterplot to visualize the relationship between Flight Number and Orbit type
  - Line chart to visualize the launch success yearly trend
- GitHub URL of completed EDA with data visualization notebook:
  - https://github.com/jameslglueck/capstone/blob/main/005_jupyter-labs-eda-dataviz-v2.ipynb

EDA with SQL

- SQL Queries Performed:
  - Display the names of the unique launch sites in the space mission
  - 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 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
- GitHub URL of completed EDA with SQL notebook:
  - https://github.com/jameslglueck/capstone/blob/main/004_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Map Objects Created:
  - Mapped Launch Sites
    - folium.Circle
    - folium.Marker
  - Mapped Success/Failed Launches
    - MarkerCluster
  - Calculate the distances between a launch site to proximities
    - MousePosition
- GitHub URL of completed interactive map with Folium map:
  - https://github.com/jameslglueck/capstone/blob/main/006_lab-jupyter-launch-site-location-v2.ipynb

Build a Dashboard with Plotly Dash

- Dashboard Plots and Interactions
  - Dropdown list to enable Launch Site selection
  - Pie chart to show the total successful launches count for all sites
  - Slider to select payload range
  - Scatter chart to show the correlation between payload and launch success
- GitHub URL of completed Plotly Dash lab:
  - https://github.com/jameslglueck/capstone/blob/main/SpaceX%20Module%203

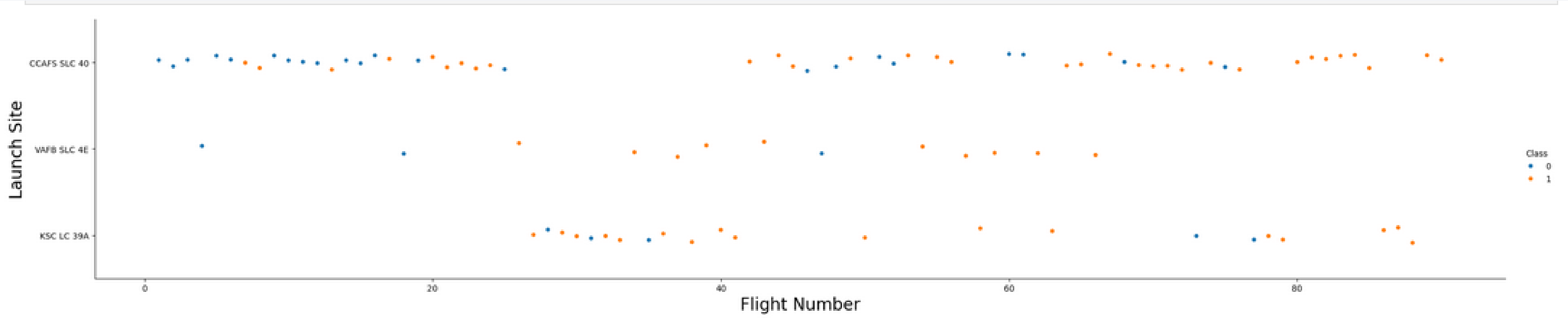
Predictive Analysis (Classification)

- Predictive Analysis Process:
  - Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y
  - Standardize the data in X then reassign it to the variable X
  - 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.
  - Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
  - Calculate the accuracy on the test data using the method score.
  - Create a support vector machine object then create a GridSearchCV object svm_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
  - Calculate the accuracy on the test data using the method score.
  - Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
  - Calculate the accuracy of tree_cv on the test data using the method score.
  - Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.
  - Calculate the accuracy of knn_cv on the test data using the method score.
  - Find the method performs best.
- GitHub URL of completed predictive analysis lab:
  - https://github.com/jameslglueck/capstone/blob/main/007_SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

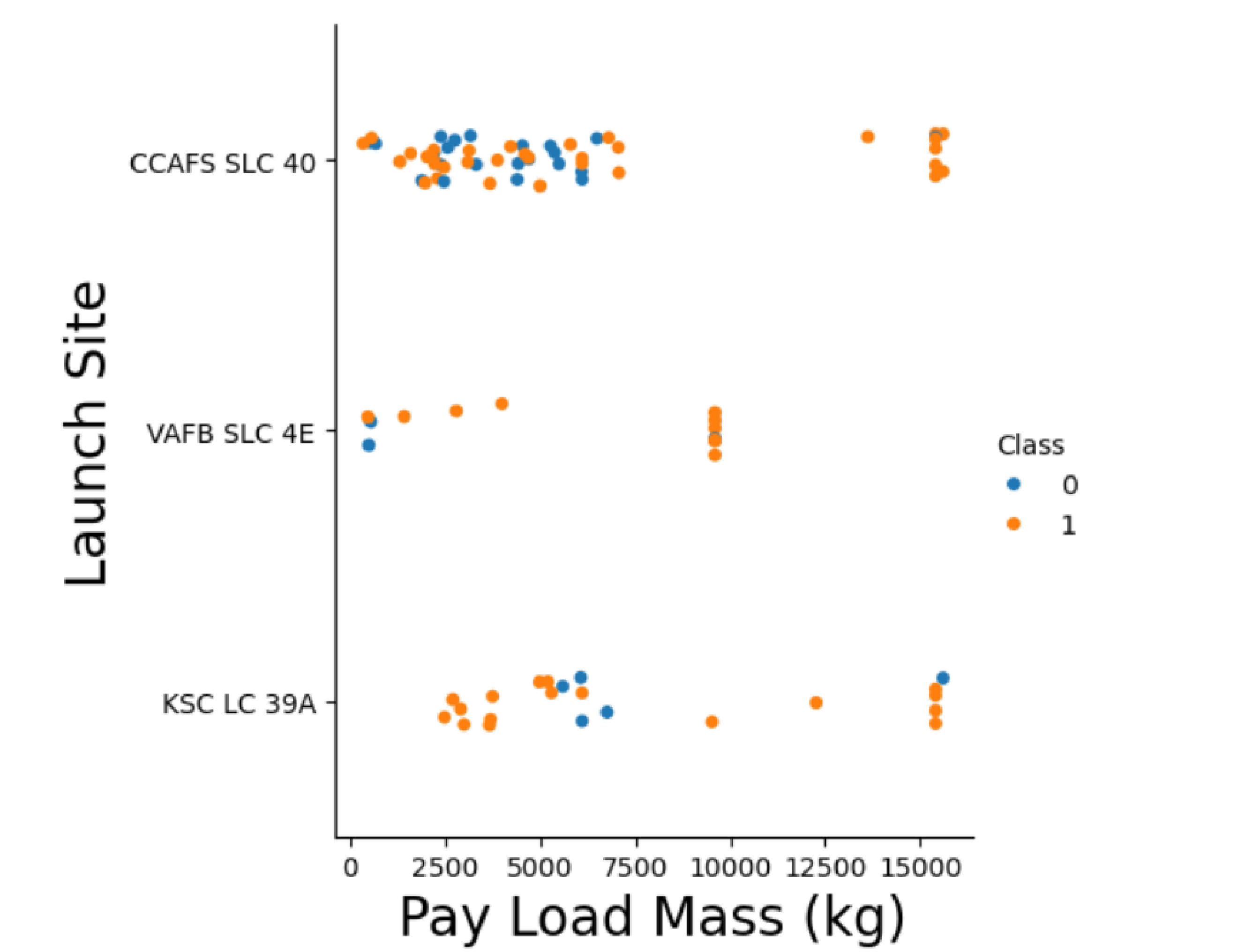
Section 2

Insights drawn from EDA

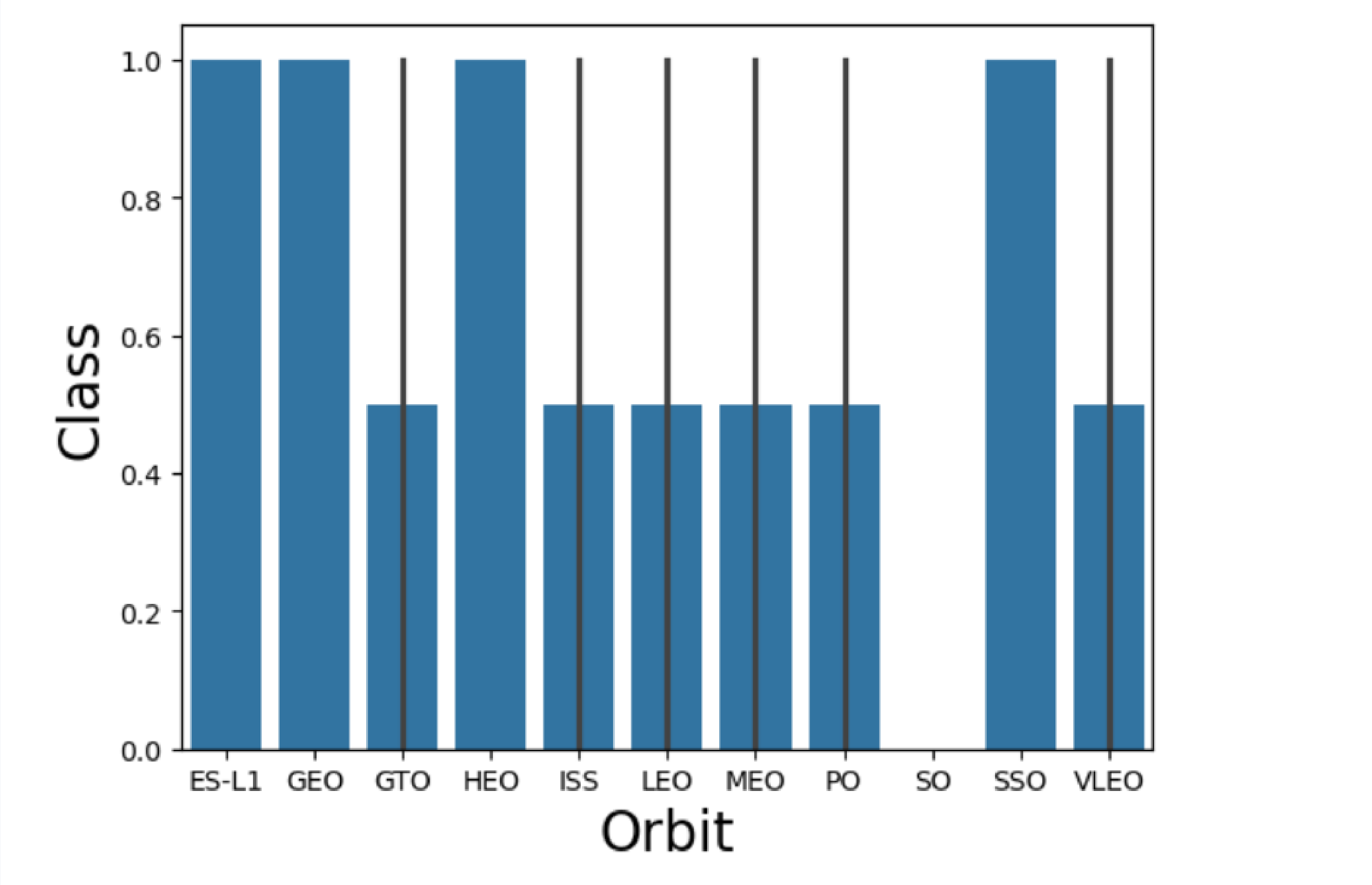
Flight Number vs. Launch Site



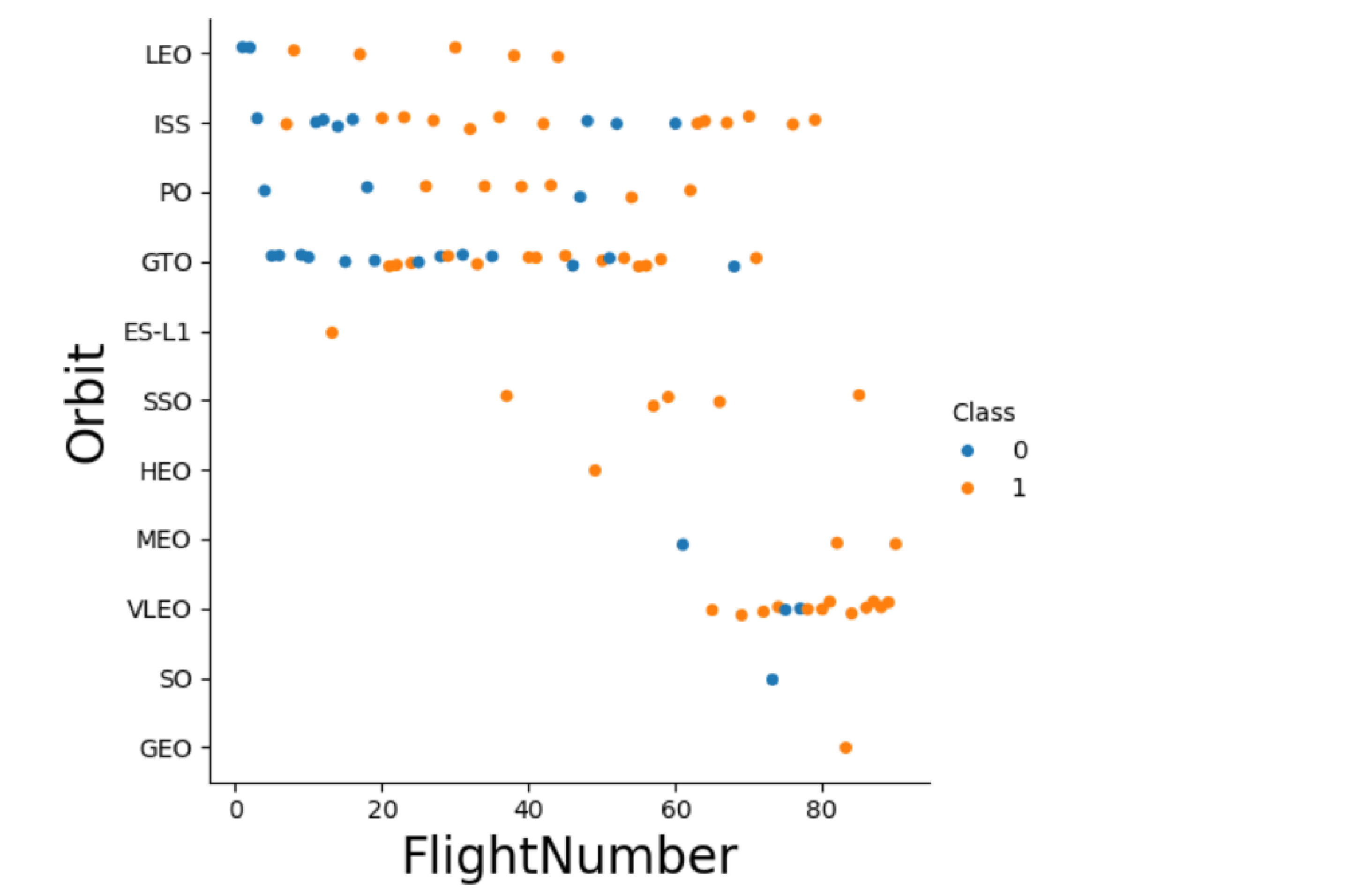
Payload vs. Launch Site



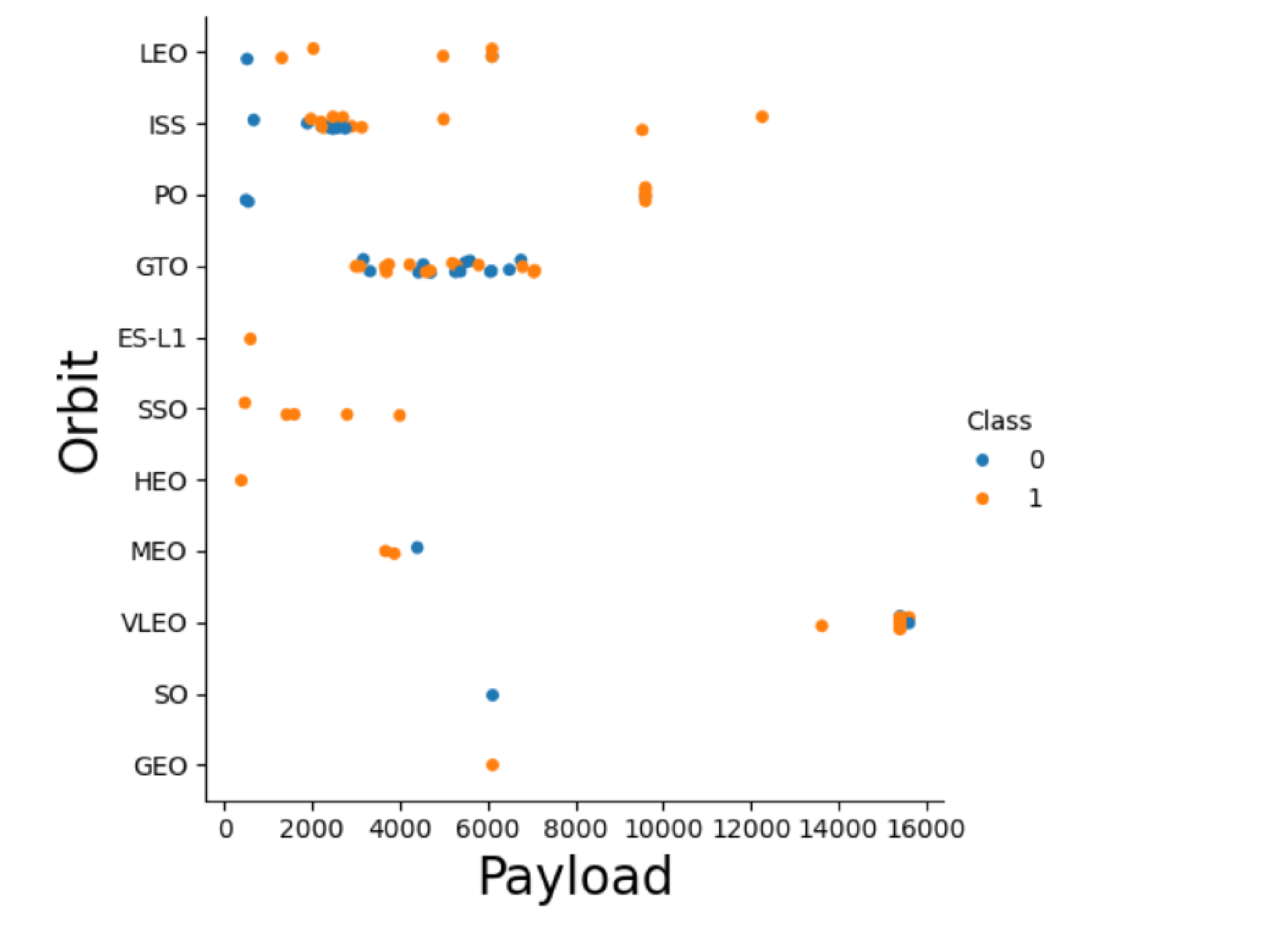
Success Rate vs. Orbit Type



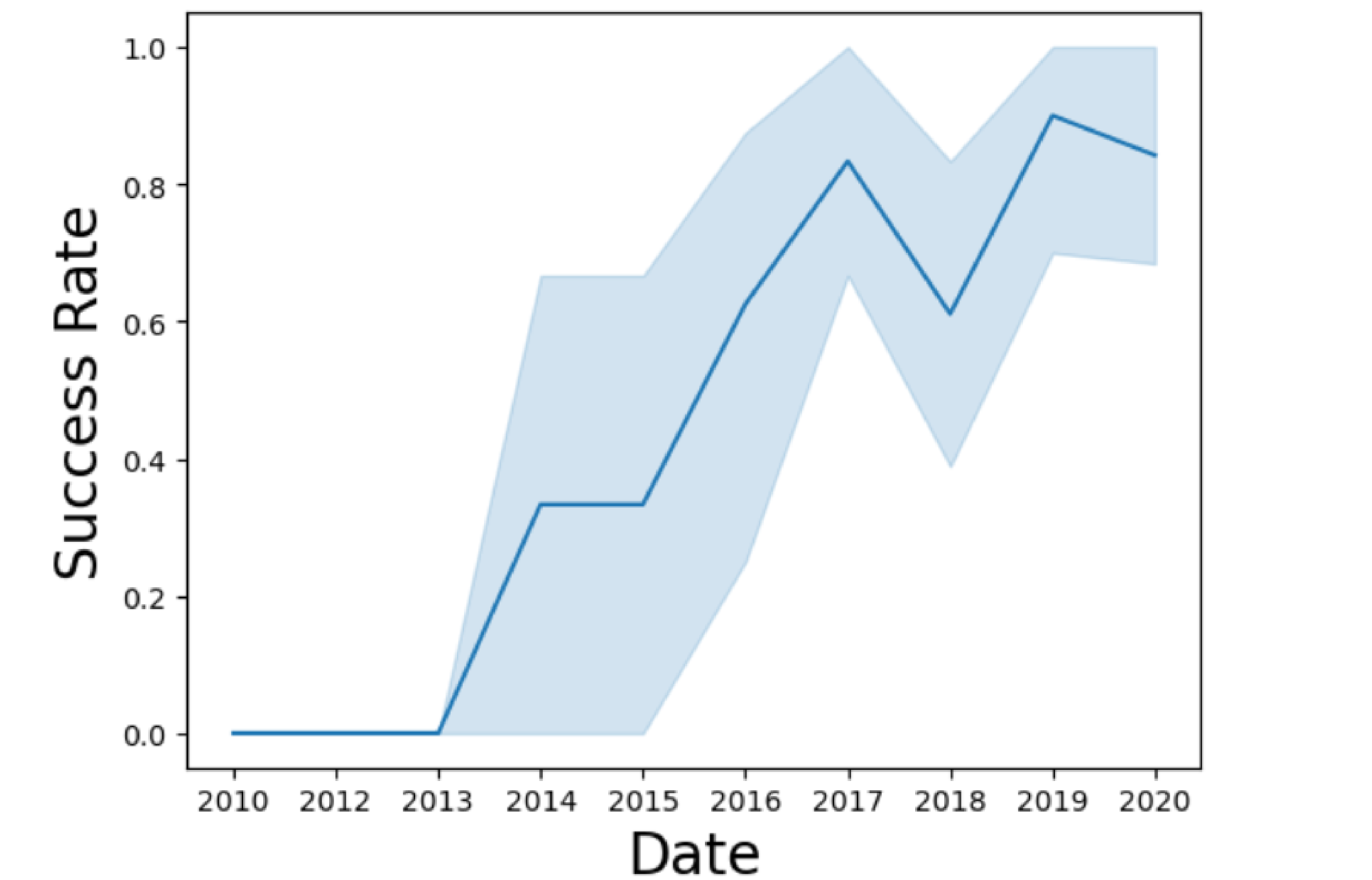
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
%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

Launch Site Names Begin with 'CCA'

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

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Ou
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (para
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 (para
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No a
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No a
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No a

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__K
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME
```

```
 * sqlite:///my_data1.db
Done.
```

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SP
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

]:

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

```
%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by
```

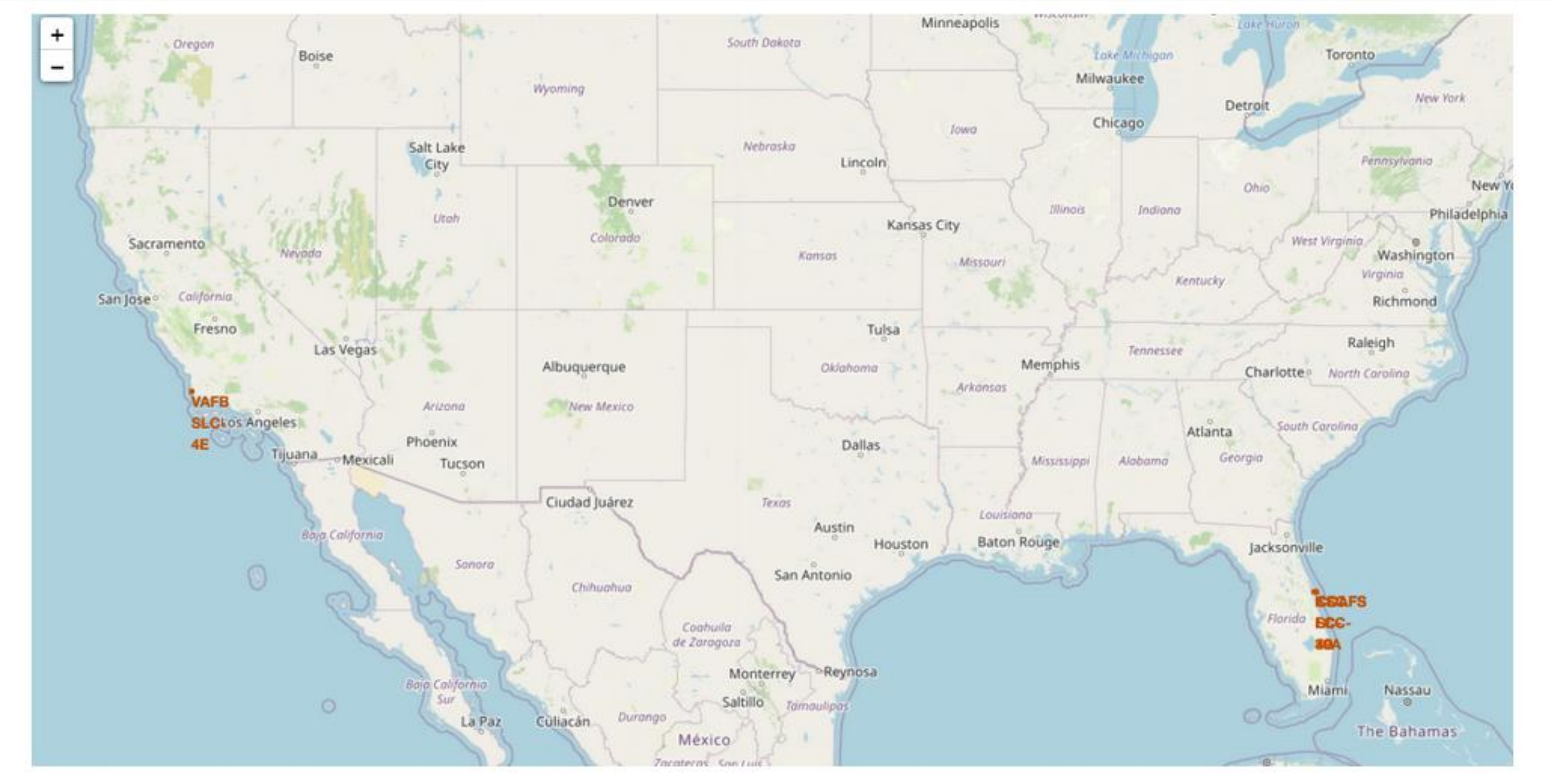
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landin
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Succ
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Suc
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Suc
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Succ
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Suc
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Suc
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Suc
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Succ

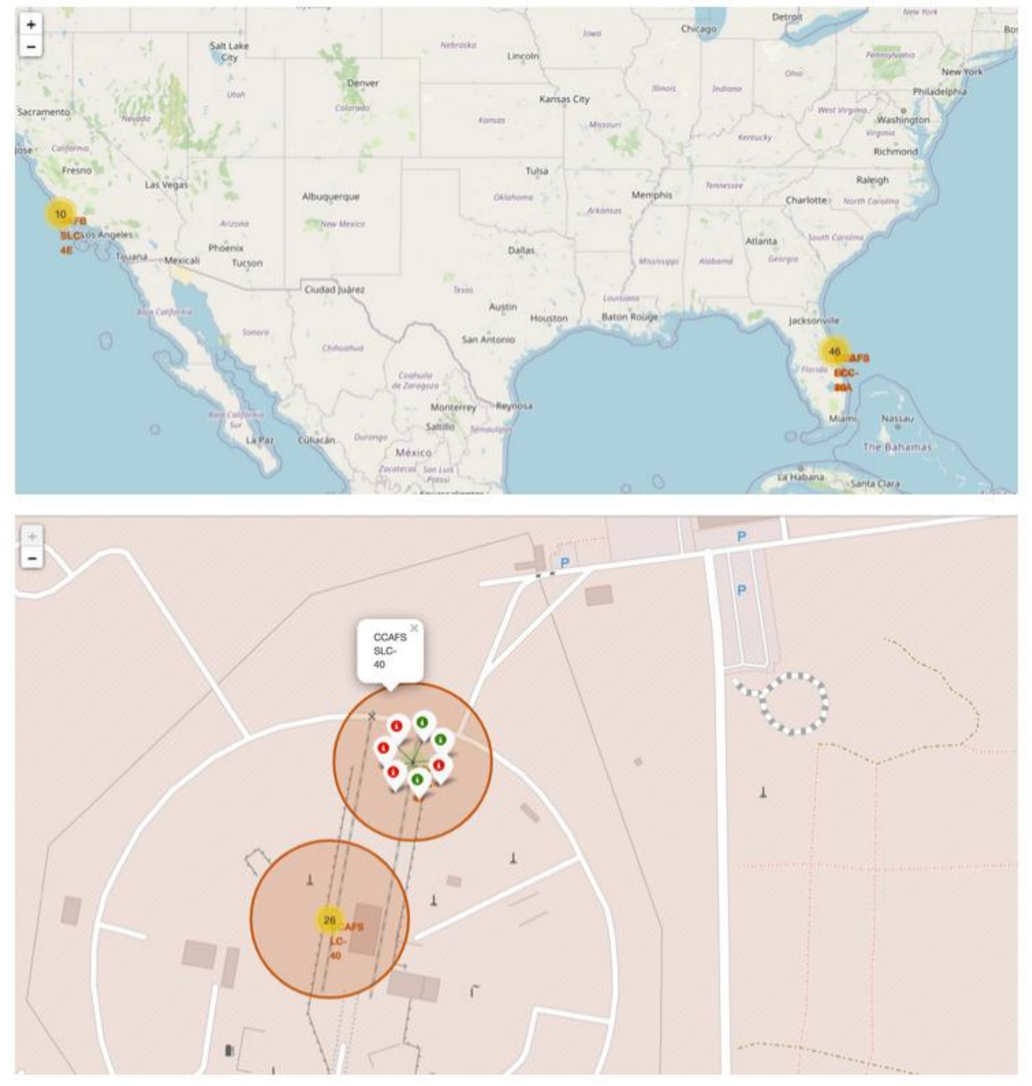
Section 3

Launch Sites Proximities Analysis

Marked Launch Sites



Marked Success/Failed Launches

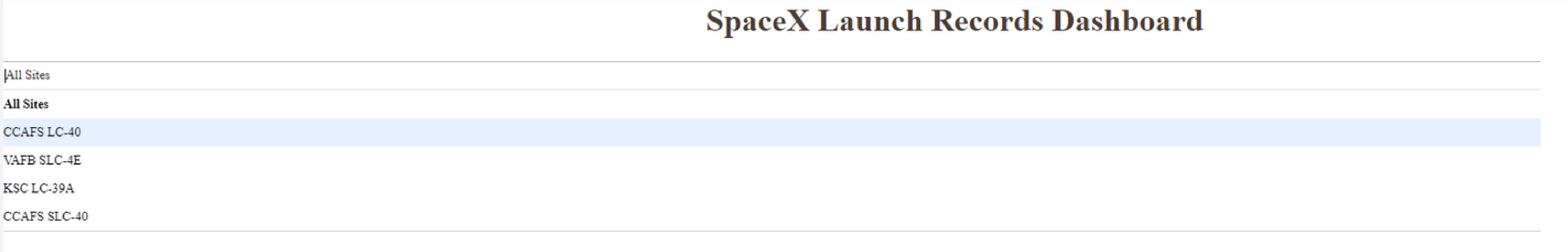


Distances Between Launch Site and Proximities

Section 4

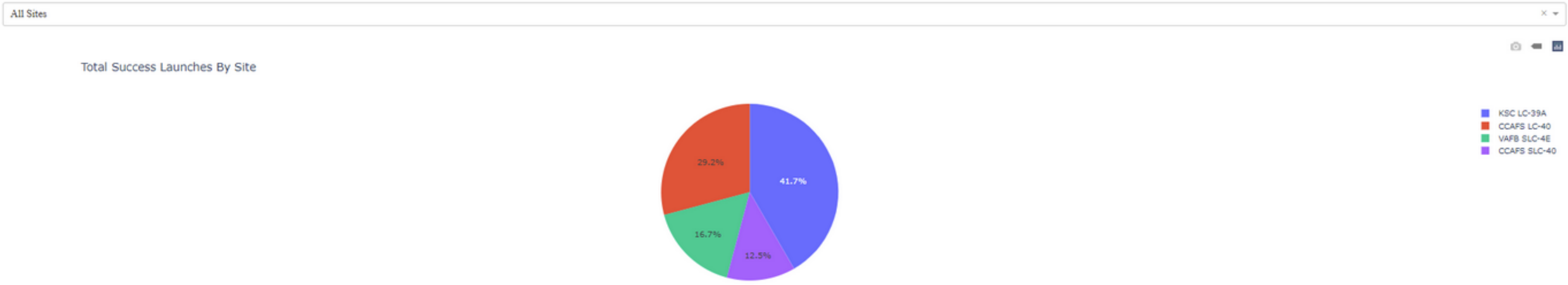
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

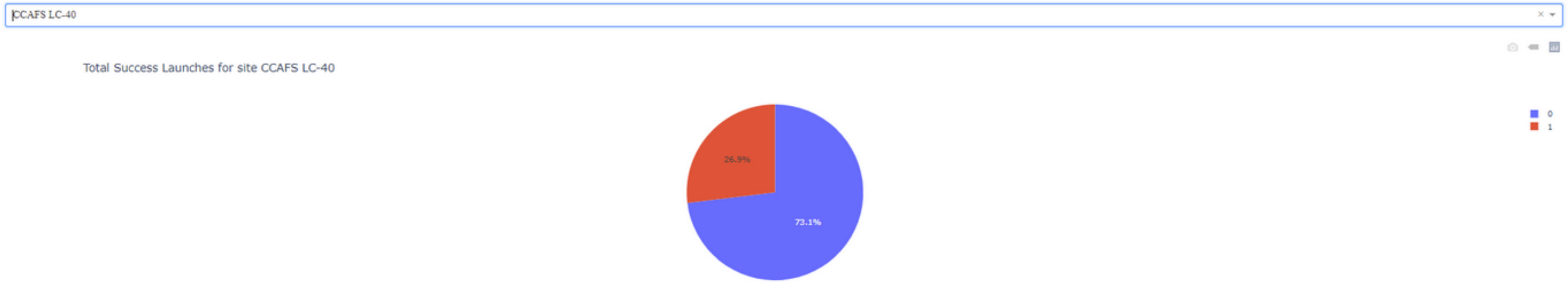


Pie Chart for Selected Launch Sites

- Pie chart for all sites are selected



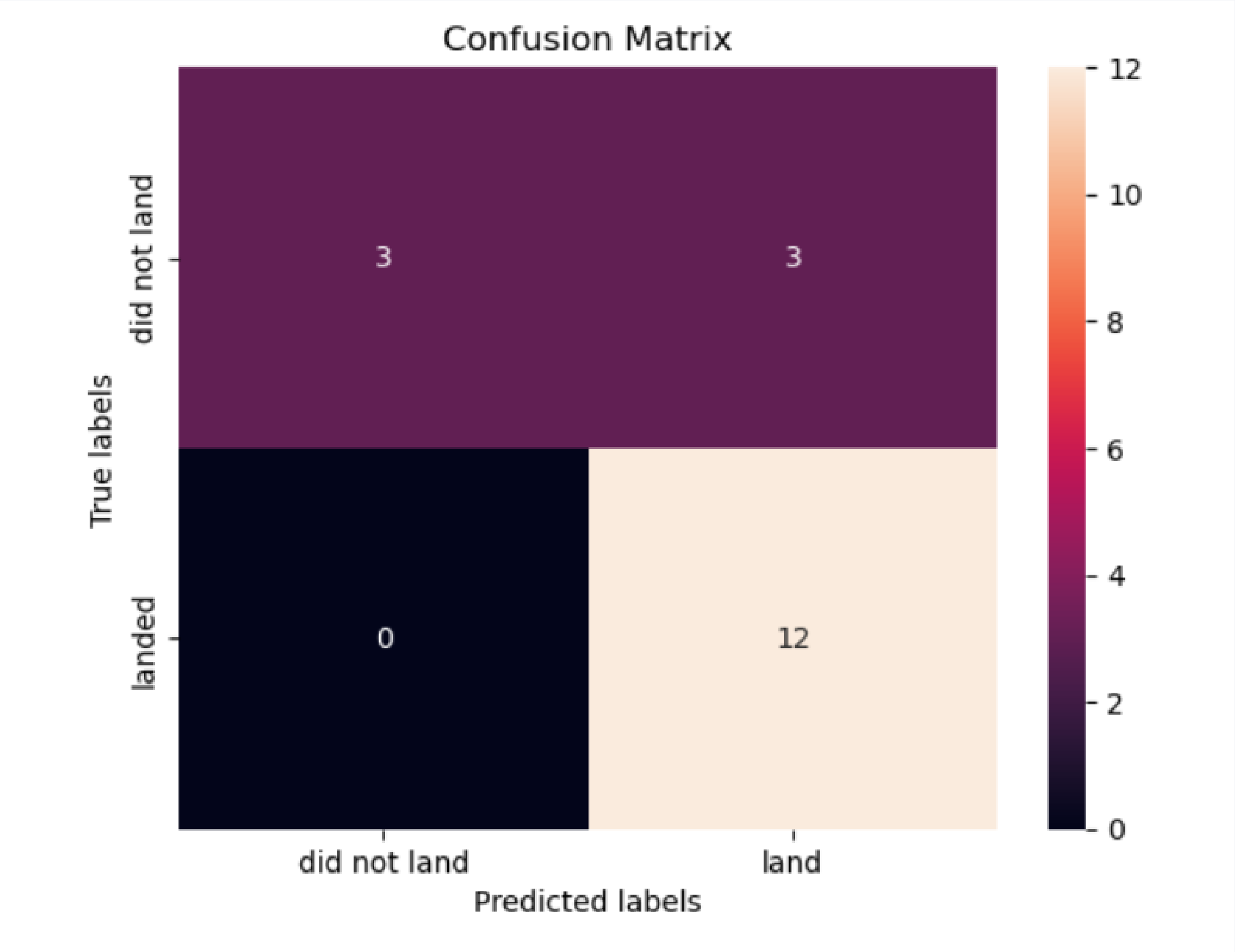
- Pie chart for is selected



Section 5

Predictive Analysis (Classification)

Confusion Matrix



Thank you!