IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Diogo Gomes
2022-11

Outline

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

PÚBLICA

Executive Summary

- Summary of methodologies
  - Data Collection (API and web scraping)
  - Data Wrangling
  - EDA - Exploratory Data Analysi, wituh further Data Visualization
  - Interactive Visual Analytics - Folium
  - Machine Learning Predictions

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

PÚBLICA

Introduction

- Project background and context
  - The object is to evaluate how viable would be for company SpaceY to compete with Space X
- Problems you want to find answers
  - Can we predict if a rocket wil land successfully?
  - Identify the features that determine the success rate or rocket landing

PÚBLICA

Section 1

Methodology

PÚBLICA

Methodology

Executive Summary

- Data collection methodology:
  - Dataset was collected using SpaceX API and web scraping
- Perform data wrangling
  - Categorical features were processed to be represented as one-hot-encoding
- 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

PÚBLICA

Data Collection

- Dataets collected from spaceX api: https://api.spacexdata.com/v4/rockets/
- Web scraping from wikipedia

PÚBLICA

Data Collection – SpaceX API

- SpaceX offers a public api

- Source code: https://github.com/diogosmg/coursera-DSCapstone/blob/master/Data%20Collection%20API.ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```


PÚBLICA

Data Collection - Scraping

- Web scraping from Wikipedia using Beautiful soup

- https://github.com/diogosmg/coursera-DSCapstone/blob/master/Data%20Collection%20API.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```


PÚBLICA

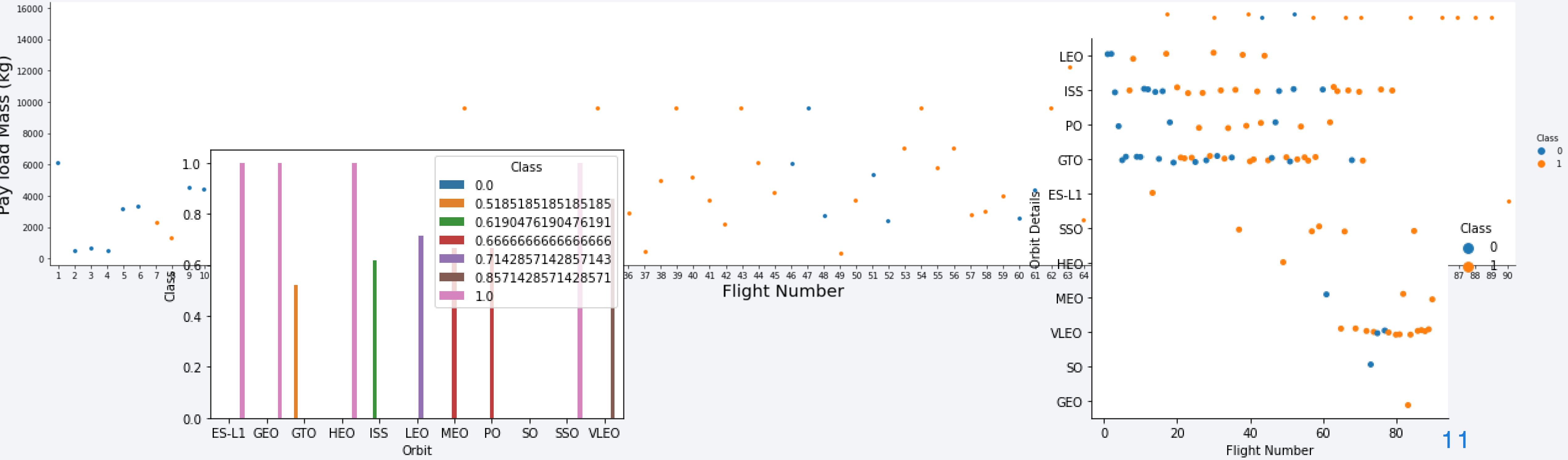
Data Wrangling

- EDA performed to determine the training labels.
- https://github.com/diogosmg/coursera-DSCapstone/blob/master/EDA%20-%20Data%20wrangling.ipynb

PÚBLICA

EDA with Data Visualization

- Scatterplots and barplots
- https://github.com/diogosmg/coursera-DSCapstone/blob/master/EDA%20with%20Data%20Visualization.ipynb



PÚBLICA

EDA with SQL

- SQL Queries:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

- https://github.com/diogosmg/coursera-DSCapstone/blob/master/EDA%20with%20SQL.ipynb

PÚBLICA

Build an Interactive Map with Folium

- Launch sites were added to the map as markers, circles and lines
- https://github.com/diogosmg/coursera-DSCapstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

PÚBLICA

Build a Dashboard with Plotly Dash

- Percentage of launches by site and payload range
- https://github.com/diogosmg/coursera-DSCapstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

PÚBLICA

Predictive Analysis (Classification)

- Dataset was loaded as pandas dataframe, and split into training and testing dataset
- We developed different ML models to identify the best performing algorithm
- https://github.com/diogosmg/coursera-DSCapstone/blob/master/MLPrediction.ipynb

PÚBLICA

Results

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

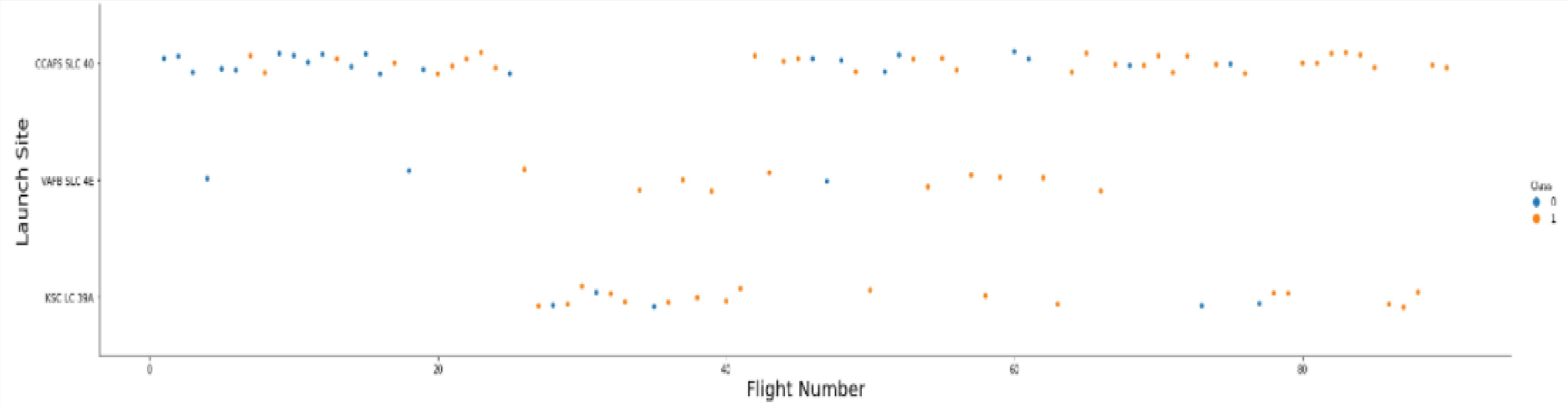
PÚBLICA

Section 2

Insights drawn from EDA

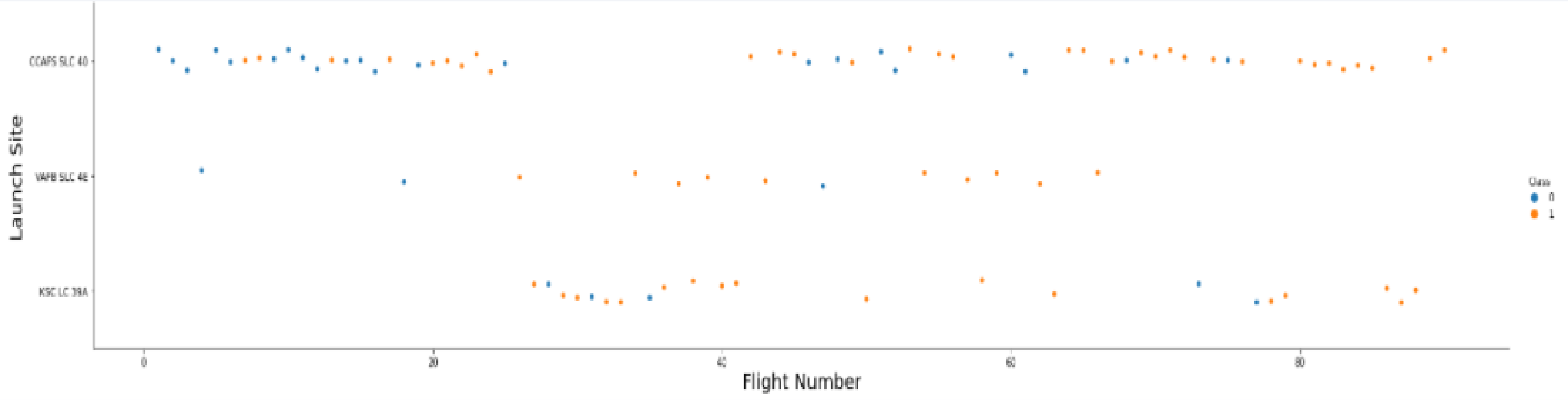
PÚBLICA

Flight Number vs. Launch Site



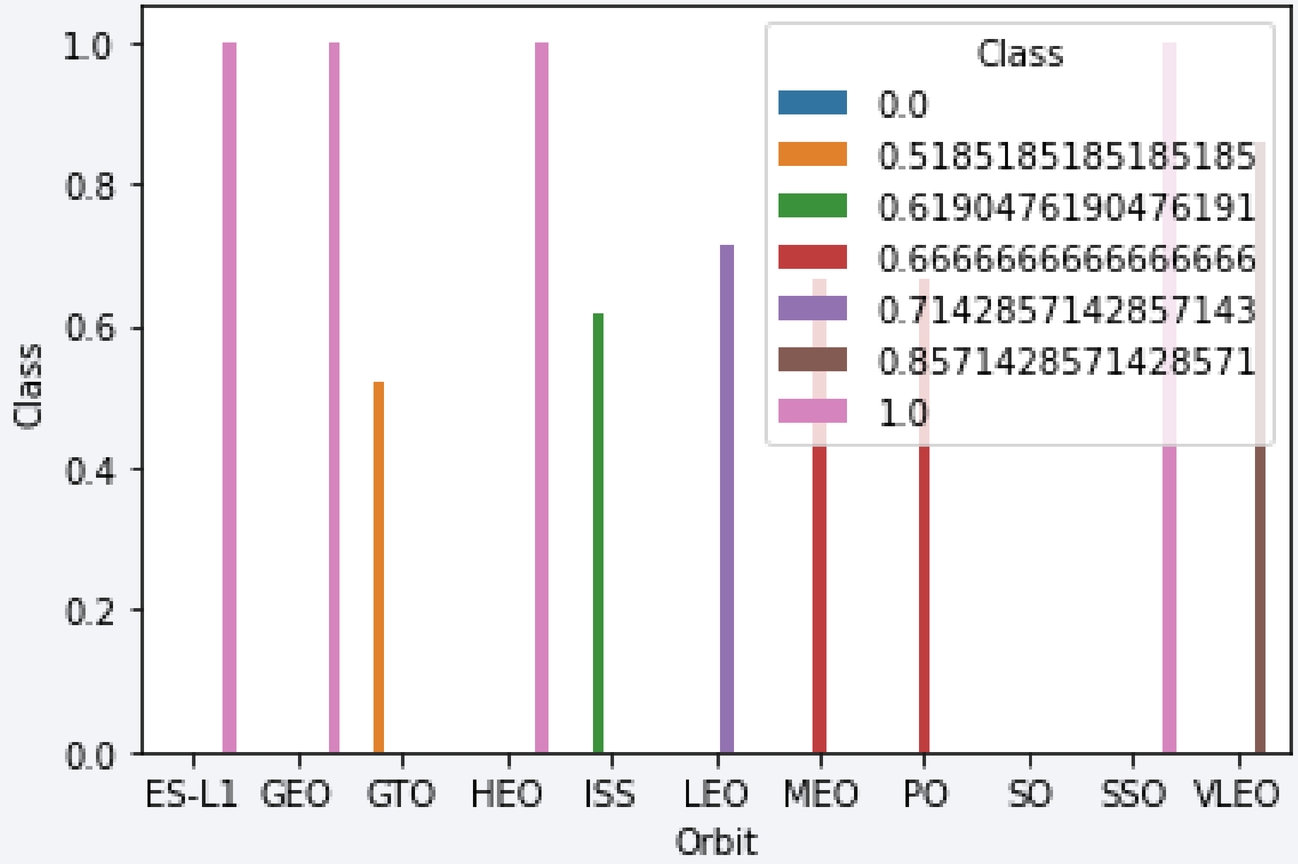
PÚBLICA

Payload vs. Launch Site

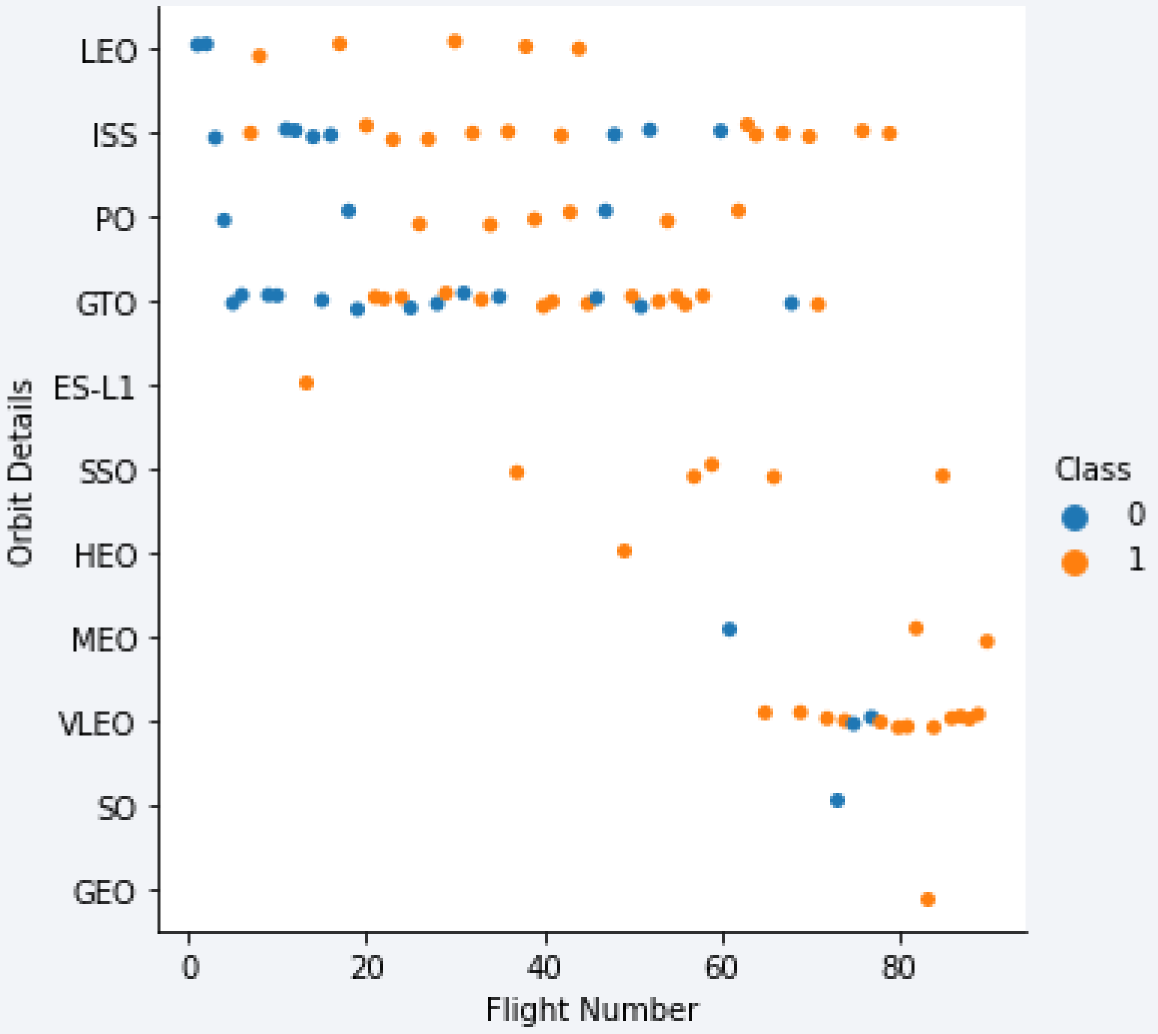


PÚBLICA

Success Rate vs. Orbit Type

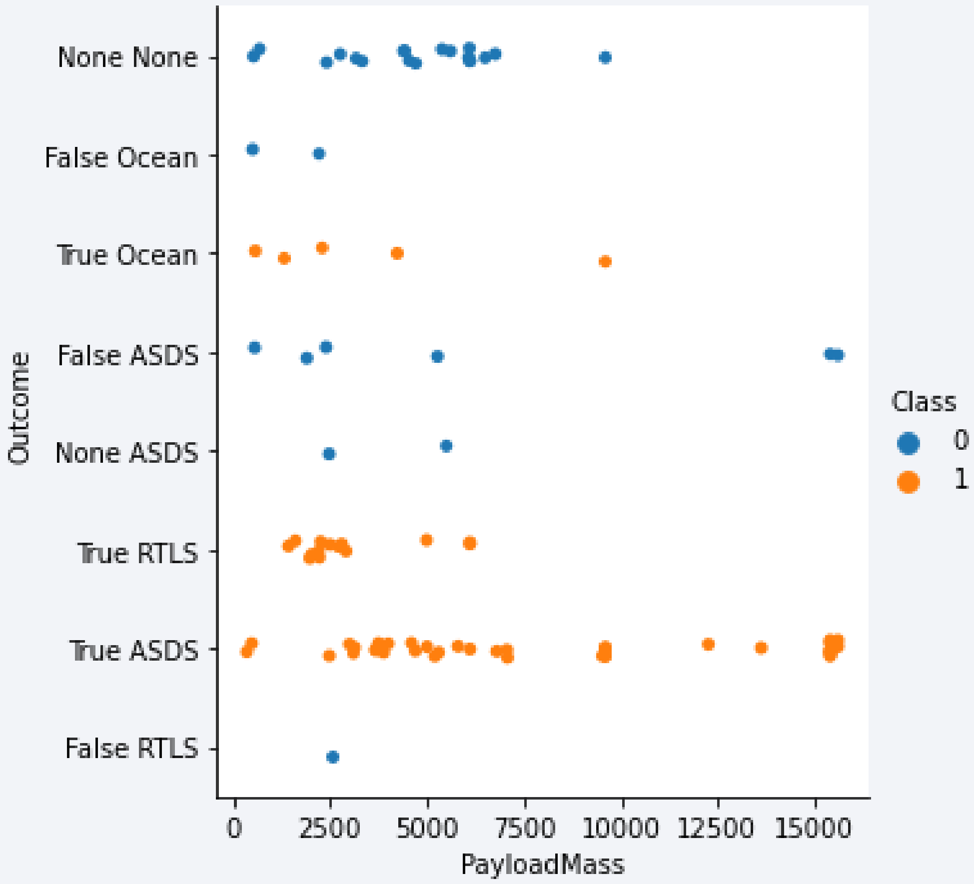


Flight Number vs. Orbit Type



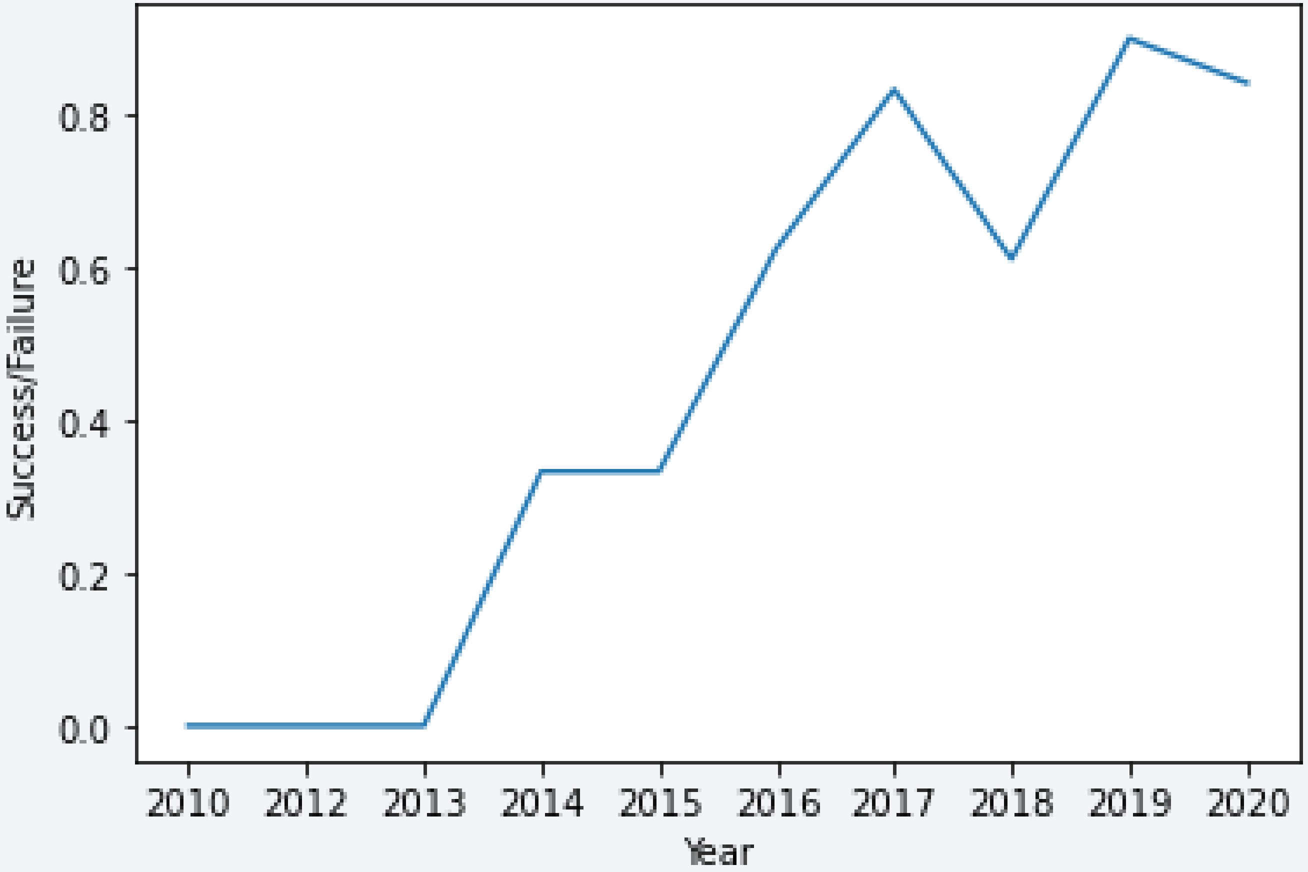
PÚBLICA

Payload vs. Orbit Type



PÚBLICA

Launch Success Yearly Trend



PÚBLICA

All Launch Site Names

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

PÚBLICA

Launch Site Names Begin with 'CCA'

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

PÚBLICA

Total Payload Mass

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

In [12]:

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

PÚBLICA

Average Payload Mass by F9 v1.1

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

In [13]:

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

PÚBLICA

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [13]:

```
%sql select min(DATE) from SPACEX;
```

```
 * ibm_db_sa://xtw67748:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lc
bludb
Done.
```

Out[13]:

1
2010-06-04

PÚBLICA

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

In [15]:

```
%sql select BOOSTER_VERSION from SPACEX where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4(
```

```
 * ibm_db_sa://xtw67748:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

Out[15]:

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

PÚBLICA

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [17]:
```
%sql select count(MISSION_OUTCOME) from SPACEX GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://xtw67748:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

Out[17]:

1
1
99
1

PÚBLICA

Boosters Carried Maximum Payload

Out[18]:

boosterversion
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

PÚBLICA

2015 Launch Records

Done.

Out[19]:

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

PÚBLICA

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

In [20]:
```
%sql SELECT LANDING__OUTCOME FROM SPACEX WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

```
 * ibm_db_sa://xtw67748:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

Out[20]:

landing__outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)

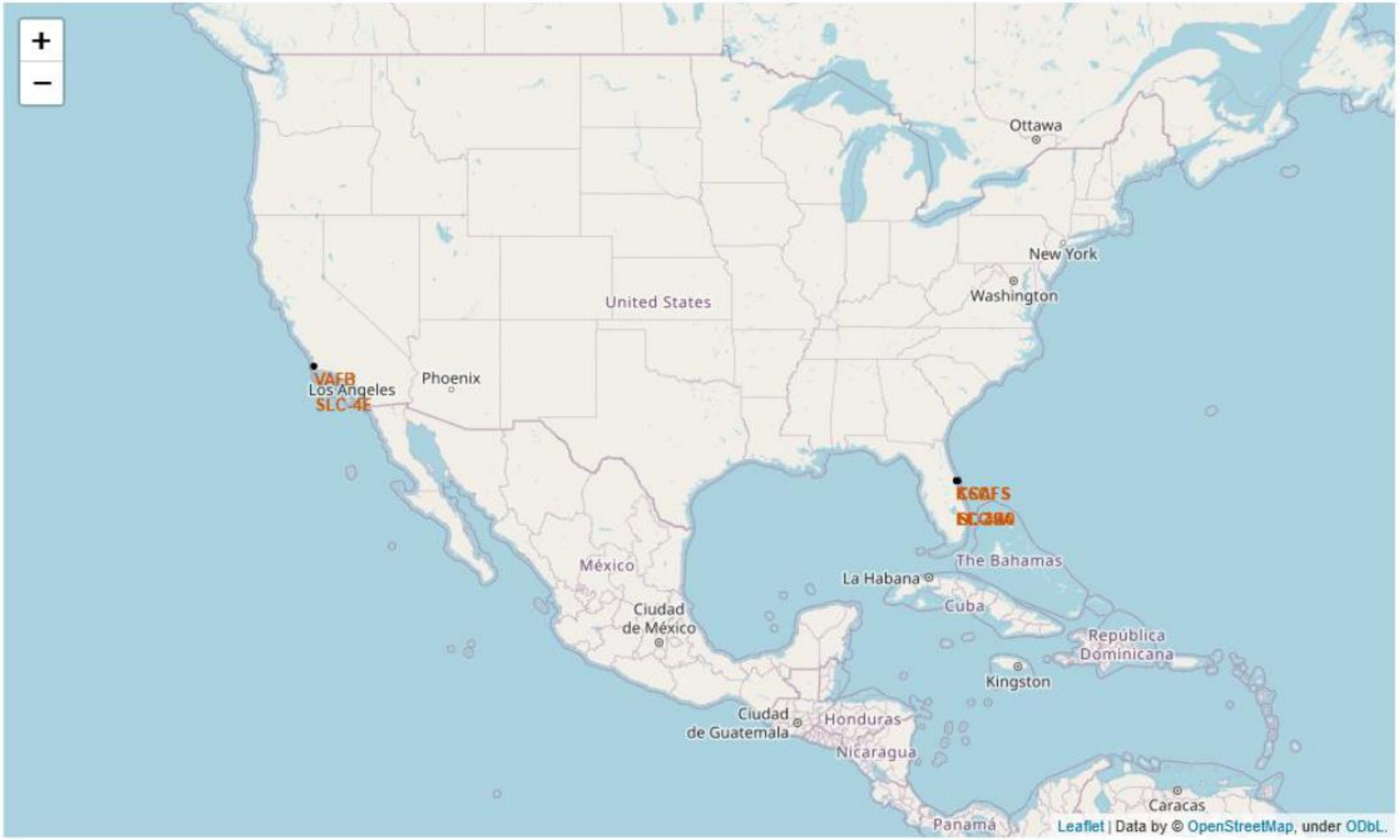
PÚBLICA

Section 3

Launch Sites Proximities Analysis

PÚBLICA

<Folium Map Screenshot 1>



<Folium Map Screenshot 2>

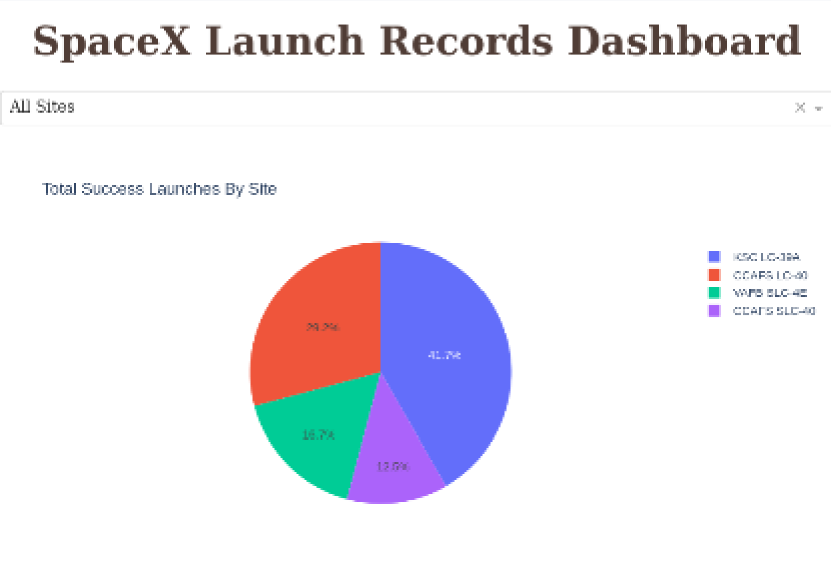
PÚBLICA

Section 4

Build a Dashboard with Plotly Dash

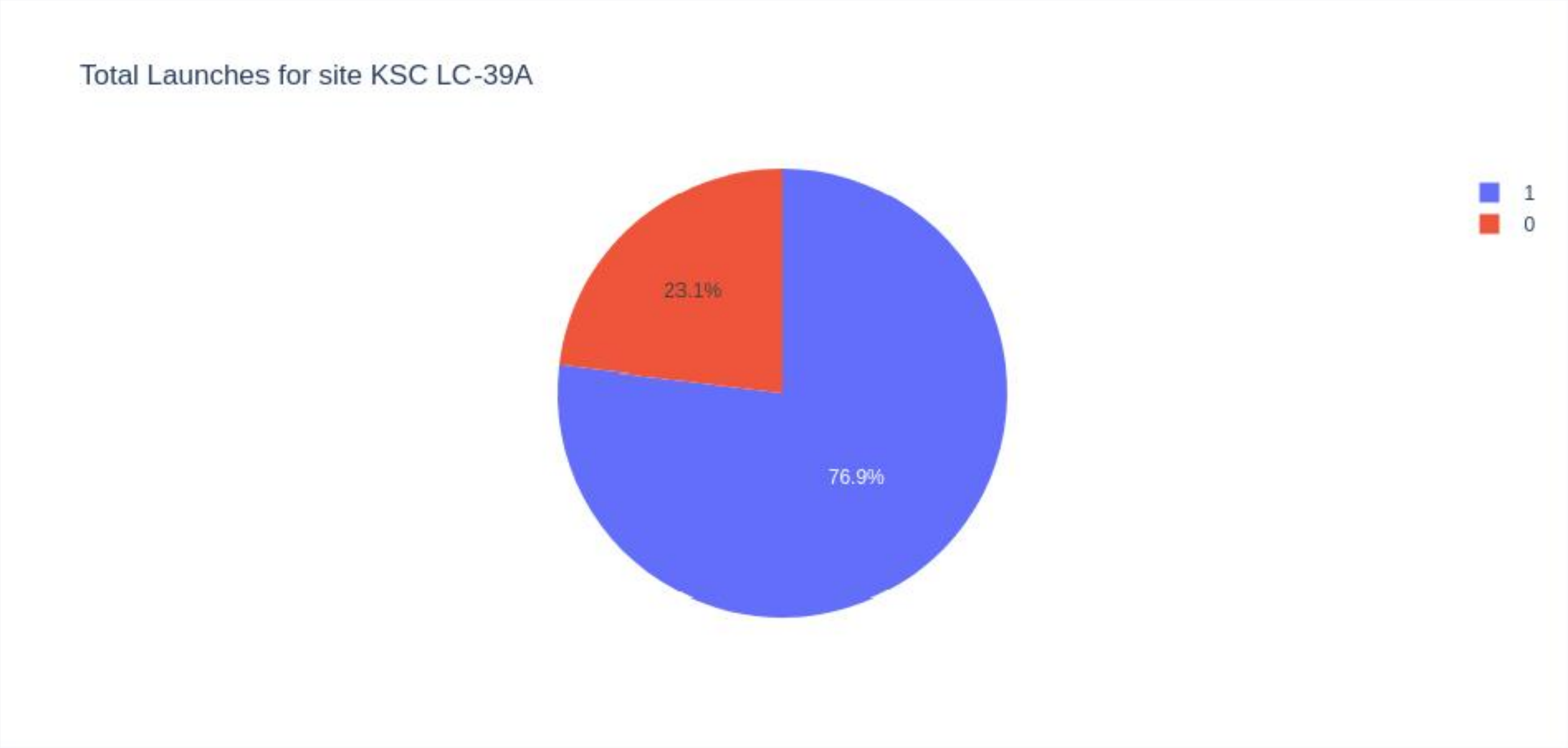
PÚBLICA

Dashboard



PÚBLICA

<Dashboard Screenshot 2>



PÚBLICA

Section 5

Predictive Analysis (Classification)

PÚBLICA

Classification Accuracy

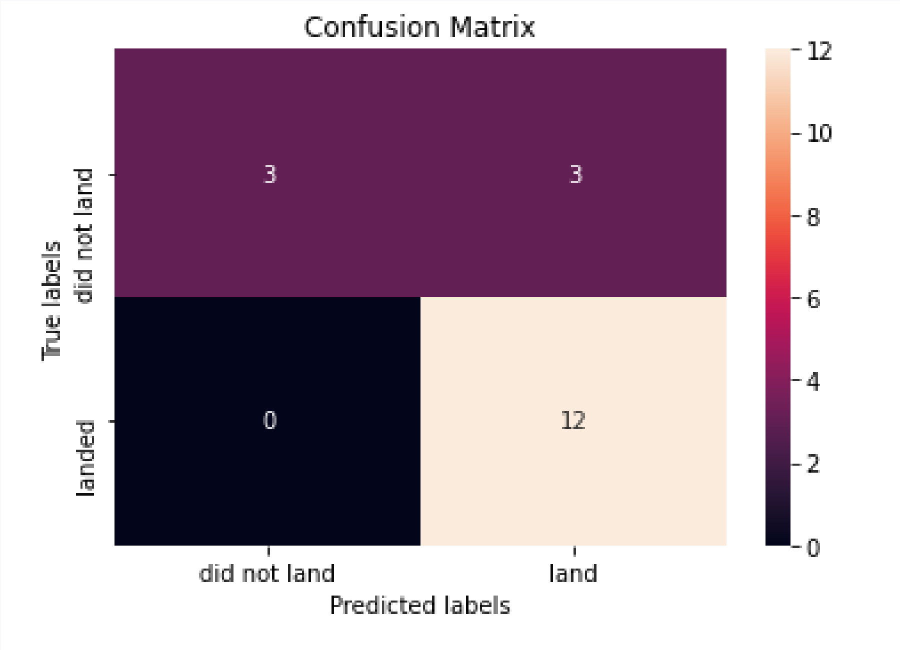
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```


PÚBLICA

Confusion Matrix



PÚBLICA

Conclusions

- Launches above 7ton are less risky
- Launch success rate increased from 2013 on.
- Decision Tree classified is the algorithm that performed the best for this task

PÚBLICA

Appendix

PÚBLICA

Thank you!

PÚBLICA