IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<MARCIO CARVALHO>
<08-12-2022>

Outline

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

Executive Summary

Summary of methodologies:

- ❑ Data Collection
- ❑ Data Wrangling
- ❑ Exploratory Data Analysis (EDA) using SQL
- ❑ Exploratory Data Analysis (EDA) using Pandas e Matplotlib
- ❑ Interactive Visual Analytics e Dashboard
- ❑ Predictive Analysis (Classification)

Summary of all results:

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

Introduction

The commercial space age is here, companies private are making space travel affordable for everyone.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars;

other providers cost upwards of 165 million dollars each, much of the savings is because

SpaceX can reuse the first stage.

So if we can predict whether the first stage will land, we can determine the cost of a launch.

The objective of the project is to analyze and predict the ideal conditions for reuse first stage of launch.

Answering the following questions:

Falcon9 rocket will land successfully and be able to generate references for future competitors?

What are the location with the best launch results?

What size of the first stage was most successful?

How accurate are the success predictions of the best launch sites?

Section 1

Methodology

Methodology

Executive Summary:

- **Data collection methodology**

  With Request to the SpaceX API e Web Scrapping

- **Perform data wrangling**

  In this step we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed and 0 means it was unsuccessful and cleaning data from null values and irrelevant columns.

- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - Logistic Regression (LR), Support Vector Machine (SVC), Decision Tree Classifier (Tree), K - Neighbors Classifier (KNN) models and evaluated for the best classifier.

Data Collection

Datasets were collected using SpaceX API at the web address: https://api.spacexdata.com/v4/

IMPORT LIBRARIES → REQUEST GET API / JSON LOADS / NORMALIZE JSON → CONSTRUCT OUR DATASET USING THE DATA JSON WE HAVE OBTAINED → FILTER THE DATAFRAME TO ONLY INCLUDE FALCON 9 LAUNCHES → DATA READY FOR WRANGLING

Datasets were collected using wiki page at the web address:
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

IMPORT LIBRARIES → REQUEST FALCON 9 LAUNCH WIKI PAGE FROM ITS URL → EXTRACT ALL COLUMN / VARIABLE NAMES FROM THE HTML TABLE HEADER → CREATING A DATAFRAME BY PARSING THE LAUNCH HTML TABLES → DATA READY FOR WRANGLING

Data Collection – SpaceX API

1. IMPORT LIBRARIES:

```python
# Requests allows us to make HTTP requests which we will u
import requests
# Pandas is a software library written for the Python prog
import pandas as pd
# NumPy is a library for the Python programming language,
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime
```

2. REQUEST GET API / JSON LOADS/ NORMALIZE JSON:

```python
# Use json_normalize meethod to convert the json result into a dataframe
import json
response = requests.get(static_json_url)
data_dict = json.loads(response.content)
data = pd.json_normalize(data_dict)
```

3. CONSTRUCTING DATASET USING JSON :

```python
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

4. FILTER DATAFRAME ONLY INCLUDE FALCON 9 LAUNCH:

```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data['BoosterVersion'] != 'Falcon 1']
data_falcon9.head()
```

5. DATA READY FOR WRANGLING:

```python
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Github URL:

https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Data Collection - Scraping

1. IMPORT LIBRARIES:

```
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

4. CONSTRUCTING DATASET USING WEB THE DATA:

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

2. REQUEST FALCON 9 LAUNCH WEB SCRAPING:

```
response = requests.get (static_url)
print(response.status_code)
print(response.content [0:100])
```

```
# Use BeautifulSoup() to create a BeautifulSoup object
soup = BeautifulSoup(response.content, "html.parser")
```

3. EXTRACT ALL COLUMN:

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```

5. Building dataframe:

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

Github URL:

https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/jupyter-labs-webscraping%20.ipynb

Data Wrangling

1. IMPORT LIBRARIES AND LOADING DATASET:

```
# Pandas is a software library written for the Python programming l
import pandas as pd
#NumPy is a library for the Python programming language, adding sup
import numpy as np
from js import fetch
import io

URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain
resp = await fetch(URL)
dataset_part_1_csv = io.BytesIO((await resp.arrayBuffer()).to_py())
```

2. NUMBER OF LAUCHES ON EACH SITE:

```
# Apply value_counts() on column
df['LaunchSite'].value_counts()
```

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

3. LANDING OUTCOMES IN SPECIFIC REGION:

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
False Ocean	2
None ASDS	2
False RTLS	1

4. FINDING THE BAD OUTCOMES TO FIND SUCCESS RATE

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
landing_class = []
for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
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

5. SUCESS RATE

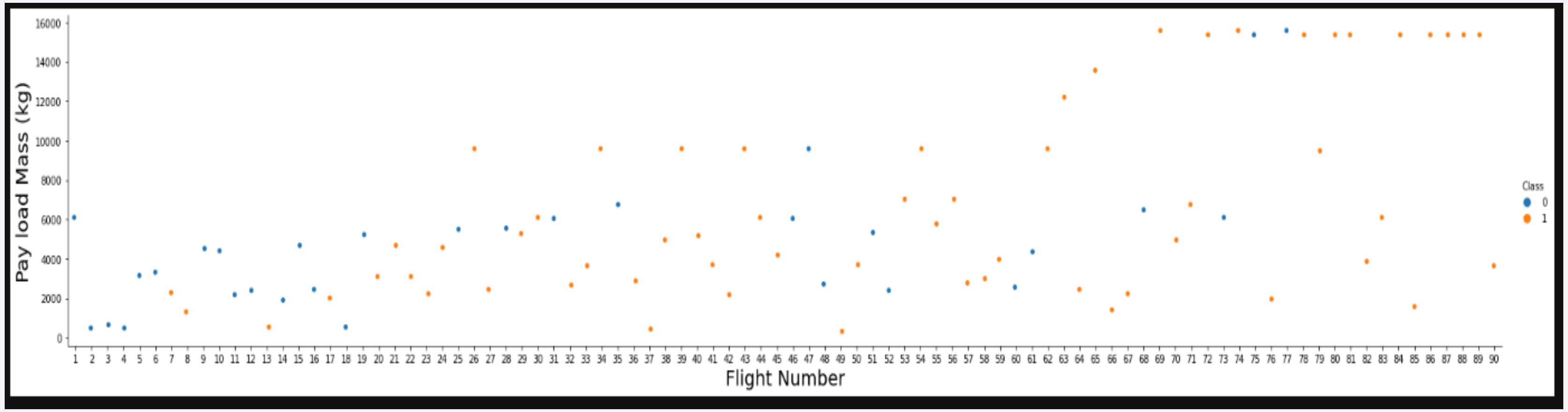
```
df["Class"].mean() * 100
```

```
66.66666666666666
```

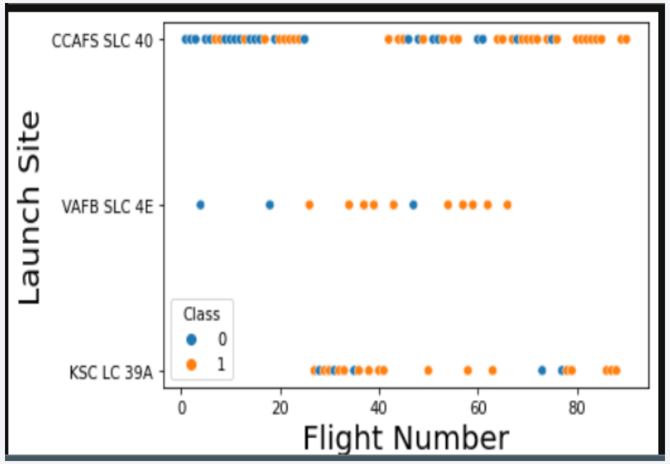
Github URL:

https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

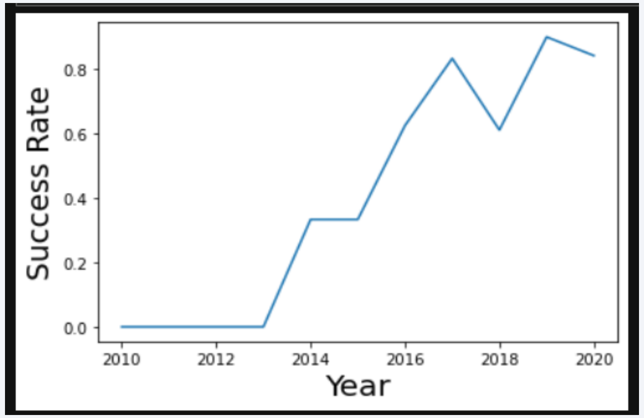
EDA with Data Visualization

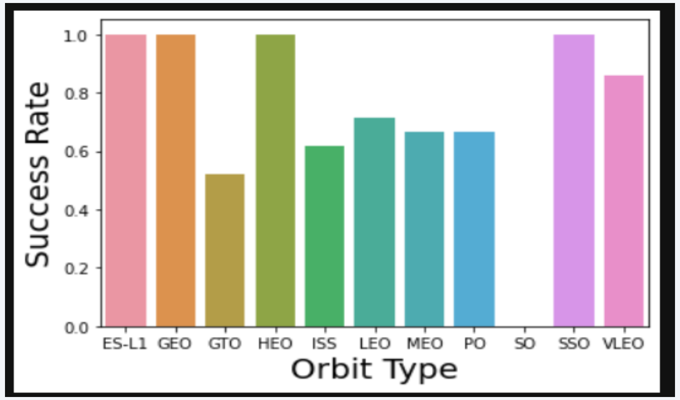


1. Plot the FlightNumber vs. PayloadMass(kg) and about the launch result



2. Relationship between Flight Number and Launch Site about the launch result





Github URL:

4. Launch success yearly trend

3. Success rate of each orbit type

https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb

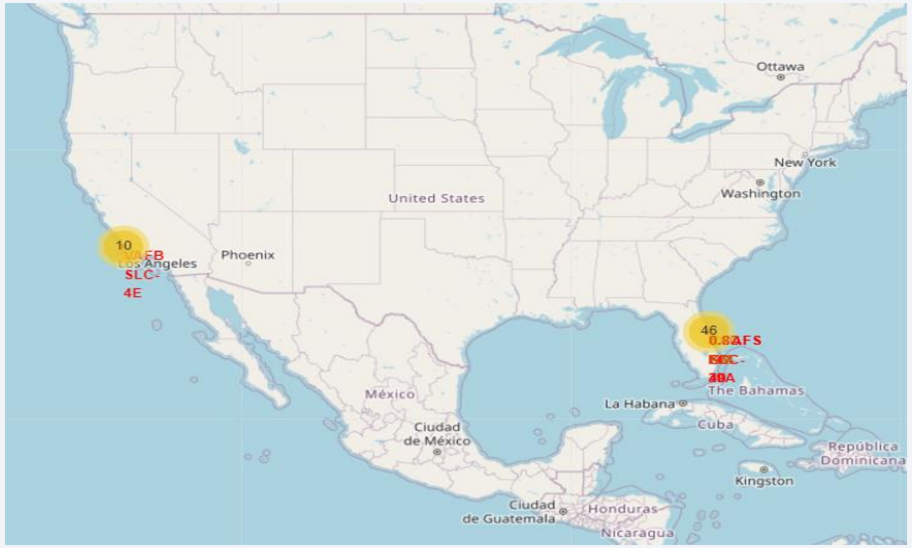
EDA with SQL

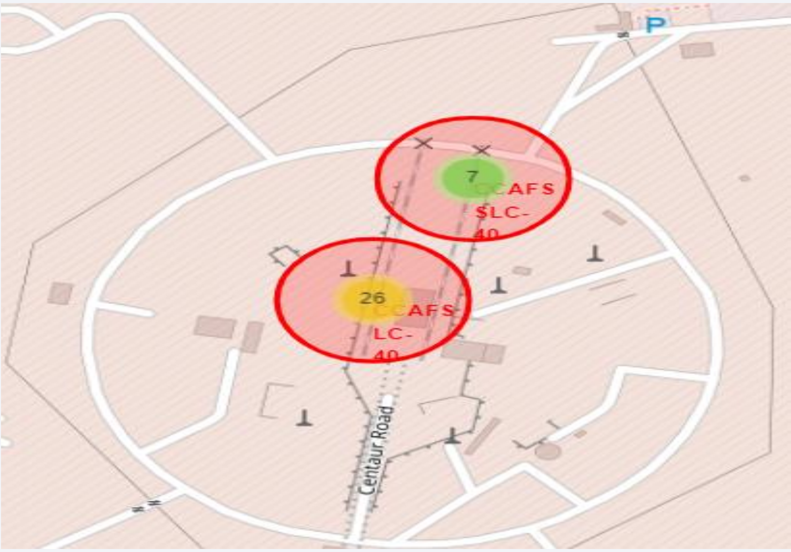
SQL queries to solve the assignment tasks:

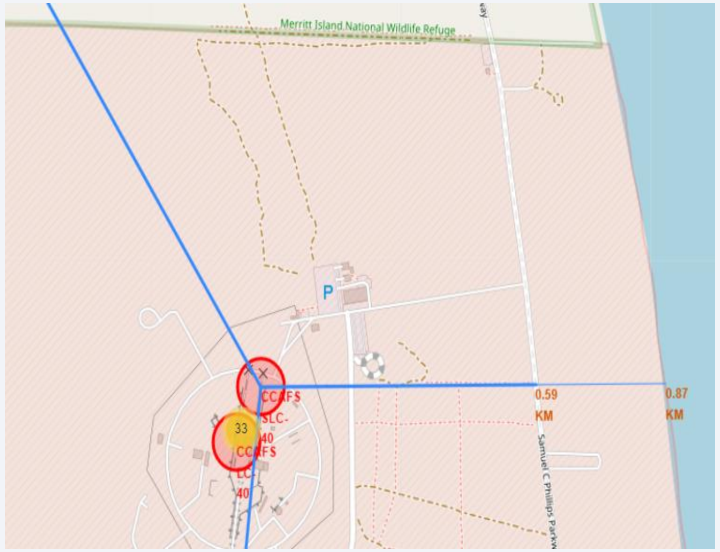
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful 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 records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Github URL: https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb

Build an Interactive Map with Folium

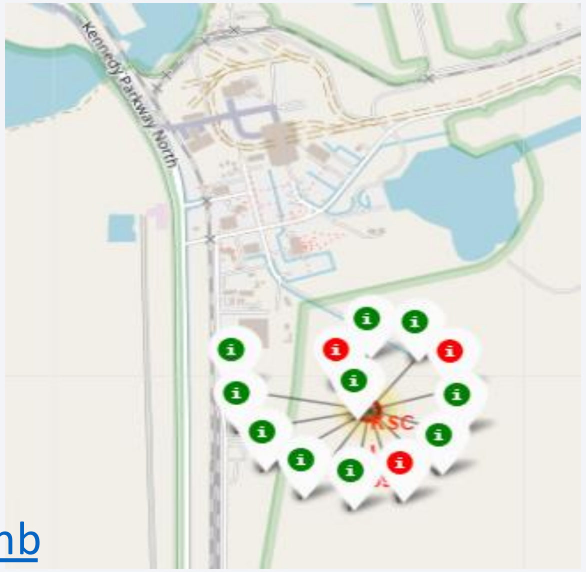




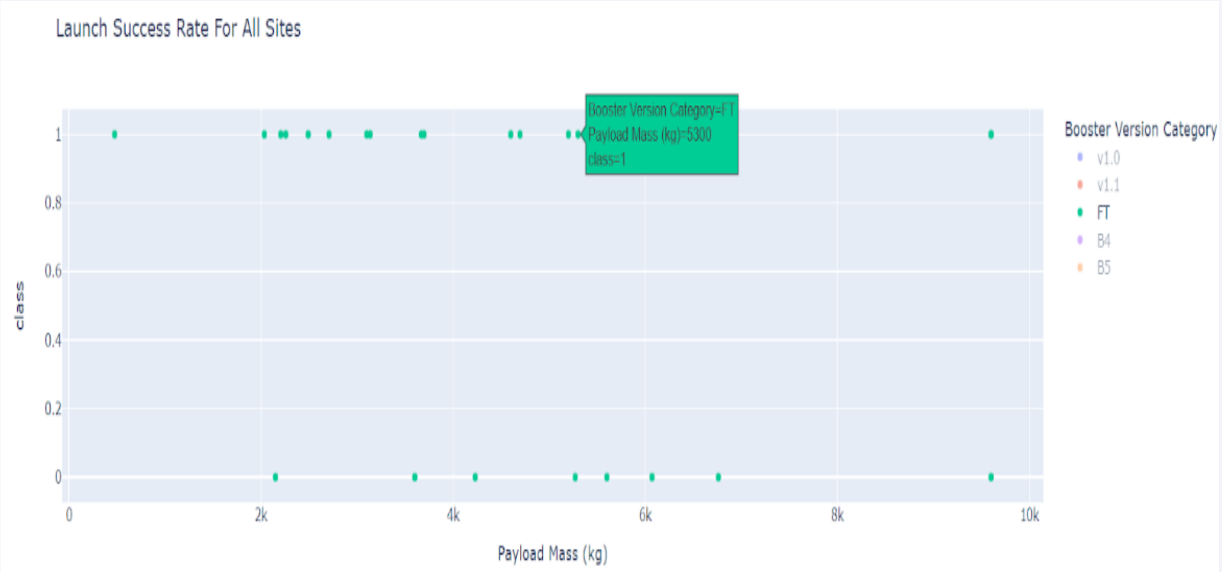


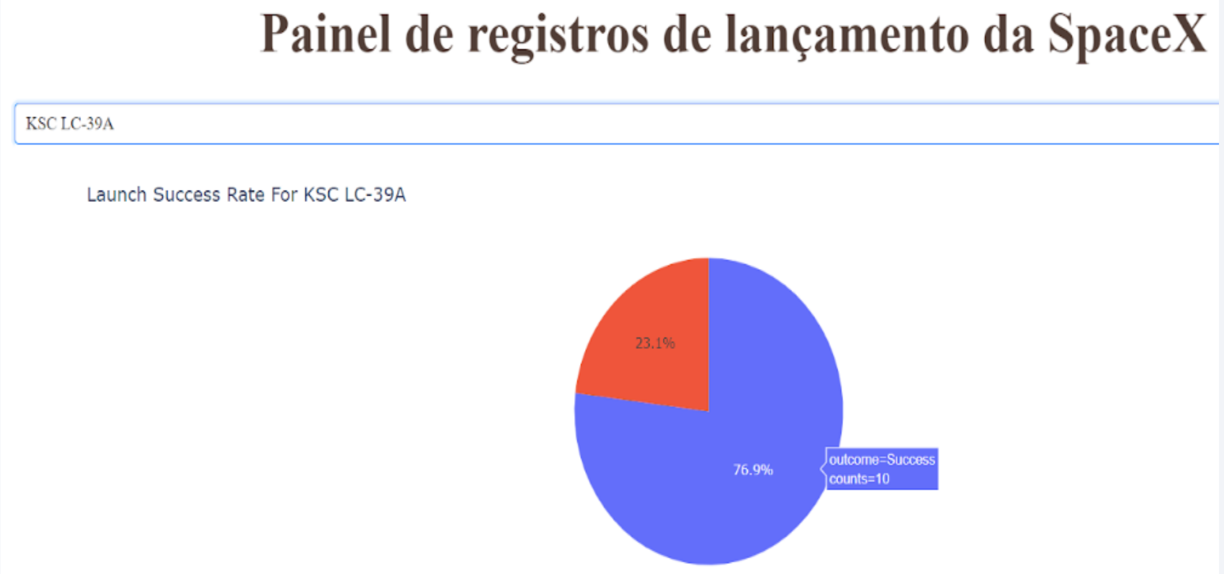
1- site_map = folium.Map - ADD map with informed coordinates
2- marker = folium.map.Marker - ADD markup or description
3- circle = folium.Circle - ADD circle at predefined coordinate
4- mouse_position = MousePosition - Add mouse position to get coordinate
5- lines=folium.PolyLine - ADD line on map
6- marker_cluster = MarkerCluster() - ADD grouping of markers
7- Site_map.add_child - ADD the previous items on the map

GITHUB: https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb



Build a Dashboard with Plotly Dash

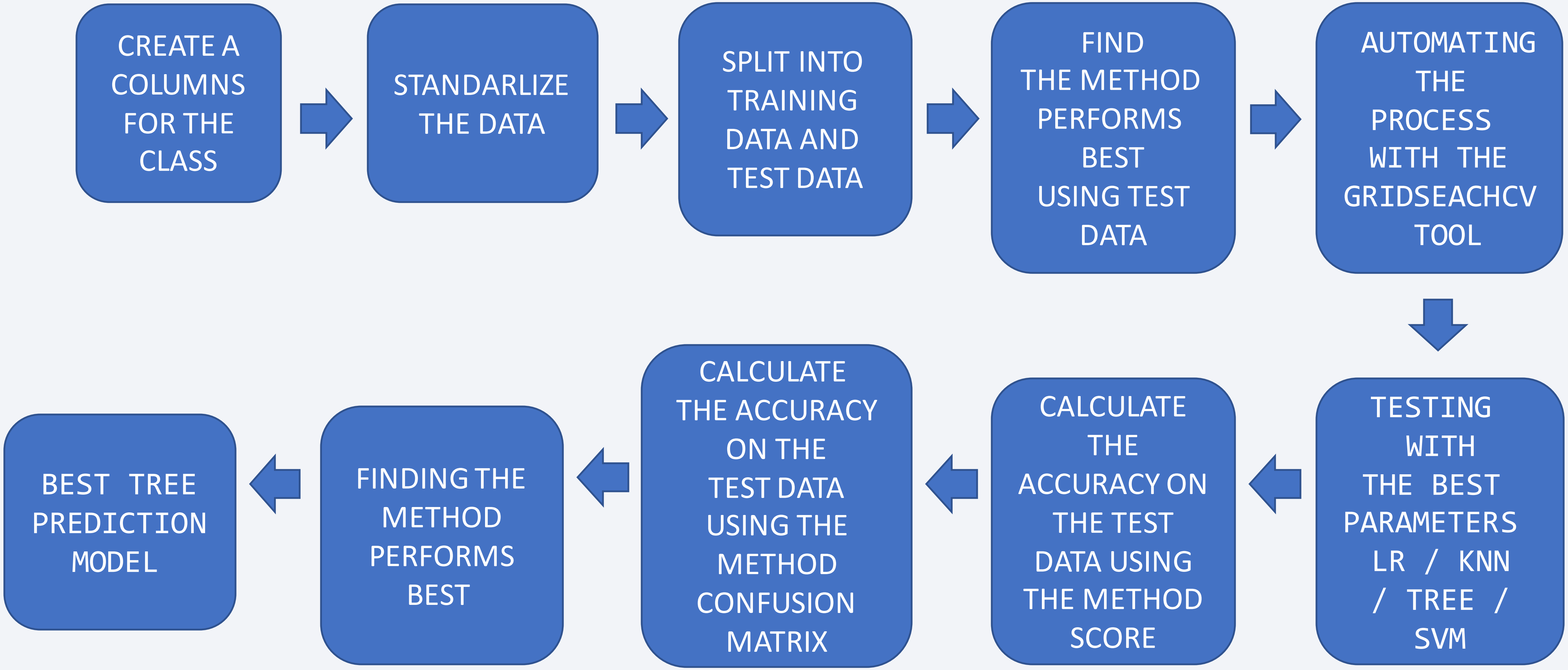




Which site has the biggest hit launches?
Which site has the highest launch success rate?
Which payload range(s) has the highest launch success rate?
Which payload range(s) has the lowest launch success rate?
Which version of F9 Booster (v1.0, v1.1, FT, B4, B5, etc.) launch success rate?

GITHUB: https://github.com/MarcioCarvalho2022/Python-Basics-for-data-Science-Project/blob/main/dash_final.py

Predictive Analysis (Classification) - To be continued



Predictive Analysis (Classification)

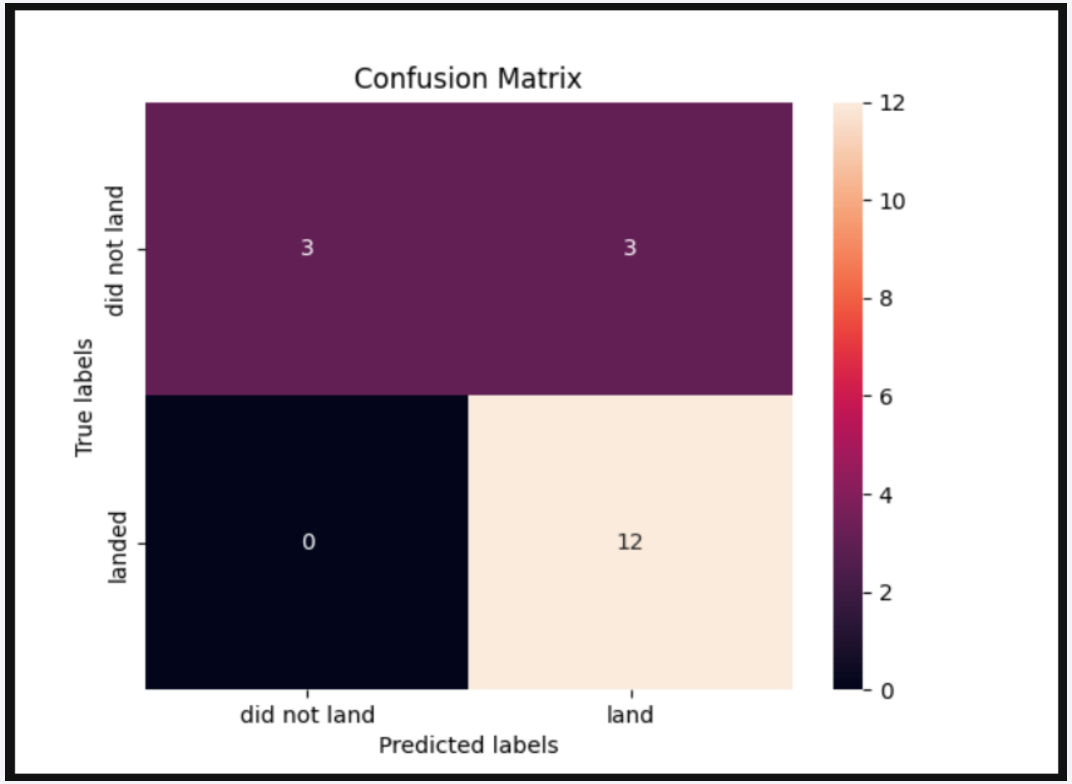
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()

tree_cv=GridSearchCV(tree, parameters, cv=10, scoring='accuracy')
tree_cv.fit(X_train,Y_train)

print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
print('Accuracy on test data is: {:.3f}'.format(tree_cv.score(X_test, Y_test)))

Accuracy on test data is: 0.944
```



GitHub URL:https://github.com/MarcioCarvalho2022/Final-Project-Data-Science-Coursera/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

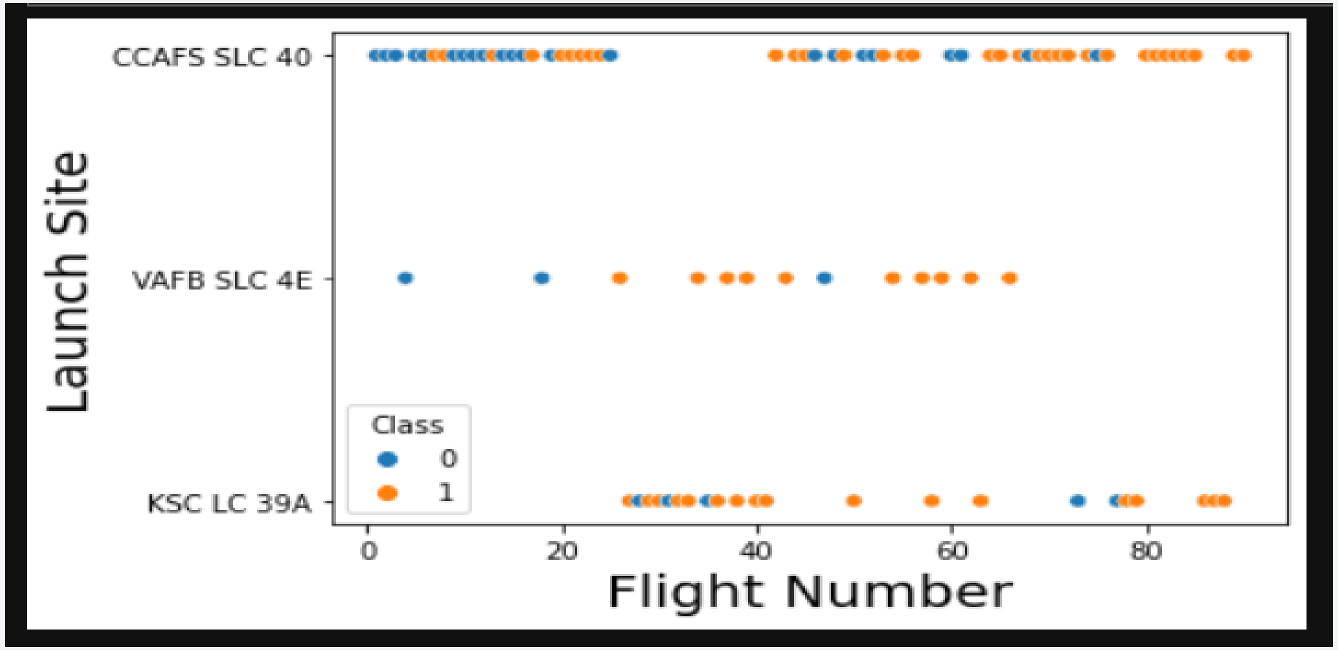
Results

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

Section 2

Insights drawn from EDA

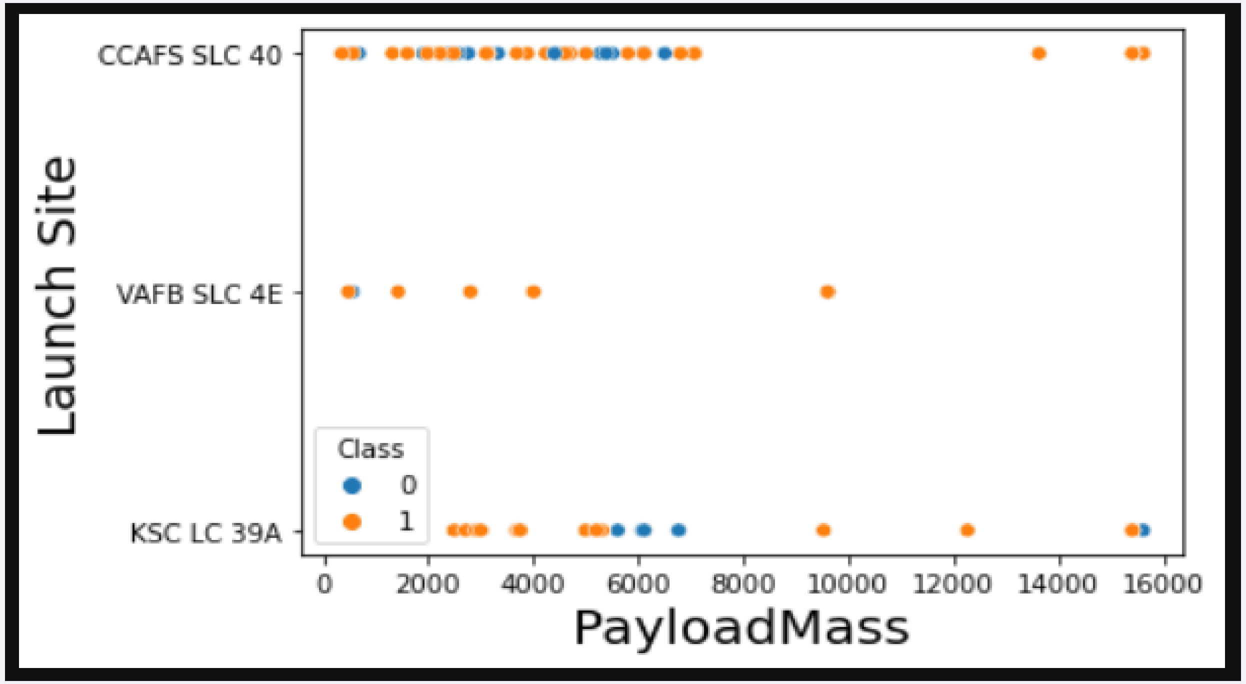
Flight Number vs. Launch Site



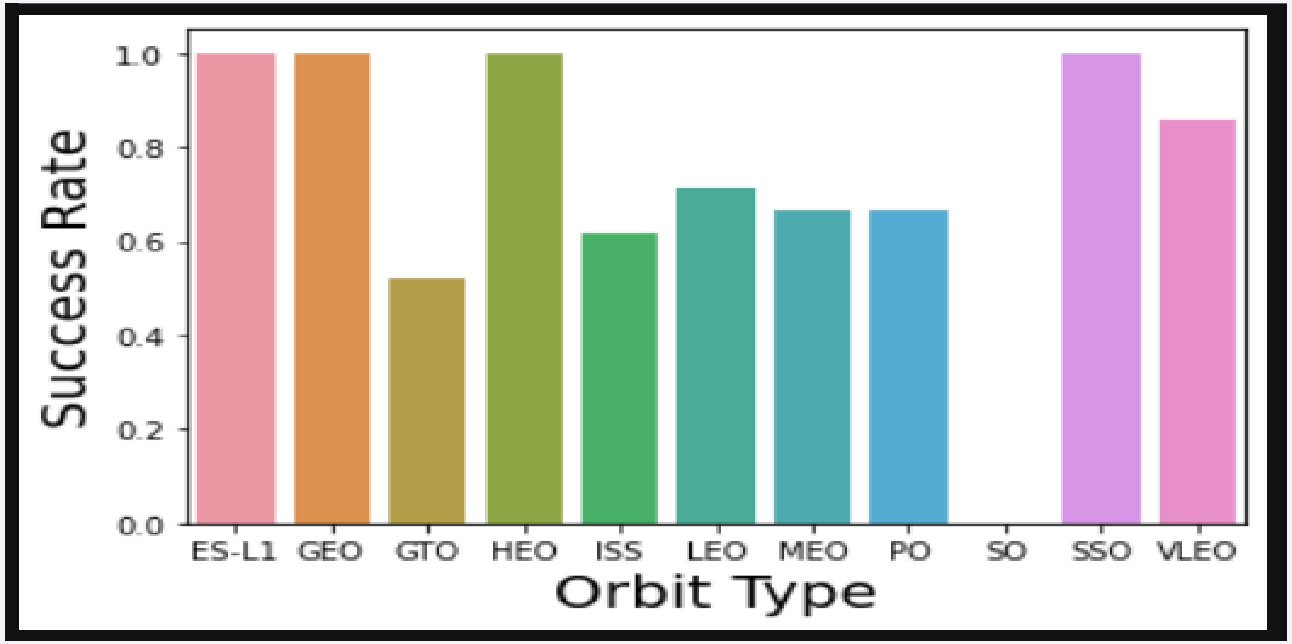
- ✓ Success rate PFB SLC 4E and KSC LC 39A is higher CCAFS SLC 40 even with lower launch amounts.
- ✓ As more experience launches on each site, the success rate increases.

Payload vs. Launch Site

- ✓ The payload with the highest launch success rate is between 2000-4000kg
- ✓ The payload with the lowest launch success rate is between 4000-8000kg
- ✓ The payload between 8000 and 15000kg has not failed to launch any site.



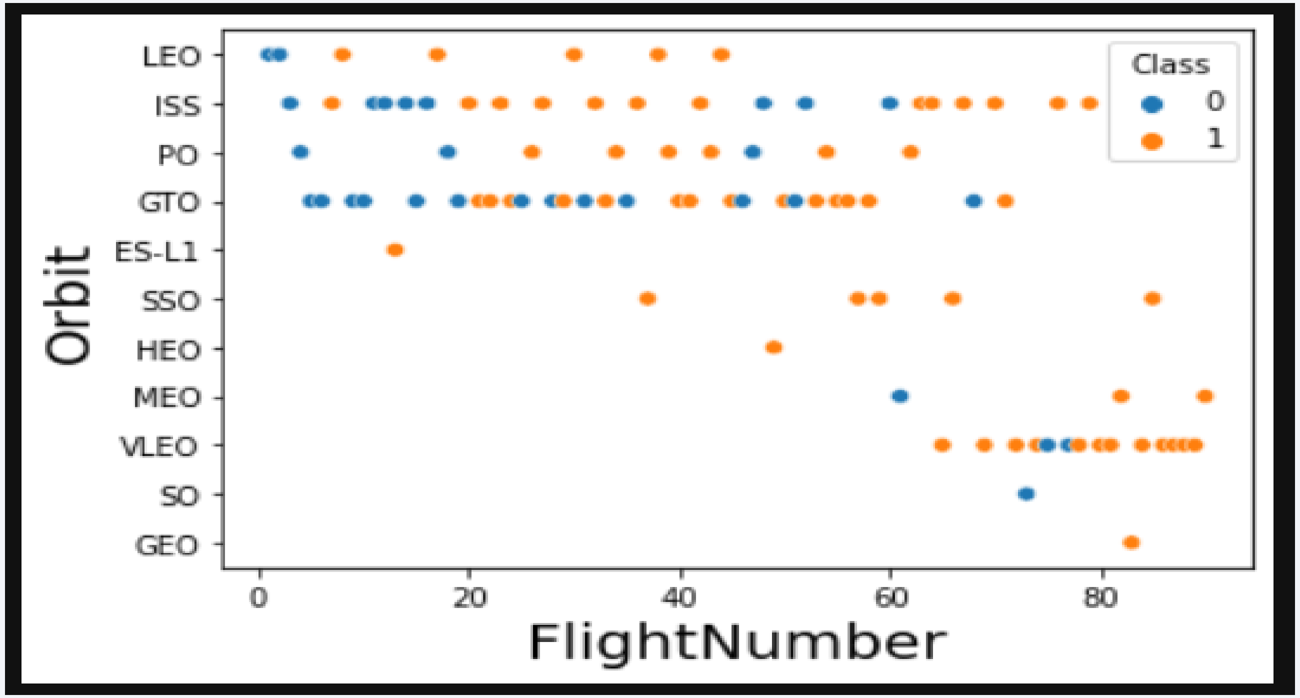
Success Rate vs. Orbit Type



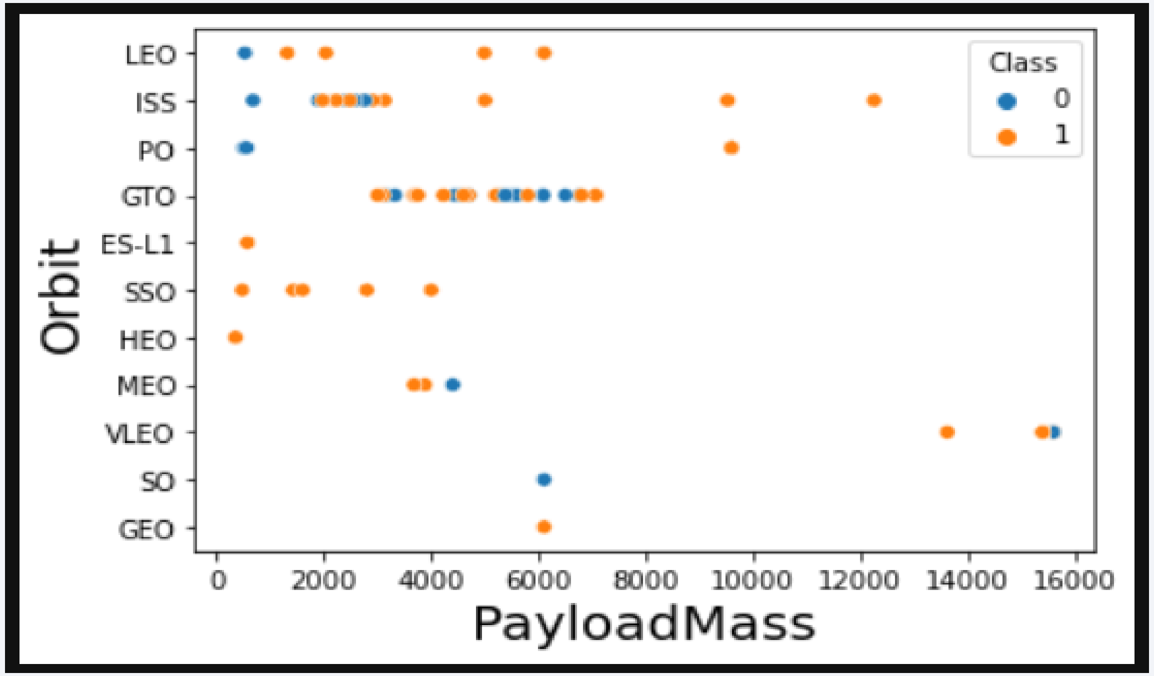
✓ The ES-L1, GEO, HEO, SSO orbits have 100% success rates.

Flight Number vs. Orbit Type

- ✓ Due to the greater proximity to the earth operating at the observation points, the VLEO orbit has been more used in recent launches.
- ✓ LEO's orbit has increased the rate of launches with more experience.
- ✓ The GTO orbit has a very large variance of success and failure and should be avoided until you find out the real reason.



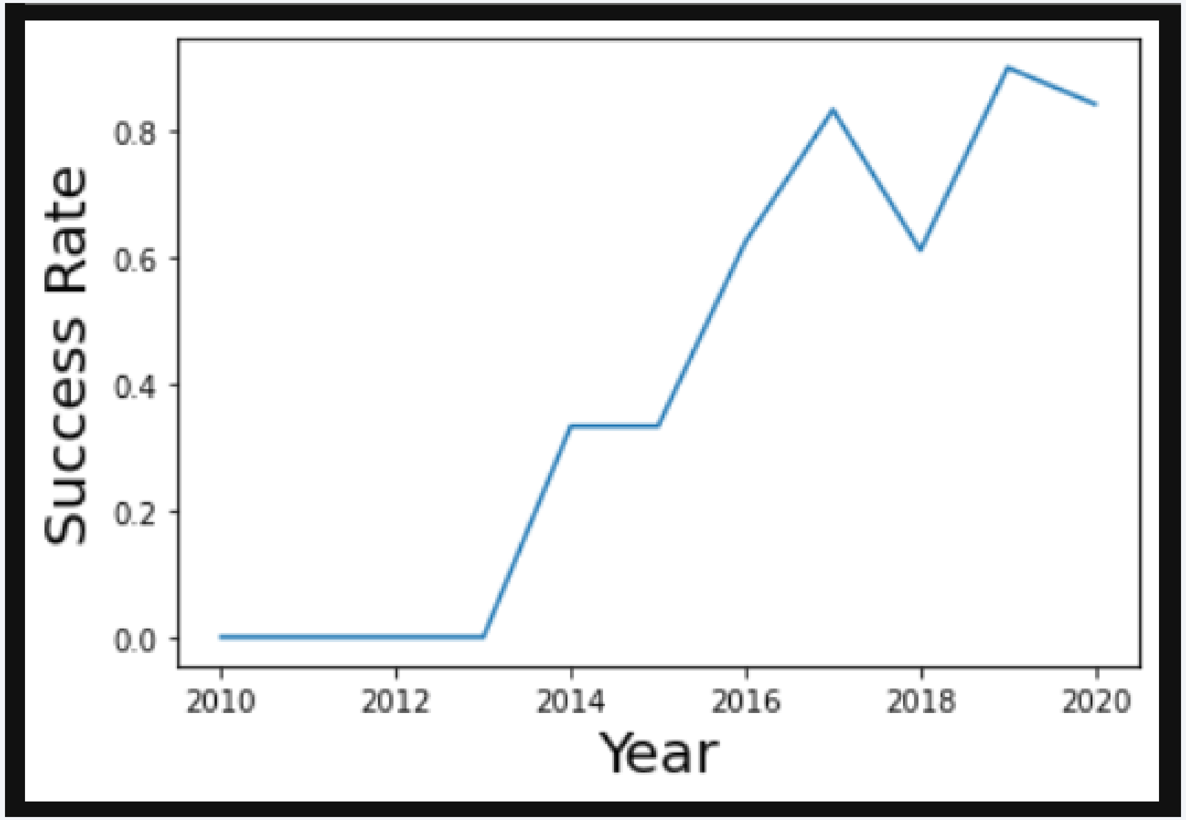
Payload vs. Orbit Type



- ✓ For the LEO, ISS and PO orbits after a payload of 4000kg the success rate was 100%.
- ✓ SSO orbit had 100% success rate with payloads up to 4000kg.
- ✓ GTO orbit has unstable success rate regardless of payload.

Launch Success Yearly Trend

- ✓ The success rate from 2013 had a great growth until 2017, with the growth being interrupted in 2018 and the rate increasing again in 2019. It is possible that the experience had a great impact on success.
- ✓



All Launch Site Names

DISTINCT filter was used to show unique values

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1

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

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

Launch Site Names Begin with 'CCA'

To show the 5 lines LIMIT was used

```
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 _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

To find the result we use SUM

```
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

To find the result we use AVG

```
sql SELECT AVG (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version like 'F9 v1.1%'

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

AVG (PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

To find the result we use MIN

```
sql SELECT MIN(Date) FROM SPACEXTBL WHERE Mission_outcome LIKE 'Success%'

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

MIN(Date)
01-03-2013

Successful Drone Ship Landing with Payload between 4000 and 6000

To find the result we use BETWEEN

```
sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landing _Outcome" = 'Success (drone ship)'
```

```
 * 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

To find the result we use COUNT

```
sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME like 'Success%'
```

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

Mission_Outcome	COUNT(MISSION_OUTCOME)
Success	100

```
sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME = 'Failure (in flight)'
```

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

Mission_Outcome	COUNT(MISSION_OUTCOME)
Failure (in flight)	1

Boosters Carried Maximum Payload

To find the result we use MAX and Subquery

```
sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

To find the result we use SUBST(DATE,Y,M)

```
sql SELECT substr(Date, 4, 2) AS Month, substr(Date, 7, 4) as Year,BOOSTER_VERSION, "Landing _Outcome",launch_site FROM SPACEXTBL where substr(Date
```

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

Month	Year	Booster_Version	Landing _Outcome	Launch_Site
01	2015	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04	2015	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

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

To find the result we use SUBST(DATE,Y,M) AND BETWEEN

```
sql Select "Landing _Outcome",COUNT(*) AS qty FROM SPACEXTBL WHERE substr(Date,7)||substr(Date,4,2)||substr(Date,1,2) between '20100604' and '20170
```

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

Landing _Outcome	qty
Success (drone ship)	5
Success (ground pad)	3

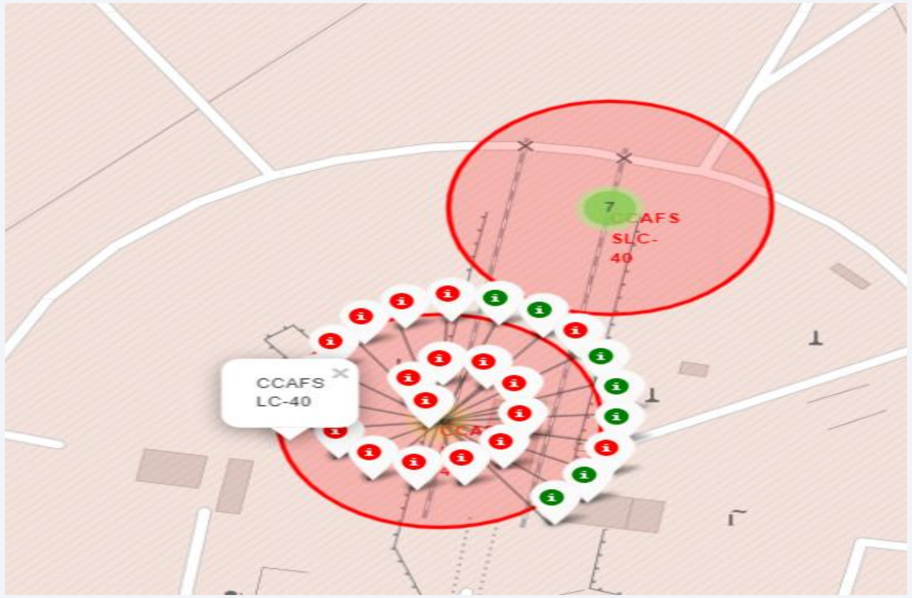
Section 3

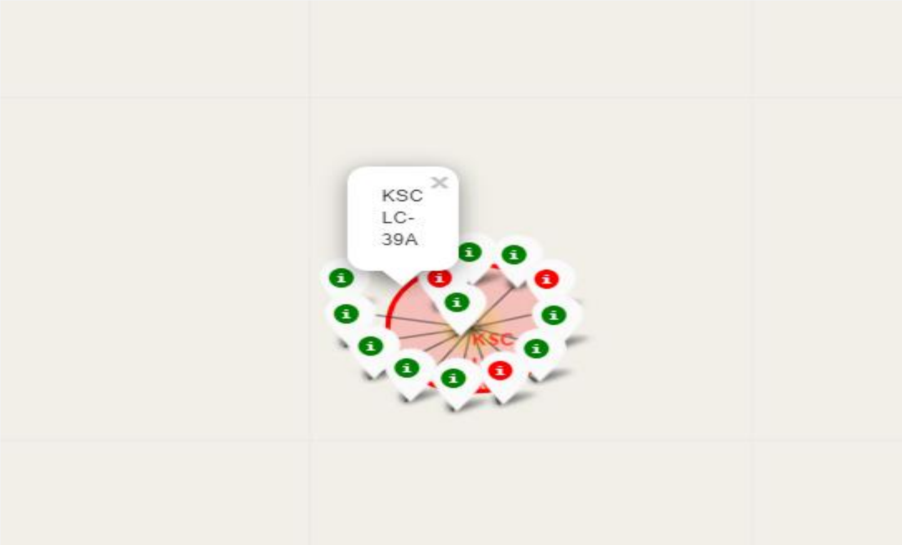
Launch Sites Proximities Analysis

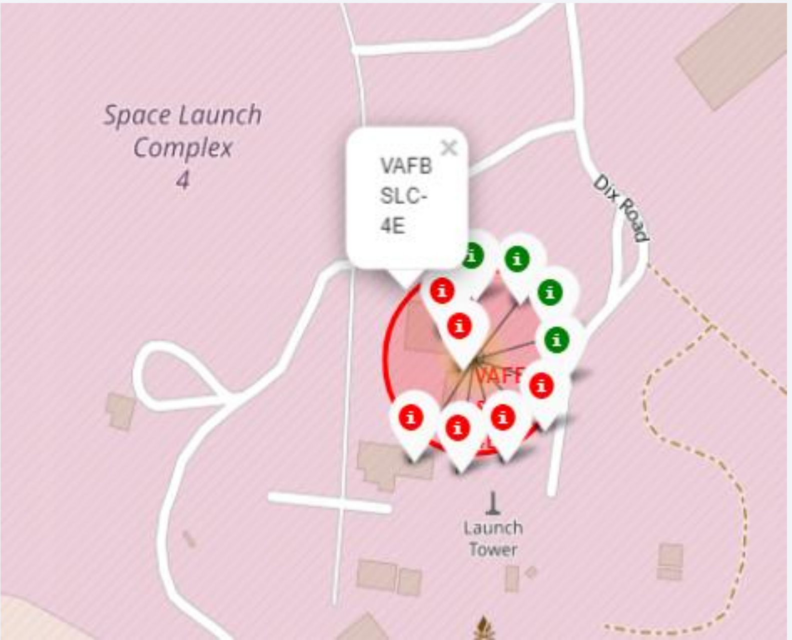
ALL LAUNCH SITES

✓ ALL LAUNCH LOCATIONS ARE NEAR THE SEA

Launch results marked by colors on the map







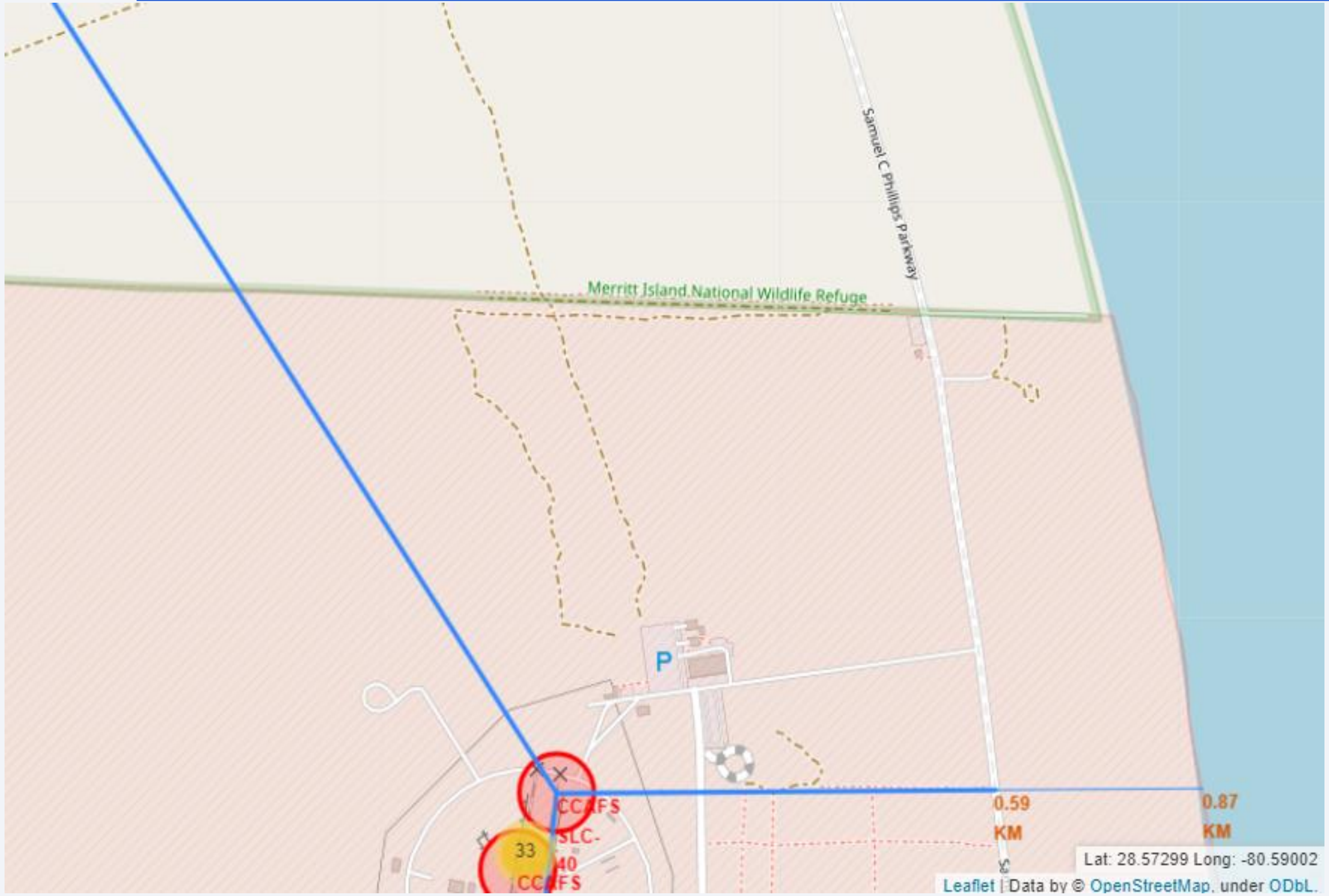
Landing Outcome:

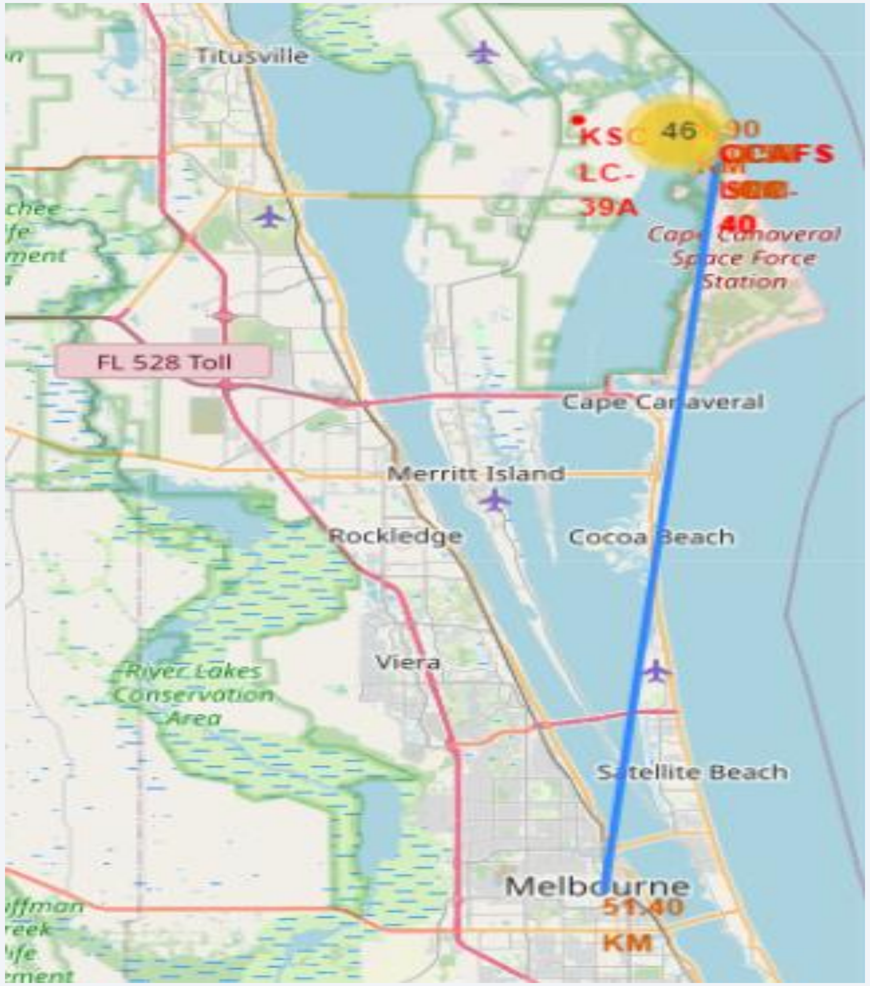
Green = success

Red = failure

✓ KSC LC 39A HAD THE BEST SUCCESS RATE

Launch site and surroundings



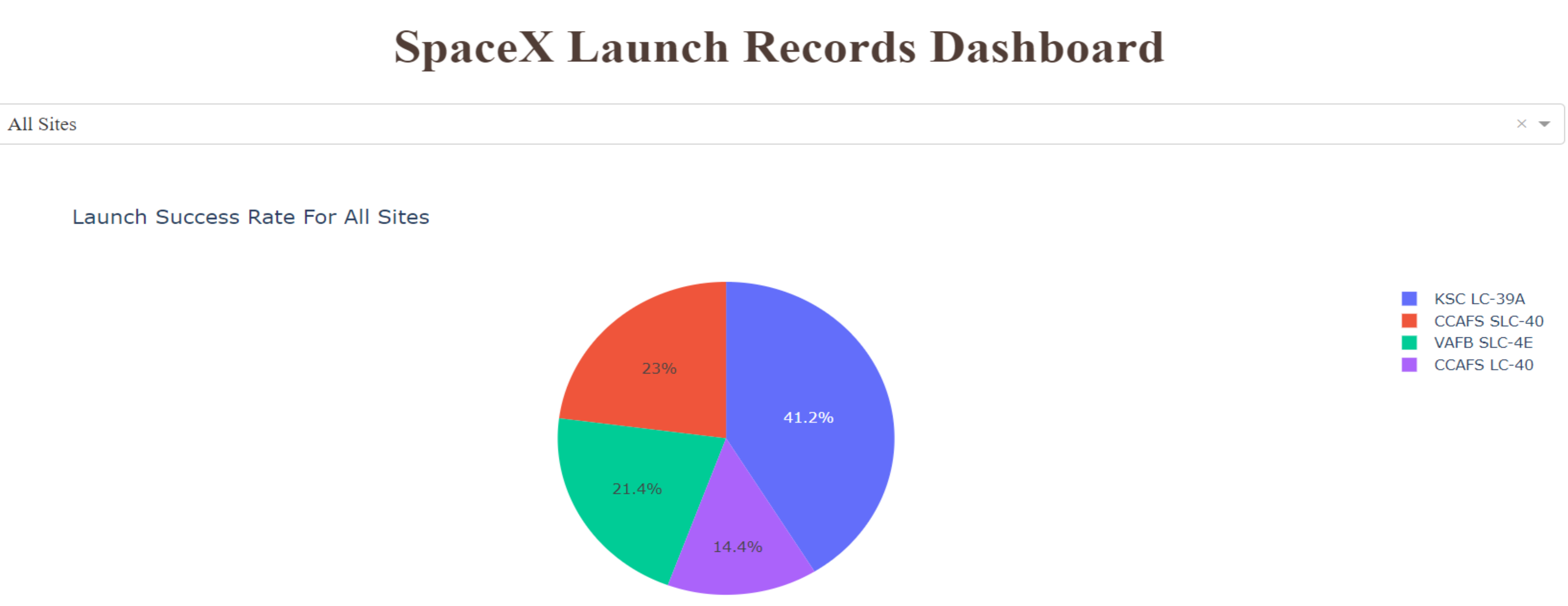


Launch sites have reasonable distances to railroads, highways, the coast and large urban centers

Section 4

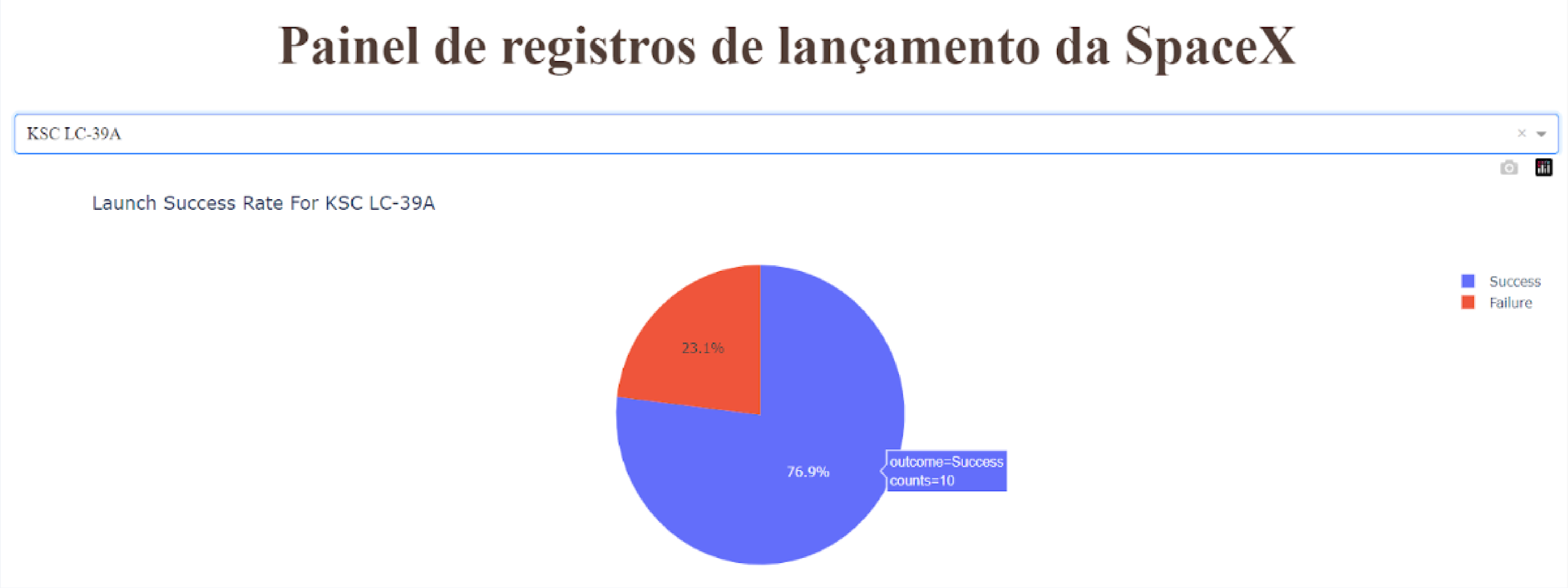
Build a Dashboard with Plotly Dash

Launch success count for all sites



- ✓ KSC LC 39A has the highest hit score with 41,2%
- ✓ CCAFS LC-40 has the lowest success score with 14,4%

Launch site with highest success rate

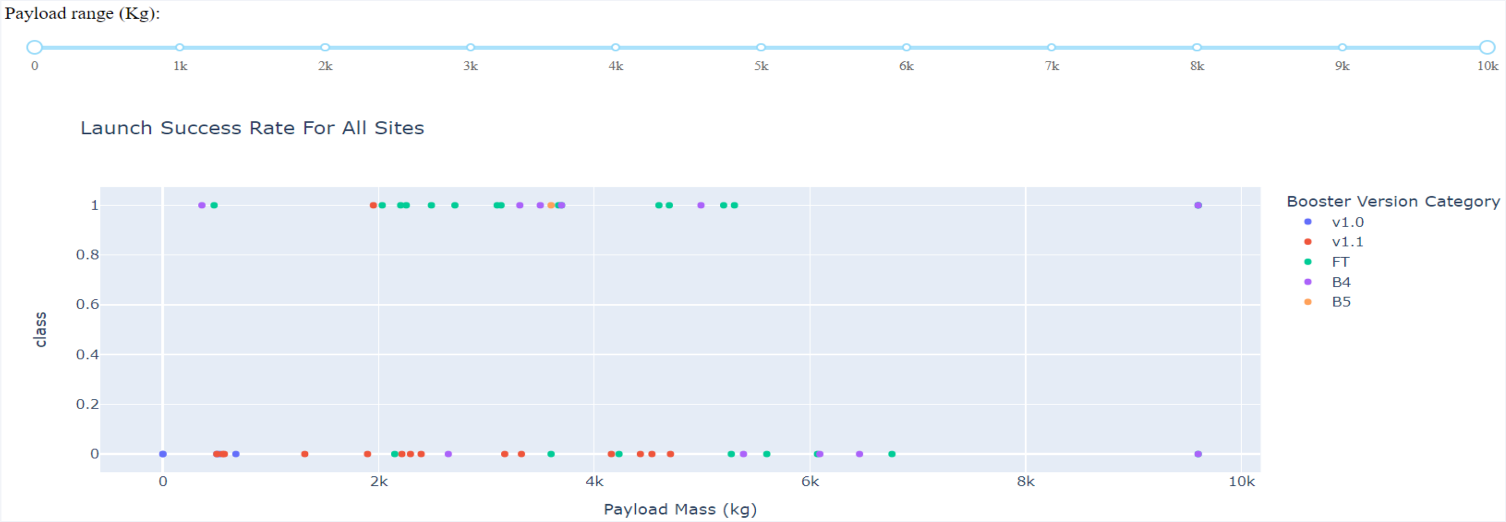


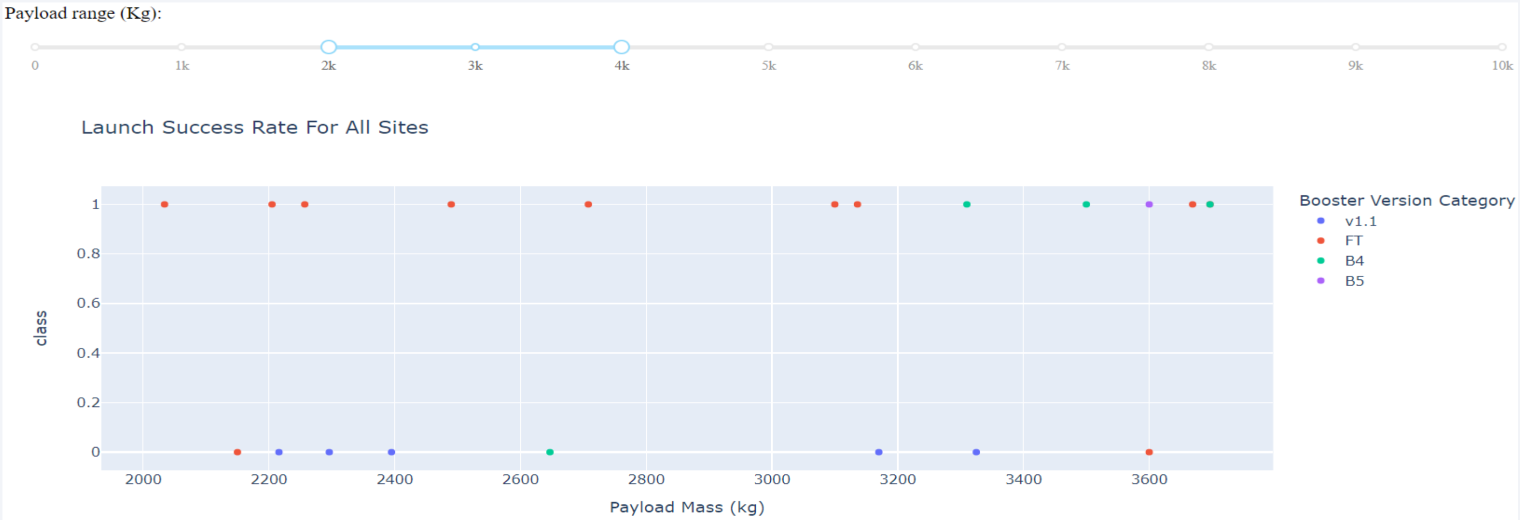
✓ KSC LC 39A had a 76,9% success rate.

✓ KSC LC 39A had a 23,1% failure rate.

Payload vs. Launch Outcome for all sites

- ✓ Booster version V1.0 and V1.1 with load up to 4000kg has the lowest success rate.
- ✓ Payloads between 4000 and 8000 kg had the lowest success rate for all sites.



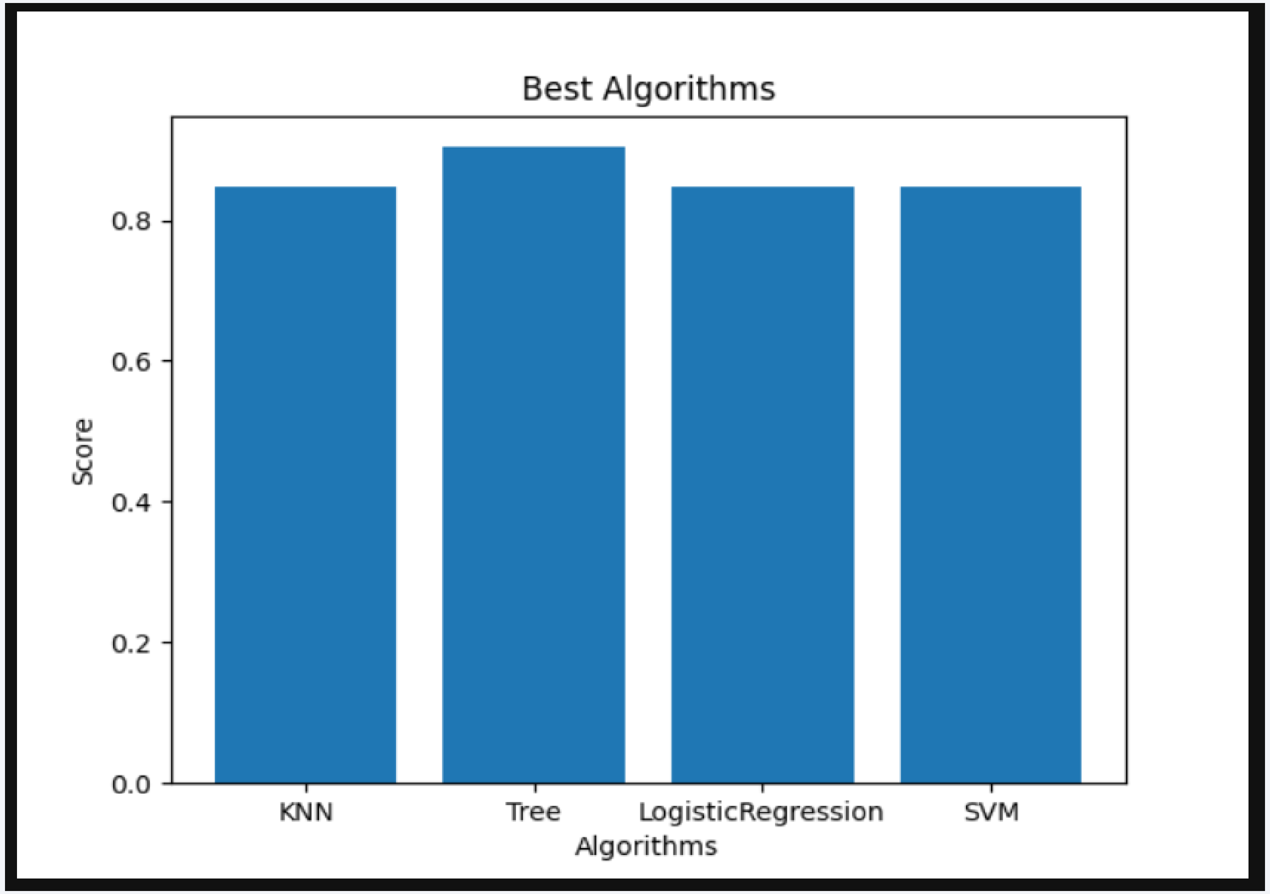


- ✓ Booster version FT in the payload range between 2000-4000kg has the highest success rate.

Section 5

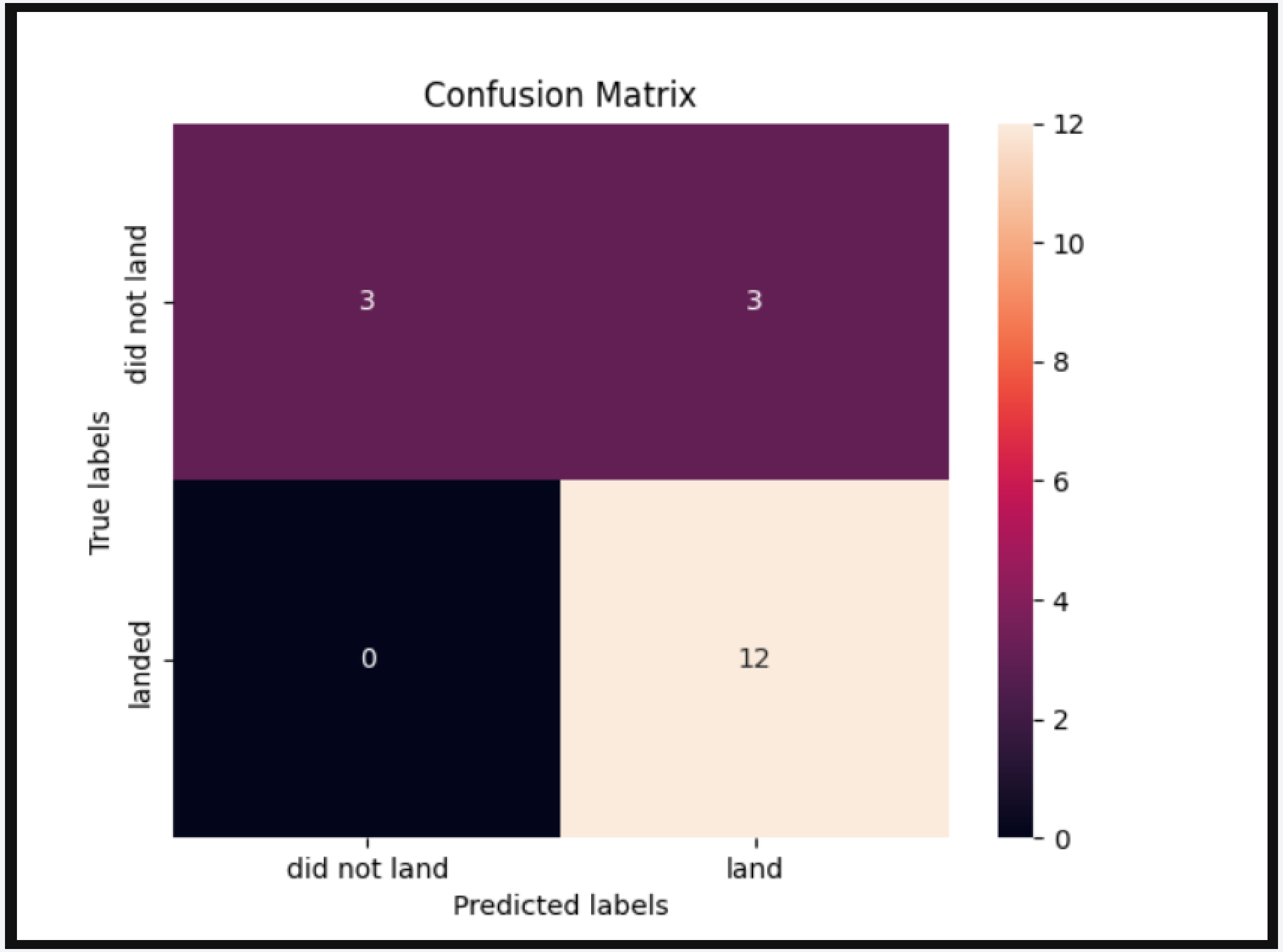
Predictive Analysis (Classification)

Classification Accuracy



✓ Decision Tree model has the highest classification accuracy

Confusion Matrix



- ✓ Decision tree model confusion matrix
- ✓ Predicted success results (TP) had an assertiveness of 80% and False positive (FP) 20%.
- ✓ With advancing technology and new calculation algorithms, the results for true negatives and false positives tend to become more accurate.

Conclusions

- ✓ Launch site with the most successful result was the KSC LC 39A with a payload of up to 5500kg for larger loads do not use Booster version FT, for the other Boosters there is not enough data.
- ✓ Booster version V1.0 and V1.1 with load up to 4000kg has the lowest success rate.
- ✓ Payloads between 4000 and 8000 kg had the lowest success rate for all sites.
- ✓ Booster version FT in the payload range between 2000-4000kg has the highest success rate.
- ✓ The success rate from 2013 had a great growth until 2017, with the growth being interrupted in 2018 and the rate increasing again in 2019. It is possible that the experience had a great impact on success.
- ✓ For the LEO, ISS and PO orbits after a payload of 4000kg the success rate was 100%.
- ✓ SSO orbit had 100% success rate with payloads up to 4000kg.
- ✓ GTO orbit has unstable success rate regardless of payload.
- ✓ Due to the greater proximity to the earth operating at the observation points, the VLEO orbit has been more used in recent launches.
- ✓ LEO's orbit has increased the rate of launches with more experience.
- ✓ The GTO orbit has a very large variance of success and failure and should be avoided until you find out the real reason.

Appendix

Acknowledgment:

"I thank God first.
There are all the professors, classmates and family for their support and help given to this eternal learner."

- Marcio Carvalho

https://www.coursera.org/

https://cloud.ibm.com/

https://github.com/

https://www.infnet.edu.br/infnet/home/

https://www.google.com.br/

https://github.com/MarcioCarvalho2022/Final-Project-Data-Science-Coursera

Thank you!