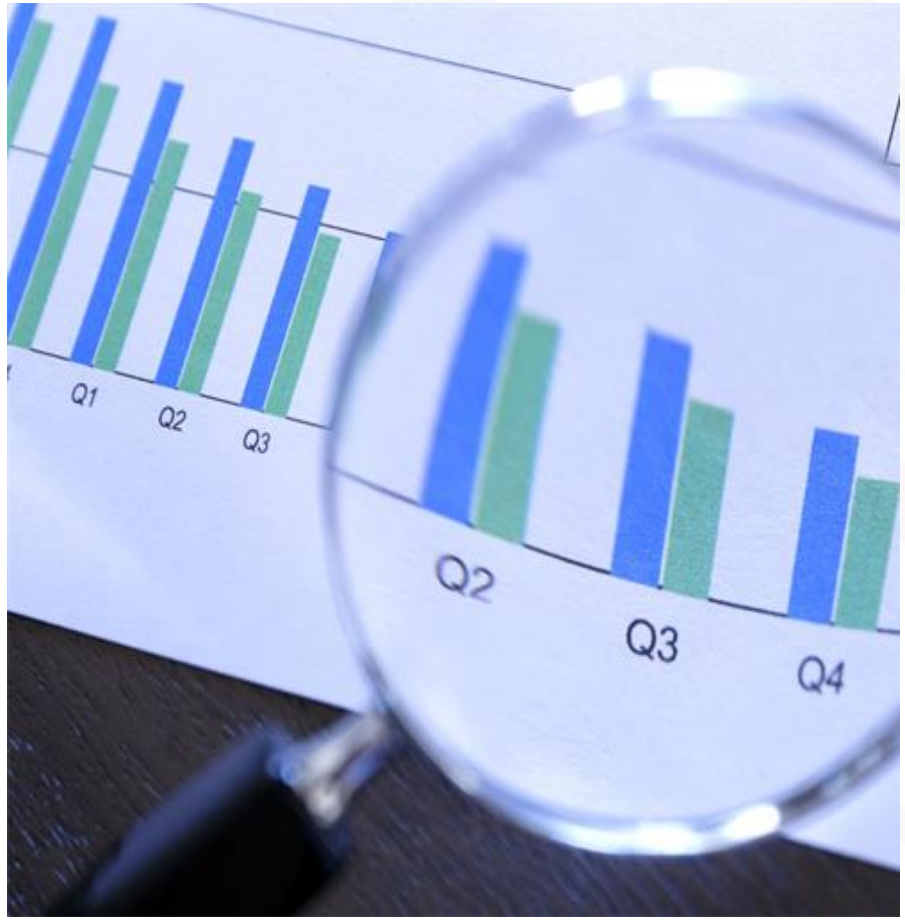


SpaceX Falcon 9 Landing Analysis

KAMY KAMBERE MUISSA

March 2025

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- We are in the era of space exploitation
- The SpaceX Falcon 9 has one major advantage over its competitors: the reusability of its first stage.
- We present an analysis of the various SpaceX Falcon9 first-stage landings

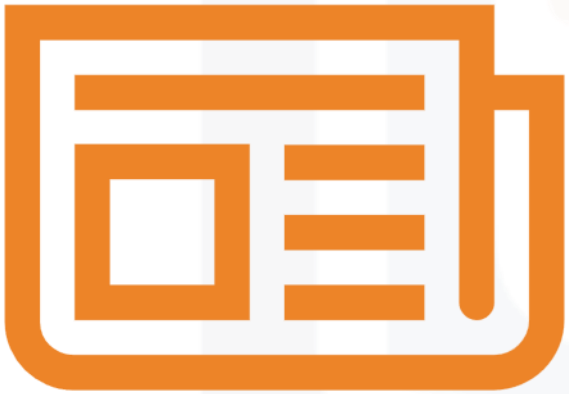
- Data Collection and Wrangling
- Data Exploration
  - Data Analysis with SQL
  - Statistical Analysis
  - Data Visualization
- Build, evaluate and refine predictive models for more exciting insights
- Present and discuss analysis results

INTRODUCTION

- Companies are making space travel affordable for everyone
- This is still very expensive and can cost up to 165 million dollars
- Among them, The Falcon9 of SpaceX seems to be the most interesting and least expensive, 62 million dollars
- This is because SpaceX manages to reuse the first stage
- As Data scientist we will work on a new rocket company, SpaceY, which will compete with SpaceX
  - Analyze different missions from different launch sites
  - Analyze landing success/failure rates for these launches
  - Determine, using Machine Learning techniques, whether the first stage will land correctly and can be reused.

METHODOLOGY

- To carry out this analysis, we will use an approach based on Machine Learning techniques.
- This includes:



- Data collection
- Data preprocessing
- Data modeling
- Model Training and Evaluation

RESULTS

DATA COLLECTION AND WRANGLING

- Request to the SpaceX API
- Clean the requested data

```
# Takes the dataset and uses the rocket column to call the API and append the data to
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, t

```
# Takes the dataset and uses the launchpad column to call the API and append the data
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload` we would like to learn the mass of the payload and the orbit that i

```
# Takes the dataset and uses the payloads column to call the API and append the data t
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

DATA COLLECTION AND WRANGLING

- Data Wrangling:
  - To find some patterns in the data
  - To determine what would be the label for training supervised models.

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

```
[10]: # Number of occurrence on each orbit
      # Apply value_counts on Orbit column
      Number_occurrence_per_Orbit = df['Orbit'].value_counts()
      Number_occurrence_per_Orbit
```

```
[10]: Orbit
      GTO      27
      ISS      21
      VLEO     14
      PO        9
      LEO       7
      SSO       5
      MEO       3
      HEO       1
      ES-L1     1
      SO        1
      GEO       1
      Name: count, dtype: int64
```

INTERACTIVE VISUAL ANALYTICS METHODOLOGY

To visualize the data we used:

- **Line Plot:** A line plot displays the relationship between two continuous variables over a continuous interval, showing the trend or pattern of the data
- **Scatter Plot:** A scatter plot visualizes the relationship between two continuous variables, displaying individual data points as dots on a two-dimensional plane, allowing for the examination of patterns, clusters, and correlations
- **Bar Plot:** A bar plot represents categorical data with rectangular bars, where the height of each bar corresponds to the value of a specific category, making it suitable for comparing values across different categories
- **Pie Chart:** A pie chart represents the proportion or percentage distribution of different categories in a dataset using sectors of a circular pie

PREDICTIVE ANALYSIS METHODOLOGY

For the purposes of prediction, the data will be divided into Train Data and Test Data, and we will use the following algorithms

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K Nearest Neighbors

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. The training data and test data should be assigned to the following labels.

`X_train, X_test, Y_train, Y_test`

[10]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y , random_state=2,test_size=0.2, shuffle=True)
```

we can see we only have 18 test samples.

[11]:
```
Y_test.shape
```

[11]:
```
(18, 1)
```

DATA EXPLORATION AND DATA PREPARATION

- SpaceX has made rocket launch data available through an API
- After cleaning the data and in particular managing missing data, we will focus on the Falcon 9 data.
  - Data mining shows that there are 4 launch sites:
    1. CCAFS LC-40
    2. VAFB SLC-4E
    3. KSC LC-39A
    4. CCAFS SLC-40
  - Each launch aims to a dedicated orbit: We have identified 11 different orbits

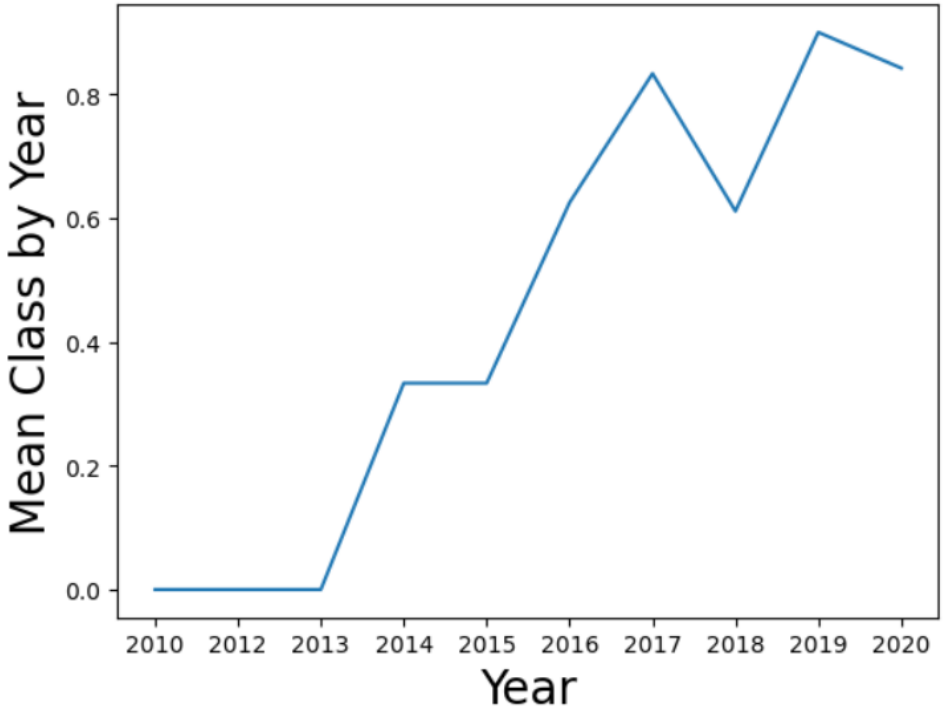
VISUALIZATION RESULTS

The analyses focused on the impact of certain parameters on the success or failure of the 1st stage landing, such as

- Relationship between Flight Number and Launch Site
- Relationship between Payload Mass and Launch Site
- Relationship between success rate of each orbit type
- Relationship between Flight Number and Orbit type
- Relationship between Payload Mass and Orbit type

Visualization of the launch success yearly trend

We can observe that the success rate since 2013 kept increasing till 2020



DATA ANALYSIS - SQL RESULTS

The objective here is to:

- Understand the SpacXx DataSet
- Load the dataset into the corresponding table in a Db2 database
- Execute SQL queries to answer assignment questions

Display the names of the unique launch site

```
[12]: %sql select distinct Launch_Site from
```

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

[12]:

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

DATA ANALYSIS AND VISUALIZATION - interactive map with Folium results

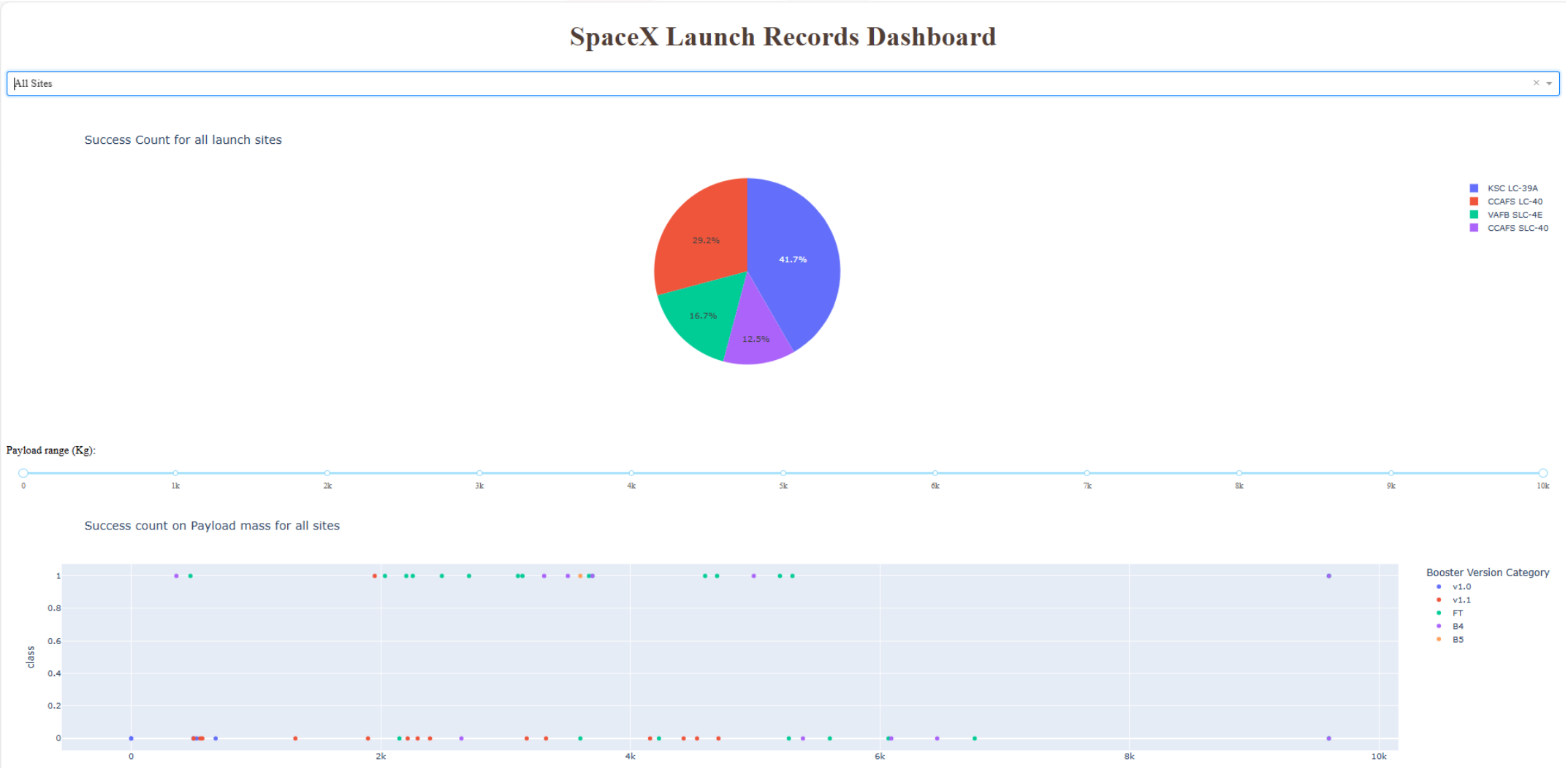
- The action included
  - Mark all launch sites on a map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities

DATA VISUALIZATION - DASHBOARD

An interactive Dashboard has been added to answer questions such as

- Which site has the largest successful 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 F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

DASHBOARD 1


SpaceX Launch Records Dashboard
All Sites
Success Count for all launch sites
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Success count on Payload mass for all sites
class
Booster Version Category
v1.0
v1.1
FT
B4
B5

DISCUSSION

- In the following section we will discuss different models for predicting the outcome of the landing based on the available data and parameters.
- To achieve this, we proceed as follows:
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data

OVERALL FINDINGS & IMPLICATIONS

Several prediction models were applied to the data, including

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K Nearest Neighbors

	Score
Decision Tree	0.889
Logistic Regression	0.833
SVM	0.833
KNN	0.833

CONCLUSION

- As the flight number increases over years, the first stage is more likely to land successfully
- We can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

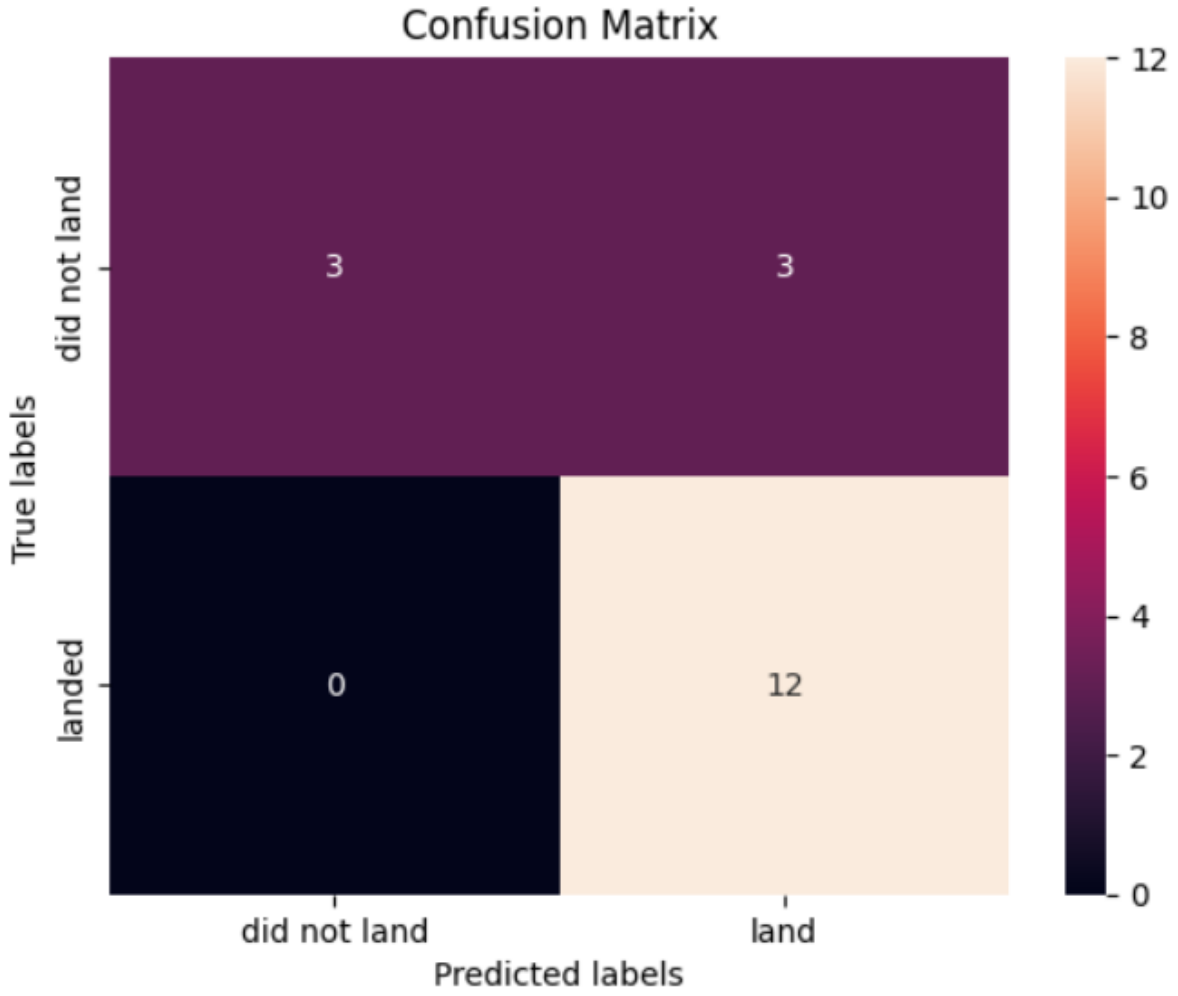
- For a better prediction, the Decision Tree method seems to be the most suitable because it offers the best score.

APPENDIX

Logistic Regression Confusion Matrix

- logistic regression object with a GridSearchCV object logreg_cv with cv = 10

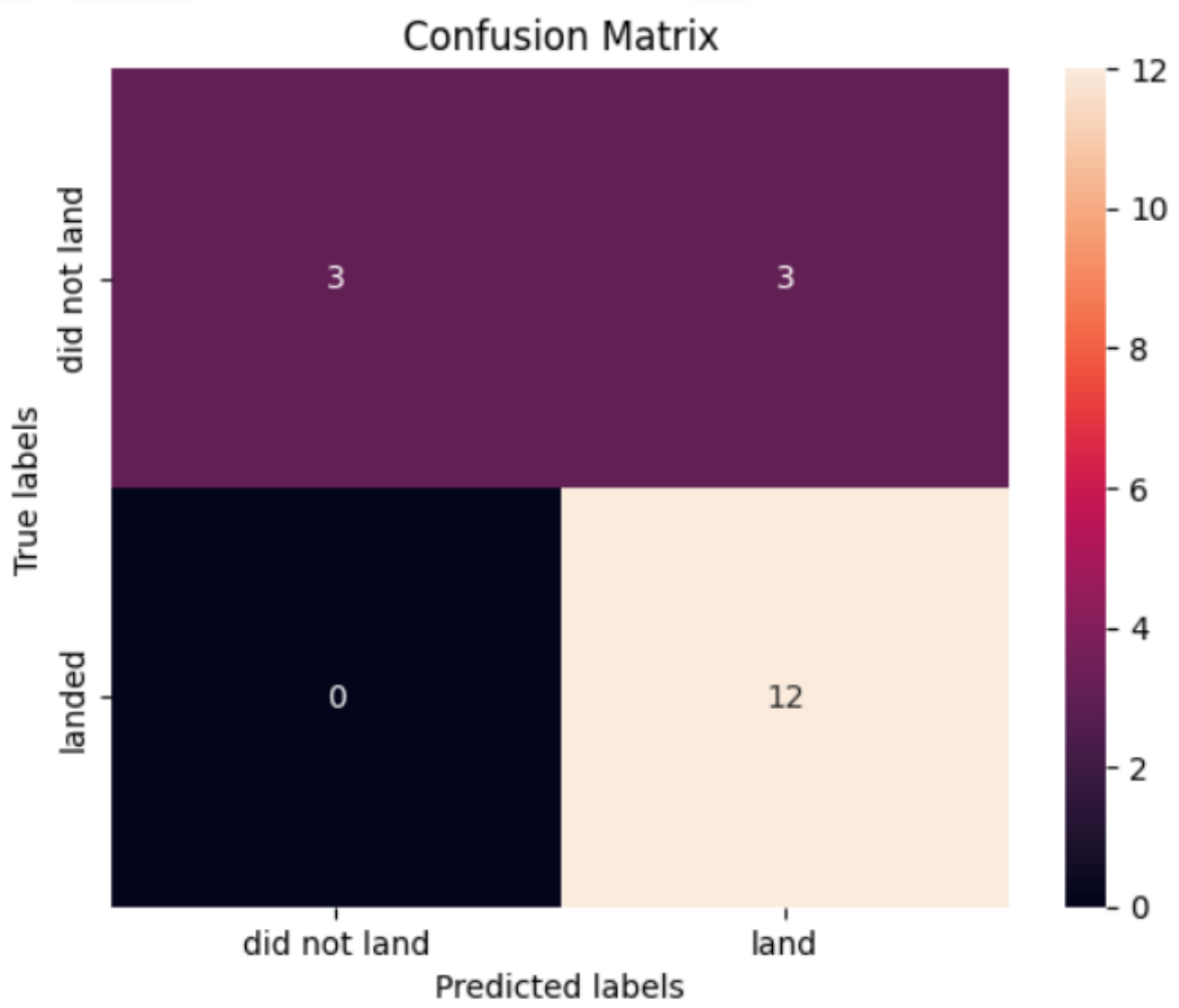
- Score method - Accuracy: 0.8333333333333334



Support Vector Machine Confusion Matrix

- support vector machine object with a GridSearchCV object svm_cv with cv = 10

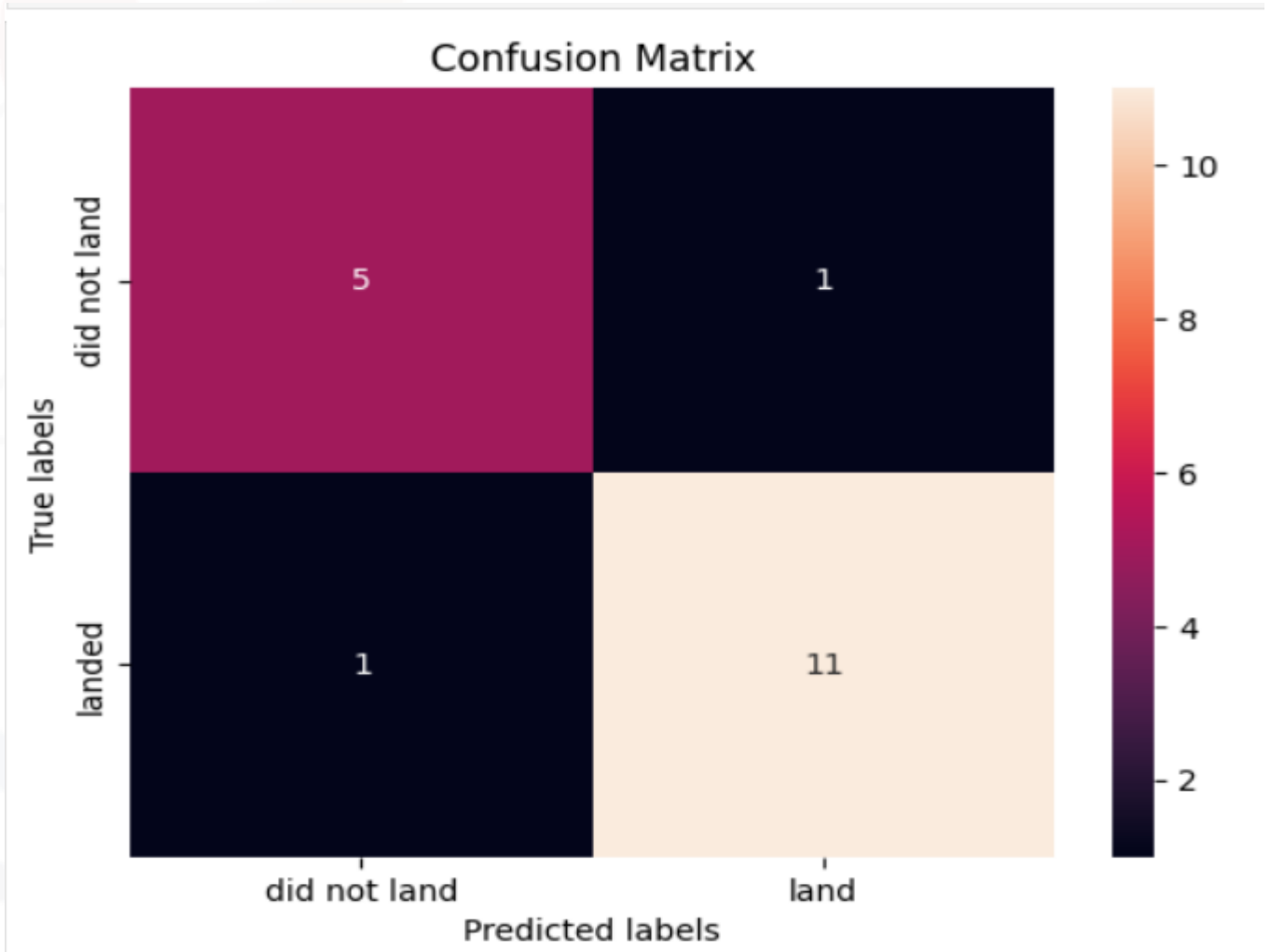
- Score method - Accuracy: 0.8333333333333334



Decision Tree Confusion Matrix

- decision tree classifier object with a GridSearchCV object tree_cv with cv = 10

- Score method - Accuracy: 0.8888888888888888



0.8888888888888888

K Nearest Neighbors Confusion Matrix

- k nearest neighbors object with a GridSearchCV object knn_cv with cv = 10

- Score method - Accuracy: 0.8333333333333334