National College of Ireland

# Identification and Classification of Exoplanets using Light Intensity

MSc Research Project
Data Analytics


Aaditya Balkrishna Garude
Student ID: x20208596

School of Computing
National College of Ireland

Supervisor: Dr. Christian Horn

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Aaditya Balkrishna Garude |
| **Student ID:** | x20208596 |
| **Programme:** | Data Analytics |
| **Year:** | 2022 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Dr. Christian Horn |
| **Submission Due Date:** | 15/08/2022 |
| **Project Title:** | Configuration Manual |
| **Word Count:** | 469 |
| **Page Count:** | 10 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | Aaditya Balkrishna Garude |
| **Date:** | 15th August 2022 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration File.

Aaditya Balkrishna Garude
x20208596
Msc. in Data Analytics

## 1 Introduction

The research was implemented using ample of analysis and several experiments which were concluded on the basis of various try and error attempts.To provide details of each and every step taken for the execution of the research this configuration manual is created which contains detail information about each and every steps performed in the research right from the selection of data to the implementation of models and also knowledge about the specification of Hardware and software used in the project.

## 2 Specification of system and Requirements

The specification of the system is divided into two sections Hardware and Software.

### 2.1 Hardware Specification

Following is the Hardware configuration of the system used for the implementation of the research project.

| Components | Specification |
|---|---|
| Hardware | Dell inspiron 15 3000 |
| Processor | AMD Ryzen 5 3500U |
| RAM | 16.0 GB |
| System Type | 64-bit operating system, x64-based processor |
| Graphics | Radeon Vega Mobile Gfx |

Fig.1.Hardware Specification

## 2.2 Software Specification

The software used in the system plays a vital role in the execution of the research project as the type of advance software installed can increase the speed and produce desirable outputs.
The operating systems has Microsoft Windows 11 and the implementation of program is carried in Jupyter Notebook 6.4.12 version.Following is the Table showing the libraries installed and their versions.

| Libraries | Version |
|---|---|
| Python | 3.8.8 |
| Tensorflow | 2.7.0 |
| Sci-kit Learn | 0.24.2 |
| Seaborn | 0.11.0 |
| Imblearn | 0.8.0 |
| Pandas | 1.1.3 |
| Numpy | 1.19.2 |
| Matplotlib | 3.3.2 |

Fig.2.Software Specifications

# 3 Data Preparation

The Dataset used in the research was gathered from the opene source online Platform called "Kaggle".Kaggle offers a important and helpful platform for Machine Learning and Deep Learning projects having multiple sets of Data for research and practice.
Creation of Kaggle account is must for accessing datasets.

1. Click on Sign in and Complete the registration process.
2. On creating the account the required dataset needs to be accessed.
3. Enter the name and this will redirect you to the page shown in Fig.3
4. Download the Data directly from the Kaggle and upload it in Jupyter notebook.
5. The data is then can been loaded in as shown in following Fig. 4.

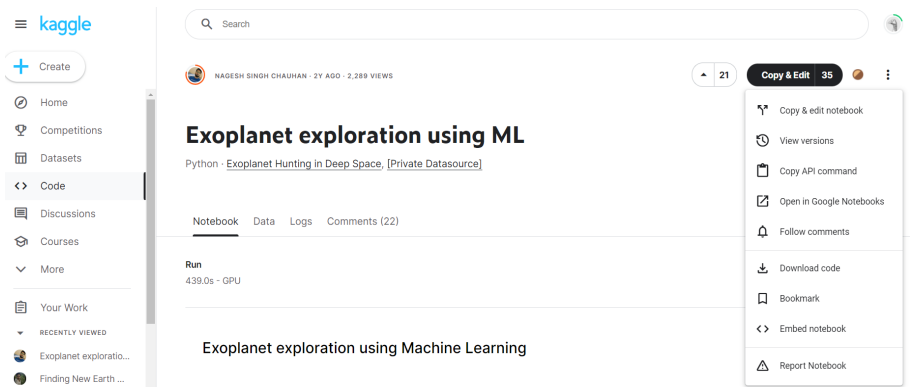


Fig.3.Data Gathering

```
In [1]: #installing all required libraries.
        import os
        import warnings
        import math
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        from pylab import rcParams
        from sklearn.metrics import mean_squared_error, mean_absolute_error
        from imblearn.over_sampling import SMOTE
        from sklearn.model_selection import train_test_split
        from sklearn import linear_model
        from sklearn.metrics import recall_score, precision_score, classification_report,accuracy_score,confusion_matrix, roc_curve, auc,
        from sklearn.preprocessing import StandardScaler, normalize
        from scipy import ndimage
        from keras.utils import np_utils
        from sklearn.metrics import classification_report
        from sklearn.metrics import plot_confusion_matrix
        from keras.utils import np_utils

In [2]: #Loading the train and test data

        exo_test = pd.read_csv('PlanetTest.csv')
        exo_train= pd.read_csv('PlanetTrain.csv')
```

Fig.4.Importing the libraries and loading the train and test data

# 4 Experimental Setup

The original Dataset is imbalance in nature. As seen in Fig.5.The Data contains 90% non-exoplanet and 10% of exoplanet.

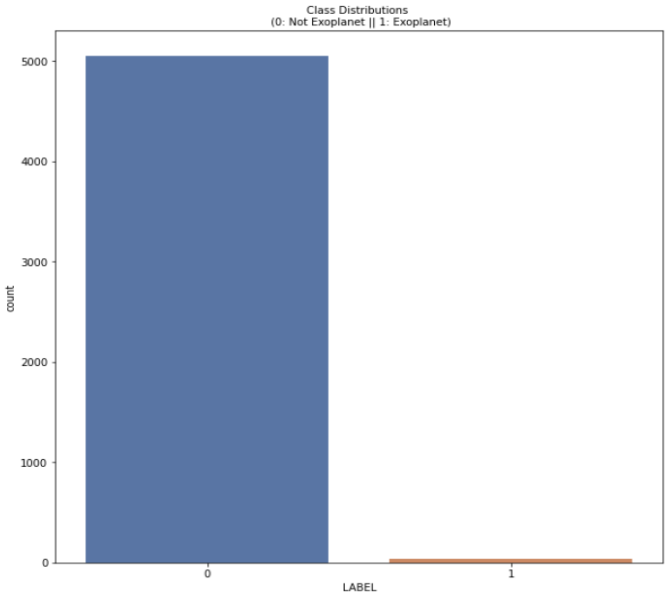


Fig.5.Imbalanced Dataset

**Exploratory Data Analysis**

```
#Plotting the correlation matrix
plt.figure(figsize=(10,10))
sns.heatmap(exo_train.corr())
plt.title('Correlation in the data')
plt.show()
```

Fig.6.Correlation Matrix

```
#Plotting gaussian histogram for non-exoplanets.
labels_1=[200,400,600]
for i in labels_1:
    plt.figure(figsize=(3,3))
    plt.hist(exo_train.iloc[i,:], bins=200)
    plt.title("Gaussian Histogram")
    plt.xlabel("Flux values")
    plt.show()
```

Fig.7. Execution of Gaussian Non-exoplanet

```
#plotting gaussian histogram for exoplanets.
labels_1=[15,30,45]
for i in labels_1:
    plt.figure(figsize=(3,3))
    plt.hist(exo_test.iloc[i,:], bins=200)
    plt.title("Gaussian Histogram")
    plt.xlabel("Flux values")
    plt.show()
```

Fig.8.Execution of Gaussian Exoplanet

```
#Scatterplot.Plotting scatterplot for relationship between two columns
sns.scatterplot(data=exo_train, x='FLUX.1', y='FLUX.6', hue='LABEL', palette=['b','r'])
plt.title('Relation of FLUX1 and FLUX6')
plt.show()
```

Fig.9.Execution of Scatter plot

```
#Plotting of pairplot for random intensities of 5 columns.
print('Pairplot for random 5 intensities')
sns.pairplot(data=exo_train[['LABEL','FLUX.1','FLUX.2','FLUX.3','FLUX.4','FLUX.5']], hue='LABEL')
plt.show()
```

Fig.10.Execution of Pairplot

```
#Detecting outliers using boxplot for 3 columns FLUX1, FLUX2, FLux3
fig, axes = plt.subplots(1, 3,figsize=(15, 6), sharey=True)
fig.suptitle('Distribution of FLUX')

sns.boxplot(ax=axes[0], data=exo_train, x='LABEL', y='FLUX.1',palette="Set2")
sns.boxplot(ax=axes[1], data=exo_train, x='LABEL', y='FLUX.2',palette="Set2")
sns.boxplot(ax=axes[2], data=exo_train, x='LABEL', y='FLUX.3',palette="Set2")
```

Fig.11.Detecting Outliers

**Data Normalization and Standardization**

```
#Data Normalization
x_train = normalized = normalize(x_train)
x_test = normalize(x_test)
```

```
#standardization of the Data for consistent values.
std_scaler = StandardScaler()
x_train = scaled = std_scaler.fit_transform(x_train)
x_test = std_scaler.fit_transform(x_test)
```

Fig.12.Data Normalization and Standardization

## 4.1 Experiment 1

Applying Machine Learning Algorithms for predicting the performance of the applied model without SMOTE technique.

### 1.Naive Bayes:

```
#Naive Bayes Without SMOTE Technique.

from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train,y_train)
prediction=classifier.predict(x_test)
print('Accuracy of Naive Bayes is', accuracy_score(prediction,y_test))
print ("\nClassification report :\n",(classification_report(y_test,prediction)))

#Confusion matrix
plt.figure(figsize=(13,10))
plt.subplot(221)
sns.heatmap(confusion_matrix(y_test,prediction),annot=True,cmap="viridis",fmt = "d",linecolor="k",linewidths=3)
plt.title("CONFUSION MATRIX",fontsize=20)

#ROC curve and Area under the curve plotting
predicting_probabilites = classifier.predict_proba(x_test)[:,1]
fpr,tpr,thresholds = roc_curve(y_test,predicting_probabilites)
plt.subplot(222)
plt.plot(fpr,tpr,label = ("Area_under the curve :",auc(fpr,tpr)),color = "green")
plt.plot([1,0],[1,0],linestyle = "dashed",color ="k")
plt.legend(loc = "best")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title("ROC - CURVE & AREA UNDER CURVE",fontsize=20)
```

```
Accuracy of Naive Bayes is 0.9912280701754386

Classification report :
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       565
           1       0.00      0.00      0.00         5

    accuracy                           0.99       570
   macro avg       0.50      0.50      0.50       570
weighted avg       0.98      0.99      0.99       570
```

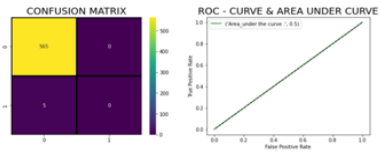


Fig.13.Naive Bayes model without SMOTE

.

### 2.Logistic Regression:

```
#Logistic Regression Without SMOTE Technique

from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(class_weight={0:100, 1:1})

lr_model.fit(x_train,y_train)

prediction=lr_model.predict(x_test)

print('Accuracy of Logistic Regression is', accuracy_score(prediction,y_test))
print ("\nClassification report :\n",(classification_report(y_test,prediction)))

#Confusion matrix
plt.figure(figsize=(13,10))
plt.subplot(221)
sns.heatmap(confusion_matrix(y_test,prediction),annot=True,cmap="viridis",fmt = "d",linecolor="k",linewidths=3)
plt.title("CONFUSION MATRIX",fontsize=20)

#ROC curve and Area under the curve plotting
predicting_probabilites = lr_model.predict_proba(x_test)[:,1]
fpr,tpr,thresholds = roc_curve(y_test,predicting_probabilites)
plt.subplot(222)
plt.plot(fpr,tpr,label = ("Area_under the curve :",auc(fpr,tpr)),color = "green")
plt.plot([1,0],[1,0],linestyle = "dashed",color ="k")
plt.legend(loc = "best")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title("ROC - CURVE & AREA UNDER CURVE",fontsize=20)
```

```
Accuracy of Logistic Regression is 0.9859649122807017

Classification report :
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       565
           1       0.00      0.00      0.00         5

    accuracy                           0.99       570
   macro avg       0.50      0.50      0.50       570
weighted avg       0.98      0.99      0.98       570
```

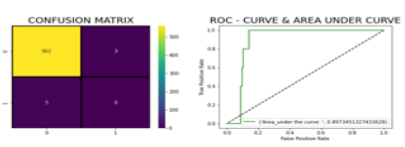


Fig.14.Logistic Regressor model without SMOTE

.

### 3.Decision Tree:

```
#Decision Tree Without SMOTE Technique
from sklearn.tree import DecisionTreeClassifier
ds_model = DecisionTreeClassifier(max_depth=5, random_state=13)

ds_model.fit(x_train,y_train)

prediction=ds_model.predict(x_test)

print('Validation accuracy of Decision Tree is', accuracy_score(prediction,y_test))
print ("\nClassification report :\n",(classification_report(y_test,prediction)))

#Confusion matrix
plt.figure(figsize=(13,10))
plt.subplot(221)
sns.heatmap(confusion_matrix(y_test,prediction),annot=True,cmap="viridis",fmt = "d",linecolor="k",linewidths=3)
plt.title("CONFUSION MATRIX",fontsize=20)

#ROC curve and Area under the curve plotting
predicting_probabilites = ds_model.predict_proba(x_test)[:,1]
fpr,tpr,thresholds = roc_curve(y_test,predicting_probabilites)
plt.subplot(222)
plt.plot(fpr,tpr,label = ("Area_under the curve :",auc(fpr,tpr)),color = "green")
plt.plot([1,0],[1,0],linestyle = "dashed",color ="k")
plt.legend(loc = "best")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC - CURVE & AREA UNDER CURVE",fontsize=20)
```

```
Validation accuracy of Decision Tree is 0.9614035087719298

Classification report :
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       565
           1       0.00      0.00      0.00         5

    accuracy                           0.96       570
   macro avg       0.50      0.48      0.49       570
weighted avg       0.98      0.96      0.97       570
```

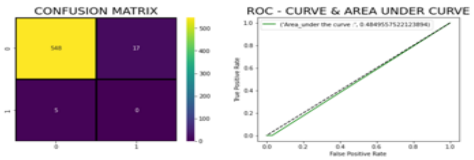


Fig.15.Logistic Regression model without SMOTE

.

## 4.2 Experiment 2

Balancing of Data using SMOTE technique.The results produced as seen in Fig.9 the data for Non-exoplanet and Exo-planet is balance in nature which will produce desired outputs.

```
#SMOTE techinque for Balancing of the imbalanced Data.
from imblearn.over_sampling import SMOTE
model = SMOTE()
ov_train_x,ov_train_y = model.fit_resample(exo_train.drop('LABEL',axis=1), exo_train['LABEL'])
ov_train_y = ov_train_y.astype('int')
ov_train_y.value_counts().reset_index().plot(kind='bar', x='index', y='LABEL',color='orange')
```

```
<AxesSubplot:xlabel='index'>
```

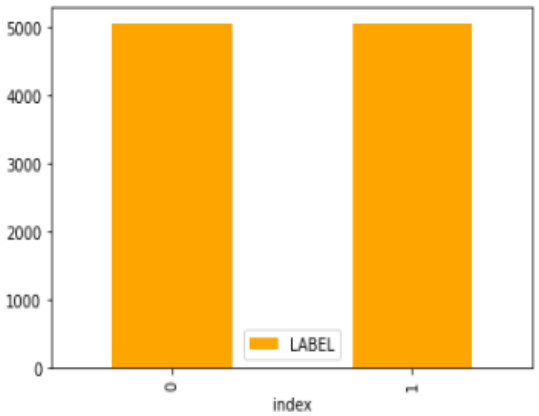


Fig.16.Balanced Dataset

. **Results produced after SMOTE Technique.**

## 1.Naive Bayes:

```
Accuracy of Naive Bayes is 0.4884488448844885

Classification report :
              precision    recall  f1-score   support

           0       0.52      0.03      0.06      1709
           1       0.49      0.97      0.65      1624

    accuracy                           0.49      3333
   macro avg       0.50      0.50      0.36      3333
weighted avg       0.50      0.49      0.35      3333

Text(0.5, 1.0, 'ROC - CURVE & AREA UNDER CURVE')
```

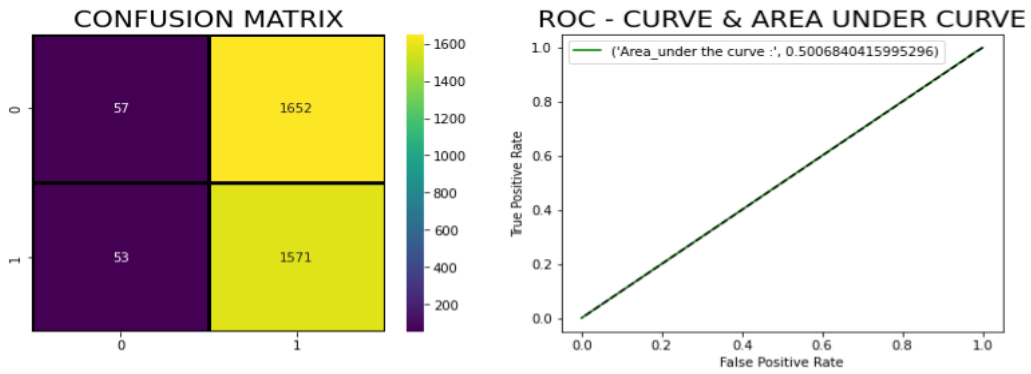


Fig.17.Naive Bayes model without SMOTE

.

## 2.Logistic Regression:

```
Accuracy of Logistic Regression is 0.6951695169516952

Classification report :
              precision    recall  f1-score   support

           0       0.72      0.66      0.69      1709
           1       0.67      0.73      0.70      1624

    accuracy                           0.70      3333
   macro avg       0.70      0.70      0.70      3333
weighted avg       0.70      0.70      0.69      3333

Text(0.5, 1.0, 'ROC - CURVE & AREA UNDER CURVE')
```

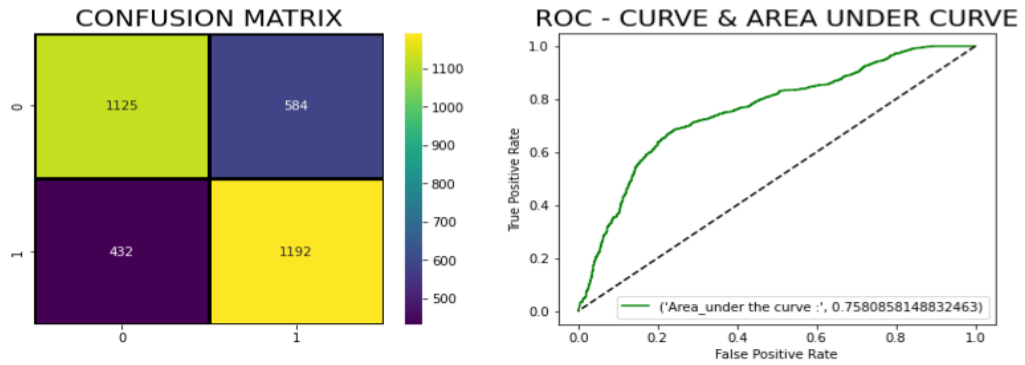


Fig.18.Logistic Regression model with SMOTE

.

## 3.Decision Tree:

```
Validation accuracy of Decision Tree is 0.9153915391539154

Classification report :
              precision    recall  f1-score   support

           0       0.99      0.85      0.91      1709
           1       0.86      0.99      0.92      1624

    accuracy                           0.92      3333
   macro avg       0.92      0.92      0.92      3333
weighted avg       0.92      0.92      0.92      3333

Text(0.5, 1.0, 'ROC - CURVE & AREA UNDER CURVE')
```

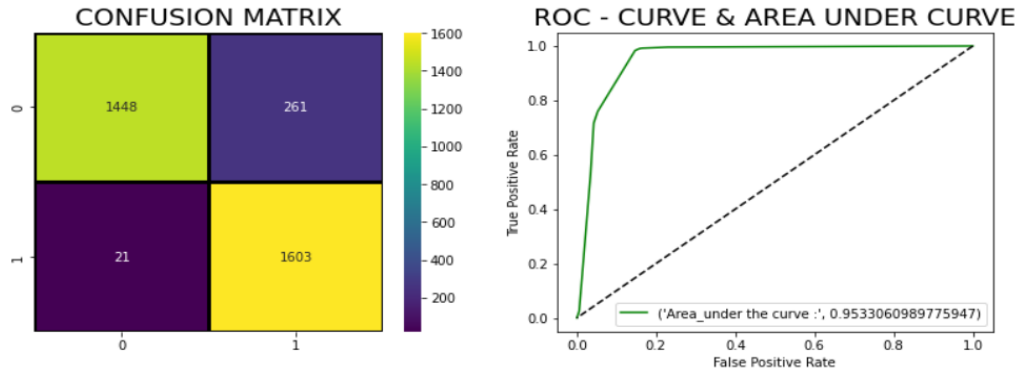


Fig.19.Decision Tree model with SMOTE

.

## References

Malik, A., Moster, B.P. and Obermeier, C. (2011). Exoplanet detection using machine learning. Monthly Notices of the Royal Astronomical Society.

McCauliff, S.D., Jenkins, J.M., Catanzarite, J., Burke, C.J., Coughlin, J.L., Twicken, J.D., Tenenbaum, P., Seader, S., Li, J. and Cote, M. (2015). AUTOMATIC CLASSIFICATION OFKEPLERPLANETARY TRANSIT CANDIDATES. The Astrophysical Journal, 806(1), p.6.

Santos, L.A. dos, Bourrier, V., Ehrenreich, D. and Kameda, S. (2019). Observability of hydrogen-rich exospheres in Earth-like exoplanets. Astronomy Astrophysics, [online] 622, p.A46. doi:10.1051/0004-6361/201833392.

Tallo, T.E. and Musdholifah, A. (2018). The Implementation of Genetic Algorithm in Smote (Synthetic Minority Oversampling Technique) for Handling Imbalanced Dataset Problem. [online] IEEE Xplore. doi:10.1109/ICSTC.201