



Nov 01, 2021

Depression Detection Algorithm

Umme Marzia Haque¹

¹American International University - Bangladesh

1



dx.doi.org/10.17504/protocols.io.bzm6p49e

Umme Marzia Haque

American International University - Bangladesh, University o...

The study has used data from YMM. The Yes/No variables that had a low correlation with target variable have been removed. To extract the most relevant features , the high correlated variables with the target variable , the Boruta method was used in conjunction with a Random Forest(RF) Classifier. To select suitable supervised learning models, the Tree-based Pipeline Optimization Tool To select suitable supervised learning models, the Tree-based Pipeline Optimization Tool (TPOTclassifier) has been employed. RF, XGBoost (XGB), Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB) have been employed in the depression identification step.has been employed. RF, XGBoost (XGB), Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB) were employed in the depression identification step.

DOI

dx.doi.org/10.17504/protocols.io.bzm6p49e

Umme Marzia Haque 2021. Depression Detection Algorithm. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.bzm6p49e>



document ,

Nov 01, 2021

Nov 01, 2021

54686

The study has used data from YMM. The Yes/No variables that had a low correlation with target variable have been removed. To extract the most relevant features , the high correlated variables with the target variable , the Boruta method was used in conjunction with a Random Forest(RF) Classifier. To select suitable supervised learning models, the Tree-based Pipeline Optimization Tool To select suitable supervised learning models, the Tree-based Pipeline Optimization Tool (TPOTclassifier) has been

employed. RF, XGBoost (XGB), Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB) have been employed in the depression identification step. RF, XGBoost (XGB), Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB) were employed in the depression identification step.

```
Start;
Read dataset;
Take the columns those contains Yes and No;
Preprocess the value with LabelEncoder;
Set the target variable;
Drop the target variable from training dataset;
Do Pearson Correlation with target variable
Set the range of high correlated values;
if(correlated variable has High correlation?) {
if(orrelated variable is in range?);
{
Keep the variable in the best subset of input features;

while(!orrelated variable is in range?)
Remove the variable;
}
while(!correlated variable has High correlation?)
Remove the variable;
}

X= High correlated variables;
call scaler.transform() on X;
y=target variable;
Set X_train, X_test, y_train, y_test;

Define Random Forest classifier to be used by Boruta;
call fit() to find all relevant features;
Review feature feature names, ranks, and decisions;
Use the subset of features to fit Random Forest model on training data;
Call feature selector.transform() to make sure same features are selected from test data
Print overall accuracy;
Print confusion matrix;

Define TPOT classifier;
call fit() to find internal_cv_score with class weight and threshold of 0.79;

Define Supervised Learning Model according to internal_cv_score;
call fit() to find all relevant features with class weight and threshold of 0.79;
Use the subset of features to fit the model on training data;
Call feature selector.transform() to make sure same features are selected from test data;
Print overall accuracy;
Print confusion matrix;
End
```

