**DAB304-Healthcare Analytics**

**Forecasting Depression Using Machine Learning Algorithms**

1. **INTRODUCTION**

Depression is a mental state of feeling low and withdrawal from doing any activity. It affects the way of thinking, perception and focus of a person. The appetite and number of sleep hours also get influenced with depression. They would either eat less or more on a case-to-case basis. The sleeping hours gets highly affected in this cycle. Often it is found that a person with depression is either experiencing sleep deprivation or is oversleeping. Feeling of despair and misery overcomes all other feelings for such persons. Sometimes such negative mindset leads to suicide. Duration of such experiences can be short or long term depending upon the recovery rate of such patients. Several factors like death of loved ones, accidents, personality disorder, side affects of certain medications, substance abuse like drugs or alcohol abuse and non-psychiatric illnesses like diabetes, cancer, Parkinson’s disease, stroke, etc. are responsible for triggering depression among people. According to WHO, more than 264 million people of all ages suffer from depression globally. As there is hardly any diagnosis to detect depression, it is important to understand the symptoms and approach the doctor at the right time. Early detection can help in better treatment as well as optimum utilisation of healthcare facilities and hence it is important to take the mental health seriously.

In this study of ‘Forecasting Depression Using Machine Learning Algorithms’ we have attempted to understand the impact of financial as well as personal factors on the mental health of any person. The dataset has 23 columns including the target variable ‘Depression’. Total number of observations are 1429. We have taken into consideration factors/variables like age, gender, marital status, number of children, education, asset value, income, expenses, investments which are the possible attributes causing depression. For example, low income can be a potential cause of depression and vice versa. Similarly, education can also influence the perspective of any individual leading to positive or negative thought process. The computational models developed so far, are lacking in efficient prediction due to which this goal could never be accomplished. So, the focus of the analysis was to overcome such challenges and to develop an effective model which could efficiently address this problem. The objective of our study was to forecast the tendency of developing depression using Logistic Regression, Random Forest and K-Nearest Neighbour machine learning models based on factors like age, gender, family, income status, demographics, etc. Thereafter we have evaluated the models by using Confusion Matrix which prints the Accuracy, Precision, F1 score, ROC and Recall score for model performance. Finally we conclude by selecting the best model with the highest accuracy score to predict development of depression in target patients.

1. **RELATED WORK**

We have explored two more research papers whose area of study is forecasting Post Traumatic Stress Disorder (PTSD) based on similar factors as is our study of forecasting depression using machine learning. It is these two research papers from which we have gathered ideas to proceed with our findings.

Identifying probable Post-Traumatic Stress Disorder (PTSD): Applying supervised machine learning to data from a UK military cohort

SUMMARY:

This paper aims at identifying PTSD among United Kingdom Armed Forces using supervised machine learning techniques. The study used the King’s Centre for Military Health Research (KCMHR) cohort. Participants to this study were asked to complete the PTSD Checklist Civilian Version (PCL-C). The armed forces personnel were asked to submit questionnaires like General Health Questionnaire and Alcohol Use Disorders Identification Test Score (AUDIT) the response to which were binary as data type. Support Vector Machines (SVM), Random Forests (RF), Artificial Neural Networks (ANN) and Bagging methods were used as supervised ML classifiers. To normalise the distribution variables were standardised using the K-nearest neighbour algorithm. The missing data proportion was less than 10% across the dataset. Data resampling, variable importance and testing was performed using 100 iterations to consider the variance and to assess the repeatability of the results. Parameter selection done using the 10-fold cross validation method. During each iteration, bootstrap method was employed to stabilise variable selection. Success of each ML was measured in terms of sensitivity, specificity and Matthews Correlation Coefficient (MCC) in all testing interations. Analyses and machine learning was done using Matlab 2016a and Image Processing Toolbox. Performance was varied between RF which provided the maximum accuracy and SVM resulted as the highest sensitivity rate. Overall an accuracy of 0.89 was achieved for all ML classifiers. The study used a dataset which was never used before hence biases in collection of data could greatly impact the ML modelling. Self report measures was used which may had introduced bias. The time of completion of the questionnaire was not considered hence temporal factors could had influenced the findings. Low sensitivity was observed in this study. Differentiation in serving and non serving army personnel was not considered in the modelling. As it is ex-servicemen had the greater probability of PTSD. Hence segregation was essential in this study, but which was not done. As conclusion it is observed that compared to the traditional self report questionnaires, supervised method of ML clearly has more advantages and can perform with great accuracy the prediction of probable PTSD.

Bridging a translational gap: using machine learning to improve the prediction of PTSD

SUMMARY:

For targeted prevention, prediction of PTSD is the foremost criterion. Present level of research has been able to identify risk indicators at group level (e.g head trauma, receiving opiates) which pertains to a subset of survivors. To increase the efficiency of early risk assessment, identifying interchangeable sets of risk indicators may be used. The goal of this study is the use of supervised machine learning (ML) to materialise interchangeable predictive combinations of risk indicators which of highest predictability. This study used data collected for the Jerusalem Trauma Outreach and Prevention Study. Data features which reflects event characteristics, emergency department (ED) records and early symptoms were collected from 957 trauma survivors within ten days of emergency admission. These data were used to predict PTSD symptoms during the course of next fifteen months. Target Information Equivalence Algorithm (TIE) was used which identified all the minimal sets of features which provides the maximum prediction of a non-remitting PTSD symptom. Thereafter from each of these sets, accuracy of the prediction was evaluated using Support Vector Machine (SVM). A repeated 10-fold cross validation was used to evaluate the predictive accuracy of the predictors and was expressed as average area under the Receiver Operating Characteristics curve (AUC) for all validation trials. The average number of Markov Boundary (MB)’s per cross validation was 800. The average number of features per MB was 18 with 13 features present in over 75% of the sets. The findings adhere to the theorised presence of multiple and interchangeable sets of risk indicators that completely and equally predict non-remitting PTSD. The ability of ML to predict with increased accuracy is a strong step towards developing algorithmic, knowledge based, personalised prediction of post traumatic psychopathology.

1. **METHODS**

We have imported the necessary libraries, i.e pandas, numpy, matplotlib, GridSearchCV, seaborn which will help us in our data analysis and visualisations. The dataset ‘b\_depressed’ was downloaded from Kaggle and since its available for free so we didn’t need any consent from the author. This dataset contains twenty-three columns like age, gender, marital status, number of children, education, asset value, income, expenses, investments which are the possible attributes causing depression. Target/dependent variable is ‘depressed’. Total number of observations were 1429. In exploratory data analysis we found that there are 20 null values in the column ‘no\_lasting\_investmen’. Elimination of null values is required to clean our dataset, this improves the data quality and thus increases overall productivity. To find the similarities between the columns ‘no\_lasting\_investmen’ and ‘lasting\_investment’ we have plotted a histogram and found that both the graphs are almost similar, indicating that data points from ‘lasting\_investment’ can replicate the values of the variable ‘no\_lasting\_investment’.

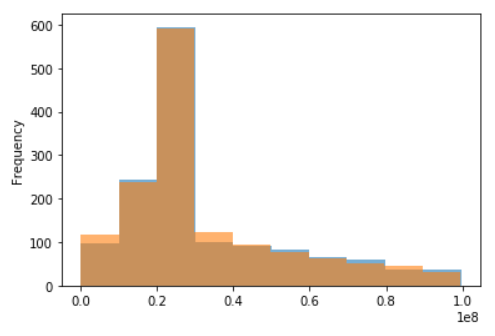


Fig. 3.1: Histogram of two independent variables, ‘no\_lasting\_investmen’ and ‘lasting\_investment’.

Hence, we have dropped the column ‘no\_lasting\_investmen’ and now we have 22 features and 1429 observations free from any null values.

Next, in order to have a glance at the distribution of the dataset we have plotted a histogram to find the frequency of patients experiencing depression and observed that around 50% of the patients was aged between 20-30 years and majorly female patients. This provides the opportunity to research on the mental health condition, its detection and prevention of female patients in the age group of 20-30 years, which may be explored in future endeavours.

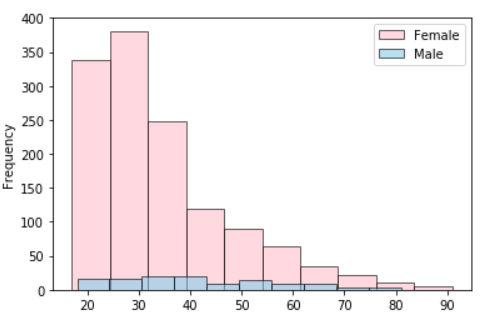


Fig. 3.2: Distribution of male and female patients with depression

Next, we explored on the distribution of number of children for unmarried patients and observed that most of the patients have one child only.

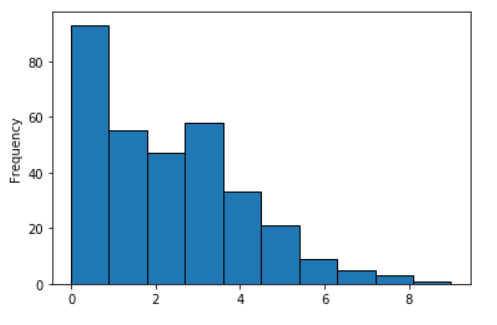


Fig. 3.3: Distribution of number of children for unmarried patients

Further, in order to have a look at the distribution of patients with and without depression, we have plotted a histogram.

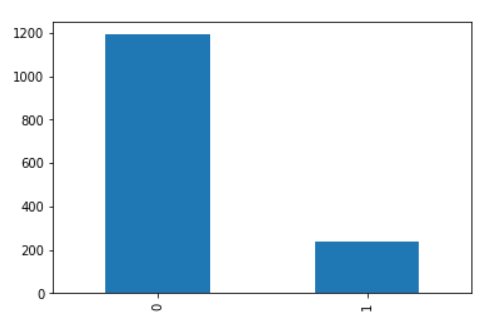


Fig. 3.4: Distribution of patients with depression and without it.

In fig. 3.4, 0 stands for patients without depression and 1 denotes patients experiencing depression. It is observed that only around 20% of the patients have depression. This imbalance of data points may affect the performance of our machine learning algorithms hence we have used Synthetic Minority Oversampling Technique (SMOTE) which will equalise the number of observations in the dataset by oversampling. The oversampling of the minority class is one approach to resolving imbalanced datasets. In the minority class, the easiest solution requires duplicating instances, as these examples do not bring any new details to the model. Instead, new examples from current examples may be synthesised.

Next, we generated a correlation matrix to quantify the influence of independent variables on our target variable ‘depression’ and observed that no single variable is highly correlated with depression but rather it’s a set of variables that are responsible to cause depression.

We have dropped the variables ‘Survey\_id’ and ‘Ville\_id’ from the dataset as these variables do not contribute any value to our machine learning models and set our target/dependent variable ‘depressed’. We have performed feature engineering to select the best input variables. Feature engineering is the process of reducing the number of input variables when a predictive model is created. In order to both minimise the computational expense of modelling and in some cases, to boost the model's efficiency, it is beneficial to reduce the number of input variables. It enables the machine learning algorithm to learn quicker, decreases model sophistication, improved accuracy and reduces overfitting. We have used SelectKBest function as our feature engineering algorithm. SelectKBest function selects the variables which has the highest correlation, positive or negative, with the dependent variable. In our study we have used SelectKBest to find the ten best input variables. They are shown as per the table below:

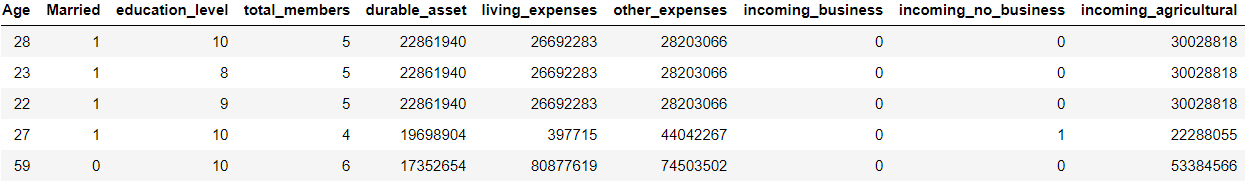


Fig. 3.5: Table showing top 10 features which are highly correlated with the dependent variable.

The last step of our data pre-processing involved using feature scaling as it can be seen that values between the independent variables differs greatly hence standardization was needed so that our machine learning algorithm does not get biased towards data points with high values which leads to incorrect prediction. Hence, we have imported PowerTransformer from sklearn.preprocessing and have scaled the features.

Now, that our dataset has been explored, cleaned and all the pre-processing step has been complete, we can proceed towards model building. Before we have split the data in train and test sets and have fit our models to such data sets, we have imported library log loss, confusion matrix and cross validation to evaluate and validate the model accuracy.. If we change the random state parameter, which is present in train test phase, each time we run our kernel, then the accuracy would be different for each iteration. Hence, consistency of the accuracy score will be lost. To avoid this we have used Repeated StratifiedKFold function which is a cross validation technique. We have used 50 iterations such that each fold will hold roughly the same proportion of samples and the mean response value is just about the same for each fold. We have used a loop function so that repeated stratified K fold function, cross validation score and confusion matrix can be coded for all 50 iterations for each model, in one single loop. In our EDA step, ref. Fig. 3.4, we have observed that there is an imbalance in our dataset, i.e around 50% of the patients are aged between 20 to 30 years and most of that segment are female patients. To eliminate this imbalance, we have used the SMOTE technique as mentioned earlier with sampling strategy being the minority class.

First, we have applied the Logistic Regression model to our cleaned and scaled dataset. Logistic Regression is a supervised learning classification algorithm used to estimate the likelihood of a target variable. The target variable ‘dependent’ is binary in nature hence we can apply this model to our dataset. Logistic Regression technique is typically used for most of the classification prediction problems. GridSearchCV was used for all the models in our study. It is a hyperparameter tuning method that is done to determine the optimum values for a given model. A model's efficiency depends significantly on the importance of hyperparameters. There is no way of determining the right values for hyperparameters in advance, but preferably, to know the optimum values, we need to find all possible values. GridSearchCV is a tool that comes in the model selection bundle for Scikit-learn. This tool helps to loop and fit the model on the training set by predefined hyperparameters. So in the end, from the described hyperparameters, we can pick the best parameters. Next, we have imported the train test split function from sklearn.model\_selection and have split the data keeping test size at 30% and random state = 0.

We have imported LogisticRegression from sklearn.linear\_model. Then we have fit the data to our model and evaluated its performance with confusion matrix. The first confusion matrix that we have used selects the best model from GridSearchCV and plots it without any iterations. Function used was total\_confusion\_matrix(best\_model\_lg, x,y), where lg stands for Logistic Regressor and x and y stands for independent and target variable respectively. Finally, we evaluated our logistic regression model by using Mean Confusion Matrix, which averages the score of accuracy, precision, F1 score, ROC, Recall for all 50 iterations and plots it.

Secondly, we have applied Random Forest (RF) model to our dataset with defined parameters like, number of features, minimum sample leafs, maximum leaf nodes, etc., and have applied GridSearchCV for hyperparameter tuning. Random forest is an algorithm for supervised learning. "The "forest" it constructs is an ensemble of decision trees, usually trained in the method of "bagging. The basic principle of the approach of bagging is that the cumulative outcome is improved by a mixture of learning styles. Simply placed, multiple decision trees are created by random forests and combined together to get a more detailed and reliable forecast. Just as we had evaluated in Logistic Regression model, here too we have evaluated our model by plotting the best confusion matrix, total confusion matrix and the mean confusion matrix. Apart from this we have also plotted a bar chart of the top seven features of the dataset as per their importance.

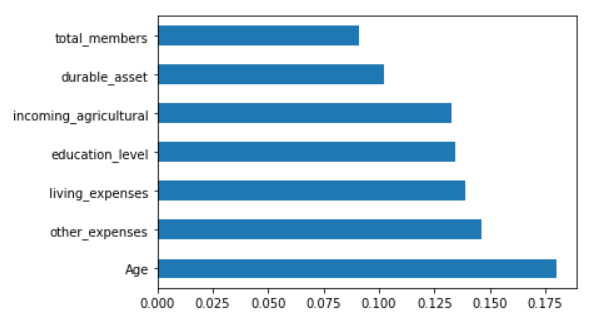


Fig. 3.6: Top seven features with respect to importance as per Random Forest model.

We can see that the variable ‘Age’ has the highest importance as per the model. This basically implies that age is the most important factor on which depression is dependent. The probability of having depression is highly correlated with the age of the patient.

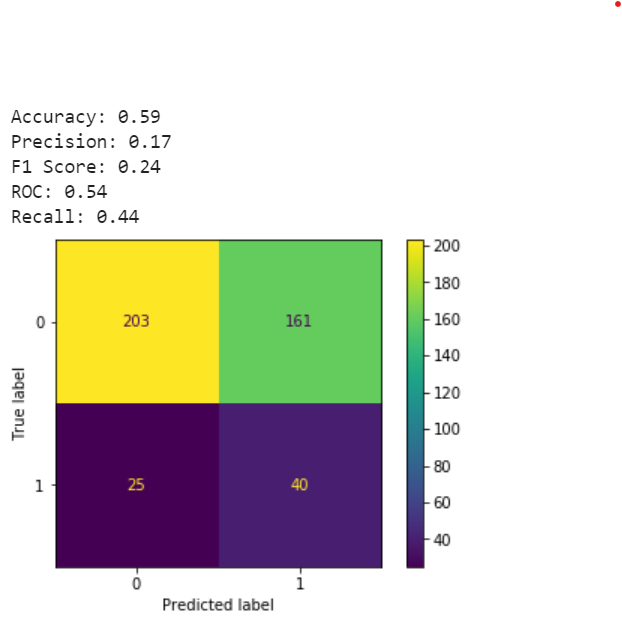
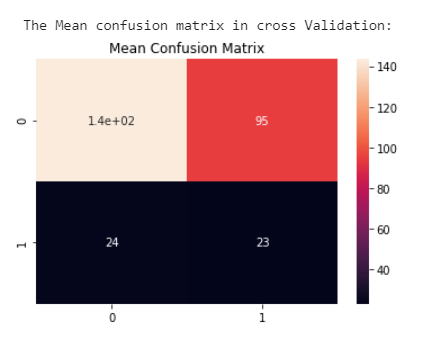
In order to improve the accuracy of our random forest model, we have also used AdaBoostClassifier technique which is one of the ensemble boosting technique used to improve the performance of the model. It assigns weights to the incorrect outcomes from each sequential iterations for any model and then passes on to the next sequence where the model performs its function again until all these weak combinations of classifications has been strengthened which in turn improves the overall accuracy of the model.

Lastly, we perform the K Nearest Neighbour (KNN) classifier. One of the simplest machine learning algorithms based on the Supervised Learning method is K-Nearest Neighbour. KNN operates under a theory that implies that each data point that falls next to each other is of the same class. That means related items are close to each other. The basic version of the algorithms for the K-nearest neighbour classifier is to predict the target label by finding the nearest neighbour class. Using Euclidean distance, the closest class to the point to be labelled is determined. Further, to improve the performance of our KNN model we have used the BaggingClassifier by importing its library from sklearn.ensemble import and fitting it with the KNN classifier.

1. **RESULTS**

The results obtained from applying the mentioned algorithms are discussed in this section.

**Logistic Regression**

When the logistic regression algorithm was applied, the testing accuracy that we achieved for the model is 0.59. The precision, which is basically the percentage of the correctly predicted positive values from the total predicted positives, is 0.17. The recall, which is the value for actual positives is 0.44% and the F1 value which is derived from precision and recall is 0.24. The ROC value is 0.54, which shows the ability of the model to differentiate between the target classes.

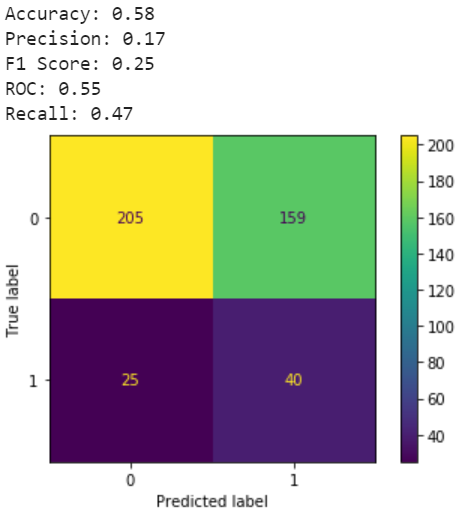
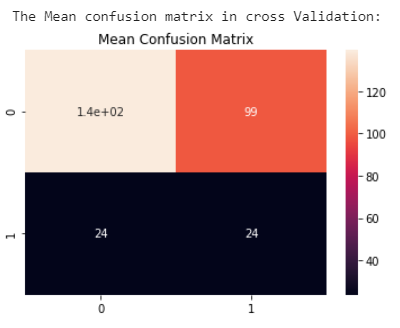
The Best model obtained from GridSearchCV:

LogisticRegression(C=0.1, class\_weight='balanced', random\_state=0, solver='liblinear’)

To improve the model, we applied “RepeatedStratifiedKFold” technique, and the mean values generated from those iterations were shown above in the right.

The above technique also couldn’t improve the accuracy of the model, instead the accuracy values are affected negatively.

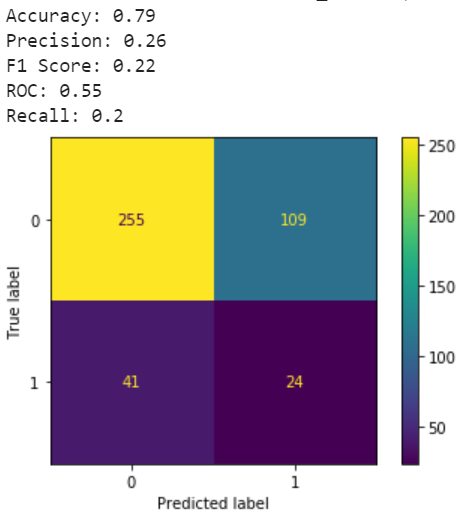
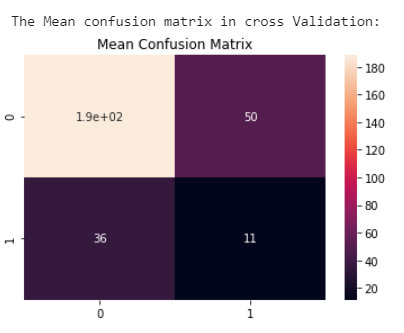
The initial model has given biased results, where a single class was predicted more. To overcome this, we applied AdaBoost technique, which is an iterative ensemble method that tries to build a strong classifier by merging multiple poorly performing classifiers. Due to this high accuracy strong classifier is achieved.

The results could slightly be improved.

**Random Forest**

The random forest, a machine learning technique, based on ensemble method was also used in the analysis to build the model, which has given us the below confusion matrix:

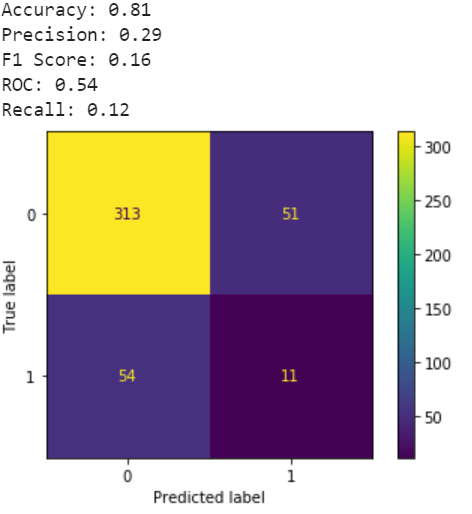
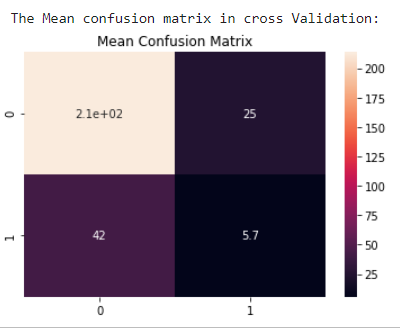
The testing accuracy that we achieved for the model is 0.79, which is quite good. The precision value is 0.26. The value for the recall is 0.20% and the F1 value which is derived from precision and recall is 0.22. The ROC value in random forest is 0.55.

Best model obtained from GridSearchCV:

RandomForestClassifier(bootstrap='True', class\_weight='balanced', max\_depth=25, max\_features='log2', max\_leaf\_nodes=17, min\_samples\_leaf=3, n\_estimators=25, n\_jobs=7, random\_state=0)

The repeated stratified K-fold cross validator was also tried in this technique to check if this could improve the model accuracy but unfortunately this technique couldn’t further improve the results.

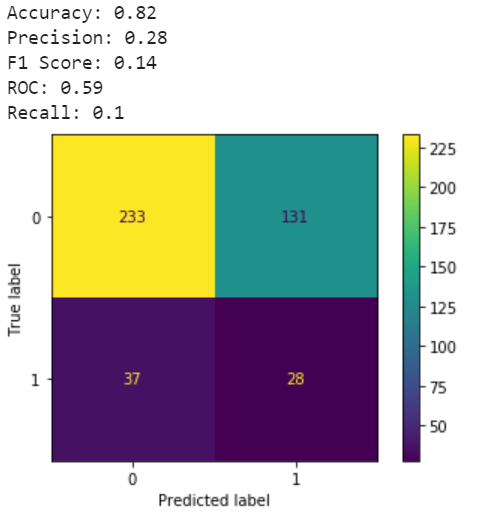
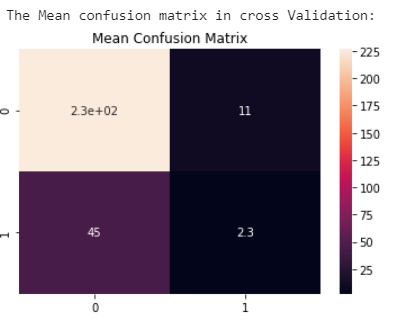
The AdaBoost technique was applied to increase the efficiency of the classifier, which gave us the below results:

It can be seen that the accuracy has improved with the use of AdaBoost. This shows that the model has performed well.

**kNN (k-Nearest Neighbors) Classification**

Lastly, we have also built the model using k-Nearest Neighbors, which is a simple algorithm.

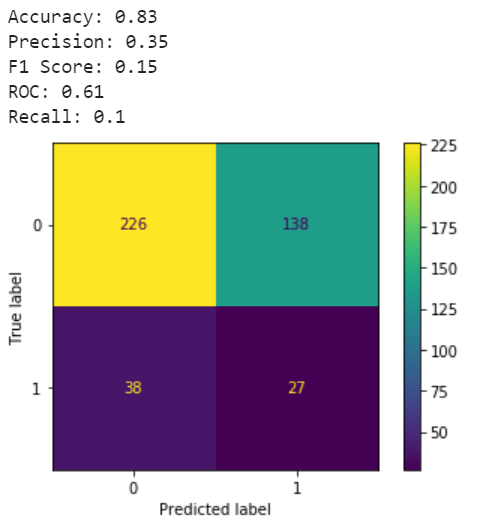
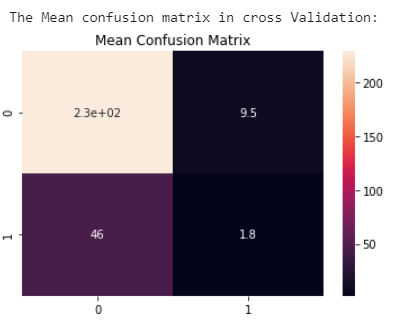
 

Here, the testing accuracy that we achieved for the model is 0.82, which is again quite good. The precision value is 0.28. The value for the recall is 0.10% and the F1 value which is derived from precision and recall is 0.14. The ROC value in random forest is 0.59. The repeated k fold technique has given us the results shown in the mean confusion matrix, where the accuracy has been affected negatively.

Best model obtained from GridSearchCV:

KNeighborsClassifier(n\_neighbors=9, weights='distance’, random\_state=0)

With the kNN algorithm, we applied bagging technique, which can improve the weak classifier’s accuracy of prediction.

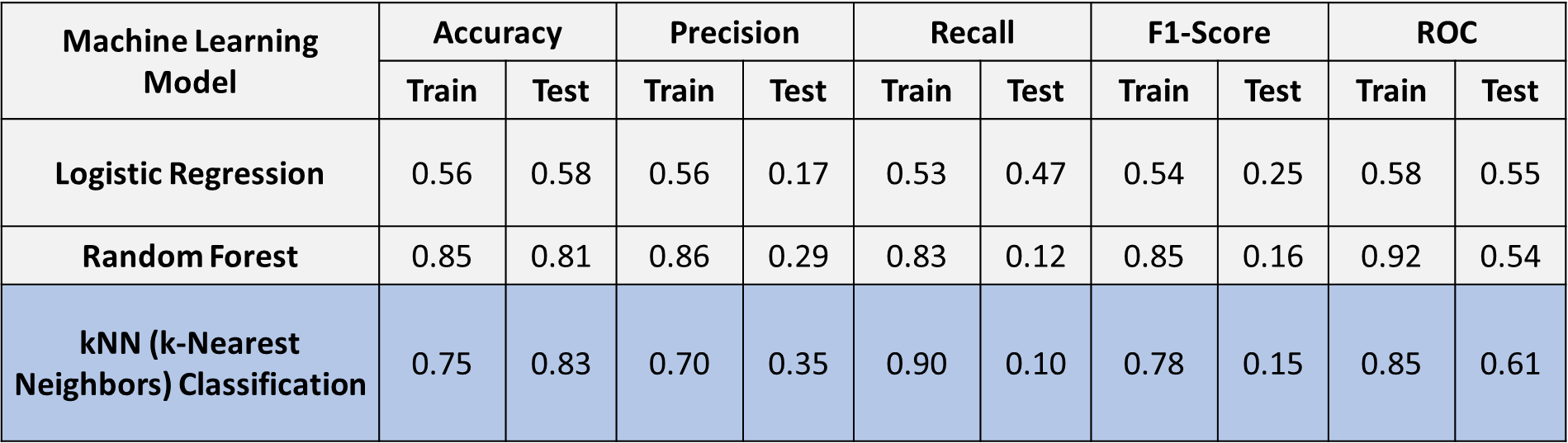
 

It can be seen that the bagging technique has slightly improved the results.

1. **DISCUSSION**

The algorithms that were applied in the analysis, for building the model, have performed differently on training and testing datasets. The AdaBoost and Bagging techniques have definitely improved the outcomes slightly and the model comparison will be done on the values based on these techniques.

Accuracy Comparison of Models



For our analysis on depression detection, kNN (k-Nearest Neighbors) classification model has outperformed the other two algorithms - Logistic Regression and Random Forest Classifier. The kNN classifier has achieved an accuracy of 83% for the testing data, which shows that the model has performed quite well.

In other terms, it can be interpreted as in the application of depression detection, the model framed will predict the possibility of depression with an accuracy of 83% provided the information on variables like age, expenses, education, family members and income, which are the best predicting parameters for this algorithm.

It is to be pointed out that the model has predicted many false positive values, resulting in low precision and recall value, eventually causing low F1-Score. This is unfavourable to the analysis but at the same time, it depends on the data quality. It is already shown through exploratory data analysis that the data size is quite small and the total number of people who have suffered depression is 20% of the whole dataset, which stands as a valid reason for biasedness in prediction for single class i.e. “not depressed”.

To overcome this, the boosting techniques – AdaBoost and Bagging were applied, which helped in improving the performance of the model only up to a certain extent.

In the analysis, there were several challenges, where the utmost important was the initial data search, which was rigorous as statistics on depression were not easily available and/or freely accessible. Also, the dataset has been taken from a website Kaggle.com, which is not a trusted source. Hence, data reliability was a concern.

1. **CONCLUSION**

The focus of the analysis was to understand the impact of financial as well as personal factors on the mental health of a person, which could reduce the probability of depression. Since an early demonstration in this field can provide adequate information so as to identify individuals who are at the risk of obtaining this disorder.

The models obtained from the algorithms, Random Forest and kNN (k-Nearest Neighbors) Classification, could effectively serve the purpose of the study as both the models performed well taking into consideration the data limitation in terms if size and quality. If the real life reliable dataset could be obtained, these built models are ready to serve the purpose of early detection and thereby, the clinical treatment.

For future work, it is suggested to use the deep learning algorithms, which could intensify the model. Also, features relating to the patient behaviour, the social environment and geographical factors, if obtained, can be considered for the study.

1. **CONTRIBUTION**

|  |  |
| --- | --- |
| **Task** | **Member Name** |
| Data Search | All Group Members |
| Data Preprocessing | All Group Members |
| Exploratory Data Analysis | All Group Members |
| Algorithms | Random Forest (RF) – Bishaw  kNN (k-Nearest Neighbors) Classification – Priyank  Logistic Regression – Kriti  K-Fold and Boosting Techniques – Roshan |
| Evaluation | All Group Members |
| Presentation | All Group Members |
| Report | All Group Members |

1. **REFERENCES**

Dataset Link:

<https://www.kaggle.com/diegobabativa/depression>

Previous Studies:

<https://scihub.wikicn.top/10.1080/09638237.2018.1521946>

<https://scihub.wikicn.top/10.1186/s12888-015-0399-8>

Search Links:

<https://www.mayoclinic.org/diseases-conditions/depression/symptoms-causes/syc-20356007>

<https://www.healthline.com/health/depression>

<https://www.psychiatry.org/patients-families/depression/what-is-depression>

<https://www.who.int/news-room/fact-sheets/detail/depression#:~:text=Depression%20is%20a%20common%20mental,affected%20by%20depression%20than%20men>.

<https://deepai.org/machine-learning-glossary-and-terms/random-forest>

<https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html>

<https://www.saedsayad.com/k_nearest_neighbors_reg.htm>

<https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>

<https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>

<https://machinelearningmastery.com/how-to-know-if-your-machine-learning-model-has-good-performance/>

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

<https://en.wikipedia.org/wiki/Depression_(mood)>

<https://www.researchgate.net/publication/327509650_Identifying_probable_Post-Traumatic_Stress_Disorder_Applying_supervised_machine_learning_to_data_from_a_UK_military_cohort/link/5bf69d3892851c6b27d1f29b/download>

<https://scihub.wikicn.top/10.1186/s12888-015-0399-8>

<https://stats.stackexchange.com/questions/253086/selectkbest-feature-selection-python-scikit-learn>

<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

<https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>

<https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/>

<https://medium.com/@venkatasujit272/overview-of-cross-validation-3785d5414ece#:~:text=Repeated%20Stratified%20K%2DFold%20Cross,is%20divided%20into%20k%20subsets>.

<https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/>

<https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm#:~:text=Logistic%20regression%20is%20a%20supervised,probability%20of%20a%20target%20variable.&text=It%20is%20one%20of%20the,Diabetes%20prediction%2C%20cancer%20detection%20etc>.

<https://www.mygreatlearning.com/blog/gridsearchcv/>

<https://www.kaggle.com/prashant111/adaboost-classifier-tutorial>

<https://builtin.com/data-science/random-forest-algorithm>

<https://www.mygreatlearning.com/blog/bagging-boosting/#:~:text=Boosting%20is%20a%20sequential%20ensemble,learner%20to%20a%20strong%20learner.&text=During%20training%2C%20the%20algorithm%20allocates%20weights%20to%20each%20resulting%20model>.

<https://www.analyticssteps.com/blogs/how-does-k-nearest-neighbor-works-machine-learning-classification-problem>

1. **APPENDICES**

Jupyter notebook containing all the codes used for analysis in an orderly format:

final\_project\_HA.ipynb