



University of New Haven

Mental Health Prediction for Working Employees **Using Data Mining**

Submission By:

Vaishnavi Hemant Salaskar
Vedant Vijaykumar Chidgopkar
Riddhi Joshi

vsala2@unh.newhaven.edu
vchid2@unh.newhaven.edu
rjosh5@unh.newhaven.edu

To:

Prof. Shivanjali Khare

ABSTRACT

Mental Health is failed to care for, in the area of health care. Currently, approximately 1 billion people are suffering from a mental health disorder. The numbers are shocking. It is also said that 75% of the people are living without treatments. The number could increase because of the covid-19 pandemic isolating people and adjusting to the new normal, which is having a huge impact on the mental health of the people. The most common problem that people face in the United States is Mental Health. In the United States, one out of five adults suffers from a Mental Health disorder.

The data is collected from an online available dataset called Kaggle. Initially, the dataset will be cleaned before measuring its accuracy. Random Forest, Decision Tree, and Naive Bayes are classification algorithms we plan to implement. With optimization techniques Multiclass Classifier, CV Parameter Selection, and Random Subspace. Our main focus in this analysis is on working individuals.

INTRODUCTION

The state of mind of an individual depends on their mental health. For a better and healthy life, mental health is important. It depends on how an individual reacts, feels, and behaves. Most individuals are prone to stress or depression. It is difficult to evaluate whether an individual is suffering from mental health or not. Very few people have access to mental health services. Some individuals do not have access to mental health service because of their low-middle income. Getting diagnosed with a mental health disorder is vital and essential. Misdiagnosis can cause an imbalance in lifestyle. One of the most difficult tasks is detecting a mental health disorder. So, admitting, realizing, detecting, and treating mental health disorders is important.

The data has been collected by Open Sourcing Mental Illness, LTD from Kaggle. The survey dataset includes mental health attitudes, which then have been analyzed and pre-processed. The data contains different labels such as age, gender, country, self-employee, family history, work interference, seek help, etc. For better prediction, we have label encoded the data.

The analysis mainly focuses on working individuals. It increases mental health awareness in the working space and environments.

RELATED WORK

List of Related Review

1. Design of Data Mining and Evaluation System for College Students Mental Health.

It Published in 2021 by Zhang Xilin and Yi Honglian from Dalian University of Science and Technology, Dalian, China under the title International Conference on Measuring Technology and Mechatronics Automation (ICMTMA).

The author of this research uses an Apriori algorithm to suggest a data mining system for college students' mental health. The student mental health assessment system provided the data for this article. Missing values in mental health data are investigated. The goal is to discover anxiety and depression symptoms in college students, as well as their links to mental health issues. They employed the Apriori algorithm for this. Association rules mining and decision rule mining are used to discover data correlation and concurrence. Furthermore, the system has modification functions that allow for the correction of erroneous data in order to make it more complete, and accurate. 42 percent is the highest quantitative performance result.

2. Classification Algorithms based on Mental Health Prediction using Data Mining.

The articles appeared in 2020 at K J Somaiya Institute of Engineering and Information Technology, located in Mumbai, India, by Vidit Laijawala, Aadesh Aachaliya, Hardik Jatta,

and Vijaya Pinjarkar. This volume contains the proceedings of the Fifth International Conference on Communication and Electronic Systems (ICCES 2020).

Mental health is becoming a more common problem. Here the target population was working individuals that are people above the age of 18. The authors have gathered the dataset from an existing available dataset, which was provided by an OSMI (Open Sourcing Mental Illness) survey. The dataset included the data of working individuals. This dataset consists of 26 attributes for prediction and 1 predicting label. Since the dataset included data from the survey, not all the parameters were useful. It consists of Age, Gender, self_emp, family_history, work_inference, past, diagnosis, treat, etc. The labels were encoded. Most data included 2 or 3 attributes (yes, no, maybe), however, some included 5 attributes. The number of sample records used was 258.

They implemented classification algorithms such as Decision Tree, Random Forest, and Naïve Bayes. Decision Tree and Random Forest algorithms were implemented to check the accuracy. They found that the Decision tree is the most optimal algorithm by 82%. It shows the Confusing Matrix providing the accuracy of 149 instances to be correctly classified as a positive while 109 to be correctly classified as a negative. There are 258 correct classifications out of 315 instances. According to researchers, individuals with a stressful or depressing work-life should seek mental health help. However, individuals whose work life is unaffected does not suffer from mental issues.

3. Machine Learning Techniques for Stress Prediction in Working Employees.

It was published in 2018 by U Shrinivasulu Reddy, Aditya Vivek Thota, A Dharun from the National Institute of Technology, Trichy. The name of the publication is International Conference on Computational Intelligence and Computing Research.

They used the OSMI Mental Health in Tech 2017 survey [2] as a dataset to train machine learning models to assess the patterns of stress and mental health problems among tech professionals and discover the most relevant elements that contribute to these diseases.

The OSMI 2017 dataset's findings were used to train the following machine learning models: b. KNN Classifier: K-Nearest Neighbor (KNN), Decision Trees, Logistic Regression, Decision Trees, Random Forest Classifier, Boosting, and Bagging.

The model was developed using many machine learning techniques, with boosting surpassing the others in terms of precision, accuracy, and false-positive rate. The random forest classifier, on the other hand, had a higher cross-validated AUC, indicating that it is more stable. People who worked in a tech business, even if their role was not tech-related, were slightly more likely to be stressed, according to our findings. The most accurate and exact classification algorithms were boosting and random forest. Machine Learning techniques for stress and mental health condition prediction show remarkable findings with a 75.13 percent accuracy, which can be further researched.

4. Predicting Depression Levels Using Social Media Posts.

It was published in 2017 by Maryam Mohammed Aldarwish, Hafiz Farooq Ahmed from King Saud University for Health Science, Kingdom of Saudi Arabia. The name of the publication is International Symposium on Autonomous Decentralized Systems.

SNS (Social Network Sites) is an online platform where people may share their interests, feelings, and other information. Researchers have demonstrated that using user-generated context to estimate individual mental health levels is a valid method. The goal was to see if SNS user posts could assist categorize people based on their mental health status [4]. Support Vector Machine (SVM) and Nave Bayes classifiers were used to categorize the UGC. Facebook, LiveJournal, and Twitter were used to compile this data. During the training phase, they classified terms that indicated whether the user was depressed or not. After that, a text is assigned to one of the classes using the Support Vector Machine (SVM) algorithm [4]. The training dataset included a total of 6773 postings in it.

5. Predictive Analysis for Healthcare Domain using Classification Techniques.

It was published in 2017 by Shweta Sharma, Sahil Anand, Anant Kumar Jaiswal from Amity School of Engineering, Uttar Pradesh, India. The name of the publication is International Journal of Linguistics and Computing Research.

This research uses classification algorithms to perform predictive analysis in the healthcare domain. They used the mental healthcare dataset, which came from a 2014 survey on the IT industry's mental health. In the dataset mixer, there are 1259 records of

various questions about mental health and some personal inquiries. The purpose of this study was to raise awareness about mental health issues and to help individuals who are suffering from them. These classification algorithms are included in this study paper: Naive Bayes, J48, and Neural Network. They've also employed Weka as a data classification and prediction tool. The document also defines terms like supervised and unsupervised learning, data training and testing, the percentage split, the use of a training set, and mean.

THE PROPOSED METHOD

Data Mining Techniques

- **Naive Bayes**

A probabilistic classifier, the Naive Bayes classification algorithm. It is based on probability models with high independence assumptions. Often, independence assumptions have no bearing on reality. As a result, they are recognized as naive. Bayes' theorem can be used to create probability models. In a supervised learning situation, you can train the Naive Bayes algorithm depending on the type of the probability model.

- **Decision Tree**

A decision tree is created using the Tree Classification algorithm. The model established can be stated as a collection of decision rules, and decision trees are simple to comprehend and adapt. Even in big databases with different amounts of training samples and a large number of attributes, this technique grows well. The output of Decision Tree Classification is a binary tree-like structure. The target variable is predicted using rules in a Decision Tree model. The Tree Classification algorithm generates a simple explanation of the data's underlying distribution.

- **Random Forest**

An ensemble of decision trees called a random forest consists of an overwhelming number of individual trees. Our model predicts a class based on the most votes from each tree in the random forest.

Optimization Techniques

- **Multiclass Classifier**

In supervised machine learning, multiclass classification is a common challenge.

Given a dataset of m training instances, each of which contains information in the form of multiple features and a label, the problem is to find the best solution. Each label represents a class in which the training example is included. We have a finite number of classes in multiclass categorization. There are n characteristics in each training example.

- **CV Parameter Selection**

The process of selecting the best set of parameters for a model is known as hyperparameter tuning. It is recommended that you search the hyper-parameter space for the optimal cross-validation score estimator. To maximize the hyperparameter space for an estimator, various cross-validation techniques can be applied. Grid Search cross-validation is a prominent strategy for optimizing the hyperparameter space and picking a robust model for various machine learning algorithms.

- **Random Subspace**

Random Subspace Ensemble is a machine learning approach that combines predictions from various decision trees trained on subsets of the training dataset's columns. Randomly altering the columns used to train each contributing member of the ensemble introduces diversity into the ensemble, which can improve performance when compared to utilizing a single decision tree. Other decision tree ensembles include bootstrap aggregation (bagging), which produces trees from diverse samples of rows from the training dataset, and random forest, which incorporates principles from bagging and the random subspace ensemble.

THE EXPERIMENTAL RESULTS & DISCUSSION

Data Mining Techniques used:

1. Naive Bayes:

In performing the classification, results were obtained that indicated 255 Instances are correctly classified, with an accuracy of 78.2209%. Incorrectly Classified Instances are 71 with inaccuracy of 21.7791%. Total Number of Instances 326.

The screenshot shows the Weka Explorer interface with the Naive Bayes classifier selected. The 'Test options' section shows 'Percentage split' at 66%. The 'Classifier output' section displays the following results:

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
Correctly Classified Instances      255      78.2209 %  
Incorrectly Classified Instances    71      21.7791 %  
Kappa statistic                    0.4798  
Mean absolute error                0.2906  
Root mean squared error            0.3849  
Relative absolute error            62.8622 %  
Root relative squared error        80.094 %  
Total Number of Instances          326  
  
=== Detailed Accuracy By Class ===  


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-------|
| Weighted Avg. | 0.952   | 0.517   | 0.764     | 0.952  | 0.848     | 0.517 | 0.826    | 0.854    | Yes   |
|               | 0.483   | 0.048   | 0.851     | 0.483  | 0.616     | 0.517 | 0.826    | 0.800    | No    |

  
=== Confusion Matrix ===  


| a   | b  | <-- classified as |
|-----|----|-------------------|
| 198 | 10 | a = Yes           |
| 61  | 57 | b = No            |


```

The 'Result list' on the left shows the selected model: '21:10:30 - bayes.NaiveBayes'.

It is pretty excellent that we have a Precision Value of 0.764 here. As well, there are 198 examples of True(yes), 61 examples of False(no) and 57 examples of true

negatives. The ROC curve is constructed after classification has been completed by plotting the true positive rate (TPR) against the false positive rate (FTR).

In the margin curve is also calculated with Naive Bayes. The margin is defined as the difference between the probability predicted for a class and the highest probability predicted for another class.

2. Decision Tree:

The results of the classification showed that 326 instances are correctly and incorrectly classified with 257 and 69 instances, respectively. We also find the Mean absolute error which is 0.3297. Alongside Root mean squared error 0.4037.

The screenshot displays the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Percentage split' at 66%. The 'Classifier output' pane contains the following text:

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
Correctly Classified Instances      257           78.8344 %  
Incorrectly Classified Instances    69           21.1656 %  
Kappa statistic                    0.4904  
Mean absolute error                 0.3297  
Root mean squared error             0.4037  
Relative absolute error             71.3338 %  
Root relative squared error         84.0105 %  
Total Number of Instances          326  
  
=== Detailed Accuracy By Class ===  


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PAC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-------|
| Weighted Avg. | 0.966   | 0.525   | 0.764     | 0.966  | 0.854     | 0.537 | 0.760    | 0.794    | Yes   |
|               | 0.475   | 0.034   | 0.889     | 0.475  | 0.619     | 0.537 | 0.760    | 0.661    | No    |

  
=== Confusion Matrix ===  


| a   | b  | <-- classified as |
|-----|----|-------------------|
| 201 | 7  | a = Yes           |
| 62  | 56 | b = No            |

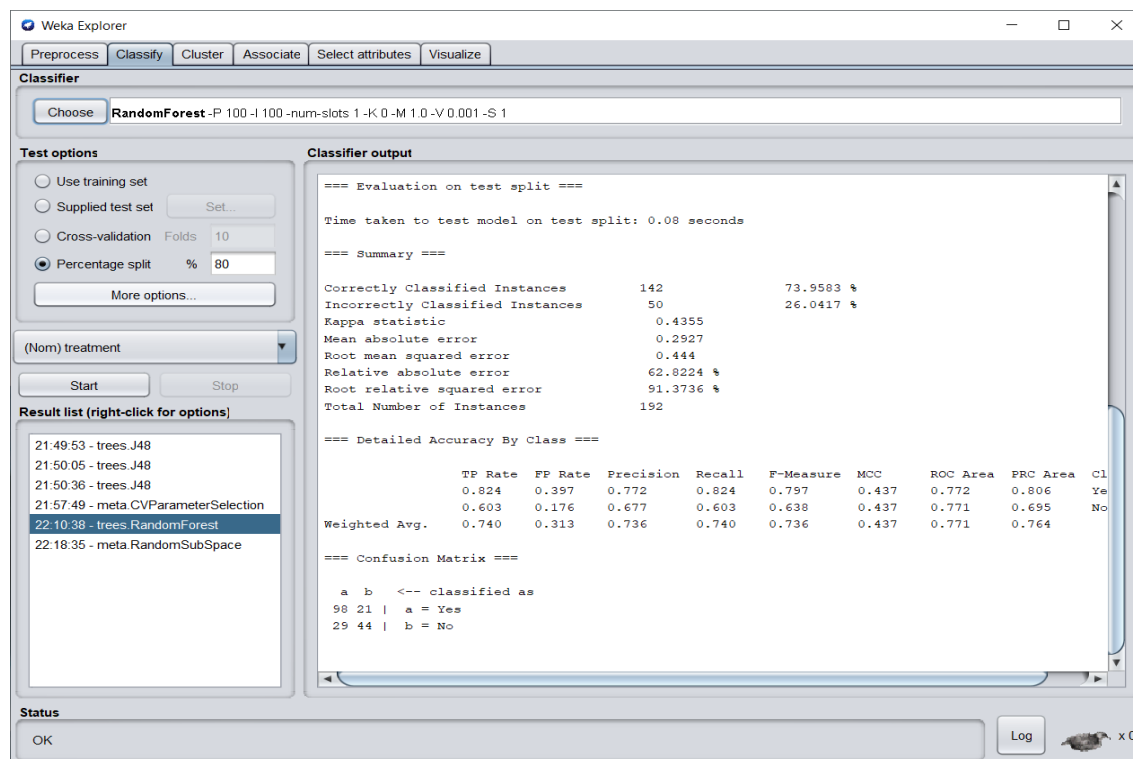

```

The 'Result list' on the left shows a list of classifiers, with '15:07:55 - trees.J48' selected. The 'Status' bar at the bottom indicates 'OK'.

True positives have confusion matrix values of 201 while false negatives have confusion matrix values of 56. The precision value is 0.764. We found out the recall value for the same 0.966. For the decision tree, observations that we found is the work interface is a parameter where data is being split and the weightage of attributes.

3. Random Forest

Our results indicate that 142 instances have been correctly classified with an accuracy of 73.9583 percent. The estimated number of incorrectly classified instances is 50, with an accuracy rate of 26.0417%.



The screenshot shows the Weka Explorer interface with the Random Forest classifier selected. The classifier output window displays the following results:

==== Evaluation on test split ====

Time taken to test model on test split: 0.08 seconds

==== Summary ====

Metric	Value	Percentage
Correctly Classified Instances	142	73.9583 %
Incorrectly Classified Instances	50	26.0417 %
Kappa statistic	0.4355	
Mean absolute error	0.2927	
Root mean squared error	0.444	
Relative absolute error	62.8224 %	
Root relative squared error	91.3736 %	
Total Number of Instances	192	

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.740	0.313	0.736	0.740	0.736	0.437	0.771	0.764	

==== Confusion Matrix ====

```

a b <-- classified as
98 21 | a = Yes
29 44 | b = No

```

The status bar at the bottom shows 'OK' and a 'Log' button.

Similarly, the Precision Value is 0.736. 98 True examples are included, 29 False examples are included, and 44 true negative instances are included.

Optimization Technique used:

1. Multiclass Classifier

Our technique split 80% of the data into training and test set sets. As a result, 79.1667% of the data was correctly classified. A total of 152 instances are correctly classified, while 40 are incorrectly classified. The inaccuracy rate for the incorrectly classified instances is 20.833%.

The screenshot displays the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'MultiClassClassifier -M 0 -R 2.0 -S 1 -W weka.classifiers.bayes.NaiveBayes -output-debug-info'. The 'Test options' section shows 'Percentage split' at 80%. The 'Classifier output' pane displays the following results:

```
=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      152           79.1667 %
Incorrectly Classified Instances    40           20.8333 %
Kappa statistic                    0.5358
Mean absolute error                 0.2951
Root mean squared error            0.3911
Relative absolute error             63.3321 %
Root relative squared error        80.4787 %
Total Number of Instances         192

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cl
               0.908    0.397    0.788     0.908    0.844     0.548    0.833    0.868    Ye
               0.603    0.092    0.800     0.603    0.688     0.548    0.833    0.780    No
Weighted Avg.   0.792    0.281    0.793     0.792    0.784     0.548    0.833    0.834

=== Confusion Matrix ===

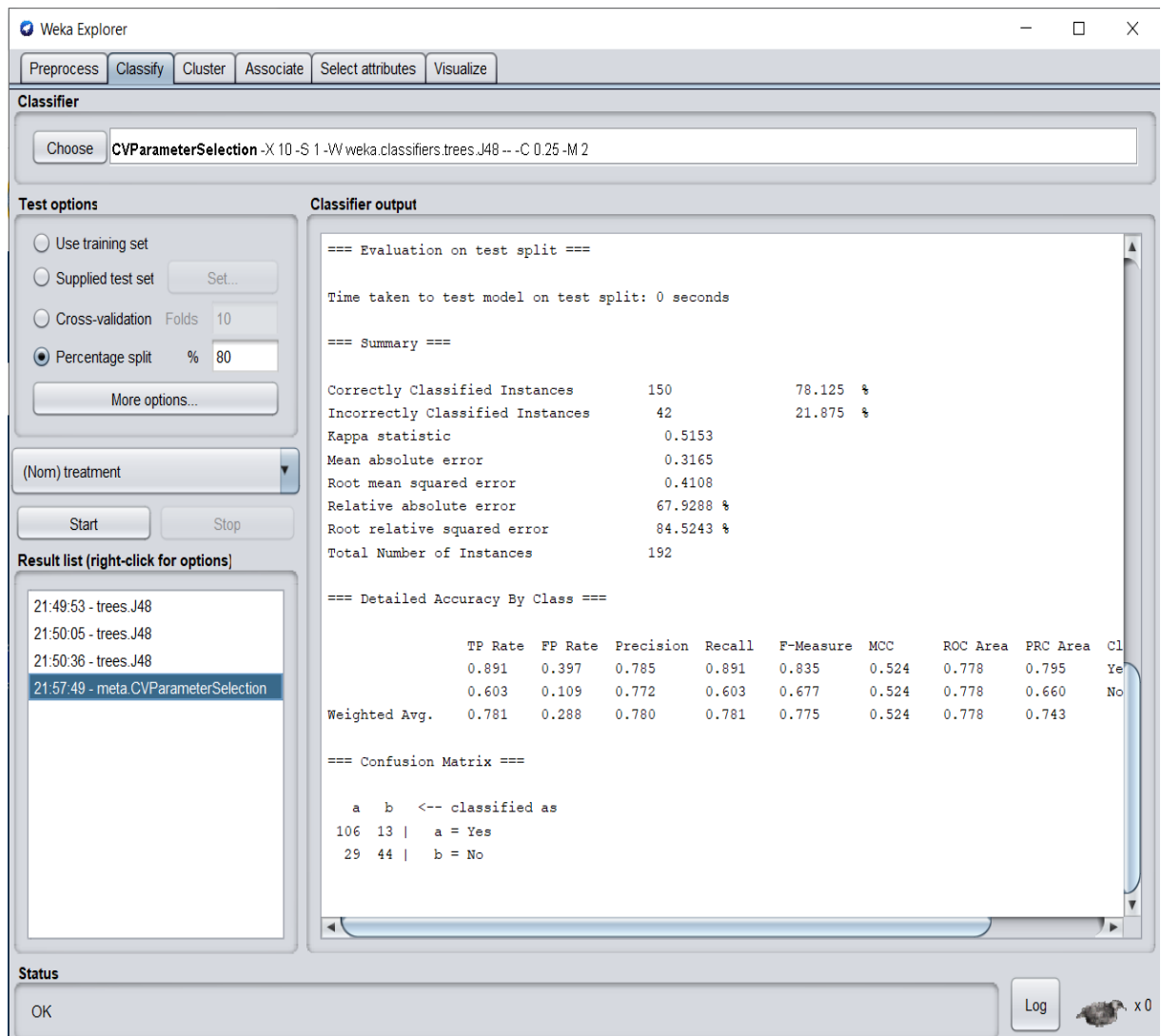
  a  b  <-- classified as
108 11 |  a = Yes
 29 44 |  b = No
```

The 'Result list' on the left shows three entries: '21:20:35 - bayes.NaiveBayes', '21:21:03 - bayes.NaiveBayes', and '21:22:29 - meta.MultiClassClassifier' (highlighted). The 'Status' bar at the bottom shows 'OK'.

Our accuracy rates were improved from 78.22% to 79.166% using Multiclass Classifier, a technique for optimizing Naive Bayes Classification.

2. CV Parameter Selection

The dataset was divided into 80 percent training sets. For correctly classified 150 instances, we obtained an accuracy of 78.125%, while for incorrectly classified 42 instances, we obtained an accuracy of 21.875%.



The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'CVPParameterSelection'. The test options are set to 'Percentage split' at 80%. The classifier output window displays the following results:

```
=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      150      78.125 %
Incorrectly Classified Instances    42      21.875 %
Kappa statistic                    0.5153
Mean absolute error                 0.3165
Root mean squared error             0.4108
Relative absolute error             67.9288 %
Root relative squared error         84.5243 %
Total Number of Instances          192

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Cl
              0.891    0.397    0.785     0.891    0.835     0.524    0.778    0.795    Ye
              0.603    0.109    0.772     0.603    0.677     0.524    0.778    0.660    No
Weighted Avg.   0.781    0.288    0.780     0.781    0.775     0.524    0.778    0.743

=== Confusion Matrix ===

  a  b  <-- classified as
106 13 | a = Yes
 29 44 | b = No
```

The result list on the left shows the selected option: '21:57:49 - meta.CVPParameterSelection'.

The precision and recall values that we obtained by using this method are 0.78 and 0.781, respectively.

3. Random Subspace

We 80 percent of the dataset was considered a training set, which resulted in an accuracy of 75.5208% when 145 instances were correctly classified while 47 incorrectly classified. We found that 24.4792 percent of the occurrences achieved accuracy. This technique provides a precision of 0.754 and a recall of 0.755.

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'RandomSubSpace' with the command: `-P 0.5 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0`.

Test options:

- ☐ Use training set
- ☐ Supplied test set (Set...)
- ☐ Cross-validation (Folds: 10)
- ☒ Percentage split (%: 80)

Classifier output:

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.35 seconds

=== Summary ===

Correctly Classified Instances      145      75.5208 %
Incorrectly Classified Instances    47      24.4792 %
Kappa statistic                    0.45
Mean absolute error                 0.3833
Root mean squared error             0.4216
Relative absolute error             82.2695 %
Root relative squared error         86.7468 %
Total Number of Instances          192

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Cl
          0.891   0.466   0.757    0.891   0.819     0.464    0.792    0.821   Ye
          0.534   0.109   0.750   0.534   0.624     0.464    0.792    0.731   No
Weighted Avg.   0.755   0.330   0.754   0.755   0.745     0.464    0.792    0.787

=== Confusion Matrix ===

  a  b  <-- classified as
106 13 |  a = Yes
 34 39 |  b = No
```

Result list (right-click for options):

- 21:49:53 - trees.J48
- 21:50:05 - trees.J48
- 21:50:36 - trees.J48
- 21:57:49 - meta.CVPParameterSelection
- 22:10:38 - trees.RandomForest
- 22:18:35 - meta.RandomSubSpace

Status: OK

Results:

Classification Techniques	Accuracy %	Precision	Time Taken
Naive Bayes Classifier	78.2209%	0.796	0 sec
Decision Tree	78.8344%	0.809	0 sec
Random Forest	73.9583%	0.736	0.08 sec

Optimization Techniques	Accuracy %	Precision	Time Taken
Multiclass Classifier (Naive Bayes)	79.166%	0.793	0 sec
CV Parameter Selection (Decision Tree)	78.125%	0.780	0 sec
Random Subspace (Random Forest)	75.5208%	0.754	0.35 sec

CONCLUSION

By performing this project, we learned working of different data mining techniques theoretically and practically. For data cleaning and building model, we used Jupyter Notebook and Weka respectively. We achieved accuracy in the range of 75% to 80%. For Naive Bayes, Decision Tree, and Random Forest we recorded accuracies of 78.22%, 78.83%, and 73.95% respectively. Furthermore, to improve model prediction and accuracy we applied optimization techniques. The results we got were impressive, we achieved accuracy of 79.16% for a multiclass classifier. The Naive Bayes Classifier was determined to be the most accurate algorithm based on the accuracy achieved. Because of high accuracy with low execution time.

FUTURE WORK

Our plan for the future is to develop an application based on this model to improve the working environment of an organization. By doing so it will help employees to improve their mental health if there is any. In addition to this our goal is to explore other domains of society such as healthcare, educational, nonprofit and so on, to make a better society. For improving model prediction and accuracy we are exploring the option of including more data as well as increasing the number of attributes as per requirement. Moreover, to achieve more accurate output our idea is to implement neural network mechanisms in model.

REFERENCES

- <https://ieeexplore-ieee-org.unh-proxy01.newhaven.edu/stamp/stamp.jsp?tp=&arnumber=9137856> | Classification Algorithms based Mental Health Prediction using Data Mining.
- <https://ieeexplore-ieee-org.unhproxy01.newhaven.edu/stamp/stamp.jsp?tp=&arnumber=8782395> | Machine Learning Techniques for Stress Prediction in Working Employees.
- <https://ieeexplore-ieee-org.unhproxy01.newhaven.edu/stamp/stamp.jsp?tp=&arnumber=9410139> | Design of Data Mining and Evaluation System for College Students' Mental Health
- https://cphfs.in/myadmin/Submitted_pdf/IJLCR-0008up.pdf | Predictive analysis using classification techniques in healthcare domain.
- **Dataset Reference:** <https://www.kaggle.com/osmi/mental-health-in-tech-survey>
- <https://www.who.int/news/item/27-08-2020-world-mental-health-day-an-opportunity-to-kick-start-a-massive-scale-up-in-investment-in-mental-health#:~:text=Mental%20health%20is%20one%20of,every%2040%20seconds%20by%20suicide>
- Repository: [GitHub - vsala2/DataMining](#)