LEVERAGING MACHINE LEARNING FOR PREDICTING MENTAL HEALTH TREATMENT OUTCOMES: A COMPREHENSIVE STUDY

G Sruthi
sruthig2005@gmail.com
ECE- Course

Abstract:

This study explores the application of machine learning algorithms in predicting mental health treatment outcomes among university students and IT employees. Using a dataset from various outlets, including mental health clinics and online platforms, the study analyzes demographic and behavioral attributes to identify key indicators of mental well-being and treatment-seeking behaviors. The dataset encompasses features such as gender, occupation, family history of mental health issues, and access to care options. Through data preprocessing, exploratory data analysis (EDA), and model building using six classifiers (Logistic Regression, KNN, Random Forest, Decision Tree, Support Vector Machine, and Ensemble), the study aims to provide insights into predicting mental health treatment outcomes and improving support strategies.

1. Introduction:

The use of machine learning algorithms to predict mental health outcomes has garnered significant attention for its potential to illuminate the complex factors influencing mental well-being. This study systematically investigates the efficacy of various machine learning methodologies in forecasting mental well-being and treatment-seeking behaviors among university students and IT employees. The study uses a diverse dataset encompassing demographic and behavioral attributes to pinpoint key indicators associated with mental health outcomes.

Through a rigorous process involving a systematic literature review, data collection from multiple sources, and comprehensive data preprocessing techniques, this study establishes the foundation for analyzing the dataset using various statistical models. Employing exploratory data analysis (EDA) to gain insights into variable distribution and relationships, the study systematically evaluates six classifiers—Logistic Regression, KNN, Random Forest, Decision Tree, Support Vector Machine, and an Ensemble method—to construct predictive models for mental health treatment outcomes.

By systematically applying machine learning algorithms and statistical techniques, this study seeks to contribute to the expanding field of knowledge on predicting mental health treatment outcomes.

2. Literature Review:

The study used machine learning methodologies such as random forest, k-nearest neighbors, and naïve Bayes to forecast mental well-being among university students. Employing pre-existing infrastructure, data collection, and analysis centered on health behaviors and demographic factors. The findings revealed that the random forest algorithm exhibited the most robust accuracy in predicting mental well-being, followed by k-nearest neighbors and naïve Bayes. Furthermore, the study identified ten prominent indicators associated with mental well-being, offering valuable insights for implementing cost-effective support strategies for university students.

The paper adhered to PRISMA guidelines, conducting a thorough review of studies since 2008 investigating the correlation between mental health and workplace productivity. Although specific technologies and algorithms were not explicitly mentioned, the study's findings indicated that individuals with mental disorders experienced increased work loss days due to absenteeism and presenteeism, contrasting with those with better mental health who displayed lower absenteeism rates and reduced presenteeism hours. Methodologically, the research utilized regression analysis, linear regression, logistic regression models, two-part models, Poisson regressions, Kaplan-Meier survival curves, and Cox's proportional hazards model to explore this relationship. Remarkably, treatment for major depressive disorder was linked to decreased absenteeism and improved job performance, underscoring the significant impact of mental health on productivity in the workplace.

The study systematically reviewed 30 research articles concerning machine learning methodologies in predicting mental health issues. It explored a range of techniques including supervised and unsupervised learning, ensemble learning, neural networks, and deep learning. The research process encompassed planning, conducting, evaluating, and discussing, with a focus on identifying pertinent documents and assessing machine learning performance. Findings involved the classification of mental health disorders such as schizophrenia, bipolar disorder, anxiety, depression, posttraumatic stress disorder, and children's mental health issues, with distinct accuracies reported for various classification tasks.

The study applies machine learning algorithms such as Logistic Regression, Random Forest, Decision Tree Classifier, and Naïve Bayes to forecast mental health disorders among IT employees. Methodologies encompass data preprocessing, graph mapping, and employing various algorithms for precise output prediction. Python is utilized for pattern recognition and implementing insights. Findings indicate that Random Forest, Decision Tree Classifier, and LAD Tree classifiers yield effective results with minimal variance, suggesting potential for future application on larger datasets to enhance accuracy.

3. Methodology:

About Data:

The data utilized in this study were gathered through surveys administered to individuals sourced from a variety of outlets such as mental health clinics, community organizations, and online platforms. The sample encompassed individuals with diverse demographic characteristics, including varying genders, ages, occupations, and geographical locations. This diverse sample aimed to offer comprehensive insights into mental health and treatment-seeking behaviors across a broad spectrum of demographics.

The dataset included a range of features, such as Timestamp, Gender, Country, Occupation, Self-Employed status, Family History of mental health issues, and whether individuals sought Treatment. Additionally, behavioral attributes such as Days Indoors, Growing Stress levels, Changes in Habits, Mental Health History, Mood Swings, Coping Struggles, Work Interest, Social Weakness, participation in Mental Health Interviews, and access to Care Options were included in the dataset. These features provided a multifaceted view of individuals' mental health statuses and behaviors related to seeking treatment, allowing for a comprehensive analysis of the factors influencing mental health outcomes.

Data Collection:

The dataset utilized in this study was sourced from Kaggle, a widely used platform for sharing datasets and conducting data-driven research. It likely comprised responses to surveys or questionnaires about mental health and individuals' behaviors related to seeking treatment.

Data Preprocessing:

The provided code snippet demonstrates the initial data preprocessing steps for a mental health dataset. It encompasses the following key steps:

Loading the Dataset: The dataset is loaded from a CSV file, which serves as the foundation for subsequent analysis.

Converting Categorical Columns: Categorical columns within the dataset are converted to appropriate data types. This ensures that categorical variables are represented accurately and efficiently for analysis.

Handling Missing Values: Missing values are addressed meticulously. For categorical columns, missing values are replaced with the mode, while for numerical columns, they are replaced with the mean. This step ensures the integrity of the dataset and facilitates meaningful analysis.

Dropping Unnecessary Columns: Columns deemed unnecessary for analysis, such as the 'Timestamp' column, are dropped. This streamlines the dataset and focuses attention on relevant variables.

Displaying Dataset Information: Comprehensive information about the dataset is presented before and after preprocessing. This includes details on data types, unique values, and percentages of missing values. Such insights are invaluable for understanding the characteristics and quality of the dataset

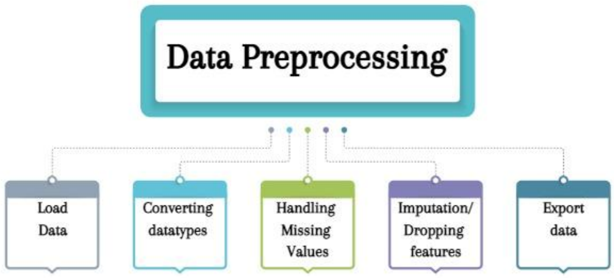


Fig1:Data preprocessing Steps applied to our Dataset

Exploratory Data Analysis (EDA)

The code snippet outlines a series of EDA tasks designed to glean insights from the dataset:

Generating Statistical Summaries: Statistical summaries are generated to provide a concise overview of the dataset. Descriptive statistics offer insights into central tendencies, variability, and distributions of variables.

Visualizing Distribution of Variables: The distribution of the target variable ('treatment') and categorical variables is visualized using count plots. Additionally, the distribution of numeric variables is depicted using histograms. These visualizations offer insights into the frequency and spread of values across different variables.

Exporting Processed Data

The code also includes functionality to export the preprocessed data to a new CSV file. This step ensures that the processed dataset can be stored and utilized for subsequent analyses or shared with collaborators.

By executing these preprocessing and EDA steps, researchers can gain a comprehensive understanding of the mental health dataset, uncover meaningful insights, and lay the groundwork for further analysis and interpretation.

Data Visualization and Analysis

The collected dataset was subjected to comprehensive visualization and analysis to gain insights into the distribution of various categorical variables about mental health and treatment-seeking behaviors.

Overall Plot

A 4x4 grid of subplots was created to visualize the distribution of categorical variables. Each subplot represents a different categorical variable, including features such as Gender, Country, Occupation, and others. The distribution of each variable was visualized using count plots, providing an overview of the frequency of different categories within each feature. The use of a smaller figure size for each plot ensured clarity and ease of interpretation.

Distribution of Categorical Variables

Further exploration of the dataset involved examining the distribution of each categorical variable individually. Count plots were generated for each variable, with bars representing the frequency of each category. The plots were ordered by the frequency of categories to facilitate comparison and interpretation. Additionally, a consistent color palette ('Set3') was utilized across all plots for coherence and visual appeal.

The visualizations revealed valuable insights into the distribution of demographic characteristics, mental health history, and attitudes toward seeking mental health care among the participants. For instance, the distribution of treatment-seeking behavior indicated variations in the proportion of individuals seeking treatment across different demographic groups and geographical locations.

These visualizations serve as a preliminary exploration of the dataset, providing a foundation for further analysis and hypothesis testing. The insights gleaned from the visualizations will inform subsequent stages of data analysis, including the identification of potential predictors of mental health treatment utilization and the development of predictive models.

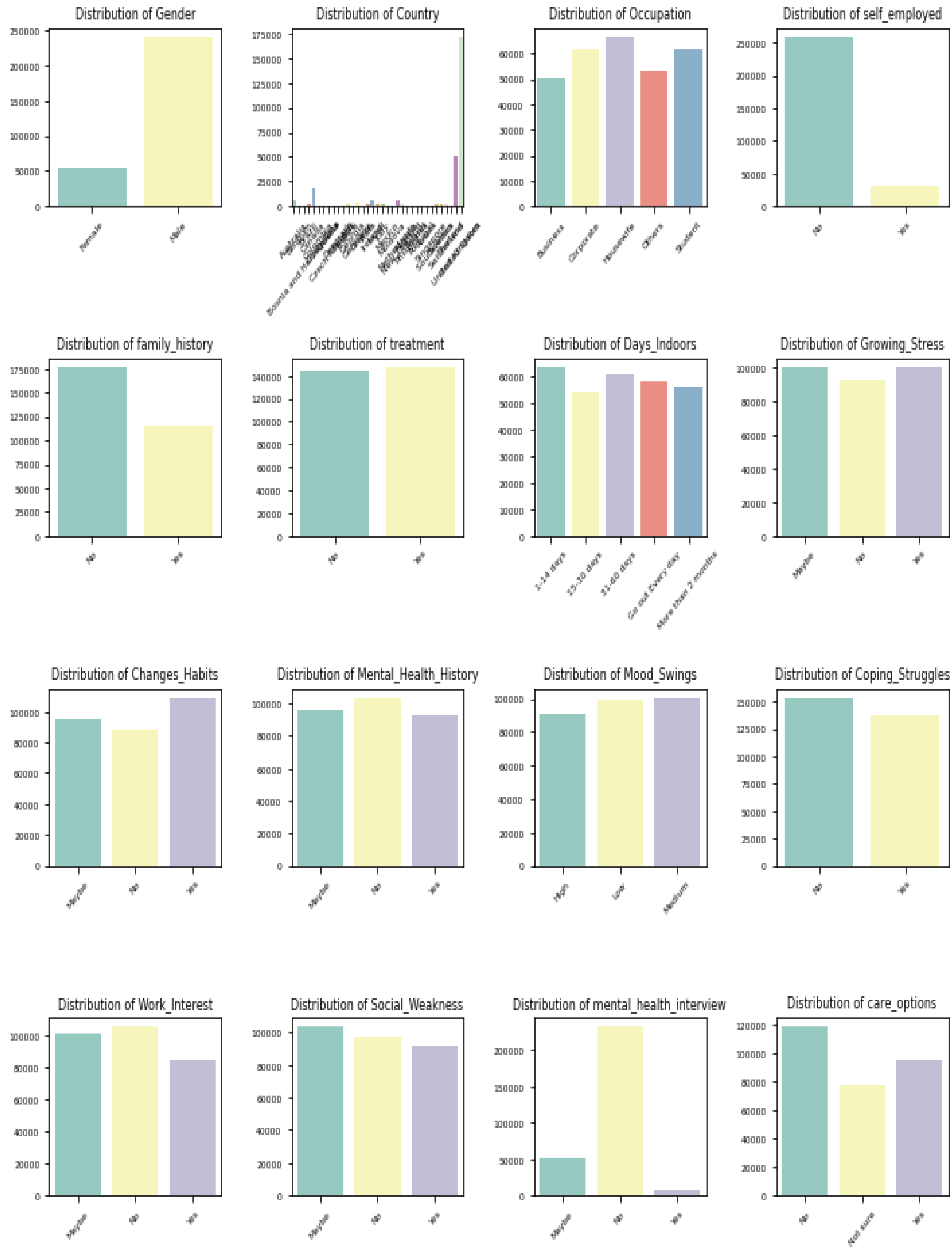


Fig-2: Visualization plots of all features

Starting Model

The process of developing and assessing a binary classification model for predicting whether individuals have received treatment for mental health issues involved several crucial steps. Initially, a preprocessed dataset was loaded and reduced to 50,000 samples. Categorical variables were converted into dummy variables, and the dataset was divided into training and testing sets using an 80-20 split.

Subsequently, feature selection was carried out using SelectKBest with f_regression scoring. This involved experimenting with different numbers of top features (ranging from 1 to 10) and varying numbers of folds for cross-validation (3 or 5). For each fold, a classifier was trained on the training set and evaluated on the validation set, computing metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, Accuracy, Precision, Recall, F1-Score, Specificity, and Balanced Accuracy (BCA Score)

Once all combinations of features and folds were tested, the model was trained on the entire training set using the selected top features. It was then evaluated on the test set using the same metrics. This process was repeated for six classifiers: KNN,

Random Forest, Decision Tree, Support Vector Machine, Logistic Regression, and an ensemble of the aforementioned five.

The results were sorted by accuracy, and the top five combinations of features and folds were chosen and presented. Finally, the findings were saved to an Excel file, providing a comprehensive method for model construction and assessment.

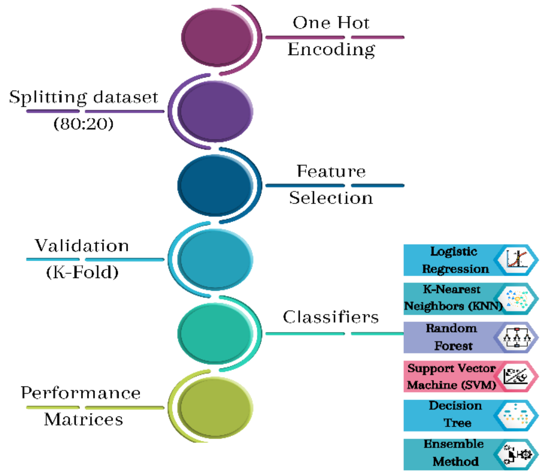


Fig-3: Model building steps

Model building and Evaluation of each classifier

We built a predictive model to determine the likelihood of individuals seeking mental health treatment using various classifiers: Random Forest, Support Vector Machine (SVM), Decision Tree, Logistic Regression, K-Nearest Neighbors (KNN), and an Ensemble method (Voting Classifier). The dataset was preprocessed to handle categorical variables and feature selection was performed using SelectKBest with f_regression.

A. *Logistic Regression Classifier:*

Effect & Resolution: Logistic Regression provides insights into feature importance and is effective for binary classification tasks.

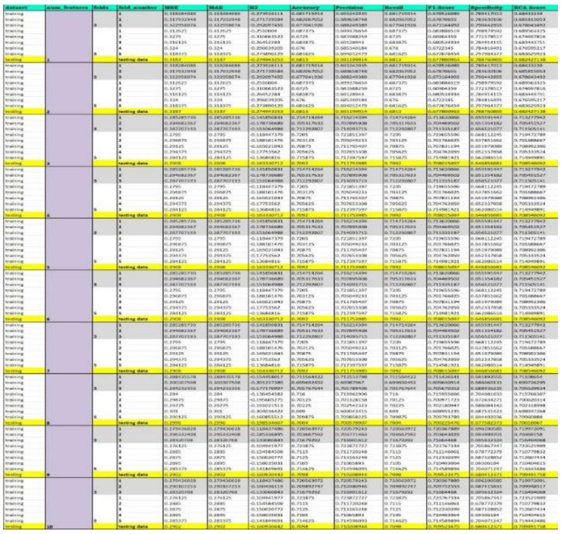
Model Building and Evaluation: The Logistic Regression model was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set. The model was then tested on the holdout test set to evaluate its performance.

Implementation:

```
logistic_classifier = LogisticRegression()

logistic_classifier.fit(X_train_fold, y_train_fold)

y_pred_fold = logistic_classifier.predict(X_val_fold)
```



dataset	num_features	folds	fold_number	MSE	MAE	R2	Accuracy	Precision	Recall	F1 Score	Specificity	BCA Score

Fig 4: Results of Logistic Regression Classifier

Model Building and Evaluation Summary:

The Logistic Regression model is built by estimating probabilities using a logistic function. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.

B. *Random Forest Classifier*

Effect & Resolution: Random Forest handles high-dimensional data and nonlinear relationships, reducing overfitting by averaging multiple decision trees.

Model Building and Evaluation: The Random Forest model was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set. The model was then tested on the holdout test set to evaluate its performance.

Implementation:

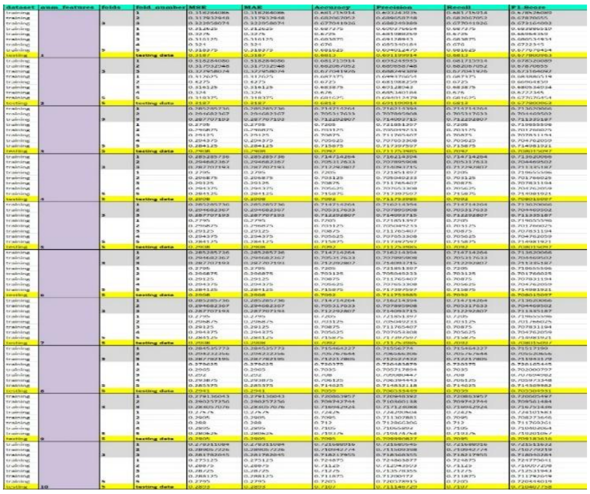
```
rf_classifier = RandomForestClassifier()

rf_classifier.fit(X_train_fold, y_train_fold)

y_pred_fold = rf_classifier.predict(X_val_fold)
```

Summary:

The Random Forest model is built by creating multiple decision trees during training and outputting the mode of the classes of individual trees. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.



dataset	num_features	folds	fold_number	MSE	MAE	Accuracy	Precision	Recall	F1-Score

Fig 5: Results of Random Forest Classifier

C. Support Vector Machine (SVM) Classifier

Effect & Resolution: SVM effectively separates classes in the feature space, particularly in high-dimensional or non-linearly separable data.

Model Building and Evaluation: The SVM model was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set. The model was then tested on the holdout test set to evaluate its performance.

Implementation:

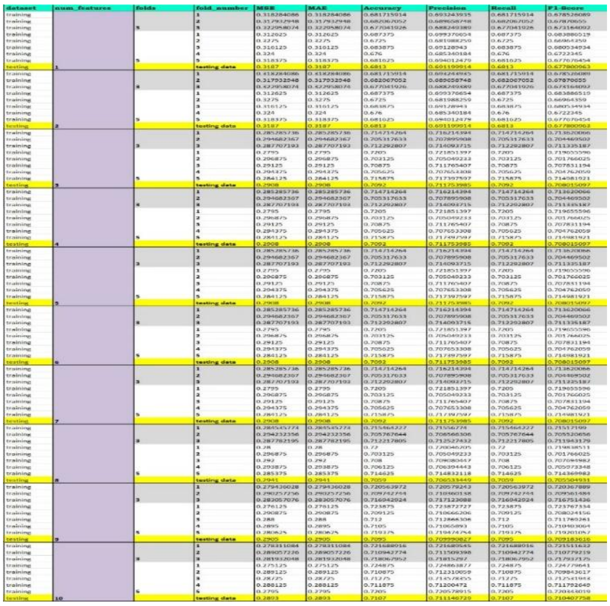
```
svm_classifier = SVC()

svm_classifier.fit(X_train_fold, y_train_fold)

y_pred_fold = svm_classifier.predict(X_val_fold)
```

Model Building and Evaluation Summary:

The SVM model is built by finding the hyperplane that best separates different classes in the feature space. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.



dataset	num_features	folds	fold_number	MSE	MAE	Accuracy	Precision	Recall	F1-Score

Fig 6: Results of Support Vector Machine (SVM) Classifier

D. Decision Tree Classifier

Effect & Resolution: Decision Trees handle numerical and categorical data, are easy to interpret, and capture complex relationships.

Model Building and Evaluation: The Decision Tree model was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set.

The model was then tested on the holdout test set to evaluate its performance.

Implementation:

```
dt_classifier = DecisionTreeClassifier()

dt_classifier.fit(X_train_fold, y_train_fold)

y_pred_fold = dt_classifier.predict(X_val_fold)
```

Model Building and Evaluation Summary:

The Decision Tree model is built by constructing a tree-like model of decisions and their possible consequences. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.

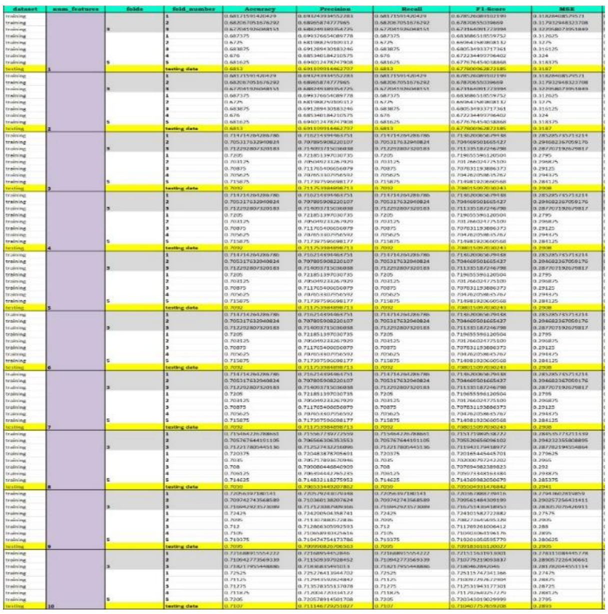


Fig 7: Results of Decision Tree Classifier

E. K-Nearest Neighbors (KNN) Classifier

Effect & Resolution: KNN captures local patterns and is robust to noisy data, considering the similarity of instances.

Model Building and Evaluation: The KNN model was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set. The model was then tested on the holdout test set to evaluate its performance.

Implementation:

```
knn_classifier = KNeighborsClassifier()

knn_classifier.fit(X_train_fold, y_train_fold)
```

Model Building and Evaluation Summary:

The KNN model is built by storing all instances corresponding to training data points in n-dimensional space. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.

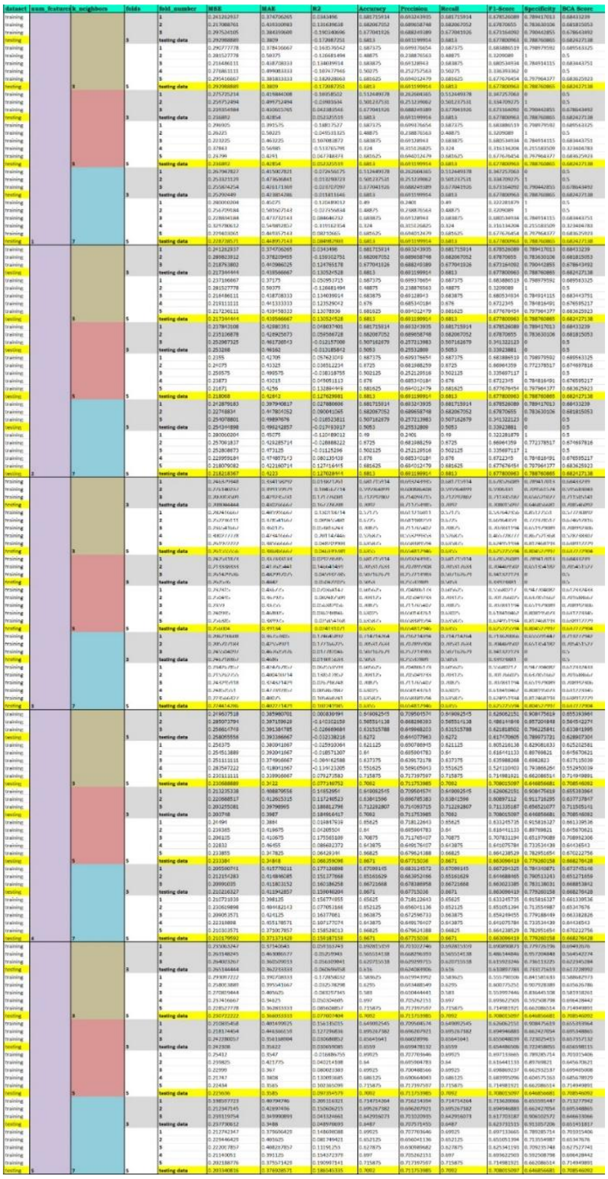


Fig 8: Results K-Nearest Neighbours (KNN) Classifier

F. Ensemble Method (Voting Classifier)

Effect & Resolution: Voting Classifier combines predictions to improve accuracy and stability, effective for complex datasets.

Model Building and Evaluation: The Voting Classifier was trained on the selected features and evaluated using k-fold cross-validation (k=3,5) on the training set. The model was then tested on the holdout test set to evaluate its performance.

Implementation:

```
ensemble_model = VotingClassifier(estimators=[
    ('knn', knn_classifier),
    ('logistic', logistic_classifier),
    ('dt', dt_classifier),
    ('rf', rf_classifier)
], voting='hard')
ensemble_model.fit(X_train_fold, y_train_fold)
y_pred_fold = ensemble_model.predict(X_val_fold)
```

Model Building and Evaluation Summary:

The Voting Classifier combines the predictions of multiple individual classifiers to improve overall performance. It is evaluated using k-fold cross-validation on the training set and tested on the holdout test set.

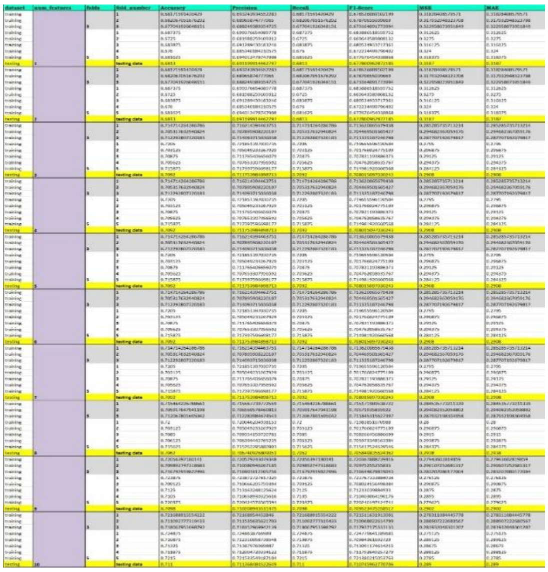


Fig 9: Results Ensemble Method Classifier

4.Results:

Classifier	Feature	Fold	Accuracy
LOGISTIC REGRESSION	10	5	0.723875
	9	5	0.723875
	10	3	0.720564
	9	3	0.720564
	4	5	0.7205001
KNN-MODEL	10(K=7)	5	0.7240001
	9(K=7)	5	0.7230001
	10(K=7)	3	0.720939
	10(K=7)	5	0.720375
	9(K=7)	5	0.7197501
RANDOM MODEL	10	5	0.7252501
	9	5	0.7242501
	10	3	0.721689
	9	3	0.720864
	4	5	0.7205001
DECISION TREE	10	5	0.7252501
	9	5	0.7242501
	10	3	0.721689
	9	3	0.720564
	4	5	0.72050001
SUPPORT VECTOR MACHINE	10	5	0.7107
	9	5	0.7095
	3	5	0.7092
	4	5	0.7092
	5	5	0.7092
ENSEMBLE METHOD	10	5	0.71101
	9	5	0.7098
	3	5	0.7092
	4	5	0.7092
	5	5	0.7092

Fig 10: Top best results from each classifier

Based on the provided table, it appears that the Random Model and Decision Tree performed the best in terms of accuracy, achieving scores of 0.7252501 and 0.7252501, using feature selection of 10 features and 5-fold cross-validation respectively. These models outperformed Logistic Regression, KNN-Model, Support Vector Machine, and Ensemble Method. However, it's important to note that the performance can vary depending on the specific dataset and problem at hand. Further analysis could include evaluating other metrics like precision, recall, and F1-score to gain a more comprehensive understanding of the models' performance.

5.Conclusion

In conclusion, our study systematically evaluated six machine learning classifiers for predicting mental health treatment outcomes among university students and IT employees. The Random Forest and Decision Tree models emerged as the top performers, achieving the highest accuracy scores of 0.7252501 with 10 features and 5-fold cross-validation, respectively. These models outperformed Logistic Regression, KNN, Support Vector Machine, and the Ensemble Method.

However, it's crucial to consider that model performance can vary depending on the dataset and problem context. Further analysis could include evaluating additional metrics such as precision, recall, and F1-score to gain a more comprehensive understanding of the models' performance. Overall, our study underscores the potential of machine learning algorithms in predicting mental health outcomes and highlights the importance of systematic evaluation and interpretation to inform effective support strategies and enhance workplace productivity.

6. REFERENCES

1. "Machine Learning-Based Prediction of Mental Well-Being Using Health Behavior Data from University Students" authored by Hanif Abdul Rahman, Madeline Kwicklis, Mohammad Ottom, Areekul Amornsriwatanakul, Khadizah H. Abdul-Mumin, Michael Rosenberg, and Ivo D. Dinov.

2. PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews

3. Machine Learning Approaches in Predicting Mental Health Problems

4. Prediction of Mental Health Using Machine LearningAuthors: Dr. J. Arokia Renjit, Adlin Sajeesha M.J, Sangavai V.D, Sree Devi D.S