KOMAL KUMARI(MT19124), DIMPY VARSANI (MT19022), AKANKSHA DEWANGAN(MT19049)

Report

Abstract:

Mental health is a major factor for human beings. It will affect emotional, psychological and social factors of an individual which determines the thought process of an individual. Healthy mind impacts on potential and productive work. Mental health plays important roles in every phase of life from childhood to adolescence throughout adulthood to old age. With growing age our social surrounding puts out a lot of thoughts, positive ones lead to a healthy & progressive mind and the negative ones contribute to mental illness like stress,depression, social anxiety etc. For good mental health it is important to determine the mental illness. Machine learning is one of the areas that will be good in predicting onset mental illness.
This kind of prediction model will help the society as a monitoring tool for individuals to deviate individual behavior. We had used various machine learning algorithms such as logistic regression, KNN classifier, Decision tree classifier, Random forest, Bagging with decision tree classifier, Boosting of Adaboostclassifier, along with stacking of various algorithm together to identify the mental health in a target group. Target groups are working professionals, college and school going students. Other than supervised learning algorithms we also applied unsupervised learning that is a clustering algorithm named as agglomerative clustering technique. Along with the machine learning techniques some artificial intelligence networks were also tried like Neural network, Deep learning neural network for predicting mental health. We have used evaluation metrics such as Accuracy, precision, recall and AUC-ROC curves.

Dataset:

This dataset is from a 2014 survey that measures attitudes towards mental health and frequency of mental health disorders in the tech workplace. It has 1259 rows and 27 attributes which help to determine the mental illness of individuals.

Below are the details of the attribute:

- Timestamp
- Age
- Gender
- Country
- state: If you live in the United States, which state or territory do you live in?
- self_employed: Are you self-employed?
- family_history: Do you have a family history of mental illness?
- treatment: Have you sought treatment for a mental health condition?
- work_interfere: If you have a mental health condition, do you feel that it interferes with your work?
- no_employees: How many employees does your company or organization have?
- remote_work: Do you work remotely (outside of an office) at least 50% of the time?
- tech_company: Is your employer primarily a tech company/organization?
- benefits: Does your employer provide mental health benefits?
- care_options: Do you know the options for mental health care your employer provides?
- wellness_program: Has your employer ever discussed mental health as part of an employee wellness program?
- seek_help: Does your employer provide resources to learn more about mental health issues and how to seek help?

- anonymity: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?
- leave: How easy is it for you to take medical leave for a mental health condition?
- mental health consequence: Do you think that discussing a mental health issue with your employer would have negative consequences?
- phys health consequence: Do you think that discussing a physical health issue with your employer would have negative consequences?
- coworkers: Would you be willing to discuss a mental health issue with your coworkers?
- supervisor: Would you be willing to discuss a mental health issue with your direct supervisor(s)?
- mental health interview: Would you bring up a mental health issue with a potential employer in an interview?
- phys health interview: Would you bring up a physical health issue with a potential employer in an interview?
- mental vs physical: Do you feel that your employer takes mental health as seriously as physical health?
- obs_consequence: Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?
- comments: Any additional notes or comments

Methodology:

This section includes the process of building the prediction models along with the analysis of the dataset and reason to select the particular features along with evaluation metrics which assess the prediction system.

A. Preprocessing:

-Replacing synonymes with similar meaning : there are different type of synonyme words for male, female and transgender:

```
male=["male", "m", "male-ish", "maile", "mal", "male (cis)", "make", "male ", "man","msle", "mail", "malr","cis man", "Cis Male", "cis male"]
transgender=["trans-female", "something kinda male?", "queer/she/they", "non-binary","nah", "all", "enby", "fluid", "genderqueer", "androgyne",
female=["cis female", "f", "female", "woman",  "femake", "female ","cis-female/femme", "female (cis)", "femail"]
```

they are replace with ‘male’,’female’,’transgender’.

```
['female' 'male' 'transgender']
```

As gender is categorical column, and after label encoding it would have only 3 values 0, 1 and 2.

-Replace missing values with mean:

In Age columns there are a lot of missing values which are filled by 0 so the age should have some number so that model will train well on the basis of age too if that is an important feature, the reason why we replaced 0 values of age with mean value of all the ages in that column.

-Generating category instead of NaN values:

In ‘self employed’ columns there are only 0.014% of self employed so let's change NaN to NOT self_employed. And in the ‘work interfere’ column there are only 0.20% of self work_interfere so let's change NaN to "Don't know”.

-Label encoding of categorical columns:

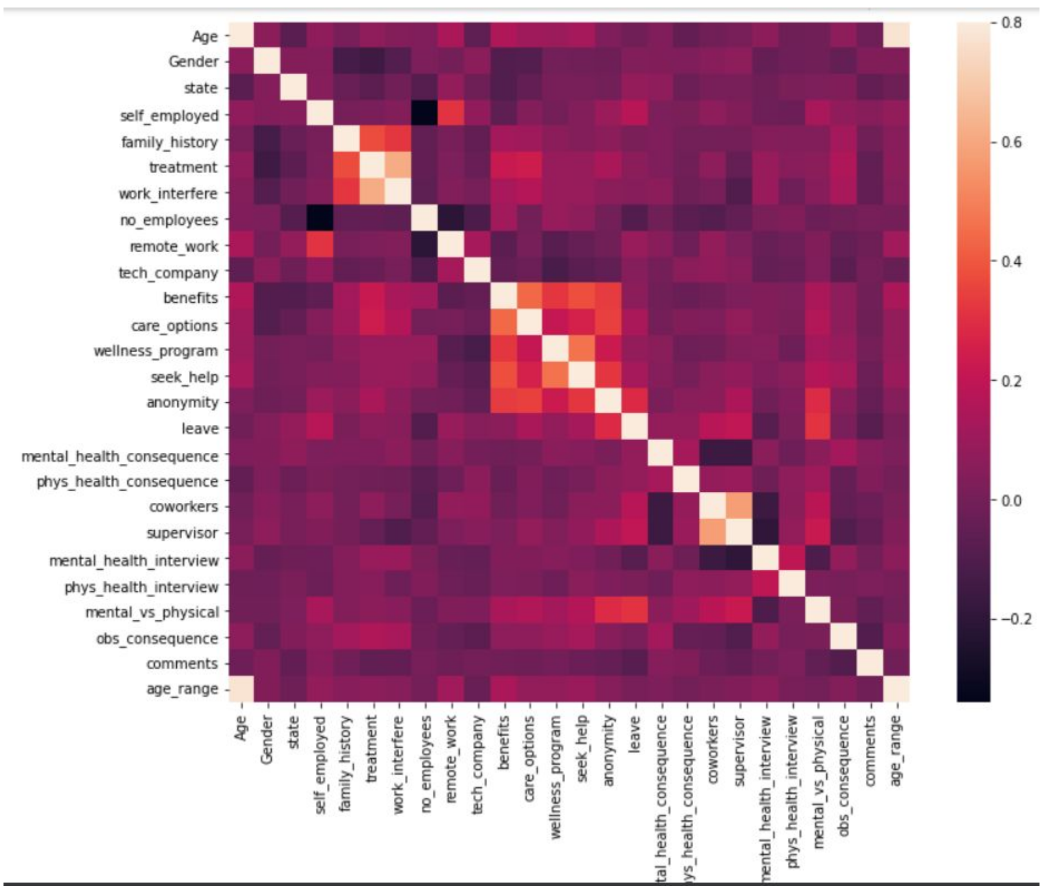
All the columns except the 'Age' is label encoded according to the unique values present in each column. Following columns are label encoded:

```
Timestamp  :  [   0    1    2 ... 1241 1242 1243]
Age  :  [19 26 14 13 15 17 21 24  5 11 18  9 28 23 16 12 22 20 32  6  0 10  8  4
  1  7 27  3 25 37 40 35 36 30  2 38 39 29 42 33 43 31 34 41 44]
Gender  :  [0 1 2]
state  :  [10 11 29 38 37 18 30  2  4 16 28 22 15  8 33 42 43 39 26 32  6 19 20  1
  3  7  5 23 44 31 12 40 24 13  0 27 25 35 41 36  9 21 34 45 14 17]
self_employed  :  [0 1]
family_history  :  [0 1]
treatment  :  [1 0]
work_interfere  :  [2 3 1 4 0]
no_employees  :  [4 5 2 1 0 3]
remote_work  :  [0 1]
tech_company  :  [1 0]
benefits  :  [2 0 1]
care_options  :  [1 0 2]
wellness_program  :  [1 0 2]
seek_help  :  [2 0 1]
anonymity  :  [2 0 1]
leave  :  [2 0 1 3 4]
mental_health_consequence  :  [1 0 2]
phys_health_consequence  :  [1 2 0]
coworkers  :  [1 0 2]
supervisor  :  [2 0 1]
mental_health_interview  :  [1 2 0]
phys_health_interview  :  [0 1 2]
mental_vs_physical  :  [2 0 1]
obs_consequence  :  [0 1]
comments  :  [103  66  37  92 113 120  42 109  39  76 126  75  78  71  64  60  19  16
  32  87 152  96  24  79   3 142 101 146  69  88  50 108   6 150  33  38
 105  47 118  83  20 131 135  15  49  48 112 151  73   4 100  95  54  11
  68  74 153  46 116 107 104  28 158  29  80 121   2  56  26  65  44 130
  61  90 127  17  91  81 134  10  55  52  63 143 157  86  40  30  23  18
 144  21  84 114  58  34  31  93 117  57 124  82  62 106  36   8 149 128
 122 132   0  59  13 141   5 147  97   1  35 119 139  22 123 148 102  98
 133  45 136  43 125  70 138  25  72   9 137 111 140  51  94  89 156 115
  14 145 110  27  67  99 155  41  85 129   7  53 154  77  12]
age_range  :  [2 1 0 3]
```

B.Exploratory data analysis [EDA]:

After all the above steps of preprocessing we now analyse the relations between attributes as well as the analysis of each individual column

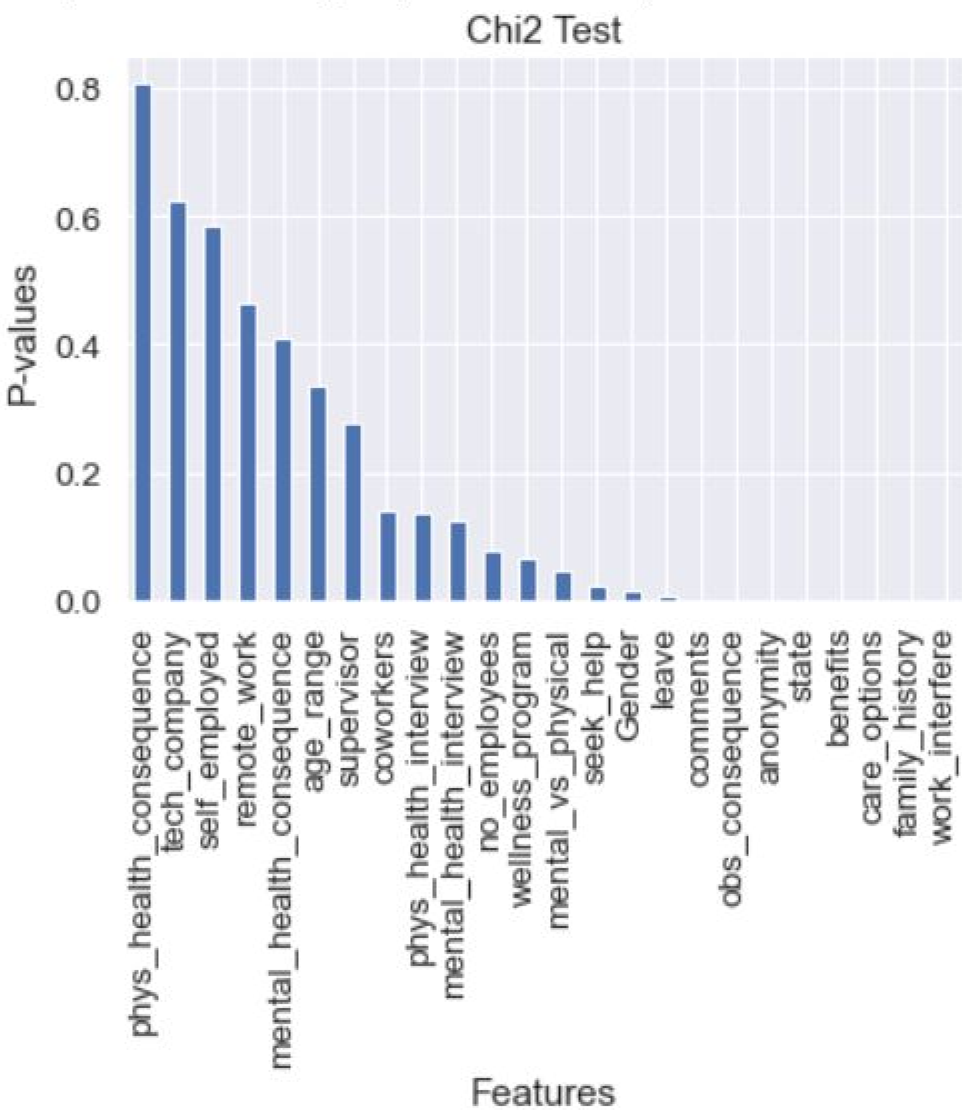
-Numerical Correlation matrix:
Here we plot the correlation matrix between each attribute to see how much percentage two attributes are similar to each other. The following matrix color ranges are between o to 1. Diagonal matrices are white in color means are highly correlated i.e. 100% similar and the dark ones totally dissimilar columns.
Here 'age' and 'age_range' are totally similar because its color is totally whitei.e. Having values are almost the same, Other than that 'self employment' and 'no_employees' are having totally different values, else all other attributes are 40 to 50% similar only in terms of column values.



-Categorical Correlation matrix:

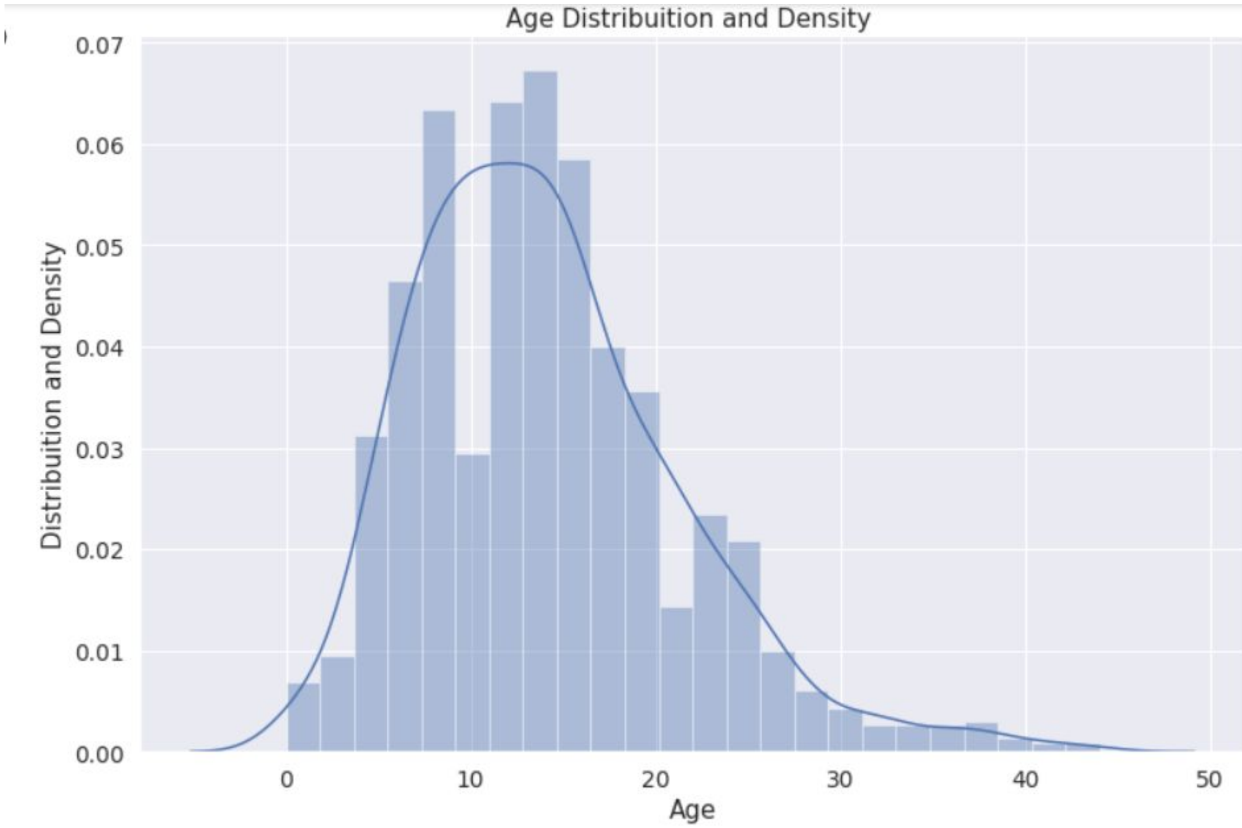
As here all the columns are categorical so we apply the 'chi2' correlation method to find the similarity of the categorical feature with the target variable 'treatment'. The p-values determine the importance of the feature for precision. A higher p-value implies lower correlation with the target variable and vice-versa.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdfb35f3970>
```



-Distribution of Age attributes :

Here the interpretation is the age range between 0 to 80 are showing the gaussian distribution bell shaped curve. Which tells us that 65 to 70% present a dataset for peoples are of age 10 to 16 say adolescence age. Rest observations in high peaks are also of age 6 to 10 are 65% in the dataset are importantly observed for mental health rest all are around 20 to 25 % of age 20 to40 age range are observed.

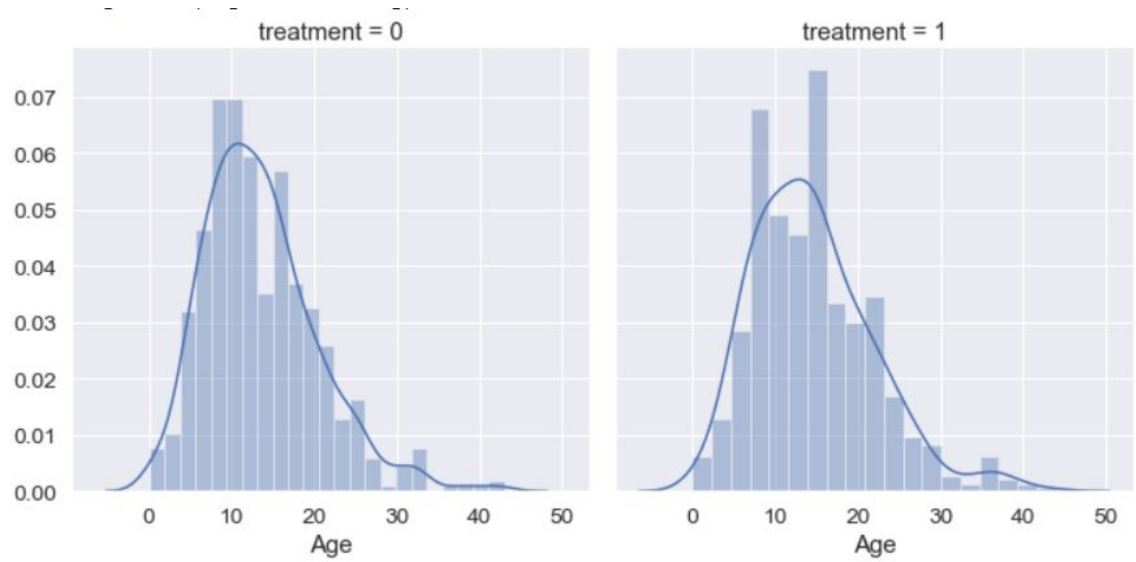


- **Age distribution along with differentiating on the basis of treatment or no_tratement.**

  Interpretation is

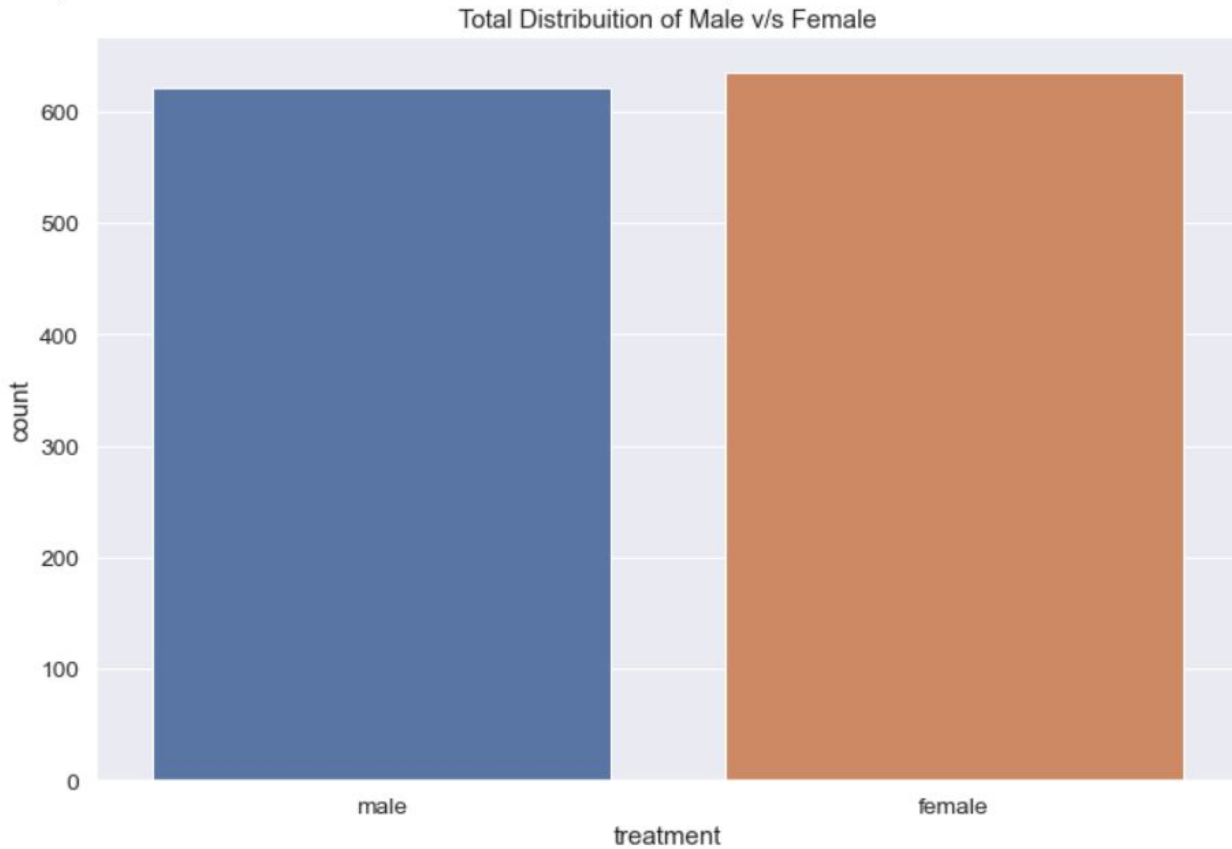
  A. treatment=0, observation of that age group who are not treated yet as per mental illness.
     So the age group of 5 to 10 are around 70% in strength and need treatment and the rest 20 to 25 age group peoples age are around 10 to 30% who are untreated.
  B. Treatment =1, observation of age groups who are treated well after observing their mental illness.
     In high peaks 9 and 16 aged childrens treated after

getting information of mental illness, their strength is 65 to 70 % in the overall dataset. And rest 10 to 30 age groups only 30% strength people treated well on mental illness.



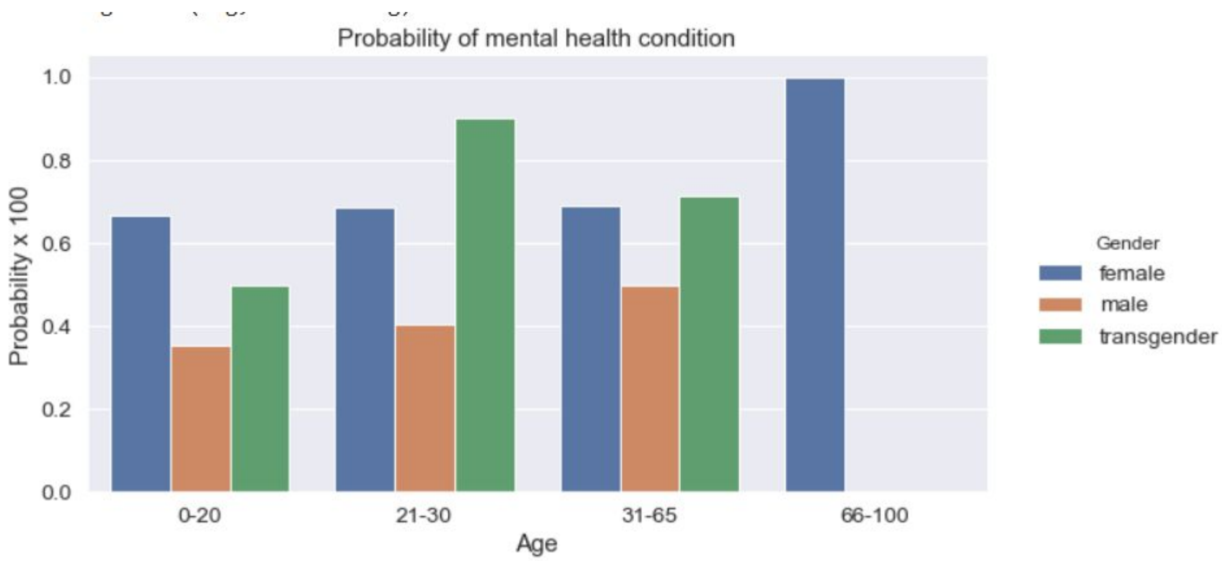
- **Now let observed the number of peoples who are treated well by looking male and female:**

Interpretation is both male and female treatment distribution is 650 for male and 680 for female almost equally in count.

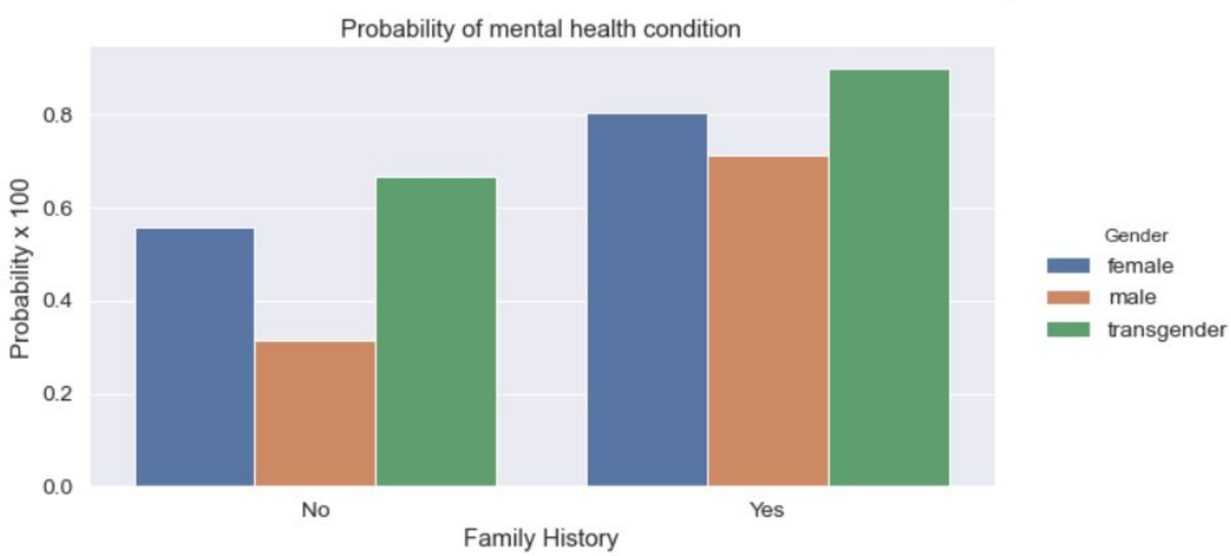


-Below graph shows the observation of probability of peoples on the basis of their gender and which age group treatment as per their age group among them.

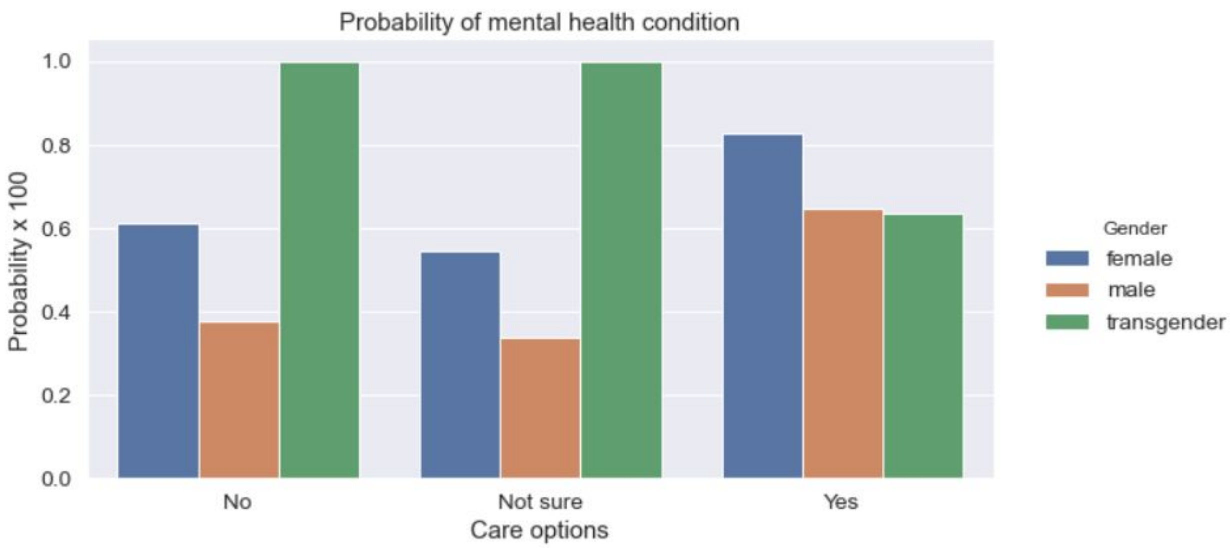
It is observed that Age between 66 to 100 female treatment probability is 1, about males treatment compared to female nd transgnder are very less probability of treatment mental health and transgender are in good prababilty range in every age range group.



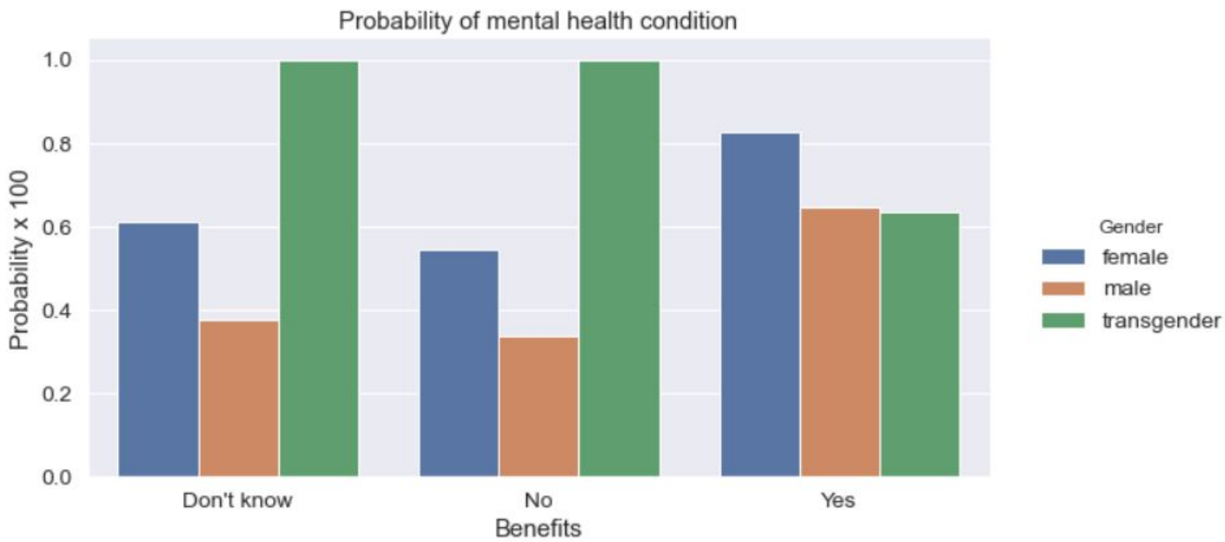
- **Observation of gender on the basis of family history:** It will be observed whether people have any history of mental illness. It interprets that 0.25 male are not having anty ,mental illness family history and 0.64 to 0.7 males have mental illness family history. Now about transgeder group more than 60% were having no illness history for mental health and more than 80% having mental illness history in their family. In overall observation transngender having highest mental illness history in their g=family and males are the least in mental illness history group.



- **Observation of genders on the basis of care options :** the options for mental health care your employer provides were provided or not by providing 3 replies yes, no or not_sure. Interpretation is Mostly about 90% trangender group are in cases of no and not sure for care provided, where as male and female around 10 to 50 % in this cases of sure and not sure. Else about the 'yes' category of care provided females are highest in range then male and trangender or we can say transgnder are least with probability around 60%.

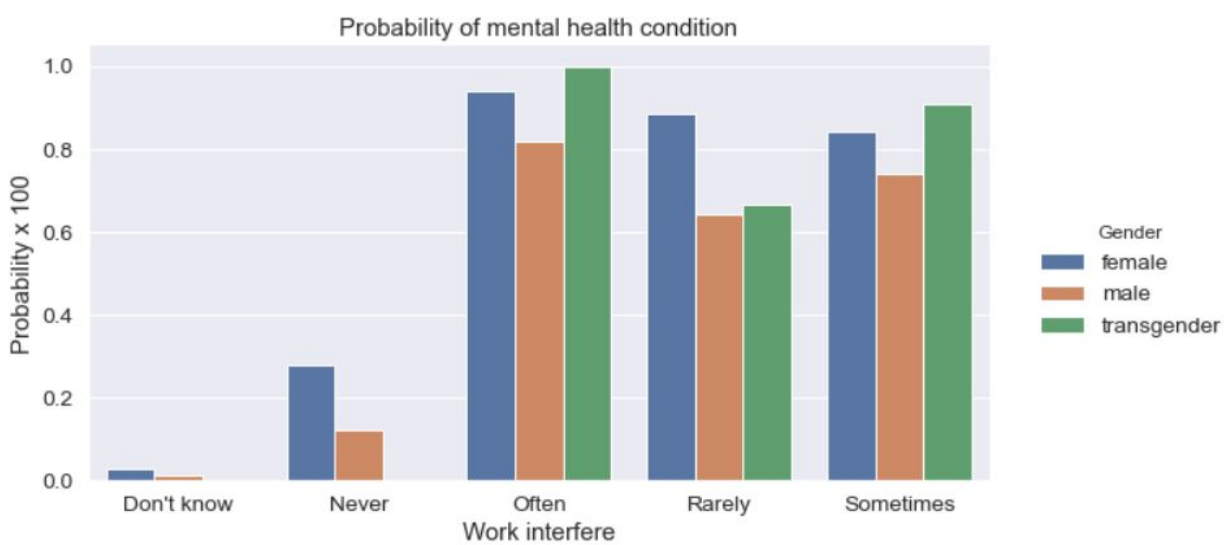


-Observation of gender on the basis of benefits: The employer provides mental health benefits or not in the category of yes, no and 'Don't know. ', So the useful intrepretaion is transgender are hrighest in group of non benfits as they are of higher percentage of no & dont know, and females are the highest which are getting benefits as an employee for mental health if you observe 'yes' catagory for gender.



-Observation on the basis of work interfered attribute:

Check " If you have a mental health condition, do you feel that it interferes with your work'. So it is observed that females in all categories have feelings of interference then male and transgeneder.



C.Scaling and Fitting:

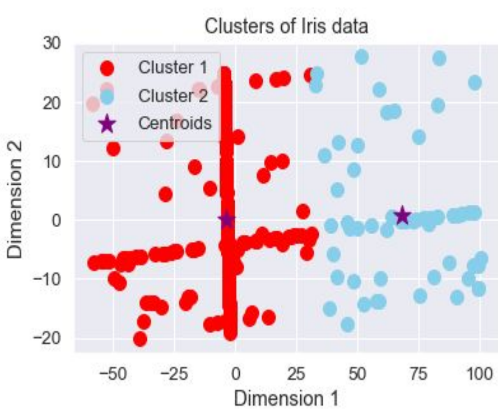
Now here we scaled down the attribute 'Age' and 'treatment' values in range 0 to 1, by min-max scalar conversion because their values are too high and need to be in some range so that they can't divert the model for the correct prediction of mental illness. Because sometimes irrelevant higher values may vary the correct predictions.

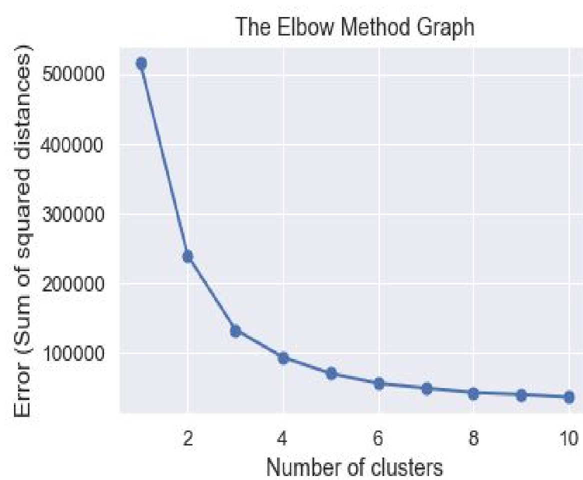
Following formula for minmax conversion:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

D.KMeans Clustering: The Kmeans clustering have been used to cluster the points into optimal number of clusters

The optimal number of clusters have been found using the elbow method by searching over several values of k. As per the elbow method the optimal number of clusters are 2.





E.Feature selection:

Here we are doing feature selection so that we can train the columns based on relevant columns only and reduce the overfitting by not training the model with each and every feature. It is an important task to remove features not relevant to the prediction.

Below are the few features selection method used:

- Select-K Best: this select feature has the best correlation with the target variable based on the chi2, mutual information or f_classif score.
- Variance Threshold: this selects the feature which has variance above a given threshold. It helps to retains feature with maximum information content
- Linear Discriminant Analysis: this select feature while training the model using the train data. This is an embedded feature selection technique.

F. Split Train-Test:

Split the dataset into a 80:20 ratio of train test data set first to train the model on 80% dataset and then by keeping ground truth of test data of 20% we can check the accuracy or correctness of the prediction model.

G. Cross validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample , to skill the model to perform well on unseen dataset. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. Here we used k=5 becoming 5-fold cross-validation.

H. GridsearchCV

GridsearchCV is defined as exhaustive search over specified parameter values for an estimator. It is used through predefined hyperparameters and fits your model or estimator on your training set. Hence we can select the best parameters from the listed hyperparameters

1. K Nearest Neighbour: Grid search for finding the best value for parameter K. The value of K searched moved across 1 to 31.

```
GridSearch best score 0.7652444444444445
GridSearch best params {'n_neighbors': 27}
GridSearch best estimator KNeighborsClassifier(n_neighbors=27)
```

2. Random Forest: Fine Tuning of parameters like min_sample_split, max_dept, score function, number of features.

```
GridSearch best score 0.8353015873015874
GridSearch best params {'criterion': 'gini', 'max_depth': None, 'max_features': 10, 'min_samples_leaf': 9, 'min_sampl
es_split': 9}
GridSearch best estimator RandomForestClassifier(max_features=10, min_samples_leaf=9, min_samples_split=9,
                       n_estimators=20)
```

3. Decision Tree Classifier: Fine Tuning of parameters like min_sample_split, max_dept, score function, number of features.

```
GridSearch best score 0.8313142857142857
GridSearch best params {'criterion': 'gini', 'max_depth': 3, 'max_features': 15, 'min_samples_leaf': 5, 'min_samples_
split': 5}
GridSearch best estimator DecisionTreeClassifier(max_depth=3, max_features=15, min_samples_leaf=5,
                       min_samples_split=5)
```

I.Models:

There are following models were used in prediction mental illness:

1.Logistic Regression:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. Following parameter are used:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

2.KNN classifier:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. Here KNN is used for classification. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.[3]

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=27, p=2,
                     weights='uniform')
```

3.Decision Tree Classifier:

Decision Tree is a Supervised learning technique that is used here as classification .It is mostly preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.[4]

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=3, max_features=6, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=7, min_samples_split=8,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

4.Random Forest:

It is supervised algorithm technique and here it is used for classification.random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.[5] It will reduce bias and variance, Hence reduce overfitting.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=8, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=20,
                       n_jobs=None, oob_score=False, random_state=1, verbose=0,
                       warm_start=False)
```

5.Bagging:

Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction[6]. Here bagging with the Decision tree classifier.

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                        class_weight=None,
                                                        criterion='gini',
                                                        max_depth=None,
                                                        max_features=None,
                                                        max_leaf_nodes=None,
                                                        min_impurity_decrease=0.0,
                                                        min_impurity_split=None,
                                                        min_samples_leaf=1,
                                                        min_samples_split=2,
                                                        min_weight_fraction_leaf=0.0,
                                                        presort='deprecated',
                                                        random_state=None,
                                                        splitter='best'),
                  bootstrap=True, bootstrap_features=False, max_features=1.0,
                  max_samples=1.0, n_estimators=10, n_jobs=None,
                  oob_score=False, random_state=None, verbose=0,
                  warm_start=False)
```

6.Boosting:

Boosting is an ensemble algorithm that reduces variance and bias, where we convert weak learners to strong learners.

AdaBoost is short for Adaptive Boosting and is a very popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier”, and here Decision tree is the classifier which is used as inside adaboosting for classification.

```
AdaBoostClassifier(algorithm='SAMME.R',
                   base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                         class_weight=None,
                                                         criterion='entropy',
                                                         max_depth=1,
                                                         max_features=None,
                                                         max_leaf_nodes=None,
                                                         min_impurity_decrease=0.0,
                                                         min_impurity_split=None,
                                                         min_samples_leaf=1,
                                                         min_samples_split=2,
                                                         min_weight_fraction_leaf=0.0,
                                                         presort='deprecated',
                                                         random_state=None,
                                                         splitter='best'),
                   learning_rate=1.0, n_estimators=500, random_state=None)
```

7.Stacking:

“Stacking”; for short is an ensemble machine learning algorithm. It involves combining the predictions from multiple machine learning models on the same dataset, like bagging and boosting.[7] Here we stacked KNN(n=1), random forest and gaussian naive bayes classifiers. Meta classifier used is logistic regression.

```
StackingClassifier(average_probas=False,
                   classifiers=[KNeighborsClassifier(algorithm='auto',
                                                     leaf_size=30,
                                                     metric='minkowski',
                                                     metric_params=None,
                                                     n_jobs=None, n_neighbors=1,
                                                     p=2, weights='uniform'),
                                RandomForestClassifier(bootstrap=True,
                                                       ccp_alpha=0.0,
                                                       class_weight=None,
                                                       criterion='gini',
                                                       max_depth=None,
                                                       max_features='auto',
                                                       max_leaf_nodes=None,
                                                       max_samples=None,...
                   meta_classifier=LogisticRegression(C=1.0, class_weight=None,
                                                      dual=False,
                                                      fit_intercept=True,
                                                      intercept_scaling=1,
                                                      l1_ratio=None,
                                                      max_iter=100,
                                                      multi_class='auto',
                                                      n_jobs=None, penalty='l2',
                                                      random_state=None,
                                                      solver='lbfgs',
                                                      tol=0.0001, verbose=0,
                                                      warm_start=False),
                   store_train_meta_features=False, use_clones=True,
                   use_features_in_secondary=False, use_probas=False,
                   verbose=0)
```

8.DNNClassifier:

DNNClassifier for deep models that perform multi-class classification. Model having characteristics :

```
num_folds=5  # number of folds
epochs=200 # number of epochs
batch_size=64 # batch size
epochs =200
```

And here in prediction we put a threshold on prediction that if predicted value is greater than 0.5 then it will be 1 else it will be 0.

10.gaussianNB:

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

11.SVM classifier:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms. Here svm is used as a classifier. Here we create a decision boundary which can segregate n dimensional space into classes, so that we can put points on the correct category. The best boundary is known as hyperplane that separate two classes.

```
SVC(C=1, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.005, kernel='rbf',
    max_iter=-1, probability=True, random_state=9, shrinking=True, tol=0.001,
    verbose=False)
```

J.Evaluation Metric:

Evaluating a Classification Model. This function will evalue:

1. **Classification accuracy**: percentage of correct predictions
2. **Confusion matrix**: Table that describes the performance of a classification model
   True Positives (TP): we correctly predicted that they do have diabetes **True Negatives (TN)**: we correctly predicted that they don't have diabetes **False Positives (FP)**: we incorrectly predicted that they do have diabetes (a "Type I error") Falsely predict positive
   False Negatives (FN): we incorrectly predicted that they don't have diabetes (a "Type II error") Falsely predict negative False Positive Rate
3. **Precision of Positive value**
4. **AUC**: is the percentage of the ROC plot that is underneath the curve .90-1 = excellent (A) .80-.90 = good (B) .70-.80 = fair (C) .60-.70 = poor (D) .50-.60 = fail (F) And some others values for tuning processes.

5. **F1-Score**

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

TP = number of true positives
FP = number of false positives
FN = number of false negatives

6. **Cross Validation Score :** Mean test accuracy on the K Fold validation structure
7. **Recall**

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

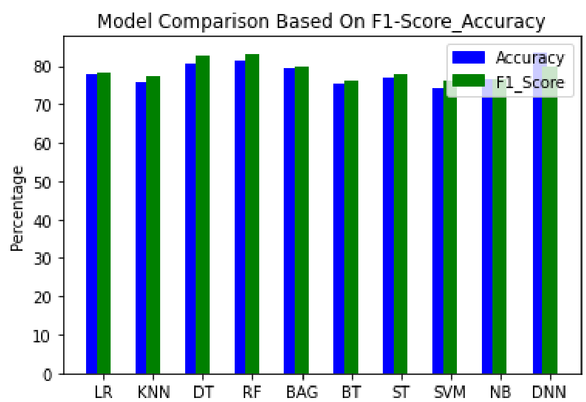
$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

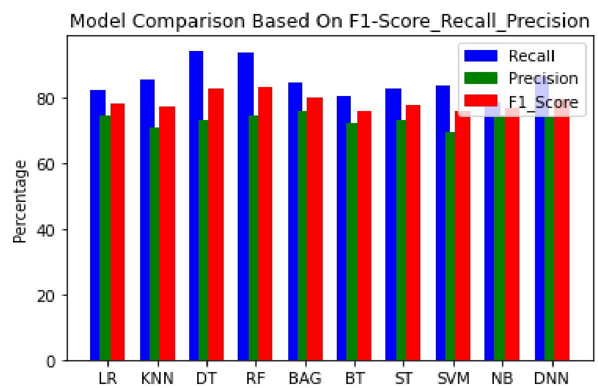
8. **Specificity**

Sensitivity = TP/(TP+FN)
Specificity = TN/(TN+FP)

Results:

Models	Accuracy	Specificity	F1 -score (%)	Precision (%)	Recall (%)	AUC- score	Cross-validation AUC
Logistic Regression	77.77	73.64	78.29	74.81	82.11	88.21	77.87
KNN Classifier	75.79	66.66	77.49	70.94	85.36	76.01	80.29
Decision Tree Classifier	80.55	67.44	82.56	73.41	94.3	80.87	80.41
Random Forest	81.34	69.76	83.03	74.67	93.49	81.63	88.83
Bagging	79.36	74.41	79.99	75.91	84.55	79.48	85.7
Boosting	75.39	70.54	76.15	72.26	80.48	75.51	86.81
Stacking	76.98	71.31	77.86	73.38	82.92	77.12	84.94
SVM Classifier	74.2	65.11	76.01	69.59	83.73	74.42	84.3
Naive Bayes Classifier	76.58	74.41	76.67	74.61	78.86	76.64	87.14
Deep Neural Network	83.571	71.31	79.69	74.12	86.217	78.74	83.78

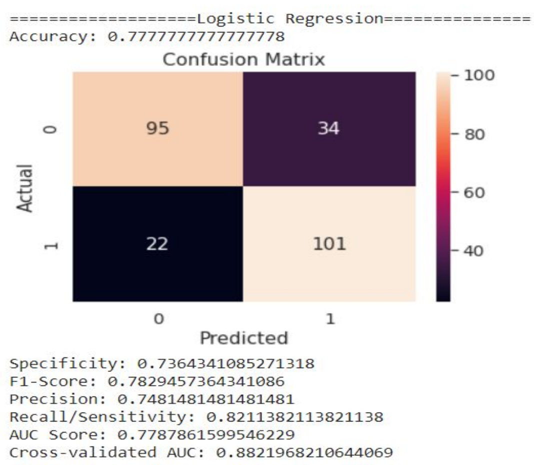


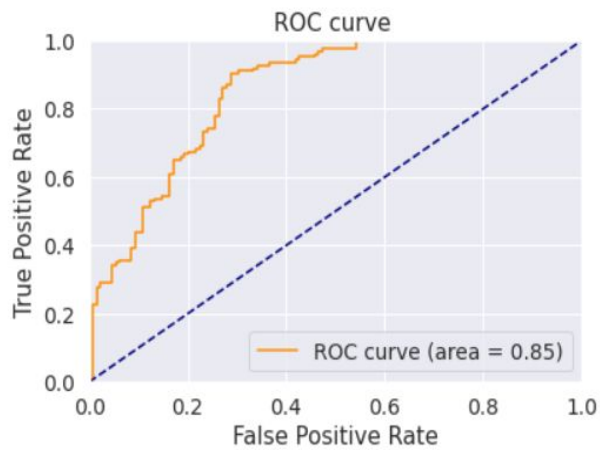


Analysis of result:

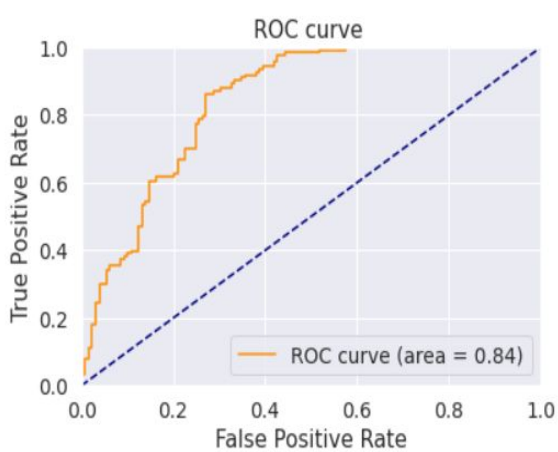
All graphs of results will be plotted here along with their analysis.

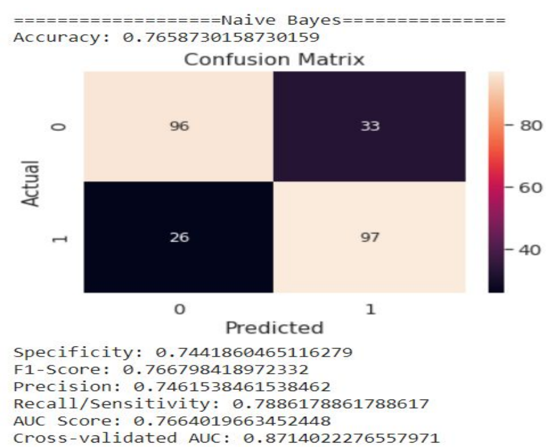
1.Logistic Regression:





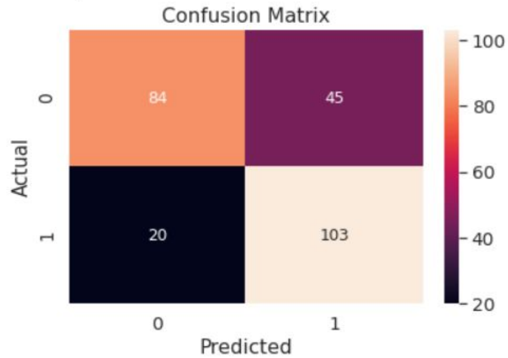
2.NaiveBayes Classifier:



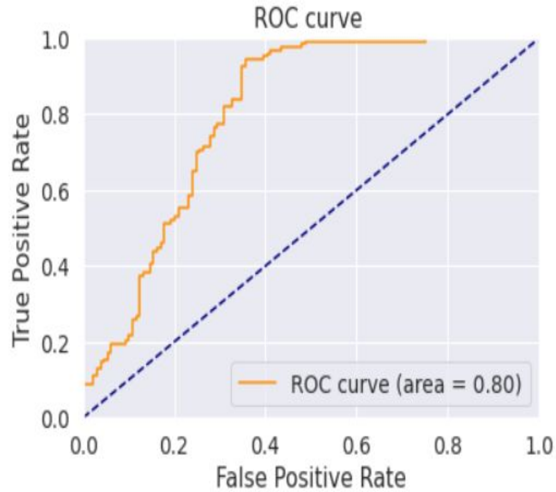


3.SVM classifier:

```
==================Support Vector Machine==============
Accuracy: 0.7420634920634921
```

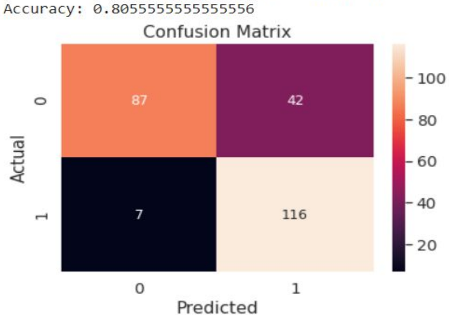



```
Specificity: 0.6511627906976745
F1-Score: 0.7601476014760147
Precision: 0.6959459459459459
Recall/Sensitivity: 0.8373983739837398
AUC Score: 0.7442805823407072
Cross-validated AUC: 0.8430936442440073
```

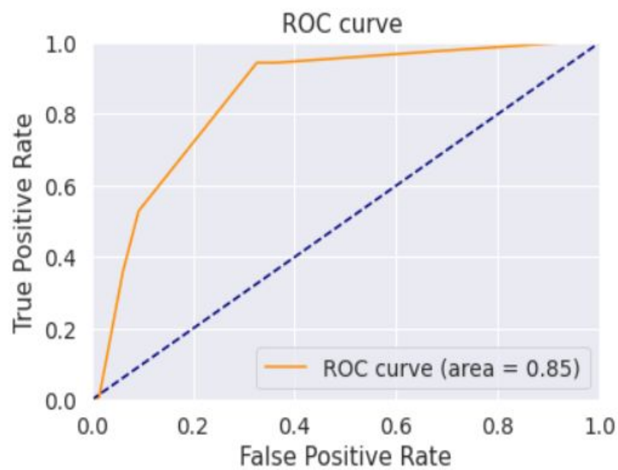


4.Decision Tree Classifier:

```
=================Decision Tree Classifier==============
Accuracy: 0.8055555555555556
```

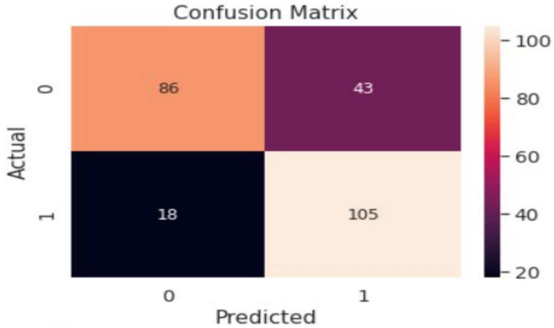



```
Specificity: 0.6744186046511628
F1-Score: 0.8256227758007118
Precision: 0.7341772151898734
Recall/Sensitivity: 0.943089430894309
AUC Score: 0.8087540177727359
Cross-validated AUC: 0.804198405850523
```

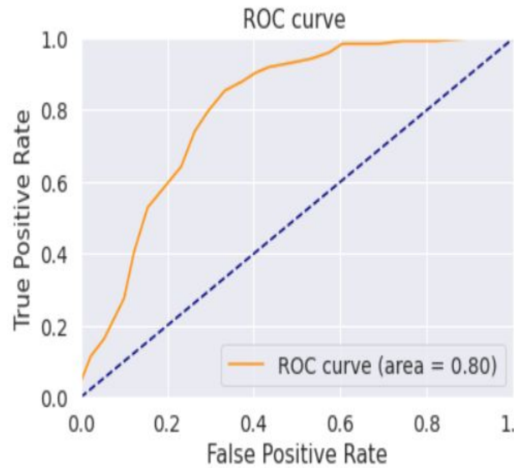


5.KNN classifier:

```
===================K Nearest Neighbours===============
Accuracy: 0.7579365079365079
```

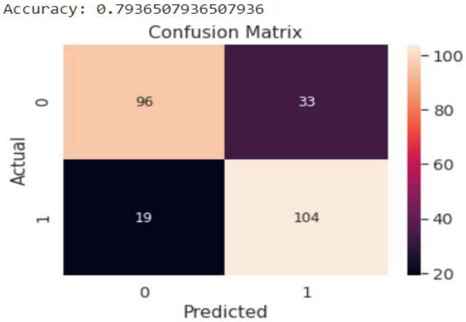



```
Specificity: 0.6666666666666666
F1-Score: 0.7749077490774907
Precision: 0.7094594594594594
Recall/Sensitivity: 0.8536585365853658
AUC Score: 0.7601626016260163
Cross-validated AUC: 0.8029599836404111
```

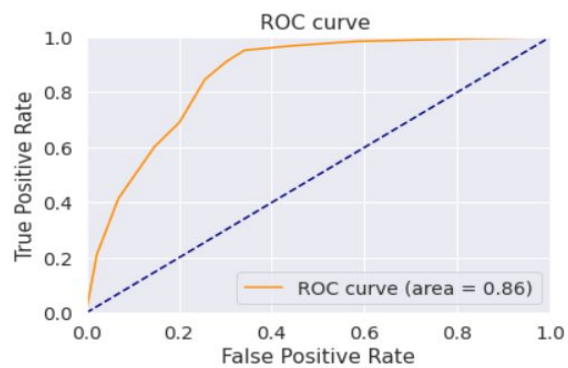


6.Bagging:

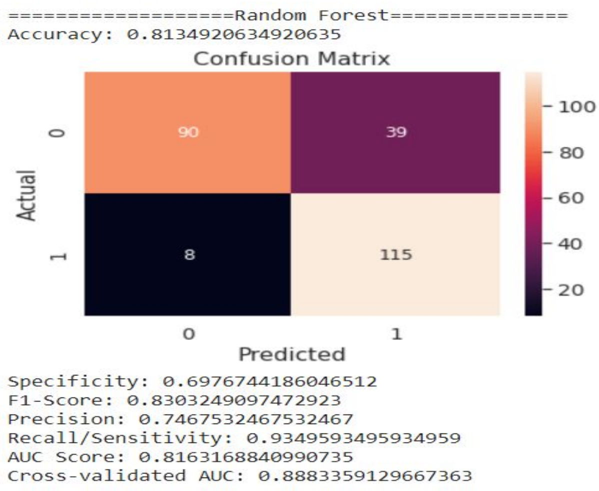
```
===================Bagging Decision Tree Classifier===============
Accuracy: 0.7936507936507936
```

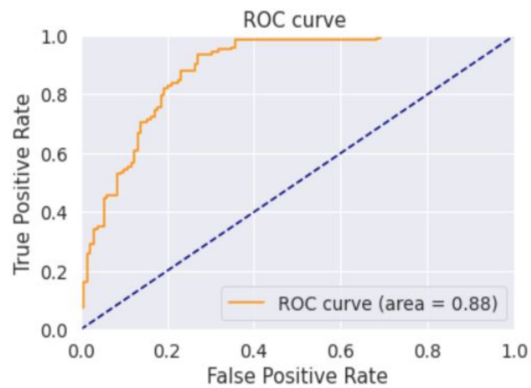



```
Specificity: 0.7441860465116279
F1-Score: 0.7999999999999999
Precision: 0.7591240875912408
Recall/Sensitivity: 0.8455284552845529
AUC Score: 0.7948572508980905
Cross-validated AUC: 0.8570599054567399
```

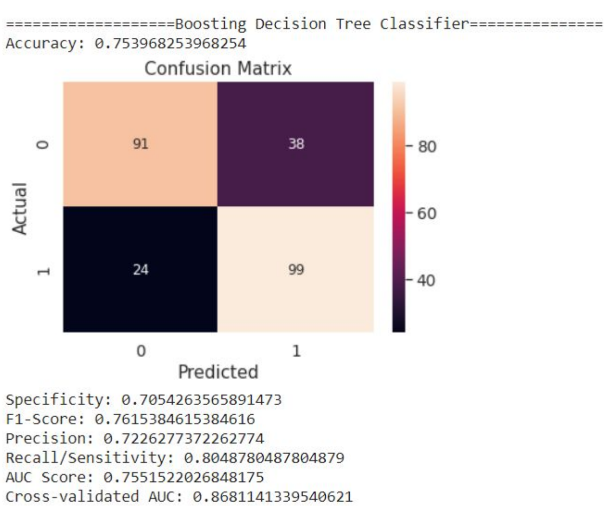


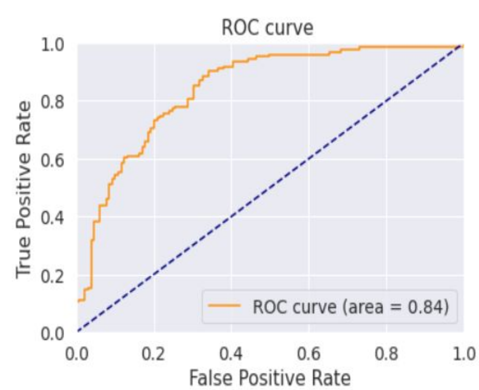
7.Random forest





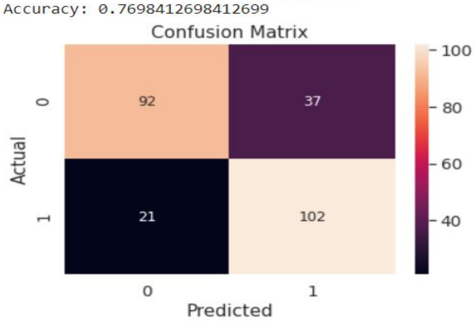
8.Boosting:

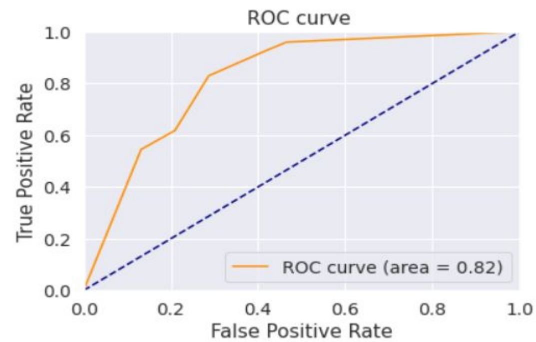




9.Stacking:

```
==================Stacking Classifier==============
Accuracy: 0.7698412698412699
```

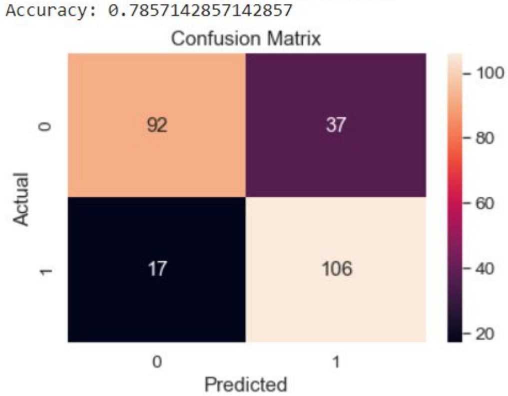





```
Specificity: 0.7131782945736435
F1-Score: 0.7786259541984732
Precision: 0.7338129496402878
Recall/Sensitivity: 0.8292682926829268
AUC Score: 0.771223293628285
Cross-validated AUC: 0.8494902638901987
```

10.DNNClassifier:

```
==================Deep Neural Network===============
Accuracy: 0.7857142857142857
```




```
Specificity: 0.7131782945736435
F1-Score: 0.7969924812030075
Precision: 0.7412587412587412
Recall/Sensitivity: 0.8617886178861789
AUC Score: 0.7874834562299111
```

Conclusion

Select k-Best has been used to extract the top 20 best features using the chi2 correlation parameter.K- Means clustering has been used in order to classify the data points into two clusters.Grid Search CV has been used to fine tune the parameters of machine learning models.The best machine learning model as per the research paper has been the Random Forest Classifier with 81% as train accuracy.The novelty introduced was the application of Deep Neural network technique which gave the test accuracy as 83%.

References:

[1].https://machinelearningmastery.com/k-fold-cross-validation/#:~:text=Cross%2Dvalidation%20is%20a%20resampling,k%2Dfold%20cross%2Dvalidation.
[2].https://www.kaggle.com/kairosart/machine-learning-for-mental-health-1/log
[3].https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
[4].https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
[5].https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm
[6].https://www.geeksforgeeks.org/ml-bagging-classifier/
[7].https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/#:~:text=Stacked%20Generalization%20or%20%E2%80%9CStacking%E2%80%9D%20for,dataset%2C%20like%20bagging%20and%20boosting.