HarvardX PH125.9x Data Science Capstone Project

Predicting-Mental-Health

Author: Bouchikhi Alaa Eddine

Date: December 2024

# Introduction

Mental health is a vital aspect of overall well-being, significantly influencing the quality of life for individuals and communities. In recent years, we have witnessed a marked increase in depression rates worldwide, highlighting the urgent need to understand the factors that contribute to its onset. Depression goes beyond mere feelings of sadness; it is a complex mental health condition that affects how individuals think and behave. If left unaddressed, it can lead to serious complications.

This study aims to explore mental health data to identify the key factors that may increase the risk of developing depression. By analyzing a comprehensive dataset derived from survey responses, we seek to uncover the relationships between various variables, such as age, gender, academic and work-related stress, sleep patterns, and dietary habits.

The primary objective of this project is to build a predictive model that can effectively identify individuals at risk of depression. Through in-depth data exploration and the application of machine learning techniques, we hope to provide valuable insights that can inform early interventions and improve treatment strategies.

The study will unfold in several phases, starting with exploratory data analysis, followed by data preparation, and culminating in the construction and evaluation of the predictive model. By taking this comprehensive approach, we aim to enhance our understanding of depression and contribute to better mental health outcomes within the community.

# 1. Data Preparation

This helps to understand the size of the dataset, which is a fundamental step in EDA.

```
> # Get data format
> data_shape <- dim(train_df)
> cat("The train_df has", data_shape[1], "rows and", data_shape[2], "columns.\n")
The train_df has 139001 rows and 20 columns.
>
> # Use glimpse to get information about columns
> glimpse(train_df)
Rows: 139,001
Columns: 20
$ id                                  <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, …
$ Name                                <chr> "Aaradhya", "Vivan", "Yuvraj", "Yuvraj", "Rhea", "Vani", "Ritvik", "Rajveer", …
$ Gender                              <chr> "Female", "Male", "Male", "Male", "Female", "Female", "Male", "Male", "Female"…
$ Age                                 <dbl> 49, 26, 33, 22, 30, 59, 47, 38, 24, 42, 55, 51, 39, 29, 50, 23, 49, 56, 50, 45…
$ City                                <chr> "Ludhiana", "Varanasi", "Visakhapatnam", "Mumbai", "Kanpur", "Ahmedabad", "Tha…
$ `Working Professional or Student`   <chr> "Working Professional", "Working Professional", "Student", "Working Profession…
$ Profession                          <chr> "Chef", "Teacher", NA, "Teacher", "Business Analyst", "Finanancial Analyst", "…
$ `Academic Pressure`                 <dbl> NA, NA, 5, NA, NA, NA, NA, NA, 2, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
$ `Work Pressure`                     <dbl> 5, 4, NA, 5, 1, 2, 5, 3, NA, 4, 3, 1, 2, 4, 1, 2, 5, 5, 5, 2, 3, 3, 5, 2, 5, 2…
$ CGPA                                <dbl> NA, NA, 8.97, NA, NA, NA, NA, NA, 5.90, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
$ `Study Satisfaction`                <dbl> NA, NA, 2, NA, NA, NA, NA, NA, 5, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
$ `Job Satisfaction`                  <dbl> 2, 3, NA, 1, 1, 5, 2, 4, NA, 1, 2, 2, 5, 4, 1, 2, 3, 3, 2, 4, 1, 1, 5, 3, 2, 1…
$ `Sleep Duration`                    <chr> "More than 8 hours", "Less than 5 hours", "5-6 hours", "Less than 5 hours", "5…
$ `Dietary Habits`                    <chr> "Healthy", "Unhealthy", "Healthy", "Moderate", "Unhealthy", "Healthy", "Modera…
$ Degree                              <chr> "BHM", "LLB", "B.Pharm", "BBA", "BBA", "MCA", "MD", "B.Pharm", "BSc", "ME", "B…
$ `Have you ever had suicidal thoughts ?` <chr> "No", "Yes", "Yes", "Yes", "Yes", "No", "No", "No", "No", "Yes", "No", "No", "…
$ `Work/Study Hours`                  <dbl> 1, 7, 3, 10, 9, 7, 6, 10, 3, 7, 6, 9, 1, 6, 8, 6, 2, 0, 3, 2, 5, 6, 9, 7, 7, 2…
$ `Financial Stress`                  <dbl> 2, 3, 1, 1, 4, 5, 2, 3, 2, 2, 4, 5, 3, 5, 5, 4, 1, 5, 2, 3, 2, 4, 2, 5, 2, 2, …
$ `Family History of Mental Illness`  <chr> "No", "No", "No", "Yes", "Yes", "No", "No", "Yes", "Yes", "Yes", "Yes", "No", …
$ Depression                          <dbl> 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
>
```

This is also a key aspect of EDA, as it helps identify trends, patterns, and potential issues in the data.

```
> # Optionally, you can summarize the dataset
> summary(train_df)
       id             Name              Gender               Age             City           Working Professional or Student
 Min.   :     0   Length:139001      Length:139001      Min.   :18.00   Length:139001      Length:139001
 1st Qu.: 34750   Class :character   Class :character   1st Qu.:29.00   Class :character   Class :character
 Median : 69500   Mode  :character   Mode  :character   Median :42.00   Mode  :character   Mode  :character
 Mean   : 69500                                        Mean   :40.39
 3rd Qu.:104250                                        3rd Qu.:51.00
 Max.   :139000                                        Max.   :60.00

  Profession        Academic Pressure Work Pressure       CGPA        Study Satisfaction Job Satisfaction Sleep Duration
 Length:139001      Min.   :1.00      Min.   :1.000   Min.   : 5.03   Min.   :1.00       Min.   :1.000    Length:139001
 Class :character   1st Qu.:2.00      1st Qu.:2.000   1st Qu.: 6.29   1st Qu.:2.00       1st Qu.:2.000    Class :character
 Mode  :character   Median :3.00      Median :3.000   Median : 7.77   Median :3.00       Median :3.000    Mode  :character
                    Mean   :3.14      Mean   :2.999   Mean   : 7.66   Mean   :2.94       Mean   :2.974
                    3rd Qu.:4.00      3rd Qu.:4.000   3rd Qu.: 8.92   3rd Qu.:4.00       3rd Qu.:4.000
                    Max.   :5.00      Max.   :5.000   Max.   :10.00   Max.   :5.00       Max.   :5.000
                    NA's   :111432    NA's   :27590   NA's   :111431  NA's   :111432     NA's   :27582
 Dietary Habits        Degree          Have you ever had suicidal thoughts ? Work/Study Hours Financial Stress
 Length:139001      Length:139001      Length:139001                         Min.   : 0.000   Min.   :1.00
 Class :character   Class :character   Class :character                      1st Qu.: 3.000   1st Qu.:2.00
 Mode  :character   Mode  :character   Mode  :character                      Median : 6.000   Median :3.00
                                                                            Mean   : 6.251   Mean   :2.99
                                                                            3rd Qu.:10.000   3rd Qu.:4.00
                                                                            Max.   :12.000   Max.   :5.00
                                                                                             NA's   :4

 Family History of Mental Illness   Depression
 Length:139001                      Min.   :0.0000
 Class :character                   1st Qu.:0.0000
 Mode  :character                   Median :0.0000
                                    Mean   :0.1817
                                    3rd Qu.:0.0000
                                    Max.   :1.0000
```

# Exploratory

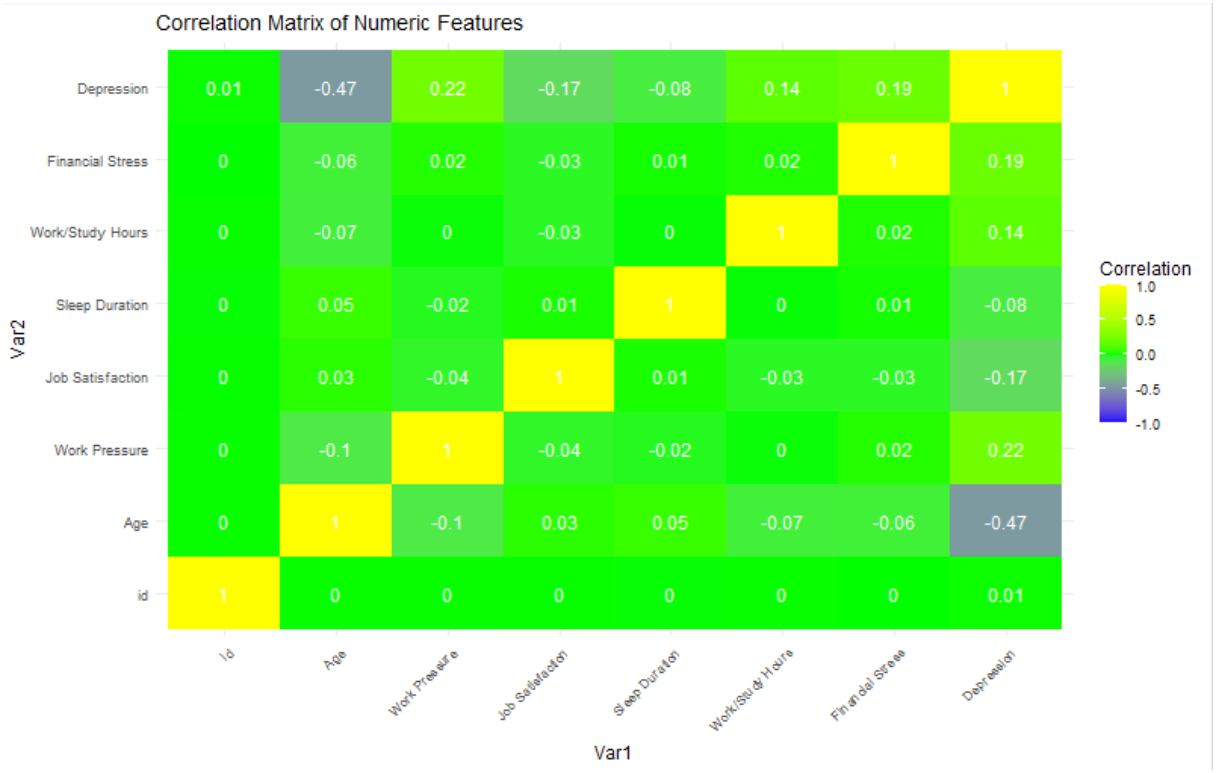
## Descriptive Statistics

We will now examine the summary statistics for both numerical and categorical data. This will help us understand the basic properties of the dataset and provide insights into the data's distribution.

The correlation matrix shows the relationships between the numeric variables in your dataset. The values range from -1 to 1:

- **Value of 1**: Indicates a perfect positive correlation between the variables, meaning an increase in one variable leads to an increase in the other.
- **Value of -1**: Indicates a perfect negative correlation, meaning an increase in one variable leads to a decrease in the other.
- **Value of 0**: Indicates no correlation between the variables.

```
> # Check if there are any remaining numeric columns
> if (ncol(numeric_cols_train) > 0) {
+     # Calculate the correlation matrix
+     correlation_matrix_train <- cor(numeric_cols_train, use = "complete.obs")
+
+     # Print the correlation matrix
+     print(correlation_matrix_train)
+ } else {
+     print("No valid numeric columns available for correlation analysis.")
+ }
                          id          Age Work Pressure Job Satisfaction Sleep Duration Work/Study Hours Financial Stress
id               1.0000000000  0.003182059   0.003008185      0.002252326   -0.002052105      0.003302584     0.0003355369
Age              0.0031820585  1.000000000  -0.098531850      0.030884963    0.053585589     -0.067579875    -0.0635541941
Work Pressure    0.0030081853 -0.098531850   1.000000000     -0.036941444   -0.021709396     -0.003319424     0.0242904064
Job Satisfaction 0.0022523262  0.030884963  -0.036941444      1.000000000    0.014546921     -0.028557811    -0.0294473020
Sleep Duration  -0.0020521051  0.053585589  -0.021709396      0.014546921    1.000000000     -0.002166086     0.0100035679
Work/Study Hours 0.0033025840 -0.067579875  -0.003319424     -0.028557811   -0.002166086      1.000000000     0.0193671397
Financial Stress 0.0003355369 -0.063554194   0.024290406     -0.029447302    0.010003568      0.019367140     1.0000000000
Depression       0.0050291774 -0.470103387   0.216379061     -0.168630837   -0.080997787      0.140635543     0.1869540357
                  Depression
id               0.005029177
Age             -0.470103387
Work Pressure    0.216379061
Job Satisfaction -0.168630837
Sleep Duration   -0.080997787
Work/Study Hours  0.140635543
Financial Stress  0.186954036
Depression        1.000000000
```




**Key Correlations with Depression**

The correlation matrix highlights important relationships among various factors impacting mental health:

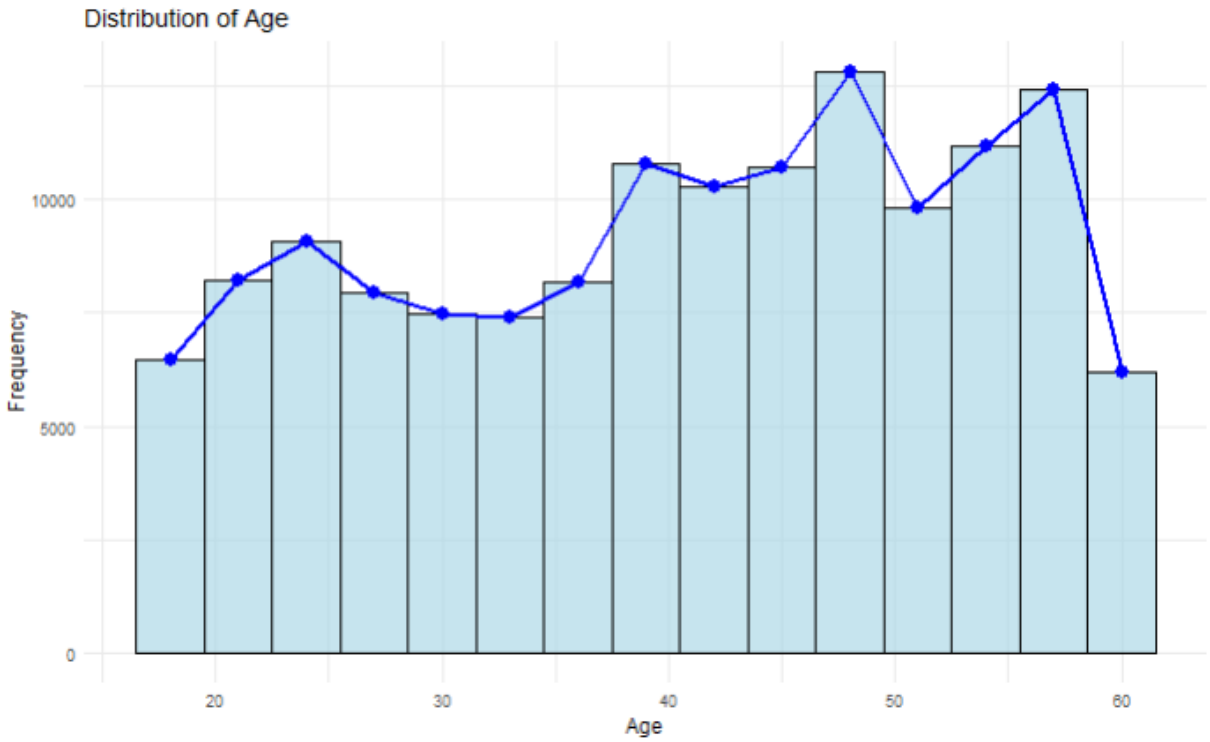
1. **Age** has a notable negative correlation with depression (-0.470), suggesting that older individuals generally experience lower levels of depression.
2. **Work Pressure** is positively correlated with depression (0.216), indicating that higher levels of work pressure are associated with increased depression, while it negatively affects job satisfaction.
3. **Job Satisfaction** is negatively correlated with depression (-0.169), meaning higher satisfaction is linked to lower depression levels.
4. **Financial Stress** also shows a positive correlation with depression (0.187), indicating that increased financial stress heightens depressive symptoms.
5. **Sleep Duration** shows weak correlations but may still play a role in overall mental health.
6. Addressing work pressure, financial stress, and enhancing job satisfaction could significantly improve mental well-being.

the p-values you obtained from the Chi-square tests for each categorical variable in relation to depression:

```
> # Load necessary library
> library(dplyr)
>
> # Updated list of categorical columns to check
> categorical_cols <- c('Gender', 'City', 'Working Professional or Student',
+                       'Profession', 'Dietary Habits', 'Degree',
+                       'Have you ever had suicidal thoughts ?', 'Family History of Mental Illness')
>
> # Check for missing columns
> missing_columns <- setdiff(categorical_cols, colnames(train_df))
> if (length(missing_columns) > 0) {
+     cat("The following columns are missing in train_df:\n")
+     print(missing_columns)
+ }
>
> # Loop through each categorical column to run chi-square test
> for (col in categorical_cols) {
+     if (col %in% colnames(train_df)) {
+         # Filter out rows with NA values in the current column or the Depression column
+         filtered_data <- train_df %>%
+             filter(!is.na(get(col)), !is.na(Depression))
+
+         # Create contingency table
+         contingency_table <- table(filtered_data[[col]], filtered_data$Depression)
+
+         # Perform chi-square test if contingency_table has more than one level
+         if (nrow(contingency_table) > 1 && ncol(contingency_table) > 1) {
+             chi_square_result <- chisq.test(contingency_table)
+
+             # Print p-value
+             cat(paste(col, ": p-value =", chi_square_result$p.value), "\n")
+         } else {
+             cat(paste(col, ": Not enough levels for Chi-square test"), "\n")
+         }
+     } else {
+         cat(paste(col, ": Column not found in train_df"), "\n")
+     }
+ }
Gender : p-value = 0.00422266013082821
City : p-value = 2.3222375818833e-173
Working Professional or Student : p-value = 0
Profession : p-value = 1.13927838796289e-251
Dietary Habits : p-value = 0
Degree : p-value = 0
Have you ever had suicidal thoughts ? : p-value = 0
Family History of Mental Illness : p-value = 7.89344149900768e-10
>
```

In summary, all the categorical variables tested show significant associations with depression, as indicated by their p-values.

## 1. Distribution of Age

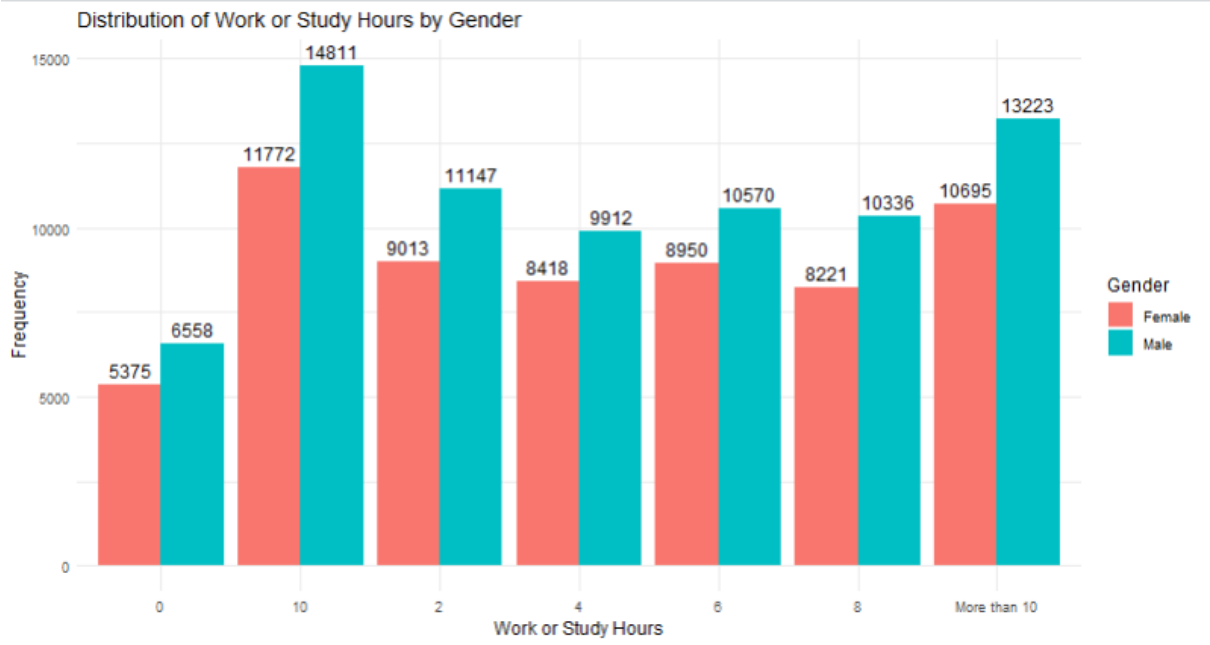


## 2. Distribution of Financial_Stress


### 1. Data Values :

- These numbers reflect the categories or numerical values representing various levels of financial stress. For Example :
    - 1 might represent very low financial stress.
    - 2 could represent low financial stress.
    - 3 signifies moderate financial stress.
    - 4 might indicate high financial stress.
    - 5 represents very high financial stress.

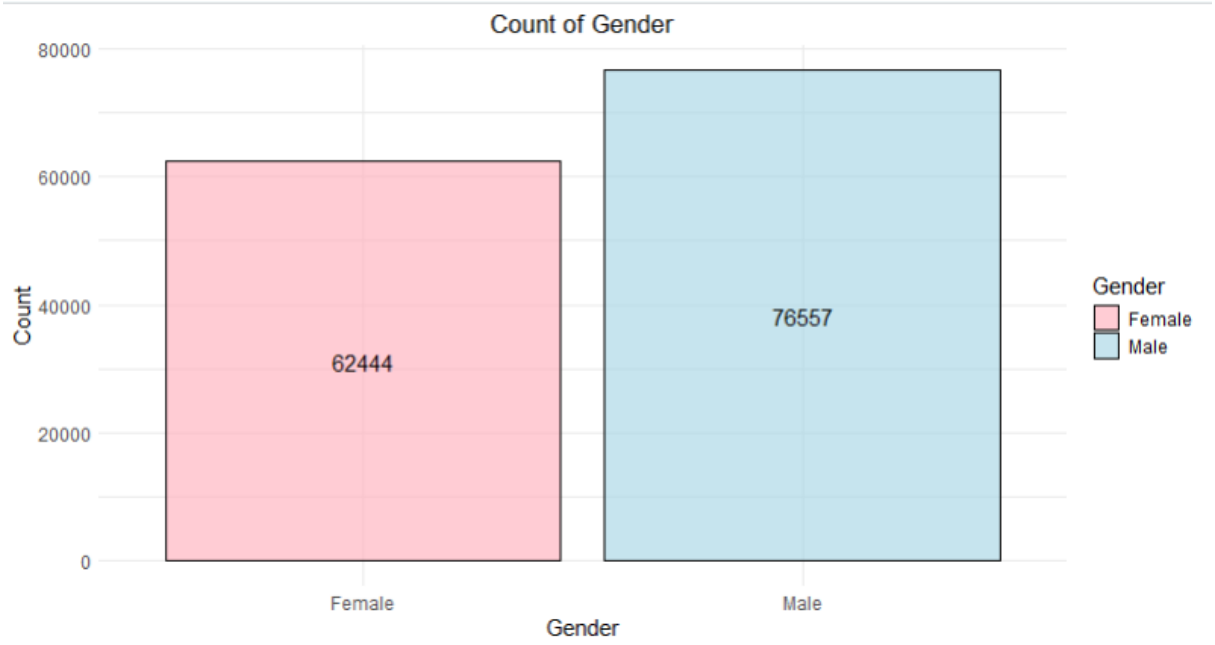
### 3. Distribution of Work_or_Study_Hours




**Interpretation of the Numbers**

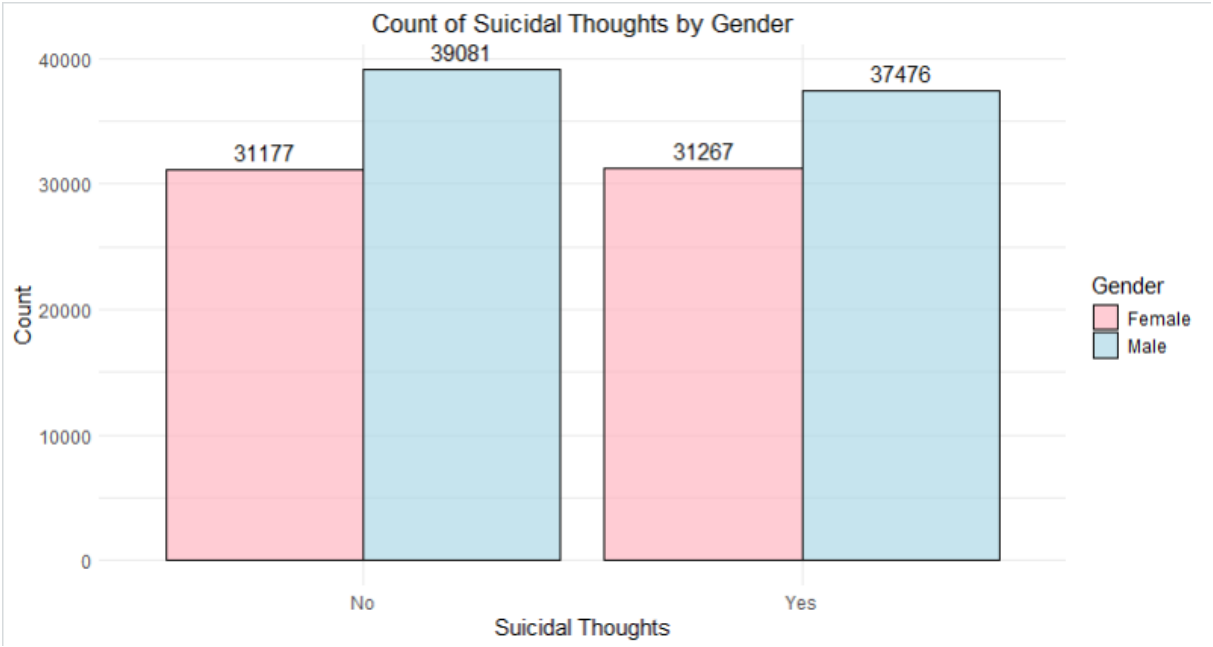
1. **0**: Represents individuals who do not engage in work or study hours, indicating no time spent in these activities.
2. **2**: Indicates individuals who spend 2 hours on work or study.
3. **4**: Represents individuals who spend 4 hours on work or study.
4. **6**: Indicates individuals who spend 6 hours on work or study.
5. **8**: Represents individuals who spend 8 hours on work or study.
6. **10**: Indicates individuals who spend 10 hours on work or study.

**4-Count plot for Gender**

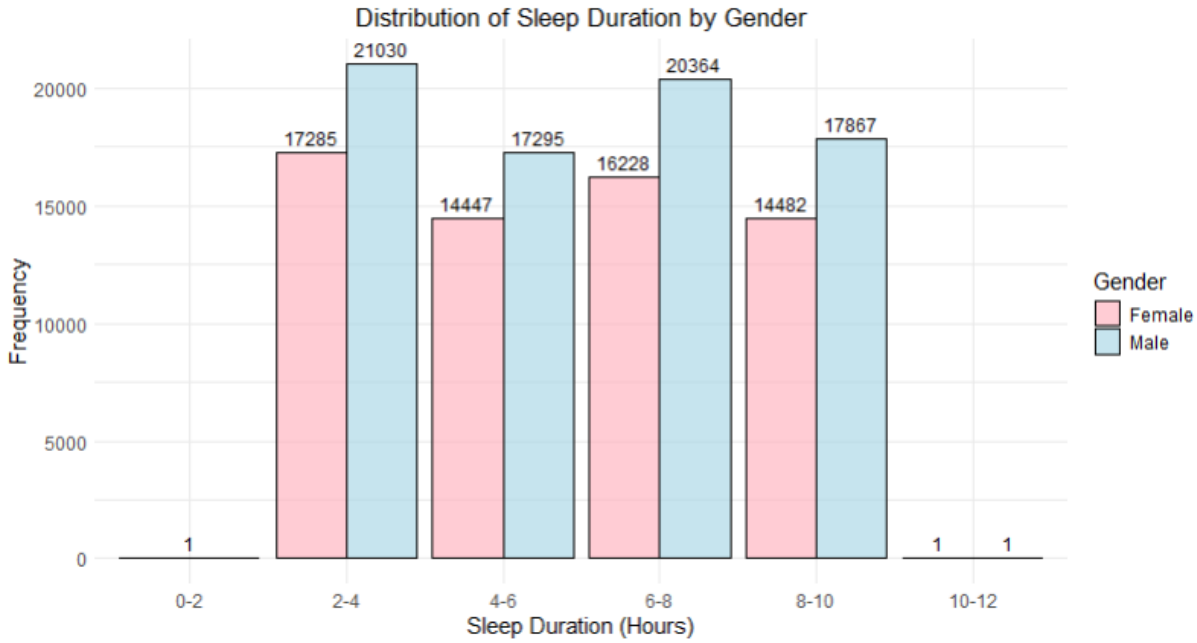



**5-Count plot for Have_you_ever_had_suicidal_thoughts**

.



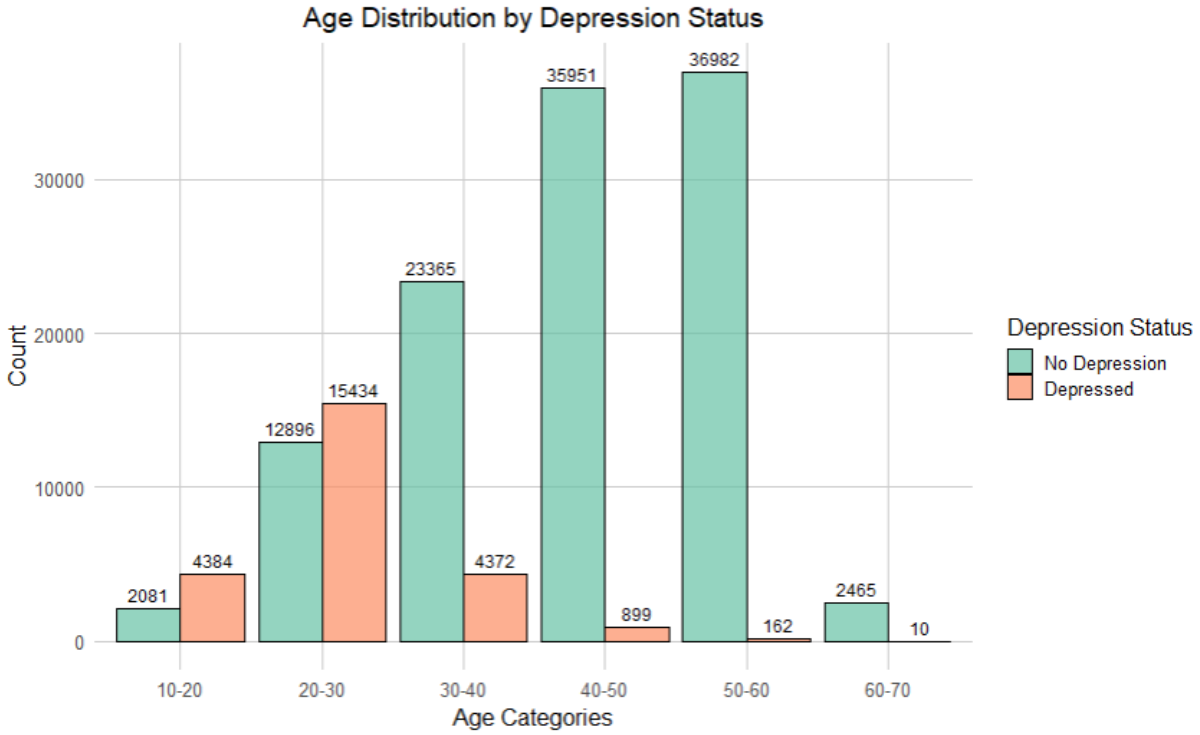
**-Distribution of Sleep_Duration**



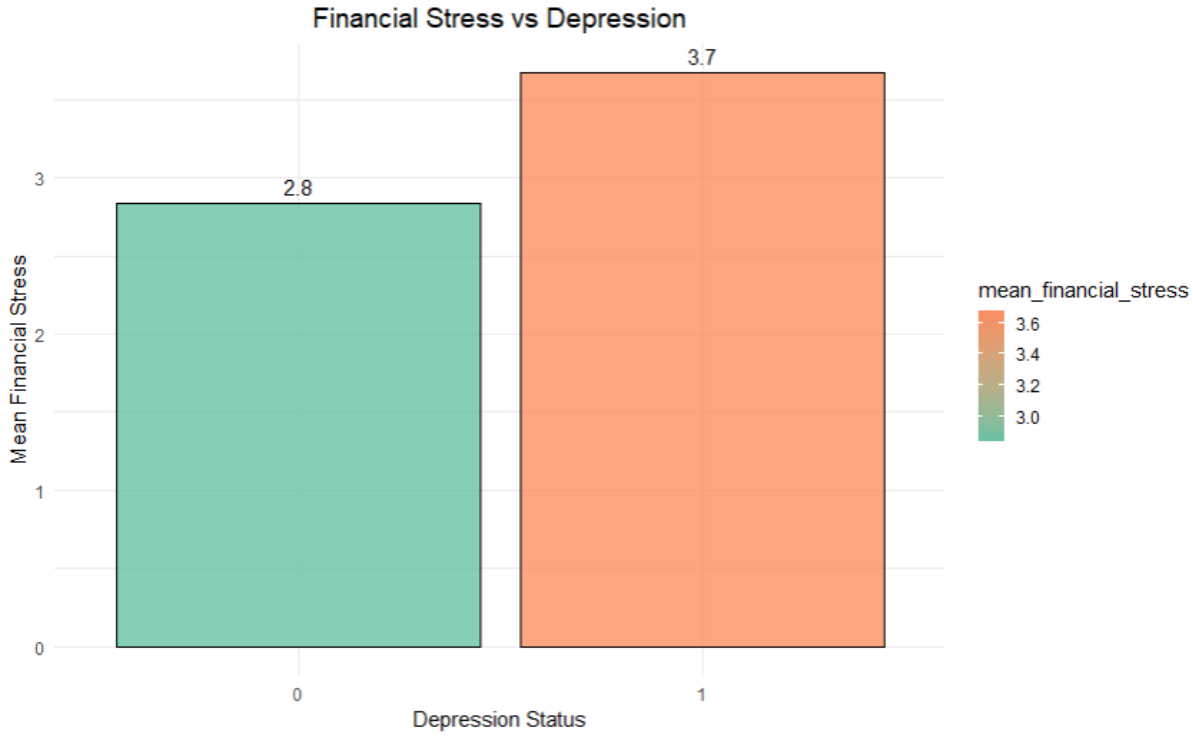

**Multivariate Exploration of Depression**

In this section, we will conduct a multivariate analysis to explore the relationships between various features in our dataset and the target variable, Depression. Understanding these relationships is crucial for identifying potential predictors of depression and gaining insights into how different factors may interact with one another.

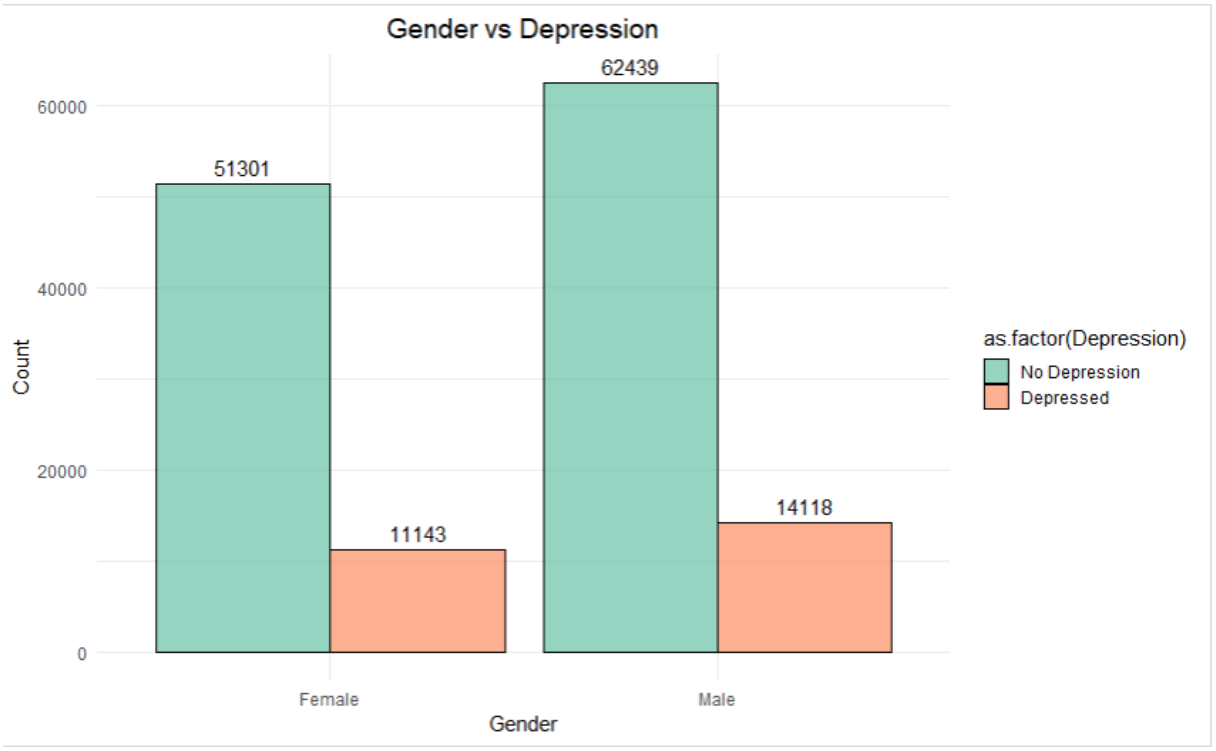
1. **Age vs Depression**




2. Financial_Stress vs Depression

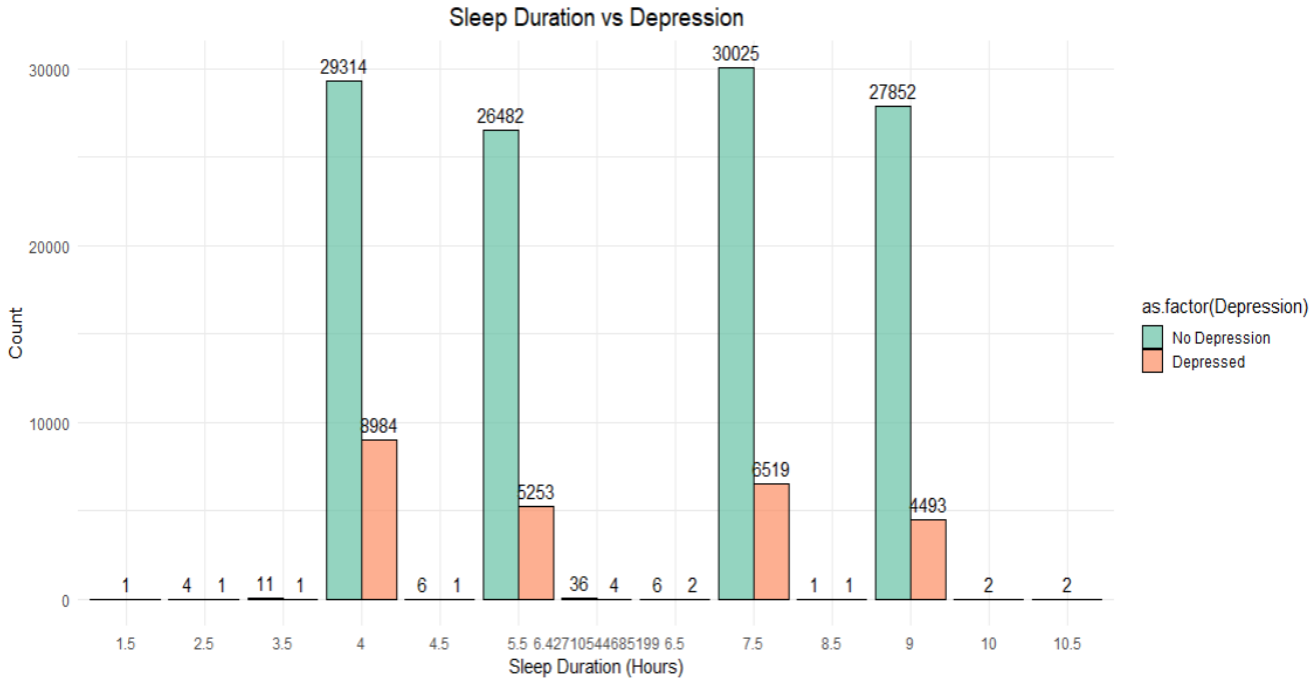


3. Gender vs Depression



4. Work_or_Study_Hours vs Depression


5. Sleep_Duration' vs Depression`



## Data Prep: Preprocessing

In data preparation, preprocessing is a crucial step to ensure high-quality results. It involves standardizing numeric features to create a uniform scale, which can significantly enhance model performance. Techniques like Z-score normalization or Min-Max scaling are commonly used to transform data, ensuring that all features contribute equally during model training.

```
>
> # Standardize numeric features for training data
> df_scaled_train <- train_df %>%
+     select(all_of(numeric_features)) %>%
+     scale() %>%
+     as.data.frame()
>
> # Normalize the standardized data for training
> df_normalized_train <- as.data.frame(lapply(df_scaled_train, function(x) (x - min(x)) / (max(x) - min(x))))
>
> # Standardize numeric features for test data
> df_scaled_test <- test_df %>%
+     select(all_of(numeric_features)) %>%
+     scale() %>%
+     as.data.frame()
>
> # Normalize the standardized data for test
> df_normalized_test <- as.data.frame(lapply(df_scaled_test, function(x) (x - min(x)) / (max(x) - min(x))))
>
> # Display the first few rows of the normalized data
> head(df_normalized_train)
        Age Work.Pressure Job.Satisfaction Sleep.Duration Work.Study.Hours
1 0.7380952     1.0000000        0.2500000      0.8333333       0.08333333
2 0.1904762     0.7500000        0.5000000      0.2777778       0.58333333
3 0.3571429     0.4997016        0.4934728      0.4444444       0.25000000
4 0.0952381     1.0000000        0.0000000      0.2777778       0.83333333
5 0.2857143     0.0000000        0.0000000      0.4444444       0.75000000
6 0.9761905     0.2500000        1.0000000      0.4444444       0.58333333
  Financial.Stress
1             0.25
2             0.50
3             0.00
4             0.00
5             0.75
6             1.00
> head(df_normalized_test)
        Age Work.Pressure Job.Satisfaction Sleep.Duration Work.Study.Hours
1 0.1190476     0.5033023        0.4905501      0.8823529       0.66666667
2 0.8571429     0.2500000        0.5000000      0.2941176       0.33333333
3 0.6666667     0.7500000        0.5000000      0.8823529       0.33333333
4 0.5714286     1.0000000        0.0000000      0.4705882       0.00000000
5 0.7142857     1.0000000        0.2500000      0.7058824       0.91666667
6 0.7142857     1.0000000        1.0000000      0.8823529       0.08333333
  Financial.Stress
1             0.75
2             0.25
3             1.00
4             0.75
5             0.50
6             0.00
>
```

## Summary of Normalized Data

The normalized training dataset (df_normalized_train) and test dataset (df_normalized_test) contain standardized numeric features, ensuring that all values are scaled between 0 and 1.

- **Training Data**: The Age, Work Pressure, Job Satisfaction, Sleep Duration, Work/Study Hours, and Financial Stress columns reflect the normalized values, with a clear distribution across different ranges.
- **Test Data**: Similar to the training data, the test dataset presents the same features, allowing for consistent comparisons and analyses.

This normalization process enhances model training by ensuring that all features contribute equally, improving the overall predictive performance.

## Convert Categorical Data into Numerical

To prepare our data for machine learning models, we need to convert categorical variables into numeric format, as most models require numerical inputs.

```
> # Load necessary libraries
> library(dplyr)
> library(caret)
>
> # Define categorical columns
> categorical_columns <- c('Gender', 'Working Professional or Student', 'Have you ever had suicidal thoughts ?', 'Dietary Habit
s', 'Family History of Mental Illness')
>
> # Convert categorical columns to factors (Label Encoding for binary and preparation for One-hot Encoding)
> train_df <- train_df %>%
+     mutate(across(all_of(categorical_columns), as.factor))
>
> test_df <- test_df %>%
+     mutate(across(all_of(categorical_columns), as.factor))
>
> # One-hot Encoding
> dummies_train <- dummyVars(~ ., data = train_df %>% select(all_of(categorical_columns)))
> train_df <- cbind(train_df %>% select(-all_of(categorical_columns)),
+                   as.data.frame(predict(dummies_train, newdata = train_df)))
>
> dummies_test <- dummyVars(~ ., data = test_df %>% select(all_of(categorical_columns)))
> test_df <- cbind(test_df %>% select(-all_of(categorical_columns)),
+                  as.data.frame(predict(dummies_test, newdata = test_df)))
>
> # Drop the first level to avoid multicollinearity (equivalent to drop_first=True in Python)
> train_df <- train_df %>% select(-matches("\\.1$"))
> test_df <- test_df %>% select(-matches("\\.1$"))
>
> # Display the head of the transformed dataframes
> print(head(train_df))
```

```
> # Display the head of the transformed dataframes
> print(head(train_df))
  id     Name Age Work Pressure Job Satisfaction Sleep Duration Work/Study Hours Financial Stress
1  0 Aaradhya  49      5.000000         2.000000            9.0                1                2
2  1    Vivan  26      4.000000         3.000000            4.0                7                3
3  2   Yuvraj  33      2.998806         2.973891            5.5                3                1
4  3   Yuvraj  22      5.000000         1.000000            4.0               10                1
5  4     Rhea  30      1.000000         1.000000            5.5                9                4
6  5     Vani  59      2.000000         5.000000            5.5                7                5
  Depression Gender.Female Gender.Male `Working Professional or Student`Student
1          0             1           0                                        0
2          1             0           1                                        0
3          1             0           1                                        1
4          1             0           1                                        0
5          0             1           0                                        0
6          0             1           0                                        0
  `Working Professional or Student`Working Professional `Have you ever had suicidal thoughts ?`No
1                                                      1                                          1
2                                                      1                                          0
3                                                      0                                          0
4                                                      1                                          0
5                                                      1                                          0
6                                                      1                                          1
  `Have you ever had suicidal thoughts ?`Yes `Dietary Habits`Healthy `Dietary Habits`Moderate
1                                          0                        1                       0
2                                          1                        0                       0
3                                          1                        1                       0
4                                          1                        0                       1
5                                          1                        0                       0
6                                          0                        1                       0
  `Dietary Habits`Unhealthy `Family History of Mental Illness`No
1                         0                                    1
2                         1                                    1
3                         0                                    1
4                         0                                    0
5                         1                                    0
6                         0                                    1
  `Family History of Mental Illness`Yes
1                                     0
2                                     0
3                                     0
4                                     1
5                                     1
6                                     0
> print(head(test_df))
      id   Name Age Work Pressure Job Satisfaction Sleep Duration Work/Study Hours
1 139001   Shlok  23      3.013209           2.9622            9.0                8
2 139002    Tara  54      2.000000           3.0000            4.0                4
3 139003   Darsh  46      4.000000           3.0000            9.0                4
4 139004   Vidya  42      5.000000           1.0000            5.5                0
5 139005 Aarohi  48      5.000000           2.0000            7.5               11
6 139006    Asha  48      5.000000           5.0000            9.0                1
```

**Results Explanation**

The preprocessing workflow applied to the dataset has resulted in a well-structured format that enhances the ability of machine learning models to predict depression. Each categorical variable has been transformed into binary features through one-hot encoding, allowing the models to interpret them effectively. This transformation ensures that the relationships between these categorical factors and the target variable (depression) can be captured accurately.

The training and test datasets now contain a variety of features that represent key demographic and behavioral factors, such as gender, employment status, suicidal thoughts, dietary habits, and family history of mental illness. For instance, the presence of a family history of mental illness may provide significant predictive power, as it often correlates with higher risks of depression. Similarly, dietary habits can influence mental health, making these features crucial for model training.

By avoiding multicollinearity ensuring that no two features convey redundant information the models can learn more efficiently, leading to better performance. The structured datasets are now primed for training machine learning algorithms, which will utilize these features to identify patterns and make predictions regarding depression.

**Conclusion**

This preprocessing workflow prepares the dataset for predictive modeling by ensuring that categorical variables are appropriately formatted and that multicollinearity is avoided. The resulting datasets are now ready for training machine learning models, which can leverage these features to predict depression effectively. Proper data preprocessing is vital for achieving high model accuracy and reliability in predictions.

## Prediction

The objective is to develop a reliable classification model that can accurately predict whether an individual is experiencing depression. By leveraging a substantial training dataset, we aim to ensure the model learns effectively from diverse examples. Once trained, we will evaluate its performance using a separate validation set to gauge its effectiveness in real-world scenarios.

```
> 
> # Load necessary libraries
> library(dplyr)
> library(caret)
> 
> # Selecting features and target variable
> X_train <- train_df %>%
+     select(-id, -Name, -Depression)
> 
> y_train <- train_df$Depression
> 
> # Split the training data into training and validation sets (80-20 split)
> set.seed(42)  # For reproducibility
> train_index <- createDataPartition(y_train, p = 0.8, list = FALSE)
> 
> X_train_split <- X_train[train_index, ]
> X_val_split <- X_train[-train_index, ]
> y_train_split <- y_train[train_index]
> y_val_split <- y_train[-train_index]
> 
> # Display the dimensions of the split datasets to verify
> print(dim(X_train_split))
[1] 111201     17
> print(dim(X_val_split))
[1] 27800    17
> print(length(y_train_split))
[1] 111201
> print(length(y_val_split))
[1] 27800
```

In summary, the **80-20** split for training and validation sets is a widely accepted practice in machine learning. This approach provides a substantial amount of data for training, allowing the model to learn effectively from diverse patterns and relationships within the dataset. By reserving **20%** of the data for validation, the model's performance can be rigorously evaluated on unseen data, ensuring it generalizes well beyond the training set. This balance helps prevent overfitting, where the model memorizes the training data instead of learning to make predictions. Ultimately, this methodology enhances the model's robustness and reliability, leading to improved performance in real-world applications.

# Model Building

## 1. Random Forest Classifier

```
> library(e1071)  # For confusionMatrix
> library(pROC)   # For ROC-AUC calculation
> library(ggplot2) # For plotting
>
> # Build the Random Forest model
> set.seed(42)  # For reproducibility
> rf_model <- randomForest(x = X_train_split, y = as.factor(y_train_split), ntree = 100)
>
> # Make predictions on the validation set
> y_val_pred_rf <- predict(rf_model, X_val_split, type = "response")
>
> # Print classification report
> conf_matrix <- confusionMatrix(y_val_pred_rf, as.factor(y_val_split))
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21865  1073
         1   890  3972

               Accuracy : 0.9294
                 95% CI : (0.9263, 0.9324)
    No Information Rate : 0.8185
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7589

 Mcnemar's Test P-Value : 3.994e-05

            Sensitivity : 0.9609
            Specificity : 0.7873
         Pos Pred Value : 0.9532
         Neg Pred Value : 0.8169
             Prevalence : 0.8185
         Detection Rate : 0.7865
   Detection Prevalence : 0.8251
      Balanced Accuracy : 0.8741

       'Positive' Class : 0

>
> # Calculate ROC-AUC score
> y_pred_prob <- predict(rf_model, X_val_split, type = "prob")[,2]
> roc_auc_rf <- roc(as.numeric(y_val_split) - 1, y_pred_prob)$auc
Setting levels: control = -1, case = 0
Setting direction: controls < cases
> print(paste("Random Forest - ROC-AUC Score:", roc_auc_rf))
[1] "Random Forest - ROC-AUC Score: 0.960090323106108"
```

The Random Forest Classifier demonstrated strong performance, achieving an accuracy of 92.94% and a Kappa statistic of 0.7589, indicating substantial agreement beyond chance. The confusion matrix reveals high sensitivity (96.09%) and a balanced accuracy of 87.41%, highlighting the model's effectiveness in identifying both positive and negative cases. Additionally, the ROC-AUC score of 0.9601 signifies excellent discrimination ability between the classes.

## 2-Gradient Boosting Machine (GBM)

```
>
> # Make predictions on the validation set
> y_val_pred_gbm <- predict(gbm_model, newdata = X_val_split, n.trees = gbm.perf(gbm_model, method = "cv"), type =
"response")
>
> # Convert probabilities to class labels
> y_val_pred_gbm_class <- ifelse(y_val_pred_gbm > 0.5, 1, 0)
>
> # Print classification report
> conf_matrix <- confusionMatrix(as.factor(y_val_pred_gbm_class), as.factor(y_val_split))
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21945  1111
         1   810  3934

               Accuracy : 0.9309
                 95% CI : (0.9279, 0.9339)
    No Information Rate : 0.8185
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7619

 Mcnemar's Test P-Value : 7.661e-12

            Sensitivity : 0.9644
            Specificity : 0.7798
         Pos Pred Value : 0.9518
         Neg Pred Value : 0.8293
             Prevalence : 0.8185
         Detection Rate : 0.7894
   Detection Prevalence : 0.8294
      Balanced Accuracy : 0.8721

       'Positive' Class : 0

>
> # Calculate ROC-AUC score
> roc_auc_gbm <- roc(as.numeric(y_val_split) - 1, y_val_pred_gbm)$auc
Setting levels: control = -1, case = 0
Setting direction: controls < cases
> print(paste("GBM - ROC-AUC Score:", roc_auc_gbm))
[1] "GBM - ROC-AUC Score: 0.968420689296224"
>
`
```

The Gradient Boosting Machine (GBM) model exhibited impressive performance, achieving an accuracy of 93.09% and a Kappa statistic of 0.7619, indicating strong agreement beyond chance. The confusion matrix shows excellent sensitivity at 96.44%, coupled with a balanced accuracy of 87.21%, reflecting effective classification across both classes. Additionally, the ROC-AUC score of 0.9684 highlights the model's superior ability to distinguish between the positive and negative cases.

## 3- Logistic Regression

```
>
> # Build the Logistic Regression model
> lr_model <- glm(y_train_split ~ ., data = X_train_split, family = binomial)
>
> # Make predictions on the validation set
> y_val_pred_lr <- predict(lr_model, newdata = X_val_split, type = "response")
>
> # Convert probabilities to class labels
> y_val_pred_lr_class <- ifelse(y_val_pred_lr > 0.5, 1, 0)
>
> # Print classification report
> conf_matrix <- confusionMatrix(as.factor(y_val_pred_lr_class), as.factor(y_val_split))
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21888  1078
         1   867  3967

               Accuracy : 0.93
                 95% CI : (0.927, 0.933)
    No Information Rate : 0.8185
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7606

 Mcnemar's Test P-Value : 1.92e-06

            Sensitivity : 0.9619
            Specificity : 0.7863
         Pos Pred Value : 0.9531
         Neg Pred Value : 0.8206
             Prevalence : 0.8185
         Detection Rate : 0.7873
   Detection Prevalence : 0.8261
      Balanced Accuracy : 0.8741

       'Positive' Class : 0

>
> # Calculate ROC-AUC score
> roc_obj <- roc(as.numeric(y_val_split) - 1, y_val_pred_lr)
Setting levels: control = -1, case = 0
Setting direction: controls < cases
> roc_auc_lr <- roc_obj$auc
> print(paste("Logistic Regression - ROC-AUC Score:", roc_auc_lr))
[1] "Logistic Regression - ROC-AUC Score: 0.968594175165763"
```

The Logistic Regression model achieved an accuracy of 93% and a Kappa statistic of 0.7606, indicating a strong level of agreement beyond chance. The confusion matrix reveals a sensitivity of 96.19% and a balanced accuracy of 87.41%, demonstrating effective classification of both classes. Furthermore, the ROC-AUC score of 0.9686 signifies excellent discrimination ability, comparable to other models tested.

## 4- XGBoost

```
> xgb_model <- xgb.train(params = xgb_params, data = dtrain, nrounds = 100)
>
> # Make predictions on the validation set
> y_val_pred_xgb <- predict(xgb_model, dval)
>
> # Convert probabilities to class labels
> y_val_pred_xgb_class <- ifelse(y_val_pred_xgb > 0.5, 1, 0)
>
> # Print classification report
> conf_matrix <- confusionMatrix(as.factor(y_val_pred_xgb_class), as.factor(y_val_split))
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21882  1004
         1   873  4041

               Accuracy : 0.9325
                 95% CI : (0.9295, 0.9354)
    No Information Rate : 0.8185
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7704

 Mcnemar's Test P-Value : 0.002694

            Sensitivity : 0.9616
            Specificity : 0.8010
         Pos Pred Value : 0.9561
         Neg Pred Value : 0.8223
             Prevalence : 0.8185
         Detection Rate : 0.7871
   Detection Prevalence : 0.8232
      Balanced Accuracy : 0.8813

       'Positive' Class : 0

>
> # Calculate ROC-AUC score
> roc_auc_xgb <- roc(as.numeric(y_val_split) - 1, y_val_pred_xgb)$auc
Setting levels: control = -1, case = 0
Setting direction: controls < cases
> print(paste("xgboost - ROC-AUC Score:", roc_auc_xgb))
[1] "xgboost - ROC-AUC Score: 0.969379778869977"
>
```

The XGBoost model demonstrated strong performance with an accuracy of 93.25% and a Kappa statistic of 0.7704, indicating a high level of agreement beyond chance. The confusion matrix indicates a sensitivity of 96.16% and a balanced accuracy of 88.13%, highlighting its effectiveness in classifying both positive and negative cases. Additionally, the ROC-AUC score of 0.9694 reflects excellent discrimination capability, making it one of the top-performing models in this analysis.

## Summary and Comparison of Model Results

Four different models were evaluated for predicting depression: Random Forest, Gradient Boosting Machine (GBM), Logistic Regression, and XGBoost. All models demonstrated good performance, but there were notable differences in their results:

## Random Forest:

Accuracy: 92.94%

Kappa: 0.7589

Sensitivity: 96.09%

ROC-AUC: 0.9601

## Gradient Boosting Machine (GBM):

Accuracy: 93.09%

Kappa: 0.7619

Sensitivity: 96.44%

ROC-AUC: 0.9684

## Logistic Regression:

Accuracy: 93.00%

Kappa: 0.7606

Sensitivity: 96.19%

ROC-AUC: 0.9686

## XGBoost:

Accuracy: 93.25%

Kappa: 0.7704

Sensitivity: 96.16%

ROC-AUC: 0.9694

## Comparison of Results

In predicting depression, four models were evaluated: **Random Forest**, **Gradient Boosting Machine (GBM)**, **Logistic Regression**, and **XGBoost**. All models performed well, with **XGBoost** achieving the highest accuracy at 93.25% and a ROC-AUC score of 0.9694, indicating excellent discrimination. **GBM** and **Logistic Regression** followed closely, showing

strong sensitivity and comparable performance. Random Forest, while slightly lower in accuracy at 92.94%, still demonstrated effective classification. Overall, these results highlight the potential of machine learning techniques in early depression detection, with **XGBoost** being the most effective choice among the models tested.

## Conclusion

The use of these models for predicting depression illustrates the potential of machine learning techniques to enhance early detection of depression. Based on the findings, **XGBoost** stands out as the most reliable option due to its high accuracy and discriminative power. However, considering the combination of multiple models could further improve predictive accuracy, ultimately contributing to better clinical outcomes and informed decision-making in healthcare.