DS5110 Project Methods Report

Title: PREDICTIVE ANALYSIS FOR DEPRESSION

Authors (Team 4):

Rahul Chandak
Dhanooram Nagaraj
Sai Ram Asish Madiraju

Dataset Description:

The dataset consists of data from 34,364 people who mentioned whether they were depressed or not and a variety of factors that could potentially help us identify any trends which lead to it. There are a total of 492 columns (a combination of categorical and numerical variables), of which 491 consist of details like age, race, what medicines they take, health issues, income, working or retired, household size, etc.

Data Tidying

Of the 492 columns, there were some columns like pulse, BMI, and sleep hours that had null values, which do not make sense. We replaced these null values with mean values.

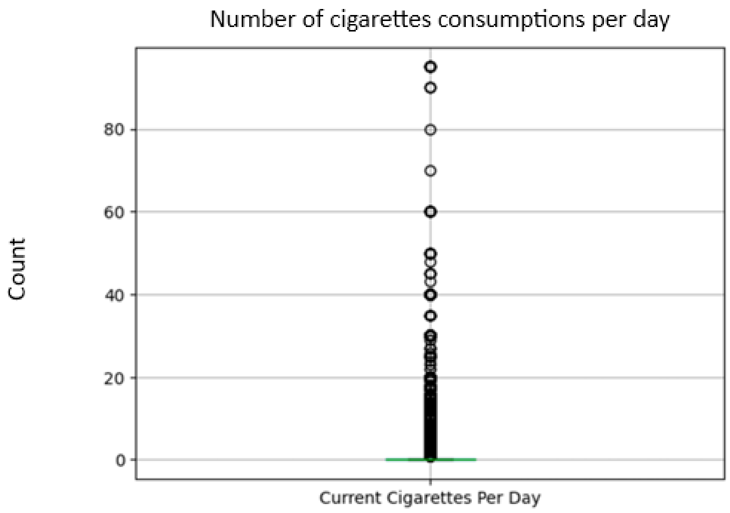
```
pulse_mean = df['Pulse'].mean()
df['Pulse'] = df['Pulse'].replace(0, pulse_mean)
BMI_mean = df['BMI'].mean()
df['BMI'] = df['BMI'].replace(0, BMI_mean)
sleephour_mean = df['Sleep Hours'].mean()
df['Sleep Hours'] = df['Sleep Hours'].replace(0, sleephour_mean)
```

Also to perform correlation analysis, we need columns to be numerical. Therefore using the label encoder from the sci-kit-learn library, we changed the categorical variables to numerical variables.We used Labelencoder to convert the categorical variable to numeric values. Where each unique value in the column will be assigned a value.

```
# Categorical to Numerical
le=LabelEncoder()
for col in string_cols:
  df[col] = le.fit_transform(df[col])
```

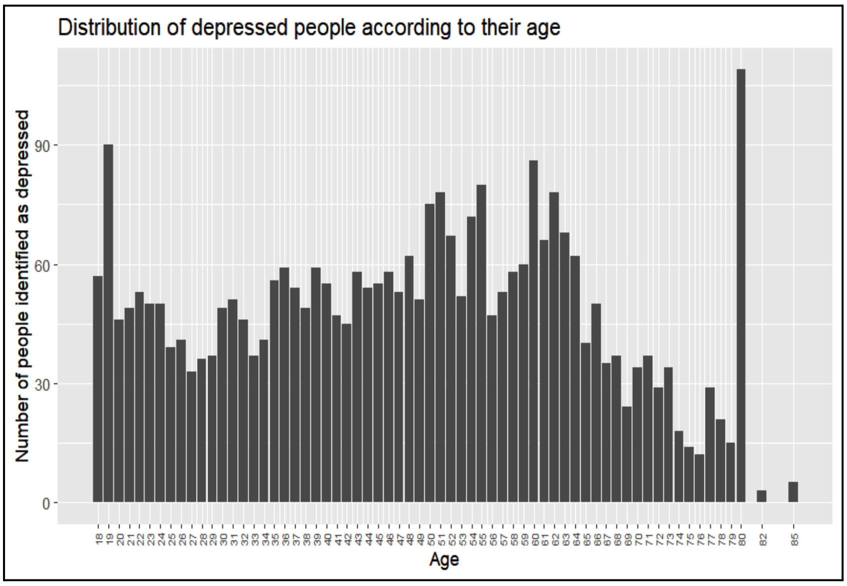
<u>Exploratory Data Analysis:</u>

We performed summary statistics to understand the mean/median and the standard deviation in all columns and we identified stats from each column. We performed a few visualizations to find outliers, like the number of cigarettes consumed per day.

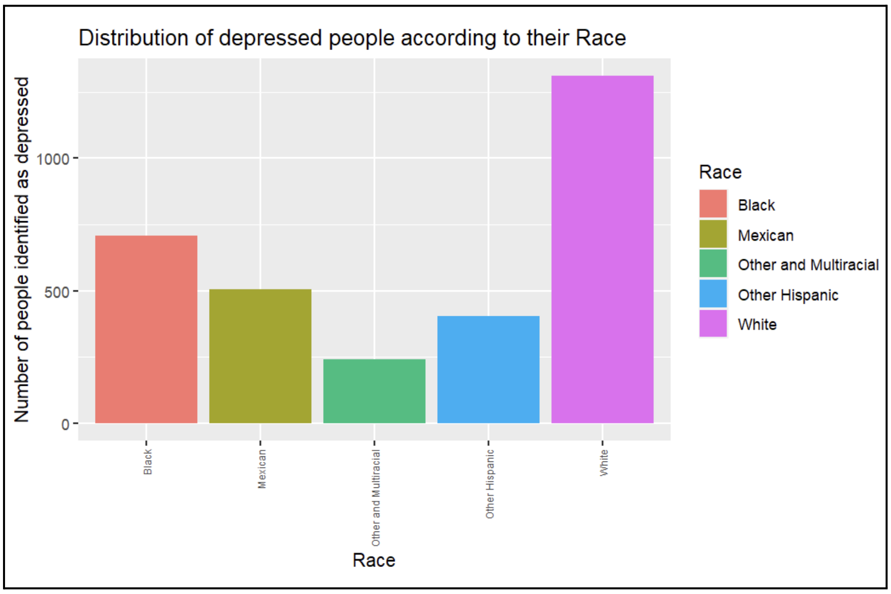


From the above visualization we can identify the general trend is around 0-5 cigarettes per day but there are extreme outliers like 80 cigarettes per day as well.

<u>Data Visualization</u>:
Plotted bar plots to visualize factors like Age and Race (Depression vs Race, Depression vs Age)



The above graph shows the trend in depressed people vs their age. We can observe that there is a peak at 19 years old and then slowly increases again at around 51-52 & 60, then it decreases. The graph conveys that 80 years old people are more depressed.



When comparing depressed people vs their race we observed that white people are more depressed but white people can also be more in the population. So we weren't able to conclude any results about depression with single variable selection, thus we explored more.

Correlation Analysis for feature selection

We identified the top 30 variables with the strongest relationship with 'Depression' by Correlation coefficients. This was also verified by evaluating and identifying the top 30 variables with the strongest relationship with 'Depression' using Pearson Correlation Coefficient.

```
#Top 30 based on correlation coefficients
for col in df.columns:
   corr_coef = df['Depression'].corr(df[col])
   corr_dict[col] = corr_coef
sorted_dict = {k: v for k, v in sorted(corr_dict.items(), key=lambda
item: abs(item[1]),reverse=True)}
top30_cols = list(sorted_dict.keys())[:30]

# Top 30 based on Pearson correlation coefficient value
pVal_dict = {}
for col in df.columns:
 corr,p_val=pearsonr(df['Depression'],df[col])
 pVal_dict[col] = p_val
```

```
sorted_dict2 = {k: v for k, v in sorted(pVal_dict.items(), key=lambda
item: abs(item[1]), reverse=False)}
top30_cols2 = list(sorted_dict2.keys())[:30]
```

Modeling

Since this was a binary classification problem, we used supervised machine learning algorithms to predict depression based on the shortlisted variables.

We split the filtered dataset into 80% training and 20% testing and fitted it across the below 5 models. For each model, we trained the model using the training dataset and calculate the accuracy using the test dataset.

Model 1: Random Forest Classifier - Accuracy = 91.64%
Random Forest is often used for Binary Classification problems, where the goal is to classify instances into one of two possible classes. The algorithm constructs multiple decision trees on various subsets of the dataset and then aggregates their predictions using a majority vote to make the final classification decision

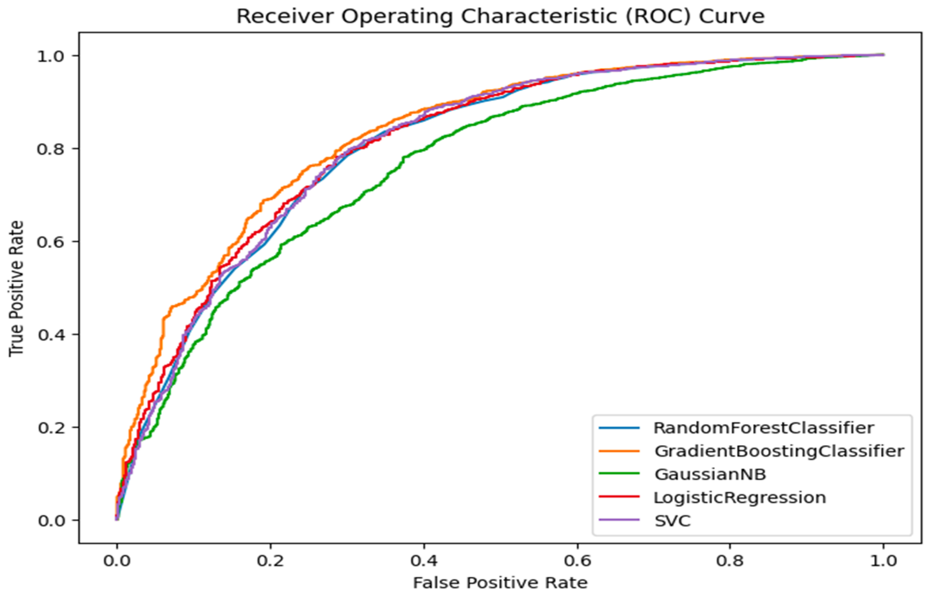
Model 2: **Gradient Boosting Classifier** - Accuracy = 91.83%
Gradient Boosting Classifier is an ensemble-based Supervised Machine Learning Algorithm used for Classification problems. It builds decision trees sequentially by focusing on the misclassified instances of the previous tree using gradient descent optimization. The final prediction is obtained by aggregating the predictions of all trees using a weighted sum.

Model 3: Naive Bayes Classifier - Accuracy = 80.72%
Naive Bayes Classifier is a Supervised Machine Learning Algorithm that is widely used for Classification problems, including Binary Classification. It calculates the probability of each class given the input features and then assigns the class with the highest probability to the new instance. Its better for high dimension dataset.

Model 4: Logistic Regression - Accuracy = 91.74%
Logistic Regression is used for binary classification algorithm that models the relationship between the input features and binary output variable using a logistic function to predict the probability of an instance belonging to the positive class.

Model 5: Support Vector Machine - Accuracy = 91%
Support Vector Machine (SVM) is a Supervised Machine Learning Algorithm used for Classification and Regression problems. It aims to find the optimal hyperplane to maximize the margin between the classes and assigns new instances to the class on either side of the hyperplane. SVM can also handle non-linearly separable data using kernel functions.

ROC Curves for all the models



As seen from the above graph, the orange curve, i.e. gradient boosting classifier, curves outwards the most which implies that of all the models tested, Gradient Boosting Classifier performs the best in giving the most accurate results. Gradient boosting is a good choice when working with structured data or when the data has many features. We took into account a total of 40 features and in conclusion this also resulted in giving the highest accuracy of all. Our accuracy results also align with the results of the above ROC curve.