DATA MINING

TERM PROJECT

1. Introduction

Our project is all about using a dataset called the 2020 Behavioral Risk Factor Surveillance System (BRFSS) Survey Data to build and test different models that predict if someone might have a depressive disorder. This dataset, named "project\_dataset-5K.csv," has already been cleaned up a bit and gives us information about 5000 people and 276 things about them. Each person's information is put into a row, and there's an important column called 'Class' that tells us if they have experienced depression ('Y') or not ('N').

Our main goal is to use this dataset to make different models that can guess if someone might have depression. We'll do a bunch of work to get the data ready, like fixing any missing information and making sure everything is in a format that our models can understand. Then, we'll pick out the most important things from the data to use in our models.

This report will walk you through what we did step by step. We'll start by looking at the data closely to see what's important. After that, we'll clean up the data, making sure it's all set for our models to use. Then, we'll pick out the best things from the data to use in our models. Finally, we'll build different models and see which one works the best for predicting depression. It's like a journey where we explore the data, clean it up, and then use it to make our models better.

In the end, we'll wrap up our report by talking about what we found from all our work. We'll talk about which model worked the best for predicting depression and what we learned from our project. We'll also suggest some ideas for future research. Our hope is that our project helps make better tools for identifying people who might be at risk of depression, so we can give them the support they need.

1. About the dataset

The project dataset is a 2020 Behavioural Risk Factor Surveillance System (BRFSS) Survey Data, which was downloaded from https://www.cdc.gov/brfss/annual\_data/annual\_2020.html and modified for this assignment.

*project\_dataset-5K.csv*, has 5000 tuples and 276 attributes.

In the dataset, each tuple is a person and *Class* is the class attribute. The class attribute value Y means a person has experienced a depressive disorder and N means a person has not experienced a depressive disorder.

#columns we are working with

Steps required:

1. Importing necessary libraries

At first, important libraries have been imported like:

* library(caret)

It is a comprehensive package that provides a unified interface for training and evaluating various machine learning models. caret simplifies the process of model training, tuning, and performance evaluation by providing a consistent framework across different algorithms.

* install.packages("fastDummies")

is a package used for creating dummy variables (also known as indicator variables or binary variables) from categorical variables in a dataset. Dummy variables are often required for modeling techniques that cannot handle categorical data directly, such as linear regression or decision trees.

* library(fastDummies)
* library(MASS)

The MASS package contains a collection of functions and datasets used in the book "Modern Applied Statistics with S" by Venables and Ripley. It includes various statistical functions and datasets for teaching and learning purposes.

* library(FSelector)

FSelector is a package for feature selection in R. It provides functions for selecting the most relevant features from a dataset based on different criteria such as information gain, chi-square, correlation coefficient, etc. Feature selection is important for improving model performance, reducing overfitting, and speeding up computation.

* library(Boruta)

Boruta is a feature selection algorithm specifically designed for random forest models. It identifies important features by comparing the importance of original features with importance of shadow features (randomly permuted features). Boruta helps in identifying relevant features while mitigating the risk of overfitting.

* library(RWeka)

RWeka is an R interface to the Weka machine learning toolkit. Weka is a collection of machine learning algorithms for data mining tasks. RWeka allows R users to access and utilize the functionality of Weka within the R environment.

* library(rsample)

rsample is a package for data splitting and resampling. It provides functions for creating different types of data splits such as training/test splits, cross-validation folds, bootstrap samples, and time series splits. Resampling techniques are essential for assessing model performance and reducing the risk of overfitting.

* library(psych)

psych is a package for psychological statistics and psychometric research. It provides functions for descriptive statistics, factor analysis, correlation analysis, reliability analysis, and various other statistical methods commonly used in psychology and related fields.

1. Loading the data

Data loading is the process of importing a dataset into the computational environment for analysis and manipulation. In the context of the project using R, we use functions provided by R packages to load the provided dataset project\_dataset-5K.csv

1. Data Pre-processing
2. Checking for Missing Values:

colSums(is.na(data)) is used to calculate the total number of missing values in each column of the dataset data.

is.na(data) returns a logical matrix where TRUE indicates missing values and FALSE indicates non-missing values.

colSums() then sums up the number of TRUE values (missing values) for each column, resulting in a vector containing the total count of missing values for each column.

This step helps in identifying which columns contain missing values and how many missing values are present in each column.

1. Imputing Missing Values:

For numerical variables, missing values are imputed with the mean of the respective column using mean(data[, i], na.rm = TRUE). The na.rm = TRUE argument ensures that missing values are excluded while calculating the mean.

For categorical variables, missing values are imputed with the mode (most frequent value) of the respective column. This is achieved by identifying the mode value using names(sort(-table(data[, i])))[1] and then replacing missing values with the mode value.

1. Calculating NearZeroVar

The nearZeroVar() function examines each numeric variable in the dataset to determine whether it has near-zero variance.

It calculates a statistic known as the "frequency ratio" for each variable. This ratio is the ratio of the frequency of the most common value to the frequency of the second most common value.

If the frequency ratio is greater than a predefined threshold (usually close to 1), the variable is considered to have near-zero variance.

The function then returns the names of variables (or a logical vector) that meet the near-zero variance criterion.

1. One-Hot Encoding

One-hot encoding is a technique used to convert categorical variables into a numerical format that can be used as input for machine learning algorithms. It is necessary because many machine learning algorithms cannot directly handle categorical data in their original form.

1. Feature selection
2. CFS

The CFS (Correlation-based Feature Selection) algorithm is a feature selection technique used in machine learning to identify the most relevant features for a predictive model.

in ourcode snippet, subset <- cfs(Class ~., data), the CFS algorithm is applied to the dataset data to select a subset of features that are most relevant for predicting the 'Class' variable (presence or absence of depressive disorder). The resulting subset likely contains the selected features identified by the CFS algorithm, which can then be used for further analysis or model building.

Overall, CFS is a powerful feature selection method that helps identify informative features for building accurate and interpretable predictive models. It is particularly useful when dealing with high-dimensional datasets where selecting the right subset of features can significantly improve model performance.

1. PCA
2. Information gain
3. Data Visualization
4. Performing Statistical Analysis:

The statistical analysis aims to understand the relationship between the selected features ('PHYSHLTH', 'GENHLTH', 'X\_PHYS14D', 'X\_RFHLTH', 'ASTHMA3') and the outcome variable 'Class' (presence or absence of depressive disorder).

By summarizing the data based on the outcome variable and selecting relevant features, the code prepares the dataset for further analysis, such as exploring correlations, identifying patterns, or building predictive models.

The resulting top\_data dataframe contains the subset of data that will be used for subsequent analyses or modeling tasks, focusing on the selected features and their association with the outcome variable.

1. Scatter Plot

A scatter plot matrix is created to visualize the correlation between multiple variables (HYSHLTH, GENHLTH, X\_PHYS14D, X\_RFHLTH, ASTHMA3).

Points in the scatter plot matrix are colored based on the Outcome variable.

This visualization helps identify potential relationships between variables and assesses their correlation with the target variable.

1. Box Plots

Box plots are created to visualize the distribution of important features (PHYSHLTH, GENHLTH, X\_PHYS14D, X\_RFHLTH, ASTHMA3) across different classes (Class).

Each box plot represents the distribution of a specific feature (xlab) within each class (ylab).

The title of each box plot indicates the feature being visualized (main).