The World Health Organization (WHO) approximates an annual fatality count of 17.9 million due to cardiovascular ailments (CVDs).

Our current engagement is with an R&D enterprise concentrating on healthcare remedies. Our company has gathered de-identified records from numerous medical facilities concerning various patients. The dataset incorporates pertinent details about each individual, encompassing personal particulars and certain medical specifics, encompassing their history of heart conditions.

1. Age: age of the patient [years]

2. Sex: sex of the patient [M: Male, F: Female]

3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]

4. RestingBP: resting blood pressure [mm Hg]

5. Cholesterol: serum cholesterol [mm/dl]

6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]

7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]

8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]

9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]

10. Oldpeak: oldpeak = ST [Numeric value measured in depression]

11. ST\_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

12. HeartDisease: output class [1: heart disease, 0: Normal]

Instructions

In the Jupyter Notebook:

1. Add an appropriate title for the project in a markdown cell.

2. Add a brief description to the same cell that explains what the project will focus on.

3. Import the libraries you think will be relevant to this project.

4. Read heart\_disease\_prediction.csv into a pandas dataframe.

5. Display the first five rows of the dataframe.

6. Print out the number of features and observations in the dataset.

Instructions

1. Display the descriptive statistics for the DataFrame.

2. In a markdown cell, add your observations or insights corresponding to the statistics. For example,

• What is the average age of the patients?

• Are there any features that have statistics that don't look right to you or that stand out? Maybe a very high or low value for a given statistic?

• Are there any missing values?

Instructions

1. Calculate the number of missing values in all columns.

2. For every categorical column in the dataset, including HeartDisease and FastingBS:

• Create a bar chart that displays the number of rows for each category of that column.

• Add appropriate axis labels, data labels and a title to each plot.

• In a markdown cell, briefly summarize any relevant insight that you gather from the plots. For example, how many patients are male and how many are female?

3. For every categorical column in the dataset, including FastingBS:

• Create a bar chart that displays the number of rows for each category of that column, grouped by HeartDisease.

• Add appropriate axis labels, data labels, a title, and a legend to each plot.

• In a markdown cell, briefly summarize any relevant insight that you gather from the plots. For example, which category of ChestPainType has a higher count for patients with heart disease?

4. [Optional] Explore any groupings and/or visualizations of the data as you see fit. For example:

• What percentage of patients with heart disease are male and over 50 years old?

• What is the median age of patients who were diagnosed with heart disease?

Instruction

1. Count the number of rows that have a 0 value for RestingBP.

2. Count the number of rows that have a 0 value for Cholesterol.

3. Based on the above two, either

• Remove those rows from the dataset, or

• Replace those values with the median value of the corresponding column and the corresponding HeartDisease value.

4. [Optional] Try a more complex approach instead of the above options.

Instructions

1. Convert the categorical features into dummy variables.

2. Create a Pearson's correlation heat map.

• You can use Seaborn's heat map() function to plot the heat map.

3. Identify from the heat map which features are reasonably correlated to HeartDisease.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Instructions

1. Split the dataset into training and validation sets.

• Select a suitable split percentage.

• [Optional] You are encouraged to experiment with this value.

• Set random\_state to a number of your choice to ensure reproducibility.

2. Create a list that contains your selected features as strings.

3. Iterate over the list. For every iteration:

• Create a k-NN classifier.

• Set a value for the number of neighbors.

• [Optional] You are encouraged to experiment with this value.

• Fit the model to the training data one feature at a time.

• Evaluate the model on the validation set.

• Print the accuracy of the model.

4. In a markdown cell, briefly describe which model performed the best out of all the selected features.

Instructions

1. Using MinMaxScaler, scale the selected features to the range (0, 1).

1.1. Create the scaler.

1.2. Fit and transform the features in the training set using the scaler and save the output.

1.3. Transform the features in the validation set.

2. Create a k-NN model.

3. Fit the model to the training data on the scaled features.

4. Evaluate the model on the scaled features in the validation set.

5. Calculate and print the accuracy of the model.

6. In a markdown cell, briefly describe how the model performed in relation to our previous model(s).

Instructions

1. Split the dataset into training and test sets.

• Select a split percentage that you think is suitable for this dataset.

• [Optional] Experiment with this value.

• Set random\_state to a number of your choice to ensure reproducibility.

2. Scale the training set.

3. Create a dictionary that stores the parameters and values you want to search over as key-value pairs.

4. Instantiate a k-NN model.

5. Create a GridSearchCV instance using the above model and the dictionary.

• Make sure to set scoring to accuracy.

6. Fit the above instance on the scaled features and corresponding labels.

7. Print out the best score and the best parameters obtained from the grid search.

• Refer to the attributes listed in the documentation in order to print these values.

8. In a markdown cell, briefly discuss the model's accuracy and the parameters that produced that result and compare it to our previous attempts.

Instructions

1. Scale the same features from the test set.

2. Predict the model's output on the scaled features from the test set using the best estimate.

3. Calculate and print the accuracy of the model's prediction.