Higher School of Computer Science and Mathematics of Monastir
Department of Computer Science
Engineering in computer science, specialty software engineering

Machine learning algorithms

Heart Disease Prediction

Objective:

- Use data visualization library in python: **Seaborn**, Matplotlib, Plotly, GGplot
- Data preparation
- Understand and interpret data visualization : Interpret charts, diagrams and statistics : central tendency, dispersion, distribution, outliers, presence of clusters, correlation, independent versus dependent variables, important versus not important variables

I. Data Description

Data set information: data collection, data source, dimension of the data set, number of features, number of records.

Feature information: type of features, decision variables, target variable, feature description.

II. Data cleaning

Missing data, noise, data cleaning, outliers, transform some date to categorical variables, find duplicated data.

III. Data exploration

- Data summary: min, max, mean, std, Q1, Q3, modes, median.
- Univariate exploration: histogram, boxplot, quantile plot, barplot, pie chart, violinplot
- Bi-variate exploration:
  - Histogram of the distribution of a continuous variable **X** concerning different values of the categorical variable **Y**.
  - Correlation of numerical variables.
  - Barplot of a categorical variable **X** concerning different values of a categorical variable **Y**.
  - Mosaïc plot of categorical variables
  - Boxplot of a continuous variable **X** concerning different values of a categorical variable **Y**.
  - Scatter plot
  - Pairplot
  - Clustermap

- Heatmap
- Class wise distribution of all features (hist, density, boxplot)
- Data distribution of attributes with box plot (density with boxplot)
- Categorical variable versus target probability distribution curve
- Bivariate analysis using bivariate plot
- Strip plot
- Regression analysis between a numerical variable and a categorical variable using regression analysis using a line plot

IV. Multivariate analysis

- Heatmap
- Hierarchical clustering using cluster map
- Missing value identification using heat map
- Joint plot
- Scatter plot
- Pairplot
- Rug plot
- Sub-plot
- Facet grid plot
- 3D scatter plot

Good luck ! 📊

Higher School of Computer Science and Mathematics of Monastir
Department of Computer Science
Engineering in computer science, specialty software engineering

Machine learning algorithms

Heart Disease Prediction

Question 1: Split the data into train and test while using defined function **train_test_split** from sklearn.

Explain the parameters ***test_size*, *train_size*, *random_state*, *shuffle*** *and* ***stratify***.

Question 2: create a python function **data_splitting** that s*plits data into Train and Test Set.*

Algorithm : calculate the size of your data. Define the proportion of train and test set. Shuffle the data. Generate a mask. Split the data according to mask.

Split the data into train and test while using the function **data_splitting**.

Question 3: Run the Knn algorithm for k=1.

- Use train_test_split to split your data into a training set and a testing set.
- Import KNeighborsClassifier from scikit learn.
- Create a KNN model instance with n_neighbors=1
- Fit this KNN model to the training data.
- Use the predict method to predict values using your KNN model and X_test.

Question 4: value of k

Set k_list as the possible k values ranging from 1 to kmax.

For each value of k in k_list:

Use sklearn KNearestNeighbors() to fit train data.

Predict on the test data.

Compute the error

Plot the curve of the error as function of k

Which value of k will you use and why.

Question 5:

- Retrain your model with the best K value (up to you to decide what you want)

- Create a confusion matrix and classification report.
- Compute independently the metrices: Precision, recall, F-measure, accuracy, ROC-curve, AUC.

Question 6 : Compare the train set and test set accuracy.

Question 7: How the distance impacts the predictions of knn.

Use different distance metrics and observe the results.

Question 9 : *Hyperparameter Tuning using grid search*

We will use three hyperparamters- n-neighbors, weights and metric.

1. n_neighbors: Decide the best k.
2. weights: Check whether adding weights to the data points is beneficial to the model or not. ‘uniform’ assigns no weight, while ‘distance’ weighs points by the inverse of their distances meaning nearer points will have more weight than the farther points.
3. metric: The distance metric to be used while calculating the similarity.

Question 10: use DecisionBoundaryDisplay to visualize the decision boundary.