**Predicting Cardiovascular Disease with Classification Machine Learning Algorithms**

- LAMA IBRAHIM ALHARBI

**Abstract**

The goal of this project was to use classification models to predict whether the patient has potential of cardiovascular disease or not in order to help improve the healthcare systems. I worked with data provided by Kaggle, dataset that were collected at the moment of medical examination. I did some feature engineering along with different classification algorithms and compare to see and achieve promising results for this problem.

**Design**

The data is provided by Kaggle, and presents a binary-class problem of Presence or absence of cardiovascular disease. Classifying statuses accurately via machine learning models would enable the hospitals to identify and improve the time and use accurate systems to diagnose the patients, and take actions faster to help them get better quickly.

**Data**

For this project, I have used dataset [Kaggle](https://www.kaggle.com/sulianova/cardiovascular-disease-dataset). It comprises a real dataset that were collected at the moment of medical examination of +70000 examples of data with 13 various attributes (12 predictors; 1 target) below table 1 describes the features in details. There are 3 types of input features: Objective: factual information; Examination: results of medical examination; Subjective: information given by the patient.

**Algorithms**

*Feature Engineering*

1. Converting age from days to years.
2. Converting categorical features to binary dummy variables.
3. Remove outliers (Unbelievable Values) from Blood pressure columns ap\_hi and ap\_lo.

*Models*

Decision Tree, random forest, logistic regression, naïve bayes, linear SVC, XGBOOST, k-nearest neighbors, support vector machine, extra tree, gradient boosting, adaBoost, perceptron, stochastic gradient descent, ridge, and bagging classifiers were used before settling on gradient boosting classifier as the model with highest accuracy performance.

*Model Evaluation and Selection*

The entire training dataset of 70,000 records was split into 80/20 train vs. holdout, and all scores reported below were calculated with 5-fold cross validation on the training portion only. Predictions on the 20% holdout were limited to the very end, so this split was only used and scores seen just once.

The official metric for data was classification rate (accuracy).

**Final Gradient Boosting Classifier scores:**

* Accuracy 0.7345
* Val accuracy 0.7366
* Precision 0.752
* Recall 0.692
* F1 0.721

**Tools**

* Numpy and Pandas for data manipulation
* Scikit-learn for modeling and data preprocessing
* Matplotlib and Seaborn for plotting
* Pandas\_profiling for interactive data visualizations