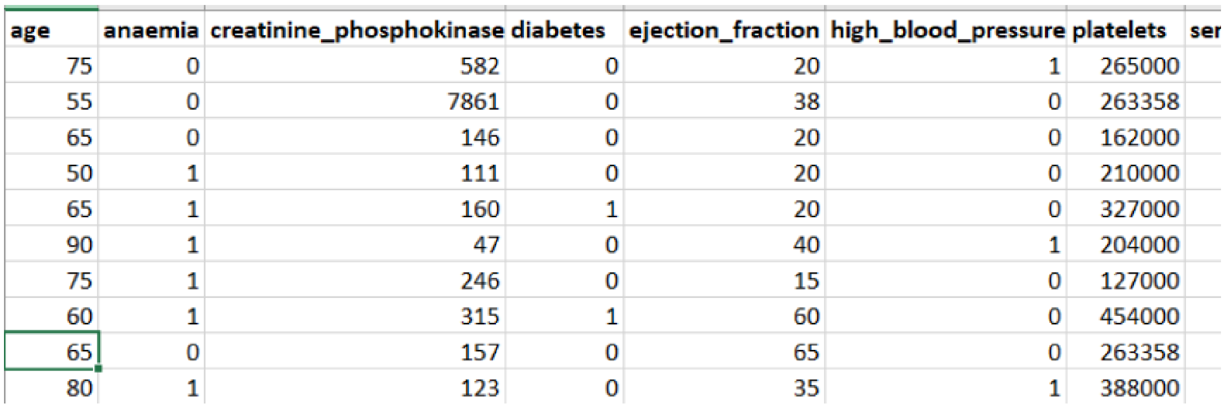
Computer Science Department
CS677 – Introduction to Machine Learning (CRN: 74016) – Fall 2023
Professor Krystyn Gutu
Project #3 | Due: December 7, 2023

This problem is a typical Classification Machine Learning task. You must build various classifiers by using the following Machine Learning models:

- Logistic Regression (LR)
- Decision Tree (DT)
- Random Forest (RF)
- XGBoost (XGB)

The dataset for exploration, modeling, and interpretability, explainability is called "Heart failure clinical records Data Set" to be found at the UCI (University California Irvine) Machine Learning Repository.

This dataset contains the medical records of 299 patients who had heart failure, collected during their follow-up period, where each patient profile has 13 clinical features.



age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	ser
75	0	582	0	20	1	265000	
55	0	7861	0	38	0	263358	
65	0	146	0	20	0	162000	
50	1	111	0	20	0	210000	
65	1	160	1	20	0	327000	
90	1	47	0	40	1	204000	
75	1	246	0	15	0	127000	
60	1	315	1	60	0	454000	
65	0	157	0	65	0	263358	
80	1	123	0	35	1	388000	

The dataset **(heart_failure_clinical_records_dataset.csv)** could be downloaded from this link:
https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records
https://archive.ics.uci.edu/ml/machine-learning-databases/00519/

Attribute Information - Thirteen (13 = 12 + 1) clinical features:
- age: age of the patient (years)
- anemia: decrease of red blood cells or hemoglobin (boolean)
- **high blood pressure:** if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary) - serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- [target] death event: if the patient deceased during the follow-up period (boolean)

Perform the following tasks:

1) Perform **Explanatory Data Analysis (EDA)** / indicate how features correlate among themselves, with emphasis to the target/label one

2) Apply **Machine Learning Modeling** on the dataset using all the above 4 algorithms. Tune (hyper-parameter tuning) each model by calling the GridSearchCV method. Indicate which combination of Hyperparameters produces the best result. Note: Use **accuracy** and **AUC-ROC** metrics when evaluating your models.

Interpretation/Explanation

3) Perform **Machine Learning Interpretability / Explanability** tasks as follows:

    a. Use the **'eli5'** library to interpret the "white box" model of **Logistic Regression.** Apply 'eli5' to visualize the weights associated to each feature.
    Use 'eli5' to explain specific predictions, pick a row in the test data with negative label and one with positive

    b. Use the **'eli5'** library to interpret the "white box" model of **Decision Tree.** Apply 'eli5' to list the feature importance ordered by the highest value.

Get an explanation for a given prediction, one positive and one negative. This will calculate the contribution of each feature in the prediction. The explanation for a single prediction is calculated by following the decision path in the tree, and adding up contribution of each feature from each node crossed into the overall probability predicted.

eli5 can also be used to explain black box models, but we will use LIME and SHAP for our two last models instead.

c. Use **LIME** to explain both the **Random Forest** and the **XGBoost** models.
   Create a LIME explainer by using the **LimeTabularExplainer** method, the main explainer to use for tabular data.
   LIME fits a linear model on a local shuffled dataset. Access the coefficients, the intercept and the **R^2** of the linear model, for both model interpretability.

   Note: If R^2 is low, the linear model that LIME fitted isn't a great approximation to your model, meaning you should not rely too much on the explanation it provides.

d. Use **SHAP** library to interpret the **XGBoost** model – specifically, TreeExplainer.
   This method of SHAP, **TreeExplainer,** is optimized for tree-based models.
   Visualize your explanations, one for positive and one for negative, using the **'force_plot'** function.

   Note; You need to establish a 'base value' in order to be used by 'force_plot'. The explainer.expected_value is the 'base value'.

   Create the feature importance plot by calling SHAP's **'summary_plot'** function, for each class / label.

4) **Predict observations,** one for positive and one for negative label, by using all four (4) models and indicate which one gives the better prediction.

   Provide output for showing the accuracy of each model as follows:
   False/True label: 0/1 (or 0/1 depending how you define labels)
   - LR: [prob_T prob_F]
   - DT: [prob_T prob_F]
   - RF: [prob_T prob_F]
   - XGB: [prob_T prob_F]

   The above calculations are derived by calling the predict_proba method.
   Note: **predict_proba(X):** Predict class probabilities for X.

Write **Python** scripts in order to complete the above tasks along with their output. All work should be done and submitted in a single **Jupyter (or Colab) Notebook.**