Project Assignement

# Provided Datasets:

1. Banknote dataset (data\_banknote\_authentication.txt)
   * Dataset information on: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>
2. Boston House Price Dataset (housing.data)
   * Dataset information on: <https://archive.ics.uci.edu/ml/datasets/Housing>
3. Ionosphere Dataset (ionosphere.data)
   * Dataset information on: <https://archive.ics.uci.edu/ml/datasets/Ionosphere>
4. Pima Indians Diabetes Dataset (pima-indians-diabetes.txt)
   * Dataset information on: <https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.names>
5. Wheat Seeds Dataset (seeds\_dataset.txt)
   * Dataset information on: <https://archive.ics.uci.edu/ml/datasets/seeds>
6. Wine Quality Dataset (winequality-white.csv)
   * Dataset information on: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

# Your tasks:

**On a dataset of your choice,**

* Define and formulate your problem as a data mining task

**Then, write python codes for**

1. Exploratory data analysis:
   * Load the data in Pandas Frame
   * Variable Identification:
     + Define the type of every variable (numerical or categorical…) and its role in the dataset (input variable or an output variable),
   * Statistics over each numerical attribute (mean, quartile, min, max, …)
   * Univariate Analysis:
     + For numerical variables: build histograms and boxplots for each numerical variable independently. These figures would give us an understanding about the variables’ central tendencies and spread.
     + For categorical variables: build a bar chart visualization that shows the frequencies in each category.
   * Bi-Variable Analysis:
     + Continuous & Continuous (Matrix): We can build a scatter plots in order to see how two continuous variables interact between each other.
     + Categorical & Categorical: A Stacked Column Chart is a good visualization that shows how the frequencies are spread between the two categorical variables.
     + Categorical & Continuous: building boxplots combined with swarmplots.
   * Detecting / Treating missing values: More of an art rather than a systematic approach and usually it depends to the problem in hand. However we describe: two different situations:
     + If for example we have only few missing values and they appear to be random we can just proceed with the deletion of these cases.
     + If we have many missing values, we don’t want to proceed into their deletions because that would end up in having a much smaller dataset which would influence the predictive model’s performance. In this case we would either replace the values with the median/mean/model or/and add another column that shows if the other variable has a missing value or not. In the latter case, ideally they newly added column should correlate with the output variable, thus creating a new variable that might be a good predictor.
   * Detecting / Treating outliers: Having many outliers in the dataset can harm the predictive model’s performance and thus would be nice to treat them. Also in this phase there is no systematic approach to deal with them. Some ideas are :
     + During the detection phase, one of the best visualization approaches to use are boxplots for univariate analysis and scatterplots for bi-variate analysis.
     + During the treatment phase, we can either delete them if they are very few or if not, we can use a special treatment like imputing them or just treat them independently by having their own predictive model.
   * Feature Engineering: During this phase we try to infer better variables/predictors out of the existing variables. Like imagine we have a date variable, we can create other new variables out of it like weekday/weekend, Monday/Tuesday…., and so on. This newly created variable can turn out to be a good predictor if it correlates somehow with the output variable.
2. Evaluate various machine learning models
   * Model selection:
     + Research and select a set of diverse machine learning models suitable for the task (e.g., decision trees, random forests, support vector machines, neural networks).
     + Implement baseline models with default hyperparameters to establish a performance benchmark.
   * Hyperparameter Tuning:
     + use techniques such as grid search, random search, to fine-tune hyperparameters for each model.
     + Explore a range of hyperparameters to identify the optimal configuration for each model.
   * Cross-Validation:
     + Perform k-fold cross-validation to assess model performance robustness and avoid overfitting.
     + Evaluate models using appropriate performance metrics (e.g., accuracy, precision, recall, F1-score, ROC-AUC).
   * Ensemble Methods:
     + Experiment with ensemble methods such as bagging, boosting, and stacking to combine predictions from multiple models.
     + Evaluate the performance improvement achieved by ensemble methods compared to individual models.
   * Comparison and Analysis:
     + Compare the performance of different models using the test set
     + Analyze the strengths and weaknesses of each model,

**Prepare a comprehensive report summarizing the findings, including visualizations, tables, and insights derived from the analysis.**