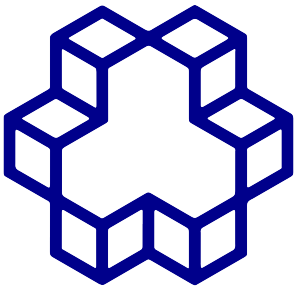
CODING ASSIGNMENT 3

Decision Tree and Naive bayes



دانشگاه صنعتی خواجه نصیرالدین طوسی

Artificial Intelligence

Instructor: Dr.Pishgoo

TAs: Amirreza Mehrzadian, Alireza Saeednia

Deadline:

1 About dataset

This dataset of breast cancer patients was obtained from the 2017 November update of the SEER Program of the NCI, which provides information on population-based cancer statistics. The dataset involved female patients with infiltrating duct and lobular carcinoma breast cancer (SEER primary cites recode NOS histology codes 8522/3) diagnosed in 2006-2010. Patients with unknown tumour size, examined regional LNs, positive regional LNs, and patients whose survival months were less than 1 month were excluded; thus, 4024 patients were ultimately included.

2 Task

you are free to choose between naive bayes and decision tree classifiers,you can even use both by voting algorithms, but at the end your accuracy score should be more than 90%. you can use the following code to see the f1-score, precision, recall, and accuracy score.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
print(classification_report(y_test, y_pred))

```

naive bayes hyperparameters:

- priors: ArrayLike — None = None,
- var_smoothing: Float = 1e-9

It's worth noting that while these hyperparameters exist, Naive Bayes classifiers are often used with their default parameters, and unlike many other models, they don't require extensive hyperparameter tuning. They are simple but powerful algorithms for predictive modeling under supervised learning algorithms.
decision tree hyperparameters:

- criterion: This is the function used to measure the quality of a split. Supported criteria are "gini" for the Gini impurity, "entropy" for the information gain, and "log_loss".
- splitter: This is the strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split
- max_depth: This is the maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- min_samples_split: This is the minimum number of samples required to split an internal node.
- min_samples_leaf: This is the minimum number of samples required to be at a leaf node.

- min_weight_fraction_leaf: This is the minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.
- max_features: This is the number of features to consider when looking for the best split.
- random_state: This controls the randomness of the estimator.
- max_leaf_nodes: This is the maximum number of leaf nodes.
- min_impurity_decrease: A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
- class_weight: Weights associated with classes in the form class_label: weight. If not given, all classes are supposed to have weight one.
- ccp_alpha: Complexity parameter used for Minimal Cost-Complexity Pruning1.