School of Computer Science Engineering and Technology
Assignment-08

Course- B.Tech	**Type-** Core
Code-23CS106	**Course Name-** Artificial Intelligence & Machine Learning
Year- 2024-2025	**Semester-** Even, **Instructor:** Prof. E.L.N. Kiran
Date– 30-01-2024	**Batch-** AIML-A,B

Implement Decision Tree Classification using OpenML Car Dataset

In machine learning, decision trees are supervised learning models with associated learning algorithms that analyze data used for classification.

Let's say that you want to make a decision of the car to buy. In order to get the right car to buy, you could go on and evaluate the level of the safety, the number of sits and doors by asking series of if like questions.

1. Import the Libraries required for Decision Tree

Import all libraries required along with visualization to completed the task on DT

2. Import the OpenML Car Dataset from Sklearn Packages

- Once downloaded the car data, prepare the dictionary format to access the data using its keys.
- Describe all the features from the dictionary, feature names.
- Setup the dataframe , describe itd details, check for missing values.
- Identify the target class and assign it to the dataframe.
- Perform exploratory analysis of dataframe using `seabon - sns package`
- the Draw the boxplot of first 10 columns to verify their role in decision to buy the car or not.

3. Train and Test Data

- Prepare the Train and Test data from the dataframe.
- Drop the `BinaryClass` column and define the dataframe with only `target` results
- Split the data into train, test using `train_test_split`

4. Train the DT using the Train Dataset

- `import sklearn.tree import DecisionClassifier`
- Apply the `model.fit` to dataset

5. Predict and Analysis the Performance of the DT Model

- Apply `model.predict(X_test)`
- Generate the accuracy_score, classification_report, confusion_matrix using `sklearn.metrics`
- Plot the decision tree and it branches using *from sklearn.tree import plot_tree, plot.figure*

6. Improve the Accuracy of Model using GridSearchCV Model

- Given the following parameters validate the GridSearchCV Model:
  `param_grid = 'C': [0.1,1, 10, 100, 1000], 'gamma': [1,0.1,0.01,0.001,0.0001], 'kernel': ['rbf']`
- Implement the model.predict
- Generate the classification_report, confusion_matrix using `sklearn.metrics`