**Assignment 3- Ben Smerd 22072922 CSE4002**

**Introduction**

Within this report, I will be discussing how I implemented a decision tree classifier to predict the class of wine from a given dataset based on the chemical attributes of the wine. I used the decision tree classifier algorithm, which is a supervised learning method which works best for classification and regression like problems. More specifically I used the ID3 decision tree algorithm which classifies the wines into three different classes. In this project I used the python programming language and the libraries of scikit-learn, pandas and matplotlib.

**Task Description**

The task for this project is to develop a machine learning model that can accurately predict a wine class given a set of input features. The model will be able to determine the output based on a set of 13 attributes- Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total Phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines and Proline. To solve this machine learning problem I used the ID3 decision tree algorithm, which works by selecting the best feature at each step, based on an evaluation criteria. The evaluation criteria used was the entropy-based which selects the optimal splits within the decision tree.

**Implementation**

*Importing Modules*

To start the process I need to import the appropriate libraries which offer the functionality to get the wine dataset and create the decision tree. Pandas was chosen because it provides powerful and efficient data manipulation capabilities and makes it easy to preprocess the wine dataset within a structured dataframe. Scikit-learn was also used because it has user-friendly and powerful machine learning algorithms- in this case the decision tree classifier. Matplotlib was also used because it can create easy to plot visualisations.

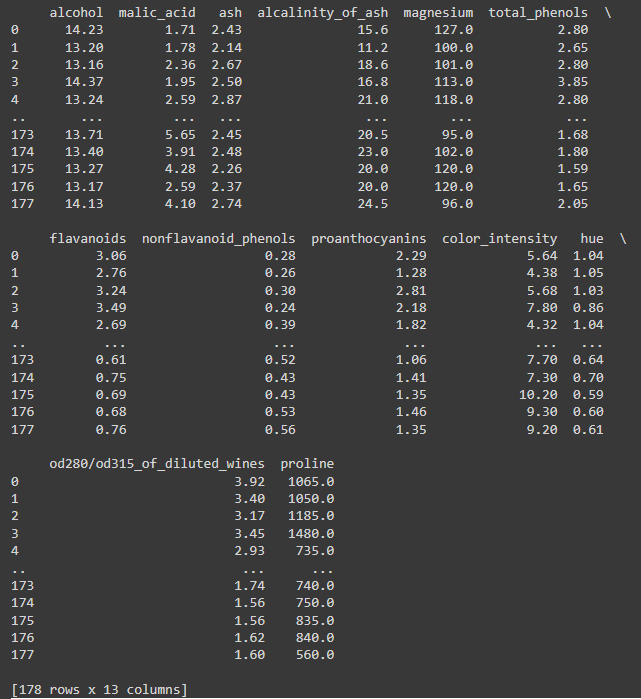
A screen shot of a computer program

Description automatically generated

*Get Dataset*

Scikit-learn provides an out of the box Wine Dataset which was used for this specific task. The dataset contains 178 samples of wine and each wine has 13 attributes as the features. The output variable will fit into one of three classes (class\_0, class\_1, class\_2).





After the Wine Dataset has been loaded into the variable *data,* I used pandas to store it into a DataFrame which makes it much more convenient to perform data manipulation. The target column of the wine dataset has also been added to the DataFrame, separating it from the input features.

A black background with white text

Description automatically generated

*Split Dataset*

To evaluate the performance of the mode, I used a basic sample split of the dataset into 80% training and 20% test sets. I used the train\_test\_split function from the sklearn.model\_selection library. It is important to split the data up so that the model can be trained on substantial portion of the data while being evaluated on unseen samples.



This split is essential in evaluating the performance of the model on an independent test set. This helps prevent cases of overfitting, where a model might memorise the training data, which leads to poor generalisation and resulting in worse performance on unseen data. The ID3 algorithm used can have problems with noise, which results in greater changes of overfitting, this is why it is important to split the data in this project to detect overfitting and assess whether pruning is necessary.

*ID3 Decision Tree*

I created an instance of the DecisionTreeClassifier from sklearn.tree, which creates a decision tree model. To make sure that the decision tree was built using the ID3 algorithm, I added in the criterion of ‘entropy’ which is based on information gain.



*Information Gain*

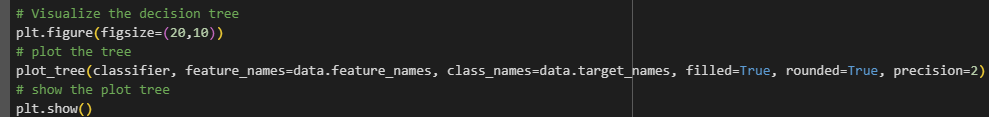
The information gain will evaluate the best feature to split the dataset on at each node of the tree. It measures how well a feature splits the data in terms of the reduction in entropy, with high information gain indicating a better split. Having now built the decision tree classifier based on the ID3 algorithm, I then trained the classifier model on the training dataset, which then builds the decision tree based on the ID3 principles for the wine data set. This is done by using the .fit method on the classifier variable.

A black background with white text

Description automatically generated

*Visualising Decision Tree*

Now that the ID3 decision tree model has been trained with the wine dataset, I wanted to then be able to visualise the decision tree. Using the powerful library of matplotlib to create a graphical representation. I used the plot\_tree function from scikit-learn and matplotlib to create it.



The results from the decision tree visualisation clearly represent the features chosen for splitting the data, which makes analysing the decision making a lot clearer in the process.

A diagram of a number system

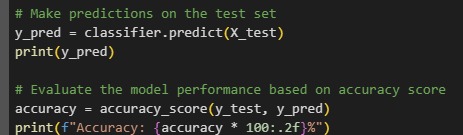
Description automatically generated with medium confidence

From the graphical representation shown above, we can see that each node in the tree represents a decision based on a specific feature. The root node as the OD280/OD315 of diluted wines feature has been determined to be the feature providing the highest information gain. From there it branches out where there are further decisions based on other features. It highlights the key roles that the branches play in differentiating the wine classes.

Because the tree is not too deep, this can represent that the model did not overfit on the noise from the data which helps in generalising future input data.

*Model Evaluation*

Once training the model has finished, I used the test dataset to evaluate how it will perform. The predictions for the test set were generated using the *predict* method and the accuracy was calculated sing the *accuracy\_score* function from scikit-learn.





The accuracy result shows 91.67% which is very high and shows that the results of the decision tree classifier are working well with generalisation and not being overfitted.

**Results and Decisions**

The accuracy of the decision tree model was 91.67%, which indicates the model perfomed well on the test dataset. Due to the small size of the data set, the high accuracy may suggest that the decision tree was able to learn a meaningful pattern from the data without overfitting and lowering its chances of generalisation. This strong test accuracy suggests that the decision tree has successfully captured the patterns and relationships between the attributes of the wines and their respective classes. Since the test set represents unseen data, this is a good indicator that the model can perform well in real-world scenarios, where it will be classifying new samples based on the same attributes with high accuracy.

As we can see in the visualisation of the decision tree, that the features alcohol and proline were used frequently to split up the data, suggesting that these features provide significant information to distinguishing between wine classes. The root node of OD280/OD315 of diluted wines played the most influential role in classifying the wine into one of three classes. Using information gain within the ID3 algorithm helped the decision tree classifier to determine this using the entropy score, which is shown in the visualisation graph.

**Future Improvements**

An area for improvement could be introducing pruning to the tree to reduce down the number of branches by getting rid of any that have little impact on the classification accuracy. It also helps with reducing overfitting and better at generalising on unseen data. Pruning can reduce the noise or irrelevant features of data so it removes complexity.

Another area of improvement could be adding in hyperparameter tuning, which would adjust areas like max\_depth, min\_samples\_split and min\_samples\_leaf that help fine-tune decision trees structure, improving generalisation.

**Alternative Approach**

I could have also used an artificial neural network (ANN), more specifically a multilayer perceptron (MLPs) for solving a classification or regression problem. A decision tree will represent knowledge symbolically which is great for arriving at a decision based on a series of if-then rules, while an ANN represents knowledge sub-symbolically, using numerical weights to capture patterns. Neural networks provide greater flexibility when dealing with more complex datasets or when relationships are less clear.

We would have 3 layers- input, middle and output layers. The input would have the 13 wine attribute features and they would get passed through one or more hidden layers where the weight would be computed and applied in non-linear activation functions (sigmoid function). This then propagates the results through the network and the output would be one of the three wine classes.