National Institute of Technology
Tiruchirappalli, Tamil Nadu – 620 015

Model Building

Date: 16.03.2022

Exercise 1: Decision Tree Algorithm

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn import datasets
from sklearn.metrics import classification_report
# Load dataset inside DataFrame
features=pd.DataFrame(datasets.load_iris()['data'],columns=datasets.load_iris()['feature_names'])
targets=pd.DataFrame(datasets.load_iris()['target'])
# Split the Dataset and train the model
featureTrain, featureTest, targetTrain, targetTest = train_test_split(features, targets, test_size=0.2)
model = DecisionTreeClassifier()
fittedModel = model.fit(featureTrain, targetTrain)
# Do predictions
predictions = fittedModel.predict(featureTest)
# Print Confusion Matrix & Accuracy Score
print (confusion_matrix(targetTest, predictions))
print (accuracy_score(targetTest, predictions))
# Print Classification Report
target_names = ['sentosa', 'versicolor', 'virginica']
print(classification_report(targetTest, predictions, target_names=target_names))
```

# Now the model is ready to take some real-time data and produce output. To feed some real-time data and find which class the data belongs to, type the following

```
list1 = [[5.1,3.5,1.4,0.2],[5.5,2.4,3.8,1.1],[6.5,3,5.2,2]]

print(fittedModel.predict(list1))
```

-- XXX --

Micro-averaged: all samples equally contribute to the final averaged metric

Macro-averaged: All classes equally contribute to the final averaged metric

Weighted-averaged: Each classes's contribution to the average is weighted by its size

So which type of averaging is preferable? As usual, this largely depends on the problem you're trying to solve. Do you have a class-imbalanced dataset? Is one class more important to get right than others? If you have an under-represented class which is important to your problem, macro-averaging may better, as it will highlight the performance of a model on all classes equally. On the other hand, if the assumption that all classes are equally important is not true, macro-averaging will over-emphasize the low performance on an infrequent class. Micro-averaging may be preferred in multilabel settings, including multiclass classification where a majority class is to be ignored.

https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/#:~:text=Macro%2Daveraged%3A%20all%20classes%20equally,is%20weighted%20by%20its%20size