Loughborough University

COP528 Applied Machine Learning (LABs)

LAB Day 05. Feature selection & Decision tree

Introduction

The aim of this session is to understand decision tree method. The experiments include 4 tasks.

Task 01. Manually build a decision tree

A training set of data is provided, please calculate the MI of each split and build a decision tree manually.

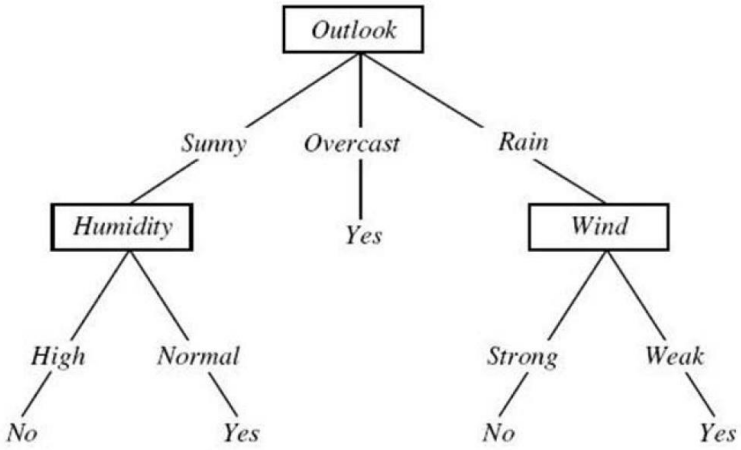
(1) Please calculate the MI of each part of the given data set.

Outlook	**Temperature**	**Humidity**	**Wind**	**PlayTennis**
sunny	hot	high	weak	No
sunny	hot	high	strong	No
overcast	hot	high	weak	Yes
rain	mild	high	weak	Yes
rain	cool	normal	weak	Yes
rain	cool	normal	strong	No
overcast	cool	normal	strong	Yes
sunny	mild	high	weak	No
sunny	cool	normal	weak	Yes
rain	mild	normal	weak	Yes
sunny	mild	normal	strong	Yes
overcast	mild	high	strong	Yes
overcast	hot	normal	weak	Yes
rain	mild	high	strong	No

(2) Please manually build a decision tree based on this dataset.

*Solutions Examples

Split data on outlook, then further calculate the entropy for two branches.



Solution:

Layer1:

Start point: Ent(PlayTennis)=$-\frac{9}{14}\log_2\frac{9}{14}-\frac{5}{14}\log_2\frac{5}{14}$=0.94

Ent(PlayTennis|Outlook)=0.69, Ent(PlayTennis|Temperature)=0.91,

Ent(PlayTennis|Humidity)=0.788, Ent(PlayTennis|Wind)=0.89

Information Gain: IG(Outlook)=0.94-0.69=0.25, IG(Temperature)=0.94-0.91=0.03,

IG(Humidity)=0.94-0.788=0.142, IG(Wind)=0.94-0.89=0.05

So choose 'Outlook' as first division point.

Layer2 (left):

Start point: Ent(Sunny)=0.97

Ent(Sunny|humidity)=0, Ent(Sunny|wind)=0.95,

Ent(Sunny|temperature)=0.399

Information Gain: IG(Temperature)=0.97-0.399=0.571, IG(Humidity)=0.97-0=0.97,

IG(Wind)=0.97-0.95=0.02

So choose 'Humidity' as division point

Layer2(right):

Start point: Ent(Rain)=0.97

Ent(Rain|wind)=0, Ent(Rain|temperature)=0.95

IG(Wind)=0.97-0=0.97, IG(Temperature)=0.97-0.95=0.02

So choose 'Wind' as division point

Task 02. Use the decision tree classifier in Sklearn to verify the tree you build from the previous practice.

Hint: The tree built by Sklearn might be different from the one built manually.

Think about why and compare them. The display function 'plot_tree()' in Sklearn and 'savefig()' in matplotlib maybe helpful.

An example of using decision tree in sklearn is given as:

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion='entropy',random_state=0, max_depth=3, min_samples_leaf=2)
model.fit(X,Y) #train your classifier using preprocessed data, saying attributes (X) and label (Y)
```

The data can be imported as follows

```
import pandas as pd
tennis = pd.read_csv('tennis.csv', header=0)
```

*Solutions Example

```
#
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn import preprocessing
import pandas as pd
import math
import matplotlib.pyplot as plt
from sklearn.tree import export_text

tennis = pd.read_csv('tennis.csv', header=0)

le = preprocessing.LabelEncoder()
tennis = tennis.apply(le.fit_transform)
X = tennis.iloc[:, :-1]
Y = tennis.iloc[:, -1]

model = DecisionTreeClassifier(criterion='gini',random_state=0, max_depth=3, min_samples_leaf=2)

model.fit(X,Y)

fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(model,
                   feature_names=['outlook','temperature','humidity','wind'],
                   class_names=['no', 'yes'],
                   filled=True)
fig.savefig('tennis_tree.png')
```

Task 03. Decision tree on Wine dataset

In the lab session of Day 03, you have made some data processing on the Wine dataset. Your task today is to use decision tree to make classification models to classify wines into 3 classes. Compare the results with a decision tree classifier based on the evaluation methods learned at the first day.

Here is the code to load the data and convert them into Pandas. You could use the numpy array with sklearn directly:

```
from sklearn.datasets import load_wine
import pandas as pd
wine = load_wine()
winedata = pd.DataFrame(data= wine.data, columns=wine.feature_names)
winedata['target'] = pd.Series(wine.target)
winedata.head(5)
```

*Solutions Example

```
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn import metrics

features, target = load_wine(return_X_y=True)
# Make a train/test split using 30% test size
X_train, X_test, Y_train, Y_test = train_test_split(features, target,
                                    test_size=0.30,
                                    random_state=0)

def eval_model(model_name, X_train, X_test, Y_train, Y_test):

    if model_name == 'DecisionTreeClassifier':
        model = make_pipeline(StandardScaler(),
                    DecisionTreeClassifier(criterion='entropy',
                                random_state=0,
                                max_depth=3,
                                min_samples_leaf=2)
                    )
    model.fit(X_train, Y_train)
    pred_test = model.predict(X_test)
    print(pred_test.shape, Y_test.shape)
    test_f1 = metrics.f1_score(Y_test, pred_test, average='micro')
    test_acc = metrics.accuracy_score(Y_test, pred_test)

    print(f'{model_name} test F1:{test_f1}, test acc:{test_acc}')

eval_model('DecisionTreeClassifier', X_train, X_test, Y_train, Y_test)
```

Task 04. Decision tree on breast cancer dataset

The breast cancer set is from the Institute of Oncology, University Medical Centre, Ljubljana, Yugoslavia, and was provided by M. Zwitter and M. Soklic, both with thanks. The dataset is a classical dichotomous dataset. Use this to build a decision tree, and evaluate it.

```
from sklearn.datasets import load_breast_cancer
dataset = load_breast_cancer()
df= pd.DataFrame(data= dataset.data, columns=dataset.feature_names)
df['target'] = pd.Series(dataset.target)
df.head(5)
features, target = load_breast_cancer(return_X_y=True)
```

*Solutions Example

```
# *Solutions Example
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
import pandas as pd
from sklearn.datasets import load_breast_cancer

features, target = load_breast_cancer(return_X_y=True)
X_train, X_test, Y_train, Y_test = train_test_split(features, target,
                                    test_size=0.30,
                                    random_state=0)

model = DecisionTreeClassifier(criterion='entropy',random_state=0, max_depth=3, min_samples_leaf=2)

model.fit(X_train,Y_train)


pred_test=model.predict(X_test)
test_f1 = metrics.f1_score(Y_test, pred_test, average='micro')
test_acc = metrics.accuracy_score(Y_test, pred_test)

print(f'test F1:{test_f1}, test acc:{test_acc}')
```