5th Week: Decision Tree and Random Forest

1. Decision Tree: Classification

In [ ]:

```
# importing tools
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import tree, metrics
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

In [ ]:

```
# reading iris dataset
iris_data = pd.read_csv('data/iris.csv')
print(type(iris_data), iris_data.shape)

X_iris = iris_data[['SepalL', 'SepalW', 'PetalL', 'PetalW']]
y_iris = iris_data['Name']
```

In [ ]:

```
# default test data size: 25%
X_train, X_test, y_train, y_test = train_test_split(X_iris, y_iris)

# Training with DecisionTree algorithm
irisDT = tree.DecisionTreeClassifier().fit(X_train, y_train)   # Decision Tree classifier
class_prediction = irisDT.predict(X_test)   # Classification, prediction

# Calculating accuracry
ac_score = metrics.accuracy_score(y_test, class_prediction)
print(f'Accuracy rate = {ac_score:.5f}')

confusion = metrics.confusion_matrix(y_true=y_test,y_pred=class_prediction)
print(confusion)

plt.figure(figsize=(14,8))
tree.plot_tree(irisDT.fit(X_train, y_train), feature_names=['PetalL','PetalW','SepalL','SepalW'
], class_names=y_train)
plt.show()
```

1. Random Forest

In [ ]:

```
# Classifying mushrooms
mshrm = pd.read_csv("data/mushroom.csv", header=None) # first column: p or e (labels), 22 featu
res
print(mshrm.shape)
print(mshrm.head())  # p: poison, e: no poison

# The ord() function in Python accepts a string of length 1 as an argument and returns the unico
de
# code point representation of the passed argument
ord('d')

# Learning the iterrows() of pandas: loop through each row of dataframe
string_series = pd.DataFrame([['r','a','n','d','o','m'],['f','o','r','e','s','t']])
for i, value in string_series.iterrows():
    for j in value:
        print(f'{i}: {ord(j)}', end=' ') #Return the Unicode code point for a one-character stri
ng.
```

In [ ]:

```
# Mushroom data preprocessing: converting single strings into numbers
label = []
data = []

for row_index, row in mshrm.iterrows():
    label.append(row.iloc[0])
    row_data = []
    for v in row.iloc[1:]:
        row_data.append(ord(v))
    data.append(row_data)

y_data = pd.Series(label)
X_data = pd.DataFrame(data)
print(y_data.shape, X_data.shape)
print(y_data.head())
print(X_data.head())
```

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X_data, y_data)

# Learning
mshrmRFC = RandomForestClassifier().fit(X_train, y_train)

# Prediction
predict = mshrmRFC.predict(X_test)

# Accuracy
ac_score = metrics.accuracy_score(y_test, predict)
print(ac_score)
confusion = metrics.confusion_matrix(y_true=y_test, y_pred=predict)
print(confusion)
cl_report = metrics.classification_report(y_test, predict)
print(cl_report)
print(mshrmRFC.feature_importances_)

# Apply Random Forest to Iris data
```