

```
In [13]: import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.neural_network import MLPClassifier
```

```
In [14]: xTestFile  ="Xtest.csv"
xTrainFile ="Xtrain.csv"
yTestFile  ="Ytest.csv"
yTrainFile ="Ytrain.csv"

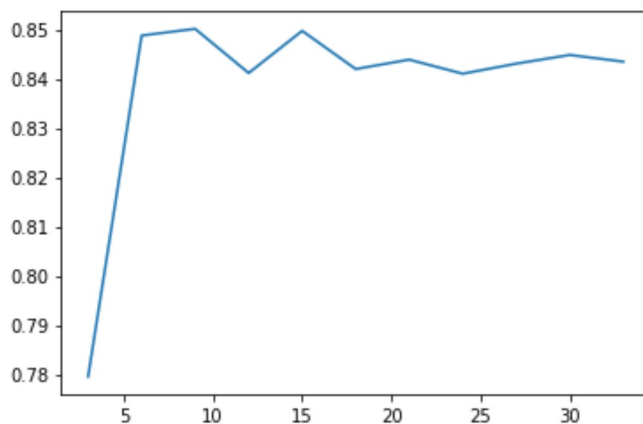
xTrain = pd.read_csv(xTrainFile).drop(["Unnamed: 0"], axis=1).values
xTest  = pd.read_csv(xTestFile).drop(["Unnamed: 0"], axis=1).values
yTrain = pd.read_csv(yTrainFile).drop(["Unnamed: 0"], axis=1).values
yTest  = pd.read_csv(yTestFile).drop(["Unnamed: 0"], axis=1).values
```

```
In [16]: #Random Forest
kFold = 5
param_grid = {'max_depth': np.arange(3,34,3)}
tree_grid = GridSearchCV(DecisionTreeClassifier(),param_grid,cv=kFold)

#Test using the training set
tree_grid.fit(xTrain,yTrain)
bestDepth = tree_grid.best_params_['max_depth']

accuracy = tree_grid.cv_results_['mean_test_score']
plt.plot(np.arange(3,34,3),accuracy)
print("Max Depth : %f" % bestDepth)
```

Max Depth : 9.000000



```
In [21]: AccScore = tree_grid.score(xTest,yTest)
print ("Accuracy Score is: %f"% AccScore)
```

Accuracy Score is: 0.867323

```
In [24]: #Neural Network
kFold = 5
param_grid = {'hidden_layer_sizes': [(100,), (200,), (300,), (400,), (500,)], 'max_iter': np.arange(100, 600, 100)}
mlp_grid = GridSearchCV(MLPClassifier(), param_grid, cv=kFold)

mlp_grid.fit(xTrain, yTrain)
bestHiddenLayer = mlp_grid.best_params_['hidden_layer_sizes']
bestIteration = mlp_grid.best_params_['max_iter']
```

```
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
    % self.max_iter, ConvergenceWarning)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
    % self.max_iter, ConvergenceWarning)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
    % self.max_iter, ConvergenceWarning)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
    % self.max_iter, ConvergenceWarning)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\Jiyoon\Anaconda3\lib\site-packages\sklearn\normal_network\multilayer_perceptron.py:921: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
```

```
In [31]: AccScore = mlp_grid.score(xTest,yTest)
print ("Accuracy Score is: %f"%AccScore)
print ("Optimal size of hidden layer"%bestHiddenLayer)
print ("Optimal number of epochs %f:"%bestIteration)
```

Accuracy Score is: 0.945029

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-31-a5c18dc7ee64> in <module>
      1 AccScore = mlp_grid.score(xTest,yTest)
      2 print ("Accuracy Score is: %f"%AccScore)
----> 3 print ("Optimal size of hidden layer"%bestHiddenLayer)
      4 print ("Optimal number of epochs %f:"%bestIteration)

TypeError: not all arguments converted during string formatting
```

```
In [33]: Neural Network shows higher accuracy than does the Decision Tree
Neural Network - 94.5%
Decision Tree - 86.7%
```