

```
In [51]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier, AdaBoostClassifier

data = pd.read_csv('C:/Users/USER/Desktop/MLENSEMBLESDOCS-02NOV2021/mnist/mnist_train.csv')
```

```
In [52]: df_X = data.iloc[:, 1:]

df_Y = data.iloc[:, 0]
```

```
In [53]: X_train, X_test, Y_train, Y_test = train_test_split(df_X, df_Y, test_size = 0.2, random_state = 4)
```

```
In [54]: data.shape
```

```
Out[54]: (60000, 785)
```

```
In [55]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60000 entries, 0 to 59999
Columns: 785 entries, label to 28x28
dtypes: int64(785)
memory usage: 359.3 MB
```

```
In [56]: #Decision Tree
dt = DecisionTreeClassifier()
dt.fit(X_train, Y_train)
```

```
Out[56]: DecisionTreeClassifier()
```

```
In [57]: dt.score(X_test, Y_test)
```

```
Out[57]: 0.8689166666666667
```

```
In [58]: #Testing for Model overfit on train data
dt.score(X_train, Y_train)

#A score of 1.0 implies that the model is 100% overfit to the training data.
```

Out[58]: 1.0

```
In [59]: #RandomForest - An ensemble of Decision Trees.  
rfc1 = RandomForestClassifier(n_estimators = 10)  
  
rfc1.fit(X_train, Y_train)
```

Out[59]: RandomForestClassifier(n_estimators=10)

```
In [60]: rfc1.score(X_test, Y_test)
```

Out[60]: 0.9383333333333334

```
In [61]: #Accuracy of DT Classifier = 0.8675  
  
#Accuracy of RandomForestClassifier(n_estimators=10) = 0.9425833333333333  
  
#i.e. there is almost 6% increase in accuracy by the usage of Random Forest Classifier over DT Classifier.
```

```
In [ ]: #Bagging Classifier. - Bootstrap Aggregation.  
#Create a Bagging classifier with a Decision Tree.  
  
bg = BaggingClassifier(DecisionTreeClassifier(), max_samples = 0.5, max_features = 1.0, n_estimators = 20)  
  
bg.fit(X_train, Y_train)  
  
#max_samples = 0.5 implies that each of our bag contains 50% of the training data  
#We can select some features at random, here we selected all features: max_features = 1.0  
#n_estimators = 20 : The number of Decision Trees we chose are 20.  
#We train the model on the training data set and see the accuracy score.  
  
print(bg.score(X_test, Y_test))
```

```
In [ ]: #Boosting - AdaBoost  
AdaBoost = AdaBoostClassifier(base_estimator = DecisionTreeClassifier(), n_estimators = 10, learning_rate = 1)  
  
AdaBoost.fit(X_train, Y_train)  
  
AdaBoost.score(X_test, Y_test)
```

```
In [ ]:
```

In []:

In []:

```
In [62]: from sklearn.ensemble import AdaBoostClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
```