Attrition Rate Analysis Decision Tree and RF Classification

Data Launching and Data Treatement:

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn import tree
Attrition_dataset = pd.read_csv("Attrition_Analysis.csv")
Attrition_dataset.head(2)
Out[15]:
  Age Attrition  ... YearsSinceLastPromotion YearsWithCurrManager
0  51        No  ...                       0                     0
1  31       Yes  ...                       1                     4

[2 rows x 24 columns]

Attrition_dataset.isnull().sum()
Out[16]:
Age                       0
Attrition                 0
BusinessTravel            0
Department                0
DistanceFromHome          0
Education                 0
EducationField            0
EmployeeCount             0
EmployeeID                0
```

```
Gender                    0
JobLevel                  0
JobRole                   0
MaritalStatus             0
MonthlyIncome             0
NumCompaniesWorked       19
Over18                    0
PercentSalaryHike         0
StandardHours             0
StockOptionLevel          0
TotalWorkingYears         9
TrainingTimesLastYear     0
YearsAtCompany            0
YearsSinceLastPromotion   0
YearsWithCurrManager      0
dtype: int64
```

```
Attrition_dataset.dtypes
Out[17]:
Age                       int64
Attrition                 object
BusinessTravel            object
Department                object
DistanceFromHome          int64
Education                 int64
EducationField            object
EmployeeCount             int64
EmployeeID                int64
Gender                    object
```

```
JobLevel                  int64
JobRole                  object
MaritalStatus            object
MonthlyIncome             int64
NumCompaniesWorked      float64
Over18                   object
PercentSalaryHike         int64
StandardHours             int64
StockOptionLevel          int64
TotalWorkingYears       float64
TrainingTimesLastYear     int64
YearsAtCompany            int64
YearsSinceLastPromotion   int64
YearsWithCurrManager      int64
dtype: object
```

```
Attrition_dataset['NumCompaniesWorked'].mean()
Out[18]: 2.6948303347756775
```

```
Attrition_dataset['TotalWorkingYears'].mean()
Out[19]: 11.279936378095888
```

```
Attrition_dataset = Attrition_dataset.fillna(Attrition_dataset.mean().round())
```

Encoding Categorical Variables:

```
label_encoder = preprocessing.LabelEncoder()
Attrition_dataset['Attrition']= label_encoder.fit_transform(Attrition_dataset['Attrition'])
Attrition_dataset['BusinessTravel']= label_encoder.fit_transform(Attrition_dataset['BusinessTravel'])
```

```
Attrition_dataset['Department']= label_encoder.fit_transform(Attrition_dataset['Department'])
Attrition_dataset['EducationField']= label_encoder.fit_transform(Attrition_dataset['EducationField'])
Attrition_dataset['Gender']= label_encoder.fit_transform(Attrition_dataset['Gender'])
Attrition_dataset['JobRole']= label_encoder.fit_transform(Attrition_dataset['JobRole'])
Attrition_dataset['MaritalStatus']= label_encoder.fit_transform(Attrition_dataset['MaritalStatus'])
```

Random Forest Algorithm to find imp Variables:

```
Attrition_dataset.columns
Out[22]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',
       'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
features = ['Age', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender',
       'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'PercentSalaryHike',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']


from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators= 1000, max_features= 2, oob_score=True)
rf_model.fit(X= Attrition_dataset[features], y= Attrition_dataset['Attrition'])
```

Out[23]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features=2,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=1000,
                       n_jobs=None, oob_score=True, random_state=None,
                       verbose=0, warm_start=False)
```

```
print("RF Model Accuracy:", rf_model.oob_score_)
```

RF Model Accuracy: 1.0

```
for feature,imp in zip(features,rf_model.feature_importances_):
    print(feature,imp)
```

Age 0.09757303206804788

BusinessTravel 0.02804982021856164

Department 0.026028826146555425

DistanceFromHome 0.06921050482762917

Education 0.04109189659639124

EducationField 0.041026886637150005

Gender 0.018315056318010236

JobLevel 0.037713801498944856

JobRole 0.056155556979458315

MaritalStatus 0.03929196041694974

MonthlyIncome 0.09540104858060328

NumCompaniesWorked 0.055591908126381234

PercentSalaryHike 0.06532596856696687

StockOptionLevel 0.03410877365115445

TotalWorkingYears 0.08518050196317473

TrainingTimesLastYear 0.04445830380709171

YearsAtCompany 0.06955229301918192

YearsSinceLastPromotion 0.04289525033397936

YearsWithCurrManager 0.05302861024376809

Generating Decision Tree Model:

```
predictors = ['Age', 'MonthlyIncome', 'TotalWorkingYears']
tree_model = tree.DecisionTreeClassifier(max_depth= 6, max_leaf_nodes= 10)
tree_model.fit(X= Attrition_dataset[predictors], y= Attrition_dataset['Attrition'])
```

Out[26]:

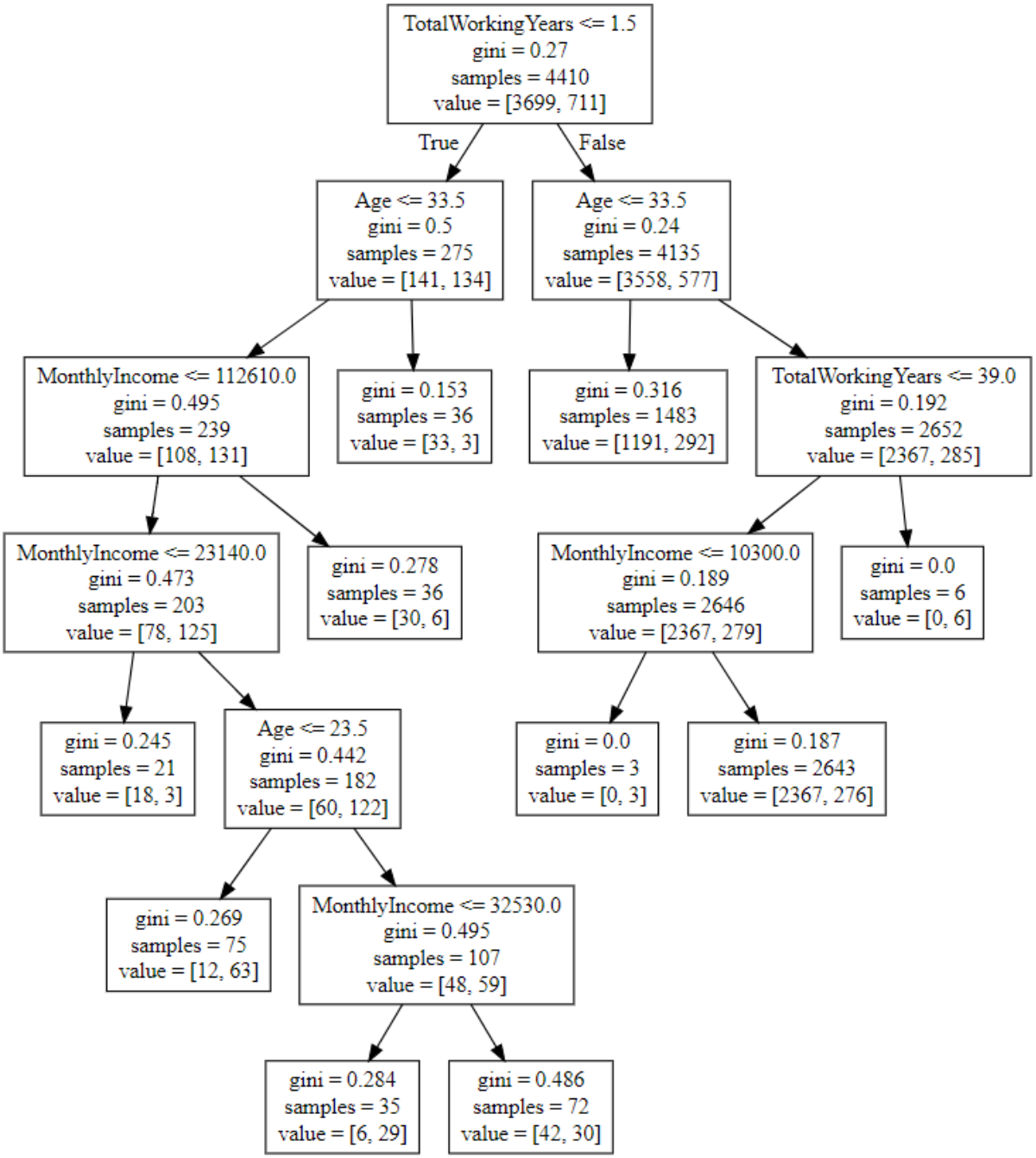
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=6, max_features=None, max_leaf_nodes=10,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

```
with open("Attrition_DTree1.dot","w") as f:
    f = tree.export_graphviz(tree_model,feature_names= ['Age', 'MonthlyIncome', 'TotalWorkingYe ars'] , out_file=f)
```

```
print("DTree Model Accuracy:", tree_model.score(X= Attrition_dataset[predictors], y= Attrition_dataset['Attrition']))
```

DTree Model Accuracy: 0.8575963718820862

Decision Tree:



Rules:

Attrition- NO

1. If Total Working years is less than 1.5 and age is greater than 33.5, then there is low probability of Attrition
2. If Total Working years is less than 1.5, age is less than 33.5 and Monthly Income greater than 112610, then there is low probability of Attrition
3. If Total Working years is less than 1.5, age is less than 33.5 and Monthly Income less than 23140, then there is low probability of Attrition
4. If Total Working years is less than 1.5, age is in range 23.5 to 33.5 and Monthly Income is in range 32530 to 112610, then there is low probability of Attrition
5. If Total Working years is greater than 1.5 and age is less than 33.5, then there is low probability of Attrition
6. If Total Working years is in range of 1.5 to 39, age is greater than 33.5 and Monthly income greater than 10300, then there is low probability of Attrition

Attrition- YES

1. If Total Working years is greater than 39 and age is greater than 33.5, then there is high probability of Attrition
2. If Total Working years is in range of 1.5 to 39, age is greater than 33.5 and Monthly income less than 10300, then there is high probability of Attrition
3. If Total Working years is less than 1.5, age is less than 23.5 and Monthly Income is in range 23140 to 112160, then there is high probability of Attrition
4. If Total Working years is less than 1.5, age is in range 23.5 to 33.5 and Monthly Income is in range 23140 to 32530, then there is high probability of Attrition

Inference:

1. Based on the importance value generated with Random forest algorithm, it is seen that the features **'Age', 'MonthlyIncome', and 'TotalWorkingYears'** are more significant for decision tree generation.
2. Increasing the no. of significant features and max-depth, increases the accuracy of the model. But the Decision tree becomes complex and overfitted.
3. Decision tree generated with these features and max-depth of 6 and 10 leaf nodes provides **85.76%** accuracy in classifying the record as Attrition(Y/N)