Bank Loan Analysis DTree and RF Classification

Data Launching and Data Treatment:

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn import tree
loan_data = pd.read_excel("Bank_Personal_Loan_Modelling.xlsx", sheet_name= "Data")
loan_data.head(2)
Out[30]:
   ID  Age  Experience  ...  CD Account  Online  CreditCard
0   1   25           1  ...           0       0           0
1   2   45          19  ...           0       0           0

[2 rows x 14 columns]

loan_data.isna().sum()
Out[31]:
ID                  0
Age                 0
Experience          0
Income              0
ZIP Code            0
Family              0
CCAvg               0
Education           0
Mortgage            0
```

```
Personal Loan        0
Securities Account   0
CD Account           0
Online               0
CreditCard           0
dtype: int64

loan_data.columns
Out[32]:
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
```

Random Forest Algorithm to find imp Variables:

```
from sklearn.ensemble import RandomForestClassifier
features = ['Age', 'Experience', 'Income', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Securities Account',
       'CD Account', 'Online', 'CreditCard']
rf_model = RandomForestClassifier(n_estimators= 1000, max_features= 2, oob_score= True)
rf_model.fit(X= loan_data[features], y = loan_data['Personal Loan'])
Out[33]:
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: 0.9876

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

Age 0.044558101231166485
Experience 0.044094972870022286
Income 0.3423930953690072
Family 0.09992091608770354
CCAvg 0.17749688920649698
Education 0.16856727513885572
Mortgage 0.0447311616825908
Securities Account 0.005781446102837379
CD Account 0.054289942682517835
Online 0.0083073338401806
CreditCard 0.009858865788621314
```

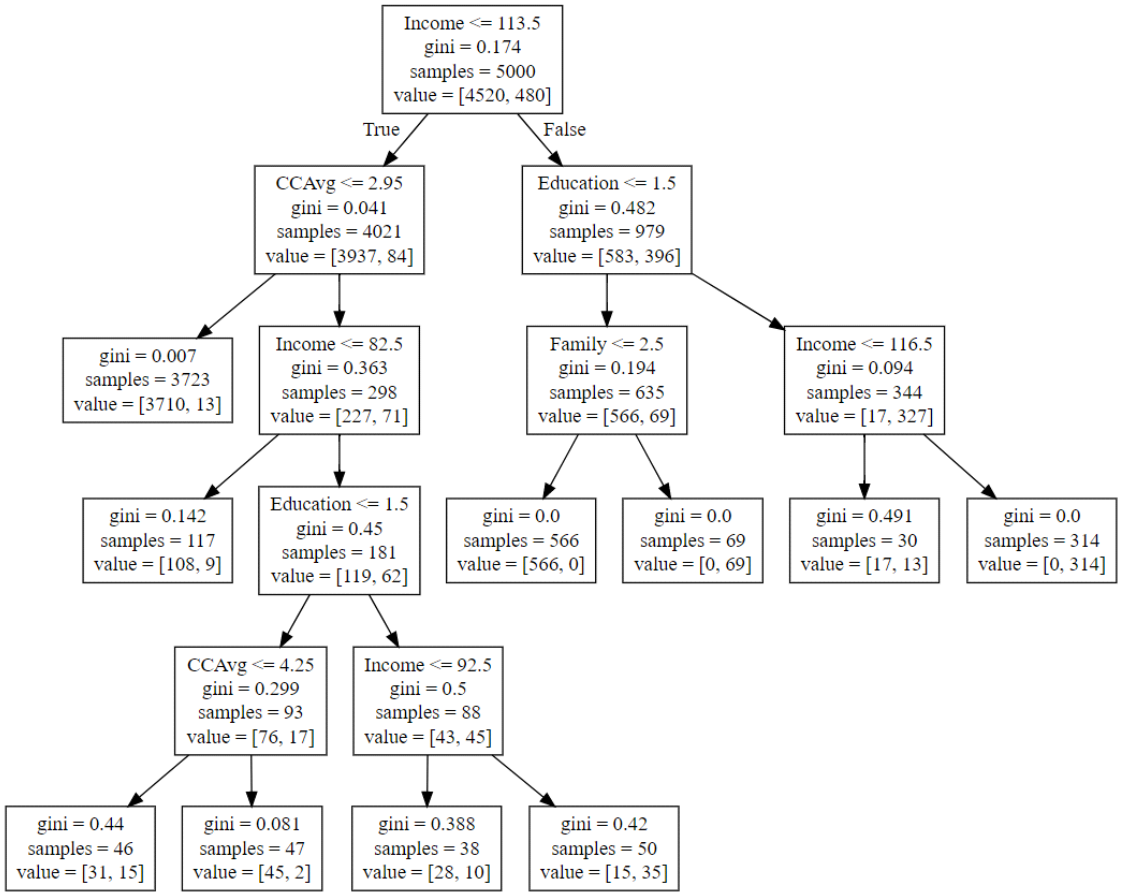
Generating Decision Tree Model:

```
predictors = loan_data[['Income','Family','CCAvg','Education']]
tree_model = tree.DecisionTreeClassifier(max_depth= 8, max_leaf_nodes= 10)
tree_model.fit(X= predictors, y = loan_data['Personal Loan'])
Out[36]:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
```

```
                max_depth=8, 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("Loan_Dtree1.dot","w") as f:
f = tree.export_graphviz(tree_model, feature_names=['Income','Family','CCAvg','Education'], out_file=f)

print("DTree Model Accuracy:", tree_model.score(X= predictors, y = loan_data['Personal Loan']))
DTree Model Accuracy: 0.9846
```

Decision Tree:



Rules:

LOAN - NO

- If the CCAvg is less than 2.95 and the Income is less than 113.5, then probability of Loan(No) is high
- If the CCAvg is greater than 2.95 and the Income is less than 82.5, then probability of Loan(No) is high
- If Education is less than 1.5, CCAvg is in the range of 2.95 to 4.25 and Income is in the range of 82.5 to 113.5, then probability of Loan(No) is high
- If Education is less than 1.5, CCAvg is greater than 4.25 and Income is in the range of 82.5 to 113.5, then probability of Loan(No) is high
- If Education is greater than 1.5, CCAvg is greater than 2.95 and Income is in the range of 82.5 to 92.5, then probability of Loan(No) is high
- If Income is greater than 113.5, Education is less than 1.5 and Family less than 2.5, then probability of Loan(No) is high

LOAN - YES

- If Education is greater than 1.5, CCAvg is greater than 2.95 and Income is in the range of 92.5 to 113.5, then probability of Loan(Yes) is high
- If Income is greater than 113.5, Education is less than 1.5 and Family greater than 2.5, then probability of Loan(Yes) is high
- If Income is in range of 113.5 to 116.5 and Education is greater than 1.5, then probability of Loan(Yes) is almost equal
- If Income is greater than 116.5, Education is greater than 1.5, then probability of Loan(Yes) is high

Inference:

- Based on the importance value generated with Random forest algorithm, it is seen that the features **'Income', 'Family', 'CCAvg' and 'Education'** are more significant for decision tree generation.
- Decision tree generated with these features and max-depth of 8 and 10 leaf nodes provides **98.46%** accuracy in classifying the record as Personal Loan(Y/N)