Random Forest and Decision Tree of Bank Loan Data set

```
import pandas as pd

dataset=pd.read_excel("Bank_Personal_Loan_Modelling.xlsx",sheet_name=1)

dataset.columns
Out[5]:
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')

dataset1=dataset.drop(["ID","ZIP Code"],axis=1)

dataset2=dataset1.dropna()

dataset3=dataset2.drop_duplicates()

from sklearn.ensemble import  RandomForestClassifier

import numpy as np

dataset3["CCAvg"]=np.round(dataset3["CCAvg"])
__main__:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
rf_model=RandomForestClassifier(n_estimators=1000,max_features=2,oob_score=True)
```

```
features=['Age', 'Experience', 'Income', 'Family', 'CCAvg','Education', 'Mortgage', 'Securities Account','CD Account', 'Online', 'CreditCard']
```

```
rf_model.fit(X=dataset3[features],y=dataset3["Personal Loan"])
```

Out[14]:

```
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("OOB Accuracy")
```

OOB Accuracy

```
print(rf_model.oob_score_)
```

0.9859635051132946

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

Age 0.05078435540614227

Experience 0.05079368390317876

Income 0.36020544674410854

Family 0.10054472691540015

CCAvg 0.14035031046155444

Education 0.16919474081242009

Mortgage 0.047332554430923754

Securities Account 0.006121588396168966

CD Account 0.05434444075468111

Online 0.009447691891006398

CreditCard 0.010880460284415597

```
from sklearn import  tree

tree_model=tree.DecisionTreeClassifier()

tree_model=tree.DecisionTreeClassifier(max_depth=6,max_leaf_nodes=10)

predictors=pd.DataFrame([dataset3["Education"],dataset3["CCAvg"],dataset3[
"Income"]]).T

tree_model.fit(X=predictors,y=dataset3["Personal Loan"])
```

Out[23]:

```
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("Dtree.dot","w") as f:

f=tree.export_graphviz(tree_model,feature_names=["Education","CCAvg","Income"],out_file=f);