In [1]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
df=pd.read_excel(r'C:\Users\siyad\AppData\Local\Temp\Temp1_Dataset-20200813T141334Z-001.zip\Dataset\Bank_Personal_Loan_Modelling.xlsx',sheet_name='Data')
```

In [5]:
```
df.head()
```

Out[5]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1

In [6]:
```
df.columns
```

Out[6]:
```
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
```

In [7]:
```
df1=df.drop(['ID','ZIP Code'],axis=1)
```

In [8]:
```
df1.head()
```

Out[8]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	1	49	4	1.6	1	0	0	1	0	0	0
1	45	19	34	3	1.5	1	0	0	1	0	0	0
2	39	15	11	1	1.0	1	0	0	0	0	0	0
3	35	9	100	1	2.7	2	0	0	0	0	0	0
4	35	8	45	4	1.0	2	0	0	0	0	0	1

In [9]:
```
df1.isnull().sum()
```

Out[9]:
```
Age                 0
Experience          0
Income              0
Family              0
CCAvg               0
Education           0
Mortgage            0
Personal Loan       0
Securities Account  0
CD Account          0
Online              0
CreditCard          0
dtype: int64
```

In [10]:
```
df1.duplicated().sum()
```

Out[10]:
```
13
```

In [11]:
```
df2=df1.drop_duplicates()
```

In [12]:
```
df2.shape
```

Out[12]:
```
(4987, 12)
```

In [13]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [15]:
```
df2['CCAvg']=np.round(df2['CCAvg'])
```

```
c:\python36\lib\site-packages\ipykernel_launcher.py: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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

In [16]:
```
df2.head()
```

Out[16]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	1	49	4	2.0	1	0	0	1	0	0	0
1	45	19	34	3	2.0	1	0	0	1	0	0	0
2	39	15	11	1	1.0	1	0	0	0	0	0	0
3	35	9	100	1	3.0	2	0	0	0	0	0	0
4	35	8	45	4	1.0	2	0	0	0	0	0	1

In [17]:
```
rf_model=RandomForestClassifier(n_estimators=1000,max_features=2,oob_score=True)
```

In [19]:
```
features= ['Age', 'Experience', 'Income', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Securities Account',
       'CD Account', 'Online', 'CreditCard']
```

In [20]:
```
rf_model.fit(X=df2[features],y=df2['Personal Loan'])
```

Out[20]:
```
RandomForestClassifier(max_features=2, n_estimators=1000, oob_score=True)
```

In [21]:
```
print('OOB Score: ')
print(rf_model.oob_score_)
```

```
OOB Score:
0.986164026468819
```

In [22]:
```
for feature,imp in zip(features,rf_model.feature_importances_):
    print(feature,imp)
```

```
Age 0.050485268178769305
Experience 0.05077254724221042
Income 0.3584883339626149
Family 0.09990993464969712
CCAvg 0.13932928043077383
Education 0.17006581507047963
Mortgage 0.047443829473937195
Securities Account 0.006289596037929927
CD Account 0.056560914325943984
Online 0.00949584562388546
CreditCard 0.011158635003758095
```

In [23]:
```
from sklearn import tree
```

In [26]:
```
tree_model=tree.DecisionTreeClassifier(max_depth=6,max_leaf_nodes=10)
```

In [28]:
```
pred=pd.DataFrame([df2['Income'],df2['Education'],df2['CCAvg']]).T
```

In [29]:
```
tree_model.fit(X=pred,y=df2['Personal Loan'])
```

Out[29]:
```
DecisionTreeClassifier(max_depth=6, max_leaf_nodes=10)
```

In [31]:
```
with open('Dtree.dot','w') as f:
    f=tree.export_graphviz(tree_model,feature_names=['Income','Education','CCAvg'],out_file=f)
```

In [32]:
```
df2.shape
```

Out[32]:
```
(4987, 12)
```

In [33]:
```
tree_model.score(X=pred,y=df2['Personal Loan'])
```

Out[33]: 0.970723882093443

In [34]:
```
import statsmodels.api as sm
```

In [35]:
```
Y=df2['Personal Loan']
```

In [36]:
```
X=df2[['Age', 'Experience', 'Income', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Securities Account',
       'CD Account', 'Online', 'CreditCard']]
```

In [37]:
```
X1=sm.add_constant(X)
```

In [38]:
```
Bankloan=sm.Logit(Y,X1)
```

In [39]:
```
result=Bankloan.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.128909
         Iterations 9
```

In [40]:
```
result.summary()
```

Out[40]:

Logit Regression Results

Dep. Variable:	Personal Loan	**No. Observations:**	4987
Model:	Logit	**Df Residuals:**	4975
Method:	MLE	**Df Model:**	11
Date:	Sat, 15 Aug 2020	**Pseudo R-squ.:**	0.5930
Time:	01:08:15	**Log-Likelihood:**	-642.87
converged:	True	**LL-Null:**	-1579.7
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-12.1436	1.646	-7.379	0.000	-15.369	-8.918
Age	-0.0537	0.061	-0.874	0.382	-0.174	0.067
Experience	0.0634	0.061	1.039	0.299	-0.056	0.183
Income	0.0549	0.003	20.964	0.000	0.050	0.060
Family	0.6945	0.074	9.355	0.000	0.549	0.840
CCAvg	0.1078	0.038	2.801	0.005	0.032	0.183
Education	1.7279	0.115	15.075	0.000	1.503	1.953
Mortgage	0.0005	0.001	0.817	0.414	-0.001	0.002
Securities Account	-0.9325	0.286	-3.266	0.001	-1.492	-0.373
CD Account	3.8189	0.324	11.802	0.000	3.185	4.453
Online	-0.6706	0.157	-4.271	0.000	-0.978	-0.363
CreditCard	-1.1168	0.205	-5.450	0.000	-1.518	-0.715

In [ ]:
```

```