

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv("C:\\Users\\Admin\\Downloads\\Assignment 7\\Fraud_check.csv")
df.head()
```

```
Out[2]:
```

	Undergrad	Marital.Status	Taxable.Income	City.Population	Work.Experience	Urban
0	NO	Single	68833	50047	10	YES
1	YES	Divorced	33700	134075	18	YES
2	NO	Married	36925	160205	30	YES
3	YES	Single	50190	193264	15	YES
4	NO	Married	81002	27533	28	NO

```
In [3]: df=pd.get_dummies(df,columns=['Undergrad','Marital.Status','Urban'], drop_first=True)
```

```
In [4]: df["TaxInc"] = pd.cut(df["Taxable.Income"], bins = [10002,30000,99620], labels = ["Risk
```

```
In [5]: df = pd.get_dummies(df,columns = ["TaxInc"],drop_first=True)
```

```
In [6]: df.tail(10)
```

```
Out[6]:
```

	Taxable.Income	City.Population	Work.Experience	Undergrad_YES	Marital.Status_Married	Marital.S
590	43018	85195	14	0	1	
591	27394	132859	18	1	0	
592	68152	75143	16	1	0	
593	84775	131963	10	0	0	
594	47364	97526	9	0	1	
595	76340	39492	7	1	0	
596	69967	55369	2	1	0	
597	47334	154058	0	0	0	
598	98592	180083	17	1	1	
599	96519	158137	16	0	0	

```
In [7]: def norm_func(i):
x = (i-i.min())/(i.max()-i.min())
return (x)
```

```
In [8]: df_norm = norm_func(df.iloc[:,1:])
df_norm.tail(10)
```

```
Out[8]:
```

	City.Population	Work.Experience	Undergrad_YES	Marital.Status_Married	Marital.Status_Single	Urt
590	0.341473	0.466667	0.0	1.0	0.0	
591	0.615406	0.600000	1.0	0.0	1.0	
592	0.283703	0.533333	1.0	0.0	1.0	
593	0.610256	0.333333	0.0	0.0	0.0	
594	0.412341	0.300000	0.0	1.0	0.0	
595	0.078811	0.233333	1.0	0.0	0.0	
596	0.170058	0.066667	1.0	0.0	0.0	
597	0.737240	0.000000	0.0	0.0	0.0	
598	0.886810	0.566667	1.0	1.0	0.0	
599	0.760683	0.533333	0.0	0.0	0.0	

```
In [9]: X = df_norm.drop(['TaxInc_Good'], axis=1)
y = df_norm['TaxInc_Good']
```

```
In [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
In [19]: from sklearn.ensemble import RandomForestClassifier
```

```
In [28]: forest = RandomForestClassifier(n_estimators=200,max_depth=None,min_samples_split=2,cri
forest.fit(Xtrain, ytrain)
```

```
Out[28]: RandomForestClassifier(n_estimators=200)
```

```
In [29]: print('Train accuracy: {}'.format(forest.score(Xtrain, ytrain)))
print('Test accuracy: {}'.format(forest.score(Xtest, ytest)))
```

```
Train accuracy: 1.0
Test accuracy: 0.725
```

```
In [24]: forest_new = RandomForestClassifier(n_estimators=100,max_depth=10,min_samples_split=20,
forest_new.fit(Xtrain, ytrain)
```

```
Out[24]: RandomForestClassifier(max_depth=10, min_samples_split=20)
```

```
In [25]: print('Train accuracy: {}'.format(forest_new.score(Xtrain, ytrain)))  
         print('Test accuracy: {}'.format(forest_new.score(Xtest, ytest)))
```

```
Train accuracy: 0.8020833333333334  
Test accuracy: 0.7916666666666666
```

```
In [ ]:
```