HR Data Analysis and Modeling

May 14, 2024

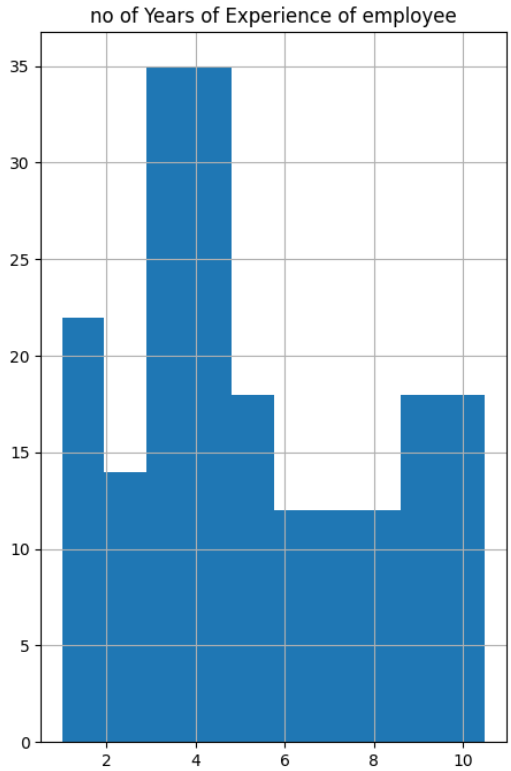
[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

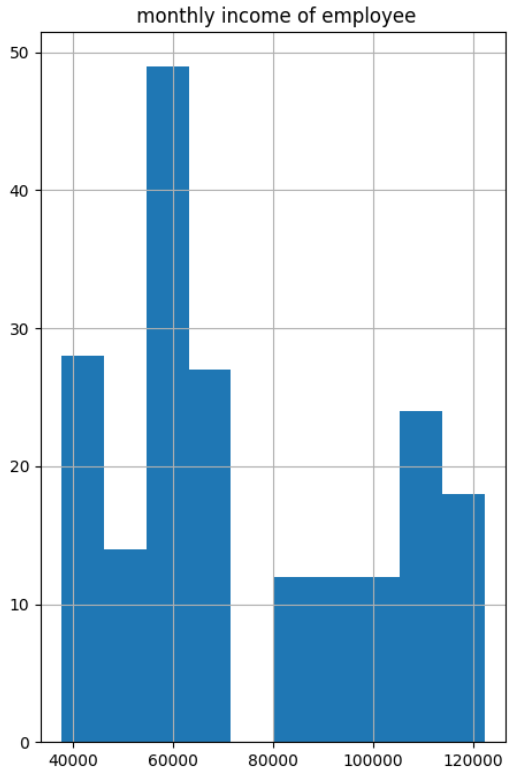
[2]:
```
df=pd.read_csv("C:HR_DT.csv")
```

[3]:
```
df.isna().sum()
df.drop_duplicates()
df.isnull().sum()
df.dropna()
df.dtypes
df.shape
```

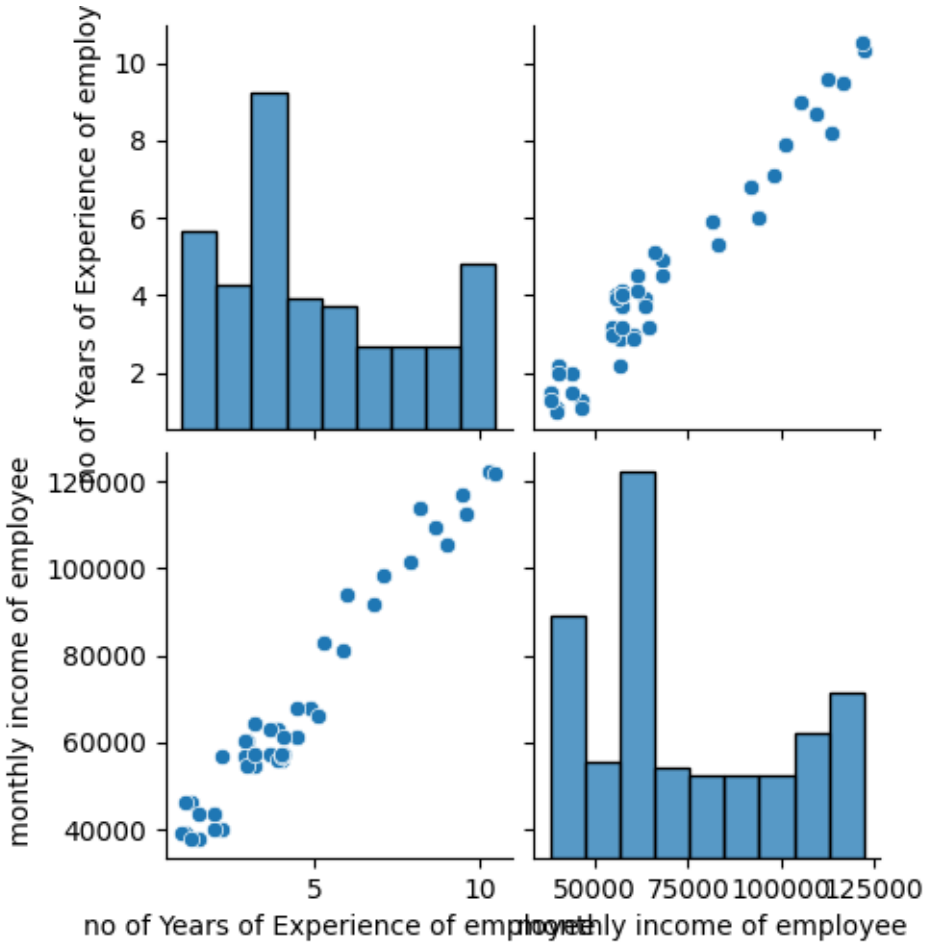
[3]:
```
(196, 3)
```

[4]:
```
# Histogram for each numerical feature
df.hist(figsize=(12, 8))
plt.show()
```



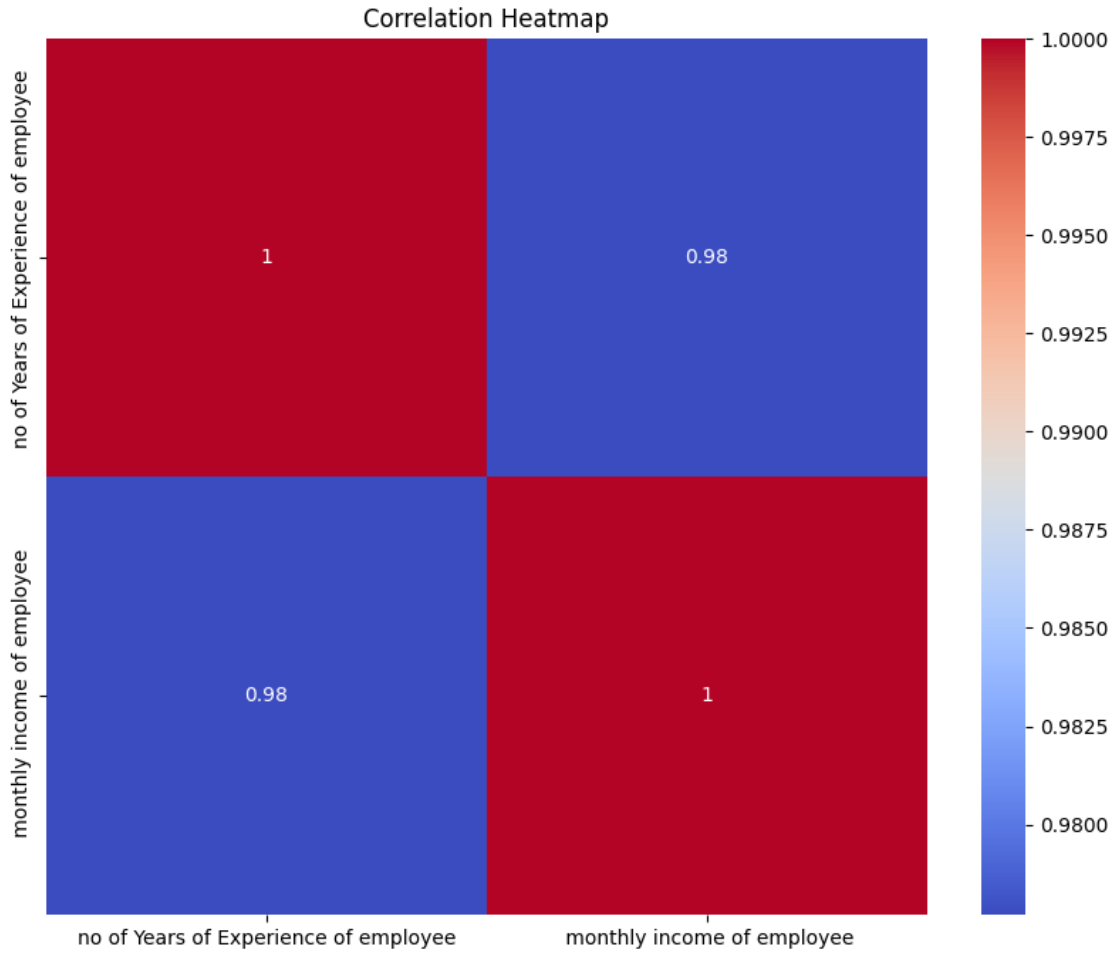


[5]:
```
# Kernel Density Estimation (KDE) plot for numerical features
sns.pairplot(df)
plt.show()
```



[6]:
```
# Heatmap to visualize correlation between features
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

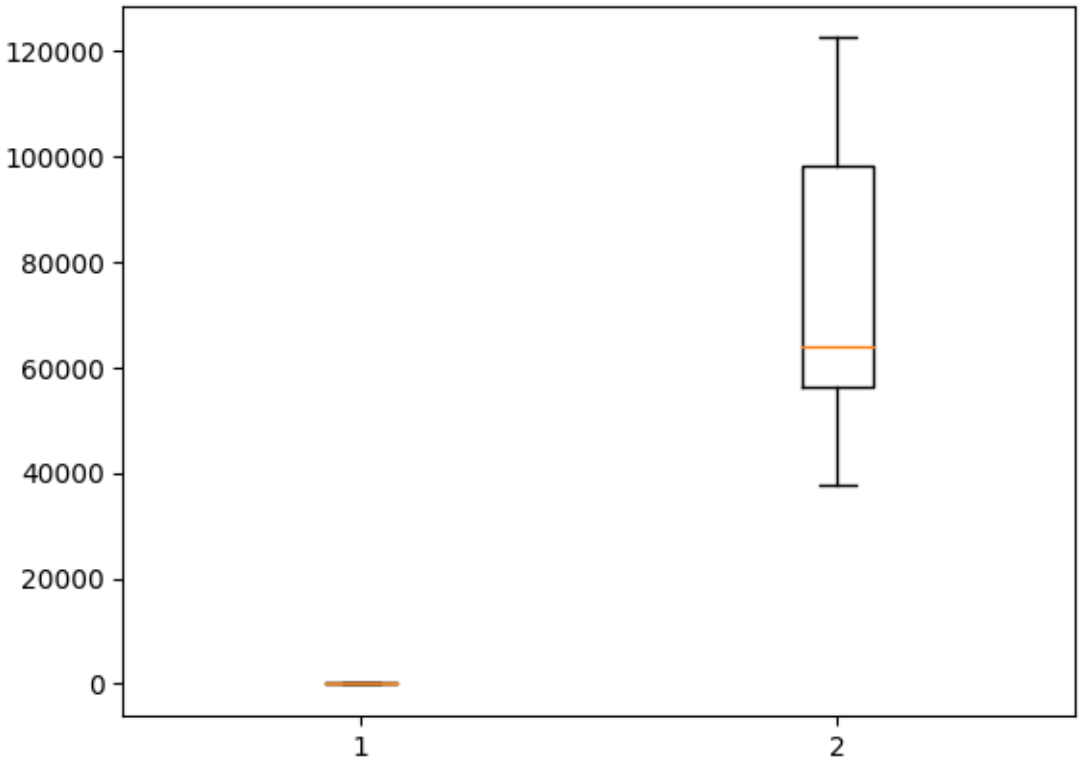
```
C:\Users\Ravi\AppData\Local\Temp\ipykernel_9420\1535863314.py:3: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```



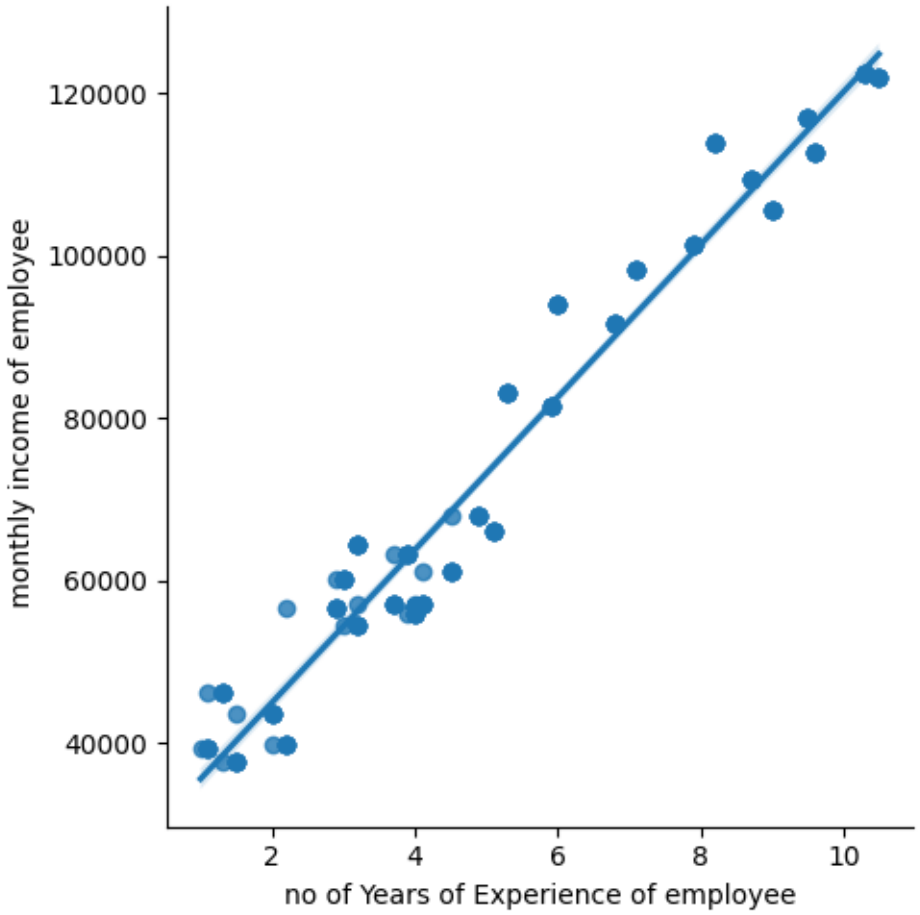
[7]:
```
df.columns
plt.boxplot(df.iloc[:,1:])
```

[7]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x211f24b7a60>,
  <matplotlib.lines.Line2D at 0x211f24b7d00>,
  <matplotlib.lines.Line2D at 0x211f24f06d0>,
  <matplotlib.lines.Line2D at 0x211f24f0970>],
 'caps': [<matplotlib.lines.Line2D at 0x211f23f8d00>,
  <matplotlib.lines.Line2D at 0x211f24b57e0>,
  <matplotlib.lines.Line2D at 0x211f24f0c10>,
  <matplotlib.lines.Line2D at 0x211f24f0eb0>],
 'boxes': [<matplotlib.lines.Line2D at 0x211f24b77c0>,
  <matplotlib.lines.Line2D at 0x211f24f0430>],
 'medians': [<matplotlib.lines.Line2D at 0x211f24b7eb0>,
  <matplotlib.lines.Line2D at 0x211f24f1150>],
 'fliers': [<matplotlib.lines.Line2D at 0x211f24f0190>,
  <matplotlib.lines.Line2D at 0x211f24f13f0>],
```

```
'means': []}
```



[8]:
```
sns.lmplot(y=' monthly income of employee', x='no of Years of Experience of␣
↪employee', data=df)
plt.show()          #df1=pd.get_dummies(df['Position of the␣
↪employee'],drop_first =True)
```



[9]:
```
# Encoding categorical variable 'Position of the employee' using LabelEncoder
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Position of the employee'] = le.fit_transform(df['Position of the␣
↪employee'])
```

[ ]:
```
```

[10]:
```
def norm_func(i):
        x = (i-i.min())        /(i.max()-i.min())
        return(x)
```

[11]:
```
df_norm=norm_func(df.loc[:, df.columns!=" monthly income of employee"])
```

[12]:
```
predict=df_norm.loc[:, df_norm.columns!=' monthly income of employee']
target=df[' monthly income of employee']
```

[13]:
```
x=df
```

[14]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(predict,target,test_size=0.3)
```

[15]:
```
from sklearn.tree import DecisionTreeRegressor as DT
```

[16]:
```
model = DT(min_samples_split = 5)

model.fit(x_train,y_train)
```

[16]:
```
DecisionTreeRegressor(min_samples_split=5)
```

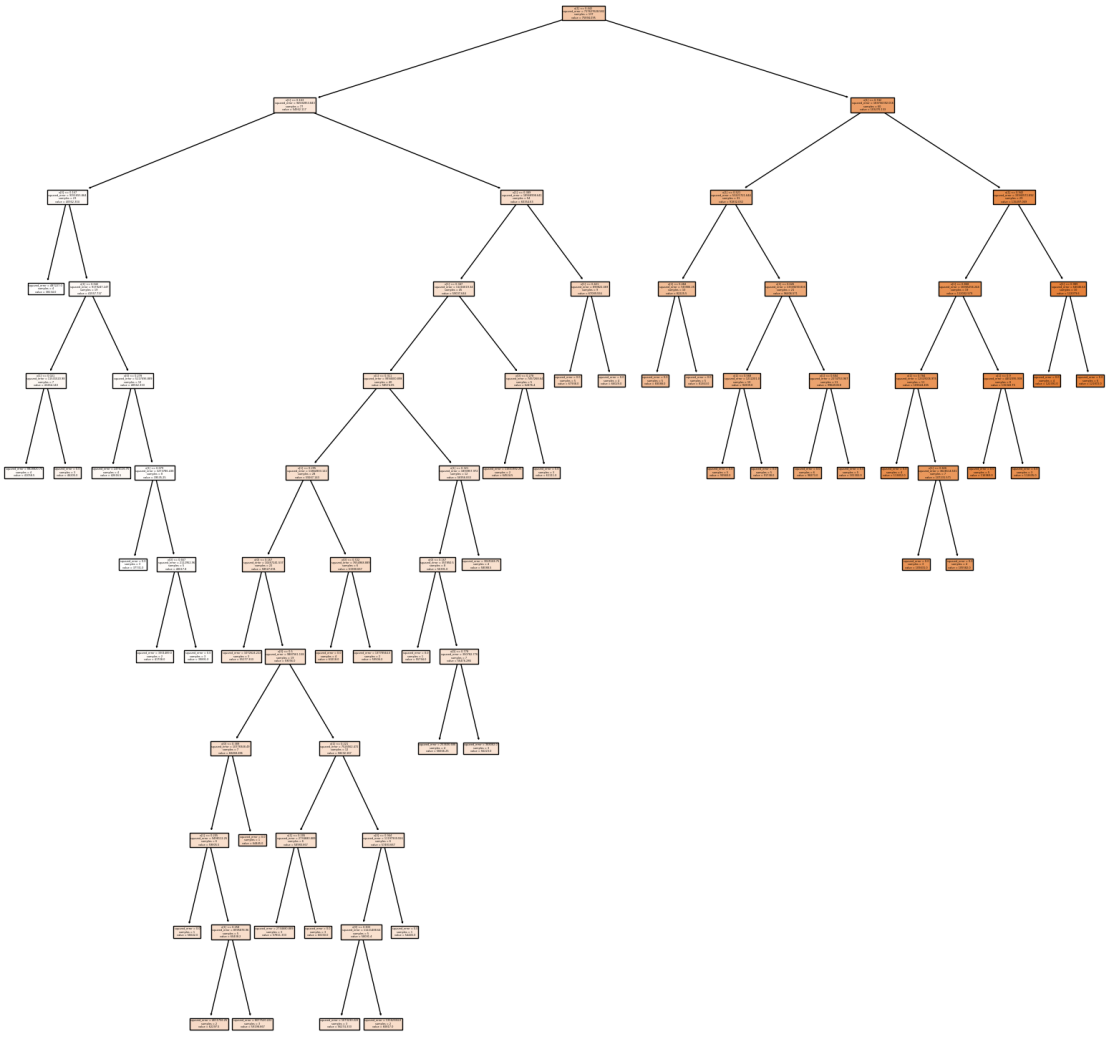
[17]:
```
from sklearn import tree
import matplotlib.pyplot as plt
plt.figure(figsize=(20,20))
tree.plot_tree(model,filled=True)
```

[17]:
```
[Text(0.5238970588235294, 0.9583333333333334, 'x[1] <= 0.442\nsquared_error =
717627628.582\nsamples = 137\nvalue = 75894.095'),
 Text(0.26348039215686275, 0.875, 'x[1] <= 0.163\nsquared_error =
94592853.843\nsamples = 77\nvalue = 54562.117'),
 Text(0.058823529411764705, 0.7916666666666666, 'x[0] <= 0.167\nsquared_error =
9351655.864\nsamples = 23\nvalue = 40962.304'),
 Text(0.0392156862745098, 0.7083333333333334, 'squared_error = 487227.0\nsamples
= 4\nvalue = 38134.0'),
 Text(0.0784313725490196, 0.7083333333333334, 'x[1] <= 0.042\nsquared_error =
9179247.247\nsamples = 19\nvalue = 41557.737'),
 Text(0.0392156862745098, 0.625, 'x[1] <= 0.021\nsquared_error =
11531520.98\nsamples = 7\nvalue = 43264.143'),
 Text(0.0196078431372549, 0.5416666666666666, 'squared_error =
8828820.75\nsamples = 4\nvalue = 41058.5'),
 Text(0.058823529411764705, 0.5416666666666666, 'squared_error = 0.0\nsamples =
3\nvalue = 46205.0'),
 Text(0.11764705882352941, 0.625, 'x[0] <= 0.278\nsquared_error =
5117696.889\nsamples = 12\nvalue = 40562.333'),
 Text(0.09803921568627451, 0.5416666666666666, 'squared_error =
2476116.75\nsamples = 4\nvalue = 42616.5'),
 Text(0.13725490196078433, 0.5416666666666666, 'x[1] <= 0.079\nsquared_error =
3273786.438\nsamples = 8\nvalue = 39535.25'),
 Text(0.11764705882352941, 0.4583333333333333, 'squared_error = 0.0\nsamples =
3\nvalue = 37731.0'),
 Text(0.1568627450980392, 0.4583333333333333, 'x[0] <= 0.667\nsquared_error =
2112952.96\nsamples = 5\nvalue = 40617.8'),
 Text(0.13725490196078433, 0.375, 'squared_error = 3301489.0\nsamples = 2\nvalue
= 41708.0'),
 Text(0.17647058823529413, 0.375, 'squared_error = 0.0\nsamples = 3\nvalue =
39891.0'),
 Text(0.4681372549019608, 0.7916666666666666, 'x[1] <= 0.389\nsquared_error =
```

```
18568998.641\nsamples = 54\nvalue = 60354.63'),
 Text(0.4068627450980392, 0.7083333333333334, 'x[1] <= 0.347\nsquared_error =
11216619.34\nsamples = 45\nvalue = 59007.644'),
 Text(0.3431372549019608, 0.625, 'x[1] <= 0.311\nsquared_error =
9994500.898\nsamples = 40\nvalue = 58574.05'),
 Text(0.27450980392156865, 0.5416666666666666, 'x[1] <= 0.295\nsquared_error =
11864903.122\nsamples = 28\nvalue = 59267.143'),
 Text(0.23529411764705882, 0.4583333333333333, 'x[0] <= 0.167\nsquared_error =
10457241.537\nsamples = 22\nvalue = 58527.091'),
 Text(0.21568627450980393, 0.375, 'squared_error = 1072624.222\nsamples =
3\nvalue = 55177.333'),
 Text(0.2549019607843137, 0.375, 'x[0] <= 0.5\nsquared_error =
9887561.158\nsamples = 19\nvalue = 59056.0'),
 Text(0.20588235294117646, 0.2916666666666667, 'x[0] <= 0.389\nsquared_error =
10776948.49\nsamples = 7\nvalue = 60468.286'),
 Text(0.18627450980392157, 0.20833333333333334, 'x[1] <= 0.205\nsquared_error =
9498112.25\nsamples = 6\nvalue = 59805.5'),
 Text(0.16666666666666666, 0.125, 'squared_error = 0.0\nsamples = 1\nvalue =
56642.0'),
 Text(0.20588235294117646, 0.125, 'x[1] <= 0.258\nsquared_error =
8995878.96\nsamples = 5\nvalue = 60438.2'),
 Text(0.18627450980392157, 0.041666666666666664, 'squared_error =
4611756.25\nsamples = 2\nvalue = 62297.5'),
 Text(0.22549019607843138, 0.041666666666666664, 'squared_error =
8077520.222\nsamples = 3\nvalue = 59198.667'),
 Text(0.22549019607843138, 0.20833333333333334, 'squared_error = 0.0\nsamples =
1\nvalue = 64445.0'),
 Text(0.30392156862745096, 0.2916666666666667, 'x[1] <= 0.221\nsquared_error =
7526562.472\nsamples = 12\nvalue = 58232.167'),
 Text(0.2647058823529412, 0.20833333333333334, 'x[1] <= 0.205\nsquared_error =
2734680.889\nsamples = 6\nvalue = 58980.667'),
 Text(0.24509803921568626, 0.125, 'squared_error = 2734680.889\nsamples =
3\nvalue = 57811.333'),
 Text(0.28431372549019607, 0.125, 'squared_error = 0.0\nsamples = 3\nvalue =
60150.0'),
 Text(0.3431372549019608, 0.20833333333333334, 'x[0] <= 0.944\nsquared_error =
11197939.556\nsamples = 6\nvalue = 57483.667'),
 Text(0.3235294117647059, 0.125, 'x[0] <= 0.833\nsquared_error =
11221488.64\nsamples = 5\nvalue = 58091.4'),
 Text(0.30392156862745096, 0.041666666666666664, 'squared_error =
1673230.222\nsamples = 3\nvalue = 56274.333'),
 Text(0.3431372549019608, 0.041666666666666664, 'squared_error =
13162384.0\nsamples = 2\nvalue = 60817.0'),
 Text(0.3627450980392157, 0.125, 'squared_error = 0.0\nsamples = 1\nvalue =
54445.0'),
 Text(0.3137254901960784, 0.4583333333333333, 'x[0] <= 0.722\nsquared_error =
7654968.889\nsamples = 6\nvalue = 61980.667'),
```

```
 Text(0.29411764705882354, 0.375, 'squared_error = 0.0\nsamples = 4\nvalue =
63218.0'),
 Text(0.3333333333333333, 0.375, 'squared_error = 13778944.0\nsamples = 2\nvalue
= 59506.0'),
 Text(0.4117647058823529, 0.5416666666666666, 'x[1] <= 0.321\nsquared_error =
1893957.972\nsamples = 12\nvalue = 56956.833'),
 Text(0.39215686274509803, 0.4583333333333333, 'x[0] <= 0.167\nsquared_error =
357850.5\nsamples = 8\nvalue = 56391.0'),
 Text(0.37254901960784315, 0.375, 'squared_error = 0.0\nsamples = 1\nvalue =
55794.0'),
 Text(0.4117647058823529, 0.375, 'x[0] <= 0.778\nsquared_error =
350782.776\nsamples = 7\nvalue = 56476.286'),
 Text(0.39215686274509803, 0.2916666666666667, 'squared_error =
253606.688\nsamples = 4\nvalue = 56666.25'),
 Text(0.43137254901960786, 0.2916666666666667, 'squared_error =
368082.0\nsamples = 3\nvalue = 56223.0'),
 Text(0.43137254901960786, 0.4583333333333333, 'squared_error =
3045168.75\nsamples = 4\nvalue = 58088.5'),
 Text(0.47058823529411764, 0.625, 'x[0] <= 0.278\nsquared_error =
7457268.64\nsamples = 5\nvalue = 62476.4'),
 Text(0.45098039215686275, 0.5416666666666666, 'squared_error =
11651982.25\nsamples = 2\nvalue = 64524.5'),
 Text(0.49019607843137253, 0.5416666666666666, 'squared_error = 0.0\nsamples =
3\nvalue = 61111.0'),
 Text(0.5294117647058824, 0.7083333333333334, 'x[1] <= 0.421\nsquared_error =
899822.469\nsamples = 9\nvalue = 67089.556'),
 Text(0.5098039215686274, 0.625, 'squared_error = 0.0\nsamples = 5\nvalue =
67938.0'),
 Text(0.5490196078431373, 0.625, 'squared_error = 0.0\nsamples = 4\nvalue =
66029.0'),
 Text(0.7843137254901961, 0.875, 'x[1] <= 0.742\nsquared_error =
183756392.016\nsamples = 60\nvalue = 103270.133'),
 Text(0.6568627450980392, 0.7916666666666666, 'x[1] <= 0.521\nsquared_error =
53221721.644\nsamples = 31\nvalue = 91832.032'),
 Text(0.6078431372549019, 0.7083333333333334, 'x[1] <= 0.484\nsquared_error =
743906.25\nsamples = 10\nvalue = 82225.5'),
 Text(0.5882352941176471, 0.625, 'squared_error = 0.0\nsamples = 5\nvalue =
83088.0'),
 Text(0.6274509803921569, 0.625, 'squared_error = 0.0\nsamples = 5\nvalue =
81363.0'),
 Text(0.7058823529411765, 0.7083333333333334, 'x[1] <= 0.626\nsquared_error =
13339290.816\nsamples = 21\nvalue = 96406.571'),
 Text(0.6666666666666666, 0.625, 'x[1] <= 0.568\nsquared_error =
1212201.0\nsamples = 10\nvalue = 92839.0'),
 Text(0.6470588235294118, 0.5416666666666666, 'squared_error = 0.0\nsamples =
5\nvalue = 93940.0'),
 Text(0.6862745098039216, 0.5416666666666666, 'squared_error = 0.0\nsamples =
```

```
5\nvalue = 91738.0'),
 Text(0.7450980392156863, 0.625, 'x[1] <= 0.684\nsquared_error =
2274753.967\nsamples = 11\nvalue = 99649.818'),
 Text(0.7254901960784313, 0.5416666666666666, 'squared_error = 0.0\nsamples =
6\nvalue = 98273.0'),
 Text(0.7647058823529411, 0.5416666666666666, 'squared_error = 0.0\nsamples =
5\nvalue = 101302.0'),
 Text(0.9117647058823529, 0.7916666666666666, 'x[1] <= 0.942\nsquared_error =
33942572.892\nsamples = 29\nvalue = 115497.069'),
 Text(0.8627450980392157, 0.7083333333333334, 'x[1] <= 0.868\nsquared_error =
16965256.244\nsamples = 19\nvalue = 112032.579'),
 Text(0.8235294117647058, 0.625, 'x[1] <= 0.784\nsquared_error =
12329106.975\nsamples = 11\nvalue = 109624.455'),
 Text(0.803921568627451, 0.5416666666666666, 'squared_error = 0.0\nsamples =
4\nvalue = 113812.0'),
 Text(0.8431372549019608, 0.5416666666666666, 'x[1] <= 0.826\nsquared_error =
3628114.531\nsamples = 7\nvalue = 107231.571'),
 Text(0.8235294117647058, 0.4583333333333333, 'squared_error = 0.0\nsamples =
3\nvalue = 109431.0'),
 Text(0.8627450980392157, 0.4583333333333333, 'squared_error = 0.0\nsamples =
4\nvalue = 105582.0'),
 Text(0.9019607843137255, 0.625, 'x[1] <= 0.9\nsquared_error =
4402395.938\nsamples = 8\nvalue = 115343.75'),
 Text(0.8823529411764706, 0.5416666666666666, 'squared_error = 0.0\nsamples =
5\nvalue = 116969.0'),
 Text(0.9215686274509803, 0.5416666666666666, 'squared_error = 0.0\nsamples =
3\nvalue = 112635.0'),
 Text(0.9607843137254902, 0.7083333333333334, 'x[1] <= 0.989\nsquared_error =
64646.64\nsamples = 10\nvalue = 122079.6'),
 Text(0.9411764705882353, 0.625, 'squared_error = 0.0\nsamples = 4\nvalue =
122391.0'),
 Text(0.9803921568627451, 0.625, 'squared_error = 0.0\nsamples = 6\nvalue =
121872.0')]
```



[18]:
```
# Prediction on Test Data
preds = model.predict(x_test)
pd.crosstab(y_test, preds, rownames=['Actual'], colnames=['Predictions'])
```

[18]:

Predictions	37731.000000	38134.000000	41058.500000	41708.000000
Actual				
37731	1	0	0	0
39343	0	0	3	0
39891	0	0	0	1
43525	0	1	0	2
46205	0	0	0	0
54445	0	0	0	0
55794	0	0	0	0

```
56642                    0              0              0              1
56957                    0              0              0              0
57081                    0              0              0              0
57189                    0              0              0              0
60150                    0              0              0              0
61111                    0              0              0              0
63218                    0              0              0              0
64445                    0              0              0              0
66029                    0              0              0              0
67938                    0              0              0              0
81363                    0              0              0              0
83088                    0              0              0              0
91738                    0              0              0              0
93940                    0              0              0              0
101302                   0              0              0              0
105582                   0              0              0              0
109431                   0              0              0              0
112635                   0              0              0              0
113812                   0              0              0              0
116969                   0              0              0              0
122391                   0              0              0              0

Predictions  42616.500000   46205.000000   55177.333333   55794.000000   \
Actual
37731                    0              0              0              0
39343                    0              0              0              0
39891                    1              0              0              0
43525                    0              0              0              0
46205                    0              3              0              0
54445                    0              0              2              0
55794                    0              0              0              0
56642                    0              0              0              0
56957                    0              0              0              2
57081                    0              0              0              0
57189                    0              0              0              0
60150                    0              0              0              0
61111                    0              0              0              0
63218                    0              0              0              0
64445                    0              0              1              0
66029                    0              0              0              0
67938                    0              0              0              0
81363                    0              0              0              0
83088                    0              0              0              0
91738                    0              0              0              0
93940                    0              0              0              0
101302                   0              0              0              0
105582                   0              0              0              0
```

109431	0	0	0	0	
112635	0	0	0	0	
113812	0	0	0	0	
116969	0	0	0	0	
122391	0	0	0	0	

			…			\
Predictions	56223.000000	56274.333333	…	83088.000000	91738.000000	
Actual			…			
37731	0	0	…	0	0	
39343	0	0	…	0	0	
39891	0	0	…	0	0	
43525	0	0	…	0	0	
46205	0	0	…	0	0	
54445	0	0	…	0	0	
55794	1	0	…	0	0	
56642	0	0	…	0	0	
56957	1	0	…	0	0	
57081	0	0	…	0	0	
57189	0	2	…	0	0	
60150	0	0	…	0	0	
61111	0	0	…	0	0	
63218	0	0	…	0	0	
64445	0	1	…	0	0	
66029	0	0	…	0	0	
67938	0	0	…	0	0	
81363	0	0	…	0	0	
83088	0	0	…	1	0	
91738	0	0	…	0	1	
93940	0	0	…	0	0	
101302	0	0	…	0	0	
105582	0	0	…	0	0	
109431	0	0	…	0	0	
112635	0	0	…	0	0	
113812	0	0	…	0	0	
116969	0	0	…	0	0	
122391	0	0	…	0	0	

					\
Predictions	93940.000000	101302.000000	105582.000000	109431.000000	
Actual					
37731	0	0	0	0	
39343	0	0	0	0	
39891	0	0	0	0	
43525	0	0	0	0	
46205	0	0	0	0	
54445	0	0	0	0	
55794	0	0	0	0	
56642	0	0	0	0	

56957	0	0	0	0
57081	0	0	0	0
57189	0	0	0	0
60150	0	0	0	0
61111	0	0	0	0
63218	0	0	0	0
64445	0	0	0	0
66029	0	0	0	0
67938	0	0	0	0
81363	0	0	0	0
83088	0	0	0	0
91738	0	0	0	0
93940	1	0	0	0
101302	0	1	0	0
105582	0	0	2	0
109431	0	0	0	3
112635	0	0	0	0
113812	0	0	0	0
116969	0	0	0	0
122391	0	0	0	0

Predictions	112635.000000	113812.000000	116969.000000	122391.000000
Actual				
37731	0	0	0	0
39343	0	0	0	0
39891	0	0	0	0
43525	0	0	0	0
46205	0	0	0	0
54445	0	0	0	0
55794	0	0	0	0
56642	0	0	0	0
56957	0	0	0	0
57081	0	0	0	0
57189	0	0	0	0
60150	0	0	0	0
61111	0	0	0	0
63218	0	0	0	0
64445	0	0	0	0
66029	0	0	0	0
67938	0	0	0	0
81363	0	0	0	0
83088	0	0	0	0
91738	0	0	0	0
93940	0	0	0	0
101302	0	0	0	0
105582	0	0	0	0
109431	0	0	0	0

```
112635                  3              0              0              0
113812                  0              2              0              0
116969                  0              0              1              0
122391                  0              0              0              2

[28 rows x 30 columns]
```

[19]:
```
np.mean(preds == y_test) # Test Data Accuracy
```

[19]:
```
0.4915254237288136
```

[20]:
```
# Prediction on Train Data
preds = model.predict(x_train)
pd.crosstab(y_train, preds, rownames = ['Actual'], colnames = ['Predictions'])
```

[20]:
```
Predictions  37731.000000   38134.000000   39891.000000   41058.500000   \
Actual
37731                   3              3              0              0
39343                   0              1              0              3
39891                   0              0              3              0
43525                   0              0              0              0
46205                   0              0              0              1
54445                   0              0              0              0
55794                   0              0              0              0
56642                   0              0              0              0
56957                   0              0              0              0
57081                   0              0              0              0
57189                   0              0              0              0
60150                   0              0              0              0
61111                   0              0              0              0
63218                   0              0              0              0
64445                   0              0              0              0
66029                   0              0              0              0
67938                   0              0              0              0
81363                   0              0              0              0
83088                   0              0              0              0
91738                   0              0              0              0
93940                   0              0              0              0
98273                   0              0              0              0
101302                  0              0              0              0
105582                  0              0              0              0
109431                  0              0              0              0
112635                  0              0              0              0
113812                  0              0              0              0
116969                  0              0              0              0
121872                  0              0              0              0
122391                  0              0              0              0
```

Predictions	41708.000000	42616.500000	46205.000000	54445.000000	\
Actual					
37731	0	0	0	0	
39343	0	0	0	0	
39891	1	1	0	0	
43525	1	3	0	0	
46205	0	0	3	0	
54445	0	0	0	1	
55794	0	0	0	0	
56642	0	0	0	0	
56957	0	0	0	0	
57081	0	0	0	0	
57189	0	0	0	0	
60150	0	0	0	0	
61111	0	0	0	0	
63218	0	0	0	0	
64445	0	0	0	0	
66029	0	0	0	0	
67938	0	0	0	0	
81363	0	0	0	0	
83088	0	0	0	0	
91738	0	0	0	0	
93940	0	0	0	0	
98273	0	0	0	0	
101302	0	0	0	0	
105582	0	0	0	0	
109431	0	0	0	0	
112635	0	0	0	0	
113812	0	0	0	0	
116969	0	0	0	0	
121872	0	0	0	0	
122391	0	0	0	0	

Predictions	55177.333333	55794.000000	…	93940.000000	98273.000000	\
Actual			…			
37731	0	0	…	0	0	
39343	0	0	…	0	0	
39891	0	0	…	0	0	
43525	0	0	…	0	0	
46205	0	0	…	0	0	
54445	2	0	…	0	0	
55794	0	1	…	0	0	
56642	1	0	…	0	0	
56957	0	0	…	0	0	
57081	0	0	…	0	0	
57189	0	0	…	0	0	

```
60150                    0              0  …              0              0
61111                    0              0  …              0              0
63218                    0              0  …              0              0
64445                    0              0  …              0              0
66029                    0              0  …              0              0
67938                    0              0  …              0              0
81363                    0              0  …              0              0
83088                    0              0  …              0              0
91738                    0              0  …              0              0
93940                    0              0  …              5              0
98273                    0              0  …              0              6
101302                   0              0  …              0              0
105582                   0              0  …              0              0
109431                   0              0  …              0              0
112635                   0              0  …              0              0
113812                   0              0  …              0              0
116969                   0              0  …              0              0
121872                   0              0  …              0              0
122391                   0              0  …              0              0

Predictions  101302.000000  105582.000000  109431.000000  112635.000000  \
Actual
37731                    0              0              0              0
39343                    0              0              0              0
39891                    0              0              0              0
43525                    0              0              0              0
46205                    0              0              0              0
54445                    0              0              0              0
55794                    0              0              0              0
56642                    0              0              0              0
56957                    0              0              0              0
57081                    0              0              0              0
57189                    0              0              0              0
60150                    0              0              0              0
61111                    0              0              0              0
63218                    0              0              0              0
64445                    0              0              0              0
66029                    0              0              0              0
67938                    0              0              0              0
81363                    0              0              0              0
83088                    0              0              0              0
91738                    0              0              0              0
93940                    0              0              0              0
98273                    0              0              0              0
101302                   5              0              0              0
105582                   0              4              0              0
109431                   0              0              3              0
```

```
112635                    0              0              0              3
113812                    0              0              0              0
116969                    0              0              0              0
121872                    0              0              0              0
122391                    0              0              0              0

Predictions  113812.000000  116969.000000  121872.000000  122391.000000
Actual
37731                     0              0              0              0
39343                     0              0              0              0
39891                     0              0              0              0
43525                     0              0              0              0
46205                     0              0              0              0
54445                     0              0              0              0
55794                     0              0              0              0
56642                     0              0              0              0
56957                     0              0              0              0
57081                     0              0              0              0
57189                     0              0              0              0
60150                     0              0              0              0
61111                     0              0              0              0
63218                     0              0              0              0
64445                     0              0              0              0
66029                     0              0              0              0
67938                     0              0              0              0
81363                     0              0              0              0
83088                     0              0              0              0
91738                     0              0              0              0
93940                     0              0              0              0
98273                     0              0              0              0
101302                    0              0              0              0
105582                    0              0              0              0
109431                    0              0              0              0
112635                    0              0              0              0
113812                    4              0              0              0
116969                    0              5              0              0
121872                    0              0              6              0
122391                    0              0              0              4

[30 rows x 40 columns]
```

[21]:
```
np.mean(preds == y_train) # Train Data Accuracy
```

[21]:
```
0.6715328467153284
```

[22]:
```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_clf = RandomForestClassifier(n_estimators=300, n_jobs=-1, random_state=32)

rf_clf.fit(x_train, y_train)
```

[22]:
```
RandomForestClassifier(n_estimators=300, n_jobs=-1, random_state=32)
```

[23]:
```
from sklearn.metrics import accuracy_score, confusion_matrix

confusion_matrix(y_test, rf_clf.predict(x_test))
accuracy_score(y_test, rf_clf.predict(x_test))
```

[23]:
```
0.5254237288135594
```

[24]:
```
# Evaluation on Training Data
confusion_matrix(y_train, rf_clf.predict(x_train))
accuracy_score(y_train, rf_clf.predict(x_train))
```

[24]:
```
0.9781021897810219
```

[26]:
```
# Hyperparameter tuning using GridSearchCV
######
# GridSearchCV

from sklearn.model_selection import GridSearchCV
rf_clf_grid = RandomForestClassifier(n_estimators=400, n_jobs=1,␣
 ↪random_state=42)
param_grid = {"max_features": [4, 5, 6, 7, 8, 9, 10], "min_samples_split": [2,␣
 ↪3, 10]}
grid_search = GridSearchCV(rf_clf_grid, param_grid, n_jobs=-1, cv=5,␣
 ↪scoring='accuracy')
grid_search.fit(x_train, y_train)
grid_search.best_params_
```

```
C:\Users\Ravi\anaconda3\lib\site-packages\sklearn\model_selection\_split.py:725:
UserWarning: The least populated class in y has only 3 members, which is less
than n_splits=5.
  warnings.warn(
```

[26]:
```
{'max_features': 4, 'min_samples_split': 3}
```

[27]:
```
# Evaluating tuned model on test data
cv_rf_clf_grid = grid_search.best_estimator_
confusion_matrix(y_test, cv_rf_clf_grid.predict(x_test))
accuracy_score(y_test, cv_rf_clf_grid.predict(x_test))
```

[27]:
```
0.8813559322033898
```

[28]:
```
# Evaluating tuned model on train data
confusion_matrix(y_train, cv_rf_clf_grid.predict(x_train))
accuracy_score(y_train, cv_rf_clf_grid.predict(x_train))
```

[28]:
```
0.9635036496350365
```

[ ]:

[ ]:

[ ]: