vsahm5via

October 18, 2024

```
[ ]: import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import r2_score
     from sklearn.metrics import accuracy_score
     from sklearn.preprocessing import OneHotEncoder
     from sklearn.preprocessing import LabelEncoder
```

```
[ ]: data = pd.read_csv('/content/document.csv')
```

```
[ ]: data.head()
```

```
[ ]:    id  carat      cut color clarity  depth  table  price     x     y     z
     0   1   0.23    Ideal     E     SI2   61.5   55.0    326  3.95  3.98  2.43
     1   2   0.21  Premium     E     SI1   59.8   61.0    326  3.89  3.84  2.31
     2   3   0.23     Good     E     VS1   56.9   65.0    327  4.05  4.07  2.31
     3   4   0.29  Premium     I     VS2   62.4   58.0    334  4.20  4.23  2.63
     4   5   0.31     Good     J     SI2   63.3   58.0    335  4.34  4.35  2.75
```

```
[ ]: data.tail()
```

```
[ ]:         id  carat        cut color clarity  depth  table  price     x     y  \
     1292  1293   0.70      Ideal     E     VS2   62.1   55.0   2952  5.71  5.75
     1293  1294   0.74      Ideal     I      IF   62.1   53.9   2952  5.79  5.81
     1294  1295   0.70       Good     E     VS1   61.0   61.0   2952  5.69  5.72
     1295  1296   0.80  Very Good     H     VS2   59.1   59.0   2953  6.02  6.07
     1296  1297   0.79    Premium     F     VS2   63.0   59.0   2953  5.84  5.80

              z
     1292  3.56
     1293  3.60
     1294  3.48
     1295  3.57
     1296  3.66
```

```
[ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1297 entries, 0 to 1296
Data columns (total 11 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   id       1297 non-null   int64
 1   carat    1297 non-null   float64
 2   cut      1297 non-null   object
 3   color    1297 non-null   object
 4   clarity  1297 non-null   object
 5   depth    1297 non-null   float64
 6   table    1297 non-null   float64
 7   price    1297 non-null   int64
 8   x        1297 non-null   float64
 9   y        1297 non-null   float64
 10  z        1297 non-null   float64
dtypes: float64(6), int64(2), object(3)
memory usage: 111.6+ KB
```

```
[ ]: data.isnull().sum()
```

```
[ ]: id         0
     carat      0
     cut        0
     color      0
     clarity    0
     depth      0
     table      0
     price      0
     x          0
     y          0
     z          0
     dtype: int64
```

```
[ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1297 entries, 0 to 1296
Data columns (total 11 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   id       1297 non-null   int64
 1   carat    1297 non-null   float64
 2   cut      1297 non-null   object
 3   color    1297 non-null   object
 4   clarity  1297 non-null   object
 5   depth    1297 non-null   float64
 6   table    1297 non-null   float64
 7   price    1297 non-null   int64
 8   x        1297 non-null   float64
 9   y        1297 non-null   float64
 10  z        1297 non-null   float64
dtypes: float64(6), int64(2), object(3)
memory usage: 111.6+ KB
```

```
[ ]: data.isnull().any()
```

```
[ ]: id          False
     carat       False
     cut         False
     color       False
     clarity     False
     depth       False
     table       False
     price       False
     x           False
     y           False
     z           False
     dtype: bool
```

```
[ ]: le = LabelEncoder()
     data['cut'] = le.fit_transform(data['cut'])
     data
```

```
[ ]:         id  carat  cut color clarity  depth  table  price     x     y     z
     0        1   0.23    2     E     SI2   61.5   55.0    326  3.95  3.98  2.43
     1        2   0.21    3     E     SI1   59.8   61.0    326  3.89  3.84  2.31
     2        3   0.23    1     E     VS1   56.9   65.0    327  4.05  4.07  2.31
     3        4   0.29    3     I     VS2   62.4   58.0    334  4.20  4.23  2.63
     4        5   0.31    1     J     SI2   63.3   58.0    335  4.34  4.35  2.75
     …    …    …    …    …     …    …    …    …    …    …    …
     1292  1293   0.70    2     E     VS2   62.1   55.0   2952  5.71  5.75  3.56
     1293  1294   0.74    2     I      IF   62.1   53.9   2952  5.79  5.81  3.60
     1294  1295   0.70    1     E     VS1   61.0   61.0   2952  5.69  5.72  3.48
     1295  1296   0.80    4     H     VS2   59.1   59.0   2953  6.02  6.07  3.57
     1296  1297   0.79    3     F     VS2   63.0   59.0   2953  5.84  5.80  3.66

     [1297 rows x 11 columns]
```

```
[ ]: le = LabelEncoder()
     data['color'] = le.fit_transform(data['color'])
     data['clarity'] = le.fit_transform(data['clarity'])
     data
```

```
[ ]:          id  carat  cut  color  clarity  depth  table  price     x     y     z
     0         1   0.23    2      1        3   61.5   55.0    326  3.95  3.98  2.43
     1         2   0.21    3      1        2   59.8   61.0    326  3.89  3.84  2.31
     2         3   0.23    1      1        4   56.9   65.0    327  4.05  4.07  2.31
     3         4   0.29    3      5        5   62.4   58.0    334  4.20  4.23  2.63
     4         5   0.31    1      6        3   63.3   58.0    335  4.34  4.35  2.75
     …     …     …    …    …      …     …     …     …    …     …    …
     1292   1293   0.70    2      1        5   62.1   55.0   2952  5.71  5.75  3.56
     1293   1294   0.74    2      5        1   62.1   53.9   2952  5.79  5.81  3.60
     1294   1295   0.70    1      1        4   61.0   61.0   2952  5.69  5.72  3.48
     1295   1296   0.80    4      4        5   59.1   59.0   2953  6.02  6.07  3.57
     1296   1297   0.79    3      2        5   63.0   59.0   2953  5.84  5.80  3.66

     [1297 rows x 11 columns]
```

```
[ ]: x = data.drop(columns='color')
     y = data.drop(columns='clarity')
```

```
[ ]: x.shape
```

```
[ ]: (1297, 10)
```

```
[ ]: y.shape
```

```
[ ]: (1297, 10)
```

```
[ ]: z.shape
```

```
[ ]: (1297, 7)
```

```
[ ]: z
```

```
[ ]: array([[0., 1., 0., …, 0., 0., 0.],
            [0., 1., 0., …, 0., 0., 0.],
            [0., 1., 0., …, 0., 0., 0.],
            …,
            [0., 1., 0., …, 0., 0., 0.],
            [0., 0., 0., …, 1., 0., 0.],
            [0., 0., 1., …, 0., 0., 0.]])
```

```
[ ]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
       ↪random_state=0)
```

```
[ ]: x_train.shape
```

```
[ ]: (1037, 10)
[ ]: x_test.shape
```

```
[ ]: (260, 10)
```

```
[ ]: y_train.shape
```

```
[ ]: (1037, 10)
```

```
[ ]: y_test.shape
```

```
[ ]: (260, 10)
```

```
[ ]: mlr = LinearRegression()
     mlr.fit(x_train, y_train)
```

```
[ ]: LinearRegression()
```

```
[ ]: y_prediction = mlr.predict(x_test)
     y_prediction
```

```
[ ]: array([[6.660e+02, 7.300e-01, 2.000e+00, …, 5.920e+00, 5.950e+00,
             3.510e+00],
            [4.750e+02, 7.200e-01, 2.000e+00, …, 5.730e+00, 5.770e+00,
             3.580e+00],
            [7.180e+02, 8.000e-01, 2.000e+00, …, 5.970e+00, 6.010e+00,
             3.690e+00],
            …,
            [1.111e+03, 7.300e-01, 2.000e+00, …, 5.760e+00, 5.800e+00,
             3.550e+00],
            [9.900e+02, 7.700e-01, 2.000e+00, …, 5.830e+00, 5.890e+00,
             3.640e+00],
            [3.310e+02, 7.400e-01, 4.000e+00, …, 5.740e+00, 5.790e+00,
             3.660e+00]])
```

```
[ ]: y_test
```

```
[ ]:         id  carat  cut  color  depth  table  price     x     y     z
     665    666   0.73    2      1   59.1   59.0   2846  5.92  5.95  3.51
     474    475   0.72    2      4   62.3   56.0   2819  5.73  5.77  3.58
     717    718   0.80    2      3   61.6   56.0   2856  5.97  6.01  3.69
     240    241   0.71    2      2   62.5   55.0   2788  5.71  5.65  3.55
     699    700   0.78    4      4   61.9   57.1   2854  5.87  5.95  3.66
     …     …    …    …    …    …     …     …    …    …    …
     44      45   0.32    1      4   63.1   56.0    403  4.34  4.37  2.75
     1181  1182   0.71    2      3   62.6   57.0   2930  5.67  5.70  3.56
     1110  1111   0.73    2      1   61.4   58.0   2914  5.76  5.80  3.55
     989    990   0.77    2      0   62.1   56.0   2896  5.83  5.89  3.64
```

```
330    331   0.74    4      1   63.5    56.0    2803  5.74  5.79  3.66

[260 rows x 10 columns]
```

[ ]:
```
acc = r2_score(y_test, y_prediction)
acc
```

[ ]:
```
0.9327476581191927
```

[ ]:
```
```