sta-ass-4

August 30, 2024

[ ]:
```
#Question1
import pandas as pd
d=pd.read_csv('/train (1).csv')
d
y=d['price_range']
print(y)
```

```
0       1
1       2
2       2
3       2
4       1
       ..
1995    0
1996    2
1997    3
1998    0
1999    3
Name: price_range, Length: 2000, dtype: int64
```

[ ]:
```
x=d.drop('price_range',axis=1)
print(x)
```

```
      battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  \
0               842     0          2.2         0   1       0           7
1              1021     1          0.5         1   0       1          53
2               563     1          0.5         1   2       1          41
3               615     1          2.5         0   0       0          10
4              1821     1          1.2         0  13       1          44
…               …    …           …         …  ..      …          …
1995            794     1          0.5         1   0       1           2
1996           1965     1          2.6         1   0       0          39
1997           1911     0          0.9         1   1       1          36
1998           1512     0          0.9         0   4       1          46
1999            510     1          2.0         1   5       1          45

      m_dep  mobile_wt  n_cores  pc  px_height  px_width   ram  sc_h  sc_w  \
0       0.6        188        2   2         20       756  2549     9     7
```

```
1       0.7        136        3   6        905      1988  2631    17     3
2       0.9        145        5   6       1263      1716  2603    11     2
3       0.8        131        6   9       1216      1786  2769    16     8
4       0.6        141        2  14       1208      1212  1411     8     2
…     …          …      …  ..       …       …   …   …   …
1995    0.8        106        6  14       1222      1890   668    13     4
1996    0.2        187        4   3        915      1965  2032    11    10
1997    0.7        108        8   3        868      1632  3057     9     1
1998    0.1        145        5   5        336       670   869    18    10
1999    0.9        168        6  16        483       754  3919    19     4

      talk_time  three_g  touch_screen  wifi
0            19        0             0     1
1             7        1             1     0
2             9        1             1     0
3            11        1             0     0
4            15        1             1     0
…          …      …            …   …
1995         19        1             1     0
1996         16        1             1     1
1997          5        1             1     0
1998         19        1             1     1
1999          2        1             1     1

[2000 rows x 20 columns]
```

[ ]:
```
dd=(x-x.min())/(x.max()-x.min())
print(dd)
```

```
      battery_power  blue  clock_speed  dual_sim        fc  four_g  \
0          0.227789   0.0         0.68       0.0  0.052632     0.0
1          0.347361   1.0         0.00       1.0  0.000000     1.0
2          0.041416   1.0         0.00       1.0  0.105263     1.0
3          0.076152   1.0         0.80       0.0  0.000000     0.0
4          0.881764   1.0         0.28       0.0  0.684211     1.0
…             …   …          …       …       …     …
1995       0.195725   1.0         0.00       1.0  0.000000     1.0
1996       0.977956   1.0         0.84       1.0  0.000000     0.0
1997       0.941884   0.0         0.16       1.0  0.052632     1.0
1998       0.675351   0.0         0.16       0.0  0.210526     1.0
1999       0.006012   1.0         0.60       1.0  0.263158     1.0

      int_memory     m_dep  mobile_wt   n_cores    pc  px_height  px_width  \
0       0.080645  0.555556   0.900000  0.142857  0.10   0.010204  0.170895
1       0.822581  0.666667   0.466667  0.285714  0.30   0.461735  0.993324
2       0.629032  0.888889   0.541667  0.571429  0.30   0.644388  0.811749
3       0.129032  0.777778   0.425000  0.714286  0.45   0.620408  0.858478
4       0.677419  0.555556   0.508333  0.142857  0.70   0.616327  0.475300
```

```
…           …         …          …         …    …          …         …
1995    0.000000  0.777778   0.216667  0.714286  0.70   0.623469  0.927904
1996    0.596774  0.111111   0.891667  0.428571  0.15   0.466837  0.977971
1997    0.548387  0.666667   0.233333  1.000000  0.15   0.442857  0.755674
1998    0.709677  0.000000   0.541667  0.571429  0.25   0.171429  0.113485
1999    0.693548  0.888889   0.733333  0.714286  0.80   0.246429  0.169559

            ram      sc_h      sc_w  talk_time  three_g  touch_screen  wifi
0      0.612774  0.285714  0.388889   0.944444      0.0           0.0   1.0
1      0.634687  0.857143  0.166667   0.277778      1.0           1.0   0.0
2      0.627205  0.428571  0.111111   0.388889      1.0           1.0   0.0
3      0.671566  0.785714  0.444444   0.500000      1.0           0.0   0.0
4      0.308658  0.214286  0.111111   0.722222      1.0           1.0   0.0
…           …         …         …          …        …             …     …
1995   0.110102  0.571429  0.222222   0.944444      1.0           1.0   0.0
1996   0.474613  0.428571  0.555556   0.777778      1.0           1.0   1.0
1997   0.748530  0.285714  0.055556   0.166667      1.0           1.0   0.0
1998   0.163816  0.928571  0.555556   0.944444      1.0           1.0   1.0
1999   0.978888  1.000000  0.222222   0.000000      1.0           1.0   1.0

[2000 rows x 20 columns]
```

[ ]:
```
from sklearn.model_selection import train_test_split#for dividing test and train
```

[ ]:
```
x_train,x_test,y_train,y_test=train_test_split(dd,y,test_size=0.
↪2,random_state=100)
```

[ ]:
```
print(x_train)
```

```
      battery_power  blue  clock_speed  dual_sim        fc  four_g  \
1260       0.398130   0.0         0.36       0.0  0.000000     0.0
916        0.210421   1.0         0.00       0.0  0.210526     0.0
532        0.184369   1.0         0.84       0.0  0.157895     0.0
1159       0.002004   0.0         0.92       1.0  0.000000     0.0
1584       0.983300   1.0         0.76       1.0  0.315789     0.0
…               …     …            …         …         …       …
1879       0.321977   0.0         0.52       0.0  0.000000     1.0
1895       0.566466   1.0         0.64       1.0  0.157895     0.0
1859       0.098864   1.0         0.36       0.0  0.210526     1.0
792        0.765531   0.0         0.36       1.0  0.157895     1.0
1544       0.553774   1.0         0.52       1.0  0.052632     0.0

      int_memory     m_dep  mobile_wt   n_cores    pc  px_height  px_width  \
1260    0.935484  0.777778   0.841667  0.000000  0.85   0.206633  0.012684
916     0.193548  0.888889   0.983333  0.285714  0.65   0.076020  0.038718
532     0.451613  0.888889   0.566667  0.571429  0.85   0.096429  0.429239
1159    0.612903  0.444444   0.816667  0.285714  0.00   0.319388  0.463952
1584    0.709677  0.000000   0.941667  0.428571  0.50   0.356633  0.234312
```

```
…          …        …         …        …   …         …       …
1879    0.822581  0.888889   0.891667  0.000000  0.20   0.510714  0.451268
1895    0.709677  0.888889   0.450000  0.714286  0.60   0.108163  0.082109
1859    0.274194  0.777778   0.916667  0.857143  0.60   0.175510  0.701602
792     0.580645  0.222222   0.183333  0.714286  0.75   0.098469  0.457276
1544    0.612903  0.444444   0.808333  0.000000  0.25   0.098980  0.261682

           ram      sc_h      sc_w  talk_time  three_g  touch_screen  wifi
1260  0.067611  0.714286  0.666667   0.555556      1.0           0.0   0.0
916   0.203634  0.857143  0.555556   0.722222      1.0           0.0   0.0
532   0.970069  0.642857  0.555556   0.277778      1.0           0.0   0.0
1159  0.057189  0.071429  0.000000   0.777778      1.0           0.0   0.0
1584  0.102352  0.071429  0.000000   0.055556      0.0           0.0   1.0
…        …        …         …         …       …             …   …
1879  0.302779  0.857143  0.444444   0.333333      1.0           1.0   1.0
1895  0.564137  0.428571  0.388889   0.444444      1.0           0.0   1.0
1859  0.129075  0.928571  0.055556   0.000000      1.0           1.0   0.0
792   0.608231  0.785714  0.777778   0.722222      1.0           0.0   1.0
1544  0.647247  0.285714  0.222222   0.666667      1.0           1.0   1.0

[1600 rows x 20 columns]
```

[ ]:
```
print(x_test)
```

```
      battery_power  blue  clock_speed  dual_sim        fc  four_g  \
1025       0.387442   1.0         0.80       1.0  0.684211     0.0
1208       0.325985   1.0         0.08       0.0  0.684211     1.0
1055       0.814295   1.0         0.08       1.0  0.000000     0.0
367        0.216433   0.0         0.96       1.0  0.000000     0.0
815        0.675351   1.0         0.00       0.0  0.052632     0.0
…             …   …           …        …       …     …
807        0.541750   0.0         0.32       0.0  0.263158     1.0
711        0.830327   1.0         0.20       0.0  0.052632     1.0
1541       0.215097   0.0         0.20       1.0  0.473684     1.0
1001       0.683367   0.0         0.84       1.0  0.263158     0.0
687        0.178357   0.0         0.00       1.0  0.000000     0.0

      int_memory     m_dep  mobile_wt   n_cores    pc  px_height  px_width  \
1025    0.758065  1.000000   0.158333  0.285714  0.85   0.144898  0.012684
1208    0.790323  0.333333   0.433333  0.285714  0.75   0.341837  0.202937
1055    0.274194  0.000000   0.475000  0.142857  0.15   0.208673  0.097463
367     0.516129  0.777778   0.158333  1.000000  0.85   0.750510  0.911883
815     0.806452  0.000000   0.425000  0.000000  0.70   0.169898  0.301736
…          …        …         …         …       …         …       …
807     0.129032  0.555556   0.691667  0.142857  0.45   0.048469  0.262350
711     0.290323  0.444444   0.266667  1.000000  0.15   0.034694  0.145527
1541    0.258065  0.111111   0.850000  0.571429  0.80   0.602041  0.567423
1001    0.887097  0.444444   0.366667  0.142857  0.60   0.320918  0.090120
```

```
687     0.274194  0.888889   0.658333  0.142857  0.05   0.325000  0.327770

           ram      sc_h      sc_w  talk_time  three_g  touch_screen  wifi
1025  0.018439  0.285714  0.055556   0.222222      0.0           0.0   1.0
1208  0.559327  0.357143  0.055556   1.000000      1.0           0.0   1.0
1055  0.258151  0.571429  0.111111   0.277778      0.0           0.0   0.0
367   0.710583  0.214286  0.111111   0.444444      1.0           1.0   0.0
815   0.733832  0.071429  0.277778   0.277778      0.0           0.0   1.0
…          …         …         …          …        …             …   …
807   0.984768  0.071429  0.111111   0.611111      1.0           1.0   1.0
711   0.253608  0.928571  0.388889   0.666667      1.0           0.0   0.0
1541  0.972207  0.571429  0.555556   0.388889      1.0           1.0   0.0
1001  0.184661  0.428571  0.222222   0.777778      0.0           1.0   1.0
687   0.237573  1.000000  0.055556   0.277778      0.0           0.0   1.0

[400 rows x 20 columns]
```

[ ]:
```
print(y_train)
```

```
1260    0
916     0
532     3
1159    0
1584    0
       ..
1879    1
1895    1
1859    0
792     2
1544    2
Name: price_range, Length: 1600, dtype: int64
```

[ ]:
```
print(y_test)
```

```
1025    0
1208    1
1055    1
367     3
815     2
       ..
807     3
711     1
1541    3
1001    0
687     0
Name: price_range, Length: 400, dtype: int64
```

[ ]:
```
print(x.isnull())
```

```
      battery_power   blue  clock_speed  dual_sim     fc  four_g  int_memory  \
0             False  False        False     False  False   False       False
1             False  False        False     False  False   False       False
2             False  False        False     False  False   False       False
3             False  False        False     False  False   False       False
4             False  False        False     False  False   False       False
…               …    …            …         …      …       …           …
1995          False  False        False     False  False   False       False
1996          False  False        False     False  False   False       False
1997          False  False        False     False  False   False       False
1998          False  False        False     False  False   False       False
1999          False  False        False     False  False   False       False

      m_dep  mobile_wt  n_cores     pc  px_height  px_width    ram   sc_h  \
0     False      False    False  False      False     False  False  False
1     False      False    False  False      False     False  False  False
2     False      False    False  False      False     False  False  False
3     False      False    False  False      False     False  False  False
4     False      False    False  False      False     False  False  False
…     …        …      …      …    …        …         …      …    …
1995  False      False    False  False      False     False  False  False
1996  False      False    False  False      False     False  False  False
1997  False      False    False  False      False     False  False  False
1998  False      False    False  False      False     False  False  False
1999  False      False    False  False      False     False  False  False

       sc_w  talk_time  three_g  touch_screen   wifi
0     False      False    False         False  False
1     False      False    False         False  False
2     False      False    False         False  False
3     False      False    False         False  False
4     False      False    False         False  False
…     …        …      …      …           …    …
1995  False      False    False         False  False
1996  False      False    False         False  False
1997  False      False    False         False  False
1998  False      False    False         False  False
1999  False      False    False         False  False

[2000 rows x 20 columns]
```

[ ]:
```
print(d.dtypes)
```

```
battery_power      int64
blue               int64
clock_speed      float64
dual_sim           int64
fc                 int64
```

```
four_g             int64
int_memory         int64
m_dep            float64
mobile_wt          int64
n_cores            int64
pc                 int64
px_height          int64
px_width           int64
ram                int64
sc_h               int64
sc_w               int64
talk_time          int64
three_g            int64
touch_screen       int64
wifi               int64
price_range        int64
dtype: object
```

[8]:
```
#2)Question
# Read the data with pandas and describe the data
import pandas as pd
d=pd.read_csv("/content/sample_data/california_housing_train.csv")
print(d)
print(d.describe)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0         -114.31     34.19                15.0       5612.0          1283.0
1         -114.47     34.40                19.0       7650.0          1901.0
2         -114.56     33.69                17.0        720.0           174.0
3         -114.57     33.64                14.0       1501.0           337.0
4         -114.57     33.57                20.0       1454.0           326.0
…             …         …                   …          …               …
16995     -124.26     40.58                52.0       2217.0           394.0
16996     -124.27     40.69                36.0       2349.0           528.0
16997     -124.30     41.84                17.0       2677.0           531.0
16998     -124.30     41.80                19.0       2672.0           552.0
16999     -124.35     40.54                52.0       1820.0           300.0

       population  households  median_income  median_house_value
0          1015.0       472.0         1.4936             66900.0
1          1129.0       463.0         1.8200             80100.0
2           333.0       117.0         1.6509             85700.0
3           515.0       226.0         3.1917             73400.0
4           624.0       262.0         1.9250             65500.0
…             …           …              …                  …
16995       907.0       369.0         2.3571            111400.0
16996      1194.0       465.0         2.5179             79000.0
16997      1244.0       456.0         3.0313            103600.0
```

```
16998      1298.0       478.0         1.9797             85800.0
16999       806.0       270.0         3.0147             94600.0

[17000 rows x 9 columns]
<bound method NDFrame.describe of        longitude  latitude  housing_median_age
total_rooms  total_bedrooms  \
0         -114.31     34.19                15.0       5612.0          1283.0
1         -114.47     34.40                19.0       7650.0          1901.0
2         -114.56     33.69                17.0        720.0           174.0
3         -114.57     33.64                14.0       1501.0           337.0
4         -114.57     33.57                20.0       1454.0           326.0
…            …        …                    …          …               …
16995     -124.26     40.58                52.0       2217.0           394.0
16996     -124.27     40.69                36.0       2349.0           528.0
16997     -124.30     41.84                17.0       2677.0           531.0
16998     -124.30     41.80                19.0       2672.0           552.0
16999     -124.35     40.54                52.0       1820.0           300.0

       population  households  median_income  median_house_value
0          1015.0       472.0         1.4936             66900.0
1          1129.0       463.0         1.8200             80100.0
2           333.0       117.0         1.6509             85700.0
3           515.0       226.0         3.1917             73400.0
4           624.0       262.0         1.9250             65500.0
…             …          …             …                  …
16995       907.0       369.0         2.3571            111400.0
16996      1194.0       465.0         2.5179             79000.0
16997      1244.0       456.0         3.0313            103600.0
16998      1298.0       478.0         1.9797             85800.0
16999       806.0       270.0         3.0147             94600.0

[17000 rows x 9 columns]>
```

[9]:
```
#Find data type and shape of each column
print(d.dtypes)
print(d.shape)
```

```
longitude             float64
latitude              float64
housing_median_age    float64
total_rooms           float64
total_bedrooms        float64
population            float64
households            float64
median_income         float64
median_house_value    float64
dtype: object
(17000, 9)
```

[10]:
```
#Find the target and features
x=d.drop('median_house_value',axis=1)
y=d['median_house_value']
print(x)
print(y)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0         -114.31     34.19                15.0       5612.0          1283.0
1         -114.47     34.40                19.0       7650.0          1901.0
2         -114.56     33.69                17.0        720.0           174.0
3         -114.57     33.64                14.0       1501.0           337.0
4         -114.57     33.57                20.0       1454.0           326.0
…             …         …                   …          …               …
16995     -124.26     40.58                52.0       2217.0           394.0
16996     -124.27     40.69                36.0       2349.0           528.0
16997     -124.30     41.84                17.0       2677.0           531.0
16998     -124.30     41.80                19.0       2672.0           552.0
16999     -124.35     40.54                52.0       1820.0           300.0

       population  households  median_income
0          1015.0       472.0         1.4936
1          1129.0       463.0         1.8200
2           333.0       117.0         1.6509
3           515.0       226.0         3.1917
4           624.0       262.0         1.9250
…             …           …              …
16995       907.0       369.0         2.3571
16996      1194.0       465.0         2.5179
16997      1244.0       456.0         3.0313
16998      1298.0       478.0         1.9797
16999       806.0       270.0         3.0147

[17000 rows x 8 columns]
0         66900.0
1         80100.0
2         85700.0
3         73400.0
4         65500.0
           …
16995    111400.0
16996     79000.0
16997    103600.0
16998     85800.0
16999     94600.0
Name: median_house_value, Length: 17000, dtype: float64
```

[11]:
```
#Find the null values (if yes fill the null values with '0' or mean of that␣␣
  ↪column)
print(d.isnull())
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0          False     False               False        False           False
1          False     False               False        False           False
2          False     False               False        False           False
3          False     False               False        False           False
4          False     False               False        False           False
…            …       …                   …            …               …
16995      False     False               False        False           False
16996      False     False               False        False           False
16997      False     False               False        False           False
16998      False     False               False        False           False
16999      False     False               False        False           False

       population  households  median_income  median_house_value
0           False       False          False               False
1           False       False          False               False
2           False       False          False               False
3           False       False          False               False
4           False       False          False               False
…             …           …              …                   …
16995       False       False          False               False
16996       False       False          False               False
16997       False       False          False               False
16998       False       False          False               False
16999       False       False          False               False

[17000 rows x 9 columns]
```

[12]:
```
#Normalize all the features
dd=(x-x.min())/(x.max()-x.min())
print(dd)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0       1.000000  0.175345            0.274510     0.147885        0.198945
1       0.984064  0.197662            0.352941     0.201608        0.294848
2       0.975100  0.122210            0.313725     0.018927        0.026847
3       0.974104  0.116897            0.254902     0.039515        0.052142
4       0.974104  0.109458            0.372549     0.038276        0.050435
…            …       …                   …            …               …
16995   0.008964  0.854410            1.000000     0.058389        0.060987
16996   0.007968  0.866100            0.686275     0.061869        0.081782
16997   0.004980  0.988310            0.313725     0.070515        0.082247
16998   0.004980  0.984060            0.352941     0.070384        0.085506
```

```
16999   0.000000  0.850159            1.000000     0.047924        0.046400

       population  households  median_income
0        0.028364    0.077454       0.068530
1        0.031559    0.075974       0.091040
2        0.009249    0.019076       0.079378
3        0.014350    0.037000       0.185639
4        0.017405    0.042921       0.098281
…             …           …              …
16995    0.025337    0.060516       0.128081
16996    0.033381    0.076303       0.139170
16997    0.034782    0.074823       0.174577
16998    0.036296    0.078441       0.102054
16999    0.022506    0.044236       0.173432

[17000 rows x 8 columns]
```

[17]:
```
#Split the data into train and test.
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(dd,y,test_size=0.
 ↪2,random_state=100)
print(x_train)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
14730   0.215139  0.555792            0.862745     0.048768        0.060832
9389    0.511952  0.408077            0.333333     0.056834        0.063935
7662    0.595618  0.144527            0.607843     0.008699        0.015829
9146    0.530876  0.297556            0.666667     0.043443        0.060366
13485   0.238048  0.616366            0.254902     0.079953        0.105369
…            …         …                   …            …               …
16304   0.184263  0.571732            0.607843     0.069514        0.064246
79      0.875498  0.027630            0.274510     0.030816        0.050745
12119   0.289841  0.647184            0.627451     0.028338        0.041899
14147   0.227092  0.548353            0.745098     0.037380        0.042055
5640    0.614542  0.153029            0.764706     0.044708        0.066729

       population  households  median_income
14730    0.031643    0.057227       0.152825
9389     0.040864    0.064792       0.104943
7662     0.004877    0.016609       0.195928
9146     0.032008    0.056241       0.071433
13485    0.044676    0.108864       0.214045
…             …           …              …
16304    0.028308    0.066930       0.390567
79       0.028616    0.048841       0.061261
12119    0.021021    0.038645       0.058365
14147    0.017994    0.045387       0.250003
5640     0.035791    0.066436       0.146557
```

```
[13600 rows x 8 columns]
```

[18]:
```
print(x_test)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
1559    0.710159  0.141339            0.294118     0.001423        0.000931
5775    0.613546  0.143464            0.725490     0.105865        0.152855
10247   0.436255  0.679065            0.450980     0.022512        0.028554
5234    0.618526  0.175345            0.862745     0.094715        0.095903
1416    0.714143  0.027630            0.568627     0.064690        0.060366
…           …       …                   …            …               …
9213    0.526892  0.375133            0.372549     0.070674        0.085661
15348   0.205179  0.568544            1.000000     0.083538        0.108163
14256   0.225100  0.700319            0.529412     0.109846        0.138734
5688    0.614542  0.131775            0.372549     0.048768        0.086127
6548    0.604582  0.181722            0.549020     0.067405        0.121974

       population  households  median_income
1559     0.001009    0.002138       0.146557
5775     0.103674    0.152278       0.167543
10247    0.010791    0.017596       0.184873
5234     0.047171    0.099655       0.291727
1416     0.028560    0.064463       0.231328
…           …           …              …
9213     0.041873    0.089952       0.082978
15348    0.041901    0.114290       0.184797
14256    0.072424    0.133037       0.126281
5688     0.047899    0.091268       0.065972
6548     0.052776    0.126295       0.145081

[3400 rows x 8 columns]
```

[19]:
```
print(y_train)
```

```
14730    101400.0
9389      58000.0
7662     256300.0
9146      56500.0
13485    143500.0
           …
16304    349200.0
79        69400.0
12119     55900.0
14147    232500.0
5640     206300.0
Name: median_house_value, Length: 13600, dtype: float64
```

[20]:
```
print(y_test)
```

```
1559     500001.0
5775     113600.0
10247    107200.0
5234     201000.0
1416     198500.0
           …
9213      64600.0
15348    218200.0
14256     65500.0
5688     152500.0
6548     162100.0
Name: median_house_value, Length: 3400, dtype: float64
```