Linear Regression(hp)

March 20, 2024

IMPORTING NECESSARY LIBRARIES

[10]:
```
import pandas as pd
import numpy as np
df=pd.read_csv("C:\\Users\\KENNY\\Downloads\\archive (1)\\winequality-red.csv")
df
```

[10]:
```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
…             …               …           …             …        …
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
…                  …                   …      …     …        …
1594                 32.0                  44.0  0.99490  3.45       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.42       0.75
1597                 32.0                  44.0  0.99547  3.57       0.71
1598                 18.0                  42.0  0.99549  3.39       0.66

      alcohol  quality
0         9.4        5
1         9.8        5
2         9.8        5
3         9.8        6
```

```
4         9.4        5
…        …      …
1594      10.5        5
1595      11.2        6
1596      11.0        6
1597      10.2        5
1598      11.0        6

[1599 rows x 12 columns]
```

ASSINGING DATA TO DEPENDENT AND INDEPENDENT VARIABLES

[11]:
```
x=df.drop(['quality'],axis=1)
y=df['quality']
```

SPLITTING DATA

[12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪2,random_state=42)
```

STANDARDZING DATA

[13]:
```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler=scaler.fit_transform(x_train)
# scaler=scaler.transform(x_test)
```

MODEL BUILDING

[14]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

[14]:
```
array([5.34666441, 5.05631345, 5.66446972, 5.46451484, 5.72518476,
       5.27928659, 5.03421667, 5.12623347, 5.74534288, 5.68665032,
       6.13959677, 5.23386892, 5.54991474, 5.25825299, 5.44810502,
       6.46828999, 5.15018088, 5.59105157, 6.5560658 , 5.32255751,
       5.3918385 , 5.19610791, 5.94475739, 6.36197631, 5.35484893,
       5.41907575, 6.36483321, 5.35121573, 5.172392  , 6.16987311,
       5.25263058, 5.50657406, 5.75422105, 5.39101712, 5.45331031,
       5.02757499, 6.16173243, 5.68661555, 5.6486077 , 6.165471  ,
       5.52872593, 5.24414488, 6.17724727, 5.16500868, 5.87598332,
       5.81317121, 6.41982782, 5.6059474 , 5.15232137, 5.55634632,
       5.16044852, 5.10449459, 5.58371721, 6.33425313, 4.95134985,
       4.98364804, 6.01041999, 5.40809804, 5.83802638, 5.2486897 ,
       5.60717482, 5.96630957, 5.27619063, 5.30380113, 6.4949309 ,
```

```
5.42033967, 6.34273471, 5.24618531, 6.41317317, 5.31237924,
6.41746963, 4.74315748, 5.79362039, 5.8283184 , 6.17598768,
5.29723707, 6.76198733, 5.89745261, 6.07833712, 6.43522754,
5.29499011, 6.4546625 , 5.45007864, 5.69644693, 5.72368681,
6.41233601, 5.31025119, 5.84548953, 6.31433877, 5.20585049,
6.10141578, 5.70349712, 5.78679322, 5.93173502, 5.1852885 ,
5.74819506, 5.17351769, 5.69336056, 4.99158806, 5.52004223,
5.06867029, 5.13831807, 5.84991801, 5.72612872, 5.47766711,
6.12476389, 5.73551897, 5.44180611, 6.08785125, 5.24667513,
6.68434941, 5.26499691, 6.15359147, 4.74493131, 5.82508834,
5.9872331 , 6.17033538, 5.50859099, 5.02156367, 5.83326942,
6.21086737, 5.26363047, 5.75354145, 5.38942262, 5.39641713,
5.25966957, 6.21024761, 5.69536196, 5.58586923, 5.82155344,
5.79362039, 5.14962195, 5.01142496, 6.34824026, 5.55634632,
5.08213438, 5.05668453, 5.3517036 , 5.11920475, 5.66948552,
6.01614582, 6.03912287, 6.2439487 , 5.48155178, 5.86335248,
5.26302973, 6.06162683, 5.4041289 , 5.99869245, 5.06897434,
5.70161041, 6.14167652, 5.11821365, 5.67658854, 5.79362039,
6.0891404 , 5.22103588, 5.90134727, 5.48941228, 5.93412645,
6.3118134 , 5.71785286, 6.13152024, 4.9898825 , 5.39143155,
5.63146602, 4.70626967, 5.232132  , 5.04110749, 4.99137335,
5.20669998, 5.11005631, 6.29652093, 5.48263655, 5.73380671,
5.86096397, 6.11131909, 5.38204246, 5.39418516, 5.11161705,
4.74487438, 6.34043215, 5.57642863, 6.52465957, 5.18100269,
6.37846442, 5.39147732, 5.7435927 , 6.71436012, 5.48263655,
5.42573746, 6.08035849, 5.6017508 , 6.52660959, 5.79174569,
5.32807323, 4.92850887, 5.40669848, 5.49983794, 6.12476389,
5.36974106, 5.78401123, 5.48534309, 5.02135392, 6.65592712,
5.62370825, 4.83368748, 5.73347951, 5.68074781, 6.09738854,
5.99258428, 5.16969289, 5.7770828 , 6.59697123, 6.37009025,
5.77981876, 5.46465189, 5.19009343, 5.80517998, 5.30830978,
5.09158113, 6.24863165, 6.33674607, 5.99341483, 5.16829696,
4.81289689, 5.22265229, 6.44901207, 5.48931502, 5.31886287,
5.5589884 , 5.04938167, 6.32905554, 5.98208683, 6.04415923,
6.12476389, 5.37906696, 5.72368681, 4.795237  , 5.03676054,
5.68938109, 5.01079638, 5.83995808, 6.13732216, 5.24782156,
5.56627333, 6.00210169, 5.3626292 , 6.68219105, 5.11532126,
5.78120835, 5.62454656, 5.31952796, 5.51514228, 5.20719665,
5.13154551, 5.48620652, 5.85075029, 5.71919777, 6.80397753,
6.20404528, 6.04410296, 5.38204246, 6.50598024, 5.85449947,
6.30306847, 5.05268393, 4.92613186, 5.94872379, 6.32176541,
5.18546252, 5.8361393 , 5.40120414, 5.17199122, 5.3095161 ,
5.49911144, 5.66556707, 6.21315993, 6.22227229, 5.26433184,
6.48967503, 4.95165562, 5.37197617, 5.49931461, 5.3577211 ,
5.82641444, 4.97385804, 6.03912287, 5.03990278, 5.76144224,
5.67870975, 6.57726748, 5.67261468, 5.5851728 , 4.92156862,
6.38162382, 5.10784567, 6.30108784, 6.21224582, 6.50221084,
```

```
       5.51985221, 5.16412612, 6.23283235, 5.32903476, 5.25839032,
       5.32882382, 5.89753508, 5.92128255, 6.26545355, 6.57918909,
       5.55219907, 5.56483453, 5.51937934, 5.61558301, 5.39101712,
       5.68815279, 5.23225544, 5.2805354 , 6.2724663 , 5.19707213])
```

MODEL ACCURACY

[15]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
MAE=mean_absolute_error(y_test,y_pred)
MSE=mean_squared_error(y_test,y_pred)
R2=r2_score(y_test,y_pred)
print('MAE:', MAE)
print('MSE:', MSE)
print('R2:', R2)
```

```
MAE: 0.5035304415524374
MSE: 0.39002514396395427
R2: 0.403180341279623
```

OPTIMIZED MODEL

[16]:
```
import pandas as pd
import numpy as np
df=pd.read_csv("C:\\Users\\KENNY\\Downloads\\archive (1)\\winequality-red.csv")
```

[17]:
```
x=df.drop(['quality'],axis=1)
y=df['quality']
```

[18]:
```
from sklearn.model_selection import train_test_split,GridSearchCV
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪2,random_state=42)
```

[19]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
# model.fit(x_train,y_train)
# y_pred=model.predict(x_test)
```

[20]:
```
param_grid={
    'fit_intercept':[True,False],
    'copy_X':[True,False],
    'n_jobs':[-1],
    'positive':[False,True]

}
# grid_search=GridSearchCV()
```

[21]:
```
grid_search=GridSearchCV(model,param_grid,cv=6)
grid_search.fit(x_train,y_train)
best_param=grid_search.best_params_
```

```
print('Best Parameters:',best_param)
```

```
Best Parameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': -1,
'positive': False}
```

[22]:
```
best_model=LinearRegression(**best_param)
best_model.fit(x_train,y_train)
pred=best_model.predict(x_test)
pred
```

[22]:
```
array([5.34666441, 5.05631345, 5.66446972, 5.46451484, 5.72518476,
       5.27928659, 5.03421667, 5.12623347, 5.74534288, 5.68665032,
       6.13959677, 5.23386892, 5.54991474, 5.25825299, 5.44810502,
       6.46828999, 5.15018088, 5.59105157, 6.5560658 , 5.32255751,
       5.3918385 , 5.19610791, 5.94475739, 6.36197631, 5.35484893,
       5.41907575, 6.36483321, 5.35121573, 5.172392  , 6.16987311,
       5.25263058, 5.50657406, 5.75422105, 5.39101712, 5.45331031,
       5.02757499, 6.16173243, 5.68661555, 5.6486077 , 6.165471  ,
       5.52872593, 5.24414488, 6.17724727, 5.16500868, 5.87598332,
       5.81317121, 6.41982782, 5.6059474 , 5.15232137, 5.55634632,
       5.16044852, 5.10449459, 5.58371721, 6.33425313, 4.95134985,
       4.98364804, 6.01041999, 5.40809804, 5.83802638, 5.2486897 ,
       5.60717482, 5.96630957, 5.27619063, 5.30380113, 6.4949309 ,
       5.42033967, 6.34273471, 5.24618531, 6.41317317, 5.31237924,
       6.41746963, 4.74315748, 5.79362039, 5.8283184 , 6.17598768,
       5.29723707, 6.76198733, 5.89745261, 6.07833712, 6.43522754,
       5.29499011, 6.4546625 , 5.45007864, 5.69644693, 5.72368681,
       6.41233601, 5.31025119, 5.84548953, 6.31433877, 5.20585049,
       6.10141578, 5.70349712, 5.78679322, 5.93173502, 5.1852885 ,
       5.74819506, 5.17351769, 5.69336056, 4.99158806, 5.52004223,
       5.06867029, 5.13831807, 5.84991801, 5.72612872, 5.47766711,
       6.12476389, 5.73551897, 5.44180611, 6.08785125, 5.24667513,
       6.68434941, 5.26499691, 6.15359147, 4.74493131, 5.82508834,
       5.9872331 , 6.17033538, 5.50859099, 5.02156367, 5.83326942,
       6.21086737, 5.26363047, 5.75354145, 5.38942262, 5.39641713,
       5.25966957, 6.21024761, 5.69536196, 5.58586923, 5.82155344,
       5.79362039, 5.14962195, 5.01142496, 6.34824026, 5.55634632,
       5.08213438, 5.05668453, 5.3517036 , 5.11920475, 5.66948552,
       6.01614582, 6.03912287, 6.2439487 , 5.48155178, 5.86335248,
       5.26302973, 6.06162683, 5.4041289 , 5.99869245, 5.06897434,
       5.70161041, 6.14167652, 5.11821365, 5.67658854, 5.79362039,
       6.0891404 , 5.22103588, 5.90134727, 5.48941228, 5.93412645,
       6.3118134 , 5.71785286, 6.13152024, 4.9898825 , 5.39143155,
       5.63146602, 4.70626967, 5.232132  , 5.04110749, 4.99137335,
       5.20669998, 5.11005631, 6.29652093, 5.48263655, 5.73380671,
       5.86096397, 6.11131909, 5.38204246, 5.39418516, 5.11161705,
       4.74487438, 6.34043215, 5.57642863, 6.52465957, 5.18100269,
       6.37846442, 5.39147732, 5.7435927 , 6.71436012, 5.48263655,
```

```
       5.42573746, 6.08035849, 5.6017508 , 6.52660959, 5.79174569,
       5.32807323, 4.92850887, 5.40669848, 5.49983794, 6.12476389,
       5.36974106, 5.78401123, 5.48534309, 5.02135392, 6.65592712,
       5.62370825, 4.83368748, 5.73347951, 5.68074781, 6.09738854,
       5.99258428, 5.16969289, 5.7770828 , 6.59697123, 6.37009025,
       5.77981876, 5.46465189, 5.19009343, 5.80517998, 5.30830978,
       5.09158113, 6.24863165, 6.33674607, 5.99341483, 5.16829696,
       4.81289689, 5.22265229, 6.44901207, 5.48931502, 5.31886287,
       5.5589884 , 5.04938167, 6.32905554, 5.98208683, 6.04415923,
       6.12476389, 5.37906696, 5.72368681, 4.795237  , 5.03676054,
       5.68938109, 5.01079638, 5.83995808, 6.13732216, 5.24782156,
       5.56627333, 6.00210169, 5.3626292 , 6.68219105, 5.11532126,
       5.78120835, 5.62454656, 5.31952796, 5.51514228, 5.20719665,
       5.13154551, 5.48620652, 5.85075029, 5.71919777, 6.80397753,
       6.20404528, 6.04410296, 5.38204246, 6.50598024, 5.85449947,
       6.30306847, 5.05268393, 4.92613186, 5.94872379, 6.32176541,
       5.18546252, 5.8361393 , 5.40120414, 5.17199122, 5.3095161 ,
       5.49911144, 5.66556707, 6.21315993, 6.22227229, 5.26433184,
       6.48967503, 4.95165562, 5.37197617, 5.49931461, 5.3577211 ,
       5.82641444, 4.97385804, 6.03912287, 5.03990278, 5.76144224,
       5.67870975, 6.57726748, 5.67261468, 5.5851728 , 4.92156862,
       6.38162382, 5.10784567, 6.30108784, 6.21224582, 6.50221084,
       5.51985221, 5.16412612, 6.23283235, 5.32903476, 5.25839032,
       5.32882382, 5.89753508, 5.92128255, 6.26545355, 6.57918909,
       5.55219907, 5.56483453, 5.51937934, 5.61558301, 5.39101712,
       5.68815279, 5.23225544, 5.2805354 , 6.2724663 , 5.19707213])
```

[24]:
```python
MAE=mean_absolute_error(y_test,pred)
MSE=mean_squared_error(y_test,pred)
R2=r2_score(y_test,y_pred)
# results=model.score(x,y)
print('MAE:', MAE)
print('MSE:', MSE)
print('R2:', R2)
```

```
MAE: 0.5035304415524374
MSE: 0.39002514396395427
R2: 0.403180341279623
```

[ ]:

[ ]: