linear-regression

June 4, 2023

[ ]:
```python
# Importing libraries
from pandas import read_csv
from sklearn.model_selection import train_test_split
```

[ ]:
```python
def load_dataset(fname):
  data = read_csv(fname, na_values = None)
  data = data.dropna()
  dataset = data.values
  X = dataset[:,:-1]
  y = dataset[:,-1]
  return X, y, dataset
```

[ ]:
```python
# Train Test split
X, y, dataset = load_dataset('/content/drive/MyDrive/Data Analytics/water.csv')
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,
                                               random_state = 1)
print('Train', X_train.shape, y_train.shape)
print('Test', X_test.shape, y_test.shape)
```

```
Train (1347, 9) (1347,)
Test (664, 9) (664,)
```

[ ]:
```python
from sklearn.linear_model import LinearRegression
```

[ ]:
```python
# Model building
model = LinearRegression()
model.fit(X_train, y_train)
```

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

[ ]:
```python
# Model score
rsq = model.score(X_train, y_train)
print("R-square", rsq)

# R- value is almost 0. This indicates that the model does not explain any of
# the variability of the dependent variable around its mean
```

```
R-square 0.010141427189464869
```

[ ]:

```
# Model parameters
print('w0', model.intercept_)
print('w0', model.coef_)
```

```
w0 0.17291352150492684
w0 [-7.31716423e-04  1.49773166e-04  4.19386988e-06  1.02512965e-02
  9.72553630e-05 -1.75361002e-04 -2.73436083e-03  1.63901360e-04
  3.19785213e-02]
```

[ ]:

```
# Prediction
ypred = model.predict(X_test)
```

[ ]:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
```

[ ]:

```
# Evaluate the model
MSE = mean_squared_error(y_test, ypred)
RMSE = math.sqrt(MSE)
print('Mean squared error:', MSE)
print('Root Mean squared error:', RMSE)

# The average squared difference between the predicted and actual potability
# values is 0.24. Since the value is small it shows better model performance.
```

```
Mean squared error: 0.24051887278794742
Root Mean squared error: 0.49042723495738 63
```

[ ]:

```
MAE = mean_absolute_error(y_test, ypred)
print('Mean Absolute Error', MAE)

# On an average, the absolute difference between the predicted and actual
# potability values is 0.48.
```

```
Mean Absolute Error 0.4815047595342708
```