IC 272: Lab8: Prediction using Linear and Polynomial Regression

You are given with data file "`atmosphere_data.csv`" that contains the readings from various sensors installed at 10 locations around Mandi district. These sensors measure the different atmospheric factors like temperature, humidity, pressure, rain, average light, maximum light and moisture content. The goal of this dataset is to model the atmospheric temperature.

1. Split the data from winequality-red.csv into **train data** and **test data**. Train data contain **70%** of tuples and test data contain remaining **30%** of tuples. Save the train data as `atmosphere-train.csv` and save the test data as `atmosphere-test.csv`
2. Build the simple linear regression (**straight-line regression**) model to predict temperature given pressure.
   a. Plot the best fit line on the training data where x-axis is pressure value and y-axis is temperature
   b. Find the **prediction accuracy** on the training data using *root mean squared error*.
   c. Find the **prediction accuracy** on the test data using *root mean squared error*.
   d. Plot the scatter plot of *actual temperature* vs *predicted temperature* on the test data. Comment on the scatter plot.
3. Build the simple nonlinear regression model using **polynomial curve fitting** to predict temperature given pressure.
   a. Plot the best fit curve on the training data where x-axis is pressure value and y-axis is temperature.
   b. Find the **prediction accuracy** on the training data for the different values of degree of polynomial (p = 2, 3, 4, 5) using *root mean squared error (RMSE)*. Plot the bar graph of *RMSE* (y-axis) vs different values of degree of polynomial (x-axis).
   c. Find the **prediction accuracy** on the test data for the different values of degree of polynomial (p = 2, 3, 4, 5) using *root mean squared error (RMSE)*. Plot the bar graph of *RMSE* (y-axis) vs different values of degree of polynomial (x-axis).
   d. Plot the scatter plot of *actual temperature* vs *predicted temperature* on the test data for the best degree of polynomial (p). Comment on the scatter plot and compare with that of in 2(d).
4. Compute the Pearson correlation coefficient for every attribute with the attribute *temperature* (dependent variable) on the training data. Select two attributes that are highly correlated (either positively or negatively correlated) with attribute *temperature*.
   a. Build the multiple linear regression model considering only the selected two attributes to predict temperature.
      i. Plot the best fit plane on the training data where x and y-axis are the two selected attributes z-axis is temperature.
      ii. Find the **prediction accuracy** on the training data using *root mean squared error*.
      iii. Find the **prediction accuracy** on the test data using *root mean squared error*.
      iv. Plot the scatter plot of *actual temperature* vs *predicted temperature* on the test data. Comment on the scatter plot.

b. Build the multivariate polynomial regression model considering only the selected two attributes to predict temperature.
   i. Plot the best fit surface on the training data where x and y-axis are the two selected attributes z-axis is temperature.
   ii. Find the **prediction accuracy** on the training data for the different values of degree of polynomial (p = 2, 3, 4, 5) using *root mean squared error (RMSE)*. Plot the bar graph of *RMSE* (y-axis) vs different values of degree of polynomial (x-axis).
   iii. Find the **prediction accuracy** on the test data for the different values of degree of polynomial (p = 2, 3, 4, 5) using *root mean squared error (RMSE)*. Plot the bar graph of *RMSE* (y-axis) vs different values of degree of polynomial (x-axis).
   iv. Plot the scatter plot of *actual temperature* vs *predicted temperature* on the test data. Comment on the scatter plot.

5. Compare each of the regression models (all the cases from questions 2-6) based on *RMSE*.

Note:

A. **Simple and Multiple Linear Regression**:
Import the `LinearRegression` from `sklearn.linear_model`
A code snippet for prediction using linear regression:

```
regressor = LinearRegression()
regressor.fit(x, y)
```

`x` is set of univariate or multivariate training data used for building simple of multiple linear regression. `y` is corresponding dependent variable.

```
y_pred = regressor.predict(x)
```

B. **Polynomial Curve Fitting and Polynomial Regression**:
Import the `PolynomialFeatures` from `sklearn.preprocessing`
A code snippet for prediction using linear regression:

```
polynomial_features= PolynomialFeatures(degree=p)
x_poly = polynomial_features.fit_transform(x)
```

`x` is set of univariate or multivariate training data used for building simple of multiple polynomial regression.

```
regressor = LinearRegression()
regressor.fit(x_poly, y)
```

`x_poly` is set of polynomial expansions (monomials of polynomial up to degree `p`) training data used for building simple of multiple linear regression. `y` is corresponding dependent variable.

```
y_pred = regressor.predict(x)
```