Assignment 5 report

1. Perceptron Algorithm for Classification of Iris Dataset:

- Approach
  - The Iris dataset was loaded using the `load_iris` function from the `sklearn.datasets` module.
  - The data was preprocessed by splitting it into training and test sets using `train_test_split` from `sklearn.model_selection`.
  - The perceptron algorithm was implemented in the `MultiClassPerceptron` class, with methods for forward and backward propagation, softmax activation, fitting the model, and making predictions.
  - The model was trained on the training set and evaluated on both the training and test sets using the accuracy score.
- Findings
  - The perceptron classification achieved a training accuracy of 0.88 and a test accuracy of 0.9.
  - These results indicate that the perceptron algorithm performed well in classifying the Iris dataset.
- Interpretation
  - The perceptron algorithm successfully learned a decision boundary to separate the different classes of the Iris dataset.
  - The high accuracy scores suggest that the model is able to accurately classify new, unseen instances.

2. Non-linear Feature Transformation on the Concrete Compressive Strength Dataset

- Approach

    - The Concrete Compressive Strength dataset was loaded from an Excel file using the `pd.read_excel` function.
    - The features and target variables were separated.
    - The data was split into training and test sets using `train_test_split`.
    - Non-linear feature transformation was performed by generating polynomial features of degree 2 using the `polynomial_features` function.
    - Linear regression models were trained on both the original and transformed features.
    - Predictions were made on the training and test sets, and the mean squared error (MSE) and coefficient of determination (R^2) were computed.
- Findings
    - For the polynomial-transformed features:
        - MSE was 53.09 for the training set and 55.59 for the test set.
        - R^2 was 0.81 for the training set and 0.78 for the test set.
    - For the original features:
        - MSE was 110.66 for the training set and 95.98 for the test set.
        - R^2 was 0.61 for the training set and 0.63 for the test set.
- Interpretation
    - The non-linear feature transformation using polynomial features improved the performance of the linear regression model compared to using the original features.
    - The lower MSE and higher R^2 scores for the transformed features indicate that the model was able to capture more of the underlying patterns in the data, resulting in better predictions.

3. RBFs on the California Housing Prices Dataset

- Approach
    - The California Housing Prices dataset was loaded using `fetch_california_housing` from `sklearn.datasets`.

- The data was standardized using `StandardScaler`.
- RBF features were generated by computing the RBF kernel between the data points and randomly selected centroids.
- Linear regression models were trained on the original and RBF-transformed features.
- Predictions were made on the test set, and the MSE and R^2 scores were calculated.

- Findings
  - For the original features:
    - MSE was 0.555 and R^2 was 0.576.
  - For the RBF-transformed features:
    - MSE was 0.371 and R^2 was 0.717.
- Interpretation:
  - The RBF transformation improved the performance of the linear regression model on the California Housing Prices dataset.
  - The lower MSE and higher R^2 scores for the RBF-transformed features indicate that the model was better able to capture the relationships between the features and target variable, resulting in improved predictions.