CSCN8000 – Artificial Intelligence Algorithms and Mathematics

Lab 3: California Housing Prices Prediction

Dataset file: housing_dataset.csv

Data Preprocessing Tasks (21 Points):

1. (4 points) Detect and handle outliers for the "median_house_value" field using:
    a. Apply the whiskers approach to identify outlier rows.
    b. (Optional) Apply Z-score normalization and choose appropriate threshold to match the outliers from the whiskers approach (1 bonus point)
2. (5 points) Handle missing values in the dataset:
    a. For numerical features:
        i. Choose whether mean/median is better to impute the values
        ii. (Optional) look for other ways to impute based on other categorical variables (1 bonus point)
    b. For categorical features: Choose the appropriate imputation method.
3. (3 points) Investigate the existence of errors/inconsistencies in the fields and solve them.
4. (3 points) Normalize numerical features using appropriate method based on feature characteristics.
5. (3 points) Encode categorical variables using using appropriate method based on feature characteristics.
6. (3 points) Engineer one new feature based on existing features.

Descriptive Analytics Tasks (6 Points):

1. (3 points) Investigate the distribution of housing prices across different ocean proximities in California.
2. (3 points) Analyze the relationship between median income and housing prices.

ML Model Training and Testing Tasks (21 Points):

The objective of this section is to develop a linear regression model to predict the median house value in a given area, based on several predictors from the California Housing dataset. You will implement the linear regression algorithm **from scratch, without using high-level libraries like scikit-learn (only Numpy and Pandas mainly)**. This exercise will deepen your understanding of the linear regression algorithm based on the discussed lecture content.

1. (1 point) Split the **cleaned** data from the first section into training and testing sets (e.g., 80% training, 20% testing).
2. (8 points) Implement the closed-form solution to linear regression:

$$\theta = (X^T X)^{-1} X^T y$$

   a. where θ is the vector of weights (including the bias (b)), X is the feature matrix **with a column of ones added to represent the intercept**, and y is the vector of target values.
   b. Use your implementation to compute the coefficients for your linear regression model on the ***training dataset split.***
3. (2 points) Print the learned coefficients (weights) of the model.
   a. **Comment** on which feature the model gave higher weight to in the weight vector.
4. (4 points) Utilize the learned coefficients to generate predictions on the test dataset split, where:

$$\hat{y} = X\theta$$

5. (4 Points) Implement the following evaluation metrics using NumPy functions only:
   a. Mean Absolute Error (MAE)
   b. Mean Squared Error (MSE)
6. (2 Points) Evaluate the model's performance on the test set using the implemented metrics and report your results.
   a. **Comment** on the model's performance.
7. [BONUS] (1 Point) Is there an additional metric to be used to give a more intuitive measurement of the model's performance, if yes please implement it from scratch and report the performance of your model with a comment on it.

Organization Criteria (2 Points)

1. (2 points) Provide an organized notebook at the end with clear section, markdown comments and printed outputs. Make sure no errors are printed in the final submission.

Deliverables

1. Include all your findings and task solutions in one Jupyter notebook (.ipynb) that shows all the printed cell outputs. Prepare a html version (.html) of the notebook file. Both files should be named as follows: [Full Name]_[Student ID]_[Section Number]_Lab_3.[html/ipynb].
2. Submit both .html and .ipynb files on eConestoga in Lab 3 under the Assignments section.