Summative Lab Exercise #1: Boston Pricing Prediction

Task Overview

In this assignment, your task is to predict the median value of owner-occupied homes in Boston (MEDV) using the Boston house-price dataset. You will achieve this by applying linear regression and its variants, such as Ridge Regression, Lasso Regression, etc. Your primary objectives include:

1. **Understanding the Data**: Explore the dataset to understand the relationships between input features and the target variable.
2. **Data Preprocessing**: Apply traditional preprocessing techniques to clean, normalize, and transform the data appropriately.
3. **Modeling**: Implement and evaluate linear regression models, focusing on different variants and hyperparameter tuning.
4. **Analysis and Interpretation**: Analyze the model results to draw meaningful conclusions and insights regarding feature importance and model performance.

The final deliverable will include your code implementation and a well-documented report in a 2-column format (IEEE/ACM style), highlighting your methodology, experiments, results, and conclusions.

Dataset Overview

The dataset you will work with is the Boston house-price dataset, which originates from a study by Harrison, D. and Rubinfeld, D.L. The study explored the relationship between housing prices and various socio-economic and environmental factors. The dataset is widely used for regression analysis and offers a great opportunity to delve into linear models and their variants.

Dataset Attributes

The dataset consists of 13 input features and one target variable:

Input Features:

1. **CRIM**: Per capita crime rate by town.

2. **ZN**: Proportion of residential land zoned for lots over 25,000 sq.ft.
3. **INDUS**: Proportion of non-retail business acres per town.
4. **CHAS**: Charles River dummy variable (1 if tract bounds river; 0 otherwise).
5. **NOX**: Nitric oxides concentration (parts per 10 million).
6. **RM**: Average number of rooms per dwelling.
7. **AGE**: Proportion of owner-occupied units built prior to 1940.
8. **DIS**: Weighted distances to five Boston employment centers.
9. **RAD**: Index of accessibility to radial highways.
10. **TAX**: Full-value property-tax rate per $10,000.
11. **PTRATIO**: Pupil-teacher ratio by town.
12. **B**: Calculated as 1000(Bk−0.63)21000(Bk - 0.63)^21000(Bk−0.63)2, where BkBkBk is the proportion of Black residents by town.
13. **LSTAT**: Percentage of lower-status population.

Target Variable:

1. **MEDV**: Median value of owner-occupied homes in $1000's.

Detailed Task Objectives

1. **Exploratory Data Analysis (EDA):**
    - Perform EDA to gain insights into the data distribution and relationships between features.
    - Visualize correlations between the input features and the target variable (MEDV).
    - Identify and handle any missing data or outliers.
2. **Data Preprocessing:**
    - **Normalization/Scaling**: Apply appropriate scaling techniques to ensure the features are on a similar scale, which is crucial for linear models.
    - **Encoding**: Handle categorical variables (if any) using suitable encoding methods.
    - **Feature Selection**: Analyze feature importance and potentially reduce dimensionality by selecting relevant features.
3. **Model Implementation:**
    - **Baseline Model**: Start with a simple linear regression model to establish a baseline performance.
    - **Advanced Models**: Implement Ridge Regression, Lasso Regression, and Elastic Net to improve the model by addressing overfitting and underfitting issues.
    - **Hyperparameter Tuning**: Experiment with different hyperparameters (e.g., regularization strength) to optimize model performance.
4. **Model Evaluation:**
    - Evaluate the models using metrics such as Mean Squared Error (MSE), R-squared (R^2), and Root Mean Squared Error (RMSE).
    - Compare the performance of different models and discuss the trade-offs.

5. **Analysis and Interpretation:**
    - Interpret the coefficients of the linear models to understand the impact of each feature on the target variable.
    - Discuss the significance of regularization in reducing overfitting.
    - Provide insights into which features are most influential in predicting housing prices.
6. **Documentation:**
    - Submit a comprehensive report detailing your methodology, experiments, results, and conclusions.
    - The report should be in a 2-column format (IEEE or ACM) and include sections such as Introduction, Methodology, Experiments, Results, and Conclusions.

Criteria for Evaluation

Your submission will be evaluated based on the following five rubrics, each worth a maximum of 4 points. The total possible score is 20 points.

Code Quality and Implementation (0-4 points)

- **4 points**: Code is well-organized, with clear structure and appropriate use of functions and classes. The code is thoroughly commented, easy to follow, and regression models are correctly implemented.
- **3 points**: Code is organized and functional, with adequate commenting. Regression models are implemented correctly, though there may be minor issues in code structure or clarity.
- **2 points**: Code is functional but may lack structure or sufficient comments. Some regression models might be incorrectly implemented or missing. The workflow may require effort to understand.
- **1 point**: Code is disorganized, with minimal commenting. Regression models may be missing or incorrectly implemented. The workflow is difficult to follow.
- **0 points**: Code is non-functional or missing.

Data Preprocessing and Feature Engineering (0-4 points)

- **4 points**: Comprehensive preprocessing steps are applied, including handling missing values, scaling features, and encoding where necessary. Feature selection or dimensionality reduction is well justified.
- **3 points**: Appropriate preprocessing steps are taken with some feature engineering. There might be minor issues or omissions, but overall, the preprocessing is sound.
- **2 points**: Basic preprocessing is applied, but important steps may be missing or incorrectly done (e.g., improper handling of missing values). Feature engineering is minimal.

- **1 point**: Preprocessing is poorly executed with significant issues or omissions. No feature engineering is present.
- **0 points**: No preprocessing is applied.

Model Selection, Implementation, and Hyperparameter Tuning (0-4 points)

- **4 points**: Regression models are correctly implemented. Hyperparameter tuning is extensive and well-justified, showing a good understanding of the models.
- **3 points**: Regression models are implemented, with some hyperparameter tuning. The models are compared, but the analysis may lack depth.
- **2 points**: Most regression models are implemented, but there might be issues or omissions. Hyperparameter tuning is minimal or incorrectly applied.
- **1 point**: Few or none of the regression models are correctly implemented. Little to no hyperparameter tuning is done.
- **0 points**: No models are implemented, or the implementations are entirely incorrect.

Results and Analysis (0-4 points)

- **4 points**: Results are thoroughly analyzed using appropriate metrics (e.g., MSE, R^2). The analysis includes discussions on model performance, feature importance, and the impact of preprocessing and hyperparameters. Visualizations effectively support the analysis.
- **3 points**: Results are analyzed with appropriate metrics. The analysis covers key aspects of model performance, with some discussion on feature importance. Visualizations are present but might not be fully utilized.
- **2 points**: Results are provided, but the analysis is basic or lacks depth. Metrics are used but not fully explained.
- **1 point**: Results are incomplete or poorly analyzed. Metrics may be incorrect or unexplained. Visualizations may be irrelevant or missing.
- **0 points**: No results or analysis are provided.

Documentation Quality (0-4 points)

- **4 points**: Documentation is comprehensive, well-organized, and follows the required 2-column format (IEEE/ACM). All sections (Introduction, Methodology, Experiments, Results, Conclusions) are clearly explained with justifications and references. The report meets the length requirement.
- **3 points**: Documentation is complete and organized, though there may be minor issues with clarity or formatting. All required sections are present, but some explanations might lack depth.
- **2 points**: Documentation is basic, covering required sections, but may have issues with clarity, organization, or formatting. Some sections might be incomplete or insufficiently explained.
- **1 point**: Documentation is poorly organized or missing sections. Explanations are unclear, and the format may not be followed correctly.

- **0 points**: No documentation is provided, or it does not meet the assignment requirements.

Submission Guidelines

Your submission should include the following two components, each uploaded separately:

1. **Code Implementation:**
    - **File Type**: Jupyter Notebook (.ipynb)
    - **Filename**: Name your file `Boston_House_Price_Prediction.ipynb` (for Jupyter Notebook)
    - **Content**: Ensure that your code includes:
        - Data loading and initial exploration.
        - Data preprocessing steps (e.g., handling missing values, scaling, encoding).
        - Implementation of linear regression and its variants (Ridge, Lasso, Elastic Net).
        - Hyperparameter tuning and model evaluation.
        - Results visualization and analysis.
    - **Note**: Ensure that all code cells are executed and outputs are visible before submission.
2. **Documentation:**
    - **File Type**: PDF document.
    - **Filename**: Save your report as `Boston_House_Price_Report.pdf`.
    - **Content**: The document should follow the 2-column format (IEEE or ACM style) and include the following sections:
        - **Introduction**: Briefly describe the problem, dataset, and objectives.
        - **Methodology**: Explain your approach to data preprocessing, feature selection, and model implementation.
        - **Experiments**: Detail the experiments conducted, including different models, hyperparameters, and their justifications.
        - **Results and Analysis**: Present the results of your models, including relevant metrics and visualizations, and discuss the findings.
        - **Conclusions**: Summarize the key insights and takeaways from your analysis.
        - **References**: Cite any sources or references used in your work.

2. Submission Format

- **Code Submission**:
    - Upload your Jupyter Notebook (.ipynb)
    - Ensure the file is named according to the specified naming convention.

- **Documentation Submission**:
    - Upload your PDF report as a separate file.
    - Ensure the PDF follows the 2-column format and includes all required sections.
- **Final Submission**:
    - Both the notebook/script and the PDF report should be uploaded separately to the submission platform.

3. Submission Instructions

- **Deadline**: Submit your files by August 27, 2024.
- **Submission Platform**: Upload your Jupyter Notebook and PDF report to the corresponding link in the Assignments Tab
- **Late Submissions**: Late submissions are accepted but will be capped at a maximum grade of 60%.

Important Notes

- **Code Execution**: Ensure your code runs successfully without errors. Submissions with non-functional code may receive lower grades.
- **Documentation Quality**: Pay attention to the clarity, organization, and formatting of your PDF report. A well-structured report is crucial for full marks.

Make sure both components are complete and follow the specified guidelines to ensure a smooth evaluation process.

Late Submission Policy

- Late submissions are accepted but will only be eligible for a maximum of 12 points (60%). All other criteria will be graded as specified above, but the final score will be capped.