Predicting House Prices using Scikit-Learn and Python

Name : Maryam Zubair
Student Id : 19709

Predicting House Prices

Introduction:

In this project, I am training a linear regression model on the California housing dataset using Scikit-Learn. The dataset includes information such as the population, median income, and median housing price of each block group in California. The program includes data preprocessing steps, such as feature scaling and transformation, and then trains a linear regression model on the data. It also includes evaluation of the model's performance using mean squared error and confidence intervals.

Design/Implementation :

1. The code first imports necessary modules such as NumPy, pandas, and scikit-learn. After that loading the housing dataset using pandas and splits it into training and testing sets.
2. Next, the code defines a data transformation pipeline that includes feature scaling, handling missing values, and adding extra features. It also defines a linear regression model.
3. The code then fits the pipeline and the linear regression model to the training data and evaluates the model's performance using mean squared error (MSE) and root mean squared error (RMSE).
4. Finally, the code computes a confidence interval for the test set RMSE using either t-scores or z-scores, and shows how to use joblib to persist the model to disk.

Testing:

In this project, we have evaluated the performance of a linear regression model:

1. We split the housing dataset into a training set and a test set, fit the pipeline on the training set, and evaluated the performance on the test set using the mean squared error (MSE) and root mean squared error (RMSE) metrics.
2. We also used cross-validation to estimate the generalization error of the model. We used RandomizedSearchCV to search for the best hyperparameters for the model and evaluated the performance of the model using cross-validation.

Enhancement Ideas

1. **Hyperparameter tuning:** In this project, I used default hyperparameters of the models. For future, I can use techniques such as grid search, random search, or Bayesian optimization to find the best hyperparameters for the model.
2. **Online learning:** In this project, I trained the model on the entire dataset. Whereas in the real-world scenarios, new data is generated continuously so the model needs to be updated. Which is why we can use online learning to update the model on new data in real-time.
3. **Deep learning:** Only linear regression and random forest models is used in this project. For future, I can explore the use of deep learning models such as neural networks, which can be more powerful in handling complex patterns in the data.

Conclusion:

This project provides an excellent example of how to approach a regression problem in a real-world setting and how to use various techniques to optimize the performance of a machine learning model.

Our best model achieved an RMSE of approximately 47,000 dollars, which is a significant improvement over the naive baseline. However, there is still room for improvement, and we suggested some potential directions for future work.

Bibliography/References

1. Introduction to Feature Scaling - Standardization and Normalization
2. Training and Testing Data in Machine Learning
3. Linear Regression Guide - In Python
4. How to implement Cross Validation in Scikit Learn