Oscar Eduardo Nieto Espitia - A0105090

Performance analysis for Boston Housing Prediction Model

The performance analysis of the Boston Housing Prediction Model aims to provide a comprehensive assessment of the model´s ability to predict median house values in the suburbs of Boston. The primary objectives of this analysis are to assess the model's bias and variance, discern whether it suffers from underfitting or overfitting, and apply regularization techniques to enhance its predictive power.

One of the central questions I aim to answer is whether the model has struck the right balance between complexity and simplicity. I examine the potential signs of underfitting, where the model may be too simplistic to capture the underlying patterns, as well as overfitting, where it might be memorizing the training data rather than learning from it.

Dataset:
1. **CRIM (Crime Rate):** This feature represents the per capita crime rate by town.
2. **ZN (Proportion of Residential Land):** Proportion of residential land zoned for large lots over 25,000 square feet.
3. **INDUS (Non-Retail Business):** The proportion of non-retail business acres per town, which can give insights into the economic activities in the area.
4. **CHAS (Charles River)**: This is a binary variable. It equals 1 if a tract bounds the Charles River and 0 otherwise.
5. **NOX (Nitrogen Oxides Concentration):** concentration of nitrogen oxides in parts per 10 million.
6. **RM (Average Number of Rooms):** The average number of rooms per dwelling.
7. **AGE (Proportion of Older Houses):** AGE is the proportion of owner-occupied units built before 1940.
8. **DIS (Distance to Employment Centers):** This is a weighted mean of distances to five Boston employment centers.
9. **RAD (Accessibility to Highways):**Index of accessibility to radial highways.
10. **TAX (Property Tax Rate):** Full-value property-tax rate per $10,000.
11. **PTRATIO (Pupil-Teacher Ratio):** Pupil-teacher ratio by town.
12. **B (Proportion of Black Residents):** BLACK is a derived feature, calculated as 1000 times the square of the proportion of residents of African American descent, providing demographic information.
13. **LSTAT (Lower Status Population):** LSTAT represents the percentage of the lower status population.
14. **MEDV (Median Home Value):** This is the target variable. MEDV is the median value of owner-occupied homes in thousands of dollars. It's the value I aim to predict in regression tasks.

Splitting the dataset

Using separate sets for training, testing and validation is a fundamental practice for the following reasons:

Model Performance Evaluation: One of the primary goals of machine learning is to build models that generalize well to unseen data. By splitting your dataset into three subsets, we can assess your model's performance accurately.

Identify Overfitting: The test set serves as a benchmark to check for overfitting. If a model performs well on the training data but poorly on the test data, it's a sign of overfitting.

Hyperparameter Tuning: To optimize the model's hyperparameters, we need a validation set. You can train multiple models with different hyperparameters and select the one that performs best on the validation set

In order to accomplish that I split the dataset adding a random factor:
- Train - 70%
- Test - 15%
- Validation - 15%

```python
oston_housing = datasets.boston_housing.load_data()
features = ["CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM", "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B",
"LSTAT", "MEDV"]
(X, y), _ = boston_housing

# Split the data into training (70%), validation (15%), and test (15%) sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```
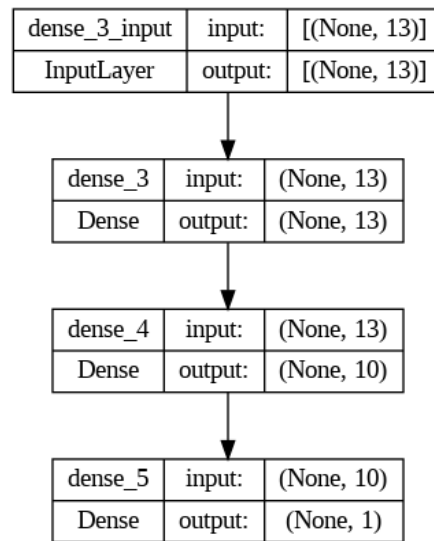
Scaling the data

The features in this dataset have different units and magnitudes. If we don´t scale the features, some features with larger magnitudes can dominate the learning process, leading to biased results. That's why que scale all our features on the different datasets (Train, Test, Validation).
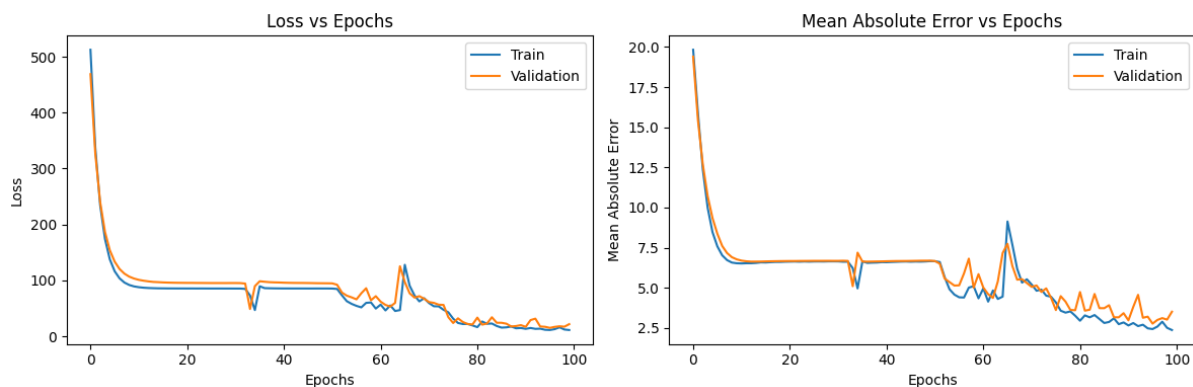
```python
scaler = RobustScaler()

X_train_prep =
scaler.transform(X_val)
X_test_prep = scaler.transform(X_test)
```

Creating the model

dense_3_input	input:	[(None, 13)]
InputLayer	output:	[(None, 13)]

dense_3	input:	(None, 13)
Dense	output:	(None, 13)

dense_4	input:	(None, 13)
Dense	output:	(None, 10)

dense_5	input:	(None, 10)
Dense	output:	(None, 1)

Performance of the model

In order to evaluate the performance of the model, I will use the Mean Squared Error (MSE) and Mean Absolute Error (MAE).

- **MSE** measures the average of the squared differences between the predicted values and the actual target values. It punishes large errors more than MAE, making it sensitive to outliers.
- **MAE** measures the average of the absolute differences between the predicted values and the actual target values. MAE is less sensitive to outliers compared to MSE, making it a more robust metric when dealing with data that may have extreme values.
- **R-squared** measures the proportion of the variance in the dependent variable that is predictable from the independent variables.

Training Loss (MSE): 26.90007972717285
Validation Loss (MSE): 35.53453063964844

Training (MAE): 3.1551523790291864
Validation (MAE): 3.657471097101931

R-squared: 0.617176070157776

There's a **Medium level of overfitting**, as indicated by the higher validation MSE compared to training MSE. This is also shown in the graph. However, this overfitting is not extreme.
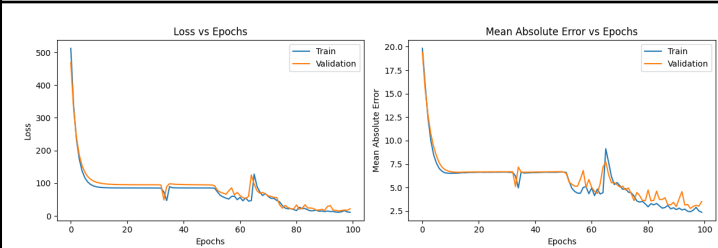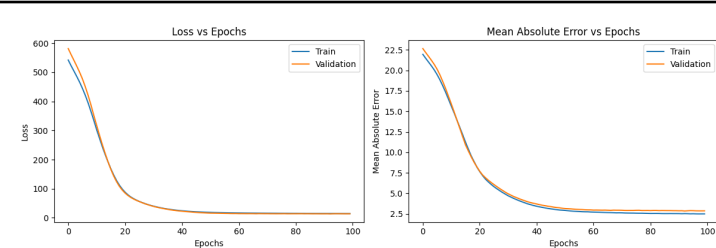
The difference between training and validation MAE is relatively small, which is a positive sign. It indicates that your model is **not suffering from significant bias**.

Indicates that the model explains approximately 61.7% of the variance in the target variable. It has a **medium level of variance**.

Model Improvements

Adjusting the Learning Rate: If the initial learning rate was too high, the model might have trouble converging and could overshoot the optimal weights. This can be seen in the first graph where the model overshoots most of the time making the MSE and MAE inconsistent.

Removing Outliers with Z-Score: Outliers are data points that significantly deviate from the majority of the data. In some cases, outliers can have a disproportionately large influence on the model's training. By removing outliers using a Z-score threshold (usually set to 3 standard deviations), I effectively eliminate data points that are far from the mean of the feature distribution.

Old Model	New Model
Training Loss (MSE): 26.90007972717285 **Validation Loss (MSE)**: 35.53453063964844 **Training (MAE)**: 3.1551523790291864 **Validation (MAE)**: 3.657471097101931 **R-squared:** 0.617176070157776	**Training Loss (MSE)**: 6.166579246520996 **Validation Loss (MSE)**: 13.156786918640137 **Training (MAE)**: 1.7316126646395928 **Validation (MAE)**: 2.6917675474415654 **R-squared:** 0.8231331654721317