If the bottom values (lower target values) are more accurately predicted than the top values (higher target values), it suggests that your model is performing better at predicting lower values of the target variable compared to higher ones. This can happen for several reasons, and here are a few possible explanations:

1. Model Bias Toward Lower Values (Underfitting for Higher Values)

If the model consistently performs better on the lower values of the target, it might be biased toward predicting lower values. This could be because the model learned more about the lower range of the target variable during training, or it may be due to a skewed distribution of the target variable.

For example, if most of the training data consists of lower target values, the model may have "learned" to predict those better while struggling with the higher values because there is less data or the relationships are more complex for those higher values.

What to check:

Inspect the distribution of your target variable in both the training and test sets. If the target variable is skewed (e.g., there are fewer higher values), the model might be more accurate at predicting the more frequent lower values.

Look at your loss function (e.g., MSE, MAE) and check if it's emphasizing certain parts of the target range.

2. Non-Linear Relationships Between Features and Target

The model may be underperforming on higher target values because there are non-linear relationships between the features and the target in that range, which the model has trouble capturing. Some models, particularly simpler ones (like linear regression), might struggle with non-linear relationships.

For example, if the higher target values are the result of more complex interactions between features, a linear model or a model with insufficient capacity might not capture this complexity.

What to check:

Check if the relationship between the features and target is non-linear, especially in the higher range of the target values. If so, consider using a more powerful model (e.g., a decision tree, random forest, or neural network) or apply transformations to your features.

3. Outliers or Noise in the Higher Target Range

If there are outliers or high variability in the higher target values, the model might not perform well on them. Outliers can significantly influence the model’s predictions and lead to poor performance in certain parts of the target range.

For instance, if the higher target values are rare or are subject to random fluctuations (noise), the model may have trouble accurately predicting them.

What to check:

Check the higher values of your target variable to see if there are any outliers or high-variance points that the model might be struggling to predict.

You can try removing outliers or using robust models like Random Forest or Gradient Boosting that can handle outliers better.

4. Incorrect Model Scaling or Transformation

If you applied scaling or transformation to your target variable (e.g., log-transforming a highly skewed target), the model might be more accurate at predicting the transformed values for lower targets but struggle with the higher values.

For example, if the target values are heavily skewed, a log transformation could make the lower values seem more predictive, but the model might not be correctly interpreting the scaling for the higher values.

What to check:

Review any transformations applied to the target variable during preprocessing (like scaling or log-transformation) and ensure that these transformations are appropriately applied and inverse-transformed when making predictions.

5. Imbalanced Data in Target Values

If the higher target values are less frequent in your dataset, the model may not have learned to predict them well. This can lead to a performance gap between lower and higher values.

For instance, if you are predicting a price, age, or some other continuous variable, and most of your training data has lower values, the model may not be able to generalize well to the higher values, especially if those occur less frequently.

What to check:

Check the distribution of your target variable and ensure that the model is exposed to enough higher value samples during training. If necessary, use techniques like stratified sampling or SMOTE (Synthetic Minority Over-sampling Technique) to balance the representation of the target variable.

6. Loss Function Not Suitable for Target Range

The loss function used to train the model could be having a greater impact on lower values than higher ones. Some loss functions (like Mean Squared Error) can disproportionately penalize errors in predictions for lower values, leading the model to focus on minimizing errors in that range.

Mean Absolute Error (MAE), for example, is less sensitive to large errors compared to Mean Squared Error (MSE), which can penalize large errors more harshly.

What to check:

Review the loss function you’re using and see if it’s giving more importance to the lower range of target values. Consider trying different loss functions or adjusting their weighting.

7. Insufficient Model Complexity

If the model lacks sufficient capacity (e.g., too few parameters or too simple a model), it may not have learned the complex patterns required to predict higher target values effectively.

What to check:

Evaluate whether the model is too simple for the complexity of the task. You might need to switch to a more complex model (e.g., Random Forest, Gradient Boosting, or Neural Networks) or adjust the model's hyperparameters (e.g., increasing the number of trees or neurons).

Conclusion

To summarize, if the bottom part of the target values (lower values) are more accurately predicted than the top part (higher values), it may indicate:

A distribution bias where the model is better at predicting more frequent lower target values.

Possible noise or outliers in the higher range of target values.

Non-linearities or complex relationships that the model isn't capturing well for higher values.

A mismatch in scaling or transformation between training and prediction.

You can address these issues by:

Analyzing the distribution of your target variable.

Ensuring sufficient representation of higher values in the training data.

Trying more complex models.

Adjusting your preprocessing steps, like feature scaling or transformations.

By exploring these avenues, you should be able to improve the model’s performance on the higher target values.