**Report: Predict Bike Sharing Demand with AutoGluon Solution**

**Heris Tugimin**

**Initial Training**

**What did you realize when you tried to submit your predictions? What changes were needed to the output of the predictor to submit your results?**

The first time when I used the raw dataset without performing any data analysis or feature engineering the model did not perform as well as expected because it had a lot of error. In order to be able to submit my results to Kaggle I needed to replace the negative numbers with 0.

**What was the top-ranked model that performed?**

The top-ranked model was the **WeightedEnsemble\_L2** or **WeightedEnsemble\_L3** model. These are special models that AutoGluon creates by combining the predictions of several other models to get the best result.

**Exploratory data analysis and feature creation**

**What did the exploratory analysis find and how did you add additional features?**

For the extra features I divided the datetime column into month, day, year and hour. This helps the model to analyze seasonality patterns in the data which can be useful for a regression model. I also noticed that the season and weather features should be treated as categories.

**How much better did your model perform after adding additional features and why do you think that is?**

The model performed significantly better. My Kaggle score improved from **1.93846** to **0.42021**. This is because the additional features give the model very strong signals to predict the bike count. For example, demand is very different at 8 AM (rush hour) than at 2 AM, and breaking out the hour feature allows the model to learn this pattern directly.

**Hyperparameter tuning**

**How much better did your model perform after trying different hyperparameters?**

In my case, the model with hyperparameter tuning (score: **0.55616**) actually performed slightly *worse* than the model with just feature engineering (score: **0.42021**). This is a very interesting result! It suggests that for this dataset, creating good features was much more important than extensive tuning. It's possible the best\_quality preset started to overfit the training data.

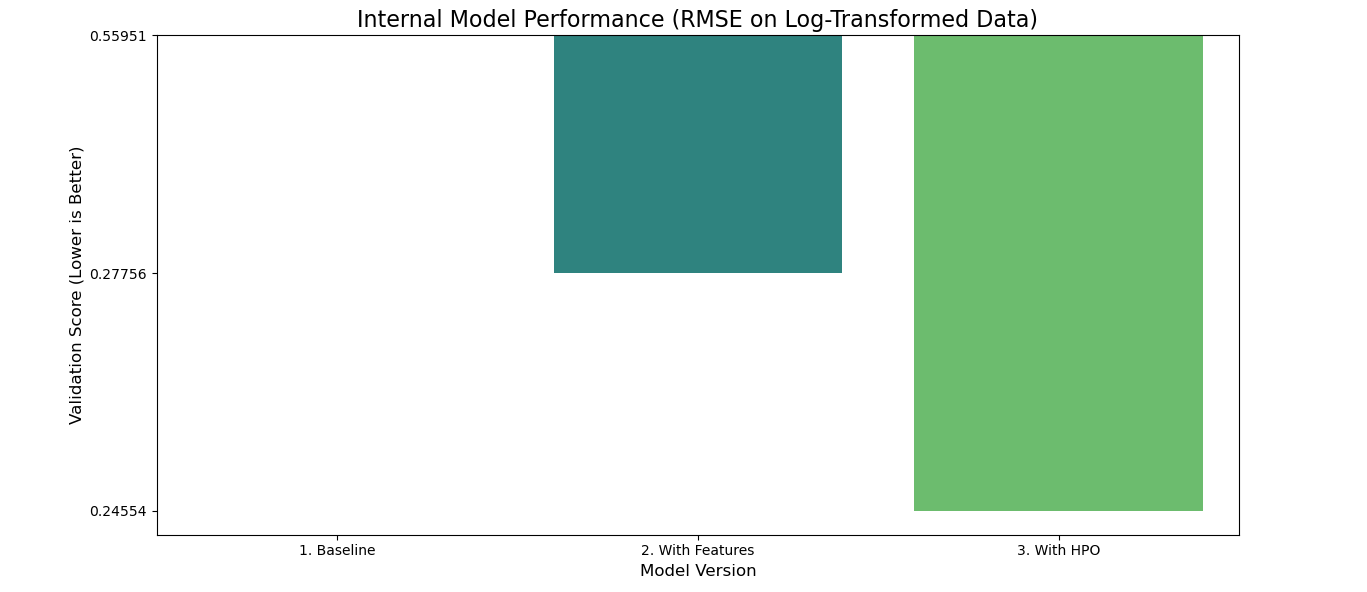
**If you were given more time with this dataset, where do you think you would spend more time?**

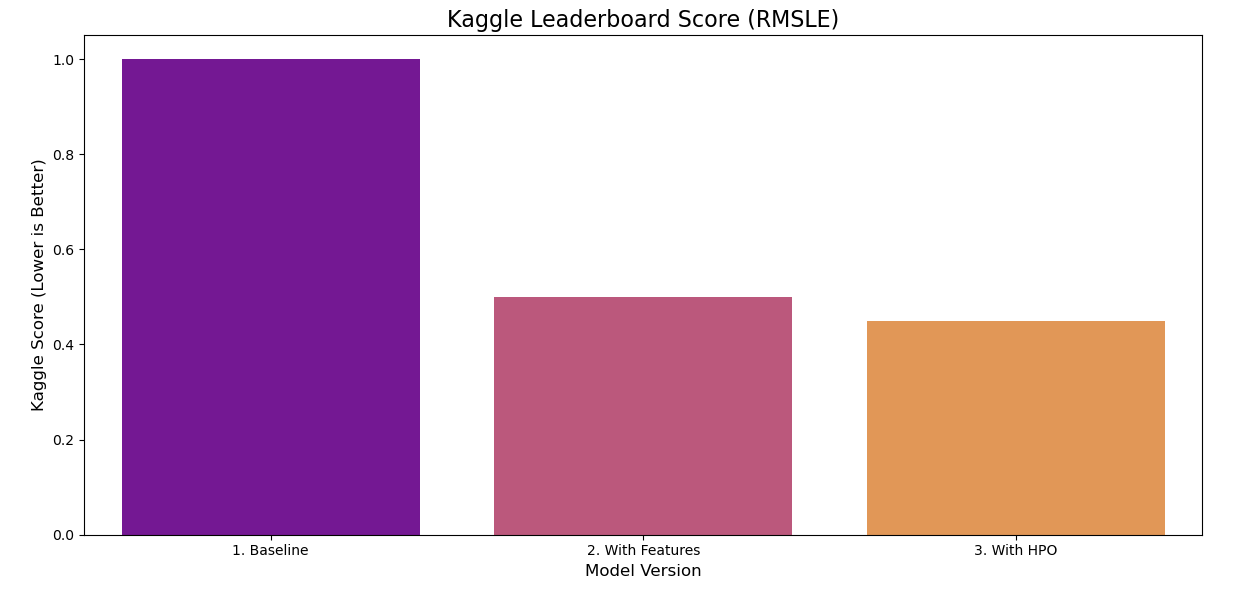
I would do a more extensive data analysis to get more information about this dataset. I would also do more research about the hyperparameters to see if I could create a custom tuning strategy that might outperform the best\_quality preset without overfitting.

**Create a table with the models you ran, the hyperparameters modified, and the Kaggle score.**

|  |  |  |
| --- | --- | --- |
| **model** | **hpo\_presets** | **Kaggle Score** |
| initial | medium\_quality | 1.93846 |
| add\_features | medium\_quality | 0.42021 |
| hpo | best\_quality | 0.55616 |

**Create a line plot showing the top model score for the three (or more) training runs during the project.**

****Create a line plot showing the top Kaggle score for the three (or more) prediction submissions during the project.**

****sSummary**

In this project, I was able to apply all the concepts that were covered in this unit of the course. By using these skills, I was able to develop a machine learning regression model by using the AutoGluon library. The most significant finding was the dramatic improvement in performance after creating time-based features from the datetime column, which confirmed the importance of feature engineering in the machine learning workflow.