Report: Predict Bike Sharing Demand with AutoGluon Solution

NAME: Ugama Benedicta Kelechi

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?

When submitting the predictions, I realized that AutoGluon can output negative values for the target variable (count). However, the Kaggle competition expects non-negative integers. Therefore, I needed to set all negative values in the predictions to 0 before saving the submission file.

What was the top ranked model that performed?

The top-performing model was an ensemble created by AutoGluon, typically WeightedEnsemble_L2. This model aggregates predictions from various base models like LightGBM, CatBoost, and Neural Networks to optimize performance.

Exploratory data analysis and feature creation

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

During EDA, I noticed that datetime had strong cyclical patterns. I extracted hour, day, and month from datetime. These features help the model understand time-of-day, day-of-week, and seasonal trends in bike demand.

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

The model's performance improved significantly (reduced RMSE) after adding these time-based features. These features provided contextual clues that the model previously lacked, such as peak commuting hours and weekend usage patterns.

Report: Predict Bike Sharing Demand with AutoGluon Solution

Hyper parameter tuning

How much better did your model perform after trying different hyper parameters?

Hyperparameter tuning helped squeeze out additional performance. By adjusting the training strategy (e.g., limiting model types, increasing training iterations), the model better generalized to unseen data. The improvement was modest but consistent.

Future Work

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

I would explore:

- Advanced time-series features (e.g., lag features, rolling averages)

- Outlier detection and removal

- Custom ensembling or stacking

- Additional categorical encoding strategies

- Deep learning models like LSTMs if time dependencies prove important

Model Table

Model Table:

model	hpo1	hpo2	hpo3	score
initial	None	None	None	0.52983
add_features	Added hour/day/month	None	None	0.46827
hpo	+custom presets	+longer training	+model limits	0.45421

Summary

Report: Predict Bike Sharing Demand with AutoGluon Solution

Using AutoGluon, I quickly built a baseline model with minimal effort, and then systematically improved it through feature engineering and hyperparameter tuning. Adding time-based features like hour, day, and month captured essential demand patterns. Hyperparameter tuning yielded modest gains. Overall, AutoML significantly accelerated the ML workflow and led to a competitive result on Kaggle.