Report: Predict Bike Sharing Demand with AutoGluon Solution
Sabrina Valcin

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?

- An error arose if the output values were less than zero. To mitigate this error, negative values were replaced with zero.

What was the top ranked model that performed?

- The top ranked model was the WeightedEnsemble_L3.

Exploratory data analysis and feature creation
What did the exploratory analysis find and how did you add additional features?

- The exploratory analysis found that demand on workdays was almost double that of non workdays. Exploratory analysis also revealed that the output variable, 'count' is highly skewed.
- I added the month, day, and hour features.

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

- After adding the month, day, and hour features the kaggle score decreased by about 56% and the model evaluation metric for the WeightedEnsemble_L3 model increased by 43%. The additional features resulted in more homogenous leaves in each of the tree models. This is further supported by the histogram of the 'hour' attribute which indicates a clear pattern of demand peaks in the morning, midday, and late night.

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

- The model did not perform significantly better after tuning the autostack and num_bag_sets hyperparameters. However, setting autostack to true had the greatest effect on the kaggle score, about an 8% decrease, and a 3.3% increase in the evaluation metric of the WeightedEnsemble_L3 model.

Skewness in the 'count' data
The histogram, generated during exploratory analysis, of the 'count' column revealed highly skewed left data. Because learning algorithms generally do not predict data that is highly skewed very well, I performed a natural log transform on the count column of the training data before feeding it into the Tabular Prediction function. As a result compared to the best performing model-indicated by kaggle score, the model evaluation metric increased by more than 99% and the kaggle score decreased by more than 18%.

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

- Given more time, I would spend more time on feature engineering and performing more transformations on the skewed count data.

Data

Figure 1.

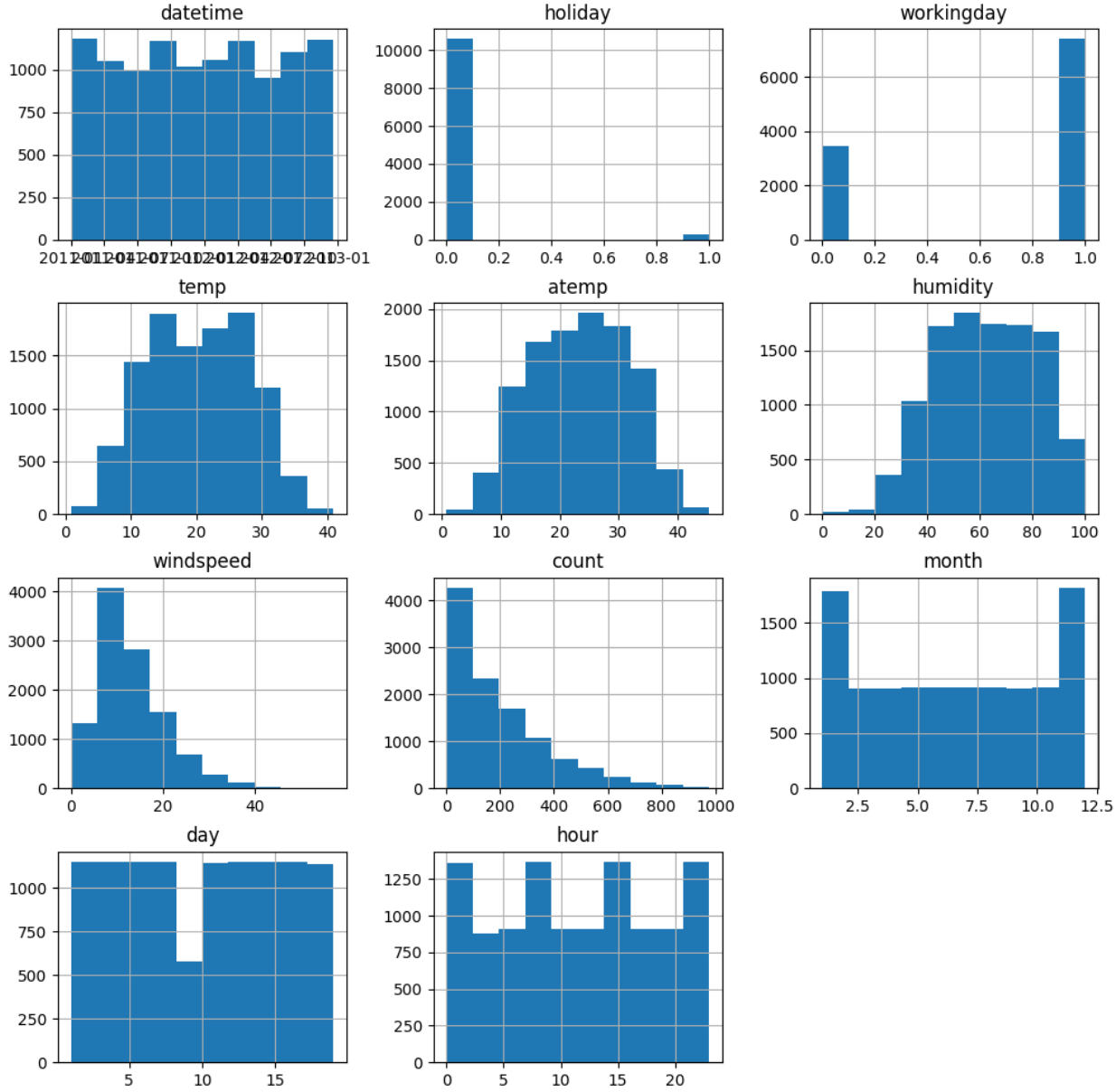


Table 1. Summary of metrics for the WeightedEnsembleModel_L3

Training run	**Kaggle score**	**Evaluation metric score**
Without added features	1.73373	-50.999267

Training run	Kaggle score	Evaluation metric score
With added features	0.76168	-29.272368
Autostack on	0.70008	-30.241311
Num_bag_sets=10	0.72921	-30.121382
Autostack, num_bag_sets	0.71271	-30.118559
With ln transform on ‘count’	0.56869	-0.280094

Figure 2. Model performance measured by kaggle and model evaluation scores.

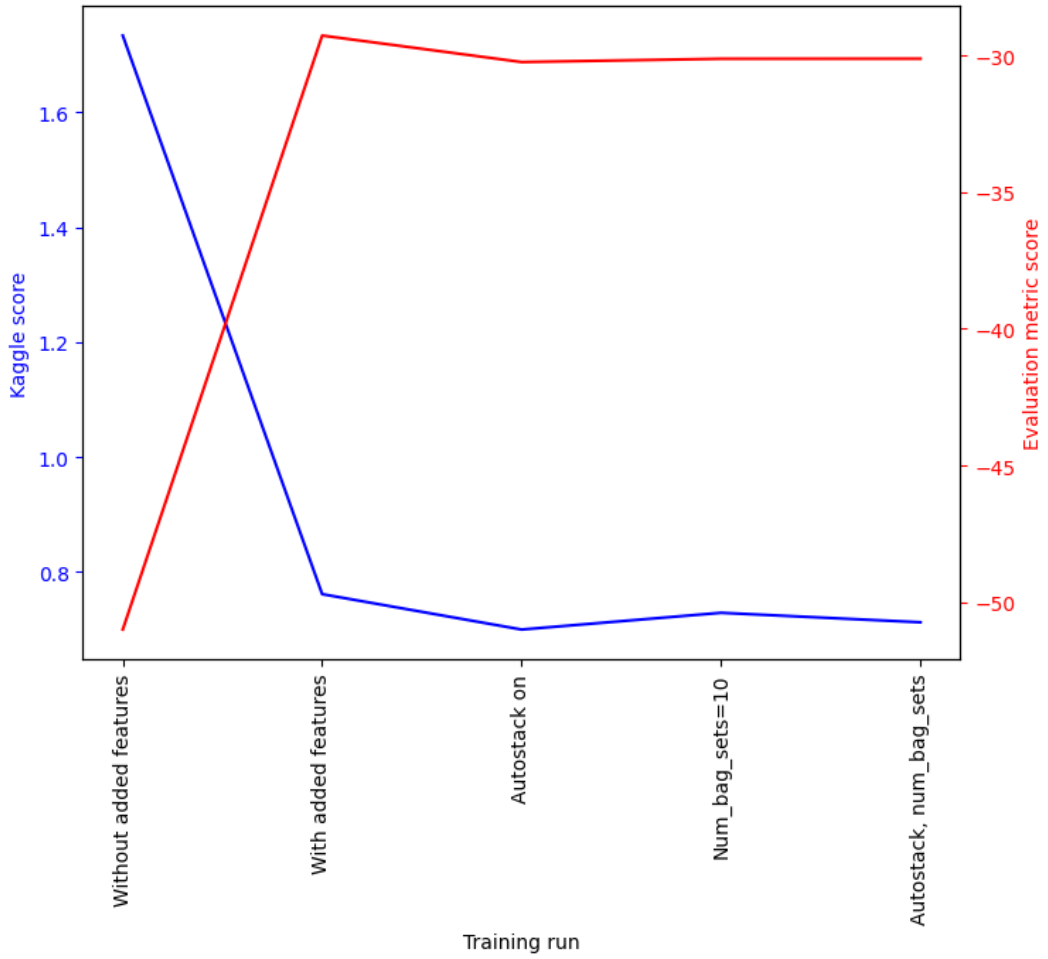
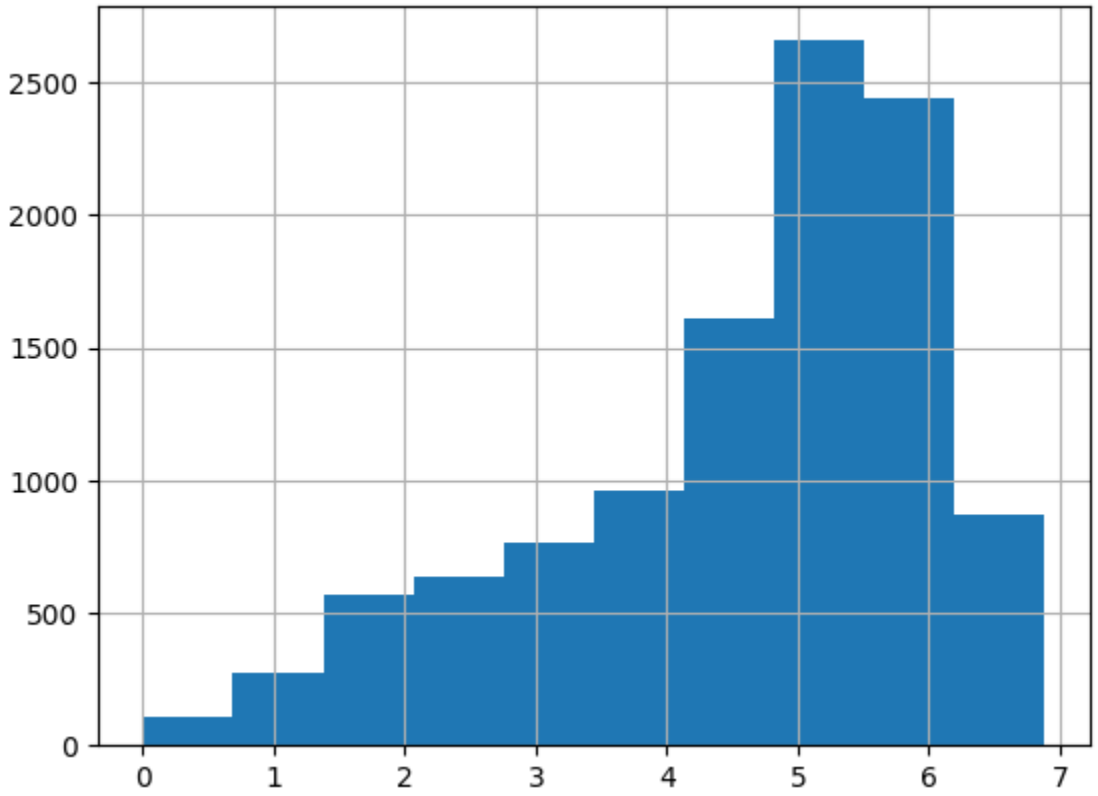


Figure 3. Distribution of count data after performing natural log transform.



Summary

Adding the day, hour, and month features improved the performance of the models. However, the models did not improve much when tuning the autostack and num_bag_sets. This led me to consider transforming the 'count' label as the histogram(Figure 1) indicated the data was highly skewed. After performing the natural log transform (Figure 3), the kaggle score improved by 18% and model evaluation metric improved by 99%.

`auto_stack`	Whether AutoGluon should automatically utilize bagging and multi-layer stack ensembling to boost predictive accuracy. Set `auto_stack` to `"True"` if you are willing to tolerate longer training times in order to maximize predictive accuracy. This automatically sets the `num_bag_folds` and `num_stack_levels` arguments based on dataset properties. Valid values: string, `"True"` or `"False"`. Default value: `"False"`.

Chnage auto_stack to True

`num_bag_sets`	Number of repeats of kfold bagging to perform (values must be greater than or equal to 1). The total number of models trained during bagging is equal to `num_bag_folds` * `num_bag_sets`. This parameter defaults to one if `time_limit` is not specified. This parameters is disabled if `num_bag_folds` is not specified. Values greater than one result in superior predictive performance, especially on smaller problems and with stacking enabled. Valid values: integer, range: [`1`, `20`]. Default value: `1`.