Report: Predict Bike Sharing Demand with AutoGluon Solution

Prosperity Oguama

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?

All negative values had to be set to zero. Kaggle will reject submissions with negative values because the total number of bikes rented per time can only be greater than or equal to zero.

What was the top ranked model that performed?

The top ranked model was WeightedEnsemble_L3. It had a validation score of -53.130615.

Exploratory data analysis and feature creation

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

Histogram:

The *temp*, *atemp,* and *humidity* features are normally distributed. This indicates that the values near the mean occur more frequently than those far from the mean and are consistent with most climate-related data.

The dependent variable (*count*) has a right (positive) skewed distribution. This could indicate higher rentals at the beginning of the year.

The *windspeed* feature is also positively skewed.

There were more working days than holidays recorded in the dataset.

Heatmap:

temp and *atemp* have an equally weak positive correlation with the target variable (*count*).

humidity has a weak negative correlation with the target variable.

The *hour* feature has a moderate positive (and the highest) correlation with the target variable. It is, therefore, a good additional feature for predicting *count*.

Time Series:

The time series plot of count vs hour buttresses the findings from the heatmap plot. There is a non-linear increase in the number of bikes rented as the hour increases. The peak time of bike rentals is around 4pm.

The time series plot of count vs month indicates that more bikes were rented in September than in any other month of the year.

Additional features:

I extracted the hours from the *datetime* column and added them as a new column (*hour*) before training the second model (add_features).

Before training the third model (hpo), I extracted the months from the *datetime* column and added them as a new column (*month*)

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

The model's performance improved by 62.59% after adding the *hour* feature.

This is because the *hour* feature has a moderate positive correlation with the target variable. This means that more bikes are likely to be rented later in the day than earlier.

Hyper parameter tuning

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

First hyperparameter tuning (hpo): Involved doubling the training time and removing the NeuralNetFastAI model because of its low performance in the previous training models.

The model had better kaggle and top model scores than the initial training model, but a poorer kaggle score than the model with the added *hour* feature. Its Kaggle score was 0.75975.

Increasing the training time beyond a certain threshold has little effect on the performance of a machine learning model and could lead to overfitting.

Second hyperparameter tuning (hpo1): Involved selecting only three models - TabularNeuralNetMxnetModel, LightGBM, and RandomForest - with their default hyperparameter settings. This model had the best kaggle score of all the models trained (0.48923).

The top-performing model from the first three training runs (WeightedEnsemble_L3) also took the longest time to train. Decreasing the number of models that made up the ensemble in the last training run significantly reduced the training time. It, however, lowered the top model score.

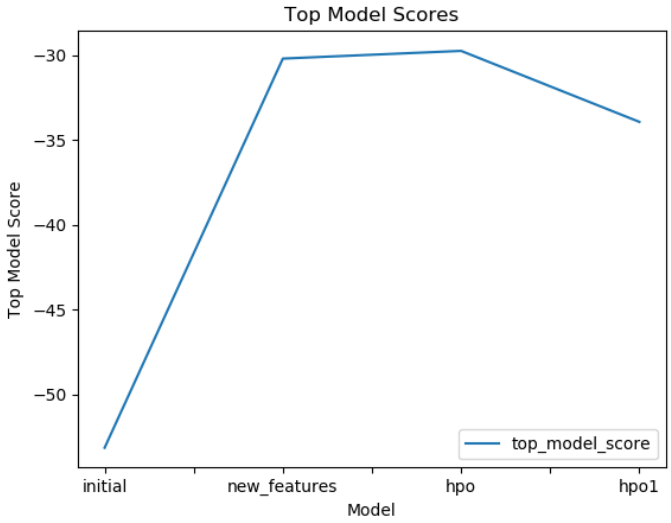
If you were given more time with this dataset, where do you think you would spend more time?

I would spend more time engineering the dataset's features.

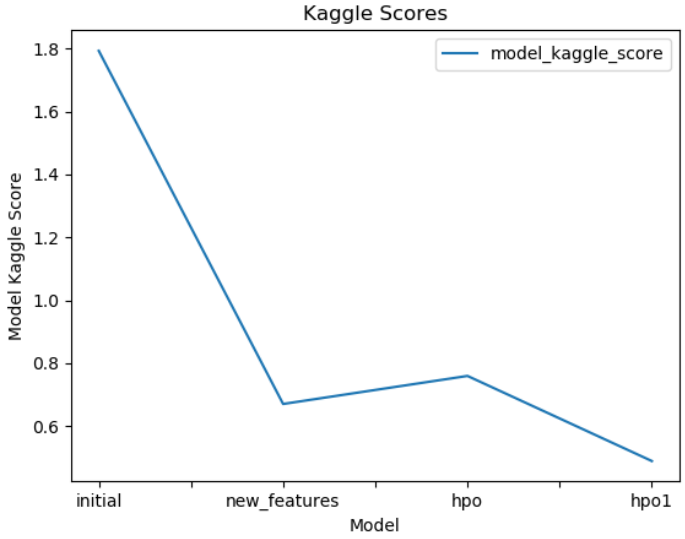
Create a table with the models you ran, the hyperparameters modified, and the kaggle score.

S/N	model	hyperparameters	time_limit	excluded_model_type	score
1	Initial	N/A	600	N/A	1.79693
2	add_features	N/A	600	NeuralNetFastAI	0.67222
3	hpo	N/A	1200	N/A	0.75975
4	hpo1	NN, GBM, RF	600	N/A	0.48923

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.



Summary

- Selecting three models (TabularNeuralNetMxnetModel, LightGBM, and RandomForest) with their default hyperparameter settings significantly improved the kaggle score.
- The top-performing model from the first three training runs (WeightedEnsemble_L3) also took the longest time to train. Decreasing the number of models that made up the ensemble in the last training run significantly reduced the training time. It, however, lowered the top model score.
- Adding the hour feature significantly improved the model's scores because of its correlation with the target variable.
- Adding the *month* feature had no significant effect on the model's scores because it had little correlation with the target variable.
- Doubling the training time (from 600 seconds to 1200 seconds) did not have a significant effect on the top model and kaggle scores.