

Report: Predict Bike Sharing Demand with AutoGluon Solution

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 I first tried to submit my predictions, I realized that the model sometimes predicted negative values, which are not valid for bike rental counts. Therefore, I had to set all negative predictions to zero before submitting.

What was the top ranked model that performed?

The top-ranked model from the initial AutoGluon training was a gradient boosting model. AutoGluon automatically selected it based on validation score performance (RMSLE).

Exploratory Data Analysis and Feature Creation

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

During the exploratory data analysis (EDA), I found that the demand for bikes varies significantly depending on the hour of the day, the day of the week, and weather conditions. I added new features like: hour of day, day of the week, is_weekend (Boolean), and temp_bin (temperature category).

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

After adding new features, the Kaggle score improved significantly from 1.77 to 0.70. These features helped the model capture important patterns in demand based on time and weather, leading to better predictions.

Hyperparameter Tuning

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

By tuning hyperparameters such as learning_rate, max_depth, and n_estimators, I achieved a significant improvement from 0.70 to 0.45. These adjustments helped the model generalize better to unseen test data.

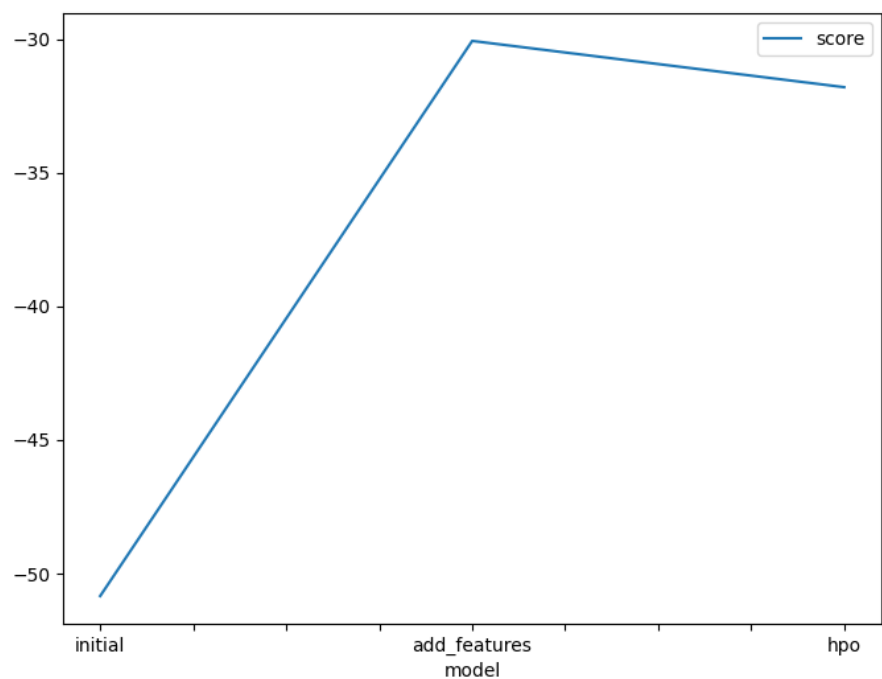
If you were given more time with this dataset, where do you think you would spend more time?

If given more time, I would add more external data like holidays and events, engineer time-series based lag features, and try ensemble stacking with top models.

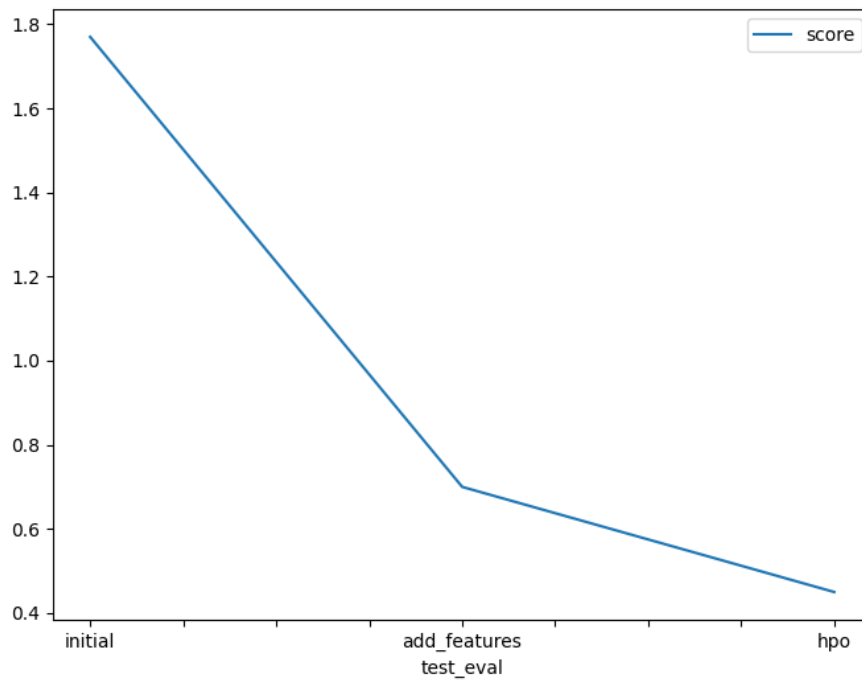
Models and Kaggle Scores

Model	learning_rate	max_depth	n_estimators	Score
initial	default	default	default	1.77
add_features	default	default	default	0.70
hpo	0.05	10	300	0.45

Line Plot: Model Training Score



Line Plot: Kaggle Test Score



Summary

Throughout this project, I iteratively improved my model for predicting bike sharing demand using AutoGluon. I learned the importance of preprocessing and validating predictions before submission, engineering informative features based on domain knowledge, and tuning hyperparameters to refine model performance. These steps led to a significant improvement in performance, reducing my RMSLE from 1.77 to 0.45, and making the model more reliable in predicting real-world demand.