Report: Predict Bike Sharing Demand with AutoGluon Solution

BISOLA IBIWOYE

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?

kaggle does not accept a negative score, so i ensure to turn all scores to zero before submission.

What was the top ranked model that performed?

The initial training done in Autogluon have the highest test score of 1.80161 from Kaggle. The best model was WeightedEnsemble_L3.

Exploratory data analysis and feature creation

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

Exploratory analysis revealed the "datetime" feature values squished together. The first feature engineering done was split "datetime" into "day", "month", "year" and the "datetime" column was dropped. The histogram performed for EDA showed "Season", "holiday", "workingday", "weather" features are more of bar graphs than histograms, this show that these features are categorical variables with "holiday" and "workingday" showing binary classifications where values are in between 0 and 1. For more feature engineering, "season" and "weather" features were made categorical.

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

The first model had the best score on kaggle. The score after the feature engineering was "0.46423". This show the model did not improve.

Hyper parameter tuning

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

The model did better after training it with the first model "GBM" including hyperparameter and hyperparameter_tune_kwargs arguments. The kaggle score for the model was "0.4837". The second model that the data was trained on is "XGB" with the kaggle score of "0.49923" and the last model the data was trained on was "KNN" and it had the highest score in the hyperparameter tuning section with a kaggle score of "1.09996".

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

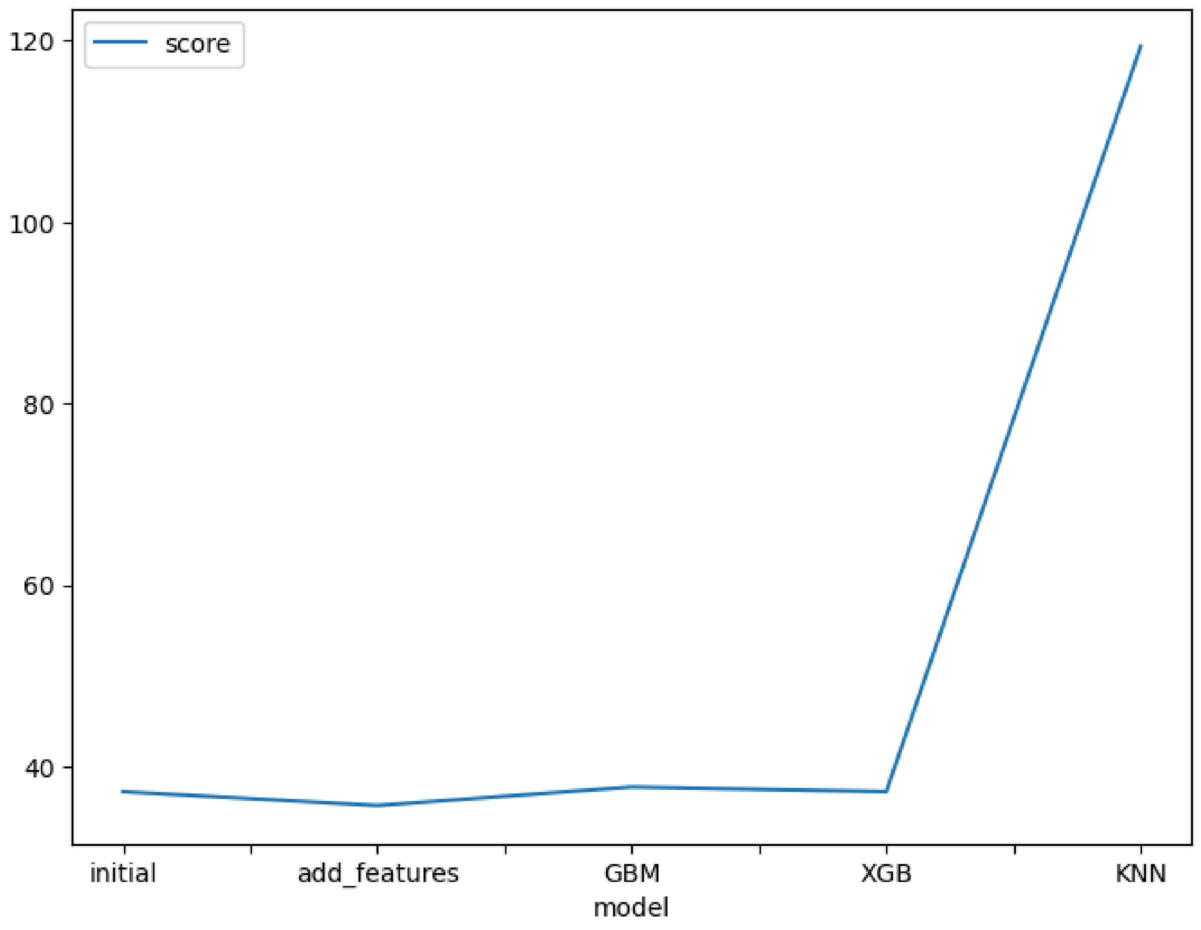
I will spend more time tuning the "KNN" model as it is the bestperformingg model out of the three models.

In [1]:
```
### a table with the models ran, the hyperparameters modified, and the kaggl

import pandas as pd
pd.DataFrame({
    "model": ["initial", "add_features", "GBM", "XGB", "KNN"],
    "Time Limit": [600, 600, 600, 600, 600],
    "Architecture Used": ["Default", "Default", "GBM", "XGB", "KNN"],
    "Training Data": ["Standard", "New Features Added", "GBM Hyperparameters
                      "XGB Hyperparameters Added", "KNN Hyperparameters Adde
    "score": [1.80161, 0.46423, 0.4837, 0.49923, 1.09996]
})
```

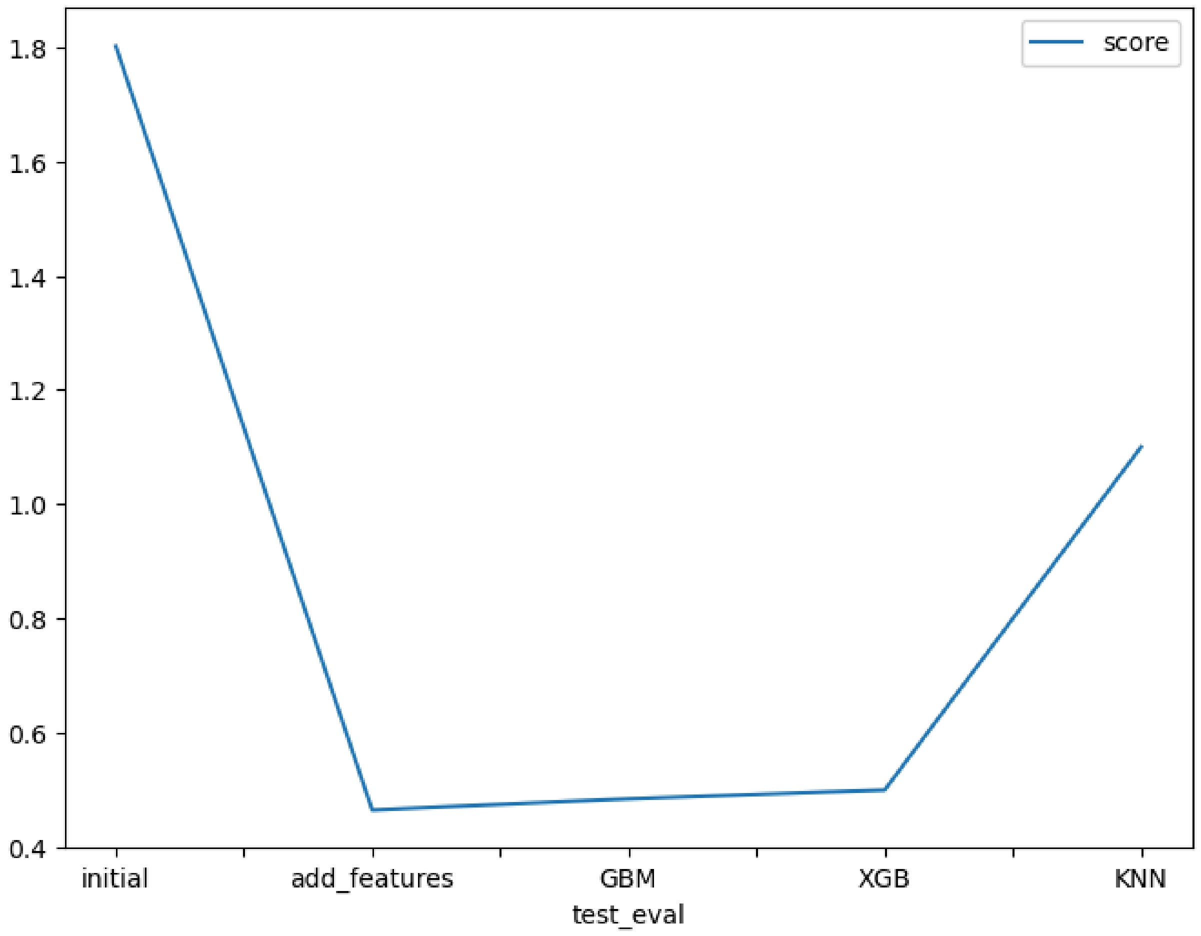
Out[1]:

	model	Time Limit	Architecture Used	Training Data	score
0	initial	600	Default	Standard	1.80161
1	add_features	600	Default	New Features Added	0.46423
2	GBM	600	GBM	GBM Hyperparameters Added	0.48370
3	XGB	600	XGB	XGB Hyperparameters Added	0.49923
4	KNN	600	KNN	KNN Hyperparameters Added	1.09996

In [2]:
```
# Taking the top model score from each training run and creating a line plot

fig = pd.DataFrame(
    {
        "model": ["initial", "add_features", "GBM", "XGB", "KNN"],
        "score": [37.2043112, 35.686201, 37.719401, 37.204312, 119.365601]
    }
).plot(x="model", y="score", figsize=(8, 6)).get_figure()
```



```
In [3]: # Take the 3 kaggle scores and creating a line plot to show improvement
        fig = pd.DataFrame(
            {
                "test_eval": ["initial", "add_features", "GBM", "XGB", "KNN"],
                "score": [1.80161, 0.46423, 0.4837, 0.49923, 1.09996]
            }
        ).plot(x="test_eval", y="score", figsize=(8, 6)).get_figure()
```



Summary

Autogluon made machine learning process easier. The leaderboard() method made it possible to view the performance of each model AutoGluon has trained. The initial training had the best score, the feature engineering improved the dataset quality and the scores showed how different models did with hyperparameter and hyperparameter_tune_kwargs arguments. Autogluon is an easy way to find the best performing model(s) using the default configuration by increasing the time limit and tuning hyperparameter to allow it to test more configurations.