# Report: Predict Bike Sharing Demand with AutoGluon Solution

Javier Chapto

## 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 submitted my results, I realized that most scores in the leaderboard were higher than what I initially obtained. This made me think how there was a lot of room for improvement with feature creation and hyperparameter modification. Before submitting my results, all the predictions that were negative values had to be changed to 0.

### What was the top ranked model that performed?

The top ranked model was WeightedEnsemble_L3, with a score of -53.116201 considering root mean squared error.

## Exploratory data analysis and feature creation

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

During exploratory data analysis, I realized that dates could be fed to the model as different columns. Because we used parse_dates=["datetime"] when reading the CSV, we could create year, month, day and hours columns in the following way:

train['year'] = train['datetime'].dt.year train['month'] = train['datetime'].dt.month train['day'] = train['datetime'].dt.day train['hour'] = train['datetime'].dt.hour

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

Performance improved from 1.78635 to 0.65961 after adding new features. Mainly, because it can measure better seasonality.

## Hyper parameter tuning

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

After trying different hyper parameters, performance improved from 0.65961 to 0.48355.

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

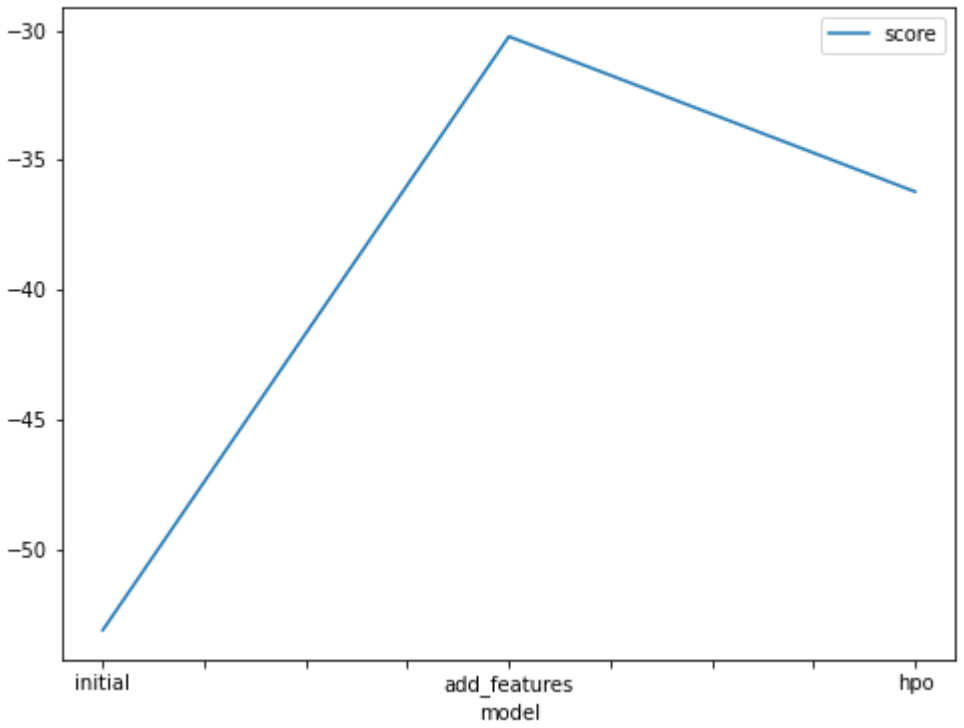
I would spend more time with feature creation and hyperparameter tuning, because both proved to be efficient strategies to improve performance.

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

| | model | number of epochs | learning rate | activation function | score |
|---|---|---|---|---|---|
| 0 | initial | default | default | default | 1.39373 |
| 1 | add_features | default | default | default | 0.46870 |
| 2 | hpo | 10 | ag.space.Real(1e-4, 1e-2, default=5e-4, log=True) | ag.space.Categorical('relu', 'softrelu', 'tanh') | 0.49696 |

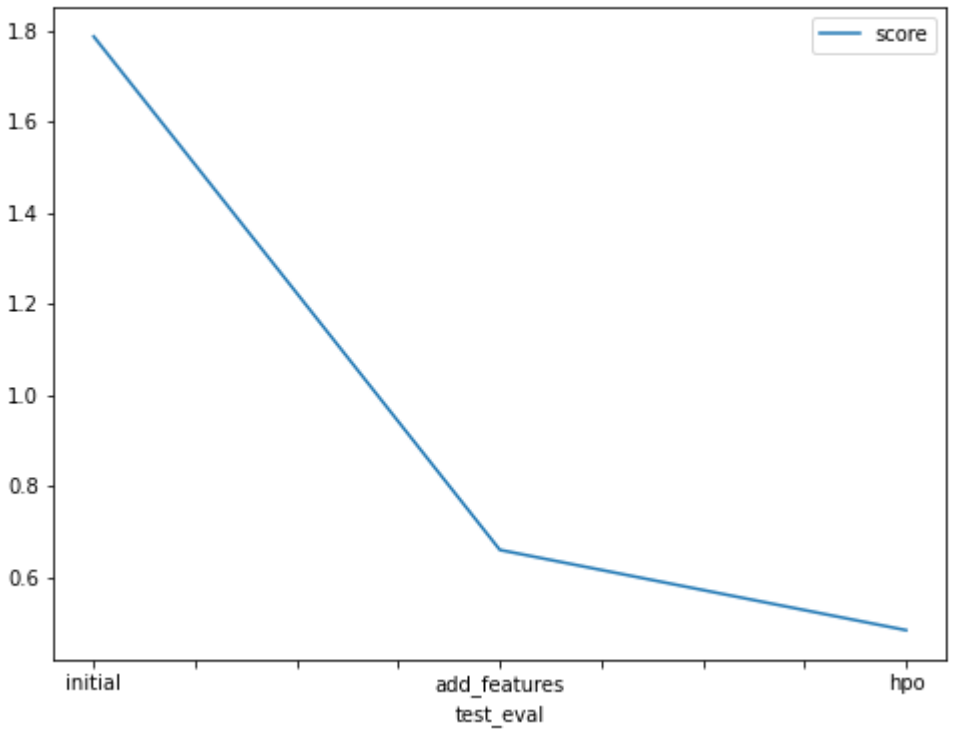
## Create a line plot showing the top model score for the three (or more) training runs during the project.

Figure 1:



Create a line plot showing the top kaggle score for the three (or more) prediction submissions during the project.

Figure 2:




## Summary

It can be observed how there was a significant improvement in the training results after creating new features (Figure 1). However, training results didn't improve after hyperparameter tuning (Figure 1). On the other hand, test results improved after creating new features and hyper parameter tuning (Figure 2).