Report: Predict Bike Sharing Demand with AutoGluon Solution
NAME Shadman Ansari

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 the model was initially run without performing feature engineering and hyperparameter tuning, I obtained the following score and models:
Top 3 models

WeightedEnsemble_L3	139.5201
ExtraTreeMSE_BAG_L2	-140.098
CatBoost_BAG_L2	-140.934

and I got Kaggle private score of: 1.32787

What was the top ranked model that performed?
WeightedEnsemble_L3 was the top ranked model with the score of -139.5201

Exploratory data analysis and feature creation
What did the exploratory analysis find and how did you add additional features?
I applied some of the basic exploratory data analysis examination by creating heatmap, using pandas methods like describe, info, shape methods to understand the data, to find out any missing value or duplications of data

I created additional features like hours, day, year using to_datetime method and using .dt functions

I also converted season and weather as categories, since initially data was int, and sklearn library was considering this variable as integer

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

Model reduced the error, and I experienced an improvement in the accuracy of predictions, and improvement in kaggle score

Below are the top three models with their scores:

WeightedEnsemble_L3	-1.9538
ExtraTreeMSE_BAG_L2	-2.0525
CatBoost_BAG_L2	-2.0898

I also got Kaggle score: 1.34676, which is a significant improvement

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

I added a few hyperparameters using

```
nn_options = {
    'dropout_prob': ag.space.Real(0.0, 0.5, default=0.1),  # dropout
probability
}

gbm_options = {
    'num_boost_round': 100,  # number of boosting rounds
    'num_leaves': ag.space.Int(lower=26, upper=66, default=36),  #
number of leaves in trees
}

hyperparameters = {  # hyperparameters of each model type
                   'GBM': gbm_options,
                   'NN': nn_options,
                  }

num_trials = 3  # try at most 3 different hyperparameter
configurations for each type of model
search_strategy = 'auto'  # tune hyperparameters using Bayesian
optimization routine with a local scheduler

hyperparameter_tune_kwargs = {
    'num_trials': num_trials,
    'scheduler' : 'local',
    'searcher': search_strategy,
}
```

I got the following top 3 models with their scores:

WeightedEnsemble_L3	-36.1319
ExtraTreeMSE_BAG_L2	-36.2494
CatBoost_BAG_L2/L1	-36.3445

and the kaggle score is: 0.51280 which is almost similar to the second approach (adding features)

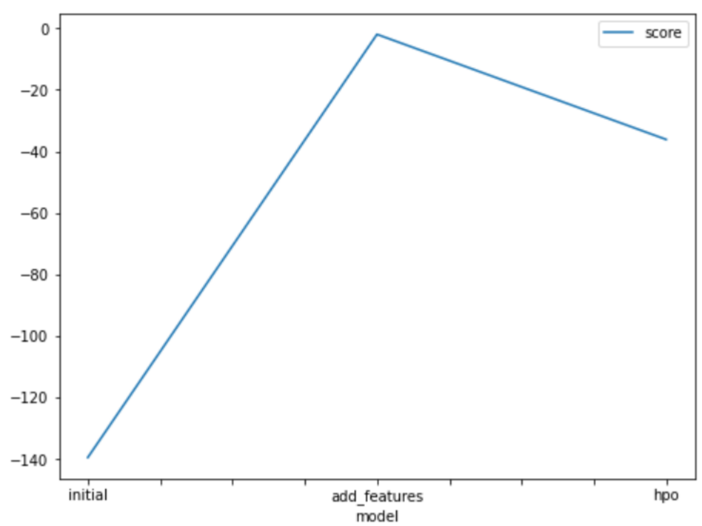
If you were given more time with this dataset, where do you think you would spend more time?
TODO: I would have spend more time on hyperparameter tuning, optamization and also would have tried individual models.

I would have also used Data wrangler pipeline in AWS Sagemaker in order to explore more possibility of features

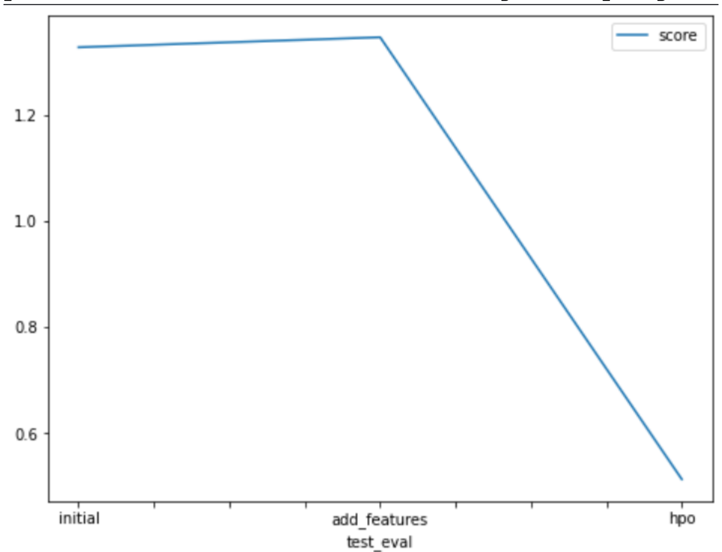
Create a table with the models you ran, the hyperparameters modified, and the Kaggle score.

	model	hpo1	hpo2	hpo3	score
0	initial	default_vals	default_vals	default_vals	1.32787
1	add_features	default_vals	default_vals	default_vals	1.34676
2	hpo	GBM: num_leaves: lower=26, upper=66	NN: dropout_prob: 0.0, 0.5	GBM: num_boost_round: 100	0.51280

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



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



Summary:

We can see that the model is improving by performing feature engineering and hyperparameter tuning.
In the project, we also learnt about Data wrangler and autogluon - which simplified the fitting of model and gives a holistic view of different models