Report: Predict Bike Sharing Demand with AutoGluon Solution

OBAFEMI OLUWADOLAPO SUCCESS

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?

I had to change the dtype of the datetime column to a dtype **of datetime64[ns]** column using the *.to_datetime()*. Nothing much happened when I tried to submit the predictions. I checked for negative values in the results and there was none.

What was the top ranked model that performed?

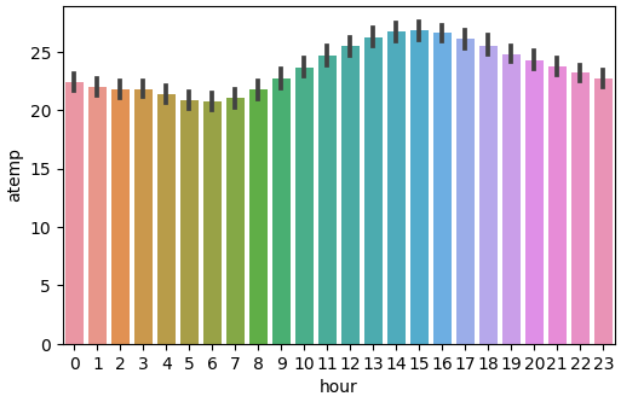
The top ranked model was the **KNeighborsUnif_BAG_L1** with a score validation of **-101.546199** which was the lowest out of the Models that trained in 10 minutes. The prediction time was also very low making it a very fast model for making the right predictions with a time of **0.055712.**

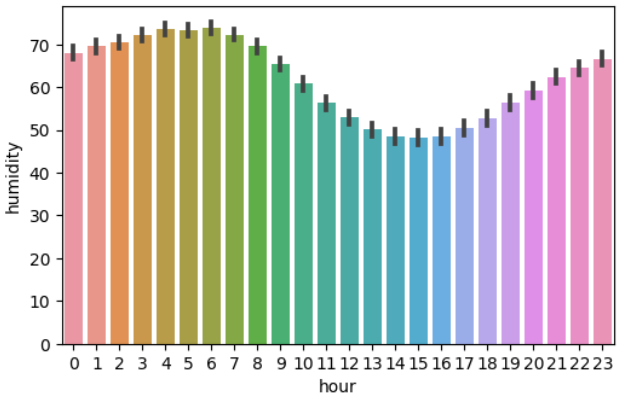
Exploratory data analysis and feature creation

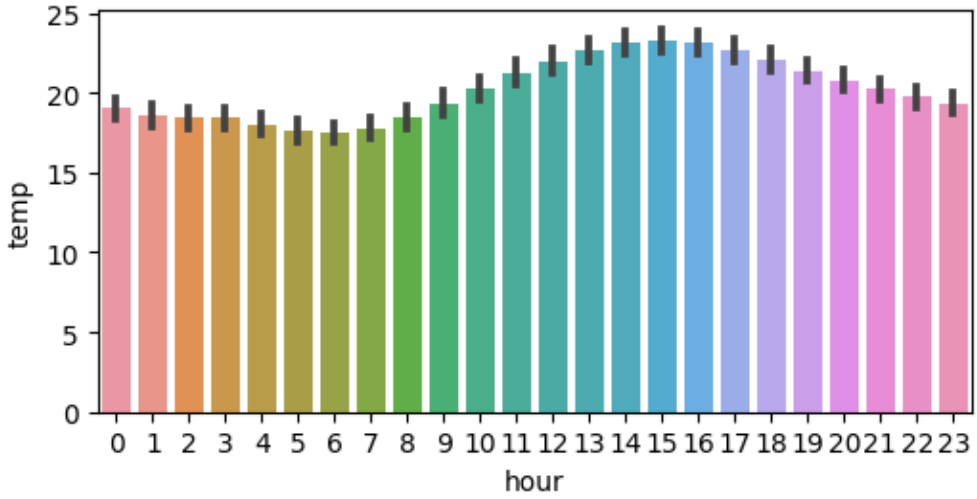
What did the exploratory analysis find and how did you add additional features?

It was noticed that the Count column is left skewed using the.hist() function. The hour column was plotted against each characteristic. As categorical columns, the season and weather columns could not be plotted. The following were discovered:

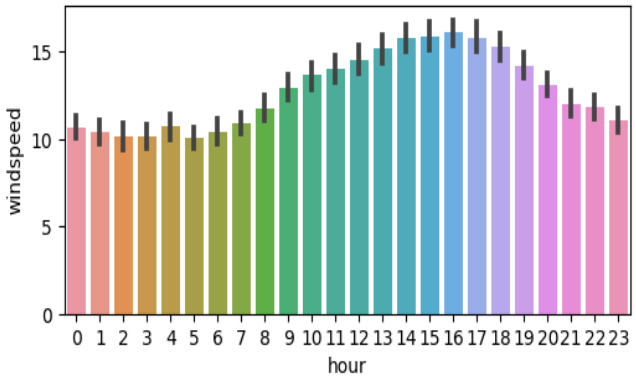
- From 2:00 p.m. to 4:00 p.m., the day's highest temperature and absolute temperature are reached.
- The early morning hours between 6 and 7 in the morning are also when the humidity is at its highest.

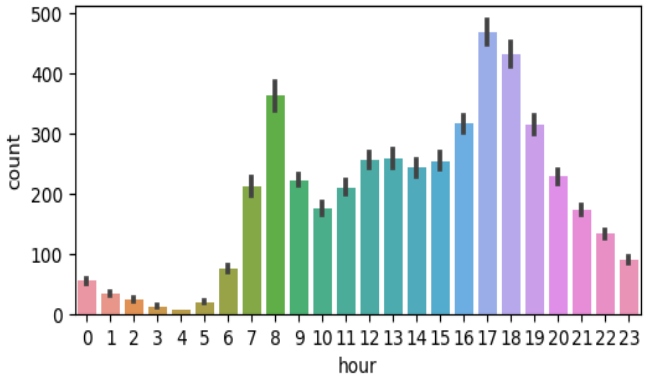




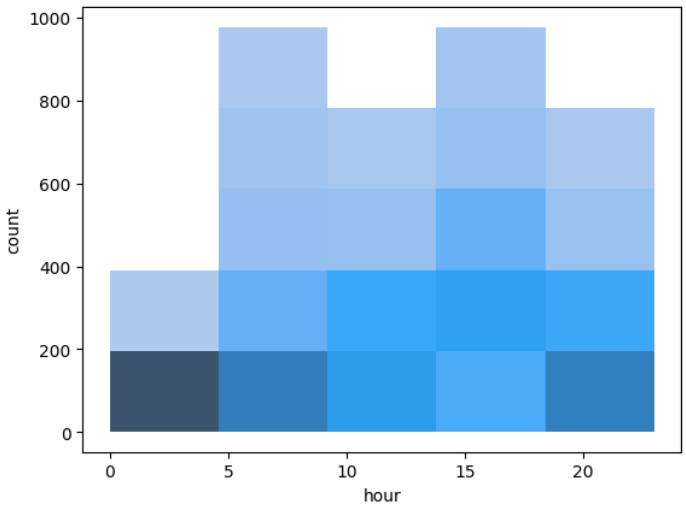


- Around 4 o'clock in the afternoon is also when the wind speed is highest.
- The second-highest surge in bike demand, as seen in the Count against Hour graph, occurs in the morning, when demand is at its lowest. This demonstrates the health-consciousness of some, who exercise first thing in the morning.The highest peak is at roughly 5:00 p.m.



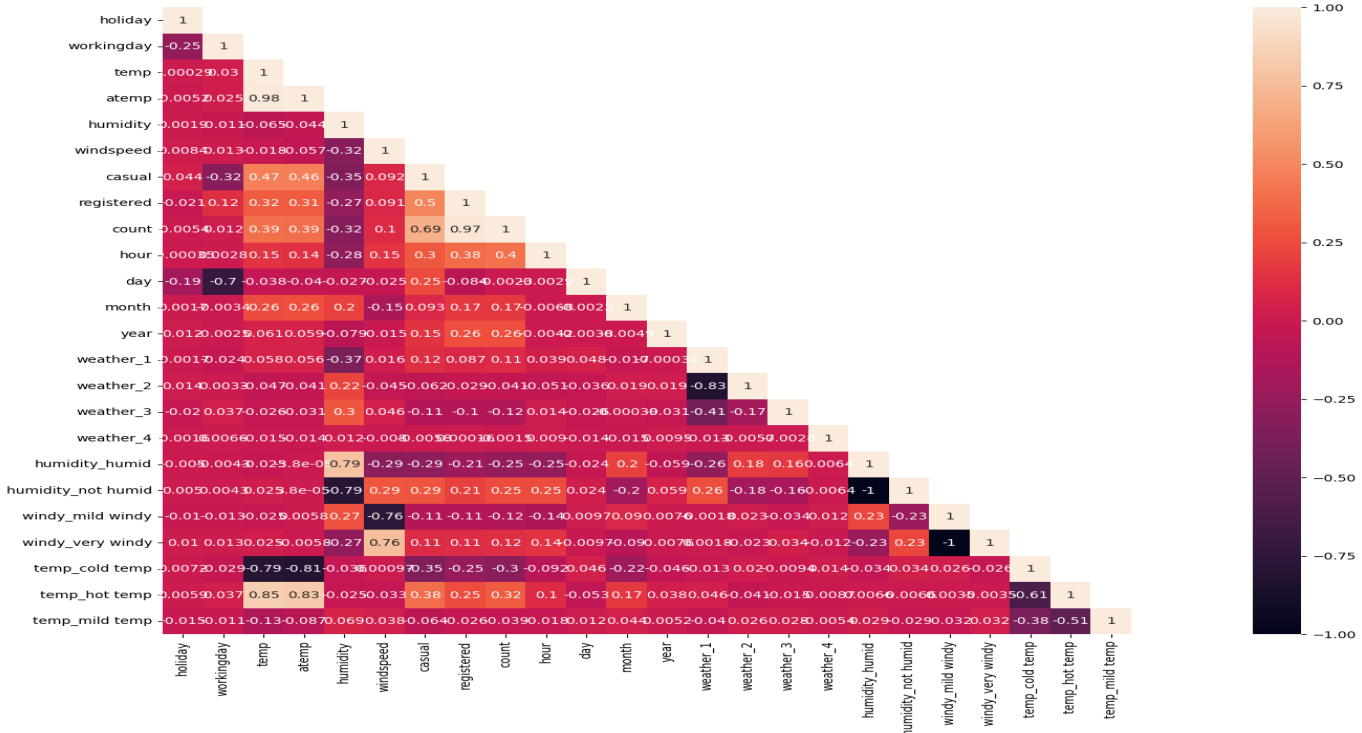


In order to update the categorical datas ["season" and "weather"] using the.get_dummies() function, I developed a new feature utilizing feature engineering. This used integers of 0s and 1s to categorize the season and weather into different groups. Below is the heatmap of count against the time of the day.



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

Following the addition of the new function, it performed way better than when feature engineering was not applied. From a kaggle score of 1.80162 to a lower score of 0.46618 (which shows that the model performed better. I added more features after the hyperparameter tuning was done from the humidity, temperature and windspeed table. Without the hyperparameter tuning after this additional features were added, a kaggle score of 0.56602 was gotten.
Below is the picture of the heatmap after more features were added.



Hyper parameter tuning

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

After each hyperparameter tweaking, they improved greatly. However, applying the hyperparameter **"XGBoost"** caused the model to perform the poorest, with the reason unknownsince XGBoost is known to perform well Although the performance was not as bad as the initial eprfromance since the Kaggle score was **0.64240.**. However, the remaining hyperparameters improved the model's performance. The **GBM** was the best method so far in terms of prediction witha kaggle score of **0.53619**. The model performed well with **“NN”** with a kaggle score of **0.59439**.

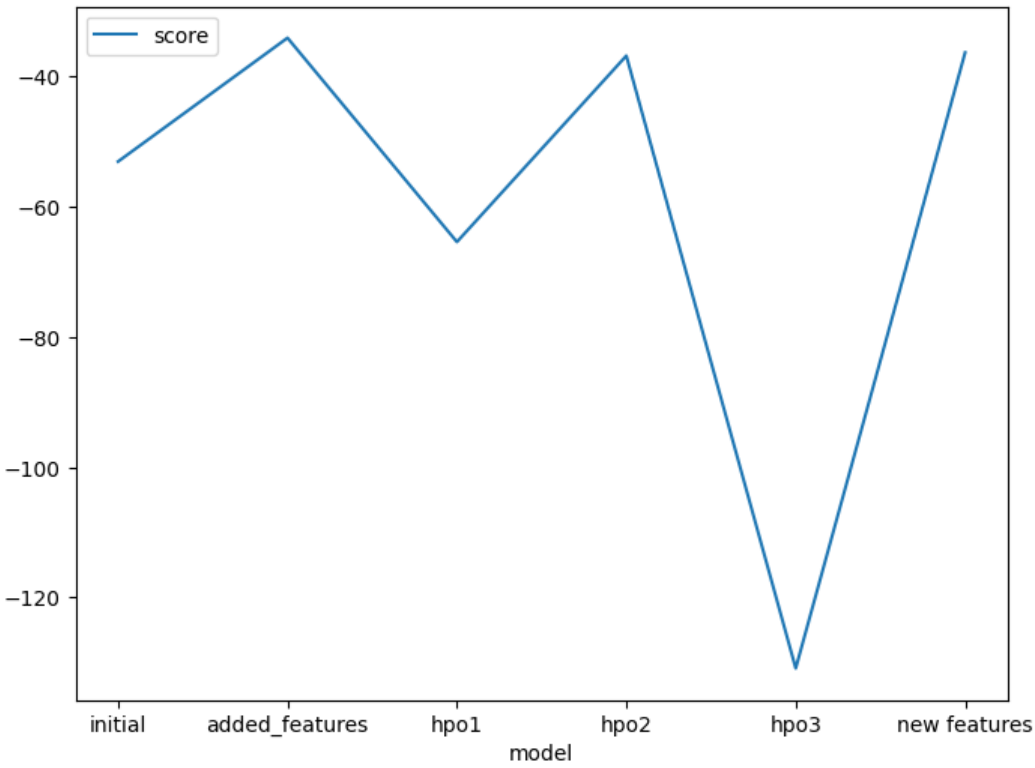
If you were given more time with this dataset, where do you think you would spend more time?

Enhancing the model, particularly using XGBoost, KNN, RF models and machine learning models. Without using AutoGluon, I will conduct additional exploratory data analysis and test out other machine learning models like the RandomForest() and SVM Models to see how each one interprets the input data.

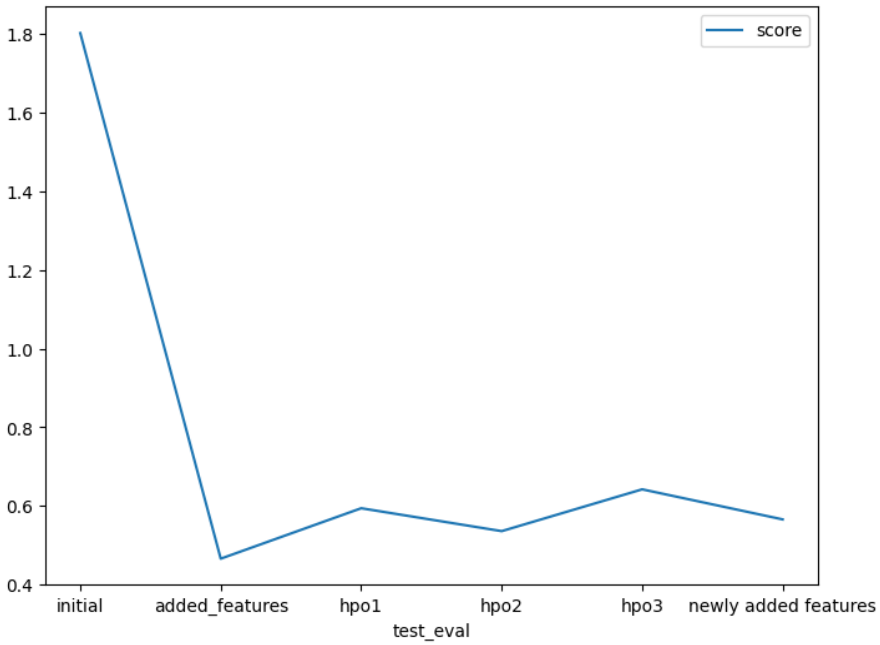
Create a table with the models you ran, the hyperparameters modified, and the kaggle score.

model	hpo1	hpo2	hpo3	add more features
initial	-53.080362	-53.080362	-53.080362	-53.080362
add_features	-34.108591	-34.108591	-34.108591	-34.108591
hpo_model_score	-65.420156	-36.850820	-130.861534	-36.348944
kaggle_score	0.594390	0.536190	0.642400	0.566020

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

To begin with, I had to run an EDA on the data to determine its current status. On the train data, the Autogluon model was applied, and predictions were created using.predict(). It received a kaggle score of roughly 1.8 after submission. Season and weather were classified, and feature engineering was then used to make better forecasts using the.getdummies() function, which was successful because a kaggle score of 0.446 was obtained.

A kaggle score was 0.59 after hyperparameter tuning using the hyperparameter "NN" and its hyperparameter kwargs. With kaggle scores of 0.54 and 0.64, respectively, a second "GBM" and third "XGB" were completed.By categorizing, using feature engineering, utilizing the Autogluon model, and using the provided standard parameters, more features were created, and an Akaggle score of 0.56 was obtained.

Before I got all these scores that seemed good, I had problems with exploratory data analysis and projections, which resulted in scores as low as 2.71 and similar low numbers. I was able to improve numerous reviews after receiving a number of them.

I want to experiment with several models, such as the SVM, RF, and others. I'll speed up submission by reducing the column to which I applied feature engineering.