Bikeshare Mini Project

Abstract: The aim of this project is to use the variables: *dteday*, *season*, *yr*, *mnth*, *hr*, etc. from the datafile to build a model that can help the user to predict the total count of rental bikes. To be clarified, 'hourly.csv' and 'daily.csv', are both used for this project. Two datasets are analyzed separately in order to compare with each other. This project includes Exploring data analysis, Data wrangling, and Model Building. For each dataset, the Linear regression model and Deep Neural Network model are both explored for this problem, and in the end, two models will be compared for their overall performance.

Dataset Selection

'hourly.csv' and 'daily.csv', are used for this project. The reason why performing data analysis on two datasets is that I want to compare the two and see which dataset gives a better result on the model building. Each dataset has its pros and cons. For 'daily.csv', the information is more precise and concentrate. However, there are only 731 total data which might interpret less information when doing the model building. The information in the 'hourly.csv' is mostly extracted from 'daily.csv', which is more completed and detailed compared to the information in 'daily.csv'. In addition, 'hourly.csv' contains 17379 total data, which is better in model building. However, since the data is collected too densely in 'hourly.csv', there might be no much difference (the distance) between each sample which could possibly fool the models.

EDA

In this part, data information is explored. Some variables are shown 'int64' but they are actually categorical variables such as *season*, *holiday*, *yr*, *workingday*, and *weathersit*. So, these variables need to be changed to the 'object' type. The other variables such as *mnth* and *weekday*. should be considered as ordinal variables, so I keep these variables as ‘int64’ type. In addition, in order to make further analysis, NAs are also checked for both of the datasets and both datasets contain no NAs.

Data Wrangling

Splitting dataset to train and test set in order to evaluate the model performance once model training is done. Both of the datasets are randomly split to train set and test set with size ratio equal to 4:1. Then, numerical variables are explored to see which numerical variables are highly correlated with *cnt* column. For ‘daily.csv’, the significant numerical variables are *temp* and *atemp*, but since these two variables are highly correlated to each other, only temp will be selected. For ‘hourly.csv’, *hr*, *temp,* and *atemp* are significant to *cnt,* so *hr* and *temp* are kept in this case. Then categorical variables are needed to one-hot coding in order to perform further analysis.

I perform an OLS model to see which variables should be selected out (p_values are larger than 0.5) from the model. For ‘daily.csv’, the final X-variables are *temp, season, holiday, yr*, and *weathersit.* For ‘hourly.csv’, the final X-variables are *temp, hr, season, holiday,* and *yr.*

Since the numerical columns in both of the datasets have already been normalized, the numerical columns are not necessarily normalized again.

Linear Regression Model

Linear Regression Model is performed on each dataset. The reason I chose the Linear Regression Model is that it is simple and common to use when predicting continuous variables. The r-squared on ‘daily.csv’ is 0.81 and the r-squared on ‘hourly.csv’ is 0.36, which indicates that the model built on ‘daily.csv’ can explain 50% more variability of the data than the model built on ‘hourly.csv’. However, if we look at the model performance on the test set, we can see that the RMSE (root mean squared error:149.26) for the model built on ‘hourly.csv’ is much lower than the one built on ‘daily.csv’ (RMSE: 821.26).

DNN Model

DNN model is built for both of the datasets. The reason why a deep neural network is chosen is that DNN can provide deep and complex learning on the dataset and it is interesting to compare how the two models perform for different datasets. By adjusting the model architectures, we can see from the validation vs train plots (Appendix I) that both models are not showing overfit problem. For ‘daily.csv’, the RMSE for the validation is around 922, and for the ‘hourly.csv’, the RMSE for the validation is around 116, which indicates that the model that built from ‘hourly.csv’ gives a better performance than the one that built from ‘daily.csv’.

Report performance on the test set: For ‘daily.csv’, the RMSE is 988.57 for the test set. For ‘hourly.csv’, the RMSE is only 117.78.

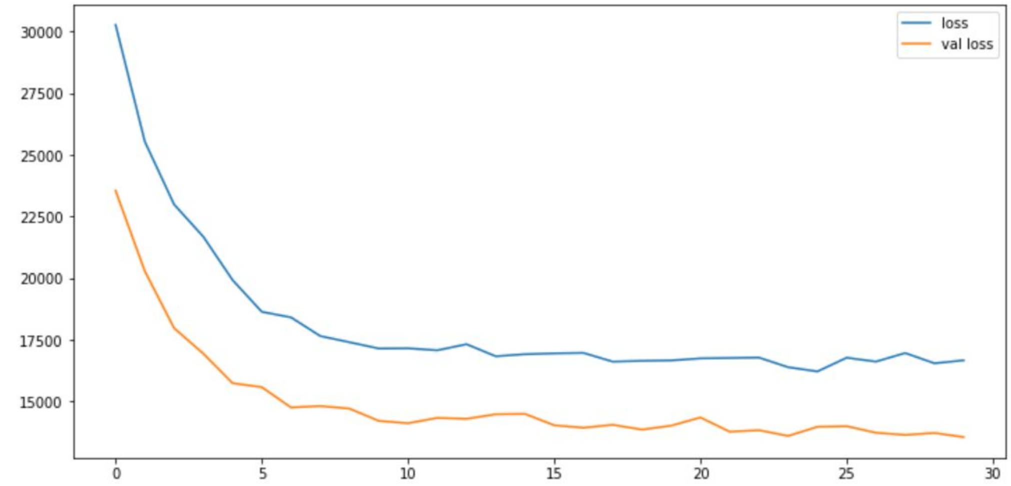
Conclusion

The overall model performance is better on the ‘hourly.csv’ than the one on the ‘daily.csv’. A lower R-squared does not necessarily mean that the model is not good. Linear regression model’s performance is better than the DNN model on ‘daily.csv’ (Appendix II), however, we can observe that the DNN model performs better than the linear regression model on ‘hourly.csv’ (Appendix III). We can conclude when the dataset size is big, a complex model will perform better than a simple model but when the data size is small, a simpler model is more preferred than a complex model.

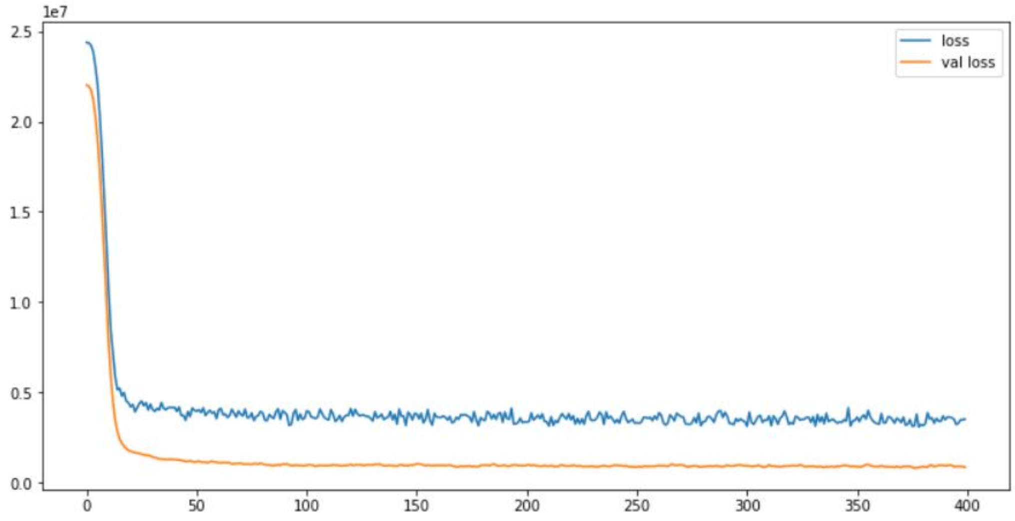
Appendix

Appendix I:

Loss plot for ‘hourly.csv’

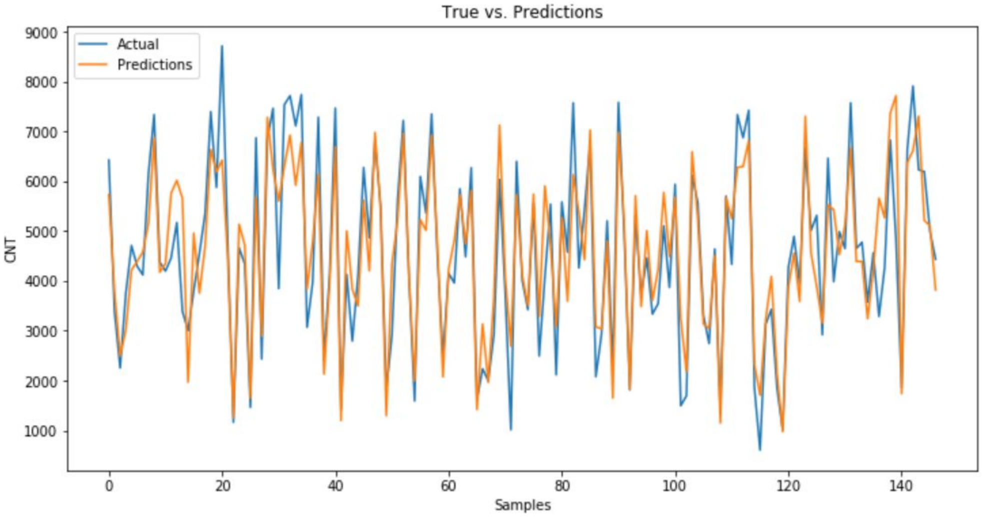


Loss plot for ‘daily.csv’

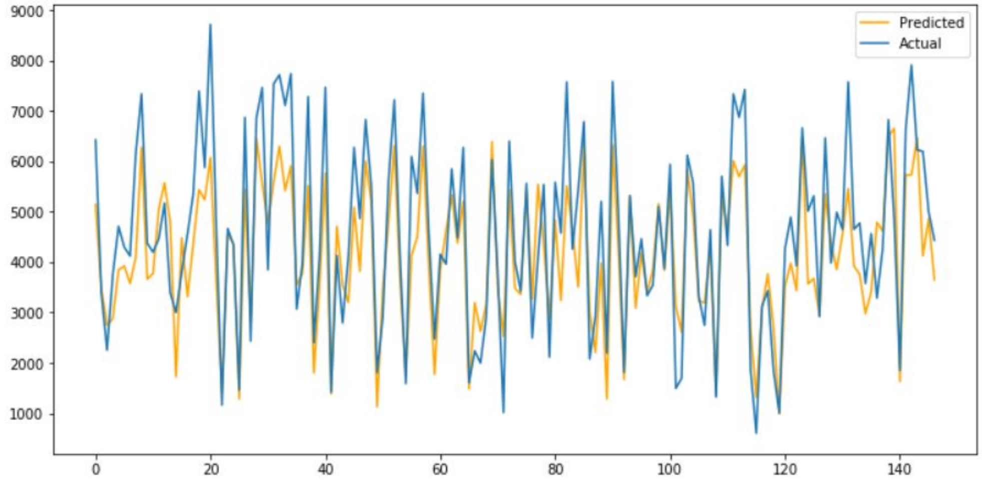


Appendix II:

'daily.csv' linear regression model:

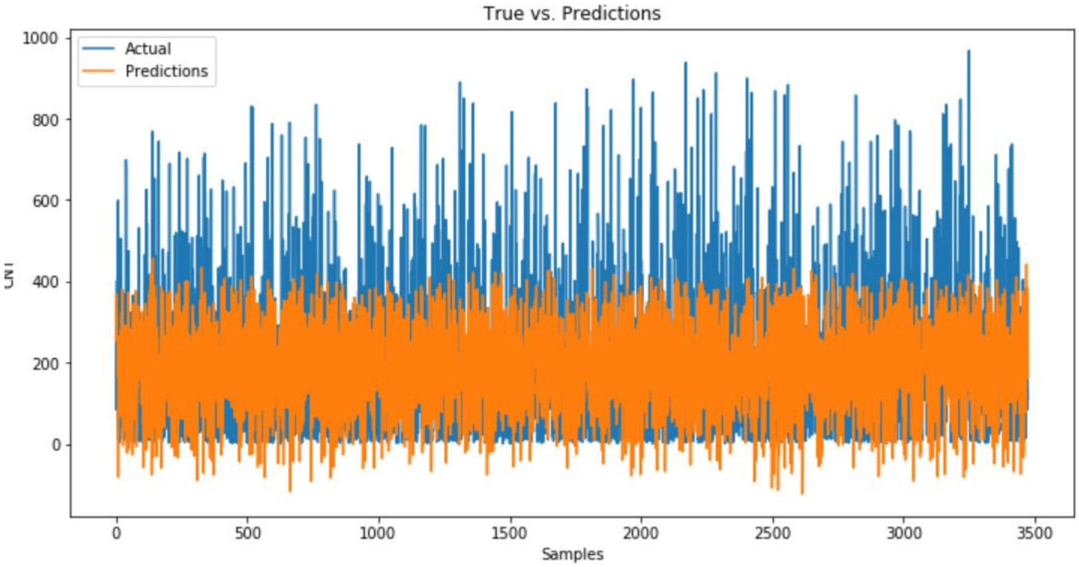


'daily.csv' DNN model:

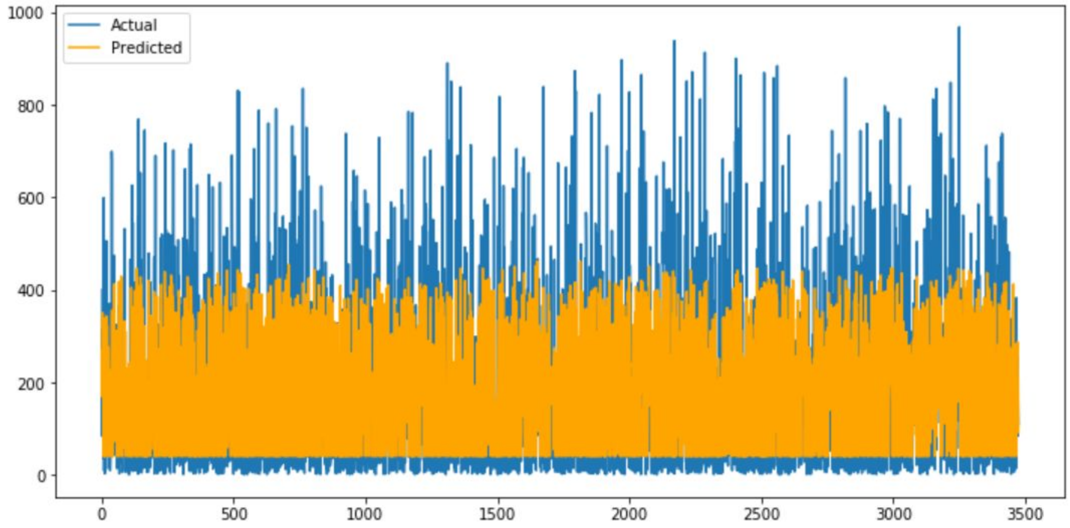


Appendix III:

‘hourly.csv’ linear regression model:



‘hourly.csv’ DNN model:


Actual
Predicted