HOMEWORK 5 – NUMERIC PREDICTION

(Total points 10)

Business Context: Sharing Economy, Bike-Sharing

Assume you work as a data scientist for a bike sharing company (example, Nice Ride Minnesota) and you are asked to analyze the dataset **bike_day.csv** containing information about bike rides, over about 2 years period. You are asked to build a numeric prediction model that can help your company predicting the demand for bike-rides, based on the available attributes.

Data Description

Each observation captures a day of the week and contains the following attributes:

- cnt_bike: the daily count of bike-sharing transactions: in other words, the count of daily bike-rides. This is your Y, the outcome variable you want to predict.
- atemp: perceived temperature for the day, in Celsius
- hum: humidity for the day
- windspeed: wind speed for the day
- temp: temperature, for the day, in Celsius
- holiday: a binary variable that takes value of 0 if the day is not a holiday day; 1 otherwise
- workingday: categorical variable that takes value No if it is not a working day; 1 otherwise

Analysis

Perform the following analysis in R.

Processing and Visualizing Data

a) (0.5) Load the data. Get a summary of the data, report it. Use ggplot to plot a histogram for the distribution of the number of bike-rides. Your plot should have a proper title, proper axis' labels, proper axis' limits. Use the information from the summary and from the histogram to comment on the distribution of the outcome variable, cnt_bike.
b) (0.4) Use the function pairs() to produce a plot of the relationships among count, atemp and hum. Comment on the relationship between count and atemp, and count and hum.
c) (0.2) Split the data into 80% training and 20% testing.

Train a K-NN model.

a) (0.1) First, decide whether you need to standardize the data or not. If yes, explain why; then decide which attributes to include, proceed with the standardization and motivate your choices. If not, explain why.

b) (0.2) Train a k-NN model on the appropriate attributes. Report the algorithm's output and identify the number of k used to trained the model.

c) (0.7) Get the predictions from the k-NN model and use ggplot to create a histogram of the distribution of the predicted bike rides. Also, create a ggplot histogram of the distribution of the true count of bike rides using the testing data. Arrange the plots next to each other. Your graphs should have a proper title, proper axis' labels and proper axis' limits. Pick two different colors for the histograms, so to distinguish them clearly. Compare the two histograms and assess how you think the k-NN model is performing, by simply looking at the plots.

d) (0.7) Compute the prediction error for the k-NN model and create a ggplot histogram for the distribution of the prediction error. Additionally, create a ggplot scatterplot of the prediction error and the true count of bike rides. The plots should have a proper title, proper axis' labels and proper axis limits. Arrange the plots next to each or on top of each other, as preferred. Describe which insights you derive from the two plots. What can we say of the k-NN model performance, by simply looking at these two plots?

e) (0.5) Compute the ME and the RMSE for the k-NN model. Report them and interpret the results.

Train a Regression Tree

f) (0.1) Decide whether you need to standardize the data or not. If yes, explain why; then decide which attributes to include, proceed with the standardization and motivate your choices. If not, explain why.

g) (0.2) Train a regression tree model on the appropriate attributes; report algorithm's output.

a) (0.3) Plot the final tree; describe which attribute-values the algorithm picked to create the tree.

h) (0.7) Get the predictions from the regression tree and use ggplot to create a histogram of the distribution of the predicted bike rides. Compare it to the histogram of the distribution of the true count of bike rides using the testing data (you can use the one produced for point c) in the k-NN analysis). Arrange the plots next to each other. Your graphs should have a proper title, proper axis' labels and proper axis' limits. Pick two different colors for the histograms, so to distinguish them clearly. Compare the two histograms and assess how you think the regression tree model is performing, by simply looking at the plots.

i) (0.7) Compute the prediction error for the regression tree and create a ggplot histogram for the distribution of the prediction error. Additionally, create a ggplot scatterplot of the prediction error and the true count of bike rides. The plots should have a proper title, proper axis' labels and proper axis limits. Arrange the plots next to each or on top of each other, as preferred. Describe which

insights you derive from the two plots. What can we say of the regression tree model performance, by simply looking at these two plots?

b) (0.5) Compute the ME and the RMSE for the regression tree. Report and interpret the results.

Train a Linear Regression

a) (0.1) Decide whether you need to standardize the data or not. If yes, explain why; then decide which attributes to include, proceed with the standardization and motivate your choices. If not, explain why.

b) (0.2) Check and comment on whether the assumption of the outcome variable being somewhat normally distributed is satisfied or not.

c) (0.3) Create a correlation matrix using the attributes used for the prediction. Evaluate whether you should exclude any attribute from the linear regression model due to multicollinearity reasons. Clearly motivate your choices, either way.

d) (0.2) Train a linear regression model on the chosen attributed and summarize the final model. Report the results.

j) (0.7) Get the predictions from the linear regression model and use ggplot to create a histogram of the distribution of the predicted bike rides. Compare it to the histogram of the distribution of the true count of bike rides using the testing data (you can use the one produced for point c) in the k-NN analysis). Arrange the plots next to each other. Your graphs should have a proper title, proper axis' labels and proper axis' limits. Pick two different colors for the histograms, so to distinguish them clearly. Compare the two histograms and assess how you think the regression tree model is performing, by simply looking at the plots.

k) (0.7) Compute the prediction error for the linear regression model and create a ggplot histogram for the distribution of the prediction error. Additionally, create a ggplot scatterplot of the prediction error and the true count of bike rides. The plots should have a proper title, proper axis' labels and proper axis limits. Arrange the plots next to each or on top of each other, as preferred. Describe which insights you derive from the two plots. What can we say of the linear regression model performance, by simply looking at these two plots?

e) (0.5) Compute the ME and the RMSE for the linear regression model. Report and interpret the results.

Produce Insights:

a) (1.5) Produce a table where you report the ME and RMSE for the three models: k-NN, regression tree and linear regression. Also, report the histograms for the distribution of the prediction errors

for k-NN, regression tree and linear regression next to each other. By looking at the errors table and the histograms, which model do you think is performing better and you would suggest your company to implement? Try to be specific when interpreting the histograms of the prediction errors. Why do you think that model(s) may be performing better than the other(s)?