Code Appendix

Taqi

12/10/2021

Appendix

Preprocessing

```
#======================#
#       Wrangling      #
#======================#
# Initial wrangling
wrangle_init <- function(data, omit_NA = TRUE, omit_idx = TRUE){
  # Boolean variables (from int to logical type)
  data$holiday <- as.logical(data$holiday)           # 0 or 1
  data$workingday <- as.logical(data$workingday)     # 0 or 1
  # Other categorical variables (from int to factor type)
  data$season <- as.factor(data$season)              # 1 to 4
  data$yr <- as.factor(data$yr)                      # 0 to 1
  data$mnth <- as.factor(data$mnth)                  # 1 to 12
  data$weekday <- as.factor(data$weekday)            # 0 to 6
  data$weathersit <- as.factor(data$weathersit)      # 1 to 4
  # Re-scale the normalized measurements
  data$temp <- data$temp * 41
  data$atemp <- data$atemp * 50
  data$hum <- data$hum * 100
  data$windspeed <- data$windspeed * 67
  # Change type of Dates (from char to Date type)
  data$dteday <- as.Date(data$dteday)
  # Remove NAs (if prompted) default value is TRUE
  if(omit_NA) { data <- na.omit(data) }
  # Remove instance column (if prompted) default value is TRUE
  if(omit_idx) { data <- data %>% select(-c("instant")) }
  # Observe christmas
  data$holiday[359] <- T; data$holiday[725] <- T
  # Return the wrangled dataset
  return(data)
}
```

```
#==========================#
#      Weekly Averages     #
#==========================#
# Compute the (1-week lagged) weekly averages of a given variable
weekly_avgs <- function(data, var){
  # Compute the averages of the variable by week
  weekly_cnts <-
```

```
    data %>%
    group_by(week) %>%
    summarize(wavg = mean({{ var }}))
  # Lag week by 1
  weekly_cnts$week <- weekly_cnts$week + 1
  # Remove excess weeks
  return(weekly_cnts %>% filter(week <= 53))
}

# Returns a dataset with an added column of weekly averages of a given variable
add_weekly_avg_var <- function(data, var, var_name){
  # Obtain the weekly averages of the desired variable
  var_avgs <- data %>% weekly_avgs({{ var }})
  # Rename the weekly average the desired variable name
  colnames(var_avgs)[2] <- var_name
  # Join the week column in the dataset by the weekly averages in the var_avgs dataframe
  return(data %>% left_join(var_avgs))
}

# Given the bike dataset, returns the dataset with week
# column and weekly averages for the three response variables
add_weekly_averages <- function(data){
  # Add the week variable to the dataset
  data <- data %>% mutate(week = ceiling(1:nrow(data)/7))
  # Add the cnt, reg, and cas weekly averages to the data
  data <- data %>% add_weekly_avg_var(cnt, "wavg_cnt")
  data <- data %>% add_weekly_avg_var(registered, "wavg_reg")
  data <- data %>% add_weekly_avg_var(casual, "wavg_cas")
  return(data)
}

#========================#
#        Subsetting      #
#========================#
# Filter for the 2011 data
in_2011 <- function(data){ return(data[(data$dteday >= "2011-01-01" & data$dteday <= "2011-12-31"),]) }
# Filter for the 2012 data
in_2012 <- function(data){ return(data[(data$dteday >= "2012-01-01" & data$dteday <= "2012-12-31"),]) }
```

Variable Selection

Predictors Selection

Predictors Selection

Response Transformation

Initial Modeling

```
# A helper function that returns a formula in the "lm" syntax
# Takes predictors, a vector of variable name strings as an input
.parseFormula <- function(predictors, response = "cnt"){
  f <- as.formula(
    paste(response,
          paste(predictors, collapse = " + "),
          sep = " ~ "))
  return(f)
}
```

```
lm.tot <- function(varset, train_data = data2011){
  model <- lm(formula = .parseFormula(predictors = varset, response = "cnt"), data = train_data)
  model
}
lm.cas <- function(varset, train_data = data2011){
  model <- lm(formula = .parseFormula(predictors = varset, response = "casual"), data = train_data)
  model
}
lm.reg <- function(varset, train_data = data2011){
  model <- lm(formula = .parseFormula(predictors = varset, response = "registered"), data = train_data)
  model
}
```

Beginning Model

Final Model

Diagonostic Analysis

Validation and Problemshooting

Refined Model

Prediction of the Yearly Growth Ratio

Prediction without the Yearly Growth Ratio