case_study

Shikhar Gupta

9/25/2017

```r
# loading data
data <- read.csv("~/housing.txt")
```

```r
# functions

# NA replacement as factor level
replace_NA <- function(x) {
    y <- addNA(x)
    levels(y) <- c(levels(x), "absent")
    x <- y
}

# missing value counting
pMiss <- function(x) {
    sum(is.na(x))/length(x) * 100
}

# variable remove
varremove <- function(df, varlist) {
    df <- df[, !(colnames(df) %in% varlist)]
    return(df)
}
```

```r
# #missing values across variables for original data
# miss_count_pre <- as.data.frame(apply(data,2,pMiss)>0)
# #visualizing distribution of missing values
# miss_col_data_pre <- data[,apply(data,2,pMiss)>0] aggr_plot
# <- aggr(miss_col_data_pre, col=c('navyblue','red'),
# numbers=TRUE, sortVars=TRUE,
# labels=names(miss_col_data_pre), cex.axis=.7, gap=3,
# ylab=c('Histogram of missing data','Pattern'))

######## Variable treatment and manual
######## selection#####################

# removing variables
data <- varremove(data, c("PoolQC", "MiscFeature", "Alley", "FireplaceQu",
    "Fence", "Id", "Utilities"))
data <- varremove(data, c("GarageYrBlt", "GarageArea", "GarageType",
    "GarageFinish", "GarageQual", "GarageCond"))


# treating Masonry related variables
data$MasVnrArea[which(is.na(data$MasVnrArea))] <- 0
data$MasVnrType[which(is.na(data$MasVnrType))] <- "None"


# treating bsmt related variables
```

```r
data_bsmt <- data %>% dplyr::select(c(BsmtQual, BsmtCond, BsmtExposure,
    BsmtFinType1, BsmtFinType2))

data <- data %>% dplyr::select(-c(BsmtQual, BsmtCond, BsmtExposure,
    BsmtFinType1, BsmtFinType2))
# dataframe with NA replaced in categorical variables where
# NA is not a missing value
data_bsmt <- data_bsmt %>% sapply(replace_NA) %>% data.frame()

data <- cbind(data, data_bsmt)

# converting month sold to factors
data$MoSold <- as.factor(data$MoSold)
```

```r
# Missing value imputation using miss forest
data_imp <- missForest(data)

# error in the imputation
data_imp$OOBerror

# final imputed dataframe
data <- data_imp$ximp
```

```r
# creating data model
x <- model.matrix(SalePrice ~ ., data = data)
y <- data$SalePrice

# creating dataframe
df1 <- as.data.frame(cbind(y, x))


norm_diagnostic <- function(m) {
    ols_rsd_qqplot(m)
    ols_norm_test(m)
}


############### OLS1###################
model1 <- lm(y ~ ., df1)
summary1 <- summary(model1)
coeff1 <- as.data.frame(summary1$coefficients)

###### RESIDUAL DIAGNOSTICS#############

### Normality test########
norm_diagnostic(model1)

##### Error Autocorrelation test######
durbinWatsonTest(model1)

##### Constant variance#######
ols_rvsp_plot(model1)

ncvTest(model1)
```

```r
############# LASSO for variable selection############

set.seed(1)  #for reproducability

# creating grid for lamda
grid.lambda <- 10^seq(10, -2, length = 100)

# cross validation to get best lambda
cv.out <- cv.glmnet(x, y, alpha = 1, lambda = grid.lambda)

best.lambda <- cv.out$lambda.min

plot(cv.out)
abline(v = log(best.lambda), col = "blue", lwd = 2)
# actual lasso model with the best lambda
final.model <- glmnet(x, y, alpha = 1, lambda = best.lambda)

Coef.Lasso <- coef(final.model)

# variables removed
var_remove <- names(Coef.Lasso[which(Coef.Lasso[, 1] == 0), 1])

# new df with reduced variables
df2 <- varremove(df1, var_remove)
```

```r
############# box cox for possible transformation############
bc <- boxcox(y ~ ., data = df2)
bc$x[which(bc$y == max(bc$y))]

df2$y <- log(df2$y)

############# OLS FIT 2############
model2 <- lm(y ~ ., df2)
summary2 <- summary(model2)
coeff2 <- as.data.frame(summary2$coefficients)

###### RESIDUAL DIAGNOSTICS############

### Normality test########
norm_diagnostic(model2)

##### Error Autocorrelation test######
durbinWatsonTest(model2)

##### Constant variance test#######
ols_rvsp_plot(model2)
ncvTest(model2)
```

```r
##### Outlier treatment###### dffits
dffits <- ols_dffits_plot(model2)
dffits$outliers
d.outlier <- dffits$outliers$Observation

# removing outlier observations: 84 observations
```

```r
df3 <- df2[-d.outlier, ]
```

```r
############# OLS FIT3#############
model3 <- lm(y ~ ., df3)
summary3 <- summary(model3)
coeff3 <- as.data.frame(summary3$coefficients)

###### RESIDUAL DIAGNOSTICS#############

### Normality test########
norm_diagnostic(model3)

##### Constant variance test#######
ols_rvsp_plot(model3)
ncvTest(model3)

# 2 variables removed after OLS
df3 <- varremove(df3, c("Condition2PosN", "Exterior2ndWd Shng"))
# rerun the above OLS again after this
```

```r
# backward subset selection for variables
k <- ols_step_backward(model3, prem = 0.05, details = FALSE)
df4 <- varremove(df3, k$removed)
```

```r
df5 <- varremove(df4, c("StreetPave", "LotShapeIR3", "LowQualFinSF",
    "BsmtFinType1Unf", "BsmtFinSF1"))


############# OLS FIT4#############
model4 <- lm(y ~ ., df5)
summary4 <- summary(model4)
coeff4 <- as.data.frame(summary4$coefficients)

###### RESIDUAL DIAGNOSTICS#############

### Normality test########
norm_diagnostic(model4)

##### Constant variance test#######
ols_rvsp_plot(model4)
ncvTest(model4)

####### Forward Selection#######
l <- ols_step_forward(model4)
plot(l)

##### Best subset selection with max of 15 predictors#######
model5 <- regsubsets(y ~ ., df5, nbest = 1, nvmax = 20, method = "exhaustive",
    force.in = NULL, force.out = NULL)

summary.out <- summary(model5)
f <- as.data.frame(summary.out$outmat)
```

```r
df6 <- df5 %>% dplyr::select("y", "BldgTypeDuplex", "BsmtFullBath",
    "BsmtQualEx", "Condition1Norm", "BsmtExposureGd", "FunctionalTyp",
    "Exterior1stBrkFace", "NeighborhoodStoneBr", "NeighborhoodNridgHt",
    "NeighborhoodSomerst", "MSZoningRM", "BldgTypeTwnhs", "LotArea",
    "NeighborhoodCrawfor", "GarageCars", "TotalBsmtSF", "OverallCond",
    "OverallQual", "YearBuilt", "GrLivArea", "FunctionalSev",
    "ScreenPorch", "SaleTypeNew", "KitchenQualTA", "NeighborhoodEdwards",
    "BldgTypeTwnhsE", "FoundationPConc", "Fireplaces", "KitchenQualGd",
    "YearRemodAdd")


############# OLS FIT5#############
model5 <- lm(y ~ ., df6)
summary5 <- summary(model5)
coeff5 <- as.data.frame(summary5$coefficients)
coeff5_interval <- confint(model5)
###### RESIDUAL DIAGNOSTICS#############

### Normality test########
norm_diagnostic(model5)

##### Constant variance test#######
ols_rvsp_plot(model5)
ncvTest(model5)

##### Error correlation test#######
durbinWatsonTest(model5)
```

```r
#### Morty data prediction######
mortydata <- df6[7, 2:length(df6)]
morty_predict <- predict(model5, newdata = mortydata, interval = "prediction",
    level = 0.95)
exp(morty_predict[2])
```