Predicting Housing Price(Machine Learning)

Keith Lee

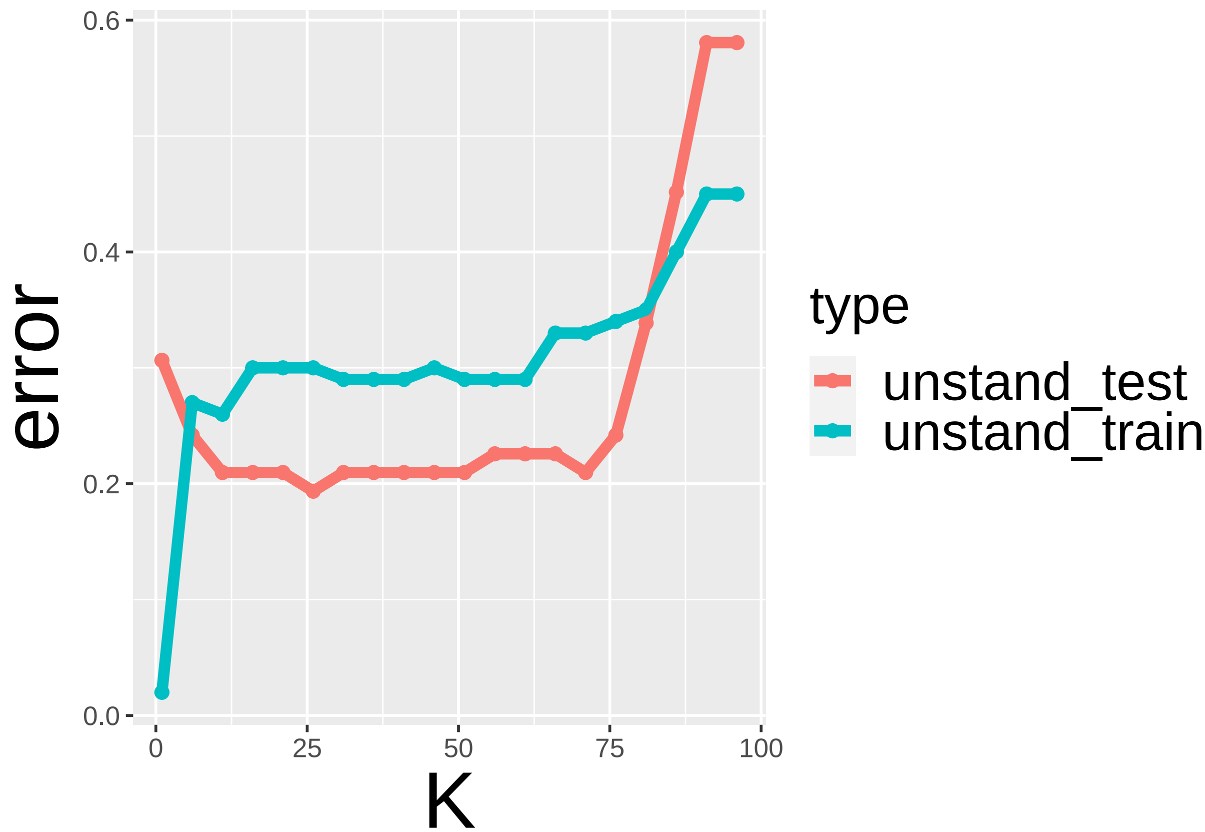
We will be predicting whether the housing price is expensive or not using the `sahp` dataset in the **r02pro** package.

#create dataset

```
library(r02pro)
library(tidyverse)
library(MASS)
my_sahp <- sahp %>%
  na.omit() %>%
  mutate(expensive = sale_price > median(sale_price)) %>%
  dplyr::select(gar_car, liv_area, oa_qual, expensive)
my_sahp$expensive <- as.factor(my_sahp$expensive)
my_sahp_train <- my_sahp[1:100, ]
my_sahp_test <- my_sahp[-(1:100), ]
```

First, I will fit KNN model of `expensive` on variables `gar_car` and `liv_area` with K-nearest number from 1 to 100 with increment of 5.

```
library(caret)
k_seq <- seq(from = 1, to = 100, by = 5)
train_error_seq <- test_error_seq <- NULL
for(k_ind in seq_along(k_seq)){
 k <- k_seq[k_ind]
 fit_knn <- knn3(expensive ~ gar_car + liv_area, data = my_sahp_train, k = k)
 pred_knn <- predict(fit_knn, newdata = my_sahp_train, type = "class")
 train_error_seq[k_ind] <- mean(pred_knn != my_sahp_train$expensive)
 pred_knn <- predict(fit_knn, newdata = my_sahp_test, type = "class")
 test_error_seq[k_ind] <- mean(pred_knn != my_sahp_test$expensive)
}
knn_re <- rbind(data.frame(K = k_seq, error = train_error_seq, type = "unstand_train"),
                data.frame(K = k_seq, error = test_error_seq, type = "unstand_test"))
mytheme <- theme(axis.title = element_text(size = 30),
        axis.text = element_text(size = 10),
        legend.text = element_text(size = 20),
        legend.title = element_text(size = 20))
ggplot(knn_re, mapping = aes(x = K, y = error, color = type)) +
  geom_point(size = 2) +
  geom_line(size = 2) +
  mytheme
```



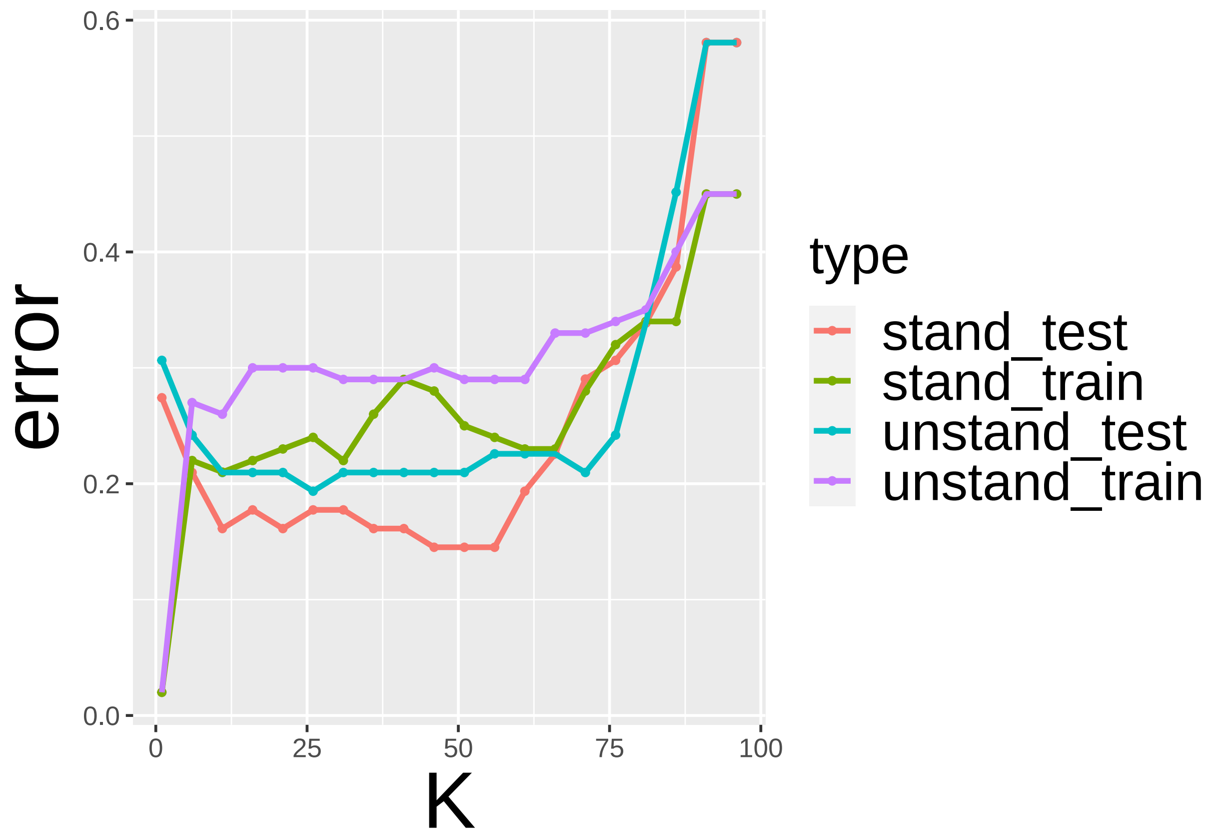
From above graph, we can state that KNN below 80 generally gives better test error. However, if we decide to pick KNN over 80, we have larger test errors.

Next, I will standardize `gar_car` and `liv_area` and repeat the same task from above and visualize the training and test error.

```
#standardization = (x - x(mean)/std(x)
stan_gar_train <- (my_sahp_train$gar_car - mean(my_sahp_train$gar_car, na.rm = T))/sd(my_sahp_train$gar_
stan_liv_train <- (my_sahp_train$liv_area - mean(my_sahp_train$liv_area, na.rm = T))/sd(my_sahp_train$li
stan_gar_test <- (my_sahp_test$gar_car - mean(my_sahp_test$gar_car, na.rm = T))/sd(my_sahp_test$gar_car
stan_liv_test <- (my_sahp_test$liv_area - mean(my_sahp_test$liv_area, na.rm = T))/sd(my_sahp_test$liv_a
new_train <- data.frame(expensive = my_sahp_train$expensive, gar_car = stan_gar_train, liv_area = stan_l
new_test <- data.frame(expensive = my_sahp_test$expensive, gar_car = stan_gar_test, liv_area = stan_liv_
```

```
stan_train_error_seq <- stan_test_error_seq <- NULL
for(k_ind in seq_along(k_seq)){
 k <- k_seq[k_ind]
 stan_fit_knn <- knn3(expensive ~ gar_car + liv_area, data = new_train, k = k)
 stan_pred_knn <- predict(stan_fit_knn, newdata = new_train, type = "class")
 stan_train_error_seq[k_ind] <- mean(stan_pred_knn != new_train$expensive)
 stan_pred_knn1 <- predict(stan_fit_knn, newdata = new_test, type = "class")
 stan_test_error_seq[k_ind] <- mean(stan_pred_knn1 != new_test$expensive)
}
stand_knn_re <- rbind(data.frame(K = k_seq, error = stan_train_error_seq, type = "stand_train"),
                data.frame(K = k_seq, error = stan_test_error_seq, type = "stand_test"))

comb_knn_re <- rbind(knn_re, stand_knn_re)
ggplot(comb_knn_re, mapping = aes(x = K, y = error, color = type)) +
  geom_point(size = 1) +
  geom_line(size = 1) +
  mytheme
```



As we can compare standardized vs. unstandardized, it shows that standardized values have relatively lower errors.

Finally, we will Logistic regression, LDA, QDA, and KNN to see the performance of the models by calculating AUC. Then we will draw ROC curve to show the performance of classification model at all classification thresholds.

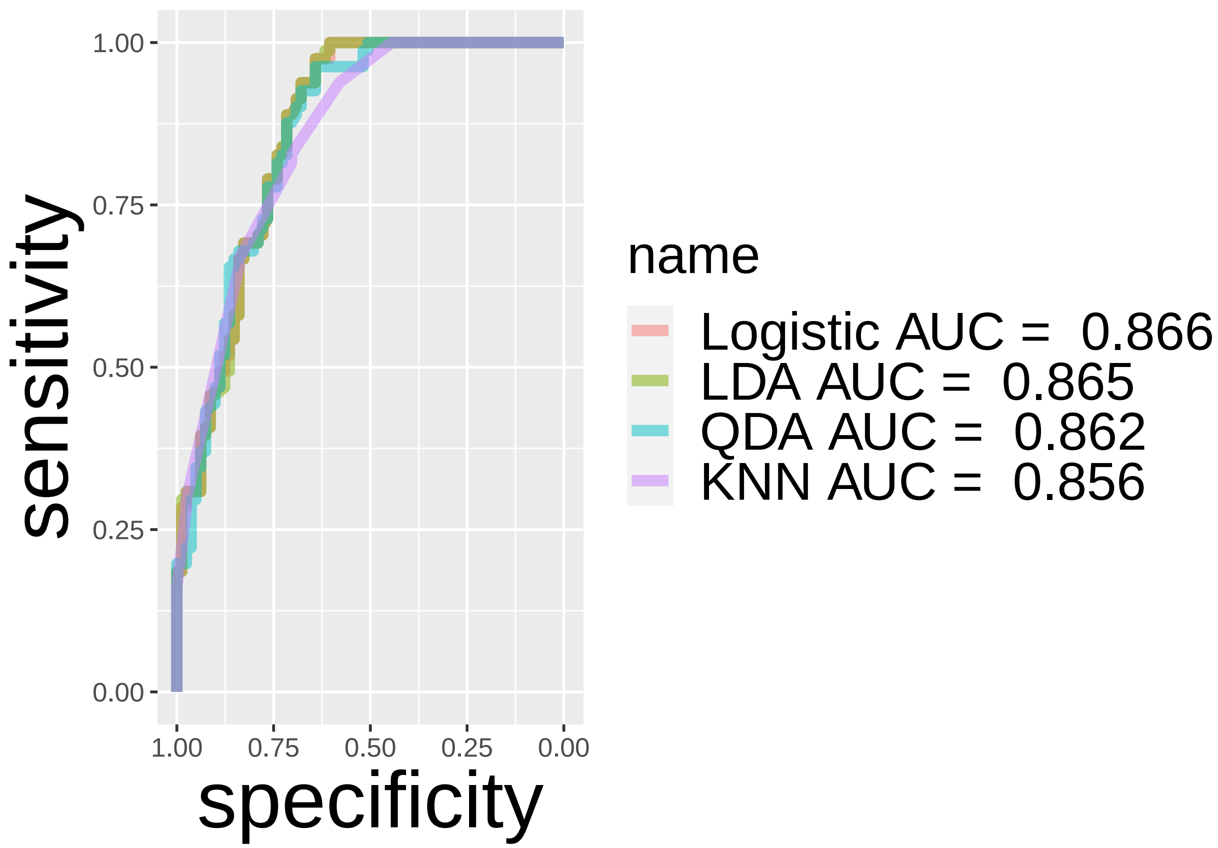
```
library(pROC)
glm <- glm(expensive ~ gar_car + liv_area, data = my_sahp, family = "binomial")
pred <- predict(glm, type = "response")
roc <- roc(my_sahp$expensive, pred)
auc <- auc(roc)

lda <- lda(expensive ~ gar_car + liv_area, data = my_sahp)
pred_1 <- predict(lda)$posterior[, 2]
roc_1 <- roc(my_sahp$expensive, pred_1)
auc_1 <- auc(roc_1)

qda <- qda(expensive ~ gar_car + liv_area, data = my_sahp)
pred_2 <- predict(qda)$posterior[, 2]
roc_2 <- roc(my_sahp$expensive, pred_2)
auc_2 <- auc(roc_2)

knn <- knn3(expensive ~ gar_car + liv_area, data = my_sahp, k = 7,prob = TRUE)
pred_3 <- predict(knn, newdata = my_sahp, type = "prob")
roc_3 <- roc(my_sahp$expensive, pred_3[ ,2])
auc_3 <- auc(roc_3)

roc_4 <- list(Logistic = roc, LDA = roc_1, QDA = roc_2,
                KNN = roc_3)
methods_auc <- paste(c("Logistic", "LDA", "QDA","KNN"),
                     "AUC = ",
                     round(c(auc, auc_1, auc_2, auc_3),3))
ggroc(roc_4, size = 2, alpha = 0.5) +
  scale_color_discrete(labels = methods_auc) +
  mytheme
```



Our graph indicates that predictions are about 86% correct, which is excellent number. We can use this model to tell whether the price is expensive or not when we have a new data set.