ML Assignment 1

Prabh Talwar

2022-10-07

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(rsample)
library(caret)
library(visdat)
```

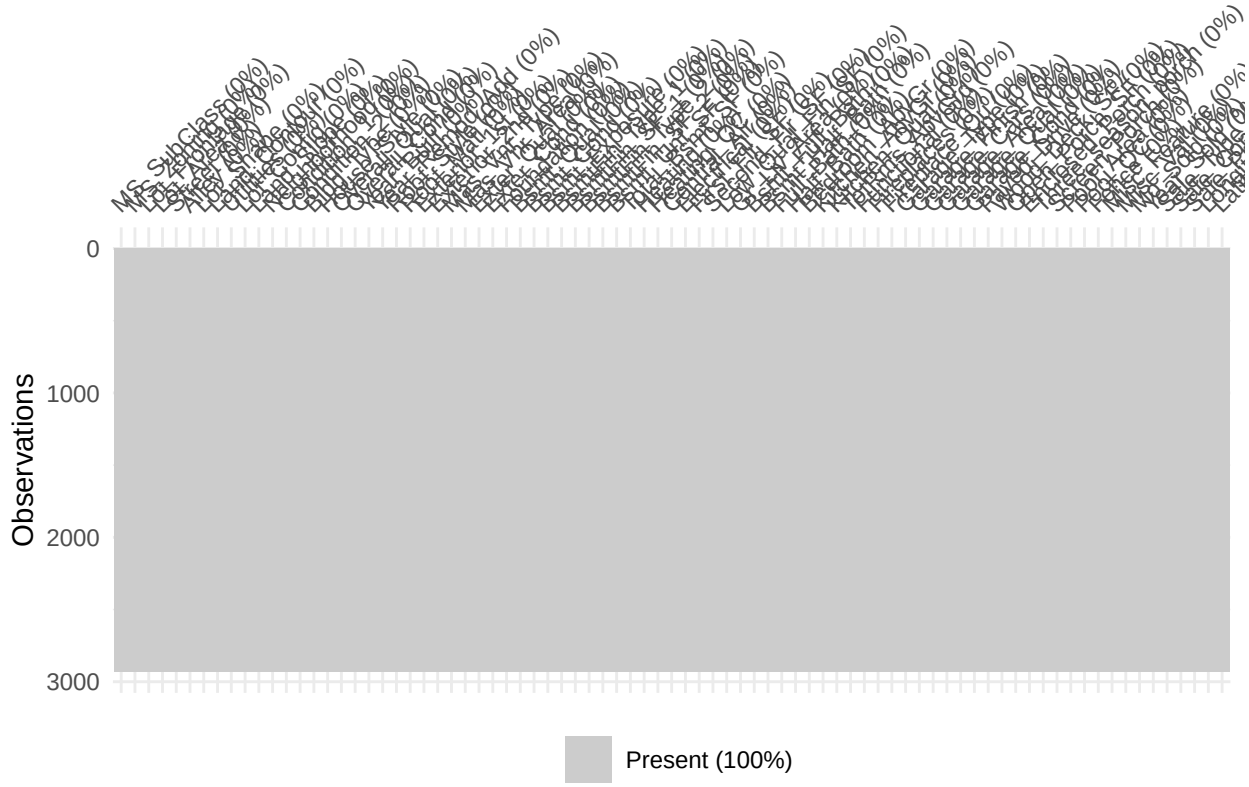
```
# Accessing data
ames <- AmesHousing::make_ames()

dim(ames)
```

```
## [1] 2930   81
```

```
# Visualizing missing data
vis_miss(ames)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



The above dataset has 2930 observations and 81 variables. The data set has both Numerical and Categorical data. While visualizing the data, we came to know that there is no missing values in the data set.

Splitting the dataset

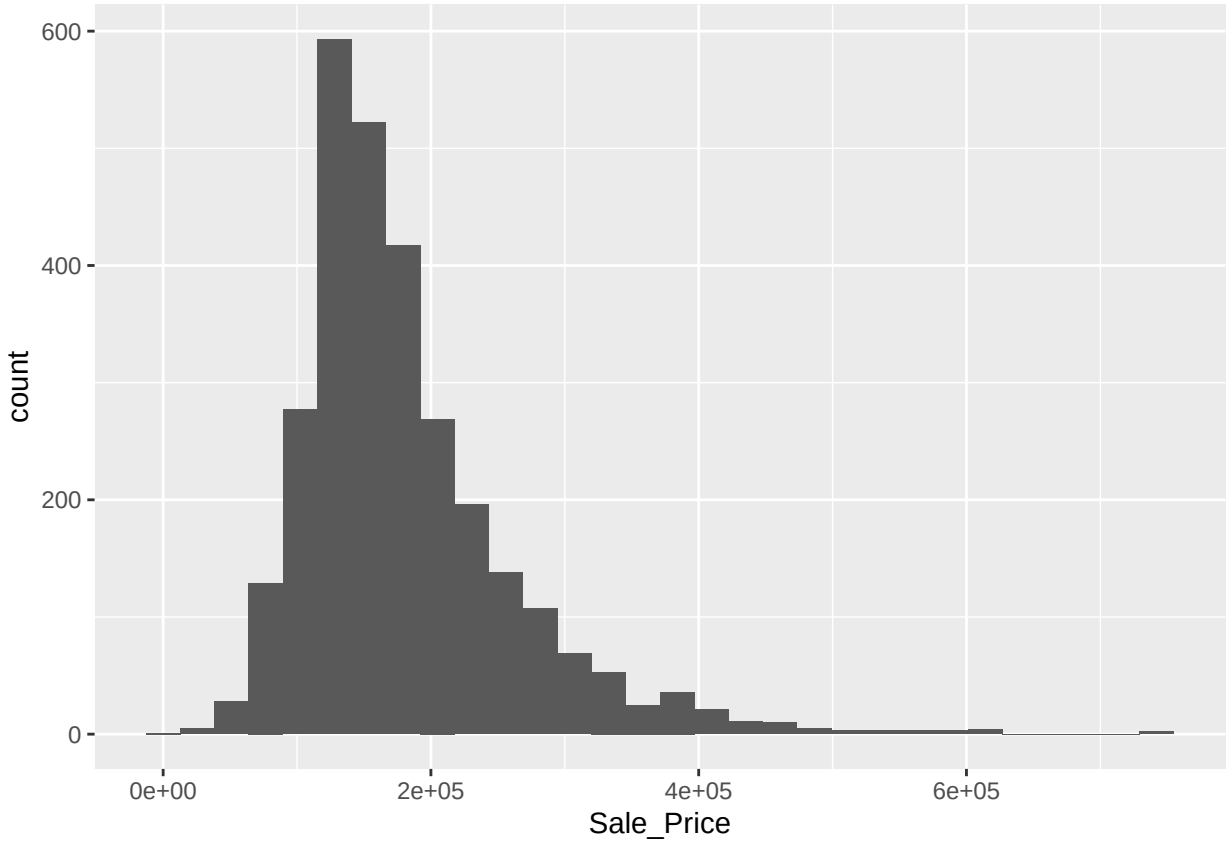
```
set.seed(123) # for reproducibility

split <- initial_split(ames, prop = 0.7, strata = "Sale_Price")
ames_train <- training(split)
ames_test <- testing(split)
```

To build a model that predicts well to our past data, we split our data in training and testing data sets. We are splitting our data into 70% training and 30% testing.We used stratified sampling here so that we can control the unbalanced distribution of the response variable as our response variable is positively skewed. Stratified sampling will ensure that our response variable, i.e; Sale Price is properly distributed in our training and testing data.

```
ggplot(data = ames, mapping = aes(x = Sale_Price)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Specify resampling strategy
cv <- trainControl(
method = "repeatedcv",
number = 10,
repeats = 5
)
```

We will be using K-fold cross validation that will randomly divides our training data set in to k folds. We are using 10 folds that will be repeated 5 times as or dat is not that big, i.e; it is $n < 1000$.

```
# Create grid of hyperparameter values
hyper_grid <- expand.grid(k = seq(2, 25, by = 1))
```

We are tuning parameters to control the complexity of the ML algorithms. we specified the hyperparameter values to 2 to 25.

```
# Tune a knn model using grid search
knn_fit <- train(
Sale_Price ~ .,
data = ames_train,
method = "knn",
trControl = cv,
tuneGrid = hyper_grid,
metric = "RMSE"
)
```

```
knn_fit
```

```
## k-Nearest Neighbors
##
## 2049 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 1844, 1844, 1843, 1844, 1844, 1845, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    2  47206.74  0.6596344  31133.25
##    3  45684.91  0.6773635  30007.71
##    4  44771.97  0.6901038  29276.09
##    5  44064.54  0.7005790  28996.93
##    6  43846.05  0.7045166  28895.48
##    7  43858.13  0.7059401  28883.74
##    8  44181.13  0.7033657  29055.05
##    9  44352.03  0.7028597  29109.00
##   10  44332.46  0.7053884  29129.62
##   11  44282.81  0.7083442  29081.39
##   12  44486.34  0.7075253  29155.41
##   13  44647.15  0.7076206  29256.68
##   14  44790.79  0.7077073  29307.11
##   15  45041.02  0.7063767  29423.86
##   16  45119.37  0.7073844  29484.00
##   17  45264.01  0.7070891  29586.22
##   18  45366.02  0.7072968  29641.92
##   19  45537.84  0.7066304  29766.52
##   20  45746.63  0.7052851  29907.89
##   21  45983.35  0.7031524  30058.55
##   22  46187.92  0.7017539  30192.39
##   23  46406.95  0.7001361  30329.98
##   24  46605.01  0.6986611  30483.94
##   25  46824.70  0.6971044  30617.33
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 6.
```

we trained k-nearest neighbor model using “knn” method with pre-specified resampling procedure trControl, grid search, and loss function “RMSE”.