NYC - K-Nearest Neighbor

Hannah Khuong

Contents

Model summary

K-Nearest Neighbors (KNN) is considered a lazy learner as it does not generate a generalized pattern of the training data. It analyzed training data only when testing data is presented. KNN algorithm calculates Euclidean distance to determine closeness of datapoints. Based on this distance, k number of observations is considered one another's 'neighbor'. In Regression KNN, a predicted value is the mean value of the top k neighbors. KNN is a simple and fast algorithm, but susceptible to local patterns. The smaller the k, the more complex the model and poses a risk of overfitting.

As the algorithm uses distance to determine neighbors, categorical variables were transformed into dummy variables. Scaling is important for KNN algorithm because we want all features to be equally important in distance calculation, regardless of the variable scale. Therefore, all features were scaled for this algorithm.

The best KNN model used Product Type with grouping method 1. The best k (chose based on lowest RMSE) was 20. The relatively small k indicates this is a quite complex model and have a tendency for overfitting. KNN seemed to overfit observation in Days Prior further from Stay Date (day 20-60) as out-sample MASE was higher than in-sample MASE at around 10 percentage-point.

Pattern-wise, KNN generated a close approximation trend of 3 regrouped Product Types. KNN also successfully picked up the declining curve of lead day 7 to Stay Date which was prominent in the Middle Level Cancellation group. The higher mean variation seen in predicted trend in Low Cancellation Group might be due to the complexity of a low k KNN model.

Metrics Functions

```
# Make functions
#MAE
get_mae = function(data, model, name){
  name <- paste(name)
  data %>%
    mutate(predict_OTB_to_survive = OTB -  predict(model, data) * OTB) %>%
    summarise(!!name := round(mae(OTB_to_survive, predict_OTB_to_survive),4))
```

```
}
#MAE
mae_by_days = function(data, model, name){
  name <- paste(name)
  data %>%
    mutate(predict_OTB_to_survive = OTB -  predict(model, data) * OTB) %>%
    group_by(days_prior_cat) %>%
    summarise(!!name := round(mae(OTB_to_survive, predict_OTB_to_survive),4))
}

#MAPE
mape_by_days = function(data, model, name){
  name <-paste(name)
  data %>%
  mutate(predict_OTB_to_survive = OTB -  predict(model, data) * OTB) %>%
    filter(OTB_to_survive != 0)%>%
  group_by(days_prior_cat) %>%
  summarise(!!name := round(mape(OTB_to_survive, predict_OTB_to_survive) ,4))
}

#MASE
mase_by_days = function(data, model, name){
  name <-paste(name)
  data %>%
    mutate(predict_OTB_to_survive = OTB - predict(model, data) * OTB) %>%
    group_by(days_prior_cat) %>%
    summarise(!!name :=  round(sum(abs(OTB_to_survive - predict_OTB_to_survive))/
                                 sum(abs(OTB_to_survive - naive_survive_pred)),digits = 4))

}
```

KNN

```
# METRICS FUNCTIONS FOR KNN

#MAE
mae_knn = function(data, model, name){
  name <- paste(name)
  data %>%
    mutate(predict_OTB_to_survive = OTB -  model$pred * OTB) %>%
    group_by(days_prior_cat) %>%
    summarise(!!name := round(mae(OTB_to_survive, predict_OTB_to_survive),4))
}

#MAPE
mape_knn = function(data, model, name){
  name <-paste(name)
  data %>%
  mutate(predict_OTB_to_survive = OTB -  model$pred * OTB) %>%
    filter(OTB_to_survive != 0)%>%
  group_by(days_prior_cat) %>%
```

```
  summarise(!!name := round(mape(OTB_to_survive, predict_OTB_to_survive) ,4))
}

#MASE
mase_knn = function(data, model, name){
  name <-paste(name)
  data %>%
    mutate(predict_OTB_to_survive = OTB - model$pred * OTB) %>%
    group_by(days_prior_cat) %>%
    summarise(!!name :=  round(sum(abs(OTB_to_survive - predict_OTB_to_survive))/
                                 sum(abs(OTB_to_survive - naive_survive_pred)),digits = 4))

}

# FUNCTIONS TO FIND BEST K
# function calculate rmse
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
# define helper function for getting knn.reg predictions
# note: this function is highly specific to this situation and dataset
make_knn_pred = function(k = 1, training, predicting) {
  pred = FNN::knn.reg(train = training[-1],
                      test = predicting[-1],
                      y = training$cxl_rate, k = k)$pred
  act  = predicting$cxl_rate
  rmse(predicted = pred, actual = act)
}
```

Benchmark (group 5)

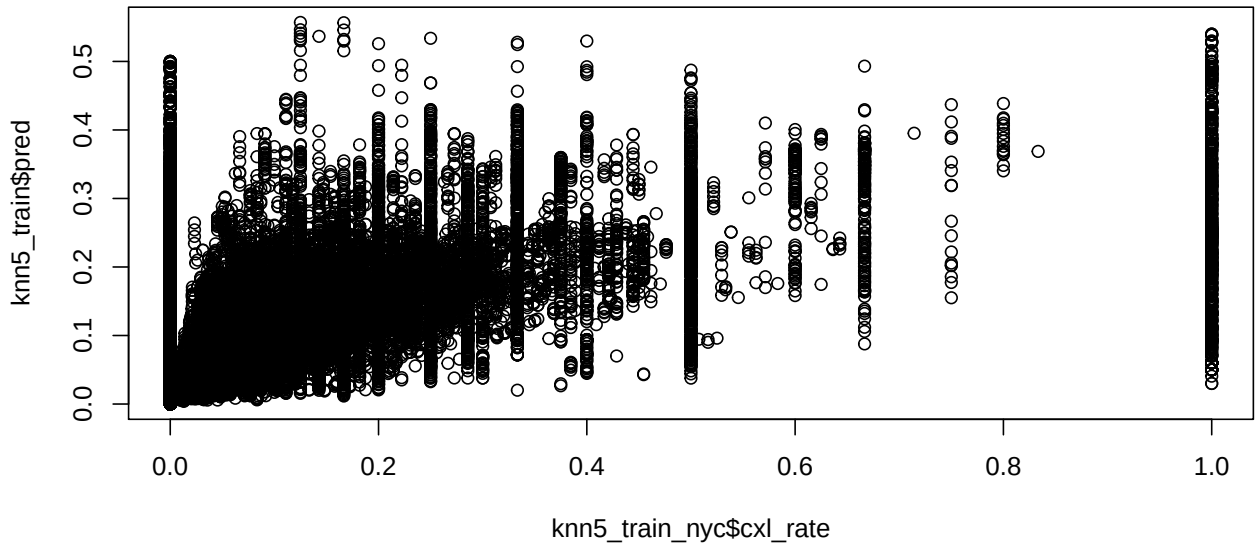
```
# CREATE DATA SET FOR KNN
#TRAIN
knn5_train_nyc <- train_nyc %>%
  select(c(cxl_rate, product_type, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type) # delete original cols after creating dummies
# y and x
knn5_train_nyc_X <- knn5_train_nyc[-1]
knn5_train_nyc_Y <- knn5_train_nyc[1]

#TEST
knn5_test_nyc <- test_nyc %>%
  select(c(cxl_rate,product_type, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type) # delete original cols after creating dummies
# y and x
knn5_test_nyc_X <- knn5_test_nyc[-1]
knn5_test_nyc_Y <- knn5_test_nyc[1]

# SCALE DATA
# calculate the pre-process parameters from the dataset
```

```r
preprocessParams <- preProcess(knn5_train_nyc_X, method=c("scale"))
# transform the dataset using the parameters
knn5_train_nyc_X <- predict(preprocessParams, knn5_train_nyc_X)
knn5_train_nyc[-1] <- knn5_train_nyc_X
knn5_test_nyc_X <- predict(preprocessParams, knn5_test_nyc_X)
knn5_test_nyc[-1] <- knn5_test_nyc_X


# DETERMINE BEST K

# define values of k to evaluate
k = c(5, 10, 25, 100, 150, 200)
# get requested train RMSEs
knn_trn_rmse = sapply(k, make_knn_pred,
                      training = knn5_train_nyc,
                      predicting = knn5_train_nyc)
# get requested test RMSEs
knn_tst_rmse = sapply(k, make_knn_pred,
                      training = knn5_train_nyc,
                      predicting = knn5_test_nyc)

# determine "best" k
best_k = k[which.min(knn_tst_rmse)]

# find overfitting, underfitting, and "best"" k
fit_status = ifelse(k < best_k, "Over", ifelse(k == best_k, "Best", "Under"))

# summarize results
knn_results = data.frame(
  k,
  round(knn_trn_rmse, 2),
  round(knn_tst_rmse, 2),
  fit_status
)
colnames(knn_results) = c("k", "Train RMSE", "Test RMSE", "Fit?")

# display results
knitr::kable(knn_results, escape = FALSE, booktabs = TRUE)
```

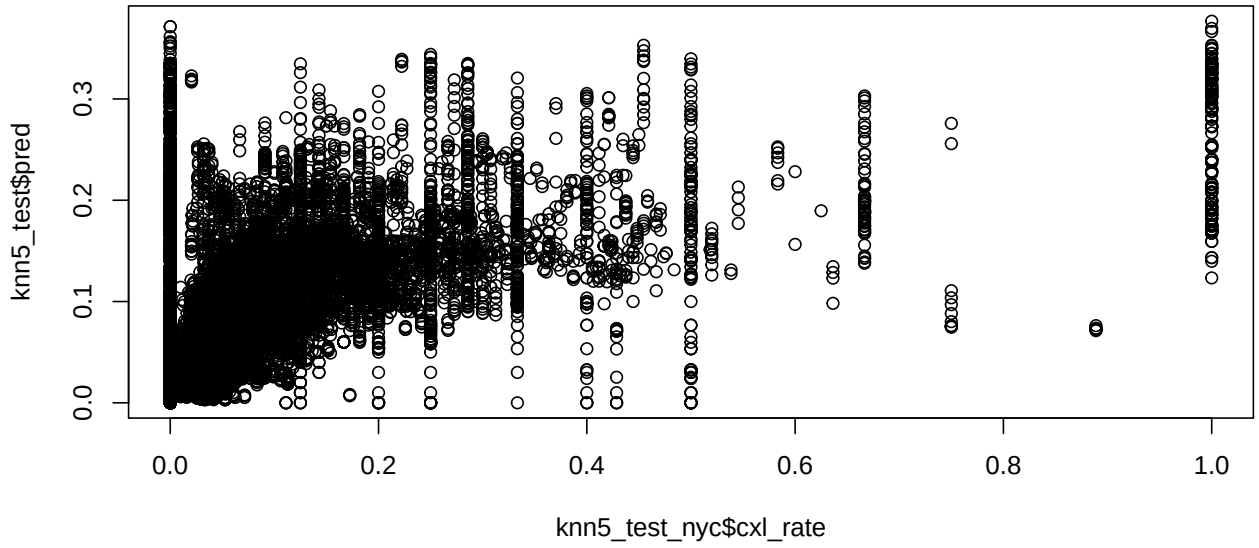
k	Train RMSE	Test RMSE	Fit?
5	0.10	0.15	Over
10	0.11	0.14	Over
25	0.12	0.14	Over
100	0.14	0.13	Best
150	0.14	0.13	Under
200	0.14	0.13	Under

```r
#KNN
knn5_train<-knn.reg(train = knn5_train_nyc_X,
                     y = knn5_train_nyc$cxl_rate, k = 100)
knn5_test<-knn.reg(train = knn5_train_nyc_X,
                   test = knn5_test_nyc_X,
                   y = knn5_train_nyc$cxl_rate, k = 100)
```

```r
plot(knn5_train_nyc$cxl_rate, knn5_train$pred)
```




```r
plot(knn5_test_nyc$cxl_rate, knn5_test$pred)
```



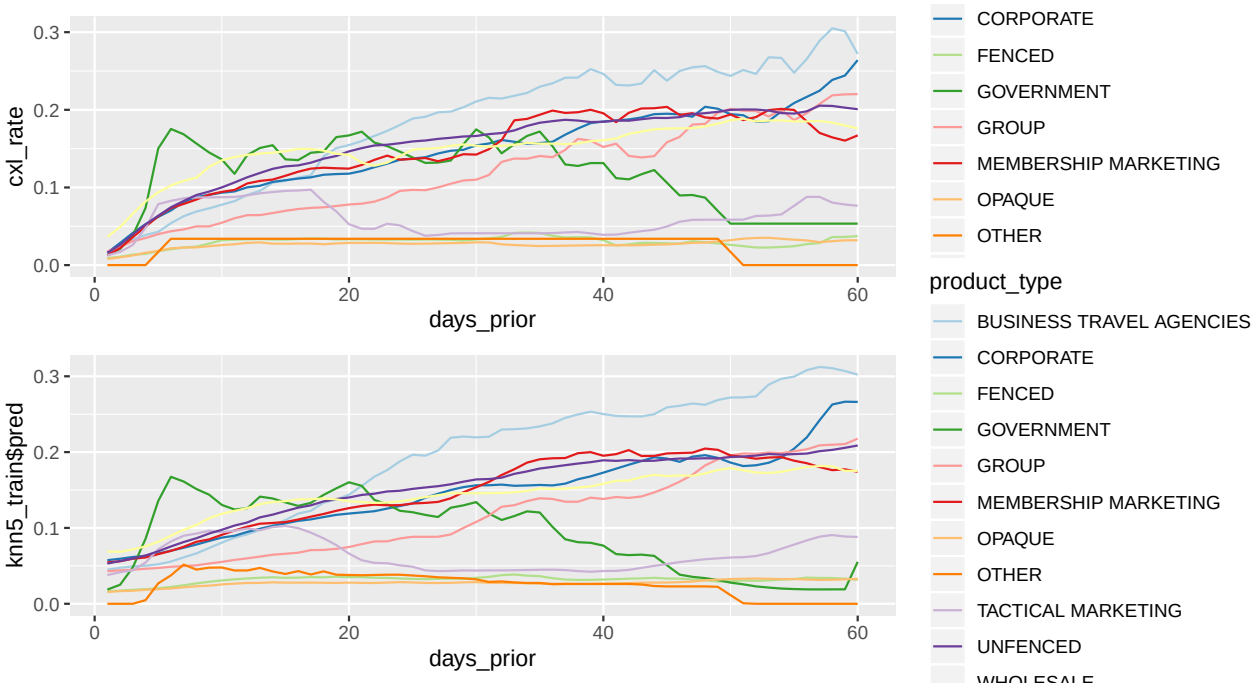

```r
# METRICS

# TRAIN (in-group)
# MAE
knn5_mae_in = mae_knn(train_nyc, knn5_train, 'knn5_mae_in')
#MAPE
knn5_mape_in = mape_knn(train_nyc, knn5_train, 'knn5_mape_in')
#MASE
knn5_mase_in = mase_knn(train_nyc, knn5_train, 'knn5_mase_in')

# TEST (out-group)
# MAE
knn5_mae_out = mae_knn(test_nyc, knn5_test, 'knn5_mae_out')
#MAPE
knn5_mape_out = mape_knn(test_nyc, knn5_test, 'knn5_mape_out')
```

Table 1: KNN 5

days_prior_cat	knn5_mae_in	knn5_mape_in	knn5_mase_in	knn5_mae_out	knn5_mape_out	knn5_mase
Day 01-07	2.0640	0.0416	0.3021	2.6060	0.0367	0.
Day 08-14	1.8087	0.0575	0.4490	2.2547	0.0453	0.
Day 15-20	1.6981	0.0613	0.4862	2.2896	0.0532	0.
Day 21-27	1.5530	0.0704	0.5140	2.2941	0.0609	0.
Day 28-60	1.0955	0.1071	0.5150	2.1852	0.1231	0.

```
#MASE
knn5_mase_out = mase_knn(test_nyc, knn5_test, 'knn5_mase_out')

#Summary table
kable(
knn5_mae_in %>%
  left_join(knn5_mape_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mase_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mae_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mape_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mase_out, c("days_prior_cat" = "days_prior_cat")),
  caption = "KNN 5"
)%>%
  kable_styling(bootstrap_options = 'condensed')
```

```
#visualize
grid.arrange(
  #original
ggplot(data = train_nyc, aes(color = product_type,x = days_prior,y = cxl_rate))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
  #product_type
ggplot(data = train_nyc, aes(color = product_type,x = days_prior,
                             y = knn5_train$pred))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),nrow = 2)
```



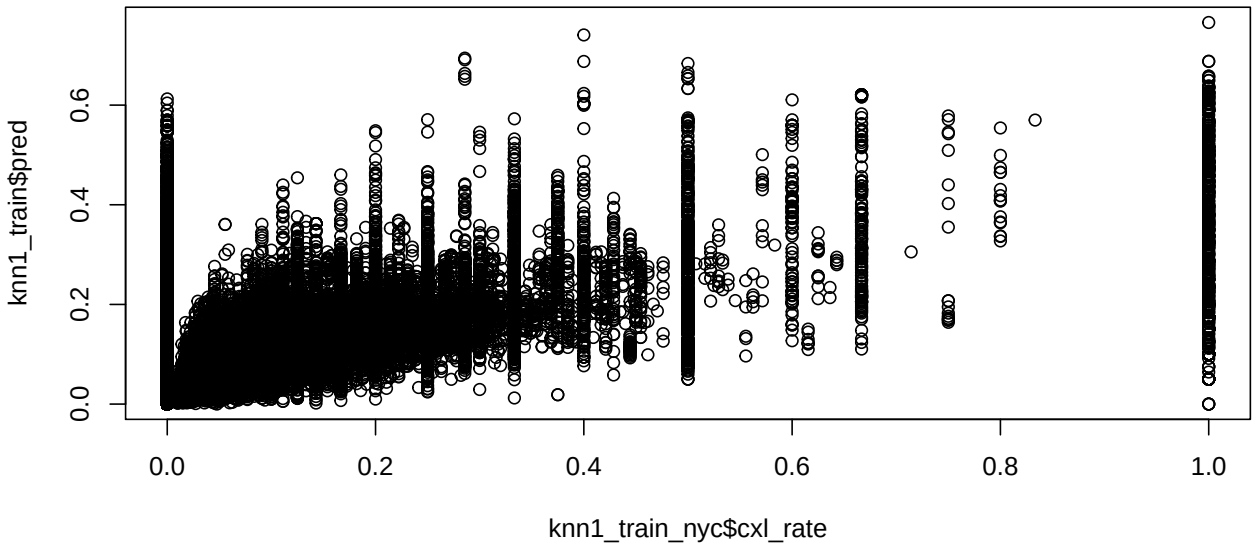
Group 1

```
# CREATE DATA SET FOR KNN
#TRAIN
knn1_train_nyc <- train_nyc %>%
  select(c(cxl_rate, product_type_regroup, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup) # delete original cols after creating dummies
# y and x
knn1_train_nyc_X <- knn1_train_nyc[-1]
knn1_train_nyc_Y <- knn1_train_nyc[1]

#TEST
knn1_test_nyc <- test_nyc %>%
  select(c(cxl_rate, product_type_regroup, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup) # delete original cols after creating dummies
# y and x
knn1_test_nyc_X <- knn1_test_nyc[-1]
knn1_test_nyc_Y <- knn1_test_nyc[1]

# SCALE DATA
# calculate the pre-process parameters from the dataset
preprocessParams <- preProcess(knn1_train_nyc_X, method=c("scale"))
# transform the dataset using the parameters
knn1_train_nyc_X <- predict(preprocessParams, knn1_train_nyc_X)
knn1_train_nyc[-1] <- knn1_train_nyc_X
knn1_test_nyc_X <- predict(preprocessParams, knn1_test_nyc_X)
knn1_test_nyc[-1] <- knn1_test_nyc_X
```

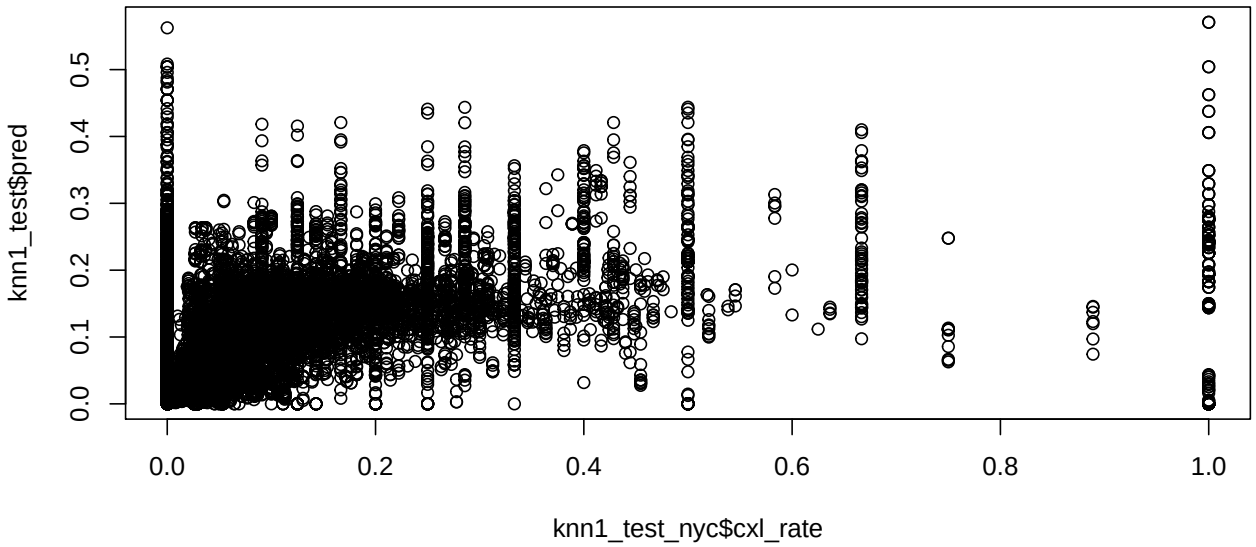
```r
# DETERMINE BEST K

# define values of k to evaluate
k = c(5, 10, 15, 20, 25, 100)
# get requested train RMSEs
knn_trn_rmse = sapply(k, make_knn_pred,
                      training = knn1_train_nyc,
                      predicting = knn1_train_nyc)
# get requested test RMSEs
knn_tst_rmse = sapply(k, make_knn_pred,
                      training = knn1_train_nyc,
                      predicting = knn1_test_nyc)

# determine "best" k
best_k = k[which.min(knn_tst_rmse)]

# find overfitting, underfitting, and "best"" k
fit_status = ifelse(k < best_k, "Over", ifelse(k == best_k, "Best", "Under"))

# summarize results
knn_results = data.frame(
  k,
  round(knn_trn_rmse, 2),
  round(knn_tst_rmse, 2),
  fit_status
)
colnames(knn_results) = c("k", "Train RMSE", "Test RMSE", "Fit?")

# display results
knitr::kable(knn_results, escape = FALSE, booktabs = TRUE)
```

k	Train RMSE	Test RMSE	Fit?
5	0.11	0.15	Over
10	0.12	0.15	Over
15	0.12	0.14	Over
20	0.13	0.14	Best
25	0.13	0.14	Under
100	0.14	0.14	Under

```r
#KNN
knn1_train<-knn.reg(train = knn1_train_nyc_X,
                    y = knn1_train_nyc$cxl_rate, k = 20)
knn1_test<-knn.reg(train = knn1_train_nyc_X,
                   test = knn1_test_nyc_X,
                   y = knn1_train_nyc$cxl_rate, k = 20)

plot(knn1_train_nyc$cxl_rate, knn1_train$pred)
```




```
plot(knn1_test_nyc$cxl_rate, knn1_test$pred)
```

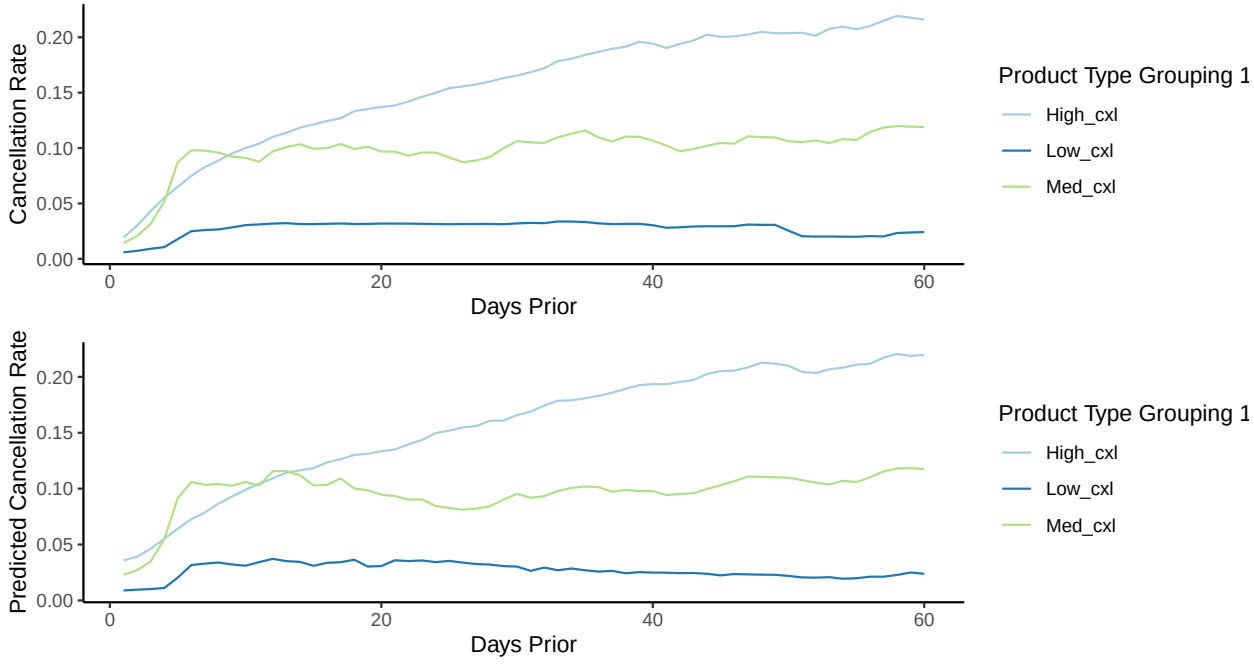



```
# METRICS

# TRAIN (in-group)
# MAE
knn1_mae_in = mae_knn(train_nyc, knn1_train, 'knn1_mae_in')
#MAPE
knn1_mape_in = mape_knn(train_nyc, knn1_train, 'knn1_mape_in')
#MASE
knn1_mase_in = mase_knn(train_nyc, knn1_train, 'knn1_mase_in')

# TEST (out-group)
# MAE
knn1_mae_out = mae_knn(test_nyc, knn1_test, 'knn1_mae_out')
#MAPE
knn1_mape_out = mape_knn(test_nyc, knn1_test, 'knn1_mape_out')
#MASE
knn1_mase_out = mase_knn(test_nyc, knn1_test, 'knn1_mase_out')
```

```
#Summary table

  data.frame(knn1_mae_in,
             knn1_mape_in[2],
             knn1_mase_in[2],
             knn1_mae_out[2],
             knn1_mape_out[2],
             knn1_mase_out[2])
```

```
##   days_prior_cat knn1_mae_in knn1_mape_in knn1_mase_in knn1_mae_out
## 1      Day 01-07      1.6602       0.0372       0.2430       2.3858
## 2      Day 08-14      1.7345       0.0576       0.4305       2.6007
## 3      Day 15-20      1.6633       0.0630       0.4762       2.5322
## 4      Day 21-27      1.4486       0.0713       0.4795       2.6460
## 5      Day 28-60      0.9589       0.0979       0.4508       2.2365
##   knn1_mape_out knn1_mase_out
## 1        0.0379        0.2717
## 2        0.0543        0.4174
## 3        0.0564        0.4996
## 4        0.0666        0.5875
## 5        0.1198        0.5524
```

```
grid.arrange(
ggplot(data = train_nyc, aes(color = product_type_regroup,x = days_prior,y = cxl_rate))+stat_summary(fu
  scale_color_brewer(palette = "Paired") +theme_classic() + labs(x = 'Days Prior', y = 'Cancellation Ra
ggplot(data = train_nyc, aes(color = product_type_regroup,x = days_prior,y = knn1_train$pred))+stat_summ
  scale_color_brewer(palette = "Paired")+theme_classic() + labs(x = 'Days Prior', y = 'Predicted Cancel
nrow = 2)
```



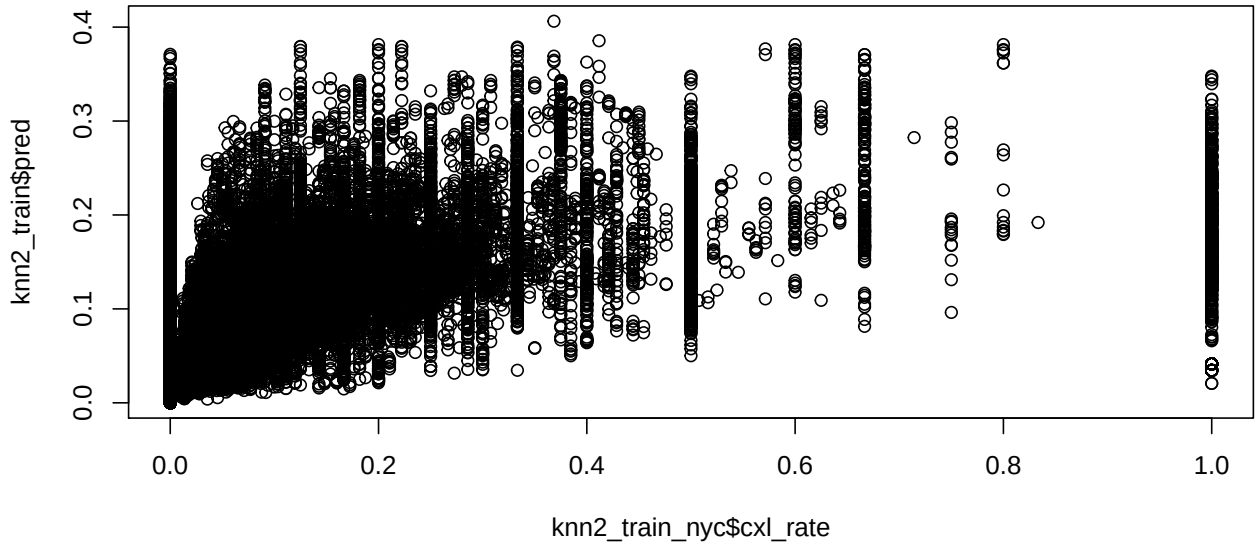
Group 2

```
# CREATE DATA SET FOR KNN
#TRAIN
knn2_train_nyc <- train_nyc %>%
  select(c(cxl_rate, product_type_regroup2, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup2) # delete original cols after creating dummies
# y and x
knn2_train_nyc_X <- knn2_train_nyc[-1]
knn2_train_nyc_Y <- knn2_train_nyc[1]

#TEST
knn2_test_nyc <- test_nyc %>%
  select(c(cxl_rate, product_type_regroup2, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup2) # delete original cols after creating dummies
# y and x
knn2_test_nyc_X <- knn2_test_nyc[-1]
knn2_test_nyc_Y <- knn2_test_nyc[1]

# SCALE DATA
# calculate the pre-process parameters from the dataset
preprocessParams <- preProcess(knn2_train_nyc_X, method=c("scale"))
# transform the dataset using the parameters
knn2_train_nyc_X <- predict(preprocessParams, knn2_train_nyc_X)
knn2_train_nyc[-1] <- knn2_train_nyc_X
knn2_test_nyc_X <- predict(preprocessParams, knn2_test_nyc_X)
knn2_test_nyc[-1] <- knn2_test_nyc_X


# DETERMINE BEST K

# define values of k to evaluate
k = c(100, 120, 145, 150, 155)
# get requested train RMSEs
knn_trn_rmse = sapply(k, make_knn_pred,
                      training = knn2_train_nyc,
                      predicting = knn2_train_nyc)
# get requested test RMSEs
knn_tst_rmse = sapply(k, make_knn_pred,
                      training = knn2_train_nyc,
                      predicting = knn2_test_nyc)

# determine "best" k
best_k = k[which.min(knn_tst_rmse)]

# find overfitting, underfitting, and "best"" k
fit_status = ifelse(k < best_k, "Over", ifelse(k == best_k, "Best", "Under"))

# summarize results
knn_results = data.frame(
  k,
```

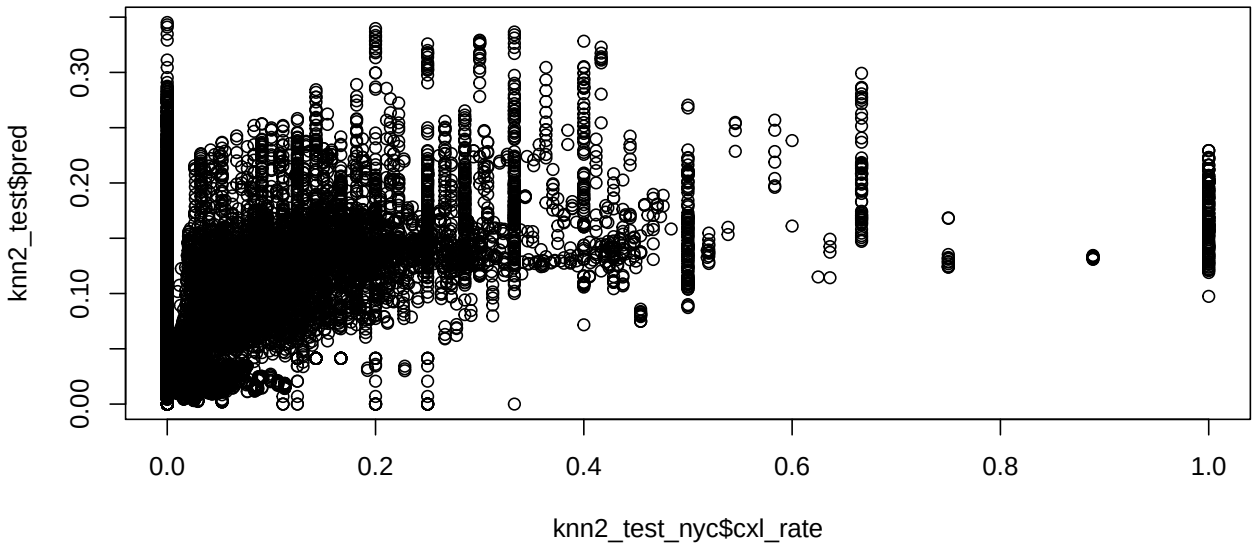
```
  round(knn_trn_rmse, 2),
  round(knn_tst_rmse, 2),
  fit_status
)
colnames(knn_results) = c("k", "Train RMSE", "Test RMSE", "Fit?")

# display results
knitr::kable(knn_results, escape = FALSE, booktabs = TRUE)
```

k	Train RMSE	Test RMSE	Fit?
100	0.14	0.14	Best
120	0.15	0.14	Under
145	0.15	0.14	Under
150	0.15	0.14	Under
155	0.15	0.14	Under

```
#KNN
knn2_train<-knn.reg(train = knn2_train_nyc_X,
                    y = knn2_train_nyc$cxl_rate, k = 145)
knn2_test<-knn.reg(train = knn2_train_nyc_X,
                   test = knn2_test_nyc_X,
                   y = knn2_train_nyc$cxl_rate, k = 145)

plot(knn2_train_nyc$cxl_rate, knn2_train$pred)
```




```
plot(knn2_test_nyc$cxl_rate, knn2_test$pred)
```



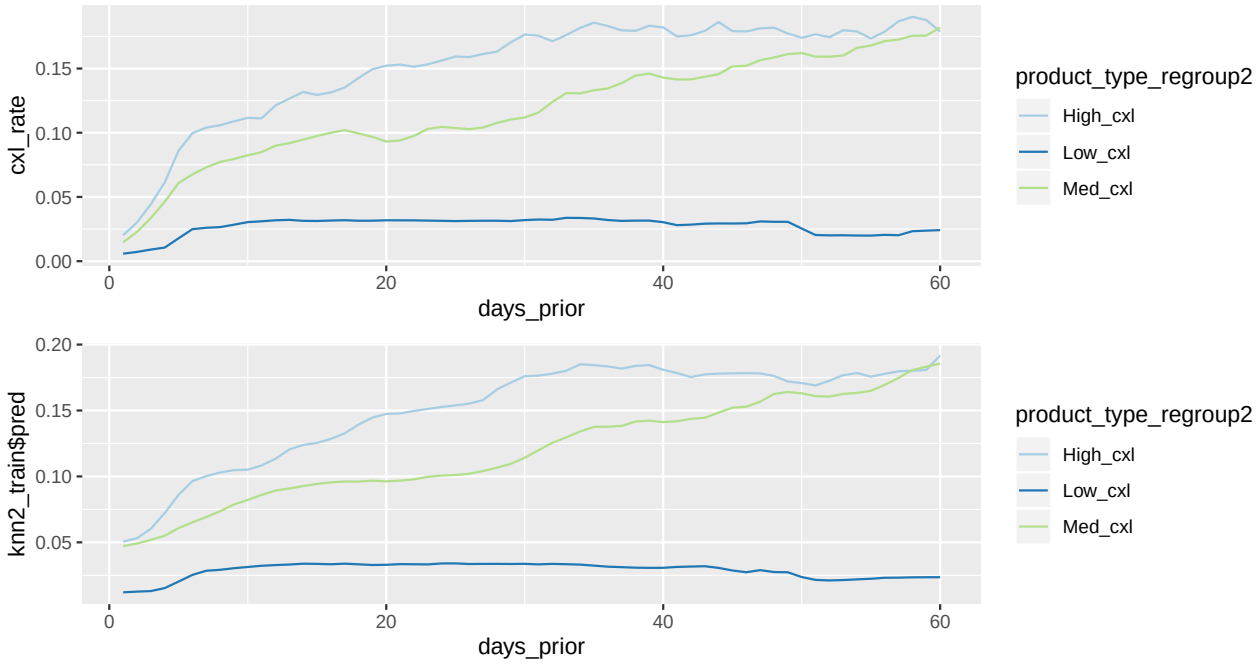

```
# METRICS

# TRAIN (in-group)
# MAE
knn2_mae_in = mae_knn(train_nyc, knn2_train, 'knn2_mae_in')
#MAPE
knn2_mape_in = mape_knn(train_nyc, knn2_train, 'knn2_mape_in')
#MASE
knn2_mase_in = mase_knn(train_nyc, knn2_train, 'knn2_mase_in')

# TEST (out-group)
# MAE
knn2_mae_out = mae_knn(test_nyc, knn2_test, 'knn2_mae_out')
#MAPE
knn2_mape_out = mape_knn(test_nyc, knn2_test, 'knn2_mape_out')
#MASE
knn2_mase_out = mase_knn(test_nyc, knn2_test, 'knn2_mase_out')

#Summary table
kable(
  data.frame(knn2_mae_in,
             knn2_mape_in[2],
             knn2_mase_in[2],
             knn2_mae_out[2],
             knn2_mape_out[2],
             knn2_mase_out[2]),
  caption = "KNN 2")%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
#visualize
grid.arrange(
  #original
ggplot(data = train_nyc, aes(color = product_type_regroup2,x = days_prior,y = cxl_rate))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
  #product_type regrouped
ggplot(data = train_nyc, aes(color = product_type_regroup2,x = days_prior,
```

Table 2: KNN 2

days_prior_cat	knn2_mae_in	knn2_mape_in	knn2_mase_in	knn2_mae_out	knn2_mape_out	knn2_mase
Day 01-07	2.1334	0.0434	0.3122	2.7603	0.0382	0.
Day 08-14	1.9618	0.0614	0.4870	2.5955	0.0504	0.
Day 15-20	1.8645	0.0658	0.5338	2.6573	0.0569	0.
Day 21-27	1.6967	0.0781	0.5616	2.5469	0.0701	0.
Day 28-60	1.1829	0.1146	0.5561	2.2237	0.1280	0.

```
                          y = knn2_train$pred))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
nrow = 2)
```



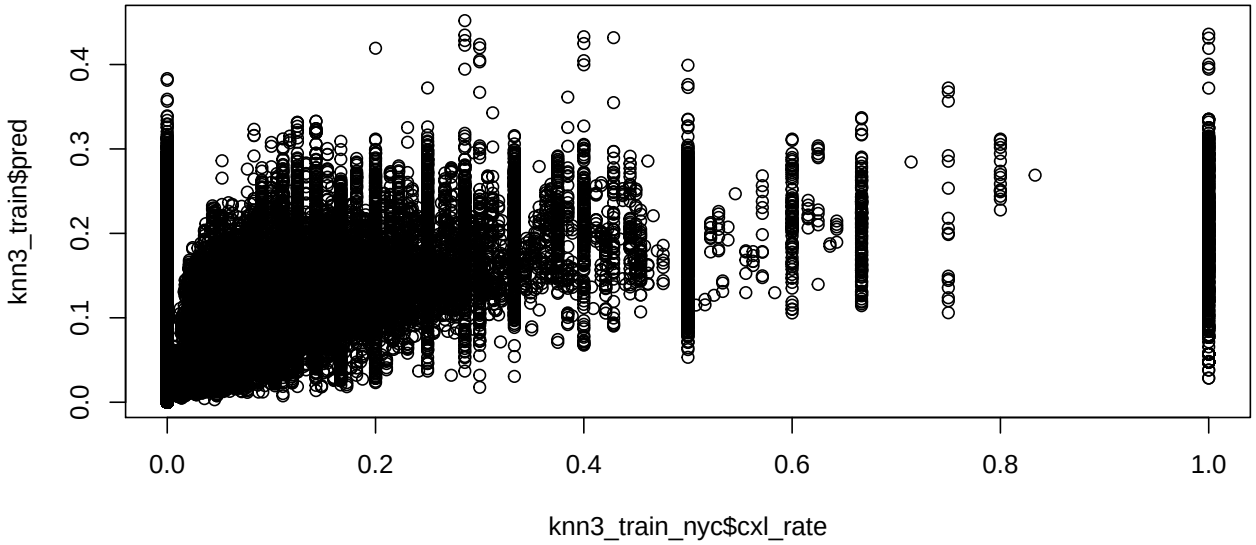
Group 3

```
# CREATE DATA SET FOR KNN
#TRAIN
knn3_train_nyc <- train_nyc %>%
  select(c(cxl_rate, product_type_regroup3, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup3) # delete original cols after creating dummies
# y and x
knn3_train_nyc_X <- knn3_train_nyc[-1]
knn3_train_nyc_Y <- knn3_train_nyc[1]

#TEST
knn3_test_nyc <- test_nyc %>%
  select(c(cxl_rate, product_type_regroup3, room_price, cummulative_gross_bookings, days_prior)) %>%
```

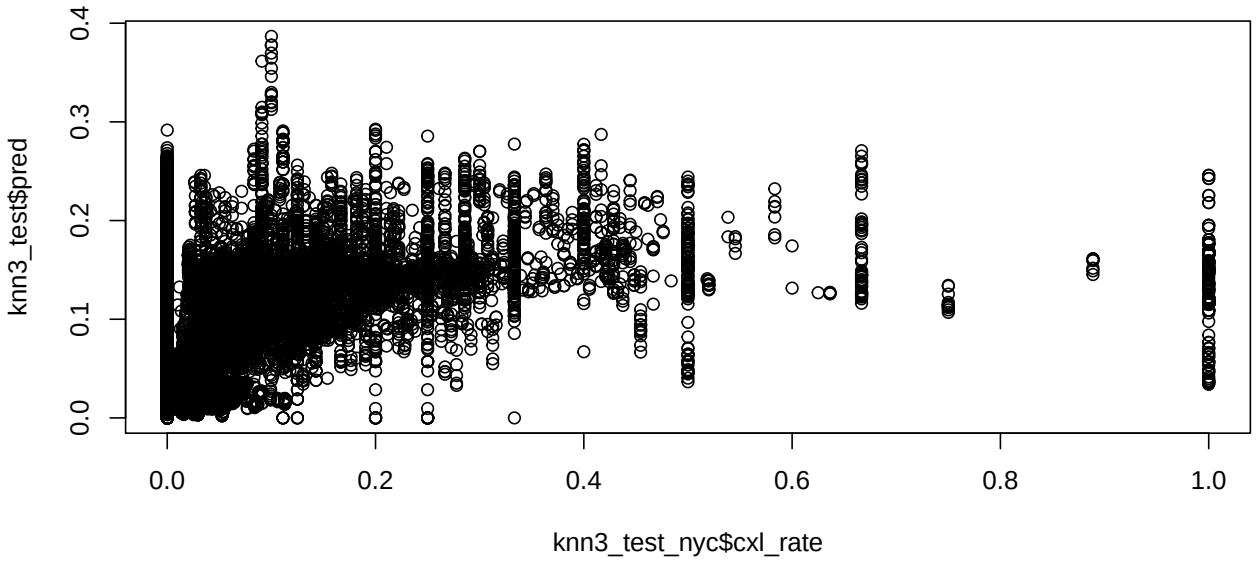
```r
  dummy_cols() %>%
  select(-product_type_regroup3) # delete original cols after creating dummies
# y and x
knn3_test_nyc_X <- knn3_test_nyc[-1]
knn3_test_nyc_Y <- knn3_test_nyc[1]

# SCALE DATA
# calculate the pre-process parameters from the dataset
preprocessParams <- preProcess(knn3_train_nyc_X, method=c("scale"))
# transform the dataset using the parameters
knn3_train_nyc_X <- predict(preprocessParams, knn3_train_nyc_X)
knn3_train_nyc[-1] <- knn3_train_nyc_X
knn3_test_nyc_X <- predict(preprocessParams, knn3_test_nyc_X)
knn3_test_nyc[-1] <- knn3_test_nyc_X


# DETERMINE BEST K

# define values of k to evaluate
k = c(100, 105, 110, 115, 120)
# get requested train RMSEs
knn_trn_rmse = sapply(k, make_knn_pred,
                      training = knn3_train_nyc,
                      predicting = knn3_train_nyc)
# get requested test RMSEs
knn_tst_rmse = sapply(k, make_knn_pred,
                      training = knn3_train_nyc,
                      predicting = knn3_test_nyc)

# determine "best" k
best_k = k[which.min(knn_tst_rmse)]

# find overfitting, underfitting, and "best"" k
fit_status = ifelse(k < best_k, "Over", ifelse(k == best_k, "Best", "Under"))

# summarize results
knn_results = data.frame(
  k,
  round(knn_trn_rmse, 2),
  round(knn_tst_rmse, 2),
  fit_status
)
colnames(knn_results) = c("k", "Train RMSE", "Test RMSE", "Fit?")

# display results
knitr::kable(knn_results, escape = FALSE, booktabs = TRUE)
```

k	Train RMSE	Test RMSE	Fit?
100	0.15	0.15	Best
105	0.15	0.15	Under
110	0.15	0.15	Under
115	0.15	0.15	Under
120	0.15	0.15	Under

```r
#KNN
knn3_train<-knn.reg(train = knn3_train_nyc_X,
                    y = knn3_train_nyc$cxl_rate, k = 105)
knn3_test<-knn.reg(train = knn3_train_nyc_X,
                   test = knn3_test_nyc_X,
                   y = knn3_train_nyc$cxl_rate, k = 105)

plot(knn3_train_nyc$cxl_rate, knn3_train$pred)
```




```r
plot(knn3_test_nyc$cxl_rate, knn3_test$pred)
```



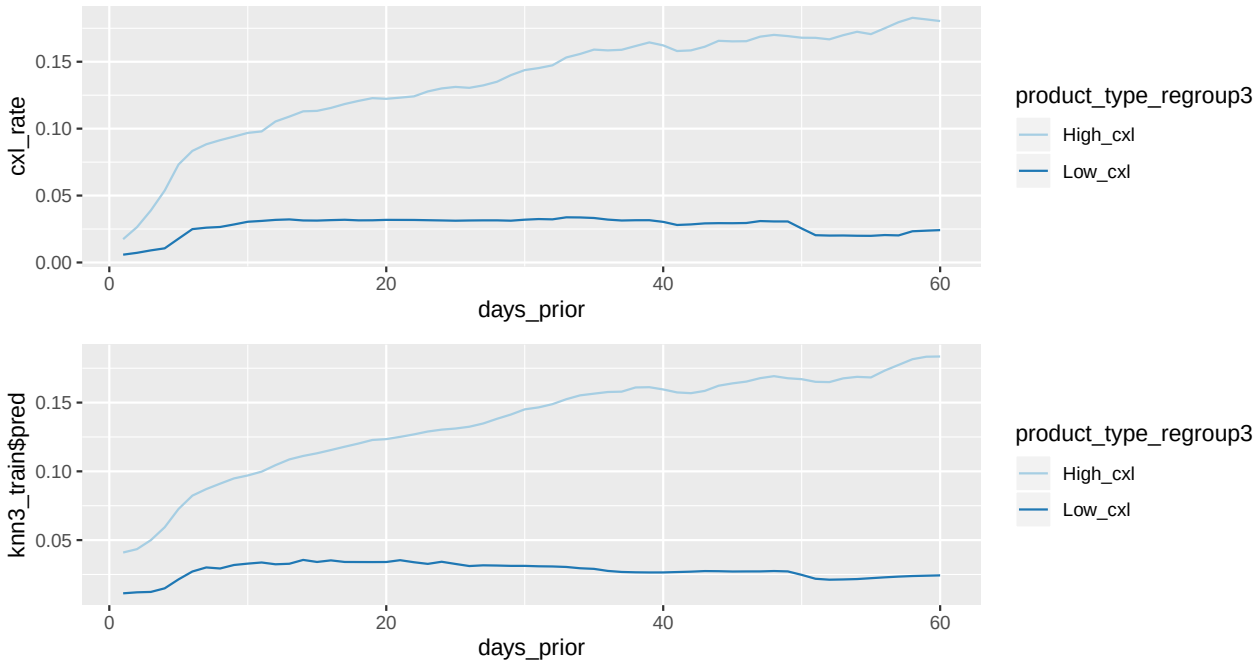

```r
# METRICS

# TRAIN (in-group)
# MAE
knn3_mae_in = mae_knn(train_nyc, knn3_train, 'knn3_mae_in')
#MAPE
knn3_mape_in = mape_knn(train_nyc, knn3_train, 'knn3_mape_in')
#MASE
```

Table 3: KNN 3

days_prior_cat	knn3_mae_in	knn3_mape_in	knn3_mase_in	knn3_mae_out	knn3_mape_out	knn3_mase
Day 01-07	1.9525	0.0426	0.2858	2.4956	0.0365	0.
Day 08-14	1.9951	0.0642	0.4952	2.6534	0.0530	0.
Day 15-20	1.9338	0.0711	0.5536	2.7798	0.0614	0.
Day 21-27	1.7404	0.0841	0.5761	2.7583	0.0767	0.
Day 28-60	1.1906	0.1159	0.5598	2.2427	0.1298	0.

```
knn3_mase_in = mase_knn(train_nyc, knn3_train, 'knn3_mase_in')

# TEST (out-group)
# MAE
knn3_mae_out = mae_knn(test_nyc, knn3_test, 'knn3_mae_out')
#MAPE
knn3_mape_out = mape_knn(test_nyc, knn3_test, 'knn3_mape_out')
#MASE
knn3_mase_out = mase_knn(test_nyc, knn3_test, 'knn3_mase_out')

#Summary table
kable(
  data.frame(knn3_mae_in,
             knn3_mape_in[2],
             knn3_mase_in[2],
             knn3_mae_out[2],
             knn3_mape_out[2],
             knn3_mase_out[2]),
  caption = "KNN 3")%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
#visualize
grid.arrange(
  #original
ggplot(data = train_nyc, aes(color = product_type_regroup3,x = days_prior,y = cxl_rate))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
  #product_type regrouped
ggplot(data = train_nyc, aes(color = product_type_regroup3,x = days_prior,
                             y = knn3_train$pred))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
nrow = 2)
```



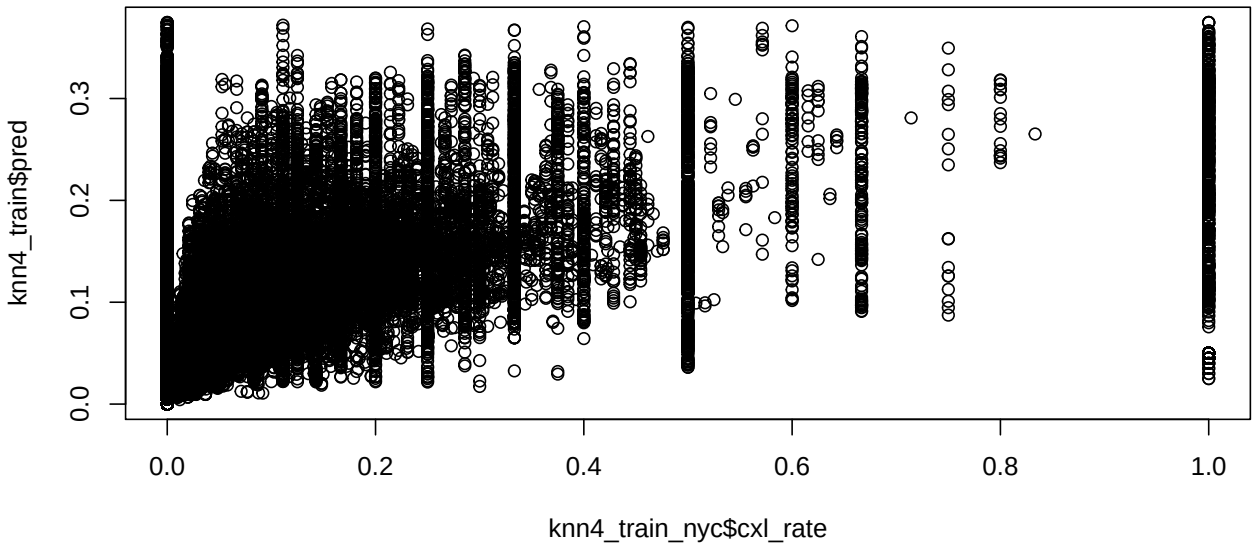
Group 4

```
# CREATE DATA SET FOR KNN
#TRAIN
knn4_train_nyc <- train_nyc %>%
  select(c(cxl_rate, product_type_regroup4, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup4) # delete original cols after creating dummies
# y and x
knn4_train_nyc_X <- knn4_train_nyc[-1]
knn4_train_nyc_Y <- knn4_train_nyc[1]

#TEST
knn4_test_nyc <- test_nyc %>%
  select(c(cxl_rate, product_type_regroup4, room_price, cummulative_gross_bookings, days_prior)) %>%
  dummy_cols() %>%
  select(-product_type_regroup4) # delete original cols after creating dummies
# y and x
knn4_test_nyc_X <- knn4_test_nyc[-1]
knn4_test_nyc_Y <- knn4_test_nyc[1]

# SCALE DATA
# calculate the pre-process parameters from the dataset
preprocessParams <- preProcess(knn4_train_nyc_X, method=c("scale"))
# transform the dataset using the parameters
knn4_train_nyc_X <- predict(preprocessParams, knn4_train_nyc_X)
knn4_train_nyc[-1] <- knn4_train_nyc_X
knn4_test_nyc_X <- predict(preprocessParams, knn4_test_nyc_X)
knn4_test_nyc[-1] <- knn4_test_nyc_X
```

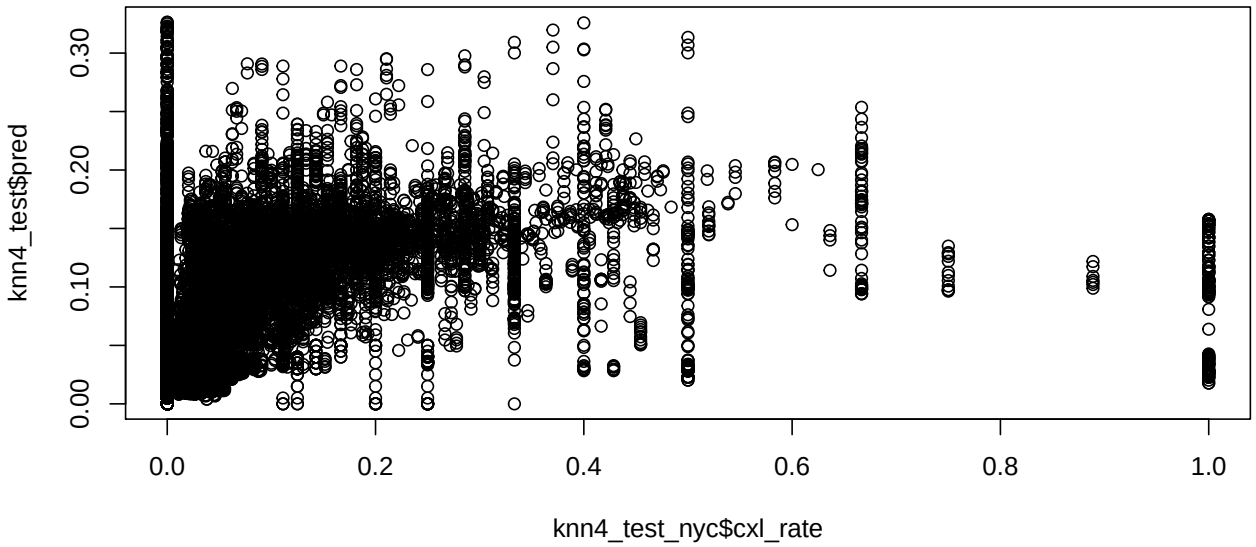
```r
# DETERMINE BEST K

# define values of k to evaluate
k = c(5, 10, 25, 100, 150, 200)
# get requested train RMSEs
knn_trn_rmse = sapply(k, make_knn_pred,
                      training = knn4_train_nyc,
                      predicting = knn4_train_nyc)
# get requested test RMSEs
knn_tst_rmse = sapply(k, make_knn_pred,
                      training = knn4_train_nyc,
                      predicting = knn4_test_nyc)

# determine "best" k
best_k = k[which.min(knn_tst_rmse)]

# find overfitting, underfitting, and "best"" k
fit_status = ifelse(k < best_k, "Over", ifelse(k == best_k, "Best", "Under"))

# summarize results
knn_results = data.frame(
  k,
  round(knn_trn_rmse, 2),
  round(knn_tst_rmse, 2),
  fit_status
)
colnames(knn_results) = c("k", "Train RMSE", "Test RMSE", "Fit?")

# display results
knitr::kable(knn_results, escape = FALSE, booktabs = TRUE)
```

k	Train RMSE	Test RMSE	Fit?
5	0.11	0.16	Over
10	0.12	0.15	Over
25	0.13	0.15	Over
100	0.14	0.15	Over
150	0.15	0.15	Over
200	0.15	0.15	Best

```r
#KNN
knn4_train<-knn.reg(train = knn4_train_nyc_X,
                    y = knn4_train_nyc$cxl_rate, k = 200)
knn4_test<-knn.reg(train = knn4_train_nyc_X,
                   test = knn4_test_nyc_X,
                   y = knn4_train_nyc$cxl_rate, k = 200)

plot(knn4_train_nyc$cxl_rate, knn4_train$pred)
```




```
plot(knn4_test_nyc$cxl_rate, knn4_test$pred)
```




```
# METRICS

# TRAIN (in-group)
# MAE
knn4_mae_in = mae_knn(train_nyc, knn4_train, 'knn4_mae_in')
#MAPE
knn4_mape_in = mape_knn(train_nyc, knn4_train, 'knn4_mape_in')
#MASE
knn4_mase_in = mase_knn(train_nyc, knn4_train, 'knn4_mase_in')

# TEST (out-group)
# MAE
knn4_mae_out = mae_knn(test_nyc, knn4_test, 'knn4_mae_out')
#MAPE
knn4_mape_out = mape_knn(test_nyc, knn4_test, 'knn4_mape_out')
#MASE
knn4_mase_out = mase_knn(test_nyc, knn4_test, 'knn4_mase_out')
```

Table 4: KNN 4

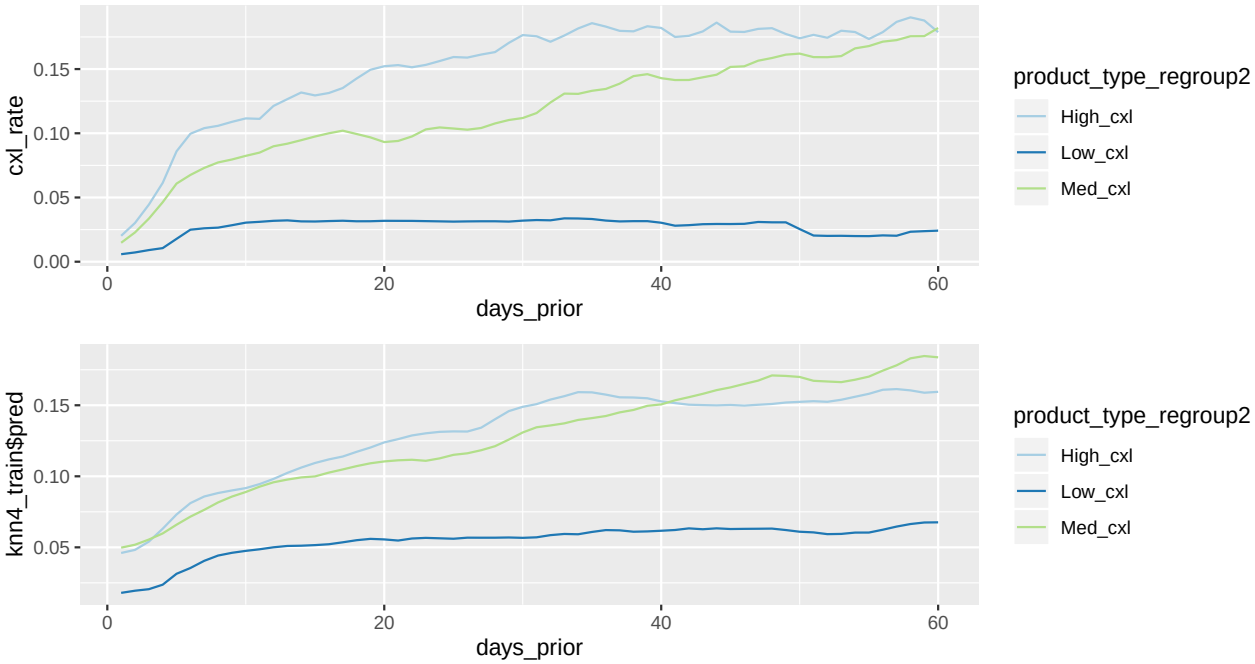
days_prior_cat	knn4_mae_in	knn4_mape_in	knn4_mase_in	knn4_mae_out	knn4_mape_out	knn4_mase
Day 01-07	2.2098	0.0456	0.3234	2.7442	0.0367	0.
Day 08-14	2.1781	0.0671	0.5407	2.5586	0.0493	0.
Day 15-20	2.1278	0.0736	0.6092	2.6451	0.0579	0.
Day 21-27	1.8995	0.0848	0.6287	2.7768	0.0720	0.
Day 28-60	1.3810	0.1228	0.6493	2.3693	0.1284	0.

```
#Summary table
kable(
knn4_mae_in %>%
  left_join(knn4_mape_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mase_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mae_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mape_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mase_out, c("days_prior_cat" = "days_prior_cat")),
  caption = "KNN 4"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
#visualize
grid.arrange(
  #original
ggplot(data = train_nyc, aes(color = product_type_regroup2,x = days_prior,y = cxl_rate))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
  #product_type regrouped
ggplot(data = train_nyc, aes(color = product_type_regroup2,x = days_prior,
                             y = knn4_train$pred))+
  stat_summary(fun.y = 'mean', geom = 'line')+
  scale_color_brewer(palette = "Paired"),
nrow = 2)
```

Table 5: In-sample metrics - MAE

days_prior_cat	knn1_mae_in	knn2_mae_in	knn3_mae_in	knn4_mae_in	knn5_mae_in
Day 01-07	1.6602	2.1334	1.9525	2.2098	2.0640
Day 08-14	1.7345	1.9618	1.9951	2.1781	1.8087
Day 15-20	1.6633	1.8645	1.9338	2.1278	1.6981
Day 21-27	1.4486	1.6967	1.7404	1.8995	1.5530
Day 28-60	0.9589	1.1829	1.1906	1.3810	1.0955



Summarise Metrics

```
# IN SAMPLE
# mae
kable(
knn1_mae_in %>%
  left_join(knn2_mae_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mae_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mae_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mae_in, c("days_prior_cat" = "days_prior_cat")) ,
  caption = "In-sample metrics - MAE"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
#mape
kable(
knn1_mape_in %>%
  left_join(knn2_mape_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mape_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mape_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mape_in, c("days_prior_cat" = "days_prior_cat")),
```

Table 6: In-sample metrics - MAPE

days_prior_cat	knn1_mape_in	knn2_mape_in	knn3_mape_in	knn4_mape_in	knn5_mape_in
Day 01-07	0.0372	0.0434	0.0426	0.0456	0.0416
Day 08-14	0.0576	0.0614	0.0642	0.0671	0.0575
Day 15-20	0.0630	0.0658	0.0711	0.0736	0.0613
Day 21-27	0.0713	0.0781	0.0841	0.0848	0.0704
Day 28-60	0.0979	0.1146	0.1159	0.1228	0.1071

Table 7: In-sample metrics - MASE

days_prior_cat	knn1_mase_in	knn2_mase_in	knn3_mase_in	knn4_mase_in	knn5_mase_in
Day 01-07	0.2430	0.3122	0.2858	0.3234	0.3021
Day 08-14	0.4305	0.4870	0.4952	0.5407	0.4490
Day 15-20	0.4762	0.5338	0.5536	0.6092	0.4862
Day 21-27	0.4795	0.5616	0.5761	0.6287	0.5140
Day 28-60	0.4508	0.5561	0.5598	0.6493	0.5150

```
  caption = "In-sample metrics - MAPE"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
#mase
kable(
knn1_mase_in %>%
  left_join(knn2_mase_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mase_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mase_in, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mase_in, c("days_prior_cat" = "days_prior_cat")),
  caption = "In-sample metrics - MASE"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
# OUT SAMPLE
kable(
knn1_mae_out %>%
  left_join(knn2_mae_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mae_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mae_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mae_out, c("days_prior_cat" = "days_prior_cat")) ,
  caption = "Out-sample metrics - MAE"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
kable(
knn1_mape_out %>%
  left_join(knn2_mape_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mape_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mape_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mape_out, c("days_prior_cat" = "days_prior_cat")),
  caption = "Out-sample metrics - MAPE"
)%>%
```

Table 8: Out-sample metrics - MAE

days_prior_cat	knn1_mae_out	knn2_mae_out	knn3_mae_out	knn4_mae_out	knn5_mae_out
Day 01-07	2.3858	2.7603	2.4956	2.7442	2.6060
Day 08-14	2.6007	2.5955	2.6534	2.5586	2.2547
Day 15-20	2.5322	2.6573	2.7798	2.6451	2.2896
Day 21-27	2.6460	2.5469	2.7583	2.7768	2.2941
Day 28-60	2.2365	2.2237	2.2427	2.3693	2.1852

Table 9: Out-sample metrics - MAPE

days_prior_cat	knn1_mape_out	knn2_mape_out	knn3_mape_out	knn4_mape_out	knn5_mape_out
Day 01-07	0.0379	0.0382	0.0365	0.0367	0.0367
Day 08-14	0.0543	0.0504	0.0530	0.0493	0.0453
Day 15-20	0.0564	0.0569	0.0614	0.0579	0.0532
Day 21-27	0.0666	0.0701	0.0767	0.0720	0.0609
Day 28-60	0.1198	0.1280	0.1298	0.1284	0.1231

```
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

```
kable(
knn1_mase_out %>%
  left_join(knn2_mase_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn3_mase_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn4_mase_out, c("days_prior_cat" = "days_prior_cat")) %>%
  left_join(knn5_mase_out, c("days_prior_cat" = "days_prior_cat")),
  caption = "Out-sample metrics - MASE"
)%>%
  kable_styling(bootstrap_options = 'condensed', full_width = F)
```

Table 10: Out-sample metrics - MASE

days_prior_cat	knn1_mase_out	knn2_mase_out	knn3_mase_out	knn4_mase_out	knn5_mase_out
Day 01-07	0.2717	0.3143	0.2842	0.3125	0.2968
Day 08-14	0.4174	0.4166	0.4259	0.4107	0.3619
Day 15-20	0.4996	0.5243	0.5485	0.5219	0.4517
Day 21-27	0.5875	0.5655	0.6124	0.6165	0.5093
Day 28-60	0.5524	0.5493	0.5539	0.5852	0.5397