Brooke O’Donnell

Project 4

KNN- Nearest Neighbor Method

We have worked on the “Churn” data set several times this semester: building both decision tree, regression tree, and random forest models to predict whether people churn or not. In this project, we will try the K Nearest Neighbor model on it. Recall that the “Churn” variable is a binary categorical variable showing whether a customer has “churned” or not. “Churn” means terminating the subscription or leaving the company.Please include the following variables in your subset: gender, SeniorCitizen, Partner Dependents, tenure, MonthlyCharges, Contract, Churn.Before you start estimating KNN models, please make sure you code all variables properly (ex. all character variables into factors.) KNN requires that all factors be coded into numeric dummy variables. Please include the following segment “step\_dummy(all\_nominal(), -all\_outcomes())”as one of the pre-processing steps (when creating recipe). This step codes all factors into dummy variables (a k-category variable is coded into k-1 dummies) except the target.

1. Build a tuning procedure for the hyperparameter of K using “tidymodels.” Please make sure you include all the following steps:
   1. split data into training and testing sets (70%-30% split)

set.seed(123)

subset.split<-initial\_split(subset1, prop=.7, strata = churn)

subset.train<-training(subset.split)

subset.test<-testing(subset.split)

* 1. specify a re-sampling procedure (10-fold cross-validation)

*# split training set 90|10*

k\_fold<-vfold\_cv(subset.train)

* 1. create a recipe and preprocessing steps

model.rec<-recipe(churn~., subset.train) %>% step\_range(all\_numeric()) %>% step\_dummy(all\_nominal(),-all\_outcomes())

model.rec %>% prep() %>% juice() %>% summary()

tenure monthlycharges churn gender\_Male seniorcitizen\_X1

Min. :0.0000 Min. :0.0000 No :3622 Min. :0.0000 Min. :0.0000

1st Qu.:0.1250 1st Qu.:0.1292 Yes:1309 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.4028 Median :0.4555 Median :1.0000 Median :0.0000

Mean :0.4508 Mean :0.4327 Mean :0.5013 Mean :0.1659

3rd Qu.:0.7639 3rd Qu.:0.6879 3rd Qu.:1.0000 3rd Qu.:0.0000

Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000

partner\_Yes dependents\_Yes contract\_One.year contract\_Two.year

Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000

1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000

Median :0.0000 Median :0.0000 Median :0.0000 Median :0.0000

Mean :0.4788 Mean :0.2999 Mean :0.2071 Mean :0.2387

3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.0000

Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000

* 1. specify the metrics you eventually need (make sure you include accuracy, sensitivity, specificity, and roc\_auc)

*# sens - out of all the positive classes how many were correctly predicted*

*#specs- out of all the negative classes how many were correctly predicted*

c.metrics<-yardstick::metric\_set(accuracy,

sens,

roc\_auc)

s.metrics<-yardstick::metric\_set(spec )

* 1. save the predicted value using model control grid

m\_control<-control\_grid(save\_pred=TRUE)

* 1. set the mode and engine for the tuning process. Please place your code in order and explain each segment using a hashtag if necessary. (To make sure that your results are reproducible, you need to set the random seeds consistently at each random step)

set.seed(123)

knn\_model<-nearest\_neighbor(neighbors= tune("K")) %>%

set\_mode("classification") %>%

set\_engine("kknn")

knn\_grid<- grid\_regular(parameters(knn\_model), levels=5)

1. Tune K in a KNN model and obtain plots for all mean metrics. Which metric(s) shows a clear pattern when K increases, which does not if any? What is the best K if you are to base your decision on the mean “accuracy” or “roc\_auc” alone?

set.seed(123)

knn.tune<-tune\_grid(

knn\_model,

model.rec,

resamples=k\_fold,

control=m\_control,

metrics=c.metrics)

set.seed(123)

knn.tune %>%

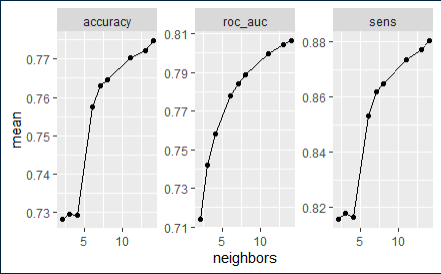
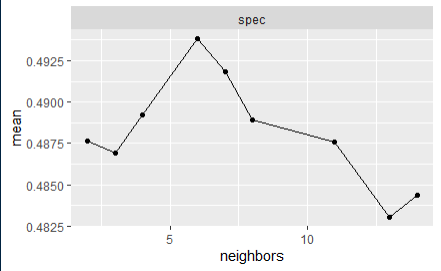
collect\_metrics() %>%

ggplot(aes(x=K,y=mean))+

geom\_point()+

geom\_line()+

facet\_wrap(~.metric, scales="free\_y")



set.seed(123)

knn.tune %>% select(id, .metrics) %>%

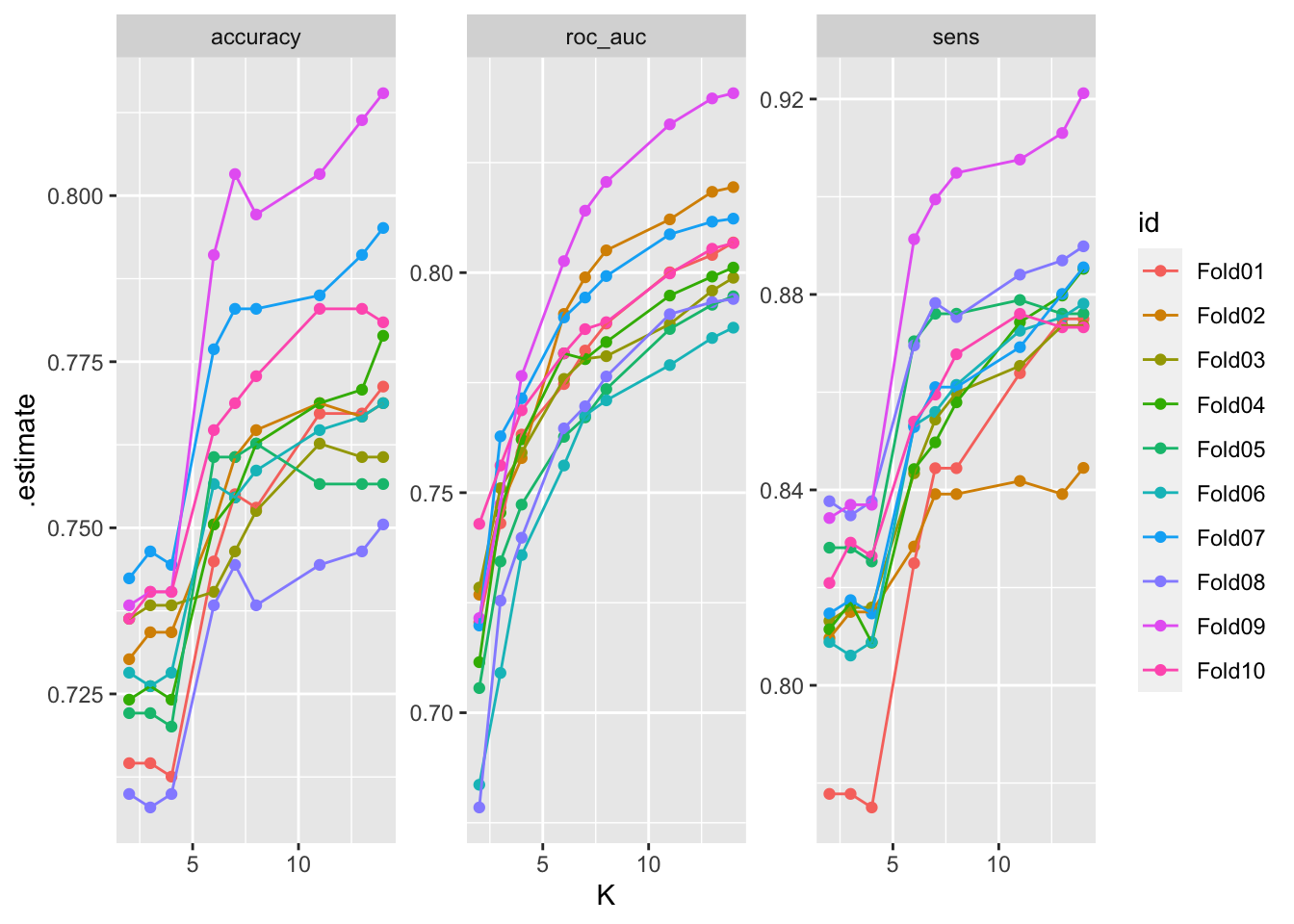
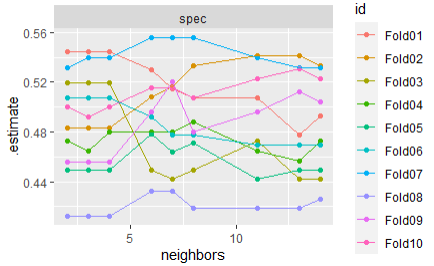
unnest(.metrics) %>%

ggplot(aes(x=K, y=.estimate, color=id))+

geom\_point()+

geom\_line()+

facet\_wrap(~.metric, scales="free\_y")



***Accuracy, roc\_auc and sens show a clear pattern when K increases these metrics also increase. If the best K was based solely on accuracy and roc\_auc I would pick the highest point which is k=14 or k=15. Specificity does not show a significant pattern.***

1. “Collect” the predicted values that have resulted from the tuning process and obtain a confusion matrix. What is the sample size according to the confusion matrix? Explain why it is much larger than the size of the training set.

pred.data<-knn.tune%>%

collect\_predictions() %>%

mutate(pred=if\_else(.pred\_No>=0.5, "No","Yes"),

pred=as.factor(pred))

pred.data %>%

conf\_mat(churn, pred)

Truth

**Prediction No Yes**

**No 27744 6046**

**Yes 4854 5735**

# ***The process used the training set sample size 10 times that is why the sample size is larger with 44379 obs. (49310\*.10=4931) (49310-4931=44379) 90 % of the training set which indicates the confusion matrix ran successfully.***

1. What are the mean accuracy and mean sensitivity that have resulted from the tuning process?

**pred.data %>% accuracy(churn, pred)**

**A tibble: 1 x 3**

**.metric .estimator .estimate**

**<chr> <chr> <dbl>**

**1 accuracy binary 0.754**

**pred.data %>% sens(churn, pred)**

**A tibble: 1 x 3**

**.metric .estimator .estimate**

**<chr> <chr> <dbl>**

**1 sens binary 0.851**

# ***Sens indicates that we will be correct 85% of the time when predicting positive cases in this dataset (yes churn).***

1. If you are to decide on the best K based on the mean “accuracy,” “sensitivity,” “specificity,” and “roc\_auc” respectively, how many different K’s are suggested? what is/are the best K? Which metric(s) seems to differ from the others in its recommendation? Justify your final decision on your choice of K.

knn.tune %>% collect\_metrics() %>% filter(.metric=="roc\_auc") %>% top\_n(mean, n=1)

K .metric .estimator mean n std\_err .config

1 14 roc\_auc binary 0.806 10 0.00485 Model9

knn.tune %>% collect\_metrics() %>% filter(.metric=="sens") %>% top\_n(mean, n=1)

K .metric .estimator mean n std\_err .config

1 14 sens binary 0.880 10 0.00600 Model9

knn.tune %>% collect\_metrics() %>% filter(.metric=="accuracy") %>% top\_n(mean, n=1)

K .metric .estimator mean n std\_err .config

1 14 accuracy binary 0.775 10 0.00608 Model9

# ***The accuracy, roc\_auc and sens all indicate that k=14 is the best decision.***

1. Use the best K you have obtained from the previous step and estimate the best model. What are the accuracy and roc\_auc of this model? Compared with the decision tree models you estimated for project 2, does the KNN model yield better or worse metrics?

model14<-nearest\_neighbor(neighbors=14) %>%

set\_mode("classification") %>%

set\_engine("kknn")

best\_model<-workflow() %>%

add\_model(model14)

add\_recipe(model.rec)

last\_fit(best\_model, subset.split) %>% collect\_metrics()

# ***.7783 was the accuracy in project 2 and .7727 was the accuracy in project 4. The metrics are similar using the knn model or decision tree model.***