MachineLearningAssignment2

#Install package if not already installed

```
#install.packages("caret")
#install.packages("lattice")
#install.packages("ggplot2")
#install.packages("fastDummies")
#install.packages("FNN")
#install.packages("e1071")
```

#load all the required libraries

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(readr)
library(fastDummies)
library(FNN)
library(gmodels)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

#Import the Universal Bank Dataset

```
UBank <- read.csv("UniversalBank.csv")
```

Remember to transform categorical predictors with more than two categories into dummy variables first ***

#Create Dummy variables for the column Education

```
UBank <- dummy_cols(UBank, select_columns = 'Education')
head(UBank)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard Education_1
## 1             0                  1          0      0          0           1
## 2             0                  1          0      0          0           1
## 3             0                  0          0      0          0           1
## 4             0                  0          0      0          0           0
## 5             0                  0          0      0          1           0
## 6             0                  0          0      1          0           0
##   Education_2 Education_3
## 1           0           0
## 2           0           0
## 3           0           0
## 4           1           0
## 5           1           0
## 6           1           0
```

#Remove ID,Zip Code & Education Columns and factor Personal loan column

```
UBank <- select(UBank, -ID, -ZIP.Code, -Education)
head(UBank)
```

```
##   Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account
## 1  25          1     49      4   1.6        0             0                  1
## 2  45         19     34      3   1.5        0             0                  1
## 3  39         15     11      1   1.0        0             0                  0
## 4  35          9    100      1   2.7        0             0                  0
## 5  35          8     45      4   1.0        0             0                  0
## 6  37         13     29      4   0.4      155             0                  0
##   CD.Account Online CreditCard Education_1 Education_2 Education_3
## 1          0      0          0           1           0           0
## 2          0      0          0           1           0           0
## 3          0      0          0           1           0           0
## 4          0      0          0           0           1           0
## 5          0      0          1           0           1           0
## 6          0      1          0           0           1           0
```

```
UBank$Personal.Loan <- factor(UBank$Personal.Loan)
```

#Partition the data into training (60%) and validation #(40%) sets

```
set.seed(123)
Train_Index=createDataPartition(UBank$Age, p=0.60, list=FALSE)
Train_Data = UBank[Train_Index,]
Validation_Data = UBank[-Train_Index,]
summary(Train_Data)
```

```
##       Age          Experience       Income           Family
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.0   Median : 64.00   Median :2.000
##  Mean   :45.31   Mean   :20.1   Mean   : 73.85   Mean   :2.395
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 99.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg           Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:2720        Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 281        1st Qu.:0.0000
##  Median : 1.500   Median :  0.00                 Median :0.0000
##  Mean   : 1.934   Mean   : 55.83                 Mean   :0.1026
##  3rd Qu.: 2.500   3rd Qu.:100.00                 3rd Qu.:0.0000
##  Max.   :10.000   Max.   :617.00                 Max.   :1.0000
##    CD.Account          Online         CreditCard      Education_1
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.05831   Mean   :0.5928   Mean   :0.2889   Mean   :0.4215
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   Education_2      Education_3
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2789   Mean   :0.2996
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
summary(Validation_Data)
```

```
##       Age          Experience        Income            Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 38.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 63.00   Median :2.000
##  Mean   :45.38   Mean   :20.11   Mean   : 73.66   Mean   :2.398
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :218.00   Max.   :4.000
##      CCAvg           Mortgage      Personal.Loan Securities.Account
##  Min.   : 0.000   Min.   :  0.00   0:1800        Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1: 199        1st Qu.:0.0000
##  Median : 1.500   Median :  0.00                 Median :0.0000
##  Mean   : 1.944   Mean   : 57.51                 Mean   :0.1071
##  3rd Qu.: 2.600   3rd Qu.:102.50                 3rd Qu.:0.0000
##  Max.   :10.000   Max.   :635.00                 Max.   :1.0000
##    CD.Account          Online         CreditCard      Education_1
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.06353   Mean   :0.6028   Mean   :0.3017   Mean   :0.4157
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   Education_2      Education_3
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2831   Mean   :0.3012
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

#Normalize the dataset and remove personal loan column

```
#Copy the original data and remove personal loan column
Train.norm.df <- Train_Data[,-7]
Valid.norm.df <- Validation_Data[,-7]
#Normalize data
norm.values <- preProcess(Train.norm.df, method=c("center", "scale"))
#Replace with the normalized data
Train.norm.df <- predict(norm.values, Train.norm.df)
Valid.norm.df <- predict(norm.values, Valid.norm.df)
```

Consider the following customer:Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1, and Credit Card = 1. Perform a k-NN classification with all predictors except ID and ZIP code using k = 1 ***

#Classify the customer with k=1

```
new.data1 = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,  Mortgage = 0, Securities.A
ccount = 0 , CD.Account = 0 ,Online = 1, CreditCard = 1, Education_1 = 0, Education_2 = 1, Education_3 = 0)
#Replace with normalized data
new.data1 <- predict(norm.values, new.data1)
#KNN Modeling
nn2 <- knn(train = Train.norm.df, test = new.data1, cl = Train_Data$Personal.Loan, k=1, prob=TRUE)
nn2
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1]
## [1,] 2655
## attr(,"nn.dist")
##           [,1]
## [1,] 0.4975307
## Levels: 0
```

Customer is classified as a 0. ***

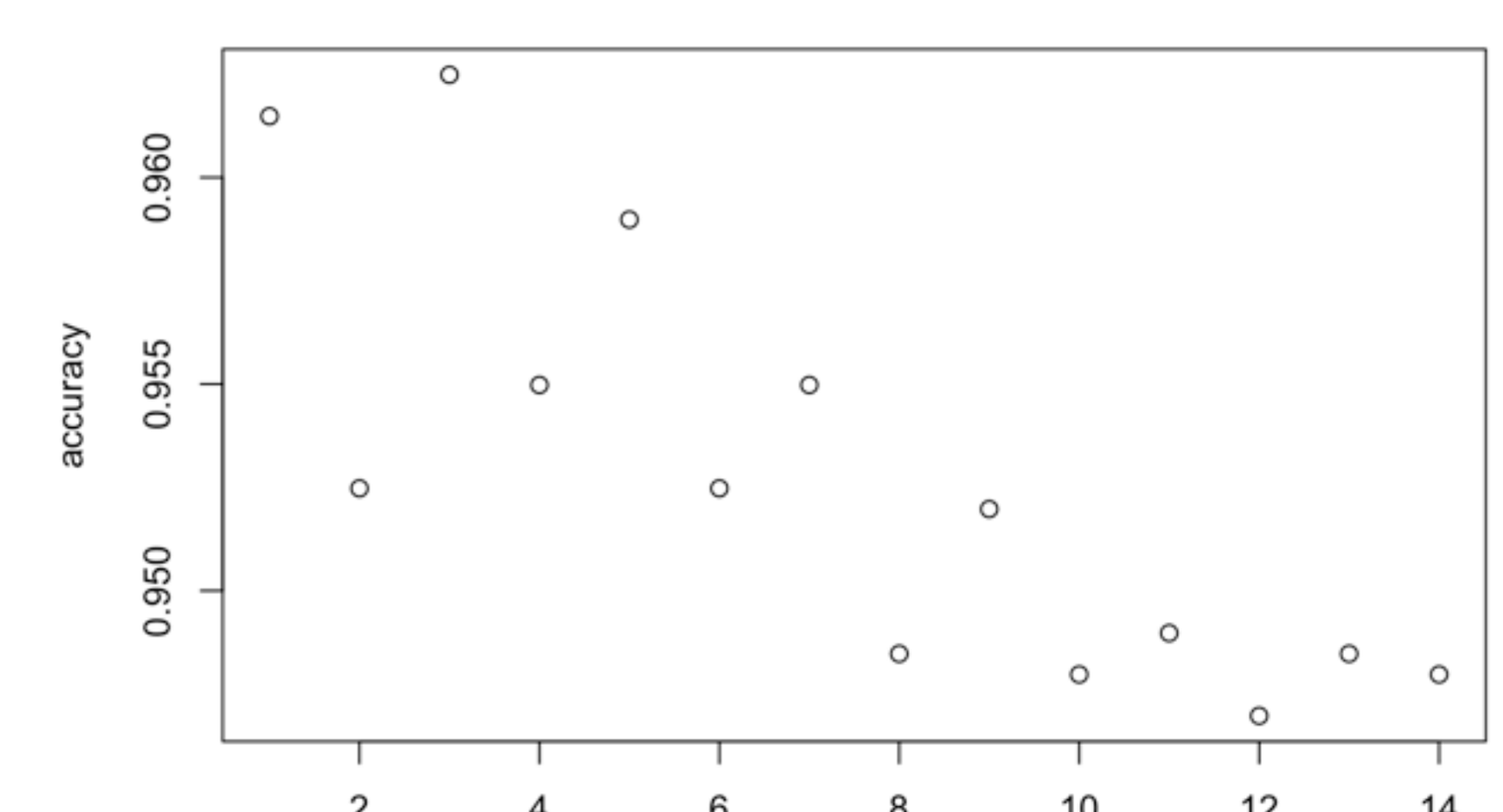
What is a choice of k that balances between overfitting and ignoring the predictor information ***

#Perform Accuracy

```
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
for(i in 1:14) {
  knn.pred <- knn(Train.norm.df, Valid.norm.df, cl = Train_Data$Personal.Loan, k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.pred,Validation_Data$Personal.Loan)$overall[1]
}
accuracy.df
```

```
##     k  accuracy
## 1   1 0.9614807
## 2   2 0.9524762
## 3   3 0.9624812
## 4   4 0.9549775
## 5   5 0.9589795
## 6   6 0.9524762
## 7   7 0.9549775
## 8   8 0.9484742
## 9   9 0.9519760
## 10 10 0.9479740
## 11 11 0.9489745
## 12 12 0.9469735
## 13 13 0.9484742
## 14 14 0.9479740
```

```
plot(accuracy.df)
```



---

the best Choice of K is 5. ***

Show the confusion matrix for the validation data that results from using the best k and explain different error types that you observe. ***

#Perform k-NN classification for validation set with k=5

```
nn <- knn(train = Train.norm.df, test = Valid.norm.df , cl = Train_Data$Personal.Loan, k=5, prob=TRUE)
knn.attributes <- attributes(nn)
```

#Show the confusion matrix for the validation data

```
confusionMatrix(nn, Validation_Data$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1790   72
##          1   10  127
##
##                Accuracy : 0.959
##                  95% CI : (0.9493, 0.9672)
##     No Information Rate : 0.9005
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7344
##
##  Mcnemar's Test P-Value : 1.624e-11
##
##             Sensitivity : 0.9944
##             Specificity : 0.6382
##          Pos Pred Value : 0.9613
##          Neg Pred Value : 0.9270
##              Prevalence : 0.9005
##          Detection Rate : 0.8954
##    Detection Prevalence : 0.9315
##       Balanced Accuracy : 0.8163
##
##        'Positive' Class : 0
##
```

Error Type I is False positives which is 72 in this case Error Type II is False negatives which is 10 in this case ***

Consider the following customer: Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1 and Credit Card = 1. Classify the customer using the best k. ***

Classify the cutomer with k=5 with training set. Also, Classify the customer with k=5 with combined training and validation set

```
Traval_data <- rbind(Train_Data,Validation_Data)
Traval_norm <- Traval_data[,-7]
#Normalize data
norm.values1 <- preProcess(Traval_norm, method=c("center", "scale"))
#Replace with normalized data
Traval_norm <- predict(norm.values1, Traval_norm)
#Consider the customer
new.data = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,  Mortgage = 0, Securities.Ac
count = 0 , CD.Account = 0 ,Online = 1, CreditCard = 1, Education_1 = 0, Education_2 = 1, Education_3 = 0)
#Normalize the test data
new.data <- predict(norm.values1, new.data)
#Classify the customer with combined Training and Validation data set
nn1 <- knn(train = Traval_norm, test = new.data, cl = Traval_data$Personal.Loan , k=5, prob=TRUE)
nn1
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 4603 2655 4345 3653 2483
## attr(,"nn.dist")
##           [,1]     [,2]      [,3]    [,4]      [,5]
## [1,] 0.4787594 0.496127 0.6343843 0.70673 0.8362888
## Levels: 0
```

```
#Also, Classify the customer with only Training data set
nn3 <- knn(train = Train.norm.df, test = new.data1, cl = Train_Data$Personal.Loan, k=5, prob=TRUE)
nn3
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 2655 2483 2573 1624 2052
## attr(,"nn.dist")
##           [,1]      [,2]      [,3]     [,4]     [,5]
## [1,] 0.4975307 0.8388543 0.9760344 1.035198 1.114043
## Levels: 0
```

With both the data sets (training and Combined training and validation), Customer is classified as 0. ***

Re-partition the data, this time into training, validation, and test sets (50% : 30% : 20%). Apply the k-NN method with the k chosen above ***

#Repartition the data and apply knn method

```
head(UBank)
```

```
##   Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account
## 1  25          1     49      4   1.6        0             0                  1
## 2  45         19     34      3   1.5        0             0                  1
## 3  39         15     11      1   1.0        0             0                  0
## 4  35          9    100      1   2.7        0             0                  0
## 5  35          8     45      4   1.0        0             0                  0
## 6  37         13     29      4   0.4      155             0                  0
##   CD.Account Online CreditCard Education_1 Education_2 Education_3
## 1          0      0          0           1           0           0
## 2          0      0          0           1           0           0
## 3          0      0          0           1           0           0
## 4          0      0          0           0           1           0
## 5          0      0          1           0           1           0
## 6          0      1          0           0           1           0
```

```
set.seed(123)
Train_Index1=createDataPartition(UBank$Age, p=0.50, list=FALSE)
Train_Data1 = UBank[Train_Index1,]
Validation_Index1 = createDataPartition(UBank$Age, p=0.30, list=FALSE)
Validation_Data1 = UBank[Validation_Index1,]
Test_index1 = createDataPartition(UBank$Age, p = 0.2,list = FALSE)
Test_data1 = UBank[Test_index1,]
#Remove the personal loan column
Train_norm1 <- Train_Data1[,-7]
Validation_norm1 <- Validation_Data1[,-7]
Test_norm1 <-Test_data1[,-7]
#Normalize the Re-partitioned data
norm.values2 <- preProcess(Train_norm1,method=c("center", "scale"))
Train_norm1<- predict(norm.values2,Train_norm1)
Validation_norm1<- predict(norm.values2,Validation_norm1)
Test_norm1<- predict(norm.values2,Test_norm1)
# KNN Modeling on training set
knnTrain <- knn(train= Train_norm1, test= Train_norm1, cl=Train_Data1$Personal.Loan, k=5, prob = TRUE)
# KNN Modeling on Validation set
knnValid <- knn(train= Train_norm1, test= Validation_norm1, cl=Train_Data1$Personal.Loan, k=5, prob = TRUE)
# KNN Modeling on Test set
knnTest <- knn(train= Train_norm1, test= Test_norm1, cl=Train_Data1$Personal.Loan, k=5, prob = TRUE)
#Combine the Training and Validation set and normalize it
Traval_data1 <- rbind(Train_Data1,Validation_Data1)
Traval_norm1 <- Traval_data1[,-7]
Test_norm2 <-Test_data1[,-7]
norm.values3 <- preProcess(Traval_norm1, method=c("center", "scale"))
Traval_norm1<- predict(norm.values3,Traval_norm1)
Test_norm2<- predict(norm.values3,Test_norm2)
#KNN modeling on test set with the combined training and Validation set
knnTest1 <- knn(train= Traval_norm1, test= Test_norm2, cl=Traval_data1$Personal.Loan, k=5, prob = TRUE)
```

Compare the confusion matrix of the test set with that of the training and validation sets ***

#Show Confusion Matrix for Training data set

```
confusionMatrix(knnTrain, Train_Data1$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2271   79
##          1    6  145
##
##                Accuracy : 0.966
##                  95% CI : (0.9581, 0.9728)
##     No Information Rate : 0.9104
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7557
##
##  Mcnemar's Test P-Value : 5.742e-15
##
##             Sensitivity : 0.9974
##             Specificity : 0.6473
##          Pos Pred Value : 0.9664
##          Neg Pred Value : 0.9603
##              Prevalence : 0.9104
##          Detection Rate : 0.9080
##    Detection Prevalence : 0.9396
##       Balanced Accuracy : 0.8223
##
##        'Positive' Class : 0
##
```

#Show Confusion Matrix for Validation data set

```
confusionMatrix(knnValid, Validation_Data1$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1340   62
##          1    7   93
##
##                Accuracy : 0.9541
##                  95% CI : (0.9422, 0.9641)
##     No Information Rate : 0.8968
##     P-Value [Acc > NIR] : 4.554e-16
##
##                   Kappa : 0.7056
##
##  Mcnemar's Test P-Value : 7.987e-11
##
##             Sensitivity : 0.9948
##             Specificity : 0.6000
##          Pos Pred Value : 0.9558
##          Neg Pred Value : 0.9300
##              Prevalence : 0.8968
##          Detection Rate : 0.8921
##    Detection Prevalence : 0.9334
##       Balanced Accuracy : 0.7974
##
##        'Positive' Class : 0
##
```

#Show Confusion Matrix for Test data set

```
confusionMatrix(knnTest, Test_data1$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 896  44
##          1   0  61
##
##                Accuracy : 0.956
##                  95% CI : (0.9414, 0.9679)
##     No Information Rate : 0.8951
##     P-Value [Acc > NIR] : 1.764e-12
##
##                   Kappa : 0.7128
##
##  Mcnemar's Test P-Value : 9.022e-11
##
##             Sensitivity : 1.0000
##             Specificity : 0.5810
##          Pos Pred Value : 0.9532
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8951
##          Detection Rate : 0.8951
##    Detection Prevalence : 0.9391
##       Balanced Accuracy : 0.7905
##
##        'Positive' Class : 0
##
```

#Show Confusion Matrix for Test data set with combined training and validation data set

```
confusionMatrix(knnTest1, Test_data1$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 895  35
##          1   1  70
##
##                Accuracy : 0.964
##                  95% CI : (0.9506, 0.9747)
##     No Information Rate : 0.8951
##     P-Value [Acc > NIR] : 4.341e-16
##
##                   Kappa : 0.7765
##
##  Mcnemar's Test P-Value : 3.798e-08
##
##             Sensitivity : 0.9989
##             Specificity : 0.6667
##          Pos Pred Value : 0.9624
##          Neg Pred Value : 0.9859
##              Prevalence : 0.8951
##          Detection Rate : 0.8941
##    Detection Prevalence : 0.9291
##       Balanced Accuracy : 0.8328
##
##        'Positive' Class : 0
##
```

Training data set Accuracy = 0.966 Validation data Set Accuracy =0.9541 Test data Set Accuracy =0.956 The classifications are most accurate on the training data set and least accurate on the validation data set. ***