

assignment2

sai prasad desineni

2/22/2022

“{r setup}

feeding in the required libraries

```
library("caret") library("dplyr") library("class") universalbank = read.csv("C:/Users/desineni/Downloads/UniversalBank.csv")
View(universalbank) universalbank$ZIP.Code = NULL universalbank$ID = NULL summary(universalbank)
```

```
#normalization of data universalbank$Personal.Loan = as.factor(universalbank$Personal.Loan)
Model_range_normalized = preProcess(universalbank, method = "range") universalbank_norm = predict(Model_range_normalized, universalbank)
View(universalbank_norm) summary(universalbank_norm)
```

```
#splitting of data into train, test and validation sets
```

```
Train_index = createDataPartition(universalbank$Personal.Loan, p = 0.6, list = FALSE) train.df = universalbank_norm[Train_index,]
validation.df = universalbank_norm[-Train_index,] print(validation.df)
```

```
#1- task1 prediction of k value
```

```
To_Predict = data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education = 1, Mortgage = 0, Securities.Account = 0,
CD.Account = 0, Online = 1, CreditCard = 1) print(To_Predict) To_Predict_norm <- predict(Model_range_normalized, To_Predict) Prediction
<- knn(train = train.df[,1:7], test = To_Predict_norm[,1:7], cl = train.df$Personal.Loan, k = 1) print(Prediction)
```

```
#2- task2 k value for the balance of overfitting and underfitting
```

```
set.seed(123) Bankcontrol <- trainControl(method = "repeatedcv", number = 3, repeats = 2) searchGrid = expand.grid(k=1:10)
knn.model = train(Personal.Loan~., data = train.df, method = 'knn', tuneGrid = searchGrid, trControl = Bankcontrol) knn.model
```

```
#3- task3 confusion matrix for the validation
```

```
predictions <- predict(knn.model, validation.df) confusionMatrix(predictions, validation.df$Personal.Loan)
```

```
#4- task4
```

```
To_Predict_norm = data.frame(Age = 40, Experience = 10, Income = 84, family = 2, CCAvg = 2, Education = 1, Mortgage = 0, Securities.Account = 0,
CD.Account = 0, Online = 1, CreditCard = 1) To_Predict_norm = predict(Model_range_normalized, To_Predict) predict(knn.model, To_Predict_norm)
```

```
#5- task5 train_size = 0.5 Train_index = createDataPartition(universalbank$Personal.Loan, p = 0.5, list = FALSE) Train.df = universalbank_norm[Train_index,]
```

```
test_size = 0.2 Test_index = createDataPartition(universalbank$Personal.Loan, p = 0.2, list = FALSE) test.df = universalbank_norm[Test_index,]
```

```

valid_size = 0.3 validation_index = createDataPartition(universalbank$Personal.Loan, p = 0.3, list =
FALSE) validation.df = universalbank_norm[validation_index,]

Trainknn = knn(train=train.df[,8], test = train.df[,8], cl = train.df[,8], k =3) Testknn <- knn(train =
train.df[,8], test = test.df[,8], cl = train.df[,8], k =3) Validationknn <- knn(train = train.df[,8], test =
validation.df[,8], cl = train.df[,8], k =3)

confusionMatrix(Trainknn, train.df[,8]) confusionMatrix(Testknn, test.df[,8])
confusionMatrix(Validationknn, validation.df[,8])

#comment - the accuracy of test set and the train set are almost same.which means the model is performing
really well,it's on point.This is a good sign ,because this avoids overfitting.
““

```