R Notebook

Loading Libraries

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## 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
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

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

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.2
```

```
library(caTools)
library(fastDummies)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
library(caTools)
library(class)
```

```
## Warning: package 'class' was built under R version 4.1.2
```

Importing Data

```
df_train <- read.csv("/Users/alishakhan/Desktop/School/FALL22/CSP571/project/NEW_DATASETS/train/joint_nu
                     stringsAsFactors = FALSE, sep = ",")

df_test <- read.csv("/Users/alishakhan/Desktop/School/FALL22/CSP571/project/NEW_DATASETS/test/joint_test
                     stringsAsFactors = FALSE, sep = ",")

df_test[is.na(df_test)] = 0

df_train$X<-NULL
df_test$X<-NULL
df_train$fraudulent<-as.factor(df_train$fraudulent)
df_test$fraudulent<-as.factor(df_test$fraudulent)
df_train$department_n_first_personp<-NULL
df_test$dep_oil<-NULL
#df_test$department_n_first_personp<-NULL
#colnames(df_train)[colSums(is.na(df_train)) > 0]
#colnames(df_test)[colSums(is.na(df_test)) > 0]

#all_equal(df_train, df_test)
#colnames(df_train)[!(colnames(df_train) %in% colnames(df_test))]
#colnames(df_test)[!(colnames(df_test) %in% colnames(df_train))]
```

Naive Bayes Classifier

```
set.seed(1)
nb_classifier<-naiveBayes(fraudulent~.,data=df_train)
y_pred<-predict(nb_classifier,newdata=df_test)
cm<-table(df_test$fraudulent,y_pred)
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
##
##    y_pred
##        0    1
##   0  715 2695
##   1    8  158
##
##                Accuracy : 0.2441
##                  95% CI : (0.2301, 0.2586)
##     No Information Rate : 0.7978
```

```
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0186
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.98893
##             Specificity : 0.05538
##          Pos Pred Value : 0.20968
##          Neg Pred Value : 0.95181
##              Prevalence : 0.20218
##          Detection Rate : 0.19994
##    Detection Prevalence : 0.95358
##       Balanced Accuracy : 0.52216
##
##        'Positive' Class : 0
##
```

Support Vector Machine Classifier

```
svm_classifier=svm(formula=fraudulent~.,data=df_train, type='C-classification',kernel='linear')
y_pred=predict(svm_classifier,newdata=df_test)
cm=table(df_test$fraudulent,y_pred)
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
##
##    y_pred
##        0    1
##   0 3389   21
##   1  110   56
##
##                Accuracy : 0.9634
##                  95% CI : (0.9567, 0.9693)
##     No Information Rate : 0.9785
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4446
##
##  Mcnemar's Test P-Value : 1.488e-14
##
##             Sensitivity : 0.9686
##             Specificity : 0.7273
##          Pos Pred Value : 0.9938
##          Neg Pred Value : 0.3373
##              Prevalence : 0.9785
##          Detection Rate : 0.9477
##    Detection Prevalence : 0.9536
##       Balanced Accuracy : 0.8479
##
##        'Positive' Class : 0
##
```

KNN

```
# Feature Scaling
train_scale <- scale(select(df_train,-c(fraudulent)))
test_scale <- scale(select(df_test,-c(fraudulent)))

#View(train_scale)
#View(test_scale)

classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = df_train$fraudulent,
                      k = 3)

#classifier_knn
# Confusiin Matrix
cm <- table(df_test$fraudulent, classifier_knn)
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
##
##    classifier_knn
##        0    1
##   0 3375   35
##   1   48  118
##
##                Accuracy : 0.9768
##                  95% CI : (0.9713, 0.9815)
##     No Information Rate : 0.9572
##     P-Value [Acc > NIR] : 2.02e-10
##
##                   Kappa : 0.7277
##
##  Mcnemar's Test P-Value : 0.1878
##
##             Sensitivity : 0.9860
##             Specificity : 0.7712
##          Pos Pred Value : 0.9897
##          Neg Pred Value : 0.7108
##              Prevalence : 0.9572
##          Detection Rate : 0.9438
##    Detection Prevalence : 0.9536
##       Balanced Accuracy : 0.8786
##
##        'Positive' Class : 0
##
```