KNN

```
# Load the caret library
library(caret)

# Train the KNN Model
k <- 5 # Choose the number of neighbors (you can tune this parameter)
model <- train(x = X_train, y = y_train, method = "knn", trControl =
trainControl(method = "none"), tuneGrid = expand.grid(k = k))

# Predict using the trained model
predictions <- predict(model, newdata = X_test)

# Evaluate the Model
confusion_matrix <- confusionMatrix(data = predictions, reference =
y_test)
print(confusion_matrix)
```

SVM

```
# Train the SVM Model
svm_model <- svm(Class ~ ., data = data.frame(X_train, Class = y_train),
kernel = "radial")

# Make predictions
svm_pred <- predict(svm_model, X_test)

# Evaluate the Model
confusion_matrix <- confusionMatrix(data = svm_pred, reference = y_test)
print(confusion_matrix)
```

SVM WITH PCA

```
# Perform PCA
pca <- prcomp(X_train, scale. = TRUE)
X_train_pca <- predict(pca, X_train)
```

```
X_test_pca <- predict(pca, X_test)

# Train and Evaluate SVM Model with PCA
svm_model_pca <- svm(Class ~ ., data = data.frame(X_train_pca, Class =
y_train))
svm_pred_pca <- predict(svm_model_pca, X_test_pca)

# Evaluate the Model
confusion_matrix <- confusionMatrix(data = svm_pred_pca, reference =
y_test)
print(confusion_matrix)
```

NAIVE BAYES

```
# Load the required library for Naive Bayes
library(e1071)

# Train the Naive Bayes Model
nb_model <- naiveBayes(Class ~ ., data = data.frame(X_train, Class =
y_train))

# Make predictions
nb_pred <- predict(nb_model, X_test)

# Evaluate the Model
confusion_matrix <- confusionMatrix(data = nb_pred, reference = y_test)
print(confusion_matrix)
```

RANDOM FOREST

```
install.packages("randomForest")
# Load the required library for Random Forest
library(randomForest)

# Train the Random Forest Model
```

```
rf_model <- randomForest(Class ~ ., data = data.frame(X_train, Class =
y_train))

# Make predictions
rf_pred <- predict(rf_model, X_test)

# Evaluate the Model
confusion_matrix <- confusionMatrix(data = rf_pred, reference = y_test)
print(confusion_matrix)
```