Short Course: KNN Classification in R

1. Introduction to KNN

- **Key idea:** Classify a new data point based on the majority class among its *k* closest neighbors.

2. Setting Up the Environment

```
# Load necessary libraries
install.packages("class")
library(class)
install.packages("datasets")
library(datasets)
```

3. Load and Prepare Data

```
# Example dataset: iris
data(iris)

# Make sure data is scaled (standardized)
iris_scaled <- scale(iris[, -5])  # all features except species
```

4. Implementing KNN

```
# Split data into training and test sets
set.seed(123)
sample_index <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris_scaled[sample_index, ]
test_data <- iris_scaled[-sample_index, ]
train_labels <- iris$Species[sample_index]
test_labels <- iris$Species[-sample_index]

# Run KNN
k <- 3  # choose k
predicted_labels <- knn(train = train_data, test = test_data, cl = train_labels,
```

```
k = k)
```

5. Evaluating Performance

```
# Confusion Matrix
table(Predicted = predicted_labels, Actual = test_labels)

# Accuracy
mean(predicted_labels == test_labels)
```

6. Choosing the Best k

```
# Loop over different k values
k_values <- 1:15
accuracy <- sapply(k_values, function(k) {
  pred <- knn(train = train_data, test = test_data, cl = train_labels, k = k)
  mean(pred == test_labels)
})

# Plot accuracy vs. k
plot(k_values, accuracy, type = "b", xlab = "k", ylab = "Accuracy")
best_k <- k_values[which.max(accuracy)]
```

Example: KNN Classification on a Subset of MNIST-like Data

1. Load or create data

For simplicity, we'll generate synthetic data resembling handwritten digits with two features for visualization ease.

```
# Generate synthetic data
set.seed(123)

# Class 0: random points around (2, 2)
```

```r
class0 <- matrix(rnorm(50*2, mean=2, sd=0.5), ncol=2)
labels0 <- rep(0, 50)

# Class 1: random points around (4, 4)
class1 <- matrix(rnorm(50*2, mean=4, sd=0.5), ncol=2)
labels1 <- rep(1, 50)

# Combine the data
data <- rbind(class0, class1)
labels <- factor(c(labels0, labels1))
```

2. Split into training and testing sets

```r
train_idx <- sample(1:100, 70)
test_idx <- setdiff(1:100, train_idx)

train_data <- data[train_idx, ]
train_labels <- labels[train_idx]

test_data <- data[test_idx, ]
test_labels <- labels[test_idx]
```

3. Run KNN

```r
library(class)

k <- 3
predictions <- knn(train = train_data, test = test_data, cl = train_labels, k = k)
```

4. Visualize Results

```r
# Plot train data
plot(train_data, col=as.numeric(train_labels)+1, pch=19, main='Training Data')
points(test_data, col=as.numeric(predictions)+1, pch=17, cex=1.5)
legend("topright", legend=levels(labels), col=1:2, pch=19)
```

5. Evaluate Accuracy

```
table(Predicted=predictions, Actual=test_labels)
mean(predictions == test_labels)
```