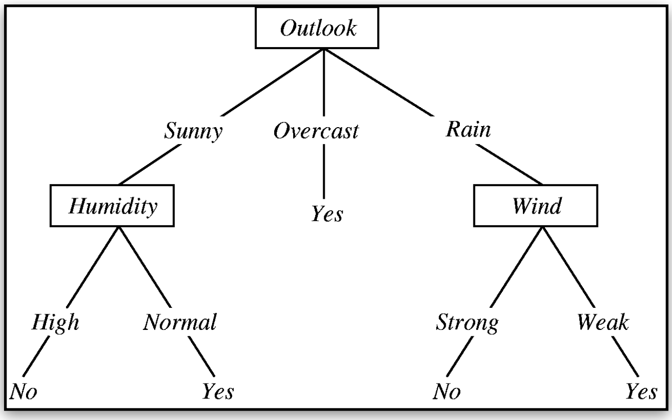
Lab Assignment-8

Write a program to classify in R with a suitable Dataset.
a) Decision Tree
b) Naïve Bayes
c) KNN
d) SVM

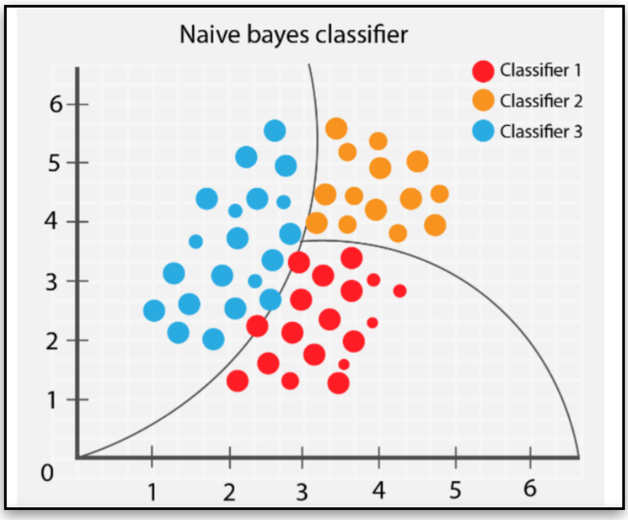
a) Decision Tree:

- Supervised learning algorithm for classification and regression tasks. Creates a tree-like model of decisions by partitioning data into subsets based on features.
- Selects the best feature to split the data at each node. Continues recursively until a stopping criterion is met.
- Uses criteria like Gini impurity and Information Gain (entropy) to decide the best split.
- Easy to understand and interpret. Handles both numerical and categorical data. Requires minimal data preprocessing.
- Prone to overfitting. Sensitive to small variations in the data.



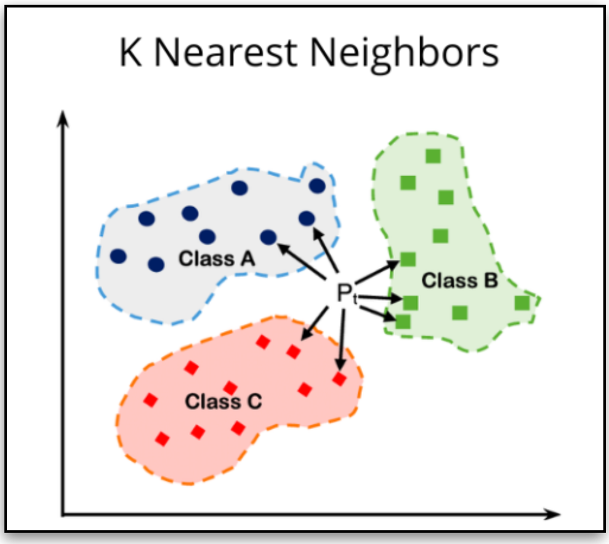
b) Naive Bayes:

- Probabilistic classification algorithm based on Bayes' theorem. Assumes independence between features.
- Calculates the posterior probability of each class based on prior probability and likelihood of features.
- Despite its "naive" assumption, it often performs well in practice, especially in text classification and spam filtering.
- Simple and easy to implement. Computationally efficient. Works well with high-dimensional data.
- Relies on the assumption of feature independence. Sensitive to the quality of training data.



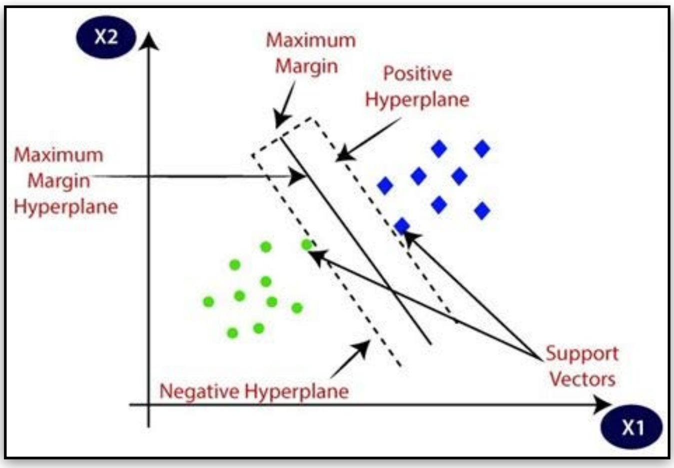
c) KNN (K-Nearest Neighbors):

- Non-parametric lazy learning algorithm for classification and regression tasks. Classifies objects based on majority vote of k nearest neighbors.
- For classification, assigns class label based on majority class among k nearest neighbors. For regression, output is average of target values of k nearest neighbors.
- Value of k must be chosen, affecting model's performance.
- Simple and easy to understand. No training phase (lazy learning). Works well with small datasets.
- Computationally expensive during testing phase. Sensitive to irrelevant features and choice of distance metric.



d) SVM (Support Vector Machine):

- Supervised learning algorithm for classification and regression tasks. Finds hyperplane that best separates classes in high-dimensional feature space.
- Aims to maximize margin between classes, distance between hyperplane and nearest data points (support vectors).
- Efficiently handles non-linearly separable data by mapping features into higher-dimensional space using kernel functions.
- Includes regularization parameter (C) to control trade-off between maximizing margin and minimizing classification errors.
- Effective in high-dimensional spaces. Memory efficient. Versatile due to kernel trick. Effective even with small datasets.
- Requires careful selection of kernel and regularization parameters. Can be computationally expensive for large datasets. Difficult to interpret model's decision boundary in high-dimensional spaces.



Code:

```r
# Install necessary packages
if (!requireNamespace("caret", quietly = TRUE)) install.packages("caret")
if (!requireNamespace("e1071", quietly = TRUE)) install.packages("e1071")
if (!requireNamespace("class", quietly = TRUE)) install.packages("class")
if (!requireNamespace("rpart", quietly = TRUE)) install.packages("rpart")

# Load libraries
library(caret)
library(e1071)
library(class)
library(rpart)

# Load the Iris dataset
data(iris)
set.seed(123)  # For reproducibility

# Splitting data into training and testing sets
indexes <- createDataPartition(iris$Species, p=0.75, list=FALSE)
train_data <- iris[indexes,]
test_data <- iris[-indexes,]

# Decision Tree model
model_dt <- rpart(Species ~ ., data = train_data, method = "class")
predictions_dt <- predict(model_dt, test_data, type = "class")

# Evaluate Decision Tree model
confusionMatrix(predictions_dt, test_data$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         12         1
  virginica       0          0        11

Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.864e-16

                  Kappa : 0.9583

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9167
Specificity                 1.0000            0.9583           1.0000
Pos Pred Value              1.0000            0.9231           1.0000
Neg Pred Value              1.0000            1.0000           0.9600
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3056
Detection Prevalence        0.3333            0.3611           0.3056
Balanced Accuracy           1.0000            0.9792           0.9583
```

 # Naive Bayes model

model_nb <- naiveBayes(Species ~ ., data = train_data)
predictions_nb <- predict(model_nb, test_data)

# Evaluate Naive Bayes model
confusionMatrix(predictions_nb, test_data$Species)

# Find the best k
set.seed(123)
k_values <- train(Species~., data = train_data, method = "knn", tuneGrid = expand.grid(k = 1:20),
trControl = trainControl(method = "cv", number = 10))

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         12         1
  virginica       0          0        11

Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.864e-16

                  Kappa : 0.9583

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9167
Specificity                 1.0000            0.9583           1.0000
Pos Pred Value              1.0000            0.9231           1.0000
Neg Pred Value              1.0000            1.0000           0.9600
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3056
Detection Prevalence        0.3333            0.3611           0.3056
Balanced Accuracy           1.0000            0.9792           0.9583
```

 # KNN model

best_k <- k_values$bestTune$k
model_knn <- knn(train = train_data[, -5], test = test_data[, -5], cl = train_data$Species, k = best_k)

# Evaluate KNN model
confusionMatrix(model_knn, test_data$Species)

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         11         0
  virginica       0          1        12

Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.864e-16

                  Kappa : 0.9583

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9167           1.0000
Specificity                 1.0000            1.0000           0.9583
Pos Pred Value              1.0000            1.0000           0.9231
Neg Pred Value              1.0000            0.9600           1.0000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3056           0.3333
Detection Prevalence        0.3333            0.3056           0.3611
Balanced Accuracy           1.0000            0.9583           0.9792
```

 # SVM model
model_svm <- svm(Species ~ ., data = train_data, method = "C-classification", kernel = "radial")
predictions_svm <- predict(model_svm, test_data)

# Evaluate SVM model
confusionMatrix(predictions_svm, test_data$Species)

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         11         1
  virginica       0          1        11

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.8134, 0.9932)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.728e-14

                  Kappa : 0.9167

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9167           0.9167
Specificity                 1.0000            0.9583           0.9583
Pos Pred Value              1.0000            0.9167           0.9167
Neg Pred Value              1.0000            0.9583           0.9583
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3056           0.3056
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.9375           0.9375
```

Environment Variables:

Data	
16L _ndexes	int [1:114, 1] 3 4 5 7 8 9 10 11 12 13 ...
iris	150 obs. of 5 variables
k_values	List of 24
model_dt	List of 14
model_nb	List of 5
model_svm	List of 31
test_data	36 obs. of 5 variables
train_data	114 obs. of 5 variables
Values	
best_k	16L
model_knn	Factor w/ 3 levels "setosa","versicolor",...
predictions_dt	Factor w/ 3 levels "setosa","versicolor",...
predictions_nb	Factor w/ 3 levels "setosa","versicolor",...
predictions_svm	Factor w/ 3 levels "setosa","versicolor",...