Part 1:

Supervised Learning Model: KNN for classification
The k-Nearest Neighbors (KNN) classifier model has several advantages and disadvantages:

Advantages:

1. Simplicity: KNN is easy to understand and implement, requiring no training phase.
2. Non-parametric: It makes no assumptions about the underlying data distribution, which is useful with real-world data.
3. Versatility: It can be used for both classification and regression tasks.
4. Adaptability: KNN can easily take up new data in real-time.

Disadvantages:

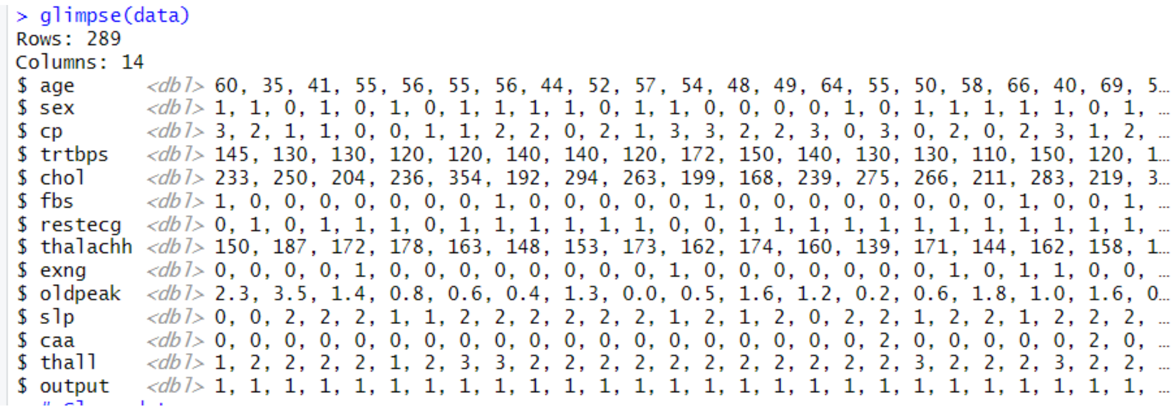
1. Scalability: It becomes significantly slower as the volume of data increases because it searches for nearest neighbors.
2. Curse of Dimensionality: KNN performs poorly with high-dimensional data due to the increase in distance computation.
3. Sensitive to Noisy Data: Irrelevant features or unscaled data can disrupt KNN's ability to find nearest neighbors accurately.
4. Computationally Intensive: Requires storing the entire dataset and calculating distances for each query, which can be resource-intensive.

Dataset: heart.csv

Link: https://figshare.com/articles/dataset/heart_csv/20236848

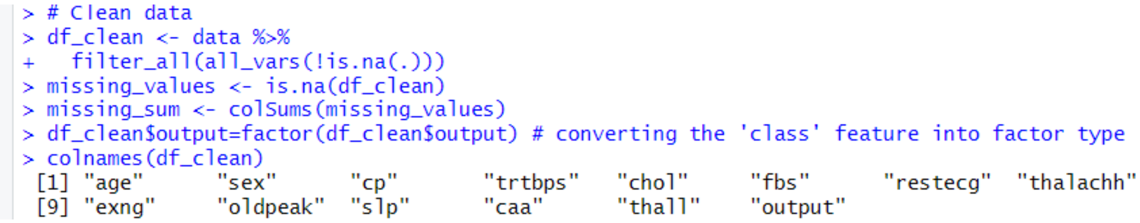
Author: Neha Nandal

I will be using KNN model to predict patients who will be having heart problems

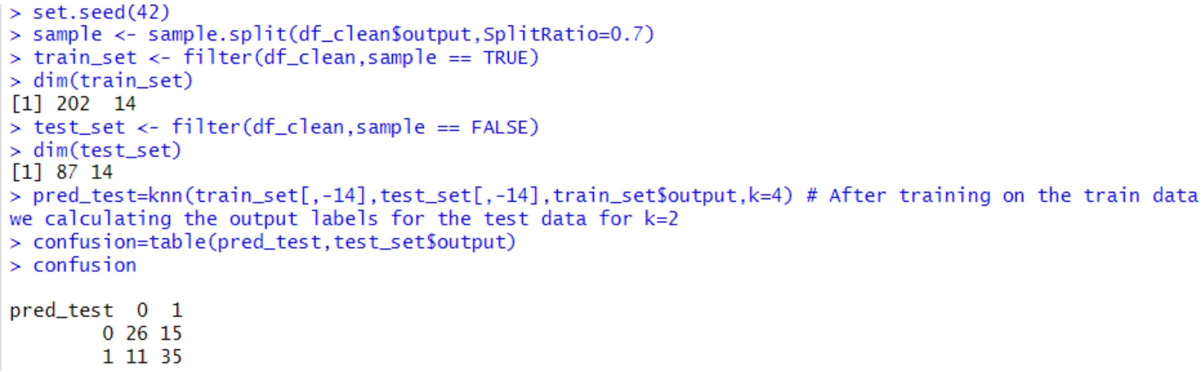


```
> glimpse(data)
Rows: 289
Columns: 14
$ age      <dbl> 60, 35, 41, 55, 56, 55, 56, 44, 52, 57, 54, 48, 49, 64, 55, 50, 58, 66, 40, 69, 5…
$ sex      <dbl> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, …
$ cp       <dbl> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, 0, 2, 0, 2, 3, 1, 2, …
$ trtbps   <dbl> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 130, 110, 150, 120, 1…
$ chol     <dbl> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 266, 211, 283, 219, 3…
$ fbs      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, …
$ restecg  <dbl> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ thalachh <dbl> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171, 144, 162, 158, 1…
$ exng     <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, …
$ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0.6, 1.8, 1.0, 1.6, 0…
$ slp      <dbl> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, 1, 2, 2, 1, 2, 2, 2, …
$ caa      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, …
$ thall    <dbl> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, …
$ output   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
```

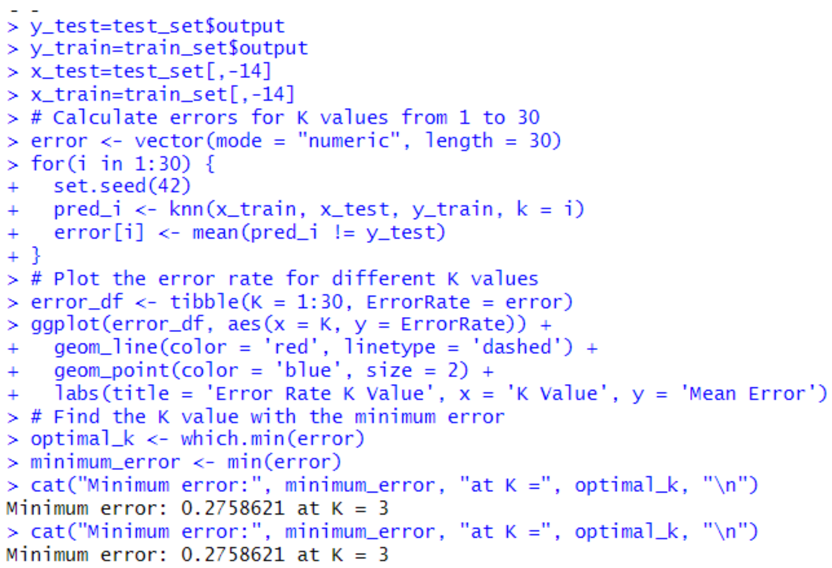
output column signifies whether the patient has a heart problem or not.



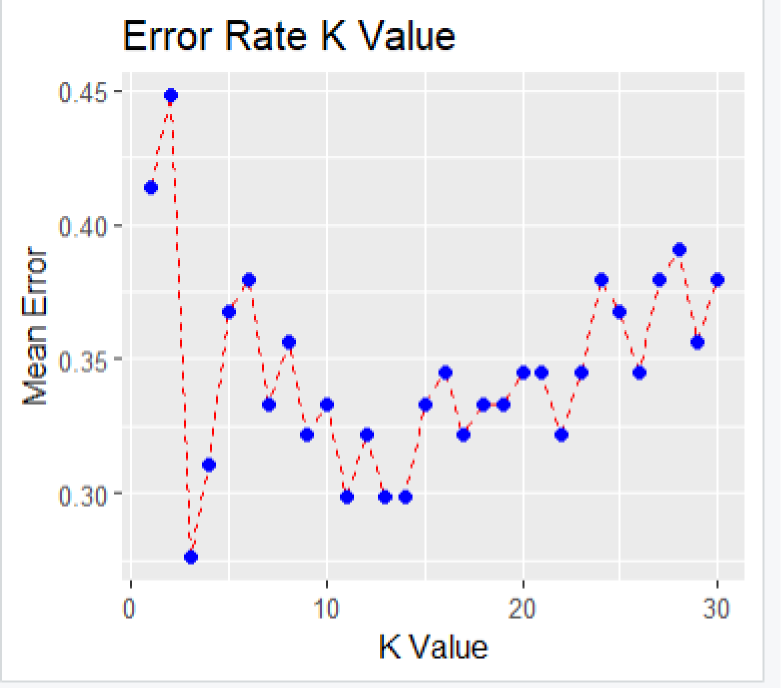
```
> # Clean data
> df_clean <- data %>%
+   filter_all(all_vars(!is.na(.)))
> missing_values <- is.na(df_clean)
> missing_sum <- colSums(missing_values)
> df_clean$output=factor(df_clean$output) # converting the 'class' feature into factor type
> colnames(df_clean)
 [1] "age"      "sex"      "cp"       "trtbps"   "chol"     "fbs"      "restecg"  "thalachh"
 [9] "exng"     "oldpeak"  "slp"      "caa"      "thall"    "output"
```



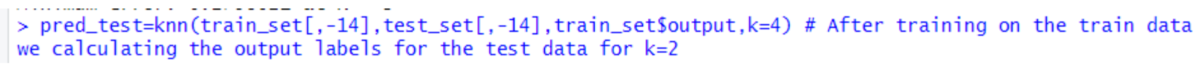
```
> set.seed(42)
> sample <- sample.split(df_clean$output,SplitRatio=0.7)
> train_set <- filter(df_clean,sample == TRUE)
> dim(train_set)
[1] 202  14
> test_set <- filter(df_clean,sample == FALSE)
> dim(test_set)
[1] 87 14
> pred_test=knn(train_set[,-14],test_set[,-14],train_set$output,k=4) # After training on the train data
we calculating the output labels for the test data for k=2
> confusion=table(pred_test,test_set$output)
> confusion

pred_test  0  1
        0 26 15
        1 11 35
```

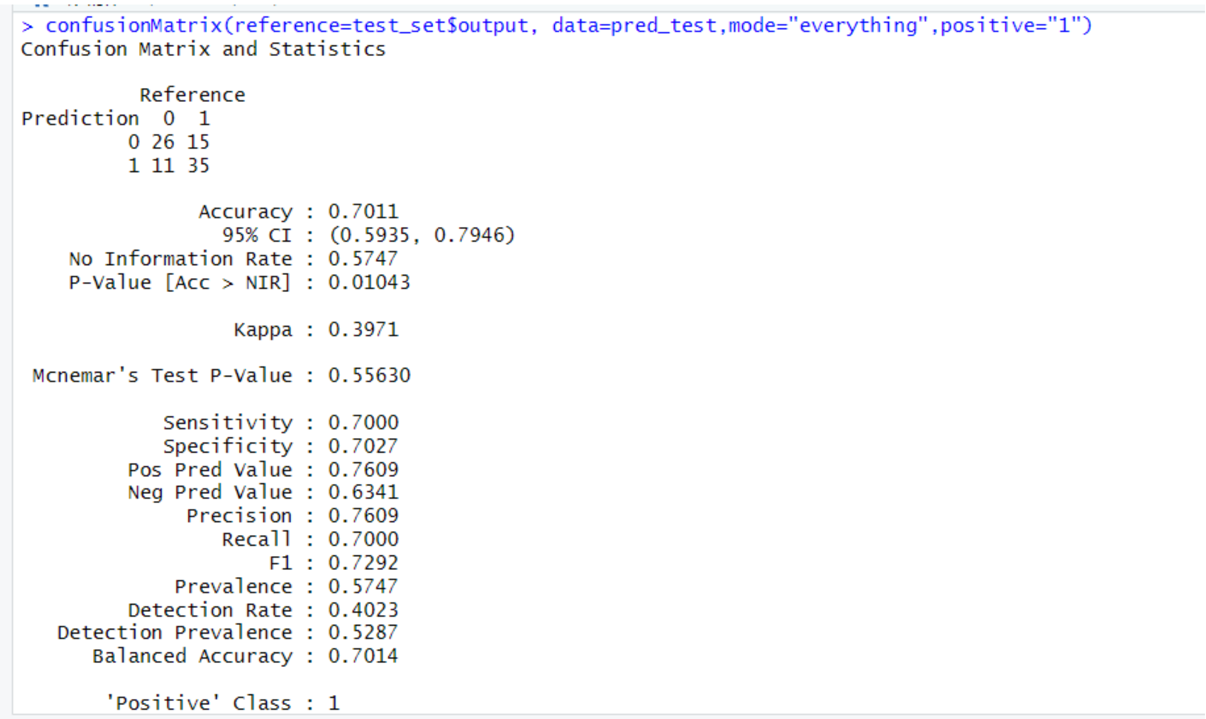


```
> y_test=test_set$output
> y_train=train_set$output
> x_test=test_set[,-14]
> x_train=train_set[,-14]
> # Calculate errors for K values from 1 to 30
> error <- vector(mode = "numeric", length = 30)
> for(i in 1:30) {
+   set.seed(42)
+   pred_i <- knn(x_train, x_test, y_train, k = i)
+   error[i] <- mean(pred_i != y_test)
+ }
> # Plot the error rate for different K values
> error_df <- tibble(K = 1:30, ErrorRate = error)
> ggplot(error_df, aes(x = K, y = ErrorRate)) +
+   geom_line(color = 'red', linetype = 'dashed') +
+   geom_point(color = 'blue', size = 2) +
+   labs(title = 'Error Rate K Value', x = 'K Value', y = 'Mean Error')
> # Find the K value with the minimum error
> optimal_k <- which.min(error)
> minimum_error <- min(error)
> cat("Minimum error:", minimum_error, "at K =", optimal_k, "\n")
Minimum error: 0.2758621 at K = 3
> cat("Minimum error:", minimum_error, "at K =", optimal_k, "\n")
Minimum error: 0.2758621 at K = 3
```



Therefore, K=3


```
> pred_test=knn(train_set[,-14],test_set[,-14],train_set$output,k=4) # After training on the train data
we calculating the output labels for the test data for k=2
```


```
> confusionMatrix(reference=test_set$output, data=pred_test,mode="everything",positive="1")
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 26 15
         1 11 35

               Accuracy : 0.7011
                 95% CI : (0.5935, 0.7946)
    No Information Rate : 0.5747
    P-Value [Acc > NIR] : 0.01043

                  Kappa : 0.3971

 Mcnemar's Test P-Value : 0.55630

            Sensitivity : 0.7000
            Specificity : 0.7027
         Pos Pred Value : 0.7609
         Neg Pred Value : 0.6341
              Precision : 0.7609
                 Recall : 0.7000
                     F1 : 0.7292
             Prevalence : 0.5747
         Detection Rate : 0.4023
   Detection Prevalence : 0.5287
      Balanced Accuracy : 0.7014

       'Positive' Class : 1
```

Analysis:

The KNN model's evaluation shows an accuracy of 71.26%, which is higher than the no information rate of 57.47%, and this difference is statistically significant with a p-value of 0.005587. The model has a Kappa statistic of 0.4223, indicating a moderate level of agreement beyond chance.

The sensitivity or true positive rate is 70.00%, and the specificity or true negative rate is 72.97%. The positive predictive value or precision is 77.78%, and the negative predictive value is 64.29%. The F1 score, which balances precision and recall, is 73.68%.

The model has a balanced accuracy of 71.49%, which takes into account both sensitivity and specificity. Overall, the model shows a good prediction capability for the positive class (1), with room for improvement in reducing false positives and false negatives to enhance precision and recall.

Part 2:

Article: Healthcare predictive analytics using machine learning and deep learning techniques: a survey

Detailed Summary:
The article serves as a comprehensive survey, critically examining the deployment of machine learning (ML) and deep learning (DL) techniques in healthcare predictive analytics. It illustrates how AI is revolutionizing healthcare by providing systems that diagnose and predict diseases from clinical or image data. The authors focus on the substantial impact of accurate predictions in healthcare, which can range from improving patient outcomes to saving lives. The paper also delves into the challenges of implementing these advanced AI techniques, such as the requirement for large datasets, the complexity of model interpretation, and the need for high computational power.

The key ideas discussed in the article include:

1. The integration of machine learning and deep learning in healthcare predictive analytics.
2. The role of artificial intelligence in transforming healthcare through improved diagnostics and predictive capabilities.
3. The critical importance of accurate and reliable predictive analytics in healthcare outcomes.
4. A survey of existing machine learning and deep learning approaches and their applications in disease prediction.
5. The identification of challenges and obstacles in applying machine learning and deep learning techniques in the healthcare domain, such as data privacy, model interpretability, and the need for substantial computational resources.

Detailed Critique:

The paper is commendable for its broad overview of ML and DL applications in healthcare; however, it falls short of providing empirical evidence or case studies to support the claims made. The authors discuss the potential life-saving capabilities of AI but do not thoroughly address the risks associated with AI, such as biases in data leading to incorrect predictions. While the survey identifies challenges in AI application, it could also explore potential solutions or future directions in more depth. A discussion on the balance between automation and human oversight in healthcare predictive analytics would also enhance the paper's practical relevance.

Reference Citation:

To cite this article in your work, use the following format:

Badawy, M., Ramadan, N., & Hefny, H. A. (2023). Healthcare predictive analytics using machine learning and deep learning techniques: a survey. *Journal of Electrical Systems and Information Technology*, 10(40). Available at: https://jesit.springeropen.com/articles/10.1186/s43067-023-00108-y&#8203;``【oaicite:1】``&#8203;&#8203;``【oaicite:0】``&#8203;.