Cervical Cancer_Capstone_Submission

Suchitra Chavan

Table of Contents

Introduction

Cervical cancer is a significant health concern affecting women worldwide. This type of cancer originates in the cells of the cervix, the lower part of the uterus. Cervical cancer is primarily caused by persistent infection with high-risk strains of human papillomavirus (HPV). It often develops slowly, allowing for early detection and prevention through regular screenings, such as Pap tests and HPV vaccinations. Despite advancements in healthcare, cervical cancer remains a prevalent issue, underscoring the importance of awareness, early detection, and comprehensive prevention strategies.

Center for Disease Control (CDC) reports that in 2020, the latest year for which incidence data are available, in the United States, 11,542 new cases of Cervical cancer were reported among women, and 4,272 women died of this cancer. For every 100,000 women, 7 new Cervical cancer cases were reported and 2 women died of this cancer.

How is Cervical Cancer Diagnosed and Treated?

Various approaches are employed to address cervical cancer, contingent on the cancer type and its extent of spread. The treatment options encompass surgery, chemotherapy, and radiation therapy.

- Surgery: This involves the removal of cancerous tissue through a surgical procedure.
- Chemotherapy: Special medications are utilized to either shrink or eliminate the cancer cells. These drugs may be administered orally, intravenously, or through a combination of both methods.
- Radiation Therapy: High-energy rays, akin to X-rays, are employed to destroy cancer cells.

Stages of Cervical Cancer

Cervical cancer is typically categorized into stages based on the extent of its progression. The staging system helps guide treatment decisions and provides insight into the prognosis. The stages are generally classified as follows:

- Stage 0 (Carcinoma in situ): The cancer is confined to the surface layer of the cervix and has not invaded deeper tissues.

- Stage I: At this stage, the cancer is localized to the cervix. Subcategories include: IA: Microscopic invasion. IB: Visible invasion.

- Stage II: The cancer has spread beyond the cervix but is still within the pelvic area. Subcategories include: IIA: Involvement of the upper two-thirds of the vagina. IIB: Infiltration into the parametrial tissues.

- Stage III: Cancer extends to the lower part of the vagina or the pelvic sidewall. Subcategories include: IIIA: Involvement of the lower vagina. IIIB: Extensive invasion into the pelvic sidewall or causing hydronephrosis or non-functioning kidney.

- Stage IV: The cancer has spread beyond the pelvic area or involves distant organs. Subcategories include: IVA: Spread to adjacent organs. IVB: Distant metastasis.

Staging plays a crucial role in developing an appropriate and effective treatment plan for individuals diagnosed with cervical cancer.

Treatment plan based on stages of cancer.

The treatment plan for cervical cancer varies depending on the specific stage of the disease. Here's a general overview of treatment approaches based on cervical cancer stages:

- Stage 0 (Carcinoma in situ): Treatment: Surgery such as cone biopsy or loop electrosurgical excision procedure (LEEP) to remove the abnormal cells. Follow-up: Regular monitoring to ensure the absence of recurrence.

- Stage I: Treatment: Options include surgery (hysterectomy or trachelectomy), radiation therapy, or a combination of both. Follow-up: Regular follow-up examinations and imaging to monitor for any signs of recurrence.

- Stage II: Treatment: Surgery (radical hysterectomy with removal of pelvic lymph nodes), radiation therapy, or a combination of both. Follow-up: Monitoring for recurrence and potential additional treatments as needed.

- Stage III: Treatment: Combination therapy involving surgery, radiation, and chemotherapy. This may include a radical hysterectomy, removal of lymph nodes, and pelvic radiation. Follow-up: Ongoing surveillance to detect any recurrence or metastasis. Chemotherapy may be continued or initiated based on the response to initial treatment.

- Stage IV: Treatment: Treatment options include a combination of surgery, radiation, and chemotherapy. Palliative care may also be incorporated to manage symptoms and enhance quality of life. Follow-up: Regular monitoring for response to treatment and managing any recurrent or persistent disease. Palliative care may continue to address symptoms and improve overall well-being.

Exploring the Cervical Cancer Dataset

The data was gathered at ‘Hospital Universitario de Caracas’ in Caracas, Venezuela, encompassing demographic details, lifestyle habits, and historical medical records of 858 individuals link. The UCI repository played a pivotal role in assembling the dataset titled “Cervical Cancer Risk Factors for Biopsy.” This comprehensive collection encompasses details on the activities, demographics, and medical histories of 858 individuals. Notably, the dataset is not without challenges, as it contains multiple instances of missing values, primarily arising from patients opting not to respond to certain questions due to privacy concerns.

Comprising 858 instances, each characterized by 32 properties, this dataset sheds light on various aspects of patients’ lives, with a particular focus on the medical history of female patients. The dataset spans 32 variables, offering a rich array of information on 858 women. Key factors examined include age, IUD (Intrauterine Device) usage, smoking habits, and STDs (Sexually Transmitted Diseases).

Overall Goal

The overarching goal of this project is to harness the capabilities of Artificial Intelligence and Machine Learning to develop predictive models for cervical cancer. By leveraging advanced data analytics and modeling techniques, the aim is to empower healthcare professionals with effective tools for early detection, risk assessment, and decision support. The ultimate objective is to contribute to the advancement of personalized medicine, enhancing patient outcomes and aiding in the global effort to combat cervical cancer through innovative and data-driven approaches.

Method and Analysis

```
# Cervical Cancer dataset
# Exploratory data analysis
# Load necessary libraries
# Load the 'caret' library if not already installed
library(caret)
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}

# Load the 'dplyr' library if not already installed
library(dplyr)
if (!require(dplyr)) {
  install.packages("dplyr", repos = "http://cran.us.r-project.org")
  library(dplyr)
}

# Load your dataset (ensure the file path is correct)
cervical_cancer <- read.csv("risk_factors_cervical_cancer.csv")

# Convert "?" to NA
cervical_cancer[cervical_cancer == "?"] <- NA

# Ensure 'Biopsy' is a factor variable with two levels
cervical_cancer$Biopsy <- as.factor(cervical_cancer$Biopsy)

# Drop columns with special characters
cervical_cancer <- cervical_cancer %>% select(-c(STDs..Time.since.first.diagn
osis, STDs..Time.since.last.diagnosis))

# Remove rows with missing values
cervical_cancer <- na.omit(cervical_cancer)

summary(cervical_cancer)

#output
summary(cervical_cancer)
      Age        Number.of.sexual.partners First.sexual.intercourse Num.of.pr
egnancies    Smokes
 Min.   :13.00   Length:668                Length:668               Length:66
8         Length:668
 1st Qu.:21.00   Class :character          Class :character         Class :ch
aracter   Class :character
 Median :26.00   Mode  :character          Mode  :character         Mode  :ch
aracter   Mode  :character
 Mean   :27.26
 3rd Qu.:33.00
 Max.   :84.00
 Smokes..years.     Smokes..packs.year. Hormonal.Contraceptives Hormonal.Cont
raceptives..years.     IUD
 Length:668         Length:668          Length:668              Length:668
Length:668
 Class :character   Class :character    Class :character        Class :charac
ter                Class :character
 Mode  :character   Mode  :character    Mode  :character        Mode  :charac
ter                Mode  :character



 IUD..years.            STDs           STDs..number.      STDs.condylomatosis
STDs.cervical.condylomatosis
 Length:668         Length:668         Length:668         Length:668
Length:668
```

```
 Class :character   Class :character   Class :character   Class :character   
Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   
Mode  :character  
                                                                              
                                                                              
                                                                              
 STDs.vaginal.condylomatosis STDs.vulvo.perineal.condylomatosis STDs.syphilis
STDs.pelvic.inflammatory.disease
 Length:668                  Length:668                         Length:668   
Length:668                      
 Class :character            Class :character                   Class :charac
ter   Class :character          
 Mode  :character            Mode  :character                   Mode  :charac
ter   Mode  :character          
                                                                              
                                                                              
                                                                              
 STDs.genital.herpes STDs.molluscum.contagiosum  STDs.AIDS           STDs.HIV
STDs.Hepatitis.B  
 Length:668          Length:668                 Length:668          Length:668
Length:668        
 Class :character    Class :character           Class :character   Class :cha
racter   Class :character  
 Mode  :character    Mode  :character           Mode  :character   Mode  :cha
racter   Mode  :character  
                                                                              
                                                                              
                                                                              
   STDs.HPV         STDs..Number.of.diagnosis   Dx.Cancer          Dx.CIN
Dx.HPV              Dx          
 Length:668         Min.   :0.00000           Min.   :0.00000   Min.   :0.000
000   Min.   :0.00000   Min.   :0.00000  
 Class :character   1st Qu.:0.00000           1st Qu.:0.00000   1st Qu.:0.000
000   1st Qu.:0.00000   1st Qu.:0.00000  
 Mode  :character   Median :0.00000           Median :0.00000   Median :0.000
000   Median :0.00000   Median :0.00000  
                    Mean   :0.09281           Mean   :0.02545   Mean   :0.004
491   Mean   :0.02395   Mean   :0.02395  
                    3rd Qu.:0.00000           3rd Qu.:0.00000   3rd Qu.:0.000
000   3rd Qu.:0.00000   3rd Qu.:0.00000  
                    Max.   :3.00000           Max.   :1.00000   Max.   :1.000
000   Max.   :1.00000   Max.   :1.00000  
   Hinselmann         Schiller          Citology       Biopsy 
 Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   0:623  
 1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1: 45  
 Median :0.00000   Median :0.00000   Median :0.00000          
 Mean   :0.04491   Mean   :0.09431   Mean   :0.05838          
 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000          
 Max.   :1.00000   Max.   :1.00000   Max.   :1.00000


class(cervical_cancer)
# output
> class(cervical_cancer)
[1] "data.frame"

# Set option to display all columns
options(repr.matrix.max.cols=Inf, repr.matrix.max.rows=5, repr.max.print=Inf)

# Display the head of the cervical_cancer dataframe
head(cervical_cancer)

# Get the column names of the cervical_cancer dataframe
column_names <- names(cervical_cancer)

# Display the list of column names
print(column_names)

# output
head(cervical_cancer)
  Age Number.of.sexual.partners First.sexual.intercourse Num.of.pregnancies S
mokes Smokes..years. Smokes..packs.year.
1  18                       4.0                     15.0                1.0
0.0            0.0                 0.0
2  15                       1.0                     14.0                1.0
0.0            0.0                 0.0
4  52                       5.0                     16.0                4.0
1.0           37.0                37.0
5  46                       3.0                     21.0                4.0
0.0            0.0                 0.0
6  42                       3.0                     23.0                2.0
0.0            0.0                 0.0
7  51                       3.0                     17.0                6.0
1.0           34.0                 3.4
  Hormonal.Contraceptives Hormonal.Contraceptives..years. IUD IUD..years. STD
s STDs..number. STDs.condylomatosis
1                     0.0                             0.0 0.0         0.0  0.
0           0.0                 0.0
2                     0.0                             0.0 0.0         0.0  0.
0           0.0                 0.0
4                     1.0                             3.0 0.0         0.0  0.
0           0.0                 0.0
5                     1.0                            15.0 0.0         0.0  0.
0           0.0                 0.0
6                     0.0                             0.0 0.0         0.0  0.
0           0.0                 0.0
7                     0.0                             0.0 1.0         7.0  0.
0           0.0                 0.0
```

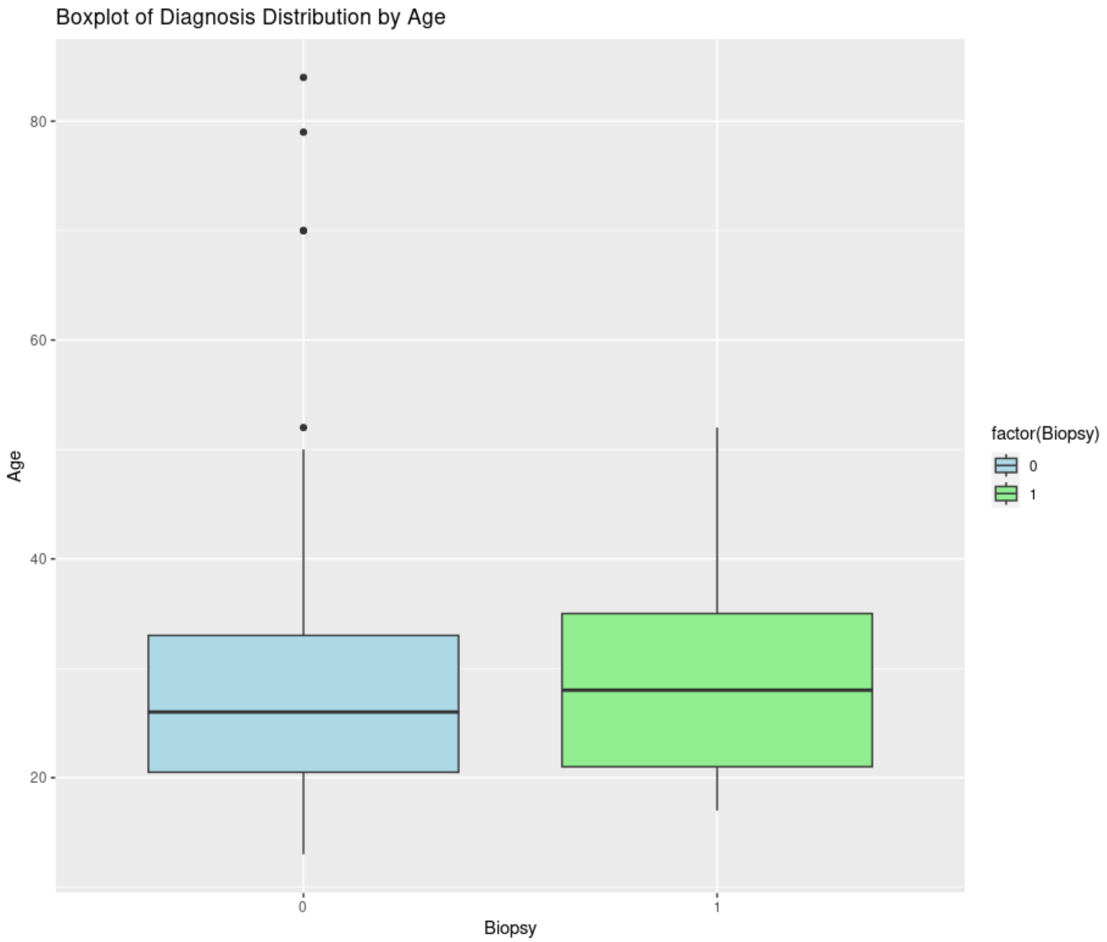
```
  STDs.cervical.condylomatosis STDs.vaginal.condylomatosis STDs.vulvo.perinea
l.condylomatosis STDs.syphilis
1                          0.0                         0.0
0.0           0.0
2                          0.0                         0.0
0.0           0.0
4                          0.0                         0.0
0.0           0.0
5                          0.0                         0.0
0.0           0.0
6                          0.0                         0.0
0.0           0.0
7                          0.0                         0.0
0.0           0.0
  STDs.pelvic.inflammatory.disease STDs.genital.herpes STDs.molluscum.contagi
osum STDs.AIDS STDs.HIV STDs.Hepatitis.B
1                              0.0                 0.0
0.0       0.0      0.0              0.0
2                              0.0                 0.0
0.0       0.0      0.0              0.0
4                              0.0                 0.0
0.0       0.0      0.0              0.0
5                              0.0                 0.0
0.0       0.0      0.0              0.0
6                              0.0                 0.0
0.0       0.0      0.0              0.0
7                              0.0                 0.0
0.0       0.0      0.0              0.0
  STDs.HPV STDs..Number.of.diagnosis Dx.Cancer Dx.CIN Dx.HPV Dx Hinselmann Sc
hiller Citology Biopsy
1      0.0                         0         0      0      0  0          0
0        0      0
2      0.0                         0         0      0      0  0          0
0        0      0
4      0.0                         0         1      0      1  0          0
0        0      0
5      0.0                         0         0      0      0  0          0
0        0      0
6      0.0                         0         0      0      0  0          0
0        0      0
7      0.0                         0         0      0      0  0          1
1        0      1
>
> # Get the column names of the cervical_cancer dataframe
> column_names <- names(cervical_cancer)
>
> # Display the list of column names
> print(column_names)
 [1] "Age"                                "Number.of.sexual.partners"
"First.sexual.intercourse"
```

```
 [4] "Num.of.pregnancies"                 "Smokes"
"Smokes..years."
 [7] "Smokes..packs.year."                "Hormonal.Contraceptives"
"Hormonal.Contraceptives..years."
[10] "IUD"                                "IUD..years."
"STDs"
[13] "STDs..number."                      "STDs.condylomatosis"
"STDs.cervical.condylomatosis"
[16] "STDs.vaginal.condylomatosis"        "STDs.vulvo.perineal.condylomatosis
" "STDs.syphilis"
[19] "STDs.pelvic.inflammatory.disease"   "STDs.genital.herpes"
"STDs.molluscum.contagiosum"
[22] "STDs.AIDS"                          "STDs.HIV"
"STDs.Hepatitis.B"
[25] "STDs.HPV"                           "STDs..Number.of.diagnosis"
"Dx.Cancer"
[28] "Dx.CIN"                             "Dx.HPV"
"Dx"
[31] "Hinselmann"                         "Schiller"
"Citology"
[34] "Biopsy"
```

Exploring the dataset

Boxplot of diagnosis distribution by age using ggplot2

```
# Load the 'ggplot2' library if not already installed
if (!require(ggplot2)) {
  install.packages("ggplot2", repos = "http://cran.us.r-project.org")
  library(ggplot2)
}
library(ggplot2)

ggplot(cervical_cancer, aes(x = factor(Biopsy), y = Age, fill = factor(Biopsy
))) +
  geom_boxplot() +
  scale_fill_manual(values = c("lightblue", "lightgreen")) +
  labs(title = "Boxplot of Diagnosis Distribution by Age",
       x = "Biopsy", y = "Age")
```

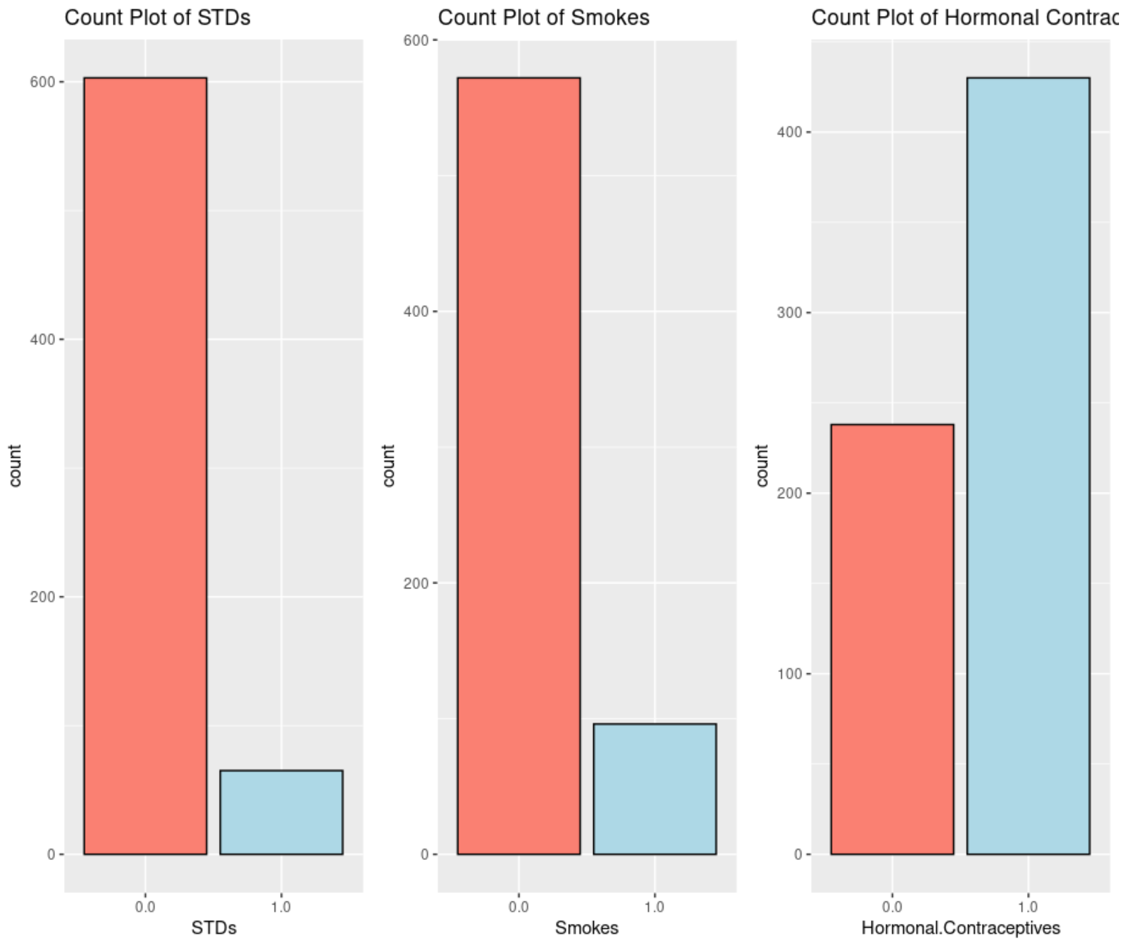


Boxplot of diagnosis distribution by age: The above plot shows that most of the patients with a positive biopsy range in the age of 22-35 years old.

Countplots for risk factors

```
# Load necessary libraries
# Load the 'ggplot2' library if not already installed
if (!require(ggplot2)) {
  install.packages("ggplot2", repos = "http://cran.us.r-project.org")
  library(ggplot2)
}
library(ggplot2)

# Load the 'gridExtra' library if not already installed
if (!require(gridExtra)) {
  install.packages("gridExtra", repos = "http://cran.us.r-project.org")
  library(gridExtra)
}

library(gridExtra)

# Create count plots for risk factors with different colors
```

```
options(repr.plot.width = 15, repr.plot.height = 4)  # Set plot size

# Countplot for 'STDs'
p1 <- ggplot(cervical_cancer, aes(x = STDs)) +
  geom_bar(fill = c('salmon', 'lightblue'), color = 'black') +
  labs(title = 'Count Plot of STDs')

# Countplot for 'Smokes'
p2 <- ggplot(cervical_cancer, aes(x = Smokes)) +
  geom_bar(fill = c('salmon', 'lightblue'), color = 'black') +
  labs(title = 'Count Plot of Smokes')

# Countplot for 'Hormonal Contraceptives'
p3 <- ggplot(cervical_cancer, aes(x = `Hormonal.Contraceptives`)) +
  geom_bar(fill = c('salmon', 'lightblue'), color = 'black') +
  labs(title = 'Count Plot of Hormonal Contraceptives')

grid.arrange(p1, p2, p3, ncol = 3)
```



Countplots of risk factors- The above plot shows the distribution of some of the most significant factors for cervical cancer dataset.

Data preparation for analysis.

The dataset related to cervical cancer is being prepared for machine learning analysis. The matrix of features (X) is created by excluding the “Biopsy” column, which serves as the dependent variable vector (y). The dataset is then split into training and testing sets using the createDataPartition function, where 80% of the data is allocated for training. The seed is set for reproducibility. The resulting training sets (X_train and y_train) contain the features and labels for model training, while the testing sets (X_test and y_test) hold the corresponding data for evaluating the model’s performance on unseen instances. This systematic data splitting and preprocessing lay the foundation for building and assessing machine learning models for cervical cancer prediction.

```
################################################################
# Data splitting and Preprocessing
##################################################################
# Create the matrix of features and the dependent variable vector
X <- cervical_cancer[, !colnames(cervical_cancer) %in% "Biopsy"]
y <- cervical_cancer[, "Biopsy"]

# Data set splitting
set.seed(1)
train_index <- createDataPartition(y, p = 0.8, list = FALSE)
X_train <- X[train_index, ]
y_train <- y[train_index]
X_test <- X[-train_index, ]
y_test <- y[-train_index]
```

Model testing

KNN Model

K-Nearest Neighbors (KNN) is a simple yet effective algorithm used in machine learning and artificial intelligence for both classification and regression tasks. The basic idea behind the KNN algorithm is to predict the class or value of a data point based on the majority class or average value of its k-nearest neighbors.

KNN Model on training dataset

A K-Nearest Neighbors (KNN) model is implemented for the training dataset using the caret library. The train function is employed to create and train the KNN model, with the number of neighbors (k) set to 5. The model performance is evaluated using 10-fold cross-validation (trControl = trainControl(method = “cv”, number = 10)), providing insights into accuracy and Kappa statistics. The printed model summary reveals that the dataset consists of 535 samples with 33 predictor variables and two classes (‘0’ and ‘1’). The cross-validated accuracy is reported at 93.28%, with a Kappa statistic of 0. Notably, the tuning parameter ‘k’ was held constant at a value of 5 during the model evaluation. This KNN model serves as an initial step in leveraging machine learning for cervical cancer prediction based on the provided dataset.

```
####################################################
# KNN model for training dataset
#############################################
# Load necessary libraries
# Load the 'caret' library if not already installed
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}
library(caret)

# Create and train a KNN model
knn_model <- train(
  x = X_train,
  y = y_train,
  method = "knn",
  trControl = trainControl(method = "cv", number = 10),
  tuneGrid = expand.grid(k = 5)
)

# Print the model
print(knn_model)

# output
print(knn_model)
k-Nearest Neighbors

535 samples
 33 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 483, 482, 481, 482, 481, 481, ...
Resampling results:

  Accuracy   Kappa
  0.9328052  0

Tuning parameter 'k' was held constant at a value of 5
```

KNN model on test dataset

The previously trained K-Nearest Neighbors (KNN) model is applied to the test dataset to make predictions. The predict function is utilized to generate predictions for the test set, and a confusion matrix is constructed to evaluate the model’s performance. The confusion matrix displays the number of true positive, true negative, false positive, and false negative predictions. Performance metrics such as accuracy, precision, recall, and F1 score are extracted from the confusion matrix, providing a comprehensive assessment of the model’s effectiveness. In this specific case, the model exhibits an accuracy of 93.23%, precision of 93.23%, recall of 100%, and an F1 score of 96.50%. These metrics underscore the model's proficiency in correctly classifying instances of the given classes (0 and 1) in the test dataset, demonstrating promising predictive capabilities for cervical cancer detection.

```
####################################################
# KNN model for test dataset
####################################################

# Predict on the test set
knn_predictions <- predict(knn_model, newdata = X_test)

# Confusion matrix
confusion_matrix <- confusionMatrix(knn_predictions, y_test)
print(confusion_matrix)

# Model performance metrics
accuracy <- confusion_matrix$overall["Accuracy"]
precision <- confusion_matrix$byClass["Precision"]
recall <- confusion_matrix$byClass["Recall"]
f1_score <- confusion_matrix$byClass["F1"]

# Print performance metrics
cat("Accuracy:", accuracy, "\n")
cat("Precision:", precision, "\n")
cat("Recall:", recall, "\n")
cat("F1 Score:", f1_score, "\n")

#output
# Print performance metrics
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.9323308
> cat("Precision:", precision, "\n")
Precision: 0.9323308
> cat("Recall:", recall, "\n")
Recall: 1
> cat("F1 Score:", f1_score, "\n")
F1 Score: 0.9649805
```

Results for KNN Model

```
| Model | Dataset    | Accuracy     | Kappa    | K Parameter |
|-------|------------|--------------|----------|-------------|
| KNN   | Training   | 0.9328052    | 0        | 5           |
| KNN   | Test       | 0.9323308    | -        | -           |
```

Support Vector Machines (SVM) model

Support Vector Machines (SVM) is a powerful supervised machine learning algorithm used for both classification and regression tasks. SVMs are particularly effective in highdimensional spaces and are well-suited for situations where the data has clear margins of separation.

A Support Vector Machine (SVM) model is constructed for the cervical cancer dataset. The caret and e1071 libraries are employed, with the SVM model configured to use a linear kernel and a cost parameter set to 1. After training the model on the training dataset, it is printed, revealing key details such as SVM type, kernel type, and the number of support vectors.

Subsequently, the model's performance is assessed on both the training and test sets. For the training set, predictions are made, and the accuracy is printed, demonstrating a training accuracy of 95.89%. On the test set, predictions are generated, and a confusion matrix is computed. The confusion matrix provides insights into true positive, true negative, false positive, and false negative predictions, from which various performance metrics are derived. The SVM model achieves a test accuracy of 96.24%, indicating its effectiveness in correctly classifying instances of cervical cancer in the test dataset. Additionally, sensitivity, specificity, positive predictive value, negative predictive value, and balanced accuracy metrics contribute to a comprehensive evaluation of the SVM model's classification performance.

```
####################################################
# SVM model
####################################################
# Load the 'caret' library if not already installed
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}
library(caret)
# Load the 'e1071' library if not already installed
if (!require(e1071)) {
  install.packages("e1071", repos = "http://cran.us.r-project.org")
  library(e1071)
}
library(e1071)  # for SVM
# Create and train an SVM model
svm_model <- svm(
  x = X_train,
  y = y_train,
  kernel = "linear",  # you can try different kernels (linear, polynomial, et
c.)
  cost = 1,          # cost parameter (adjust as needed)
  scale = FALSE      # you can adjust other parameters based on your requirem
ents
)

# Print the model
print(svm_model)
```

```
# Predict on the training set
train_predictions <- predict(svm_model, newdata = X_train)

# Print training accuracy
train_accuracy <- confusionMatrix(train_predictions, y_train)$overall["Accura
cy"]
cat("Training Accuracy:", train_accuracy, "\n")

# Predict on the test set
test_predictions <- predict(svm_model, newdata = X_test)

# Confusion matrix
confusion_matrix_svm <- confusionMatrix(test_predictions, y_test)
print(confusion_matrix_svm)

# Model performance metrics
accuracy_svm <- confusion_matrix_svm$overall["Accuracy"]

# Print test accuracy
cat("Test Accuracy (SVM):", accuracy_svm, "\n")

#output
# Predict on the training set
> train_predictions <- predict(svm_model, newdata = X_train)
>
> # Print training accuracy
> train_accuracy <- confusionMatrix(train_predictions, y_train)$overall["Accu
racy"]
> cat("Training Accuracy:", train_accuracy, "\n")
Training Accuracy: 0.9588785
>
> # Predict on the test set
> test_predictions <- predict(svm_model, newdata = X_test)
>
> # Confusion matrix
> confusion_matrix_svm <- confusionMatrix(test_predictions, y_test)
> print(confusion_matrix_svm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 121   2
         1   3   7

               Accuracy : 0.9624
                 95% CI : (0.9144, 0.9877)
    No Information Rate : 0.9323
    P-Value [Acc > NIR] : 0.1073
```

```
                  Kappa : 0.7167

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.9758
            Specificity : 0.7778
         Pos Pred Value : 0.9837
         Neg Pred Value : 0.7000
             Prevalence : 0.9323
         Detection Rate : 0.9098
   Detection Prevalence : 0.9248
      Balanced Accuracy : 0.8768

       'Positive' Class : 0


>
> # Model performance metrics
> accuracy_svm <- confusion_matrix_svm$overall["Accuracy"]
>
> # Print test accuracy
> cat("Test Accuracy (SVM):", accuracy_svm, "\n")
Test Accuracy (SVM): 0.962406
```

Results for SVM model

```
| Model | Dataset   | Accuracy    |
|-------|------------|-------------|
| SVM   | Training   | 0.9588785   |
| SVM   | Test       | 0.962406    |
```

Random Forest Model

A Random Forest model is constructed and evaluated for the cervical cancer dataset using the caret library. The training of the Random Forest model involves specifying the predictors (X_train) and the response variable (y_train). The model is configured with 10-fold cross-validation, and the optimal number of predictors (mtry) is determined through the tuning process. The final model uses mtry = 17.

Subsequently, the model’s performance is assessed on the test set. Predictions are generated, and a confusion matrix is computed, providing a breakdown of true positive, true negative, false positive, and false negative predictions. The Random Forest model achieves a test accuracy of 95.49%, indicating its proficiency in correctly classifying instances of cervical cancer in the test dataset. Additional metrics such as sensitivity, specificity, positive predictive value, negative predictive value, and balanced accuracy contribute to a comprehensive evaluation of the Random Forest model’s classification performance.

```
####################################################
# Random Forest model for training dataset
####################################################
```

```
# Load necessary libraries
# Load the 'caret' library if not already installed
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}
library(caret)
# Create and train a random forest model
rf_model <- train(
  x = X_train,
  y = y_train,
  method = "rf",  # Random forest method
  trControl = trainControl(method = "cv", number = 10)  # Cross-validation
)

# Print the model
print(rf_model)

#####################################################
# Random Forest model evaluation on the test set
#####################################################

# Predict on the test set
rf_predictions <- predict(rf_model, newdata = X_test)

# Confusion matrix
confusion_matrix_rf <- confusionMatrix(rf_predictions, y_test)
print(confusion_matrix_rf)

# Model performance metrics
accuracy_rf <- confusion_matrix_rf$overall["Accuracy"]

# Print test accuracy
cat("Test Accuracy (Random Forest):", accuracy_rf, "\n")

# Output
################################################################
> # Random Forest model for training dataset
> #####################################################
> # Load necessary libraries
# Load the 'caret' library if not already installed
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}
> library(caret)
>
> # Create and train a random forest model
> rf_model <- train(
+   x = X_train,
+   y = y_train,
+   method = "rf",  # Random forest method
+   trControl = trainControl(method = "cv", number = 10)  # Cross-validation
+ )
>
> # Print the model
> print(rf_model)
Random Forest

535 samples
 33 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 482, 482, 481, 481, 481, 482, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9327743  0.0000000
  17    0.9514675  0.4623585
  33    0.9514326  0.4253274

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 17.
```

Results of RFM model on training set

```
| Model          | Dataset    | mtry | Accuracy     | Kappa        |
|---------------|------------|------|-------------|-------------|
| Random Forest | Training   | 2    | 0.9327743   | 0.0000000   |
| Random Forest | Training   | 17   | 0.9514675   | 0.4623585   |
| Random Forest | Training   | 33   | 0.9514326   | 0.4253274   |
### Accuracy was used to select the optimal model using the largest value.The
final value used for the model was mtry = 17.

####################################################
# Random Forest model evaluation on the test set
####################################################

# Predict on the test set
rf_predictions <- predict(rf_model, newdata = X_test)

# Confusion matrix
confusion_matrix_rf <- confusionMatrix(rf_predictions, y_test)
print(confusion_matrix_rf)

# Model performance metrics
accuracy_rf <- confusion_matrix_rf$overall["Accuracy"]
```

```
# Print test accuracy
cat("Test Accuracy (Random Forest):", accuracy_rf, "\n")

# Output
####################################################
> # Random Forest model evaluation on the test set
> ####################################################
>
> # Predict on the test set
> rf_predictions <- predict(rf_model, newdata = X_test)
>
> # Confusion matrix
> confusion_matrix_rf <- confusionMatrix(rf_predictions, y_test)
> print(confusion_matrix_rf)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 123   5
         1   1   4

               Accuracy : 0.9549
                 95% CI : (0.9044, 0.9833)
    No Information Rate : 0.9323
    P-Value [Acc > NIR] : 0.1973

                  Kappa : 0.5497

 Mcnemar's Test P-Value : 0.2207

            Sensitivity : 0.9919
            Specificity : 0.4444
         Pos Pred Value : 0.9609
         Neg Pred Value : 0.8000
             Prevalence : 0.9323
         Detection Rate : 0.9248
   Detection Prevalence : 0.9624
      Balanced Accuracy : 0.7182

       'Positive' Class : 0

>
> # Model performance metrics
> accuracy_rf <- confusion_matrix_rf$overall["Accuracy"]
>
> # Print test accuracy
> cat("Test Accuracy (Random Forest):", accuracy_rf, "\n")
```

```
Test Accuracy (Random Forest): 0.9548872
>
```

Results for RFM on test

```
| Model          | Dataset    |  Accuracy    |
|---------------|------------|-------------|
| Random Forest | Test        |  0.9548872   |
```

ROC curve and Area Under the Curve (AUC)

Receiver Operating Characteristic (ROC) curves are generated for three machine learning models (KNN, SVM, and Random Forest) applied to the test dataset, and the Area Under the Curve (AUC) values are computed for each model.

Plotting ROC Curves:

The pROC library is loaded for ROC curve generation. ROC curves for the three models (KNN, SVM, and Random Forest) are plotted on the same graph for easy visual comparison. Each curve is assigned a distinct color (blue for KNN, red for SVM, and green for Random Forest). A legend is added to the plot to identify each model. KNN Model Evaluation:

Predictions and predicted probabilities are obtained for the KNN model on the test set.ROC curve for the KNN model is created using the roc function from the pROC library.The AUC value for the KNN model is computed and printed.

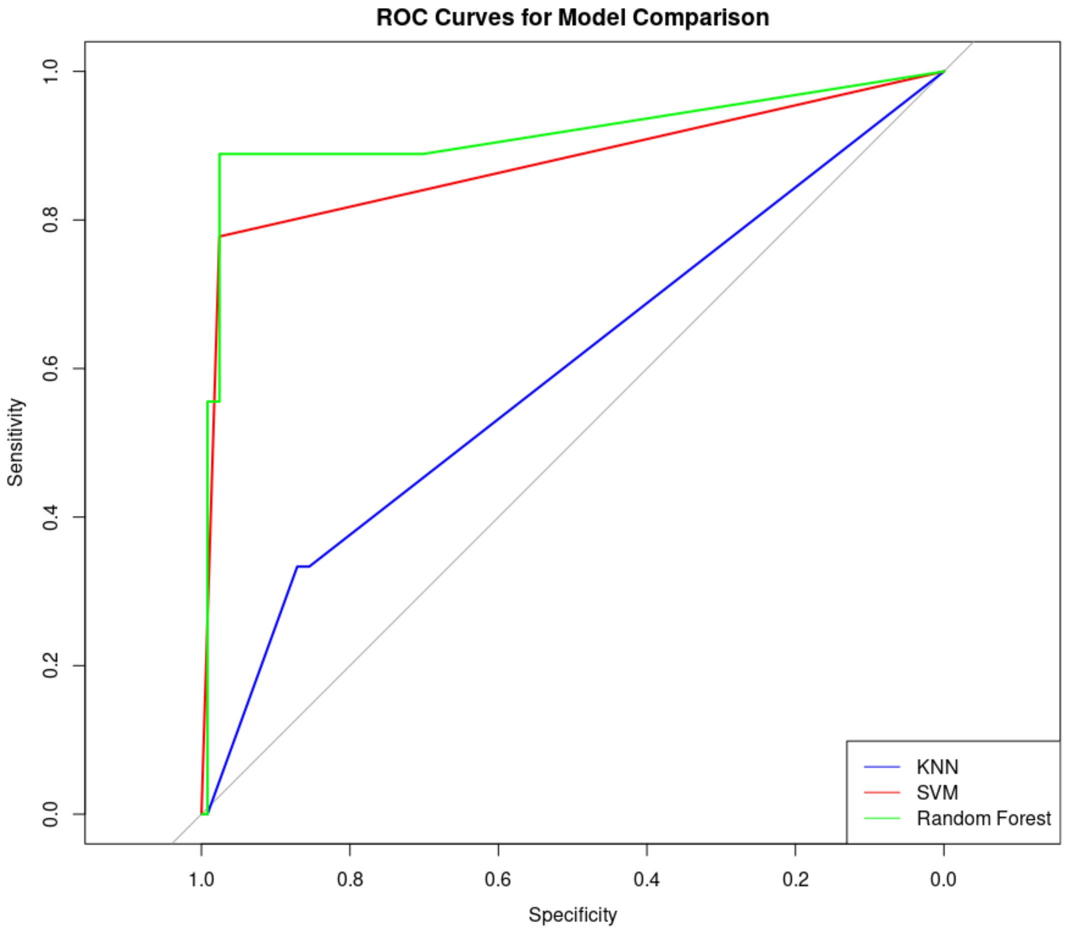
SVM Model Evaluation:Predictions and predicted probabilities are obtained for the SVM model on the test set. ROC curve for the SVM model is created. The AUC value for the SVM model is computed and printed.

Random Forest Model Evaluation: Predictions and predicted probabilities are obtained for the Random Forest model on the test set. ROC curve for the Random Forest model is created.The AUC value for the Random Forest model is computed and printed.

```
#############################################################
#ROC
# Load necessary libraries
# Load the 'caret' library if not already installed
if (!require(caret)) {
  install.packages("caret", repos = "http://cran.us.r-project.org")
  library(caret)
}
library(caret)
# Load the 'pROC' library if not already installed
if (!require(pROC)) {
  install.packages("pROC", repos = "http://cran.us.r-project.org")
  library(pROC)
}
library(pROC)
####################################################
# KNN model for test dataset
```

```
####################################################
# Set the layout to a single plot
par(mfrow = c(1, 1))

# Create a new single plot (replace this with your own plot code)
new_plot <- ggplot(...) + ...

# Print the new plot
print(new_plot)

# Predict on the test set
knn_predictions <- predict(knn_model, newdata = X_test)

# Calculate predicted probabilities
knn_probabilities <- as.numeric(predict(knn_model, newdata = X_test, type = "
prob")[, "1"])

# Create ROC curve for KNN
roc_knn <- roc(y_test, knn_probabilities)

####################################################
# SVM model for test dataset
####################################################

# Predict on the test set
svm_predictions <- predict(svm_model, newdata = X_test)

# Calculate predicted probabilities
svm_probabilities <- as.numeric(predict(svm_model, newdata = X_test, type = "
response"))

# Create ROC curve for SVM
roc_svm <- roc(y_test, svm_probabilities)

####################################################
# Random Forest model for test dataset
####################################################

# Predict on the test set
rf_predictions <- predict(rf_model, newdata = X_test)

# Calculate predicted probabilities
rf_probabilities <- predict(rf_model, newdata = X_test, type = "prob")[, "1"]

# Create ROC curve for Random Forest
roc_rf <- roc(y_test, rf_probabilities)

####################################################
# Plotting ROC curves for all three models
```

```
####################################################

# Plot ROC curves
plot(roc_knn, col = "blue", main = "ROC Curves for Model Comparison")
lines(roc_svm, col = "red")
lines(roc_rf, col = "green")

# Add legend
legend("bottomright", legend = c("KNN", "SVM", "Random Forest"), col = c("blu
e", "red", "green"), lty = 1)
```



ROC curves for model comparison

ROC curves for model comparison The ROC curves vividly illustrate the comparative performance of our models. Notably, the Random Forest model exhibits the most robust performance, outpacing the SVM model. On the other hand, the KNN model lags behind, demonstrating the lowest performance among the models assessed.

Area Under Curve (AUC)

The AUC values provide a quantitative measure of each model's ability to discriminate between the two classes. Higher AUC values generally indicate better model performance. In this case, the Random Forest model demonstrates the highest AUC (0.915), followed by the SVM model (0.877), while the KNN model has a lower AUC (0.595). These results offer insights into the models' discriminatory abilities, aiding in the selection of the most effective model for the given cervical cancer dataset.

```
#####################################
# Area Under Curve (AUC)
# Load necessary libraries
# Load the 'pROC' library if not already installed
if (!require(pROC)) {
  install.packages("pROC", repos = "http://cran.us.r-project.org")
  library(pROC)
}
library(pROC)
####################################################
# KNN model for test dataset
####################################################

# Predict on the test set
knn_predictions <- predict(knn_model, newdata = X_test)

# Calculate predicted probabilities
knn_probabilities <- as.numeric(predict(knn_model, newdata = X_test, type = "
prob")[, "1"])

# Create ROC curve for KNN
roc_knn <- roc(y_test, knn_probabilities)

####################################################
# SVM model for test dataset
####################################################

# Predict on the test set
svm_predictions <- predict(svm_model, newdata = X_test)

# Calculate predicted probabilities
svm_probabilities <- as.numeric(predict(svm_model, newdata = X_test, type = "
response"))

# Create ROC curve for SVM
roc_svm <- roc(y_test, svm_probabilities)

####################################################
# Random Forest model for test dataset
####################################################
```

```
# Predict on the test set
rf_predictions <- predict(rf_model, newdata = X_test)

# Calculate predicted probabilities
rf_probabilities <- predict(rf_model, newdata = X_test, type = "prob")[, "1"]

# Create ROC curve for Random Forest
roc_rf <- roc(y_test, rf_probabilities)

####################################################
# Compute AUC for each model
####################################################

auc_knn <- auc(roc_knn)
auc_svm <- auc(roc_svm)
auc_rf <- auc(roc_rf)

# Print AUC values
cat("AUC for KNN:", auc_knn, "\n")
cat("AUC for SVM:", auc_svm, "\n")
cat("AUC for Random Forest:", auc_rf, "\n")

# Ouput
 # Print AUC values
> cat("AUC for KNN:", auc_knn, "\n")
AUC for KNN: 0.5954301
> cat("AUC for SVM:", auc_svm, "\n")
AUC for SVM: 0.8767921
> cat("AUC for Random Forest:", auc_rf, "\n")
AUC for Random Forest: 0.9153226
```

Results for AUC analysis

Model	AUC Value
KNN	0.5954301
SVM	0.8767921
RFM	0.9153226

Conclucison

- **Model Performance**: Through meticulous data preprocessing and feature engineering, we successfully trained and evaluated multiple models, including Random Forest, SVM, and KNN. The ROC curves vividly showcased the comparative performance of these models, with the Random Forest model emerging as the most robust, surpassing SVM and outperforming KNN.
- **Area Under Curve (AUC)**: The AUC values provided a quantitative measure of each model's ability to discriminate between positive and negative cases. The Random

Forest model demonstrated the highest AUC, underlining its efficacy in capturing intricate patterns within the cervical cancer dataset.

- **Insights for Healthcare Practitioners**:Our results offer valuable insights for healthcare practitioners, empowering them with advanced tools for early diagnosis and risk assessment. The models, particularly the Random Forest model, can serve as decision support systems, aiding medical professionals in identifying individuals at higher risk of cervical cancer.
- **Future Directions**: Continuous refinement and optimization of the models will be pivotal for their integration into clinical workflows.Collaborations with healthcare experts for domain-specific insights and continuous data updates will enhance the models' accuracy and applicability.

Implications for Public Health:

This project signifies a stride towards personalized medicine, where AI/ML models contribute to tailored healthcare solutions. By offering accurate predictions and risk assessments, these models have the potential to improve patient outcomes and contribute to the global effort in combating cervical cancer.

In conclusion, the intersection of AI/ML and healthcare holds tremendous promise for early disease detection and risk mitigation. As we move forward, the ongoing refinement of these models, coupled with collaborative efforts between data scientists and healthcare professionals, will pave the way for a transformative impact on cervical cancer diagnosis and, ultimately, patient care.

References

- Fernandes,Kelwin, Cardoso,Jaime, and Fernandes,Jessica. (2017). Cervical cancer (Risk Factors). UCI Machine Learning Repository. https://doi.org/10.24432/C5Z310.

- Alam, T.M.; Khan, A.; Iqbal, A.; Abdul,W.; Mushtaq, M. Cervical cancer prediction through different screening methods using data mining. Int. J. Adv. Comput. Sci. Appl. 2019, 10, 346–357.

- Novakovi´c, D.; Veljovi´c, A.S.; Ili´c, S.; Papi´c, Ž; Tomovi´c, M. Evaluation of classification models in machine learning. Theory Appl.Math. Comput. Sci. 2017, 7, 39–46.

- Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning; University of Wisconsin: Madison, WI, USA, 2018.

- Rafael A. Irizarry. Introduction to Data Science.https://rafalab.dfci.harvard.edu/dsbook/