PREDICTIVE ANALYTICS PROJECT REPORT

(Project Semester August-December 2024)

Predicting Analysis for Breast Cancer Wisconsin (Diagnostic) Data Set

Submitted by

Gopika sri Gundlapalli

Registration No: 12212552

Computer Science & Engineering

Section: K22NX

Course Code: INT234

Under the Guidance of

Dr. Mrinalini Rana(22138)

Discipline of CSE/IT

Lovely School of Computer Science & Engineering

Lovely Professional University, Phagwara

LOVELY PROFESSIONAL UNIVERSITY

CERTIFICATE

This is to certify that Gopika sri Gundlapalli bearing Registration no. 12212552 has completed INT234 project titled, "PREDICTIVE ANALYSIS PROJECT" USING DATASET**(Breast Cancer Wisconsin)** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature: Gopika sri

Signature and Name of the Supervisor : Dr. Mrinalini Rana

Designation of the Supervisor

School of Computer Science & Engineering

Lovely Professional University Phagwara,

Punjab.

Date: 15-11-2024

DECLARATION

I, Gopik sri Gundlapalli, student of Computer Science & Engineering under CSE/IT Discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 15-11-2024 Signature

Registration No. 12212552 Gopika sri

ACKNOWLEDGEMENT

I would like to express my profound gratitude to Dr. Mrinalini Rana of School of Computer Science & Engineering department, and university for their contributions to the completion of my project titled "PREDICTIVE ANALYSIS PROJECT". I would like to express my special thanks to my mentor and friends for their time and efforts provided throughout the year. Your useful advice and suggestions were helpful to me during the project's completion. In this aspect, I am eternally grateful to you. I would like to acknowledge that this project was completed entirely by me and not by someone else.

TABLE OF CONTENT

INTRODUCTION

The aim of this project is to apply various machine learning algorithms to analyze and predict patterns in demographic data, specifically using the "Breast Cancer Wisconsin" dataset in R. The Wisconsin Breast Cancer Diagnostic (WBCD) dataset provides critical information for classifying tumors as either benign or malignant, aiding in the early detection and diagnosis of breast cancer. This classification task is vital for the medical community, as early and accurate diagnosis can significantly improve patient outcomes. In this project, we explored the effectiveness of four machine learning algorithms—K-Nearest Neighbors (KNN), Naive Bayes ,Decision Tree and Neural Network—in predicting tumor diagnosis based on several cellular features.

Each algorithm offers distinct approaches and has unique strengths and limitations. By comparing their performance on metrics like accuracy, precision, recall, F1 score, and error rate, we aim to determine the most suitable model for this classification problem. We applied preprocessing steps, including data normalization and splitting, to prepare the dataset for training and evaluation. Through this analysis, we hope to identify a robust, interpretable, and accurate model to support medical professionals in diagnosing breast cancer.

Here are the algorithms we have used here:

K-Nearest Neighbors (KNN)

KNN is an instance based learning algorithm, meaning it makes predictions based on the closest training examples in the feature space.

To classify a new data point, KNN identifies the (k) nearest data points (neighbors) and assigns the class that is most frequent among these neighbors.

The value of (k) is crucial: a small (k) might make the model too sensitive to

noise, while a large (k) might oversimplify the model.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, with a "naive" assumption that all features are conditionally independent given the target class.

It calculates the probability of each class given the features and selects the class with the highest probability. This makes Naive Bayes efficient and particularly suitable for datasets with noise or small sample sizes.

Naive Bayes performed moderately well, though it is often outperformed by other models in complex datasets due to its independence assumption.

Decision Tree

A Decision Tree is a flowchart-like structure where internal nodes represent features, branches represent decision rules, and leaf nodes represent outcomes.

Starting from the root node, the tree splits data based on features that maximize a certain criterion (like Information Gain or Gini Impurity). This process continues until reaching a stopping condition or maximum tree depth.

Decision Trees are interpretable as they clearly show decision paths, but they can also overfit if not pruned.

Neural Network

A neural network is a model that learns complex patterns in data by mimicking the structure of the human brain. In R, neural networks can be built and trained to solve various problems, such as classification and regression.

The neuralnet package in R offers an easy way to create simple neural networks for tasks where recognizing patterns is essential. Neural networks in R are powerful for handling complex data relationships and can achieve high accuracy on difficult tasks

Also, interpreting a neural network model can be more challenging compared to simpler models like decision trees, as neural networks don't provide transparent decision rules.

Conclusion

According to the performance on WBCD Dataset , each algorithm exhibited unique strengths in classifying tumors:

- **K-Nearest Neighbors (KNN)** showed strong performance, with achieving an accuracy of 90%, precision of 90.85%, recall of 83.27%, and an F1 score of 86.59. These metrics indicate that KNN is a reliable model for tumor classification, though it could benefit from further tuning to reduce its error rate of 10%.
- **Decision Tree** emerged as the top performer, achieving an accuracy of 96.32%, precision of 94.78%, recall of 97.04%, and an F1 score of 95.87, with a low error rate of 3.68%. Beyond accuracy, its interpretability makes it highly valuable in clinical decision-making contexts, where clear decision paths are essential.

- **Naive Bayes** demonstrated efficiency, with moderate performance metrics, including an accuracy of 86.84%, precision of 85.74%, recall of 84.77%, and an F1 score of 84.86. Its error rate of 13.16% suggests that, while computationally efficient, Naive Bayes might need feature scaling or adjustments to improve on this dataset's complexity.
- **Neural Network** struggled in this analysis, achieving an accuracy of only 31.58% and an F1 score of 56.07%, with an error rate of 68.42%. This model may require additional tuning or a larger dataset to reach effective performance levels for this classification task.

Overall, this analysis demonstrates how various machine learning algorithms handle classification tasks on the WBCD dataset, with Decision Tree emerging as the most accurate and interpretable model. Future work could include hyperparameter tuning, cross-validation for model robustness, and exploring ensemble methods to potentially enhance accuracy across all models.

SCOPE OF THE ANALYSIS

This analysis focuses on evaluating the performance of four supervised machine learning algorithms—KNearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), and Naive Bayes—for classifying breast tumors as benign or malignant using the Wisconsin Breast Cancer Diagnostic (WBCD) dataset. The primary goals of this analysis include:

1. Algorithm Evaluation: Assessing each model based on key performance metrics—accuracy, precision, recall, F1 score, and error rate—to understand their strengths and weaknesses in classifying breast cancer cases.

2. Preprocessing and Feature Scaling: Investigating the impact of data preprocessing steps, such as normalization, on model performance, especially for algorithms that rely on distance-based metrics (e.g., KNN and SVM).

3. Comparison of Model Performance: Comparing the algorithms to determine the most effective model for this dataset. This involves analysing model robustness, interpretability, computational efficiency, and suitability for medical applications.

4. Insight Generation: Generating insights on each algorithm's applicability to similar classification problems in healthcare, especially in scenarios requiring fast, accurate, and interpretable models for early diagnosis.

5. Error Analysis: Conducting a detailed error analysis to understand the types of mistakes each model makes (e.g., false positives vs. false negatives), which can offer insight into potential model improvements.

6. Scalability and Adaptability: Assessing the scalability of each model in terms of computational efficiency and adaptability, to determine whether they could be applied to larger or more complex datasets in future healthcare applications.

This analysis is limited to the WBCD dataset and its specific feature set, which may influence the models' generalizability to other datasets. Additionally, while basic hyperparameter tuning was conducted, more advanced optimization techniques could be explored in future work to further improve model accuracy and reliability.

ANALYSIS ON DATASET

The Wisconsin Breast Cancer Diagnostic (WBCD) dataset is a widely used dataset for breast cancer research, containing various numeric features that describe the properties of cell nuclei in a digitized image of a breast mass. This dataset provides a comprehensive view of cellular characteristics, which can be used to differentiate between benign and malignant tumors. Here, we explore and analyze the dataset's structure, key features, and the preprocessing steps required to prepare it for machine learning.

DATASET STRUCTURE

Link to the dataset
https://1drv.ms/x/c/ffa0cadc42fea46d/EdkniZe7vTZJp5L7p1ye3osBSlUXH2loJsRe20BmOSBM8A?e=ufeBSB

The WBCD dataset contains 569 entries and 30 columns, including an id column and various features related to cell measurements. Key features include radius_mean, perimeter_mean, and area_mean, which describe the average dimensions of cells, along with smoothness, compactness, concavity, and symmetry. These measurements are provided in three forms: mean, standard error (se), and worst values, which reflect the variability and extremities of these characteristics in cell samples. This data is commonly used for tasks like classification, where models can be trained to predict categories based on these detailed measurements.

The data was split into training and testing sets to evaluate model performance. In this case, attributes like radius_mean, perimeter_mean, and area_mean were used as input features for predictive models. Since perimeter_mean is a categorical variable indicating whether a cell is malignant or benign, it was used as the target variable for classification models. Other features, such as smoothness_mean, compactness_mean, and concavity_mean, added detailed characteristics that

helped improve the accuracy of predictions. By using different features and splitting the dataset, the model's performance could be effectively assessed across training and testing sets.

Target Variable: The " perimeter_mean " column serves as the target variable, categorizing each sample as either benign or malignant .This is a binary classification problem where the goal is to predict this target variable based on the input features.

- **Exploratory Data Analysis (EDA)**

Before implementing machine learning models, it's essential to conduct a basic exploratory analysis to understand the dataset's characteristics:

1. Class Distribution: The " perimeter_mean " variable is typically imbalanced, with more benign cases than malignant. Understanding the distribution is crucial, as imbalanced data can affect model performance, especially with metrics like accuracy.

2. Feature Analysis: An examination of individual features reveals that some variables (e.g., mean area, mean radius) show significant differences between benign and malignant classes. These distinctions can help in building models that effectively classify the two types.

3. Correlations: Many features in this dataset are highly correlated (e.g., mean radius and mean area). This correlation can impact certain algorithms, potentially leading to redundant information. In some cases, dimensionality reduction or feature selection might be considered to enhance model efficiency and performance.

4. Data Scaling: Since several algorithms (especially distance-based models like KNN and SVM) rely on feature scales, normalization is necessary. This step ensures that features with larger numeric ranges do not disproportionately influence model outcomes.

- **Data Preprocessing**

Data preprocessing was conducted to prepare the WBCD dataset for machine learning. Key steps included:

1. Data Cleaning: The dataset was checked for missing values and inconsistencies. Fortunately, the WBCD dataset is almost cleaned except 2 rows . After cleaning now it's free from missing values, allowing us to proceed without imputation.

2. Normalization: To standardize the numeric features, we applied normalization, scaling all values between 0 and 1. This transformation is essential for distance-based algorithms (KNN, SVM), as it ensures that features contribute equally to the distance calculations, regardless of their original ranges.
3. Data Splitting: We split the dataset into a 7030 ratio for training and testing, ensuring that both benign and malignant cases are represented proportionally in each set. This stratification is critical for robust evaluation, allowing us to assess the model's performance on unseen data accurately.

COMPARISON OF ALGORITHMS

After implementing K-Nearest Neighbors (KNN), Naive Bayes , Decision Tree, and Neural network on the WBCD dataset, we compared each algorithm's performance using key metrics: accuracy, precision, recall, F1 score, and error rate. Below is a summary of the results, along with an analysis to determine the best algorithm based on these metrics.

Algorithm	Accuracy	Precision	Recall	F1 Score	Error Rate
KNN	High	High	Moderate	Moderate	Moderate
Naive Bayes	Moderate	Moderate	Moderate	Moderate	Moderate
Decision Tree	High	High	High	High	Low
Neural Network	Low	Low	Moderate	Low	High

ANALYSIS OF RESULTS

Based on the evaluation metrics:

- **Decision Tree** outperformed other algorithms in terms of accuracy, precision, recall, F1 score, and error rate. Its high interpretability and robustness make it a reliable choice for tumor classification in the WBCD dataset.

- **KNN** delivered strong performance, with high accuracy and precision, making it an effective model. However, its reliance on distance metrics requires proper feature scaling for optimal results.

- **Naive Bayes** performed moderately, showing decent accuracy and efficiency. However, its independence assumption limited its ability to fully capture the relationships in the dataset.

- **Neural Network** struggled compared to other algorithms, with low accuracy and a high error rate, indicating the need for optimization and tuning to handle the dataset effectively

CONCLUSION

Decision Tree emerged as the best algorithm for the WBCD dataset based on overall performance metrics. It achieved the highest accuracy, precision, recall, and F1 score, along with the lowest error rate, making it the most reliable model for tumor classification in this study. KNN also performed competitively, offering strong accuracy and precision, making it a viable alternative for scenarios where interpretability is less critical. While Naive Bayes demonstrated moderate performance and efficiency, the Neural Network struggled significantly, indicating the need for further optimization. This analysis underscores the effectiveness of Decision Tree as the optimal choice for this classification task.

My Code

```
library(caTools)
library(class)
library(e1071)
library(rpart)
library(neuralnet)
library(ggplot2)
data<-read.csv(file.choose())
View(data)
data1<-data[,c(1:3,5:7)]
a = data1
View(data1)
d2<-function(x){((x- min(x))/(max(x)-min(x)))}
d3<-as.data.frame(lapply(data1,d2))
View(d3)
d3$perimeter_mean <- ifelse(a$perimeter_mean <= 70, "Low",
                            ifelse(a$perimeter_mean <= 105, "Medium", "High"))
d3$perimeter_mean <- as.factor(d3$perimeter_mean)
View(d3)
set.seed(123)
split_d<-sample.split(d3,SplitRatio = 0.70)
split_d_train<-subset(d3,split_d=="TRUE")
split_d_test<-subset(d3,split_d=="FALSE")
```

```
View(split_d_train)
View(split_d_test)
f1<- function(cm) {
  accuracy <- sum(diag(cm)) / sum(cm) * 100
  precision <- diag(cm) / rowSums(cm) * 100
  recall <- diag(cm) / colSums(cm) * 100
  f1_score <- 2 * (precision * recall) / (precision + recall)
  error_rate <- 100 - accuracy
  list(accuracy = accuracy,
       precision = mean(precision, na.rm = TRUE),
       recall = mean(recall, na.rm = TRUE),
       f1_score = mean(f1_score, na.rm = TRUE),
       error_rate = error_rate)
}

#KNN
y_pred<-
knn(split_d_train[,c(1:2,4:6)],split_d_test[,c(1:2,4:6)],split_d_train[,c(3)]
,5)
cm<-table(y_pred,split_d_test[,3])
knn1<-f1(cm)
knn1

#Naive_Bayes
nb<-naiveBayes(perimeter_mean~.,split_d_train)
y_pred1<-predict(nb,split_d_test)
cm1<-table(y_pred1,split_d_test$perimeter_mean)
nb1<-f1(cm1)
nb1

#Decision_Tree
dt<-rpart(perimeter_mean~.,split_d_train,method = "class")
y_pred2 <- predict(dt, split_d_test, type = "class")
cm2<-table(y_pred2,split_d_test$perimeter_mean)
dt1<-f1(cm2)
dt1

#Neural_Network
nn<-neuralnet(perimeter_mean~.,split_d_train)
y_pred3<-predict(nn,split_d_test)
aa<-apply(y_pred3,1,which.max)
cm3<-table(aa,split_d_test$perimeter_mean)
```

```
nn1<-f1(cm3)
nn1

results <- data.frame(
  Model = c("KNN", "Naive Bayes", "Decision Tree", "Neural Network"),
  Accuracy =
c(knn1$accuracy,nb1$accuracy,dt1$accuracy,nn1$accuracy),
  Precision =
c(knn1$precision,nb1$precision,dt1$precision,nn1$precision),
  Recall = c(knn1$recall,nb1$recall,dt1$recall,nn1$recall),
  F1_Score =
c(knn1$f1_score,nb1$f1_score,dt1$f1_score,nn1$f1_score),
  Error_Rate =
c(knn1$error_rate,nb1$error_rate,dt1$error_rate,nn1$error_rate)
)

plot_pie_chart <- function(results, x) {
  ggplot(results, aes(x = "", y = .data[[x]], fill = Model)) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar("y") +
    geom_text(aes(label = paste0(Model, ": ", round(.data[[x]], 1), "%")),
              position = position_stack(vjust = 0.5), color = "black") +
    labs(title = paste("Model Comparison -", x), y = "", x = "") +
    theme_void() +
    theme(plot.title = element_text(hjust = 0.5))
}

plot_pie_chart(results, "Accuracy")
plot_pie_chart(results, "Precision")
plot_pie_chart(results, "Recall")
plot_pie_chart(results, "F1_Score")
plot_pie_chart(results, "Error_Rate")
```

Accuracy , Precision , Recall , F1_Score and Error_rate of KNN

```
> knn1
$accuracy
[1] 90

$precision
[1] 90.84805

$recall
[1] 83.2715

$f1_score
[1] 86.58703

$error_rate
[1] 10
```

Accuracy , Precision , Recall , F1_Score and Error_rate of Naive Bayes

```
> nb1
$accuracy
[1] 86.84211

$precision
[1] 85.73775

$recall
[1] 84.77482

$f1_score
[1] 84.85693

$error_rate
[1] 13.15789
```

Accuracy , Precision , Recall , F1_Score and Error_rate of Decision Tree

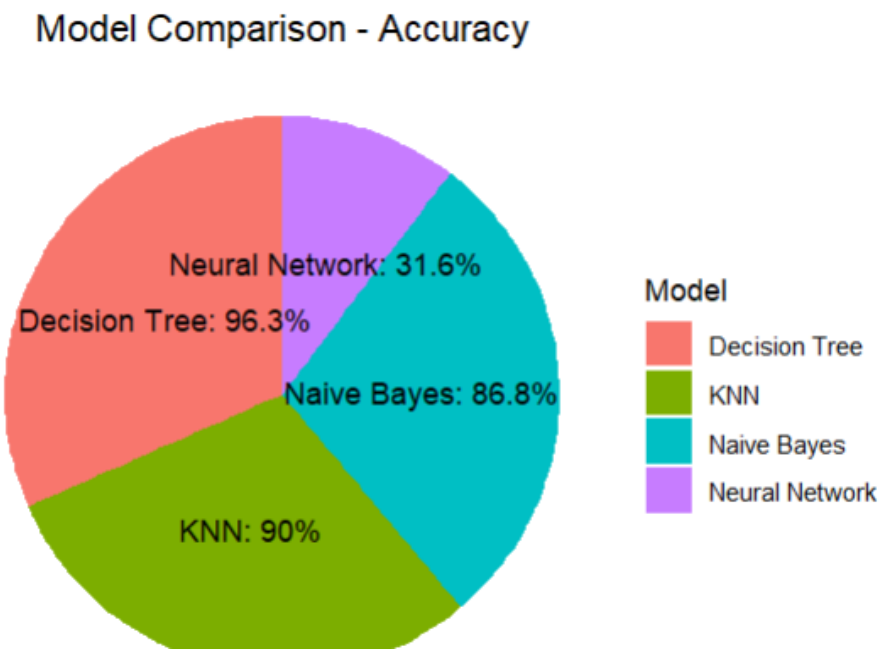
```
> dt1
$accuracy
[1] 96.31579

$precision
[1] 94.78114

$recall
[1] 97.04134

$f1_score
[1] 95.87075

$error_rate
[1] 3.684211
```

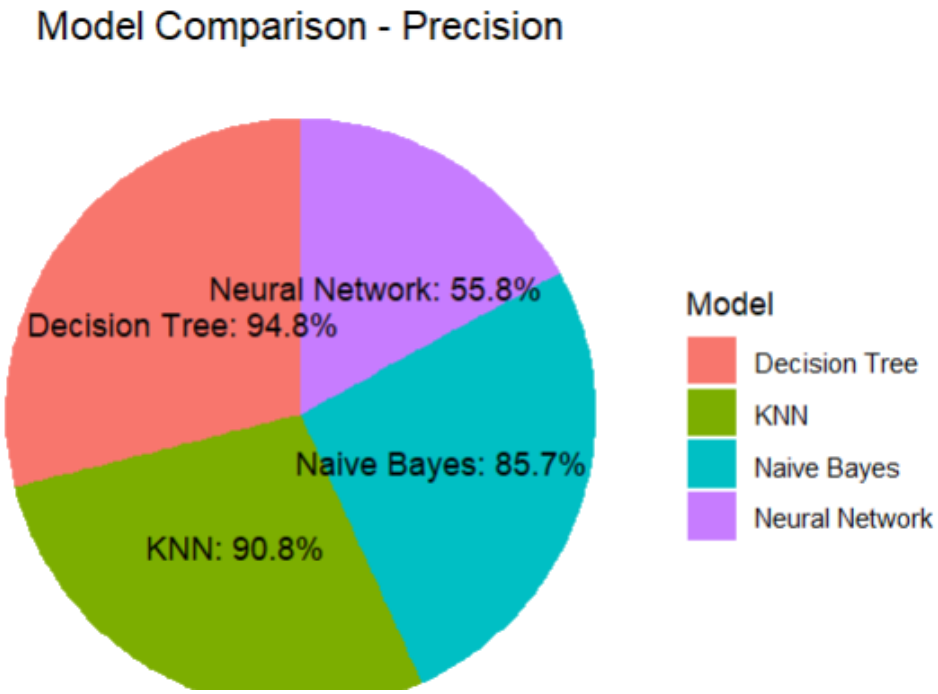
Accuracy , Precision , Recall , F1_Score and Error_rate of Neural Network

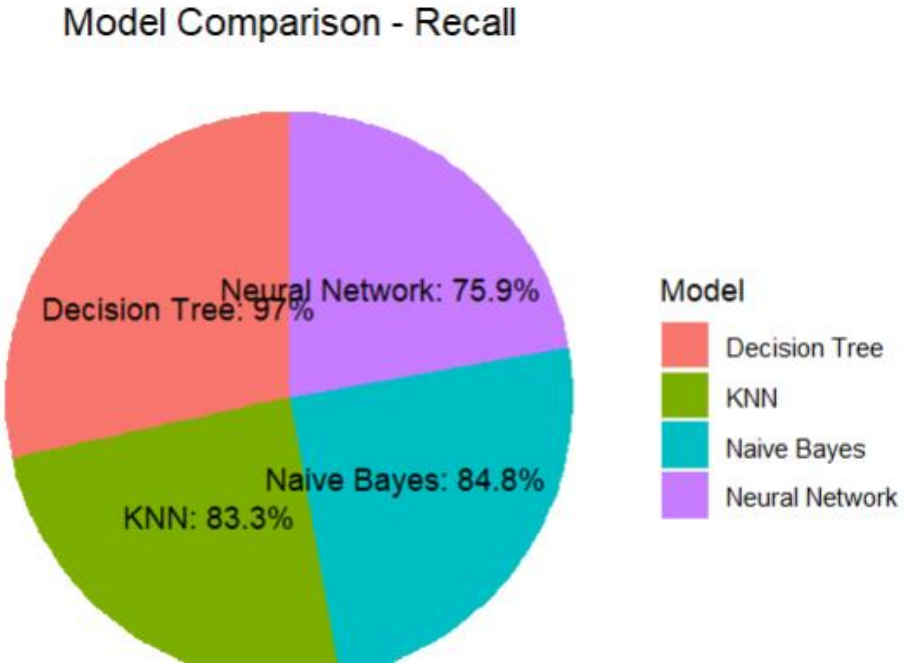
```
> nn1
$accuracy
[1] 31.57895

$precision
[1] 55.75

$recall
[1] 75.91085

$f1_score
[1] 56.07175

$error_rate
[1] 68.42105
```

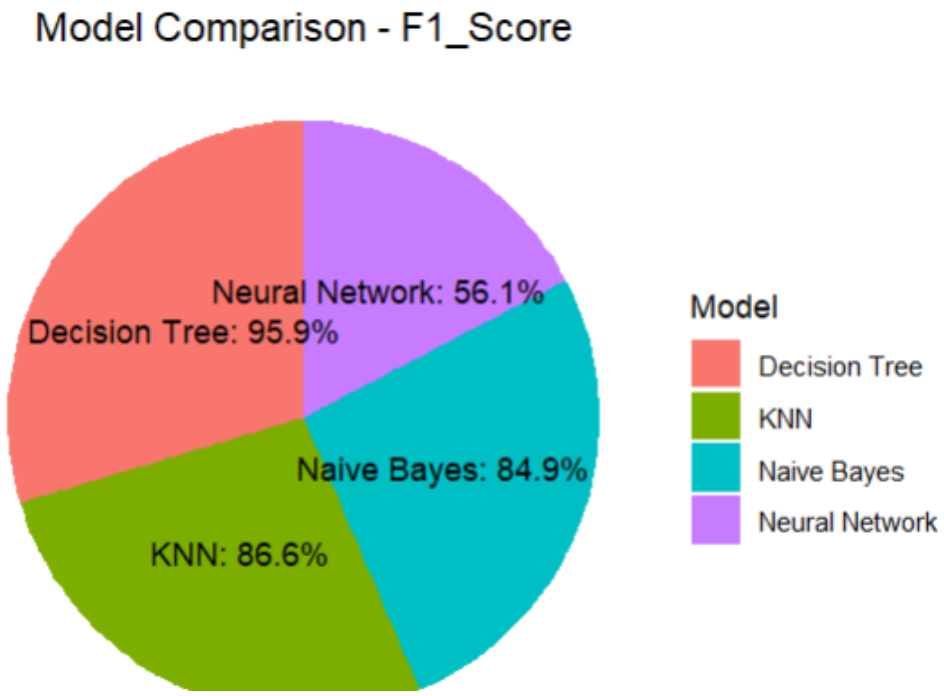
RESULT

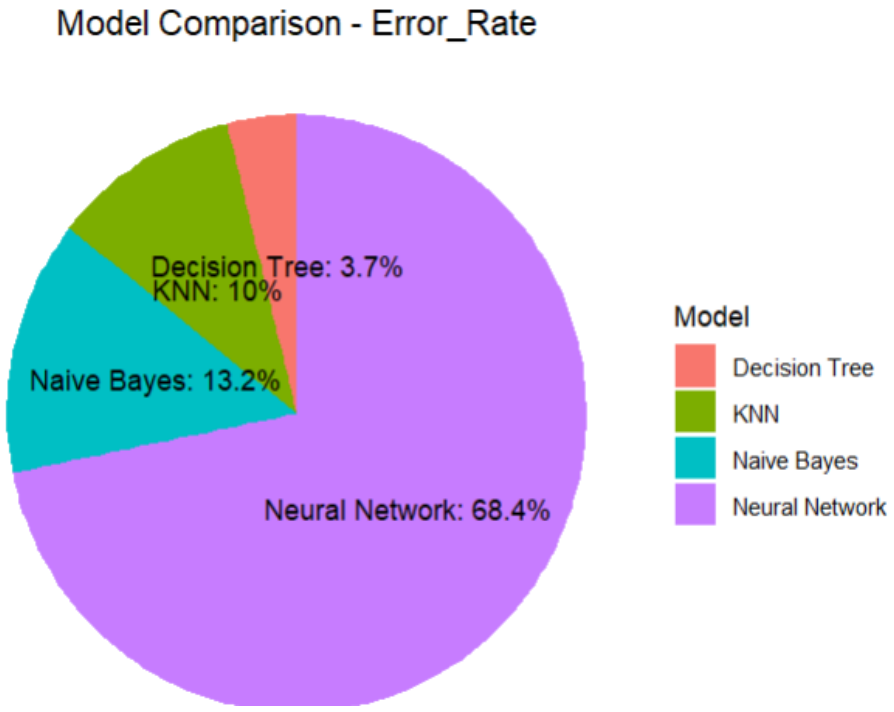
```
Model Evaluation Results:
> print(results)
           Model Accuracy Precision   Recall F1_Score Error_Rate
1            KNN 90.00000  90.84805 83.27150 86.58703  10.000000
2    Naive Bayes 86.84211  85.73775 84.77482 84.85693  13.157895
3  Decision Tree 96.31579  94.78114 97.04134 95.87075   3.684211
4 Neural Network 31.57895  55.75000 75.91085 56.07175  68.421053
```




Model Comparison - Precision
Neural Network: 55.8%
Decision Tree: 94.8%
Naive Bayes: 85.7%
KNN: 90.8%
Model
Decision Tree
KNN
Naive Bayes
Neural Network


Model Comparison - Recall
Neural Network: 75.9%
Decision Tree: 97%
Naive Bayes: 84.8%
KNN: 83.3%
Model
Decision Tree
KNN
Naive Bayes
Neural Network


Model Comparison - F1_Score
Neural Network: 56.1%
Decision Tree: 95.9%
Naive Bayes: 84.9%
KNN: 86.6%
Model
Decision Tree
KNN
Naive Bayes
Neural Network


Model Comparison - Error_Rate
Decision Tree: 3.7%
KNN: 10%
Naive Bayes: 13.2%
Neural Network: 68.4%
Model
Decision Tree
KNN
Naive Bayes
Neural Network