**Data Science and Machine Learning**

**CS3203**

Text

Description automatically generated

**Diabetes Prediction Using K-Nearest Neighbor Algorithm**

**Prepared By: Ansh Ujlayan**

**Submitted To: Ms. Shikha Mundra**

**Registration No.: 209301050**

**Batch: G1**

**Branch /Sem: B. tech CSE G /VI**

**Dehmi Kalan, Jaipur-Ajmer Expressway, Jaipur - 303 007,**

**Rajasthan, India, Phone: 0141-3999100**

**Jaipur.manipal.edu**

Table of Contents

[Introduction 3](#_Toc132490966)

[Dataset Description 4](#_Toc132490967)

[Algorithm Used 5](#_Toc132490968)

[Data Visualization 6](#_Toc132490969)

[1. Bar Chart 6](#_Toc132490970)

[2. Correlation Matrix 6](#_Toc132490971)

[Result Analysis 7](#_Toc132490972)

[1. Confusion Matrix 7](#_Toc132490973)

[2. ROC 10](#_Toc132490974)

[3. Evaluation 11](#_Toc132490975)

# Introduction

Diabetes is a chronic disease that affects millions of people worldwide. It is characterized by high levels of blood glucose resulting from defects in insulin production, insulin action, or both. Diabetes can lead to serious complications, including heart disease, stroke, blindness, kidney failure, and lower limb amputations. Early detection and management of diabetes are critical in preventing these complications and improving the quality of life of people with diabetes.

In this project, we aim to develop a machine learning model to predict the presence of diabetes in a person based on various clinical features such as Glucose, Blood Pressure, Skin Thickness, BMI, Insulin, Age, etc. The model will use the K-Nearest Neighbours (KNN) algorithm, which is a type of supervised learning algorithm used for classification and regression analysis.

The objective of this project is to provide a tool that can assist healthcare professionals in early identification of diabetes and provide personalized care to individuals with diabetes. The model can also be useful for individuals to assess their risk of developing diabetes and take preventive measures.

The dataset used in this project is the Pima Indians Diabetes Database, which is publicly available and contains clinical data of women of Pima Indian heritage aged 21 years or older. The dataset contains 768 observations with eight clinical features and one target variable indicating the presence or absence of diabetes. We will pre-process the dataset, split it into training and testing sets, and use the KNN algorithm to train and evaluate the model.

# Dataset Description

The dataset used in this project is the Pima Indians Diabetes Database, originally from the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset was collected from females of Pima Indian heritage who were at least 21 years old living near Phoenix, Arizona, USA. The data was collected from 768 female patients, with several medical predictor variables and one target dependent variable.

The medical predictor variables include:

Pregnancies: the number of times the patient has been pregnant.

Glucose: plasma glucose concentration a 2 hours in an oral glucose tolerance test.

BloodPressure: diastolic blood pressure (mm Hg).

SkinThickness: triceps skin fold thickness (mm).

Insulin: 2-Hour serum insulin (mu U/ml).

BMI: body mass index (weight in kg/(height in m)^2).

DiabetesPedigreeFunction: diabetes pedigree function, which represents the genetic influence of diabetes.

Age: age of the patient (years).

The target variable, Outcome, is a binary variable indicating whether the patient has diabetes or not. A value of 1 means the patient has diabetes, and a value of 0 means the patient does not.

It is important to note that the dataset has some missing values, particularly in columns such as Glucose, BloodPressure, SkinThickness, BMI, and Insulin. To handle these missing values, we have replaced them with the mean of their respective columns.

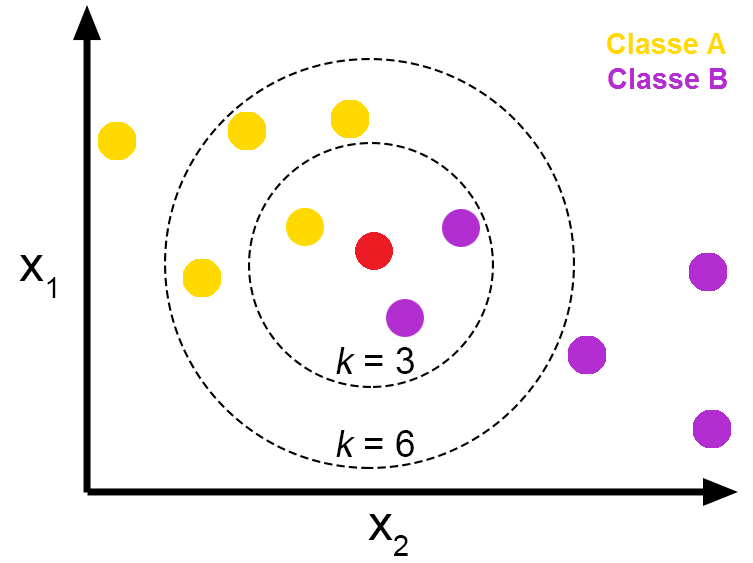
The objective of this project is to use this dataset to diagnostically predict whether a patient has diabetes based on the given predictor variables.

# Algorithm Used

K-Nearest Neighbor (KNN) is a popular classification algorithm that is widely used in machine learning for both regression and classification problems. The KNN algorithm is based on the concept that similar data points tend to belong to the same class. It is a non-parametric and lazy learning algorithm, meaning that it does not make any assumptions about the data and only stores the training data points.

In KNN, the k nearest neighbors to the input data point are considered, and the output class is determined based on the majority vote of those k neighbors. The distance between the input data point and its k neighbors is calculated using various distance metrics, such as Euclidean distance, Manhattan distance, and Minkowski distance.

One of the advantages of the KNN algorithm is its simplicity, as it requires only a few parameters to be set, namely the value of k and the distance metric. Additionally, it can be used for both binary and multi-class classification problems. However, KNN may not perform well on high-dimensional data due to the "curse of dimensionality" and can be sensitive to irrelevant features. It is also computationally expensive to use on large datasets.



*Figure 1: KNN*

# Data Visualization

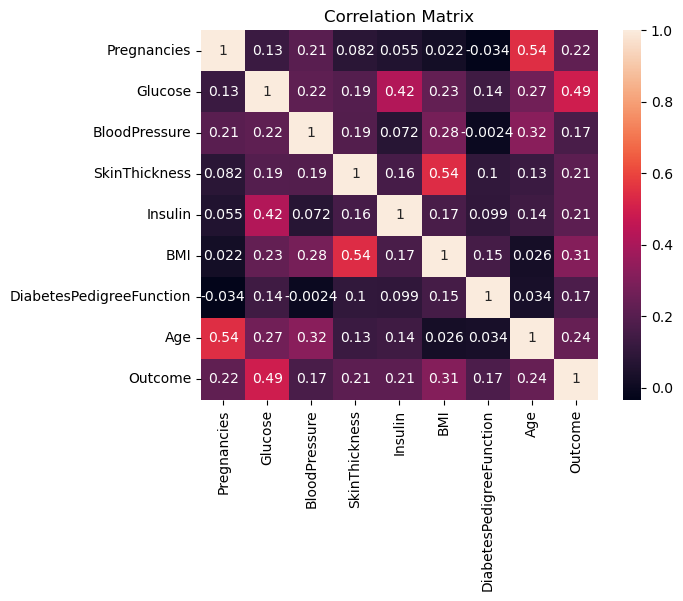
## Bar Chart

Chart

Description automatically generatedBased on the bar graph depicting the correlation between age and diabetes, it appears that the likelihood of having diabetes increases with age. The graph shows that for those in their 20s, the percentage of individuals with diabetes is relatively low, but steadily increases as age groups advance, reaching a peak in the 50-60 age group, before declining slightly in the 60s and 70s. This trend suggests that age is a significant factor in predicting diabetes, and older individuals should be more cautious and attentive to their health to avoid the onset of diabetes.

*Figure 2: Bar Graph*

## Correlation Matrix

The correlation matrix shows that age has a weak positive correlation with the target variable "Outcome", which means that as age increases, the likelihood of having diabetes slightly increases. However, the correlation coefficient is relatively small (0.24), indicating a weak linear relationship between age and diabetes. Therefore, while age

*Figure 3: Correlation Matrix*

may have some impact on diabetes risk, it is not a strong predictor and other factors may have a greater influence.

# Result Analysis

## Confusion Matrix

For 80-20 test train split:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted/Classified** | |
|  |  | **Negative** | **Positive** |
| **Actual** | **Negative** | 94 | 13 |
| **Positive** | 15 | 32 |

*Table 1: 80-20 split*

### Accuracy

Accuracy = (TP+TN)/(TP+TN+FP+FN) = **0.81**

### Precision

Precision = TP/(FP+TP) = **0.71**

### Recall / Sensitivity

Recall = TP/(FN+TP) = **0.68**

### F1-Measure

F1-Measure = 2 \* (Precision \* Recall) / (Precision +Recall) = **0.70**

### Specificity

Specificity = TN/N = **0.88**

### False Positive Rate (FPR)

FPR = FP/N = **0.12**

### False Negative Rate (FNR)

FNR = FN/P = **0.32**

### Negative Predictive Value (NPV)

NPV = TN/(TN+FN) = **0.86**

### False Discovery Rate (FDR)

FDR = FP/(FP+TP) = **0.29**

### Matthews’s correlation coefficient (MCC)

MCC = (TP\*TN) -(FP\*FN) /SQRT((TP+FP) (TP+FN) (TN+FP) (TN+FN)) = **0.57**

For 70-30 train test split:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted/Classified** | |
|  |  | **Negative** | **Positive** |
| **Actual** | **Negative** | 94 | 13 |
| **Positive** | 15 | 32 |

*Table 2: 70-30 split*

### Accuracy

Accuracy = (TP+TN)/(TP+TN+FP+FN) = **0.76**

### Precision

Precision = TP/(FP+TP) = **0.65**

### Recall / Sensitivity

Recall = TP/(FN+TP) = **0.55**

### F1-Measure

F1-Measure = 2 \* (Precision \* Recall) / (Precision +Recall) = **0.60**

### Specificity

Specificity = TN/N = **0.86**

### False Positive Rate (FPR)

FPR = FP/N = **0.14**

### False Negative Rate (FNR)

FNR = FN/P = **0.45**

### Negative Predictive Value (NPV)

NPV = TN/(TN+FN) = **0.80**

### False Discovery Rate (FDR)

FDR = FP/(FP+TP) = **0.35**

### Matthews’s correlation coefficient (MCC)

MCC = (TP\*TN) -(FP\*FN) /SQRT((TP+FP) (TP+FN) (TN+FP) (TN+FN)) = **0.43**

For 60-40 train test split:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted/Classified** | |
|  |  | **Negative** | **Positive** |
| **Actual** | **Negative** | 94 | 13 |
| **Positive** | 15 | 32 |

*Table 3: 60-40 split*

### Accuracy

Accuracy = (TP+TN)/(TP+TN+FP+FN) = **0.74**

### Precision

Precision = TP/(FP+TP) = **0.65**

### Recall / Sensitivity

Recall = TP/(FN+TP) = **0.50**

### F1-Measure

F1-Measure = 2 \* (Precision \* Recall) / (Precision +Recall) = **0.57**

### Specificity

Specificity = TN/N = **0.86**

### False Positive Rate (FPR)

FPR = FP/N = **0.14**

### False Negative Rate (FNR)

FNR = FN/P = **0.50**

### Negative Predictive Value (NPV)

NPV = TN/(TN+FN) = **0.78**

### False Discovery Rate (FDR)

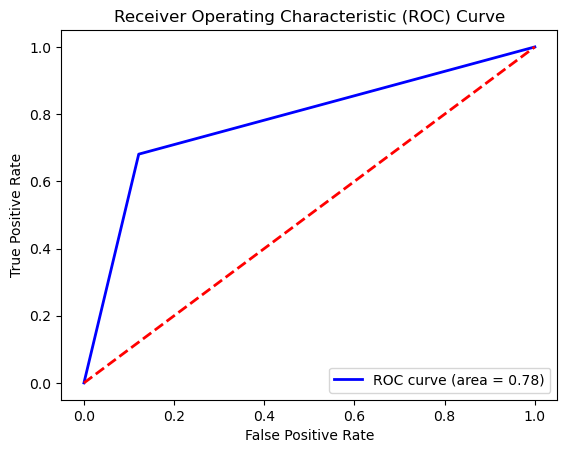
FDR = FP/(FP+TP) = **0.35**

### Matthews’s correlation coefficient (MCC)

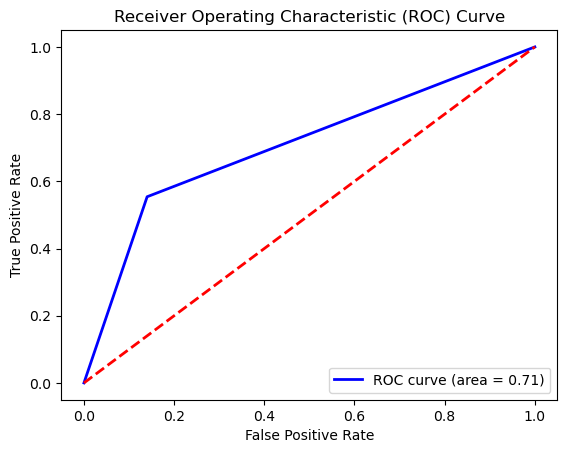
MCC = (TP\*TN) -(FP\*FN) /SQRT((TP+FP) (TP+FN) (TN+FP) (TN+FN)) = **0.40**

## ROC

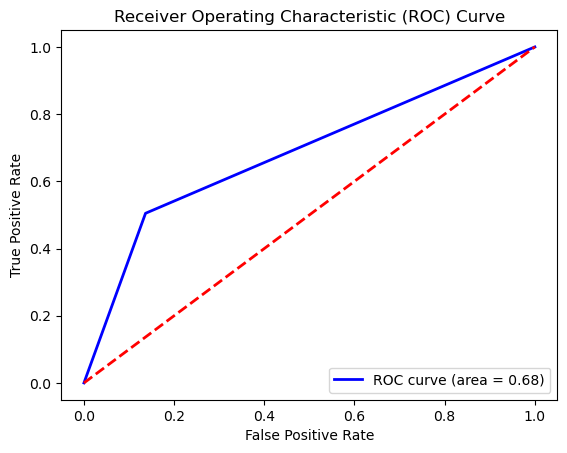
The ROC areas for the 80-20, 70-30, and 60-40 train-test splits were 0.78, 0.71, and 0.68, respectively. The 80-20 split resulted in the highest ROC area, indicating better performance of the model. However, the difference in the ROC areas for the three splits was not very significant, suggesting that the model's performance is relatively stable across different train-test splits.

For 80-20 train test split:

*Figure 4: 80-20 split*

For 70-30 train test split:

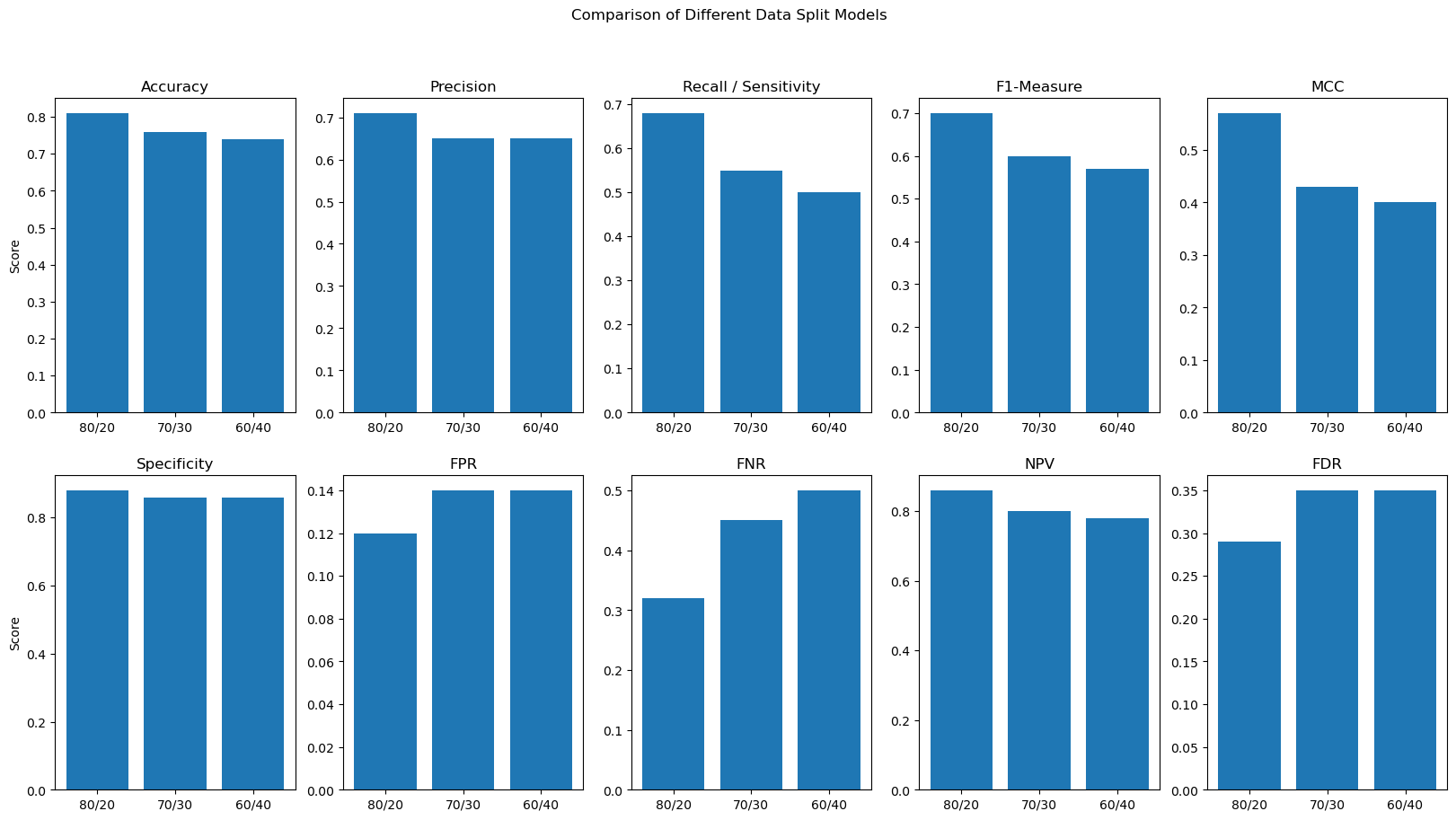
*Figure 5: 70-30 split*

For 60-40 train test split:

*Figure 6: 60-40 split*

## Evaluation

Based on the measures for three different train test ratios, it can be observed that the performance of the algorithm decreases as the ratio of training data decreases. The specificity values remain constant across all ratios, while the sensitivity, accuracy, precision, F1-score, and MCC decrease with the decrease in the training data. The false positive rate (FPR), false negative rate (FNR), negative predictive value (NPV), and false discovery rate (FDR) remain constant with minor variations across different ratios. In conclusion, it is recommended to use a higher training data ratio to achieve better performance in terms of the evaluated measures.



*Figure7: Overall evaluation of the 3 train-test models*

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithms→**  **Measures↓** | **Train test ratio 1**  **(80-20)** | **Train test ratio 2**  **(70-30)** | **Train test ratio 3**  **(60-40)** |
| **Specificity** | 0.88 | 0.86 | 0.86 |
| **Sensitivity** | 0.68 | 0.55 | 0.50 |
| **Accuracy** | 0.81 | 0.76 | 0.74 |
| **Precision** | 0.71 | 0.65 | 0.65 |
| **FPR** | 0.12 | 0.14 | 0.14 |
| **FNR** | 0.32 | 0.45 | 0.50 |
| **NPV** | 0.86 | 0.80 | 0.78 |
| **FDR** | 0.29 | 0.35 | 0.35 |
| **F1- Score** | 0.70 | 0.60 | 0.57 |
| **MCC** | 0.57 | 0.43 | 0.40 |

*Table 5: Overall Performance of the 3 train-test split models*