Grp B- Experiment-5

Aim:- Implement KNN algo^m on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on given dataset.

Theory:-

KNN Algo^m:-

KNN is one of the simplest ML algo^m based on supervised ML technique. KNN algo^m assumes the similarity bet^n the new case/data and available cases and put the new case into the category that is most similar to the available categories. KNN algo^m stores all the available data and classifies a new data point based on similarity.

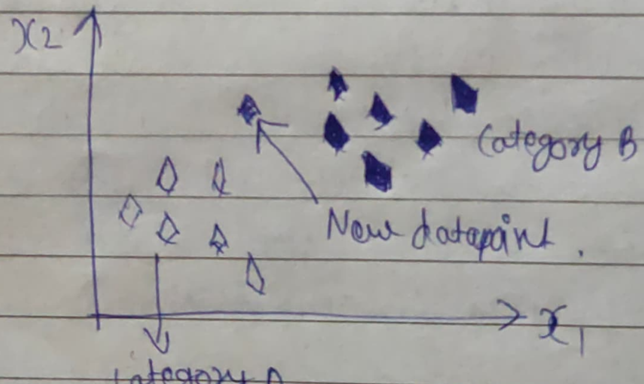
KNN algo^m can be used for Regression as well as classification but mostly it is used for the classification problems. KNN algo^m at the training phase just stores the dataset and when it gets new data, then it classifies that data into category that is much similar to new data.

Example:- Suppose, we have an image of creature that looks similar to cat and dog, but we want to know either it is cat or dog. So for this identification we can use KNN algo^m, as it works on a similarity measure. Our KNN ~~tot~~ model will find the similar ~~measor~~ features of the new dataset to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

The KNN working can be explained on the basis of the below algo^m.

1. Select the no. k of the neighbors.
2. Calculate the Euclidean distance of K no. of neighbors.
3. Take the nearest neighbors as per the calculated Euclidean distance.
4. Among these k neighbors, count the no. of data points in each category.
5. Assign the new datapoints to that category for which the no. of neighbors is max^m.
6. Our model is ready.

Suppose we have ~~a~~ new datapoint and we need to put it in required category.



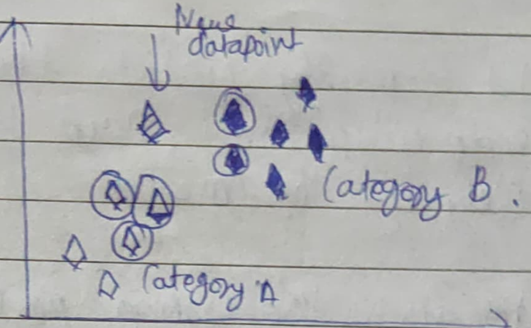
- Firstly, we will choose the no. of neighbors, so we will choose K=5.

- Next we will calculate distance bet^n datapoints.

Formula for Euclidean Distance

$$D = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}.$$

By Calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and ~~to~~ two nearest neighbors in Category B.



As we can see the 3 nearest neighbors are from category A, hence this new datapoint must belong to category A.

Confusion Matrix:-

It is a matrix of size 2x2 for binary classification with actual values on one axis and predicted on another.

		Actual	
		Negative	Positive
Predicted	Negative	True -ve	False -ve
	Positive	False +ve	True +ve

Example:- A ML model Trained to predict tumor in patients. The test dataset consists of 100 people.

		Actual	
		Negative	Positive
Predicted	Negative	60	8
	Positive	22	10

True +ve :- model correctly predicts the +ve class. In above example, 10 people who have tumors are predicted positively by the model.

True -ve :- Model correctly predicts the -ve class. In example, 60 people who don't have tumors are predicted negatively by the model.

False +ve - Model gives the wrong prediction of -ve class. In example, 22 people are predicted as positive of having a tumor, although they don't have tumor.

False -ve :- model wrongly predicts class. In example, 8 people who have tumors are predicted as -ve.

With the help of this 4 values we cancalculate True +ve rate (TPR), False -ve Rate (FPR), True Negative rate (TNR), False -ve Rate (FNR).

$$TPR = \frac{TP}{\text{Actual +ve}} = \frac{TP}{TP+FN}$$

$$FNR = \frac{FN}{\text{Actual +ve}} = \frac{FN}{TP+FN}$$

$$TNR = \frac{TN}{\text{Actual -ve}} = \frac{TN}{TN+FP}$$

$$FPR = \frac{FP}{\text{Actual -ve}} = \frac{FP}{TN+FP}$$

Precision:- Out of all positive predicted, what percentage is truely positive.

$$\text{Precision} = \frac{TP}{TP+FP}$$

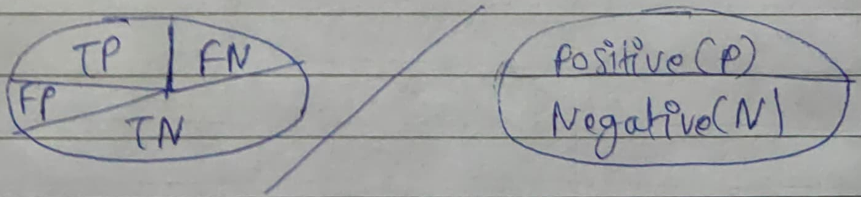
It lies bet" 0 and 1.

Recall:- Out of total +ve, what percentage is predicted +ve.

$$\text{Recall} = \frac{TP}{TP+FN}$$

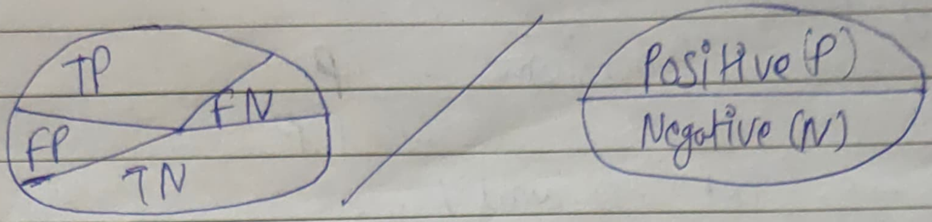
Error rate (ERR):- It is calculated as the no. of all incorrect predictions (FN+FP) divided by the total no. of dataset (P+N). The best error rate is 0.0, whereas the worst is 1.0.

$$\text{Error rate} = \frac{FP+FN}{(P+N)}$$



Accuracy :- ACC is calculated as the no. of all correct predictions divided by total no. of dataset. The best accuracy is 1.0, whereas the worst is 0.0, It can also be calculated by 1 - ERR.

Accuracy :- (TP+TN) / (P+N)



$$ACC = \frac{TP + TN}{TP+TN+FN+FP} = \frac{TP+TN}{P+N},$$

Conclusion :- In this way we have successfully applied KNN algo^{m} on Diabetes dataset and compute confusion matrix, accuracy, error rate, precision and recall on the given data set.