A Project Report submitted

For Artificial Intelligence (UCS411)

by

Kunwar Apoorvaditya 102003452

Submitted to

Niyaz Wani

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BE UNIVERSITY),

PATIALA, PUNJAB

INDIA

May 2022

BREAST CANCER DETECTION

Objective

Women are seriously threatened by breast cancer with high morbidity and mortality. The lack of robust prognosis models results in difficulty for doctors to prepare a treatment plan that may prolong patient survival time. Hence, the requirement of time is to develop the technique which gives minimum error to increase accuracy.

Breast Cancer is the most affected disease present in women worldwide. 246,660 of women's new cases of invasive breast cancer are expected to be diagnosed in the U.S during 2016 and 40,450 of women's death is estimated. The development in Breast Cancer and its prediction fascinated. The UCI Wisconsin Machine Learning Repository Breast Cancer Dataset attracted as large patients with multivariate attributes were taken as sample set.

The diagnosis of breast cancer is carried out by classifying the tumor. Tumors can be either benign or malignant. In a benign tumor, the cells grow abnormally and form a lump and do not spread to other parts of the body. Malignant tumors are more harmful than benign as these tumors tend to spread all over the body if not treated in the right time.

Machine Learning technology has developed so much that we can now help the community by detecting breast cancer on the basis of various features such as radius, texture, compactness etc. We have used the K-nearest neighbor (KNN) algorithm to develop a data model which predicts whether the tumor is benign or malignant.

The breast cancer Wisconsin (Diagnostics) dataset has been used for the training and testing of the data model created by us.

Dataset

The dataset used in this project is the UCI Breast Cancer Wisconsin (diagnostics) Dataset. It contains 569 different entries of patients who had breast cancer either malignant or benign. The dataset has 30 parameters defined on the basis of which we determine whether the tumor is malignant or benign. The dataset contains 63% benign and 37% malignant tumors data.

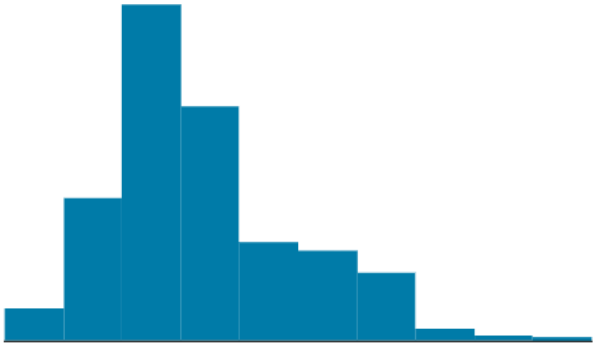
The different parameters are states as follows:

radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst

Variation in the parameters

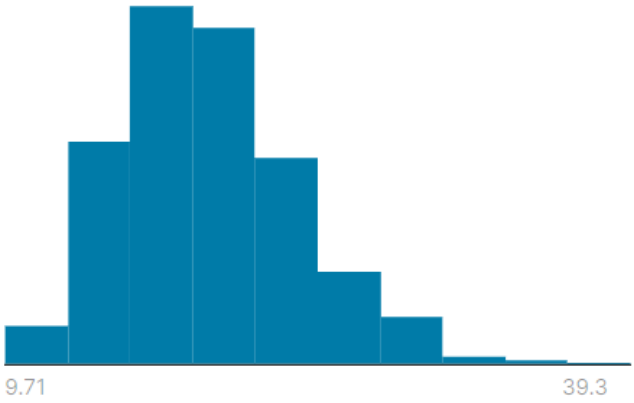
 # radius_mean

mean of distances from center to points on the perim



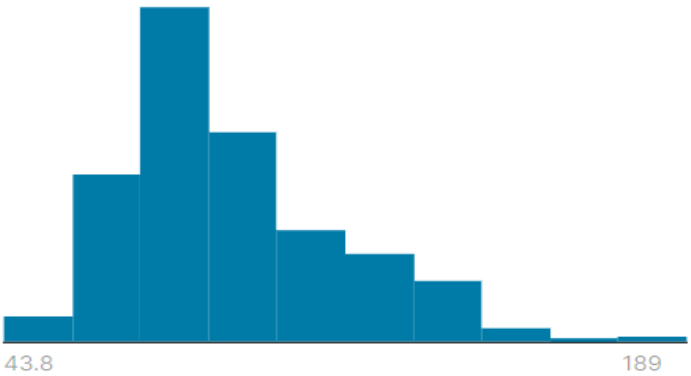
 # texture_mean

standard deviation of gray-scale values



 # perimeter_mean

mean size of the core tumor



Methodology

The Data Model in this project is made using the K-nearest neighbor algorithm. Among the supervised machine learning algorithms, K-nearest neighbors (KNN) is one of the most effective techniques. It performs classification on certain data points. The KNN algorithm is a type of supervised ML algorithm that can be used for both classifications as well as regression predictive problems. It uses 'attribute similarity' to predict the values of new data-points and then the new data point will be assigned a value based on how closely it matches the points in the training set.

Algorithm

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

The K value for our dataset is 23, because the optimal K values is the square root of the number of entries in the dataset, i.e.,569, which comes out to be 23.85372 Rounding it down to 23 is better because we want an odd K value as an odd number so that we can calculate a clear majority in the case where only two groups are possible.

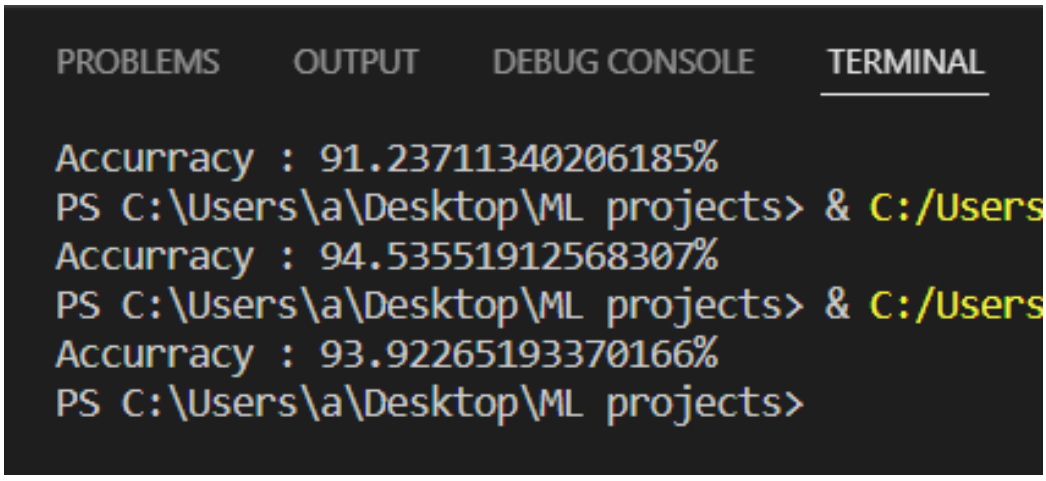
Code

breastcancer.py

```python
import math
import operator
import csv
import random

def loadDataset(filename, split, trainingSet=[],testSet=[]):
    with  open(filename, 'r') as csvfile:
        lines = csv.reader(csvfile)
        dataset=list(lines)
        for x in range(len(dataset)-1):
            for y in range(30):
                dataset[x][y]= float(dataset[x][y])
            if random.random() < split:
                trainingSet.append(dataset[x])
            else:
                testSet.append(dataset[x])

def euclideanDistance(instance1, instance2, length):
    distance=0
    for x in range(length):
        distance+=pow((instance1[x]-instance2[x]),2)
    return math.sqrt(distance)

def getNeighbors(trainingSet, testInstance, k):
    distances=[]
    length=30
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance,trainingSet[x],length)
        distances.append((trainingSet[x],dist))
    distances.sort(key=operator.itemgetter(1))
    neighbors=[]
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors

```

```python
def getResponse(neighbors):
    m =0
    b=0
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response=='M':
            m+=1
        else:
            b+=1
    if m>b:
        return 'M'
    else:
        return 'B'

def getAccuracy(testSet, predictions):
    correct=0
    for x in range(len(testSet)):
        if testSet[x][-1] is predictions[x]:
            correct+=1
    return (correct/float(len(testSet)))*100.0

def main():
    trainingSet =[]
    testSet=[]
    loadDataset(r'data.csv',0.66,trainingSet,testSet)
    predictions=[]
    k=23 #dataset length is 569 so the value of k for optimal result is sqrt(564) which round downs to 23
    for x in range(len(testSet)):
        neighbors= getNeighbors(trainingSet, testSet[x], k)
        result= getResponse(neighbors)
        predictions.append(result)
        #print('> predicted = '+ repr(result) + ', actual = ' + repr(testSet[x][-1]))
    accuracy = getAccuracy(testSet, predictions)
    print('Accurracy : '+ repr(accuracy) + '%')
main()
```

Output



Conclusion

This project helped us understand the working of data models using KNN algorithm. Machine learning approaches have been increasing rapidly in the medical field due to their monumental performance in predicting and classifying disease. The same data model can be used for other predictions as well for various types cancer depending on their dataset.
The average accuracy of the data model is 93.23% with the maximum accuracy reached 94.5%.

References

The dataset has been taken from the following link:

https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data?resource=download&select=data.csv

The data Model has been made with the help of the following videos:

https://www.youtube.com/watch?v=ULvlqwjNNAo

https://www.youtube.com/watch?v=6kZ-OPLNcgE