BREAST CANCER RISK PREDICTION

INTRODUCTION

Overview:

Breast cancer is one of the main causes of cancer death worldwide. Early diagnostics significantly increases the chances of correct treatment and survival, but this process is tedious and often leads to a disagreement between pathologists. Computer-aided diagnosis systems showed potential for improving the diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible.
Here we are developing a machine learning model where in the model gets trained by considering the parameters such as: Radius ,Texture, Perimeter, Area, Smoothness, Concavity, Concaveness, Compactness here all these parameters are taken in mean, se and overall values are been taken. And the model is been trained using Auto AI service in IBM Watson cloud and that can be deployed in an application such as web or mobile applications.
(i)Downloading dataset
(ii) Data Preprocessing
(iii) Model Building
(iv) Application Building

Purpose

Through this repository we can:

- Apply the fundamental concepts of machine learning from an available dataset
- Evaluate and interpret my results and justify my interpretation based on observed data set
- Create notebooks that serve as computational records and document my thought process.

The analysis is divided into four sections, saved in juypter notebooks in this repository

1. Identifying the problem and Data Sources
2. Exploratory Data Analysis

3. Pre-Processing the Data
4. Build model to predict whether breast cell tissue is malignant or Benign

STEPS INVOLVED IN DEVELOPING A MACHINE LEARNING MODEL FROM SCRATH.

Identifying the problem and Getting data.

Notebook goal:Identify the types of information contained in our data set In this notebook I used Python modules to import external data sets for the purpose of getting to know/familiarize myself with the data to get a good grasp of the data and think about how to handle the data in different ways.

Exploratory Data Analysis

Explore the variables to assess how they relate to the response variable In this notebook, I am getting familiar with the data using data exploration and visualization techniques using python libraries (Pandas, matplotlib, seaborn. Familiarity with the data is important which will provide useful knowledge for data pre-processing)

Pre-Processing the data

Notebook goal:Find the most predictive features of the data and filter it so it will enhance the predictive power of the analytics model. In this notebook I use feature selection to reduce high-dimension data, feature extraction and transformation for dimensionality reduction. This is essential in preparing the data before predictive models are developed.

Predictive model using K Nearest Neighbors (KNN) **Construct predictive models to predict the diagnosis of a breast tumor.** In this notebook, I construct a predictive model using KNN machine learning algorithm to predict the diagnosis of a breast tumor. The diagnosis of a breast tumor is a binary variable .

Optimizing the KNN Classifier

Construct predictive models to predict the diagnosis of a breast tumor. In this notebook, I aim to tune parameters of the KNN Classification model using scikit-learn.

LITERATURE SURVEY

Existing problem:

Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year, and also causes the greatest number of cancer-related deaths among women. In 2018, it is estimated that 627,000 women died from breast cancer – that is approximately 15% of all cancer deaths among women. While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally.

Proposed Solution:

In order to improve breast cancer outcomes and survival, early detection is critical. There are two early detection strategies for breast cancer: early diagnosis and screening. Limited resource settings with weak health systems where the majority of women are diagnosed in late stages should prioritize early diagnosis programmes based on awareness of early signs and symptoms and prompt referral to diagnosis and treatment.

Early diagnosis strategies focus on providing timely access to cancer treatment by reducing barriers to care and/or improving access to effective diagnosis services. The goal is to increase the proportion of breast cancers identified at an early stage, allowing for more effective treatment to be used and reducing the risks of death from breast cancer.

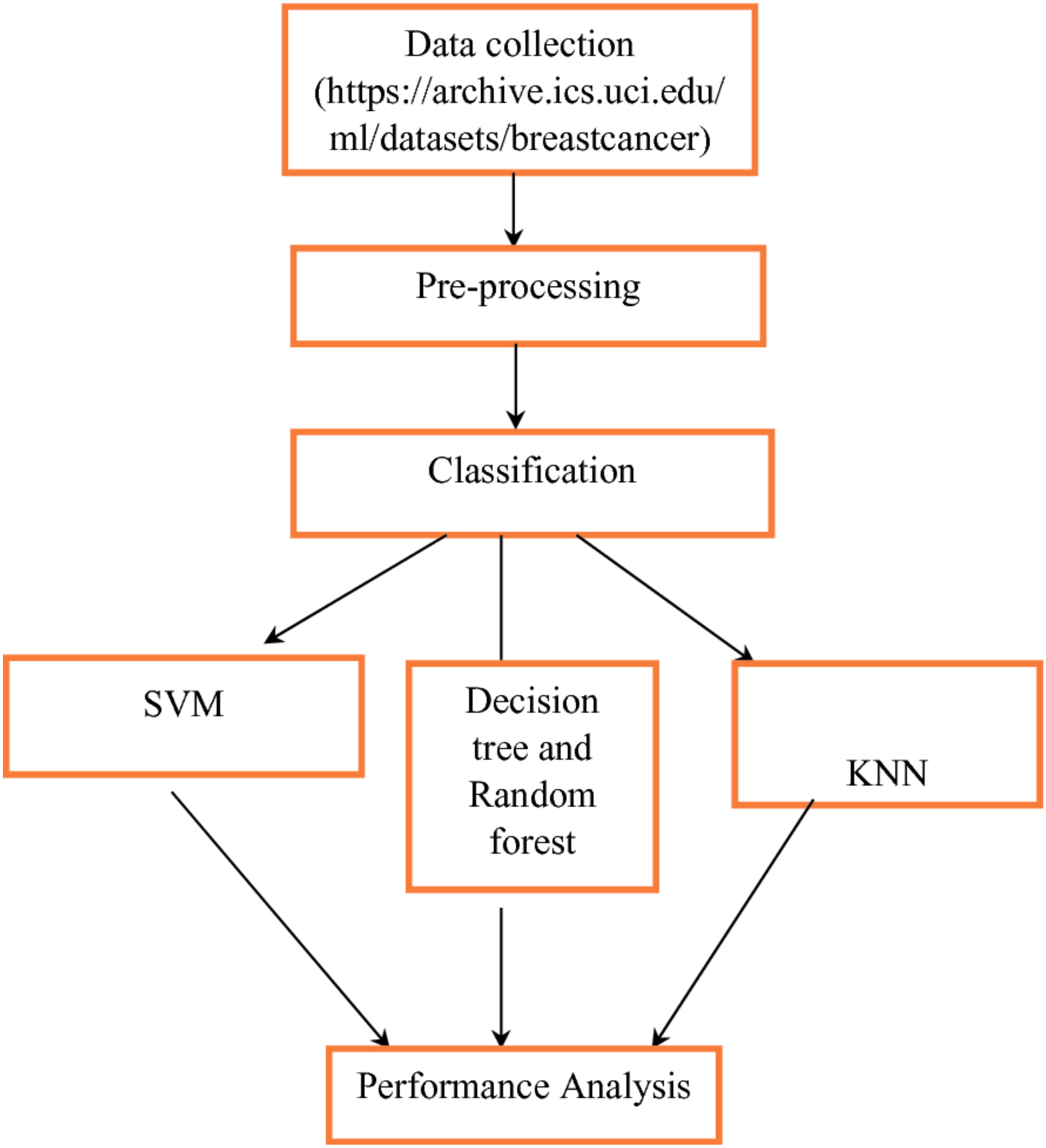
Screening consists of testing women to identify cancers before any symptoms appear. Various methods have been evaluated as breast cancer screening tools, including mammography, clinical breast exam and breast self-exam.

By working on the dataset procured through early diagnosis and screening ,we can use that in a machine learning model.

In the world of data ,we can push the best dataset into the machine learnig algorithm like KNN ,Logistic regression ,random forest,
In the source code I have deployed the model using various classification algorithms like decision tree ,logistic regression, KNN classification, and naive -bayes classification.
The predictions/analysis is also shown at the end of this repository.

THEORITICAL ANALYSIS

Block daigram:


Data collection (https://archive.ics.uci.edu/ ml/datasets/breastcancer)
Pre-processing
Classification
SVM
Decision tree and Random forest
KNN
Performance Analysis

Hardware/software designing:

A good IDE can run the application in a PC. (Linux/Windows).
Ruuning on the virtual environment is safer.
There is no need of downloading an IDE for running machine learning model.
Using Anaconda ,which comes with Jupyter Notebook.
We can use Jupyter Notebook to work on our MODEL.
The IDE must have libraries/packages installed like NumPy,
Pandas,scikit-learn,matplotlib, which can be installed by running simple command line codes on the ANACONDA COMMAND PROMT.
This model then can be put up on cloud services like IBM CLOUD SERVICES.
Through the cloud we can access the machine learning model through URL.
Any smartphone application that can run a python script can run the model.

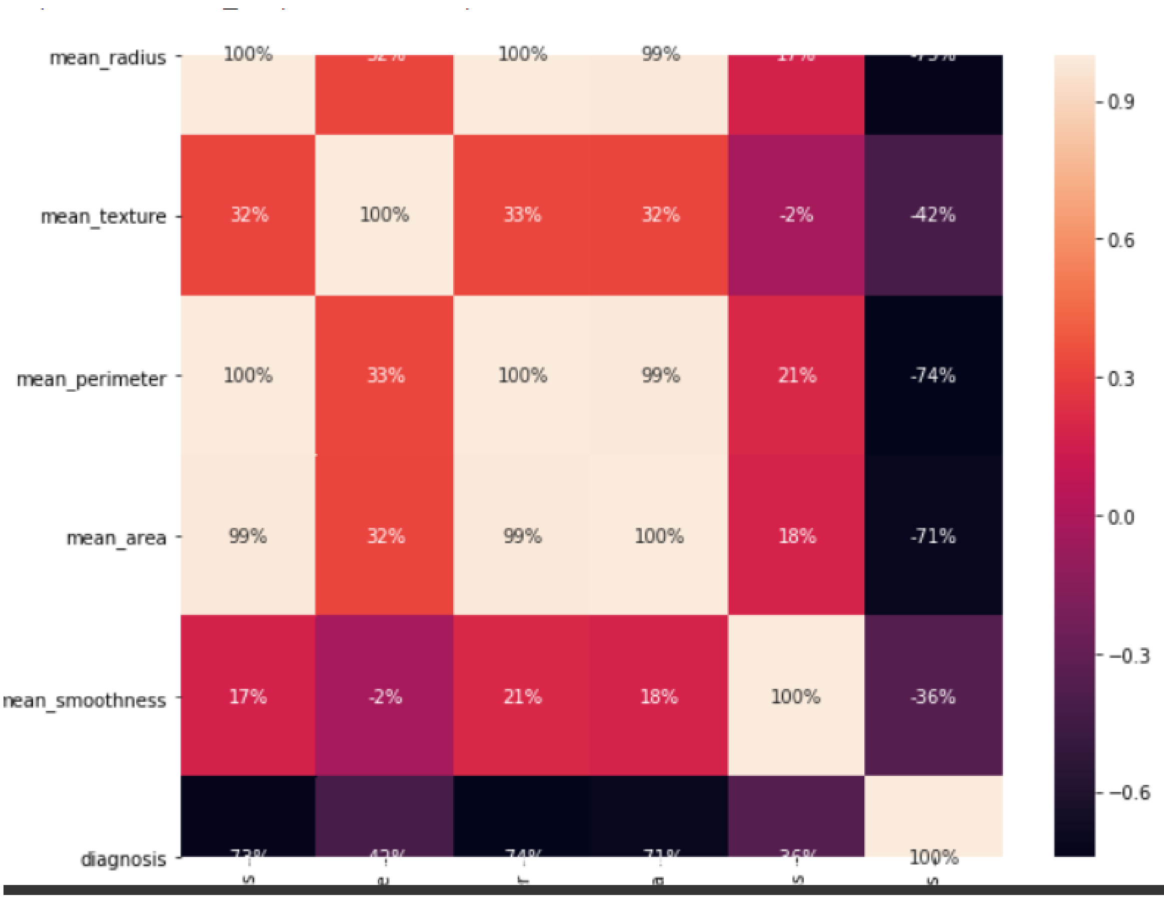
MODEL design:

For deployment, the code was fed into the IBM Watson ,this has a storage instance,machine learning instance.
For making web applictaions it has NODE-RED,which is made on the Node.js SDK.
This makes easy to make web application as well as store up in the cloud.

EXPERIMENTAL INVESTIGATIONS

The dataset in the given problem consist of 569 coloumns and 6 rows.
The prediction was to make whether according to the data from the previous data,is the new patient is likely to have breast cancer or not.
Diagnosis coloumn has the output as 0 and 1,where 1 represents patient with the cance r ,wheras 0 represents no cancer.

During analysis of the dataset , and studying the heatmap of the correlation between different features,i came to a conclusion that no part of the dataset should not be ignored/deleted.



There is no categorical data in the dataset.Batch of the data is bisected into 80:20 ratio of training test: test data.

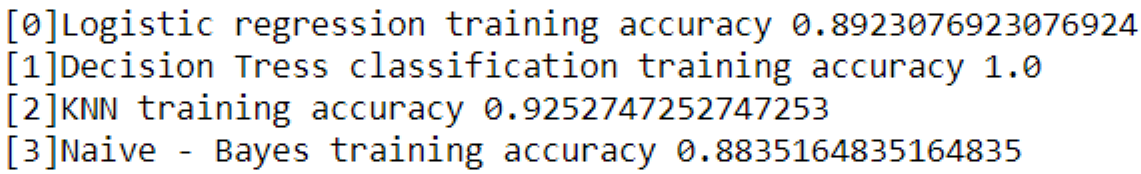
80% data is used for training .

We have kNN algo ,this is a lazy algorithm,in which time required for training the data is more as compared to the test data.

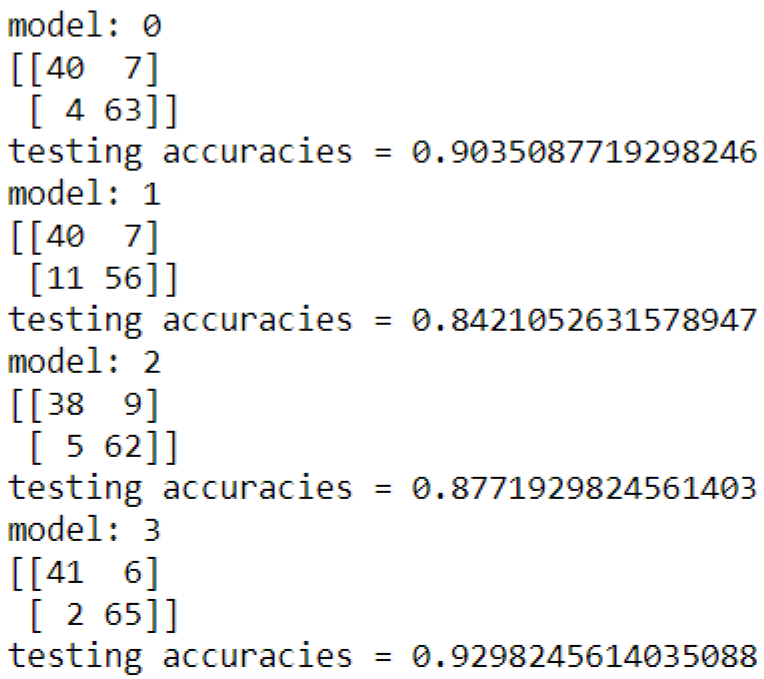
The features of the dataset is done feature scaling since the algorithm based on the distance of the features.

Now the model is run on the jupyter notebook ,and this is added on the IBM Watson service.

RESULT



```
[0]Logistic regression training accuracy 0.8923076923076924
[1]Decision Tress classification training accuracy 1.0
[2]KNN training accuracy 0.9252747252747253
[3]Naive - Bayes training accuracy 0.8835164835164835
```



```
model: 0
[[40  7]
 [ 4 63]]
testing accuracies = 0.9035087719298246
model: 1
[[40  7]
 [11 56]]
testing accuracies = 0.8421052631578947
model: 2
[[38  9]
 [ 5 62]]
testing accuracies = 0.8771929824561403
model: 3
[[41  6]
 [ 2 65]]
testing accuracies = 0.9298245614035088
```

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:
1.Both the model and application is lightweight.
2. Prediction speed is high.
3. Server side is authenticated.
4. The prediction is helpful in educational purposes.

DISADVANTAGES:
1. Node-red is not suitable for commercial purposes.
2. Predictions on missing feature can be inaccurate.

APPLICATIONS

1. Through this model we can predict , whether a patient is likely to have cancer or not ,without even doing medical tests.
2. Medical test can be modified and optimized .
3. We can analyse which starta of population(in women) ,are more likely to have it in the future.
4. The application can be run on android by SL4A.

5. It would be very useful and handy tool in healthcare.
6 . It can run on PC server very fast.
7. It bypasses the first level of manual inspection.

CONCLUSION

After to the ongoing covid-19 pandemic, more people will a urge to have pre knowledge of all medical ailments and advancements.
For better working of the model we would be needed actual and large dataset.
Since by the level of dataset in the repository ,the results are "good"

In the source code I have 4 kind of models ,but on the IBM cloud ,Node-red I have deployed it by using only KNN classification algorithm.

The model was trained on a dataset of 569 patients ,the total number of fetures were 5 . Feature scaling was very important in this problem set ,as the classification algorithm used demands uniform distancing between all features.
The model after initial testing was deployed IBM Watson and NODE-RED.
This intermediate level of machine learning is very necessary to understand to leave scope for potential developments.

FUTURE SCOPE

As per WHO breast cancer is a deadly cancer ,which develops inside the human body with out even showing simtoms.
Since in the initial days of the diesease, no symptoms are witnessed by the patient ,the dieases develops into later stages of the deadly cancer .
If by regular examination ,we can deploy the dataset into the model to predict ,to very good accuracy we can find about the cancer ,without any medical tests.
Thatswhy , this prediction algorithm has great future ahead ,if it keeps on learning from bigger dataset.

APPENDIX

SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

dataset = pd.read_csv("breast_cancer_data.csv")
dataset
dataset.isnull().any()

plt.figure(figsize=(10,8))
sns.heatmap(dataset.iloc[:,0:6].corr(),annot=True,fmt='.0%')

dataset['diagnosis'].value_counts()

x= dataset.iloc[:,0:5].values
x

y= dataset.iloc[:, -1].values
y

from sklearn.model_selection import train_test_split
x= dataset.iloc[:,0:5].values
x

y= dataset.iloc[:,5].values
y

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
x_train
y_train

from sklearn.preprocessing import  StandardScaler
sc= StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

```
x_train
x_test

# creating fuction for model prediction#
def models(x_train,y_train):
    #logistic regression classification#
    from sklearn.linear_model import LogisticRegression
    lg=LogisticRegression()
    lg.fit(x_train,y_train)
    #decision tree classification#
    from sklearn.tree import DecisionTreeClassifier
    dt= DecisionTreeClassifier(criterion='entropy',random_state=0)
    dt.fit(x_train,y_train)

    #KNN classification#
    from sklearn.neighbors import KNeighborsClassifier
    kn=KNeighborsClassifier()
    kn.fit(x_train,y_train)

    #naive-bayes classification
    from sklearn.naive_bayes import GaussianNB
    nv=GaussianNB()
    nv.fit(x_train,y_train)

    #print the model accuracy#
    print('[0]Logistic regression training accuracy',lg.score(x_train,y_train))
    print('[1]Decision Tress classification training accuracy',dt.score(x_train,y_train))
    print('[2]KNN training accuracy',kn.score(x_train,y_train))
    print('[3]Naive - Bayes training accuracy',nv.score(x_train,y_train))

    return lg,dt,kn,nv


model=models(x_train,y_train)      #prints prediction by various kinds of model#

#test modelacuracy on the test data on confusion-matrix#
from sklearn.metrics import confusion_matrix
```

```
for i in range(len(model)):
    print('model:',i)
    cm=confusion_matrix(y_test,model[i].predict(x_test))
    TP=cm[0][0]
    TN=cm[1][1]
    FN=cm[1][0]
    FP=cm[0][1]
    print(cm)
    print('testing accuracies =',(TP + TN)/(TP + TN + FP + FN))
```