Final Project

Mukul Bisht – C0857928

Neelesh Vashist – C0858518

Rohit Kumar – C0859060

Saurabh Singh – C0859334

Artificial Intelligence and Machine Learning (AIMT), Lambton College

AML 2203_1 Advanced Python AI and ML Tools

Professor Vahid Hadavi

Dec 12, 2022

Table of Contents

Abstract

Artificial intelligence and machine learning algorithm-based approaches are used in the modern medical period to enable better decision-making for medical professionals.

The PyCoders team (Mukul, Neelesh, Rohit, and Saurabh) has endeavoured to create an online platform that focuses on developing a machine learning-based health problem diagnostics application while maintaining the general usage of the machine learning method.

Objective

We must develop a tool for diagnosing health issues that will help doctors make quick decisions during the early stages of diseases using trained models and pre-set data.

Our online tool will make the most accurate predictions about a person's health issues based on their health characteristics using various Machine Learning techniques, including Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and Random Forest Classifier.

Currently, diabetes, breast cancer, and COVID-19 are the three diseases we are trying to forecast.

Datasets

We are going to use three types of datasets for our Three diseases prediction models.

Diabetes Dataset:

For our Diabetes Prediction dataset, we are using a dataset from Kaggle. The National Institute of Diabetes and Digestive and Kidney Diseases is the source of this dataset. The goal is to determine whether a patient has diabetes based on diagnostic parameters as follows:

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test
- Blood Pressure: Diastolic blood pressure (mm Hg)
- Skin Thickness: 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

- Age: Age (years)
- Outcome: Class variable (0 or 1)

Dataset: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

```
df = pd.read_csv("diabetes.csv")
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure: Diabetes Dataset

Breast Cancer Dataset:

For breast cancer detection, we use another dataset from Kaggle. This dataset's Features are calculated from a digital image of a breast mass's fine needle aspirate (FNA). They describe details of the cell nuclei visible in the picture. We will use parameters like:

- radius_mean - mean of distances from the center to points on the perimeter
- perimeter_mean – mean of the parameter
- area_mean – mean of area.
- compactness_mean - perimeter^2 / area - 1.0
- concavity_mean – the harshness of the contour's concave areas
- concave points_mean - how many concave areas are there in the contour

Dataset: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

```
# Reading the file
df = pd.read_csv("data.csv")
```

```
df.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

Figure 2. Breast Cancer Dataset

Covid – 19 Dataset:

For our Covid – 19 datasets, we collected the chest X-ray dataset from two sources can combined them. One was from the Pneumonia dataset from Kaggle, and we also used images from the covid-chest-x ray dataset from GitHub. Images from these datasets are extracted and kept in a folder labelled as 'COVID' and 'NORMAL.'

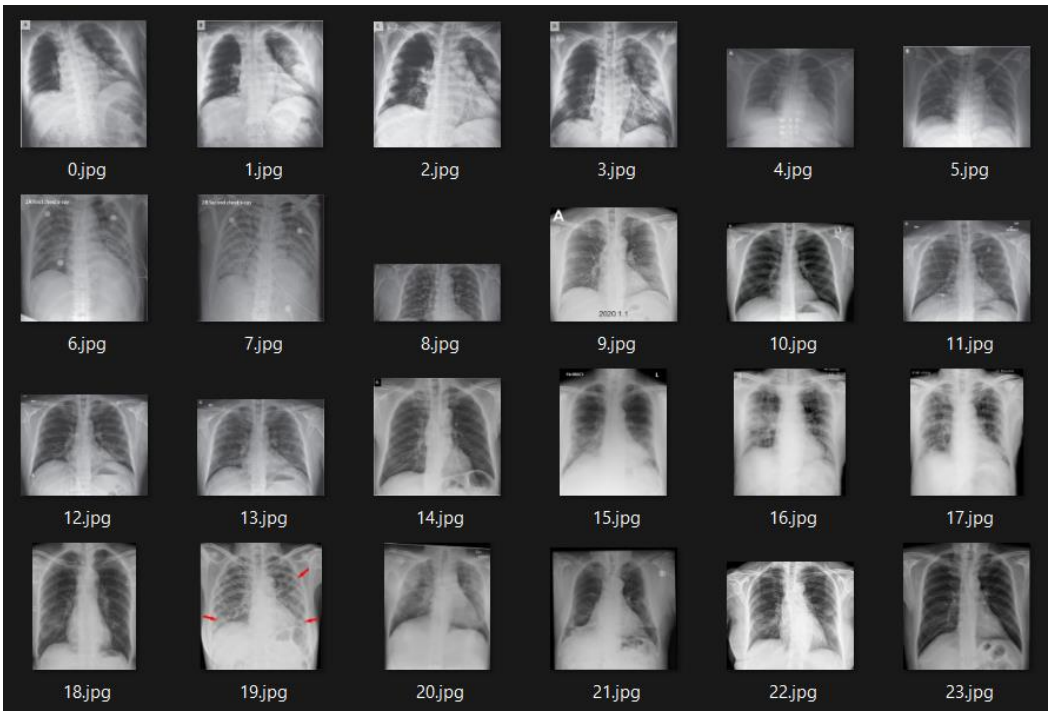

Figure 3. Covid – 19 Dataset

Diabetes Detection Model Steps

Data Cleaning:

- We drop duplicate columns.
- We replace zero values with Nan values because health parameters in this scenario can not be zero hence, they are considered as null values.

Column	Total missing Values	% of missing values
Pregnancies	0	0.00
Glucose	5	0.65
BloodPressure	35	4.56
SkinThickness	227	29.56
Insulin	374	48.70
BMI	11	1.43
DiabetesPedigreeFunction	0	0.00
Age	0	0.00
Outcome	0	0.00

Figure 4. Null values detected in diabetes model

- Then we perform outlier detection using Box-plot, and following is the output for the outlier detection:

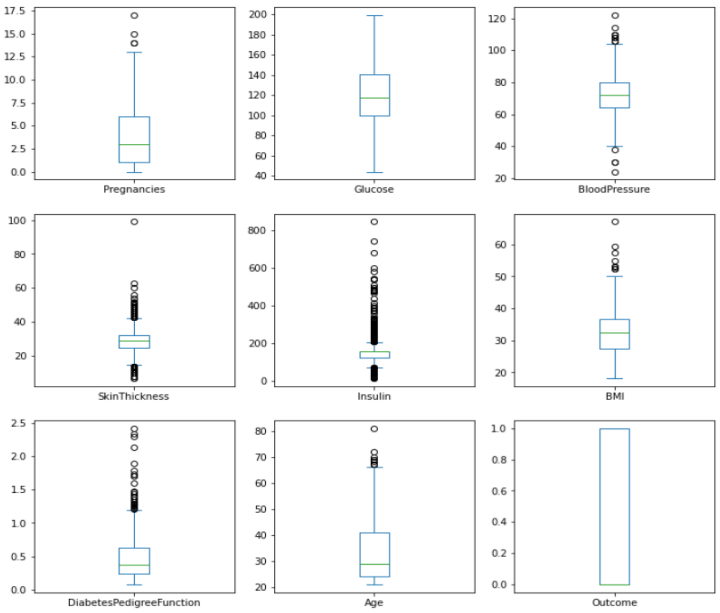

Figure 5. Outlier Detected in features

- Although the dataset contains a lot of outliers, but we believe that these datapoints are essential for diabetes prediction, so we will not remove the outlier.

Data Visualization:

- We check this distribution of the target variable 'Outcomes'

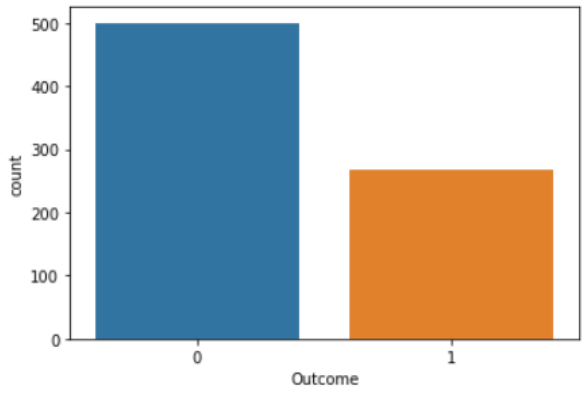

Figure 6. Distribution of diabetes outcomes.

- We check the correlation of the features using sns.heatmap and sns.pairplot, which shows the correlation between the features along with distribution of the features according to outcomes.

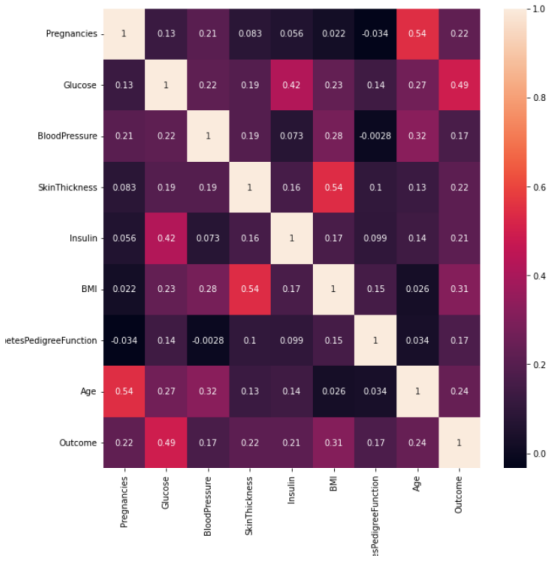

Figure 6. Heatmap showcasing the correlation between the features

Feature Scaling:

We use min-max scalar algorithm and standardization algorithm to scale the predictor values making it ready for predictions. This step will help us improve the accuracy of the model.

Data Training and Evaluation:

- We split the dataset into training and test dataset using the sklearn.model_selection library.
- We perform the Logistic regression on the dataset using the training set of unscaled datasets and scaled dataset, and compare the results.

	precision	recall	f1-score	support
0.0	0.84	0.79	0.81	100
1.0	0.65	0.72	0.68	54
accuracy			0.77	154
macro avg	0.75	0.76	0.75	154
weighted avg	0.77	0.77	0.77	154

Accuracy of the model = 76.62337662337663%

	precision	recall	f1-score	support
0.0	0.86	0.79	0.82	100
1.0	0.66	0.76	0.71	54
accuracy			0.78	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.79	0.78	0.78	154

Accuracy of the model = 77.92207792207793%

Figure 7. Results from Logistic Regression using scaled and unscaled dataset.

- Our dataset gives, 76.62% accuracy with unscaled dataset and 77.92% accuracy with scaled dataset.
- We Try to improve this by using Artificial Neural Networks (ANN) with Adam optimizer and ReLU and Sigmoid activation function and achieved accuracy of 77.27%, which is not much improvement compared to Logistic regression.
- We can stick with our regression model and drop our ANN model, but if we consider the imbalance in the outcomes in the target variable, we can improve the accuracy of our model. This can we achieved by over-sampling of our data.

- We Oversample our data and run it through our ANN model again and after we train the model with over sampled data and achieve a training set prediction accuracy of 79.31% and testing set prediction accuracy of 80.51%.

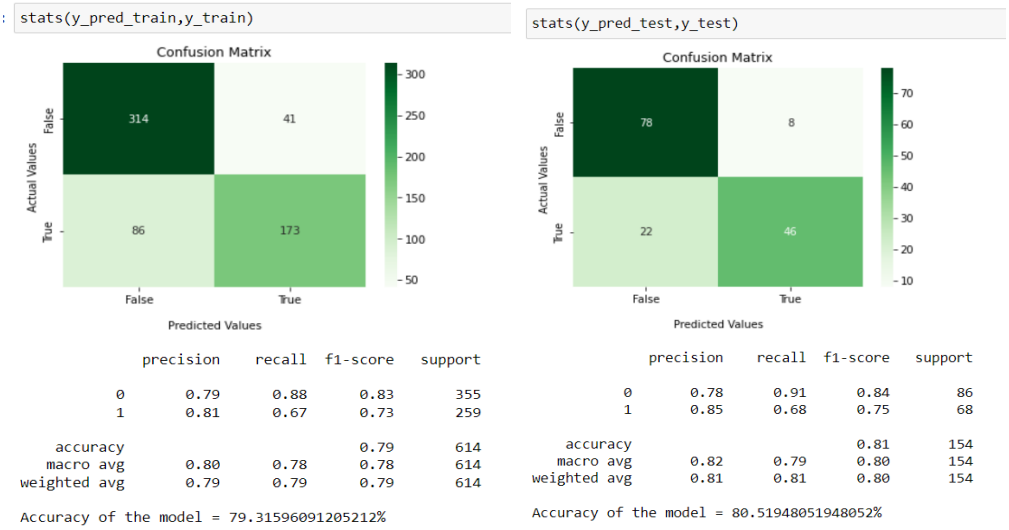

Figure 8. Accuracy from ANN with oversampled Dataset.

- We use inbuilt tensor flow library to save the model to use it for web application.

Breast Cancer Detection Model Steps

Data Cleaning

- We first drop The Id column and Unnamed column because they serve no purpose in prediction.
- Now we check for null values, but our dataset does not have any null values but there are 32 columns in which only few columns are highly correlated to the target variable so we remove redundant variables.
- We check the correlation using the heatmap provided by Seaborne library.

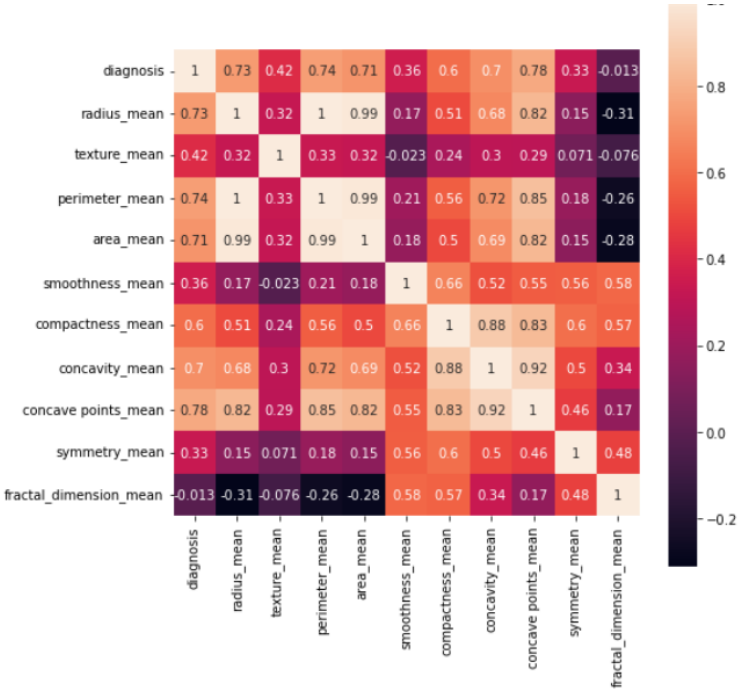

Figure 9. Correlation between features.

- After looking at the correlation between variables we chose radius_mean, perimeter_mean, area_mean, compactness_mean, concavity_mean, concave points_mean as our main parameters and removed extra columns.
- Now we perform Outlier Detection on the extracted features Using IQR.

```
IQR for column concave points_mean is 0.053689999999999995
Threashold Range for column concave points_mean where lower limit Q1 = -0.060224999999999994 and upper limit Q3 = 0.15453499999
999998
Total no of outliers for concave points_mean are 10
**********************************************************************************************
IQR for column area_mean is 362.40000000000003
Threashold Range for column area_mean where lower limit Q1 = -123.30000000000001 and upper limit Q3 = 1326.3000000000002
Total no of outliers for area_mean are 25
**********************************************************************************************
IQR for column radius_mean is 4.08
Threashold Range for column radius_mean where lower limit Q1 = 5.579999999999999 and upper limit Q3 = 21.9
Total no of outliers for radius_mean are 14
**********************************************************************************************
IQR for column perimeter_mean is 28.929999999999993
Threashold Range for column perimeter_mean where lower limit Q1 = 31.775000000000013 and upper limit Q3 = 147.49499999999998
Total no of outliers for perimeter_mean are 13
**********************************************************************************************
IQR for column concavity_mean is 0.10114000000000001
Threashold Range for column concavity_mean where lower limit Q1 = -0.12215000000000001 and upper limit Q3 = 0.28241000000000005
Total no of outliers for concavity_mean are 18
**********************************************************************************************
```

Figure 10. This is the output from the IQR

- We performed Outlier Detection using IQR and treatment using q10 and q90 quantiles and removed 100% of outliers.

Model Training and Evaluation

- After Splitting the data into a 70:30 ration of training and test dataset. We use Random Forest Classifier for breast cancer predictions.
- Model predictions using the training dataset give 100% accuracy, but this is because there are less numbers of records in our dataset.
- Our Model with the test dataset produces 94.15% accuracy. With 0.97 recall and 0.96 f1 score.

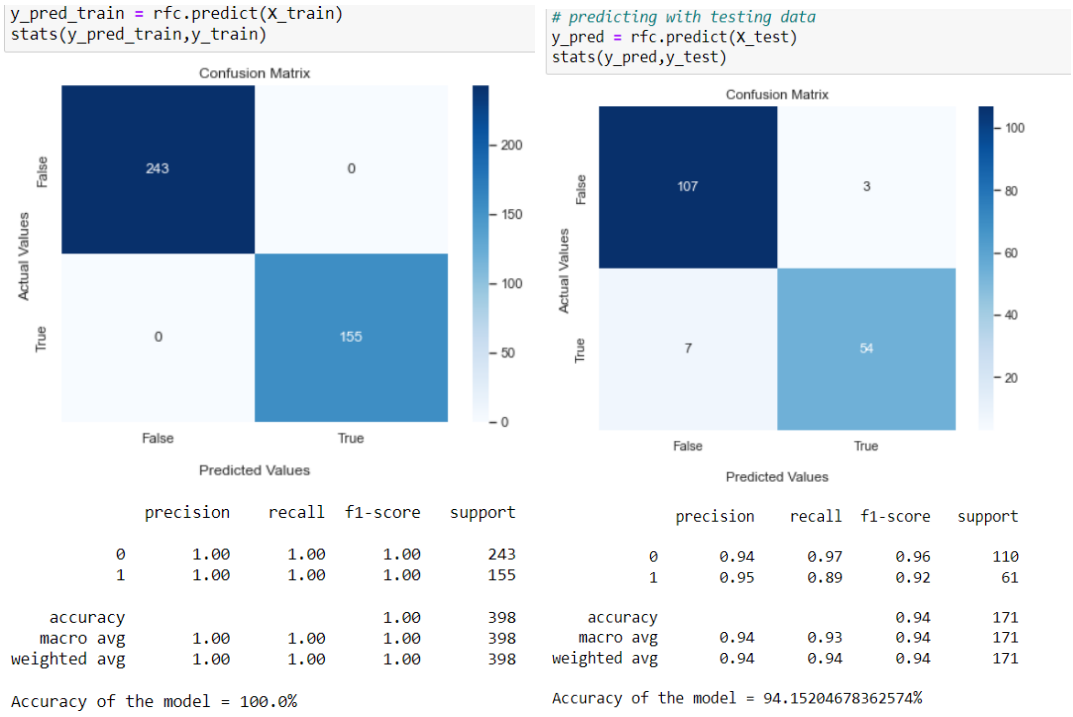

Figure 11. Results from Random Forest Classifer with traning and test breast cancer dataset

- We believe the model is over-fitted but this can be fixed if we have higher number of records.

Covid – 19 detection Model Steps

Data Processing

- After Collecting our dataset from various sources, we will use image preprocessing technique on our Image Dataset. For this pre-processing we are using Skimage library.
- We are using cv2 library to show the processing we are doing on the dataset images and we create a Sliding window element with the help or regular NumPy array.
- Firstly, we binarize the image by using rgb2gray method from Skimage.color library and create a binarized image.
- Then, we process that binary image through the erosion function to reduce the complexity of the image using the element we created.

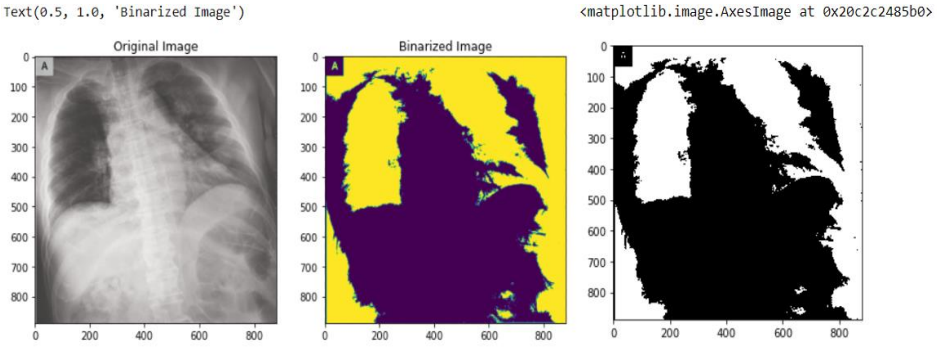

Figure 12. Image after Binarizing and going through erosion fnction.

- We save the processed images in separate folder and divide the processed and un-processed dataset into train, test, and validation datasets.

Model Training and Evaluation

- We used the convolutional neural network (CNN) algorithm to create a model sequential model with multiple layers Using the ReLU activation function for the inner layers and the Sigmoid activation function for the Final layer with an ‘Adam’ optimizer.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 conv2d_1 (Conv2D)           (None, 220, 220, 64)      18496

 max_pooling2d (MaxPooling2D  (None, 110, 110, 64)     0
 )

 dropout (Dropout)           (None, 110, 110, 64)      0

 conv2d_2 (Conv2D)           (None, 108, 108, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 54, 54, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 conv2d_3 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 26, 26, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 26, 26, 128)       0

 flatten (Flatten)           (None, 86528)             0

 dense (Dense)               (None, 64)                5537856

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 5,668,097
Trainable params: 5,668,097
Non-trainable params: 0
_________________________________________________________________
```

Figure 13. layers of the Generated CNN model

- We use ImageDataGenerator Library to generate, train, test, validation datasets from processed and unprocessed data for our CNN model.
- We then generate two models from processed and unprocessed datasets and evaluate the loss and accuracy generated from them.

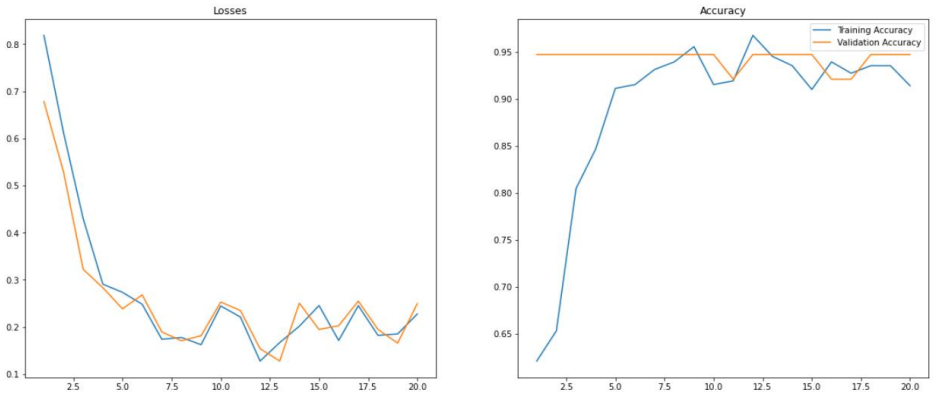

Figure 14. Loss and Accuracy chart for un-processed dataset.

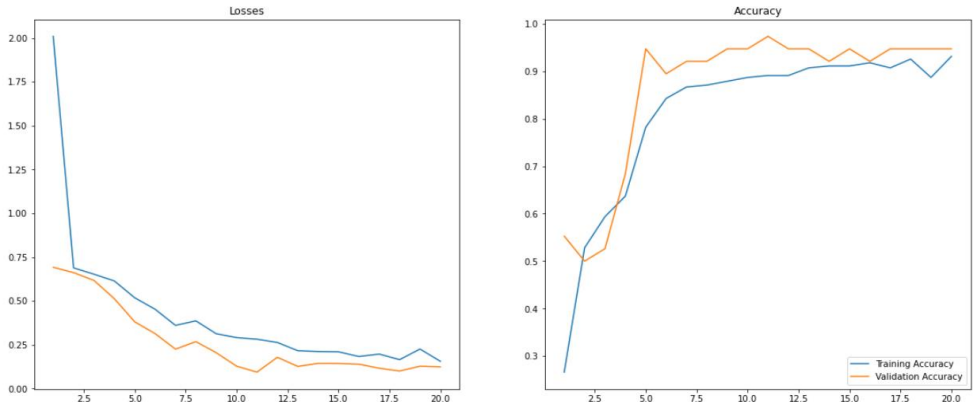

Figure 15. Loss and Accuracy chart for processes dataset.

- Now checking the datasets with the test dataset. We get the accuracy of un-processed dataset to be 97.61%, which is higher than 95.23% given by the processed dataset.

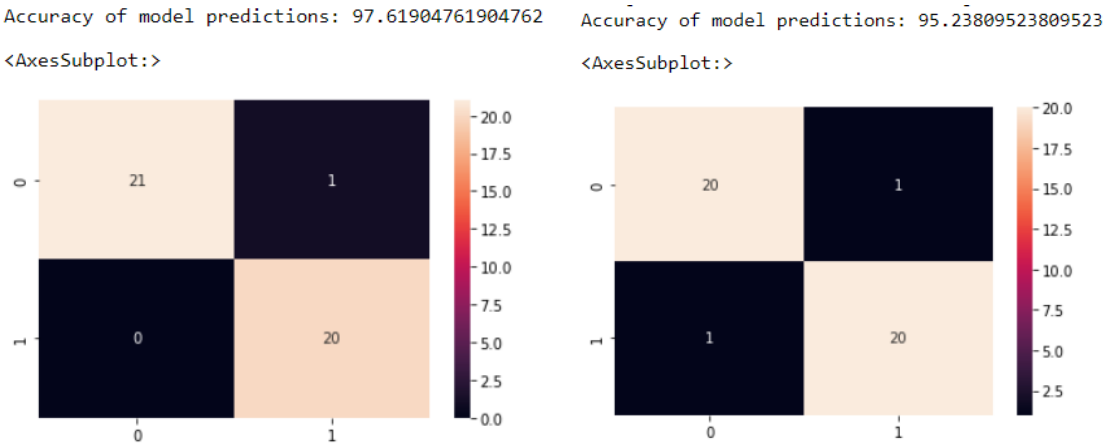

Figure 16. Confusion matrix and Accuracy of the un-processed data (Left) and the processed data (Right)

- This reduction in accuracy was expected as reducing the complexity of the image dataset will result in faster prediction at the cost of accuracy.

Front-End Designing

We used Technologies like Flask and web-development tools like html, CSS, and JS to develop the front-end of out project.

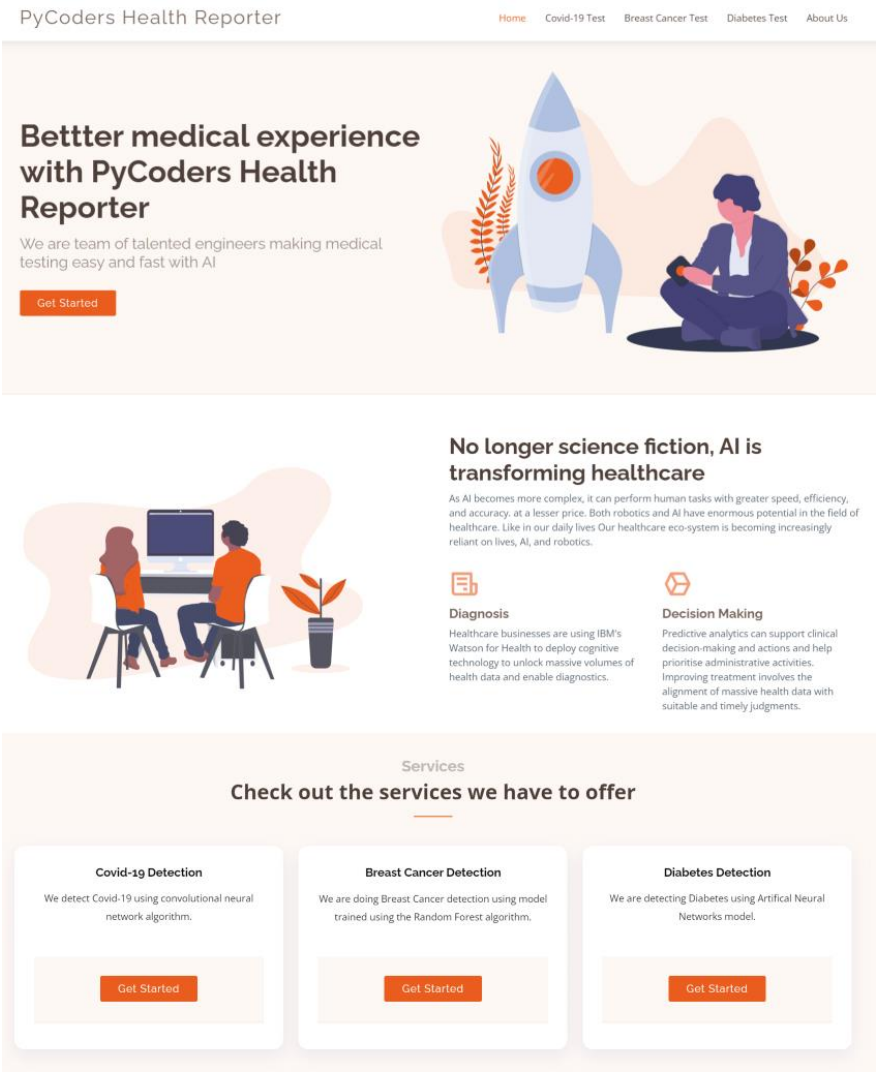

Figure 17. Homepage of our web application

- Our web app has three model prediction integrated in it.

Conclusion

- In this project we made a web application which offer intuitive UI to users/doctors to predict whether the patient is infected with Covid-19, has Breast Cancer, or has Diabetes.
- We used various machine learning techniques like Outlier Detection with IQR, Feature Tuning, Image Processing, Over Sampling, train-test-split, and algorithms like convolutional neural network (CNN), Logistic Regression, Artificial Neural Networks (ANN), and Random Forest Classifier.
- We used Flask and web development tools to develop our Front-end UI.

Future Scope

- This application can be further worked on to improve and can be used by doctors in future to give faster diagnostics.
- This application is flexible and can be extended to add more disease-detection models. The current application includes three modules related to covid 19, diabetes and breast cancer detection.
- We can add more functionality to manage customer records related to multiple diseases and can care repository for the patient.

References

Breast Cancer Wisconsin (Diagnostic) Data Set. (2016b, September 25). Kaggle. https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Diabetes Dataset. (2020, August 5). Kaggle. https://www.kaggle.com/datasets/mathchi/diabetes-data-set

Syarif, A., Azman, N., Ronal Repi, V. V., Sinaga, E., & Asvial, M. (2022). UNAS-Net: A deep convolutional neural network for predicting Covid-19 severity. *Informatics in Medicine Unlocked, 28*, 100842. https://doi.org/10.1016/j.imu.2021.100842

Chest X-Ray Images (Pneumonia). (2018, March 24). Kaggle. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

Cohen, J. P. (2020, June 10). *ieee8023/covid-chestxray-dataset*. GitHub. https://github.com/ieee8023/covid-chestxray-dataset