Neural Networks Techniques for Cancer Prediction

Problem statement:

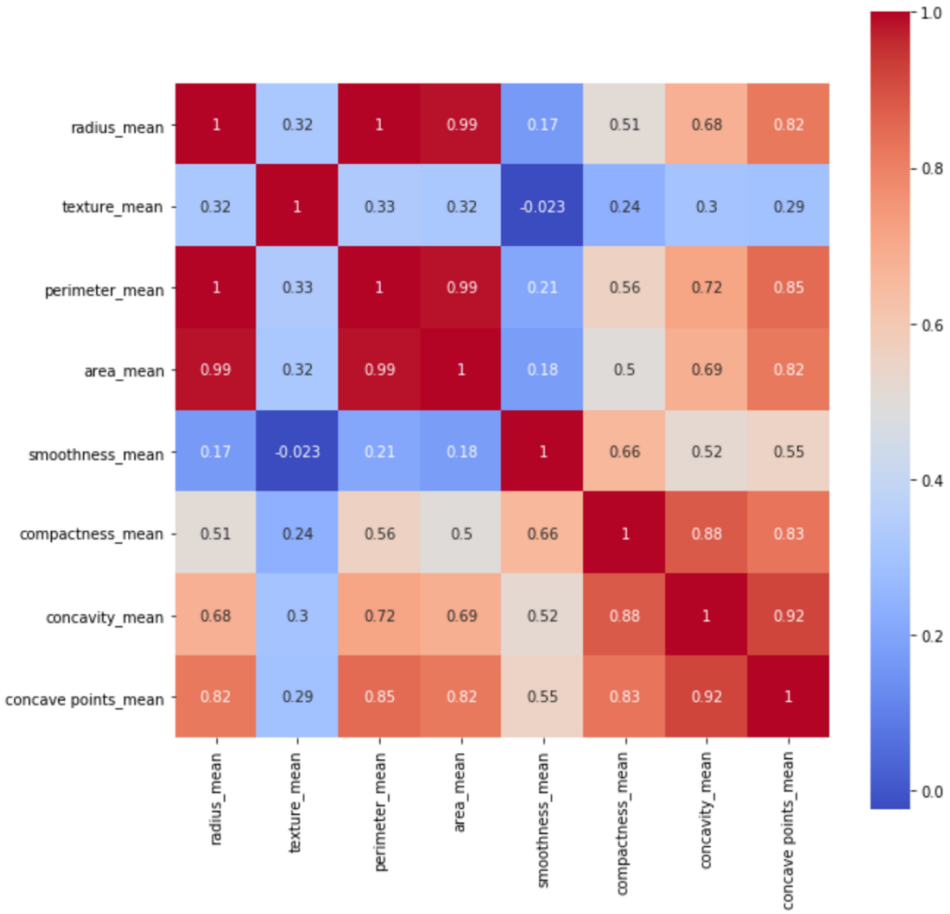
The goal is to classify whether the patient is benign on malignant. The dataset I will be using for this is the Breast Cancer Wisconsin data. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

I will be going with Neural Networks and the goal is to build the network to get a deeper understanding of the architecture of the network and experiment with the number of layers / neurons at each layer / learning rates / other hyperparameters. I will also be analyzing the results using the confusion matrix to derive deeper results.

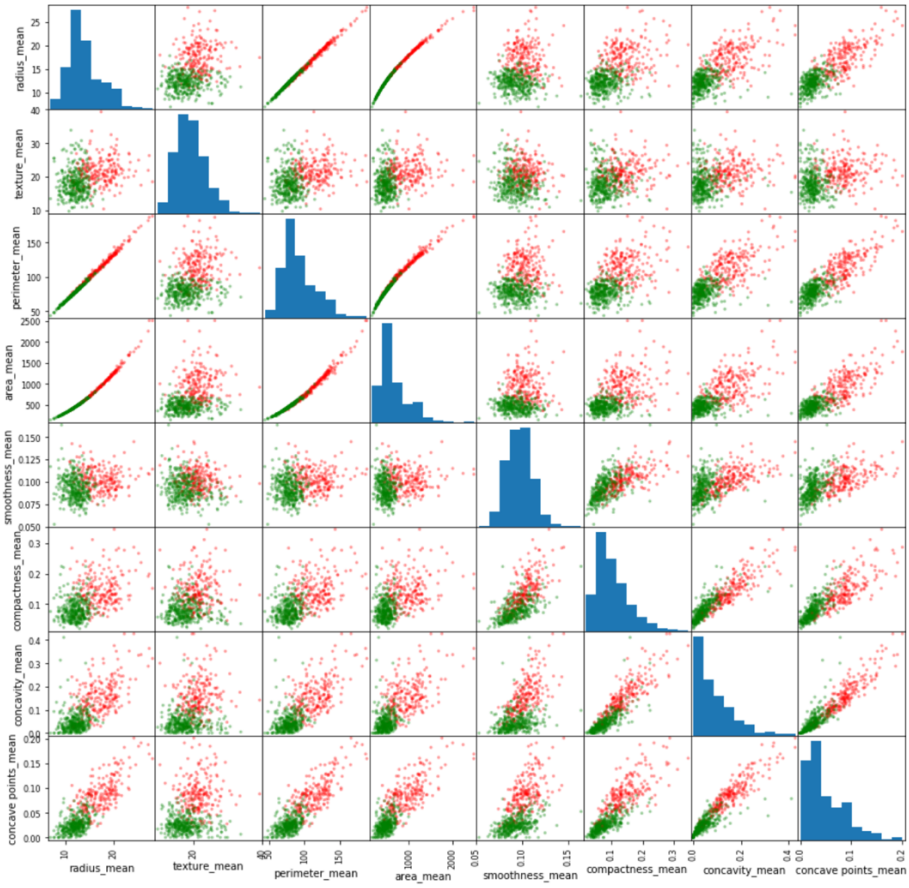
Data analysis

- I analyzed the dataset through correlation analysis, scatter plots and box plots to visualize how the data is correlated and which features are normally distributed and so on.

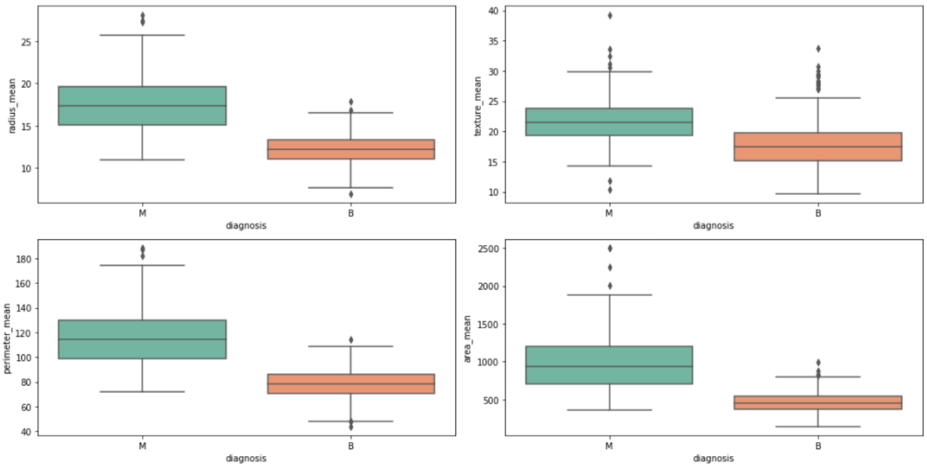
Correlation matrix:



Scatter plot:



Box plot:



3

Data preparation

- The data is separated and the classification variable is transformed to labels (0/1) which stand for (Benign/Malignant)
- Training/Test split is selected to be 80-20 and the data is scaler to fit and is normalized.

```
X = df.iloc[:,2:32]
y = df.iloc[:,1]
labelencoder_Y = LabelEncoder()
y = labelencoder_Y.fit_transform(y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Models tested

Model	Layers	Epochs	Optimizer
Model 1	2	1	Stochastic GD
Model 2	3	50	Stochastic GD
Model 3	3	100	Rms prop
Model 4	4	100	Rms prop

Results

- Model 1 is a simple 2-layer NN model which had poor performance, but as I increased the epochs / layers, I could clearly see and improvement in performance. Model 4 did not perform better than model 3 due to over-fitting the training data set. The models are saved as .h5 files which are analyzed finally in the results notebook. Each model is computed in its own notebook.

Model	Training Accuracy	Testing Accuracy
Model 1	78.02%	93.42%
Model 2	99.34%	98.25%
Model 3	99.56%	99.56%
Model 4	99.34%	98.25%

- **Confusion Matrix for Model 3** (reference mentioned in ipynb)**:**