**Joint Tech Internship Community Program**

**Generative AI Consortium (MSME)**

**SystimaNX IT Solutions Pvt Ltd.**

AI/ML Internship - Deep Learning

Assignment 1

To understand basic terminologies by working with Blood pressure measurement dataset

*Submitted by:*

S.K.Surya Prasath

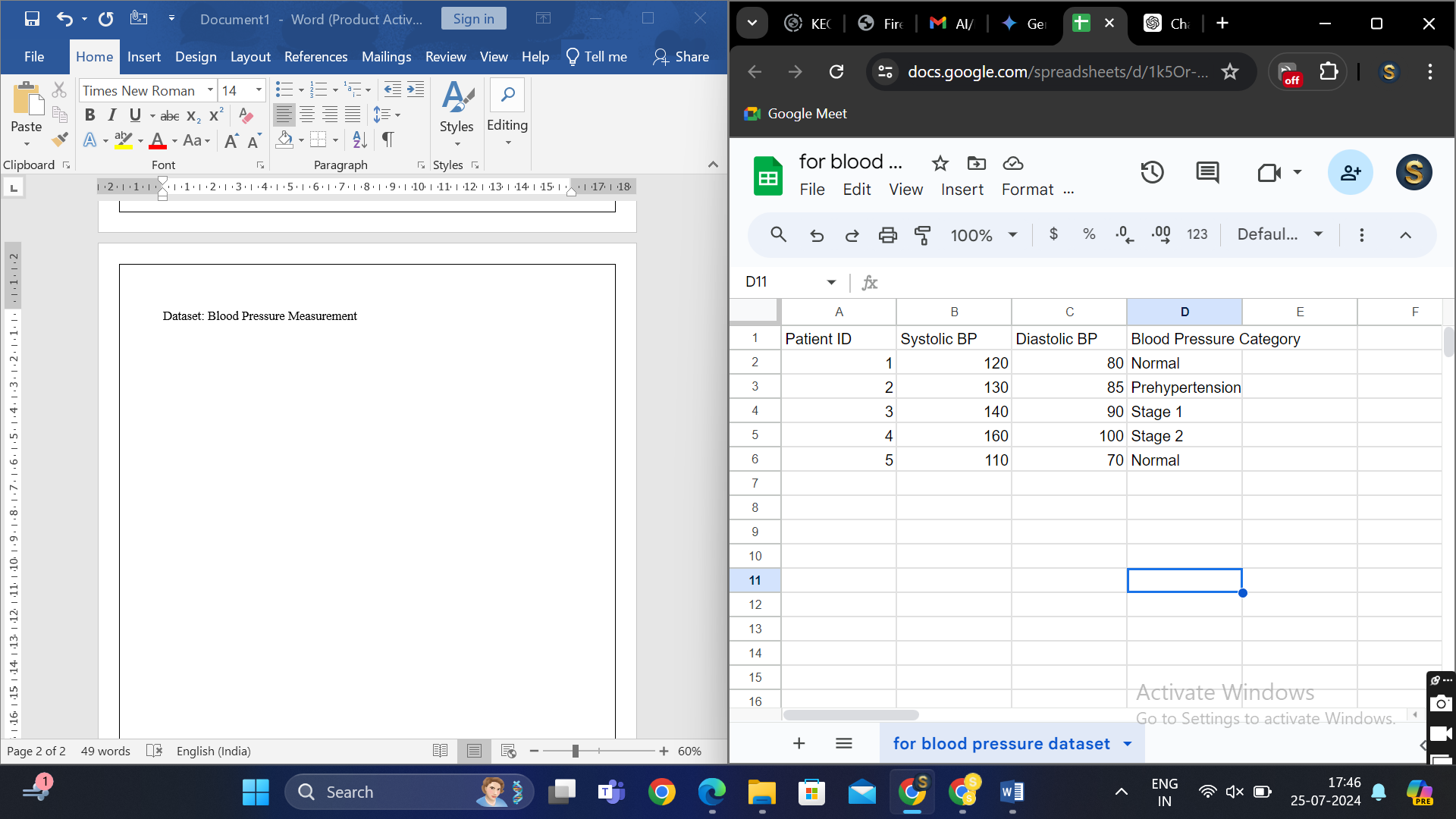
BE Computer Science

Kongu Engineering College

[suryaprasathsowmitha@gmail.com](mailto:suryaprasathsowmitha@gmail.com)

Phone: 6374578949

**Dataset:** Blood Pressure Measurement



**Problem Description:**

Input: Systolic and diastolic readings

Output: Blood Pressure category (Normal, Prehypertension, Stage1, Stage2)

**Key Terminologies and Parameters:**

1. **Neural Network:**

* A neural network is a type of algorithm with which we can perform both supervised and unsupervised learning. This has multiple connections over every neuron similar to the neural network of the brain and hence called neural network.
* Every neuron in one layer of the neural network is connected to every neuron of the next layer.
* It has many layers primarily input layer, hidden layer and output layer.

1. **Neuron:**

* The smallest unit of a neural network (basic building block) that receives input and process with the help of weights, bias and activation functions to generate output.

1. **Layers:**

* **Input layer**: Input layer has neurons equal to the number of inputs to be passed
* **Hidden layer**: It is the intermediate layer between the input and output layer where pattern identification is done.
* **Output layer**: It has neurons equal to the number of outputs.
* In our example we have very little amount of records so we can use a small number of hidden layers. Input and output layers have neurons based on the input data and output data.
* In our example we have 2 neurons in the input layer and one neuron in the output layer.

1. **Convolutional layer:**

* When we talk about Convolutional neural network (CNN) which is a little different from traditional neural network.
* In CNN we mainly process images but we can also process text
* Convolutional layer applies set of filters to extract features that is why it is mainly used in processing image.

1. **Convolutional neural network (CNN):**

* Unlike traditional neural network CNN don’t have connection between every neuron between two layers.
* CNN have convolutional layer, pooling layer and fully connected layers.
* It uses convolutional filters that helps to find local patterns.

1. **Recurrent Neural Network (RNN):**

* It is a type of neural network which is used mainly to work with sequence of data, like sentences.
* It has hidden state which maintains the memory of what it has seen throughout the sentence.

1. **Activation function:**

* The output of a neuron is dependant on many factors one such factor is activation function.
* It introduces non-linearity to the network.
* It helps to learn complex patterns.
* It helps to avoid underfit
* There are many types of activation functions each have their own applications.

1. **ReLU:**

* Rectified linear Unit is used widely in CNNs.
* It mitigates vanishing gradient problem that makes it suitable for object detection and image classification.
* **Formula:** max(0,x)
* **Range:** [0, ∞)

1. **Sigmoid:**

* Used mainly for binary classification as it gives output as either 0 or 1.
* **Formula:** 1/(1+e^-x)
* **Range:** (0,1)

1. **Tanh**

* Formula: (e^x – e^-x) / (e^x + e^-x)
* Range: (-1, 1)

1. **Softmax**

* Formula: e^x / sum of all (e^x)
* Range: (0, 1)

1. **Forward Propagation:**

* It involves multiple steps while training a neural network starting from allotting random weight then finding z using product of weight and input and then passing the value to activation function.

1. **Backpropagation:**

* When the actual value and predicted value differs the weights have to be updated for that first we have to find error value and optimize.

1. **Loss function:**

* It helps to find the difference between actual and predicted value.
* There are many types of error functions like Mean squared error.

1. **Cost function:**

* It is used to aggregate the loss over the entire training dataset.

1. **Gradient Descent:**

* It is an optimization algorithm used to minimize the cost function by adjusting the model’s weights.

1. **Learning Rate:**

* Learning rate is a hyperparameter that controls how much the model’s parameters are adjusted with respect to the gradient.

1. **Batch Size:**

* It is the number of training samples used in one forward/backward pass during training.

1. **Epoch:**

* An epoch is one complete pass through the entire training dataset.

1. **Overfitting:**

* When the model is trained too well including the noise and outliers then the output would be poor.
* The testing error will be much greater than the training error.

1. **Underfitting;**

* When the model is not trained with enough data then it can’t capture the patterns making it underfit.
* Both training and testing error are high.

1. **Training Set:**

* The portion of dataset that is used to train the model.

1. **Validation set:**

* The portion of dataset used to tune the model’s hyperparameters.

1. **Test set:**

* The portion of dataset used to asses the performance of the model.

1. **Cross Validation:**

* It helps to estimate the model’s performance and ensure that it generalizes to unseen data.

1. **Hyperparameters:**

* There are certain parameters that determines the output of the model. Learning rate, batch size, number of epochs are some of the hyperparameters.

1. **Model Parameters:**

* Parameters that gets updated from learning like weight and biases from the training data.

1. **Regularization:**

* It is a technique used to prevent overfitting with the help of a penalty.

1. **Dropout:**

* To avoid overfitting and over computation a certain amount of neurons are dropped out.

1. **Weight Initialization:**

* Initially weights can be allocated by many methods like zero initialization, random initialization.
* It depends on what activation function we use.

1. **Normalization:**

* When the data are not in the same scale computation becomes inefficient so to make all the data to a specific range we perform normalization.

1. **Standardization:**

* For the models having normal distribution we perform standardization that makes the mean as 0 and standard deviation as 1 and makes all the data accordance to it.