CS 6375

ASSIGNMENT 2

# Names of students in your group:

# Yagna Srinivasa Harsha Annadata (YXA210024)

# Vishruth Reddy Chinthi Reddy (VXC220020)

# Number of free late days used: 2

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

# Please list clearly all the sources/references that you have used in this assignment.

1. [Neural Networks - GeeksForGeeks](https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/)
2. <https://www.tensorflow.org/api_docs/python/tf/keras>

**Part2 – Programming Part**

**Google Colab link:** <https://colab.research.google.com/drive/1CEuizREFsN2JXqj2BCvgGxUfjMMDguF2?usp=sharing>

## Data Link:

**Dataset Link:** <https://raw.githubusercontent.com/YagnaAnnadata/Projects/main/LinearRegression/Dataset/winequality/winequality-red.csv>

**Original Dataset Link**: <https://archive.ics.uci.edu/dataset/186/wine+quality>

**Data Set Used:** Wine Quality

Quality of wine is the most important feature in wine making. The goal is to model wine quality based on physicochemical tests. The features which determine the quality of wine here are Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

## Variables List:

## Independent Variables:

## x1 = fixed acidity

## x2 = volatile acidity

## x3 = citric acid

## x4 = residual sugar

## x5 = chlorides

## x6 = free sulfur dioxide

## x7 = total sulfur dioxide

## x8 = density

## x9 = pH

## x10 = sulphates

## x11 = alcohol

## Dependent Variables:

## Y = Quality

**Distribution of variables:**

The below graphs show the histogram of the variables.

A group of blue and white graphs

Description automatically generated

**Correlation Matrix:**

A chart with red and blue squares

Description automatically generated

A bar graph with different colored bars

Description automatically generated

## Model Summary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activation Function | Learning rates | 0.001 | | 0.01 | 0.1 | 0.001 | | 0.01 | | 0.1 | 0.001 | 0.01 | 0.1 |
| Epoch | 1000 | | | | 5000 | | | | | 10000 | | |
| Sigmoid | Train Accuracy |  |  | |  |  |  | |  | |  |  |  |
| Test Accuracy |  |  | |  |  |  | |  | |  |  |  |
| Tanh | Train Accuracy |  |  | |  |  |  | |  | |  |  |  |
| Test Accuracy |  |  | |  |  |  | |  | |  |  |  |
| Relu | Train Accuracy |  |  | |  |  |  | |  | |  |  |  |
| Test Accuracy |  |  | |  |  |  | |  | |  |  |  |

**Plots:**

**Training and Test accuracy with different Learning Rates and Epochs for Sigmoid Function:**

A screenshot of a graph

Description automatically generated

**Training and Test accuracy with different Learning Rates and Epochs for Tanh Function:**

A graph of a function

Description automatically generated with medium confidence

**Training and Test accuracy with different Learning Rates and Epochs for Relu Function:**

A graph of a function

Description automatically generated

**Comparisons of Training and Test accuracy with different Learning Rates and Epochs for**

**Activation Functions:**

A group of blue and orange bars

Description automatically generated

## Libraries Used:

* Numpy
* Pandas
* Scikit-learn
* Matplotlib
* Seaborn
* Keras

## Conclusion:

As observed from the above table and plots, we find that Relu Activation function performs the best. Here the Relu activation function, the graph converges comparatively faster because of its functionality of choosing either zero and x.

The following is observed from the summary table:

This case displays higher Accuracy compared to all other results.

Activation Function: Relu, Learning Rate: 0.001, Epochs: 10000

Train Accuracy: **0.4500**, Test Accuracy: **0.8000**