ARtiFicial IntelLIGENCE Assignment 2

Submitted to:

Dr Ayyaz Hussan

Submitted by:

Amna Muzaffar

04072013003

BSCS(VII)

**Introduction**

This report focuses on predicting mobile prices (In Categories) through important preprocessing and the implementation of two powerful models: K-Nearest Neighbours (KNN) and Neural Networks (NN). Through careful data preparation, we ensure the quality of our input, setting the stage for accurate predictions. The KNN model leverages proximity-based classification, while the NN delves into complex patterns within the data. By examining the outcomes, this analysis aims to uncover insights into the effectiveness of these models for mobile price classification.

**Dataset Description**

The dataset includes features of mobile phones, such as battery power, Bluetooth, camera specifications, screen dimensions, and connectivity options. These features cover hardware aspects like RAM, internal memory, and camera resolution, as well as phone characteristics such as dual SIM support and 4G connectivity. The 'price\_range' column indicates the categorized price range of each mobile device.

A screenshot of a computer

Description automatically generated

**Methodology**

For KNN and ANN following steps are followed:

1. **Preprocessing:**

In Preprocessing, null and duplicate values are checked and removed. Also, categorical values are label encoded. In addition to this, for models to perform better, data is standardized.

A screenshot of a computer

Description automatically generated

We have also separated ground truth from other features and perform data preprocessing on it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

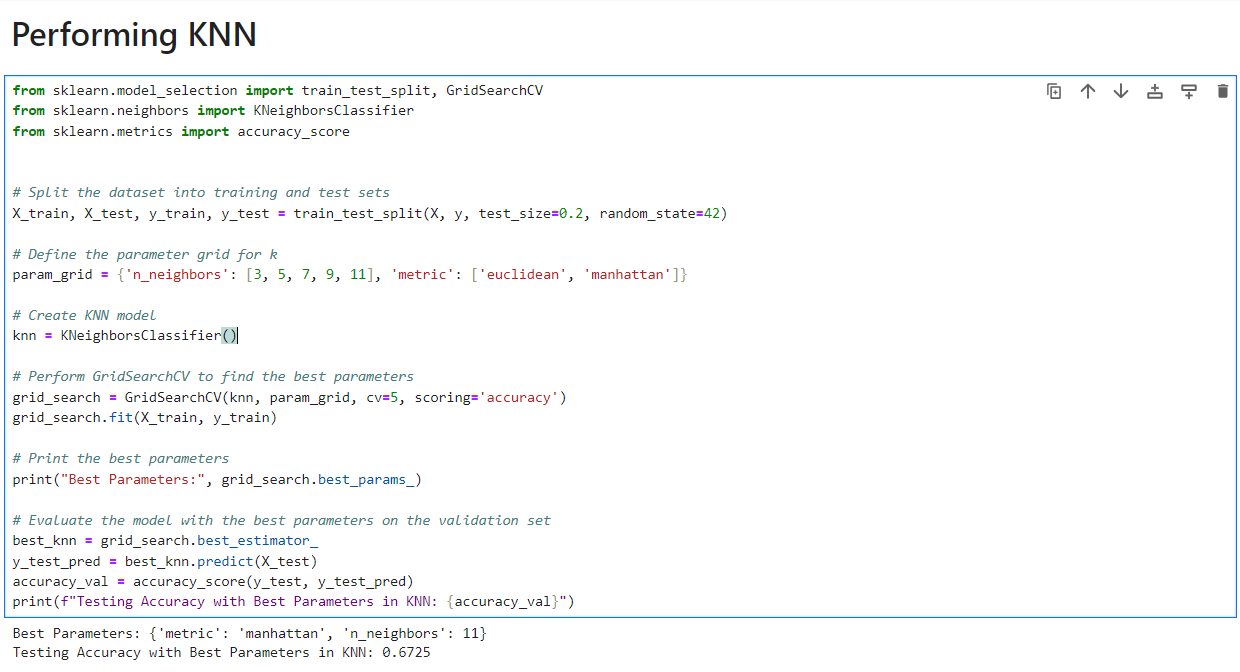
Description automatically generatedAlso Numerical Features are Standarized.

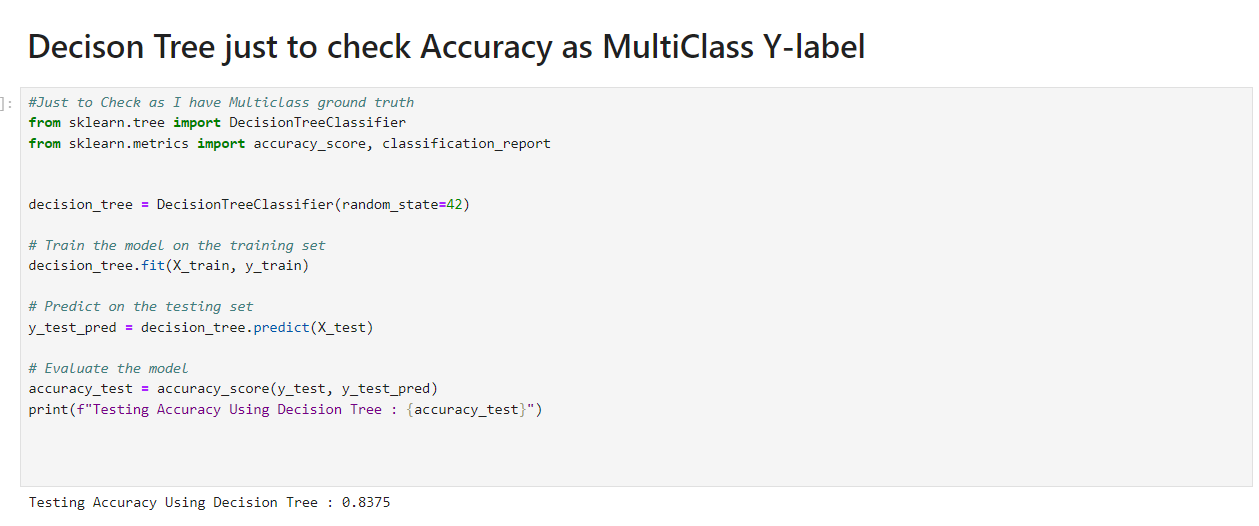
1. **Train-Test Split:**

A computer screen shot of a computer code

Description automatically generatedThe dataset was divided into training and testing sets to assess the generalization performance of both models accurately. The training set was used for model training, while the testing set was kept separate for evaluating the models' performance on unseen data.

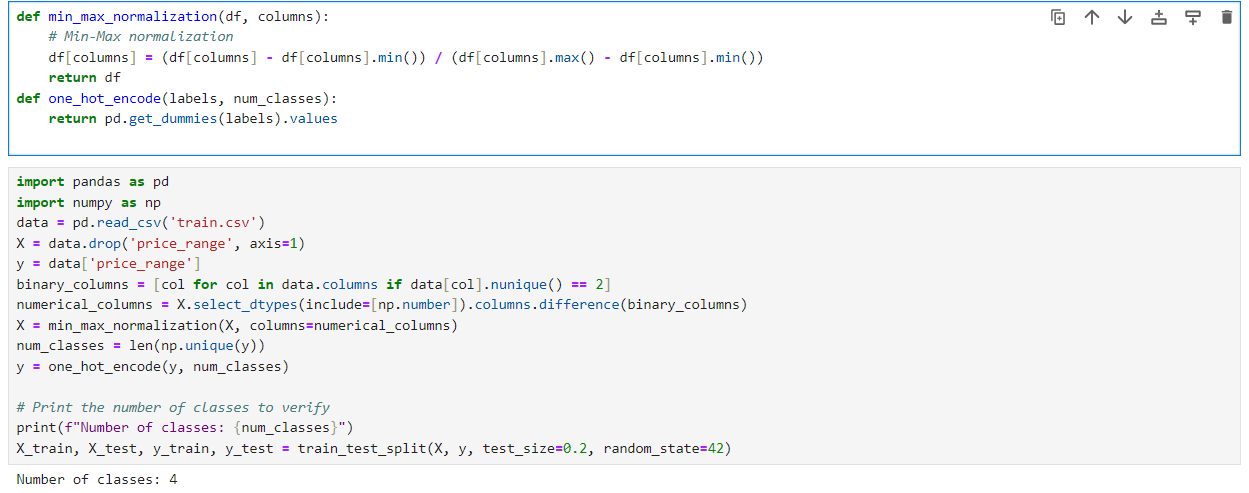
1. **Built-in KNN Model:**

By using Built-in KNN, a grid search is used to find the optimal hyperparameters for a K-Nearest Neighbors (KNN) classifier. The grid encompassed variations in the number of neighbors (3, 5, 7, 9, 11) and distance metrics ('euclidean' and 'manhattan'). Utilizing scikit-learn's GridSearchCV, the KNN model was trained and validated with a 5-fold cross-validation. The best combination of hyperparameters was then identified, and a refined KNN model was created. Finally, the accuracy of this optimized model was assessed on the validation set, providing insights into its effectiveness in predicting mobile price classifications.

But here it was observed that accuracy was very low the reason behind this was that my ground\_truth was multi class and for Multi-Classification KNN is not good approach that’s why I just check my data on decsion trees to check accuracy and found improvement in accuracy.

1. **Custom Neural Networks Implementation:**

* **Features Normalization and Ground truth One-hot-encoded.**

The min\_max\_normalization function is a common preprocessing step, especially beneficial for neural networks, where it scales numerical features to a common range. Regarding one-hot encoding, it is applied to categorical variables, converting them into a binary matrix format. That’s why we encode our ground truth.For neural networks, this is particularly useful because it allows the model to effectively interpret and learn from categorical data.

* **Code for Neural networks Implementation**

A screenshot of a computer program

Description automatically generated

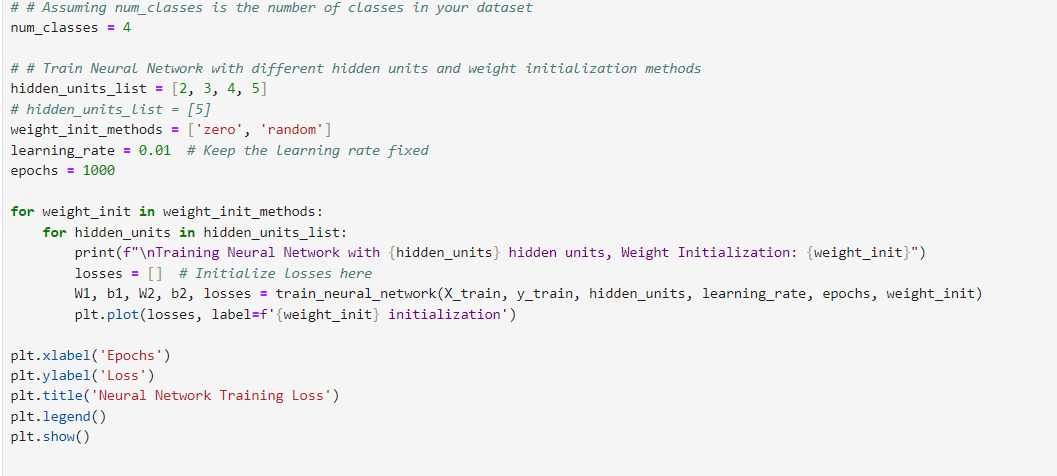
**A screenshot of a computer program

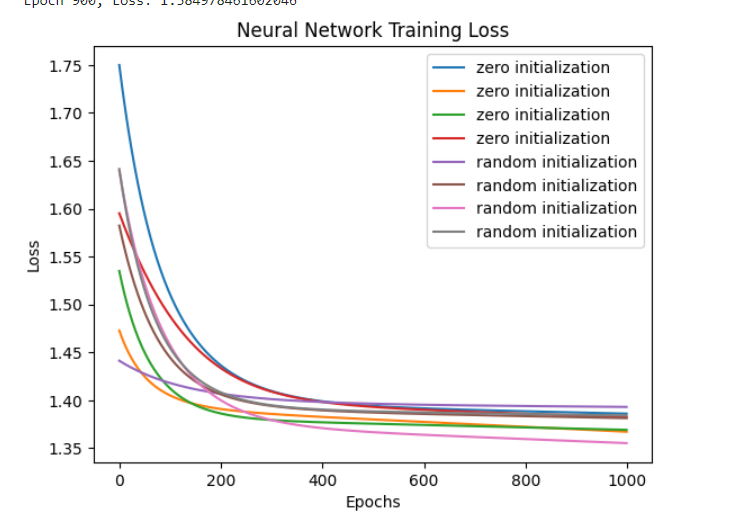
Description automatically generated**A screenshot of a computer program

Description automatically generated

* **Best Weight Initialization Method with fixed learning rate 0.01:**

I have tested neural networks with two methods of Weight Initialization (Random and Zero) and these are my findings:



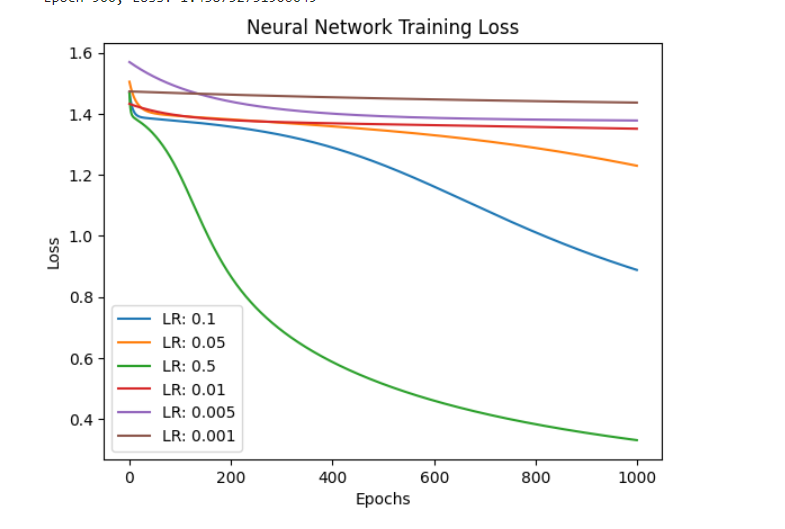


From here we can observe that for Hidden Unit 5 and with MethodRandom we have low loss. So, we can say that **Random Weight Initialization method** is best in my case.

* **Neural Networks with Random Weight Initialization and Different learning rates**

**A screenshot of a computer

Description automatically generated**

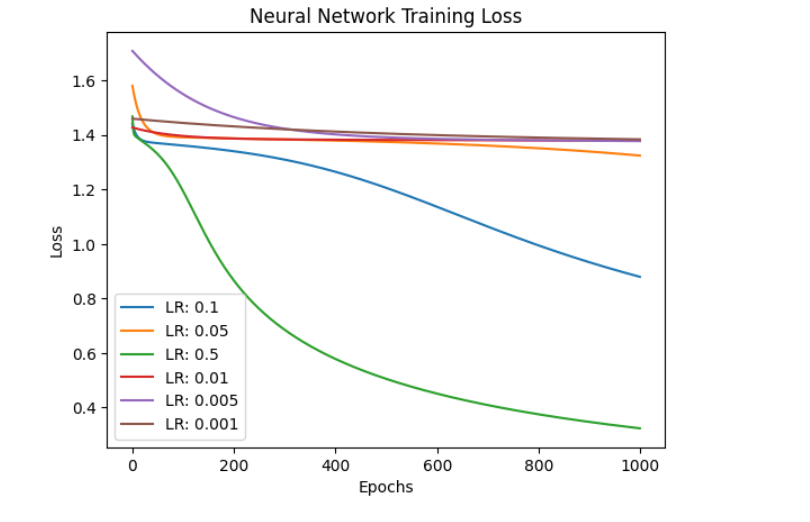
****

This graph shows loss over epochs for different learning rates, and we see that for **learning rates 0.1 and 0.5** losses are very low.

* **Neural Networks with Zero Weight Initialization and Different learning rates**

**A screenshot of a computer

Description automatically generated**

****

This graph shows loss over epochs for different learning rates, and we see that for **learning rates 0.1 and 0.5** losses are very low.

* **Neural Networks with Early stopping**

Here is code for Early stopping:

**A screenshot of a computer

Description automatically generated**

Here when I applied early stopping on my data it shows early stopping at Epoch 70272.

A close-up of a white background

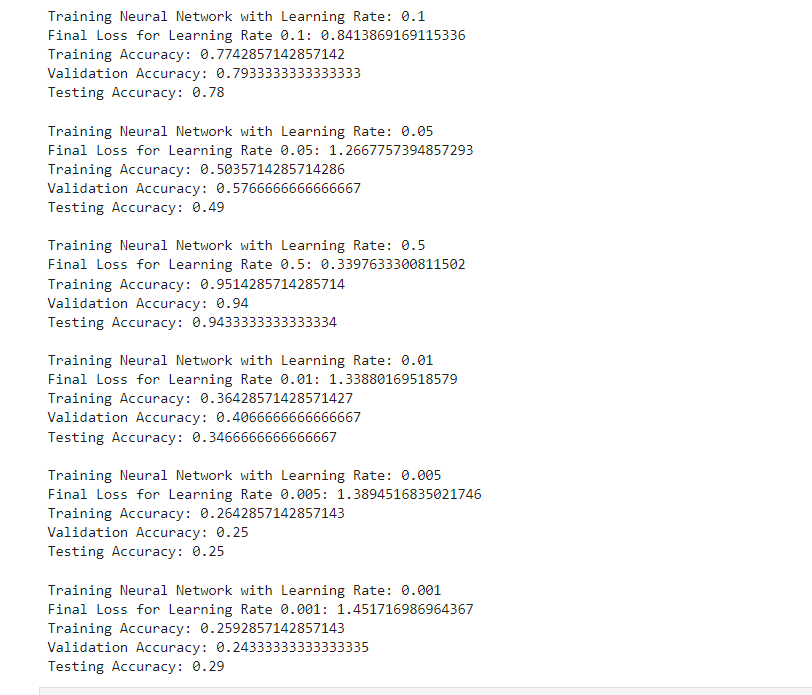
Description automatically generated

* **Neural Networks Accuracies**

Data is split into training, validation and testing set and then we performed early stopping and vvalidate our models using test sets and find their testing and training accuracies

**A computer code with text

Description automatically generated with medium confidence**



Here we have all accuracies for different learning rates and we find out that with learning rates **0.1 and 0.5** we have better **accuracies of 78% and 94%** respectively.

**Conclusion:**

In this analysis, we navigated mobile price prediction, prioritizing data preprocessing with min-max normalization and one-hot encoding. Moving to K-Nearest Neighbors (KNN), we optimized hyperparameters using grid search. Shifting to Neural Networks (NN), we compared random and zero weight initializations, explored early stopping, and fine-tuned learning rates. Our models were rigorously validated, showcasing predictive prowess on test sets. In conclusion, this report provides a concise journey from preprocessing to advanced model evaluation, offering nuanced insights and robust accuracies for mobile price prediction.