Report

This report is about the components used to create the Linear Regression model and the data preprocessing techniques used to get the most optimum results.

Step 1: Load Data

In this step, we loaded the dataset from the truncated_train.csv using pandas library

Step 2: Extract Features of the Dataset

Dataset is divided into 2 parts, Audio files and the .csv files. First we need to connect the audio files with the corresponding row in the .csv file. So we start by extracting features from each audio and storing them in a dataframe. We use Librosa library to load the audio, time series data along with the sample rate. The piptrack function is used to get the pitch of the audio. We then take mean of all the non-zero values in the pitch array to get 1 value. Similarly, we calculate the root mean square of the audio using a User defined root mean square function. Then to get the duration, we use .get_duration() function of Librosa. Spectral centroid, bandwidth and rolloff are extracted and then we take mean of them to get 1 value. Then we proceed to extract formant frequencies using parselmouth library. We use parselmouth as it is compatible with mp3 files. Finally we return all the features in the form of a dictionary.

Step 3: Preprocess the Features

Now we will convert the string attribute of age in the data into numerical values, so twenties will be 25, thirties will be 35 and so on. Then I created a user built Custom Standard Scaler class to easily be able to scale the features to make them more optimized and efficient for training. I tested a MinMax Scaler too and it didn't yield as good results as Standard Scaler so Standard Scaler is used. Then the features are converted into a DataFrame which makes it easy to manipulate. After trail and error, I found out only a few attributes yield the best results so I only excluded some attributes that didn't make a positive difference while training. The chosen attributes for training are: ['pitch', 'intensity', 'spectral_centroid', 'spectral_rolloff', 'formant_1', 'formant_2']. This will be used to train the model.

Step 4: Get Test Data

We perform the same Extract Features and Preprocessing techniques for Test data too.

Step 5: Create and Train Model

A Linear Regression model is then implemented from scratch. Learning_rate is set to 0.1 and epochs to 1000 which was finalized after trial and error. The basis of training a model is it's weights so we initialize weights to be same size as features and randomly selected for initial value, and a bias is also randomly chosen for initial value. Now in each epoch, gradient descent algorithm is run. We first predict a value from the training dataset, calculate the loss and divide it by 2 as per the formula. We then calculate the new value of gradient by first taking partial derivative of loss function for weights and bias. For bias, we simply just need to calculate it's mean for all the training set. For weights, we have to subtract the prediction of each record of training data with actual target class and multiply the same record with it too. We take sum of this for all records and then divide by total rows. And easier and efficient way to do this is to make use of the transpose functionality of a Dataframe to multiply with the error between prediction and actual value, then dividing by total number of rows. After all epochs have been run, the model will have been trained.

Step 6: Evaluation

Here we just evaluate the model using the test dataset and user defined error calculating functions that were specified in the requirements. Since continuous values can't be checked using accuracy, we resort to finding mean errors. And finally, we display a graph.

Conclusion

In summary, the report outlines the meticulous process of creating and optimizing a Linear Regression model for the given dataset. Starting with feature extraction using Librosa and Parselmouth, followed by preprocessing to standardize and convert categorical attributes, the dataset was split into training and test sets. The model was then trained from scratch using gradient descent, with evaluation conducted on the test set using mean error metrics. Despite the inherent challenges of continuous value prediction, the model demonstrated satisfactory performance, highlighting the importance of systematic methodologies and appropriate techniques in machine learning tasks.

By:

Muhid Qaiser 22i-0472
Ahmed Zubair 22i-0525

submission: 09/05/2024