Student score prediction
Sina tayebi

1- Dataset

The Student Performance dataset contains the following columns:

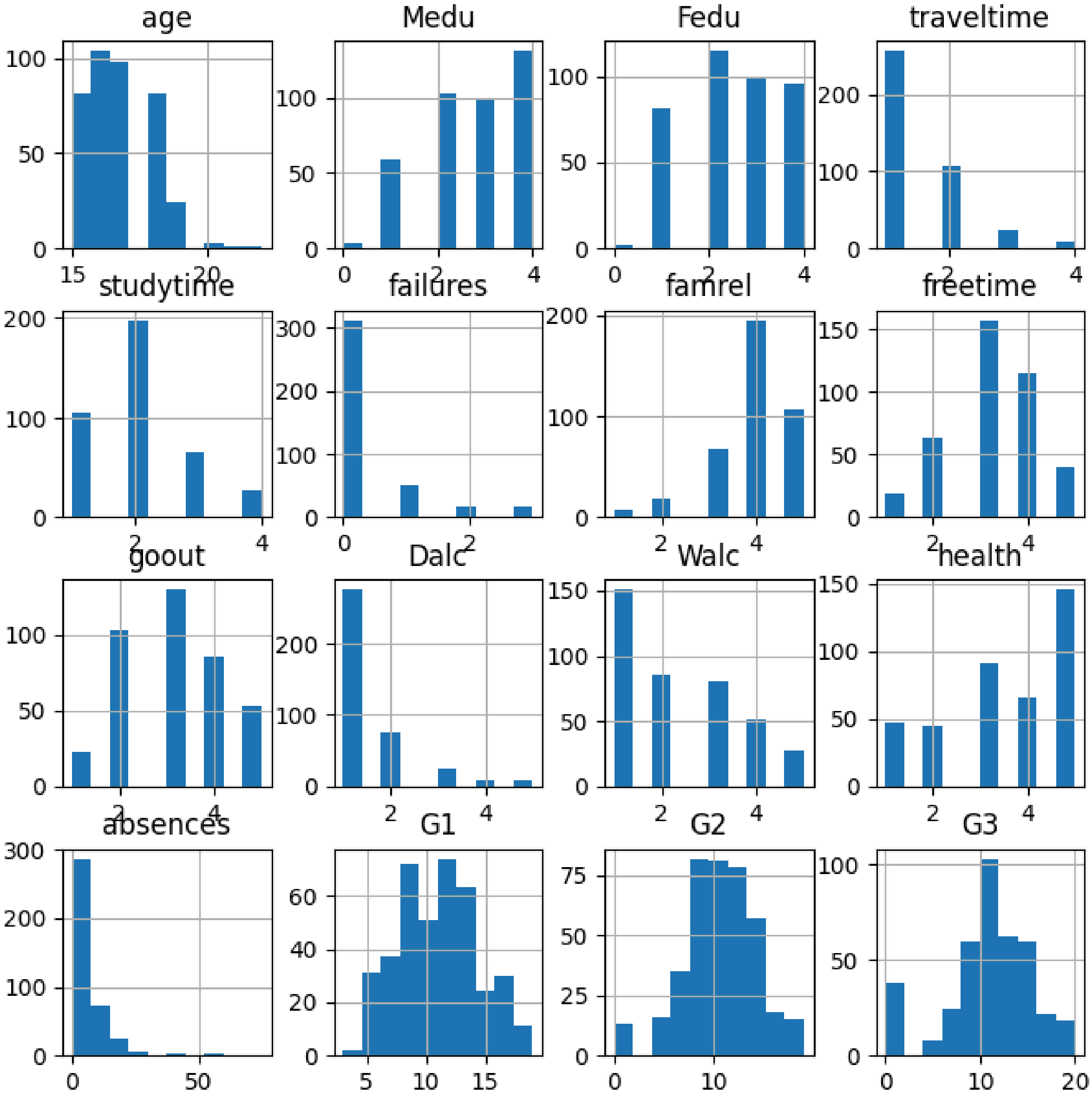
- school
- sex
- age
- address
- famsize
- Pstatus
- Medu
- Fedu
- Mjob
- Fjob
- reason
- guardian
- traveltime
- studytime
- failures
- schoolsup
- famsup
- paid
- activities
- nursery
- higher
- internet
- romantic
- famrel
- freetime
- goout
- Dalc
- Walc
- health
- absences
- G1 (First period grade)
- G2 (Second period grade)
- G3 (Final grade)

The task is predicting the final score of students, which is the G3 column. For solving this problem, we use MLPs and investigate different activation functions, optimizers and regularisation techniques.

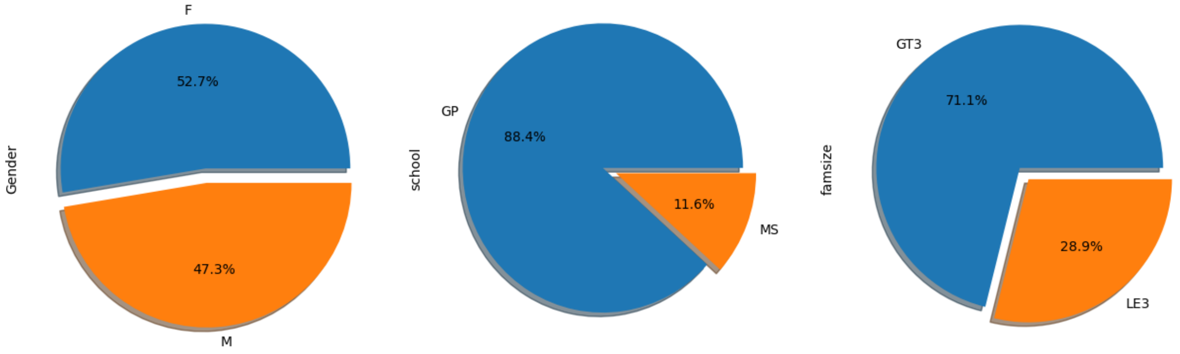
2- EDA

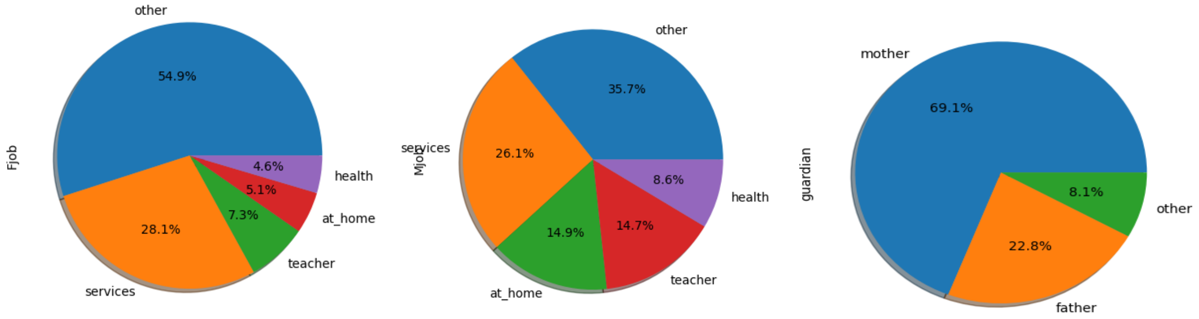
2-1- distributions

Let us first see the distribution of numerical features.

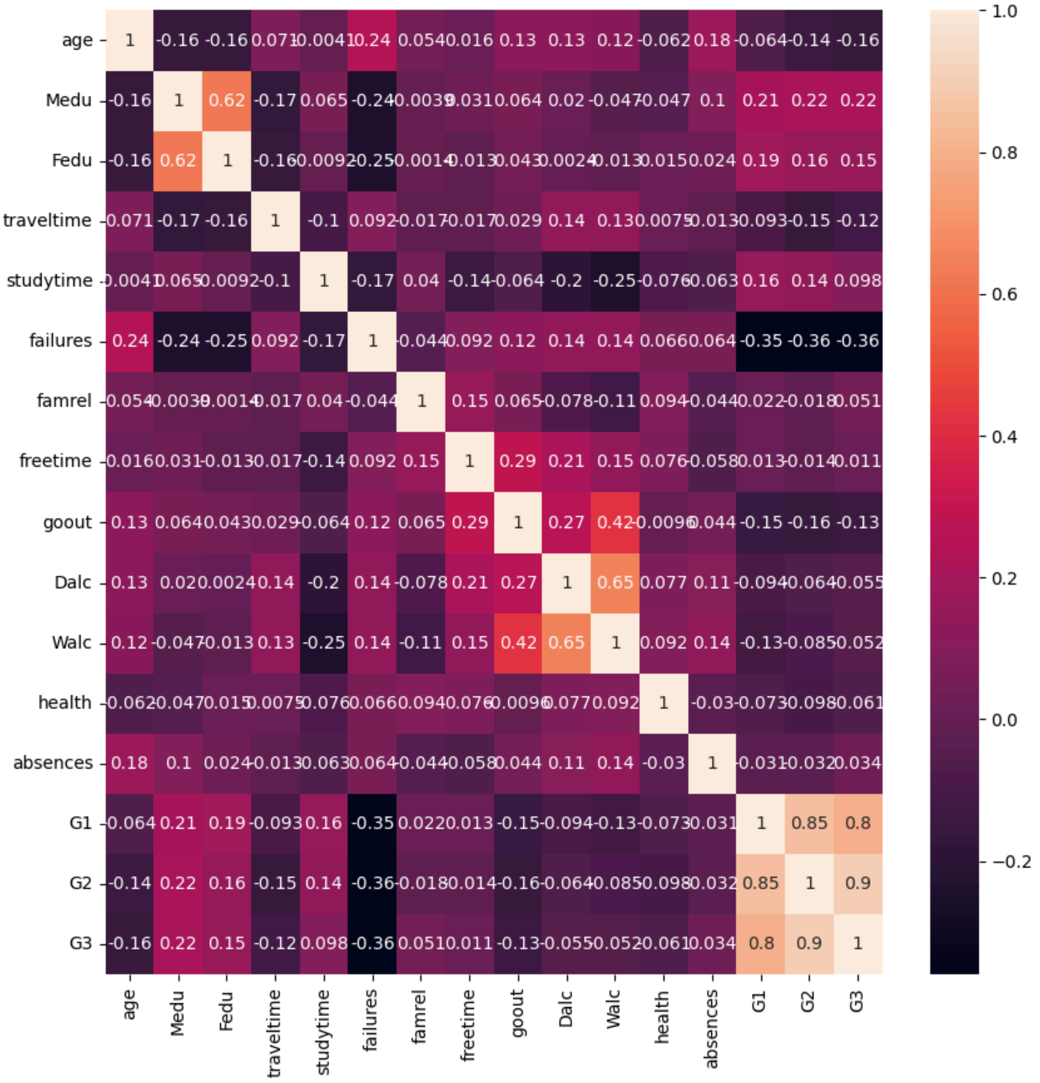


As we can see, the G2 and G3 feature has some outlier which is in point 0. We may need to handle these further. Now let us see categorical data.





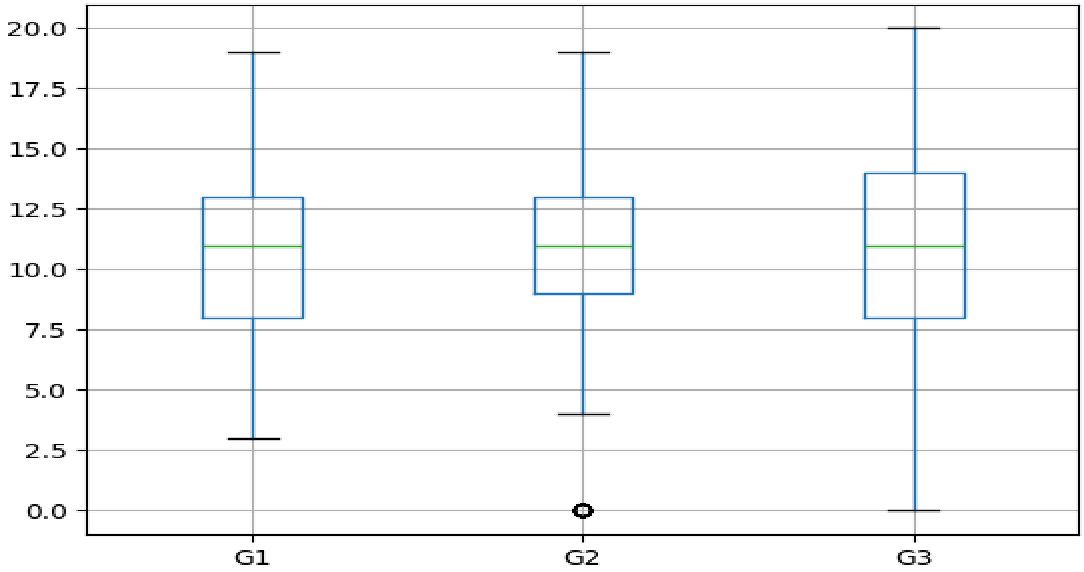
2-2- Correlations



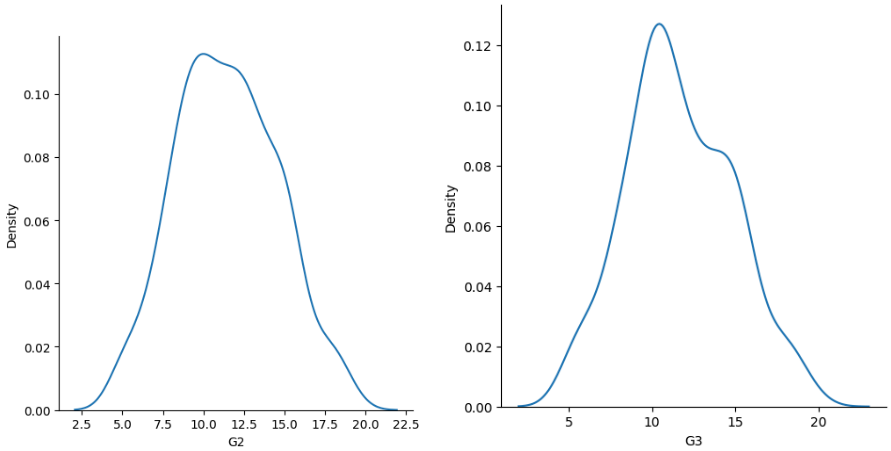
As we can see, the most correlated features with the target are, G1, and G2, also some features like failures have significant negative correlation with the target, so we expect these features to make huge participation in the training regression MLP.

2-3- outliers

Due to the sensitivity of the linear regression to outliers, we should handle it carefully! Let us see the boxplot of features.



As we can see,G2 and G3 have outliers and must be removed. We removed them using the IQR method.



Distribution of those features after removing outliers are now close to normal distribution and it is sufficient for linear regression algorithm.

3- Feature transformation

We use ordinal and one-hot encoding to encode categorical features and standardscaler to scale numerical features using sklearn pipeline.

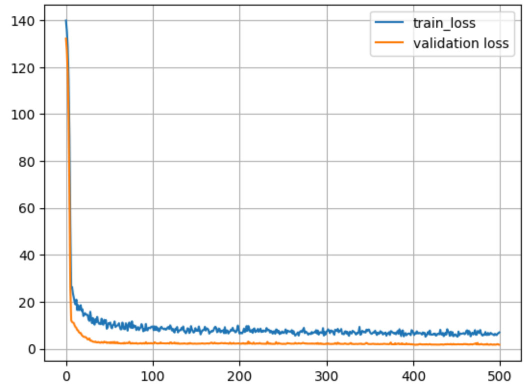
4- Training

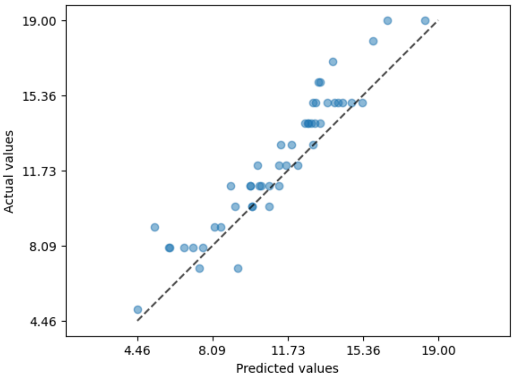
4-1- Hyperparameter tuning

We tuned hyper parameters like learning_rate, hidden layer size and number of iteration and also number of epochs using the GridsearchCV package.

4-2- Model with Adam, MSEloss, ReLU and dropout with p = 0.5

The reported result is as follows.

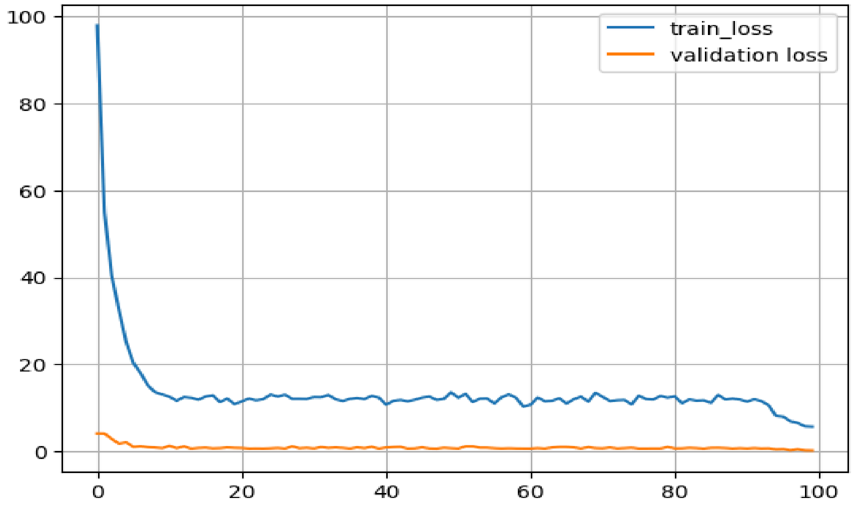




R2 Score: 0.7998968909999515, Test MSE: 2.109375238418579

This is our base result, as we can see the train loss and validation loss both converging to zero, so a few more epochs may could improve the performance, also the test r-2 score shows that model could predict 80% of variability of the data and it's not a bad result due to the low amount of data.

4-3- Model with Adam, MSEloss, Tanh and dropout with p = 0.5

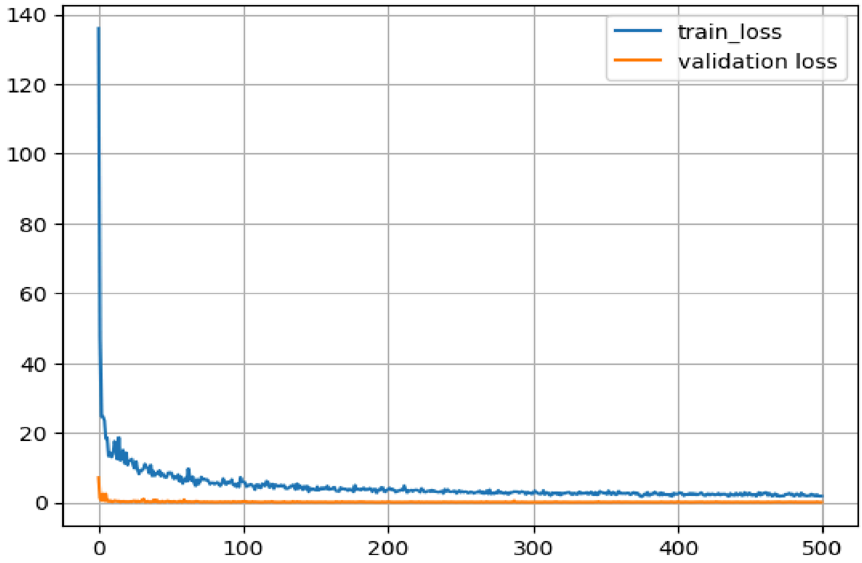


test_mse: 3.120669322264028   test_r-2: 0.703961792068118

We used the TanH activation function instead of ReLU. as we can see, the model performance over 100 epochs did not as well as ReLU, this is because of using a dropout lauer with probability equal to 0.5, so the training process using this combination is not as good as ReLU and dropout.

4-4- Model with SGD, MSEloss, LeakyReLUand dropout with p = 0.5

We investigate the optimisers and their convergence conditions in theoretical questions, and we know that Adam with tuned parameters is always better than SGD! But in this scenario, we don't really know if we choose right parameters for the Adam, so we tested the SGD with LeakyReLU activation function over 500 epochs.



test_mse: 1.6415855299401532   test_r-2: 0.8442731387836306

The result had a significant improvement! It shows that the SGD with default parameters performs better than Adam with default parameters.

5- Conclusion

According to the previous section, we can understand that the most important steps to improve the result is preprocessing the data. The r-2 score before removing outliers was 0.73 in best model architecture. So, first we must ensure that data is sufficient enough to feed the network. After that we need to precisely tune our hyperparameters like number of epochs, learning rate, dropout probability etc. also choosing the right optimiser, loss function and activation function is crucial, which we saw in section **4-4.**