Asal Rahbari – 99222044

Students' Grades

In this task we're trying to train a model on our data that will later predict the performance of a student based on some given information about them.

1) Loading Data:

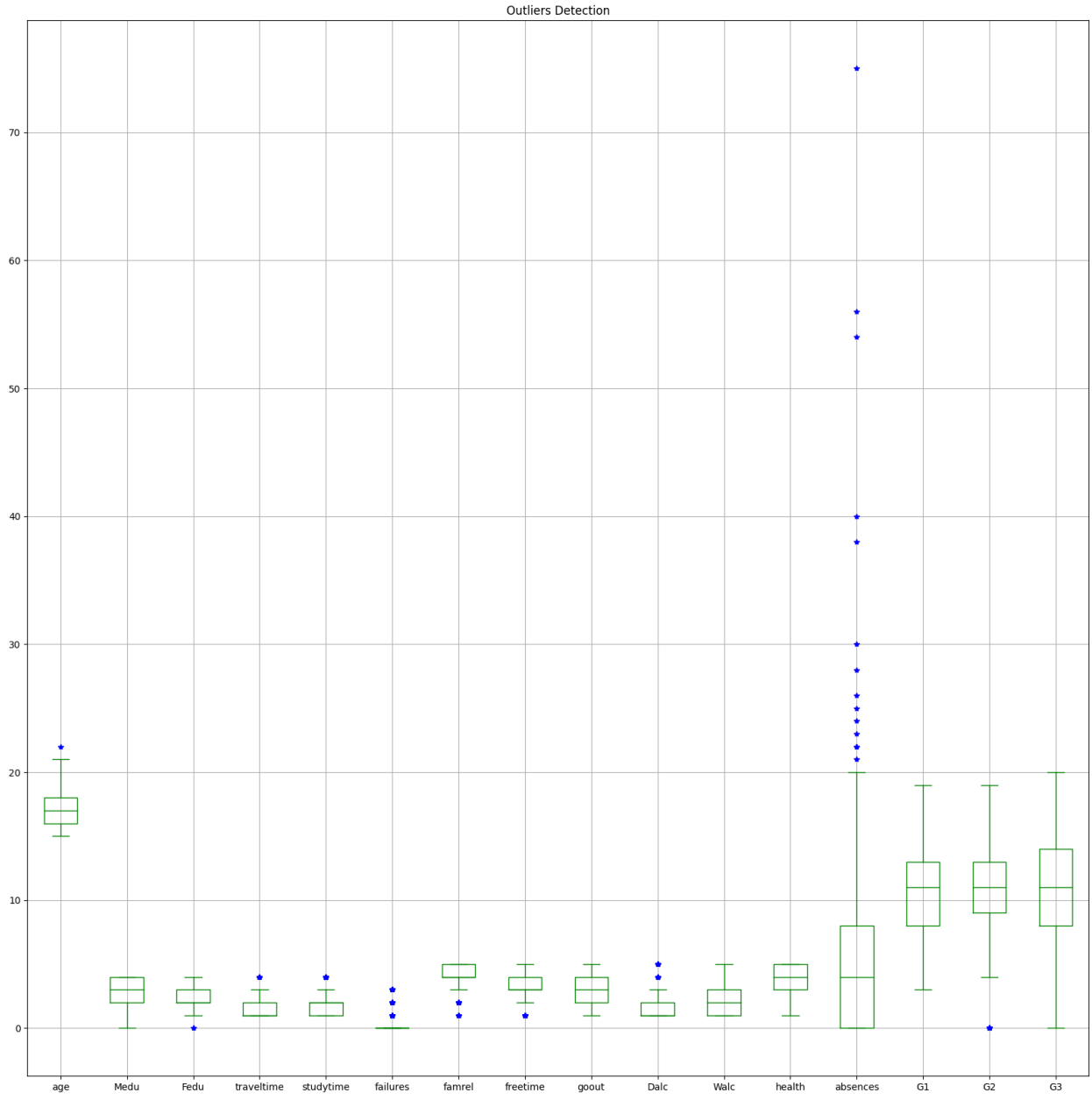
First of all, we need to become familiar with our dataset, its unique characteristics and the meaning behind each of the features. Getting to know things like the number of unique values of each feature, the type of each column, the min and max value of that feature if it's numerical, can significantly improve our performance in the Data preprocessing step. For instance, when the number of unique values of a feature is two, we'd know that the best representation for this feature is binary form. Likewise, when there are more than two unique values but not too many, we'd know that one hot encoding might be the best choice.

2) Null Values:

In this step we check for any null or meaningless values in our dataset. I defined a method called ‘null_search’ for looking for null values. Using this method, it seems that there aren’t any values missing. To make sure that all of the given values are meaningful and are not things like ‘unknown’ or other meaningless phrases, I got the list of all of the unique values for each feature to check their validity. Fortunately, all of the values seemed to be reasonable so I moved on to the next part.

3) Outliers:

Now, it’s time for checking for the existence of any outliers. We can easily do this by plotting a boxplot for each of our features. The boxplot can be customized to our liking.

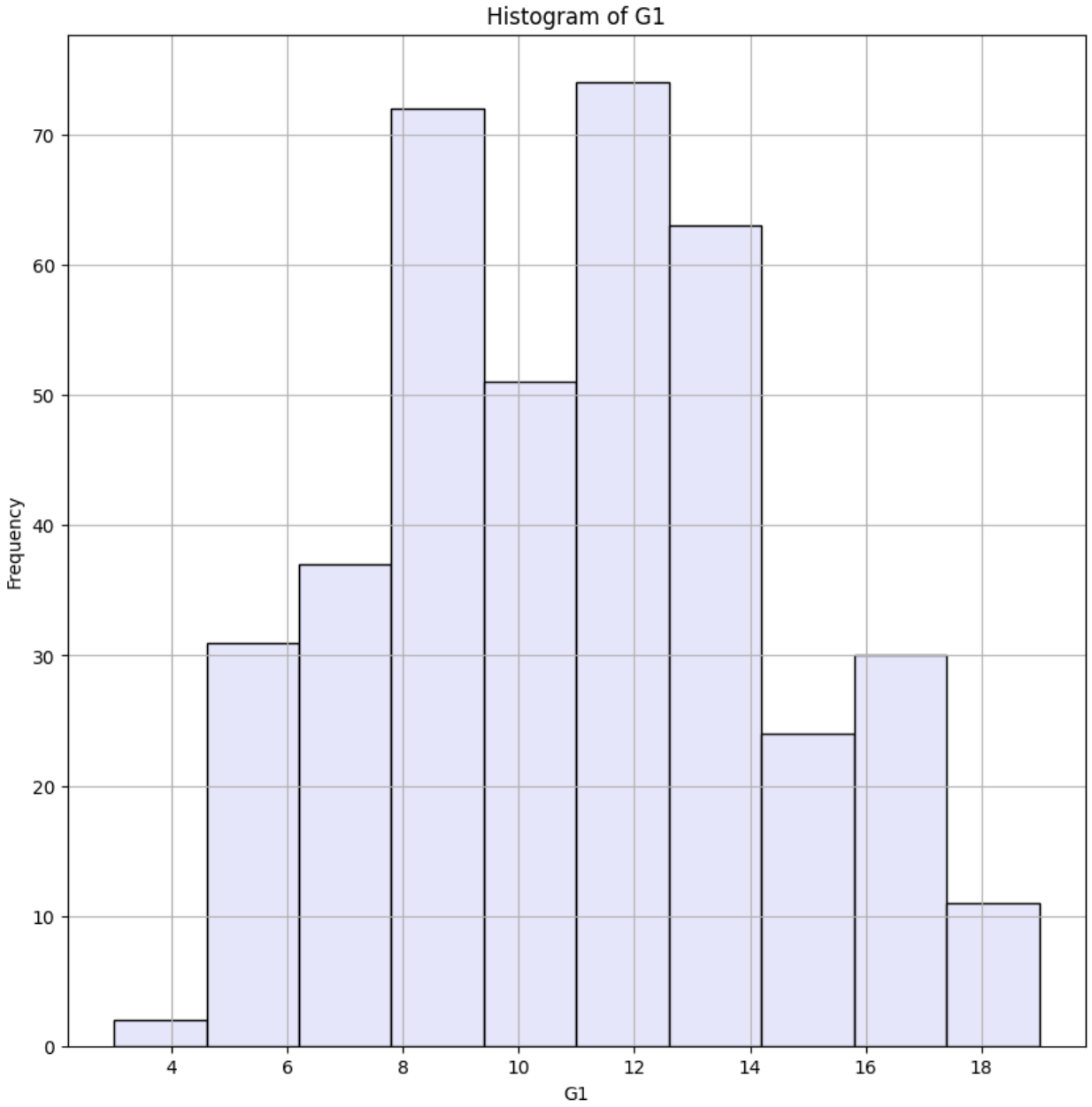


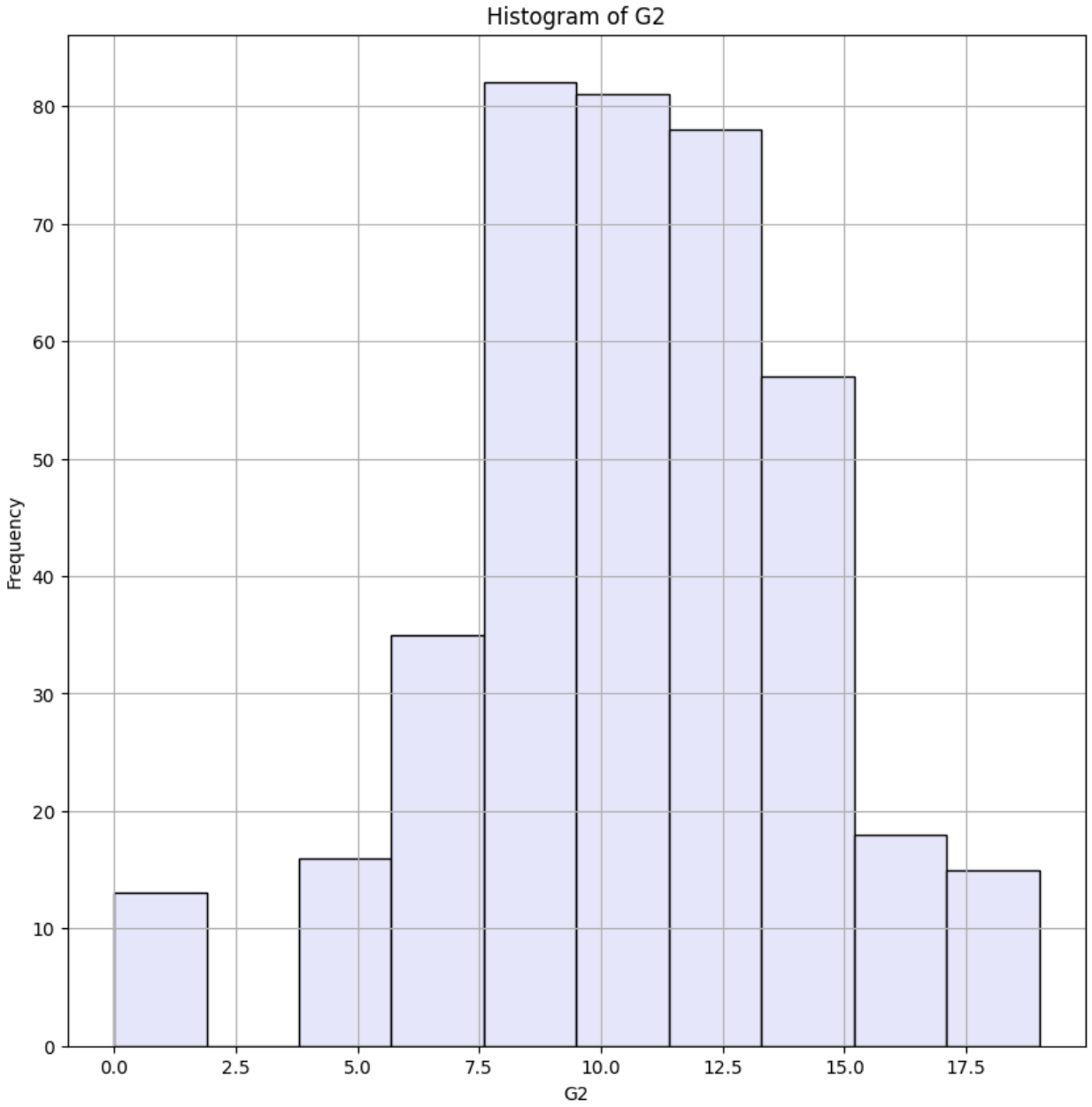
4) Visualization:

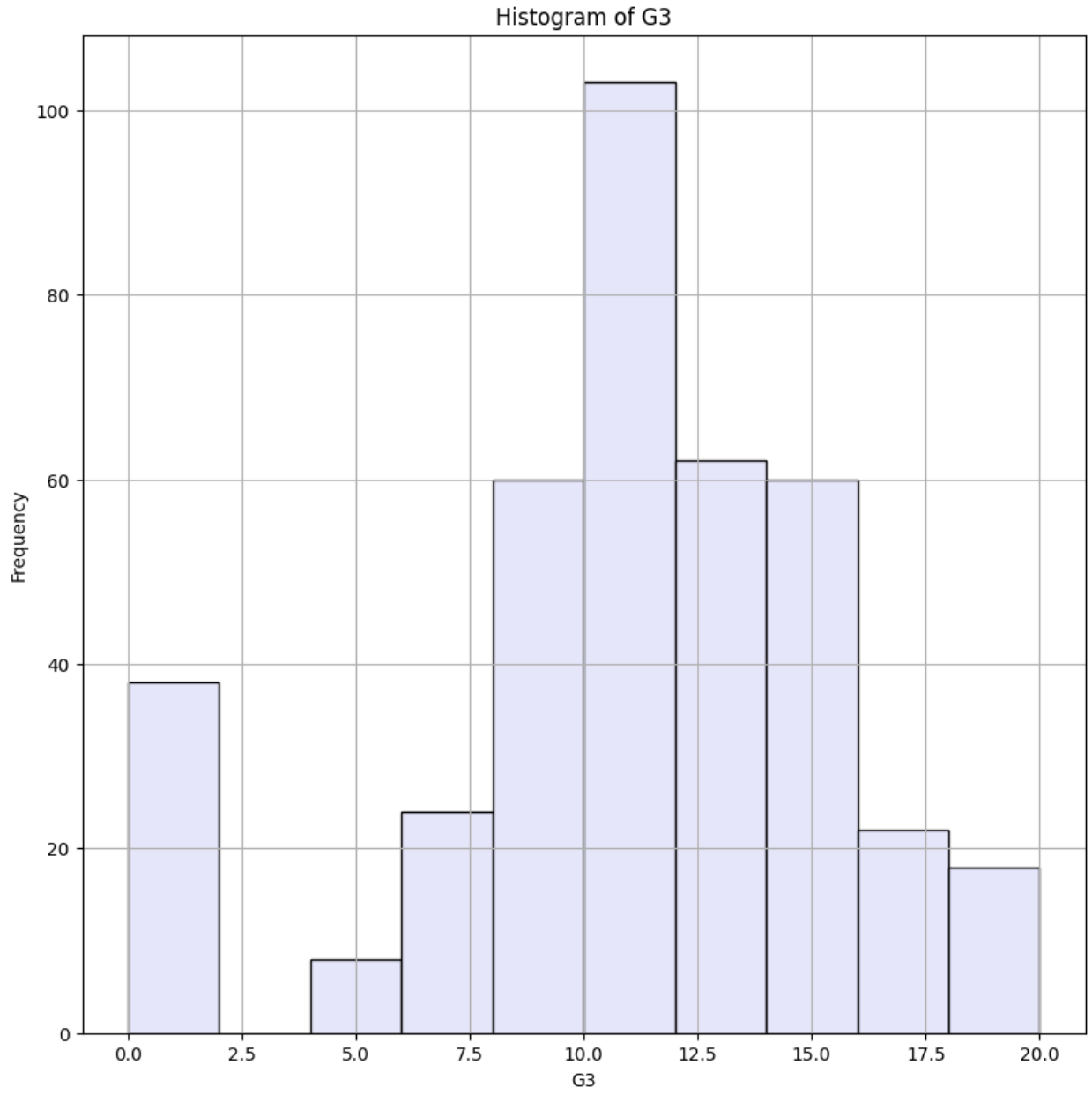
Visualizing data can help us to interpret the underlying characteristics of it better.

First I plotted histograms for some of our numerical columns: age, absences, G1, G2 and G3.

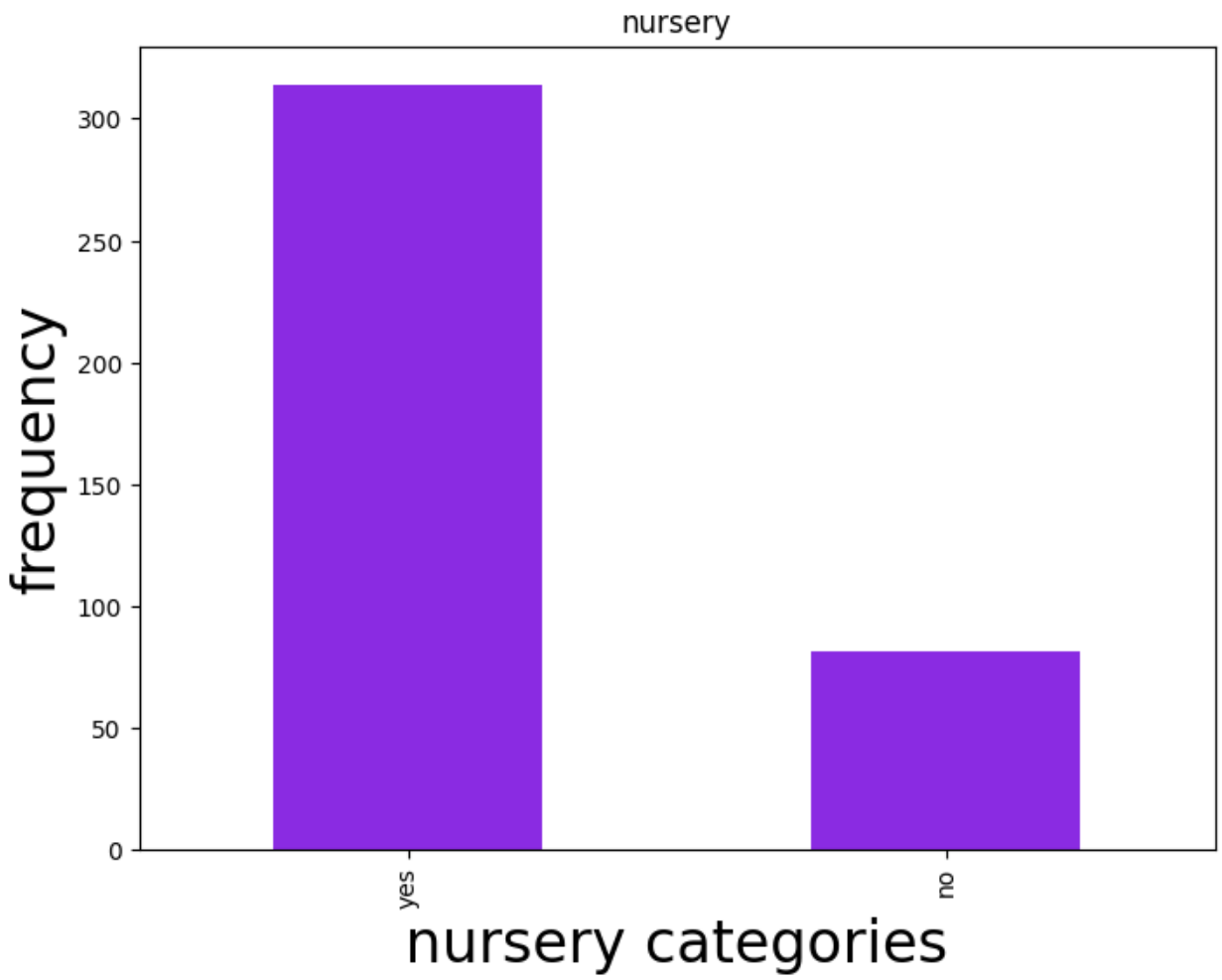
Histograms can effectively help us to see the distribution and mode of data.

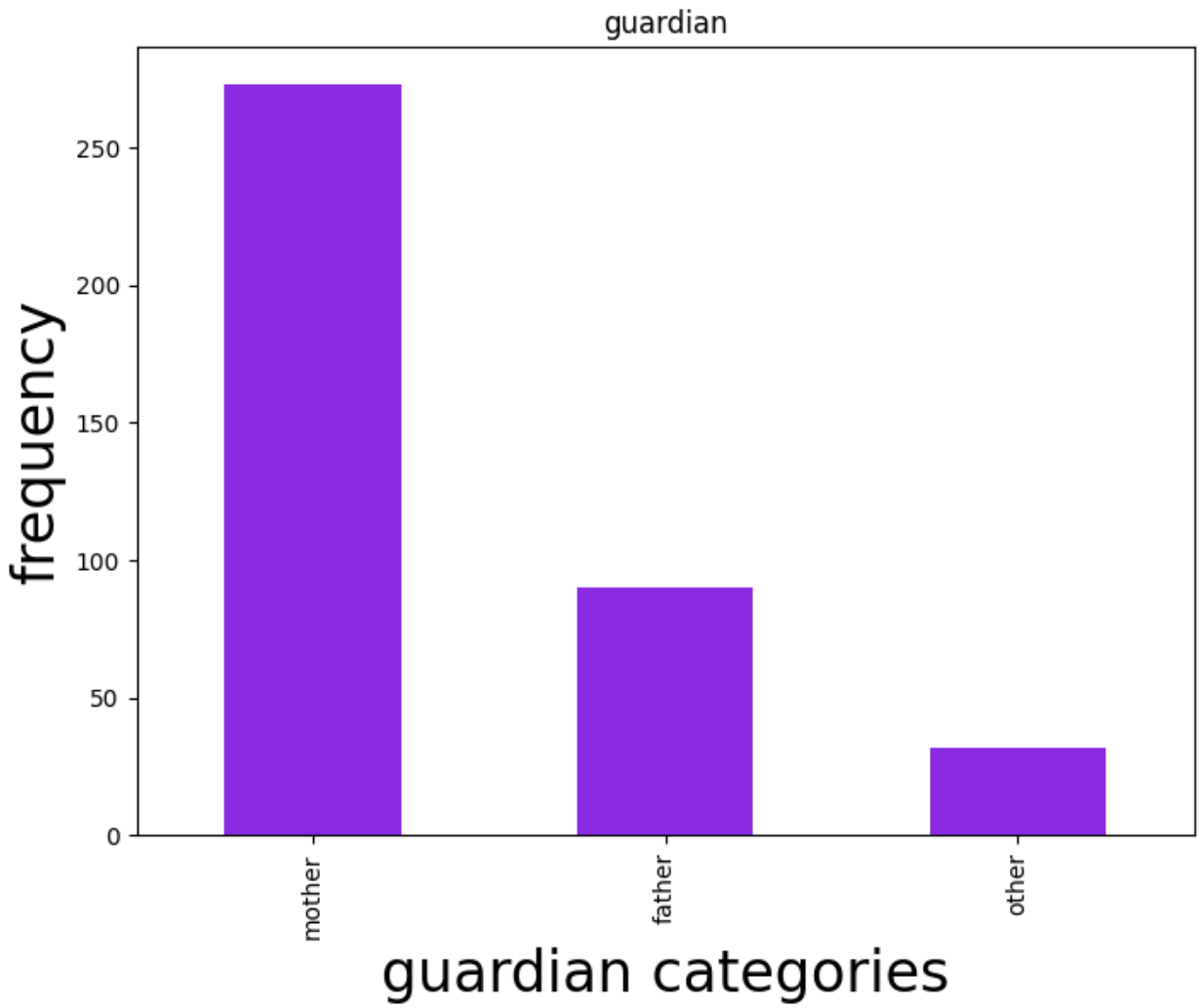

Histogram of G1
Frequency
G1
0
10
20
30
40
50
60
70
4
6
8
10
12
14
16
18


Histogram of G2
Frequency
G2
0
10
20
30
40
50
60
70
80
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5



Then for all of the categorical columns I plotted bar chart to see the frequency of each of the classes for every feature:


nursery
frequency
300
250
200
150
100
50
0
yes
no
nursery categories



5) Encoding:

For our model to be able to interpret features they must be all in the numeric form. So, we're encoding all of the categorical columns.

The easiest and best way to encode features with only two unique values is binary encoding and that's what we're doing here.

Next for all of the features that seem numeric but are actually categorical with numbers as their categories' labels, we applying one-hot encoding.

6) Target:

Since the performance of students is asked in the given task, I got the average of three given grades for students and considered the result as a metric for evaluating students' performance. So now the column 'Grade' is our target feature.

7) Scale and Normalization:

It's essential to do this step for all of our numerical columns to avoid further problems in our training process. Having different features with different scales can obviously affect our model's performance for the worse.

8) Model Training:

We first train a 3 layer MLP model with Relu activation function, sgd optimizer and MSE loss function. Then, we define a method for applying the training loop with 500 epochs and learning rate 0.001.

9) Evaluation:

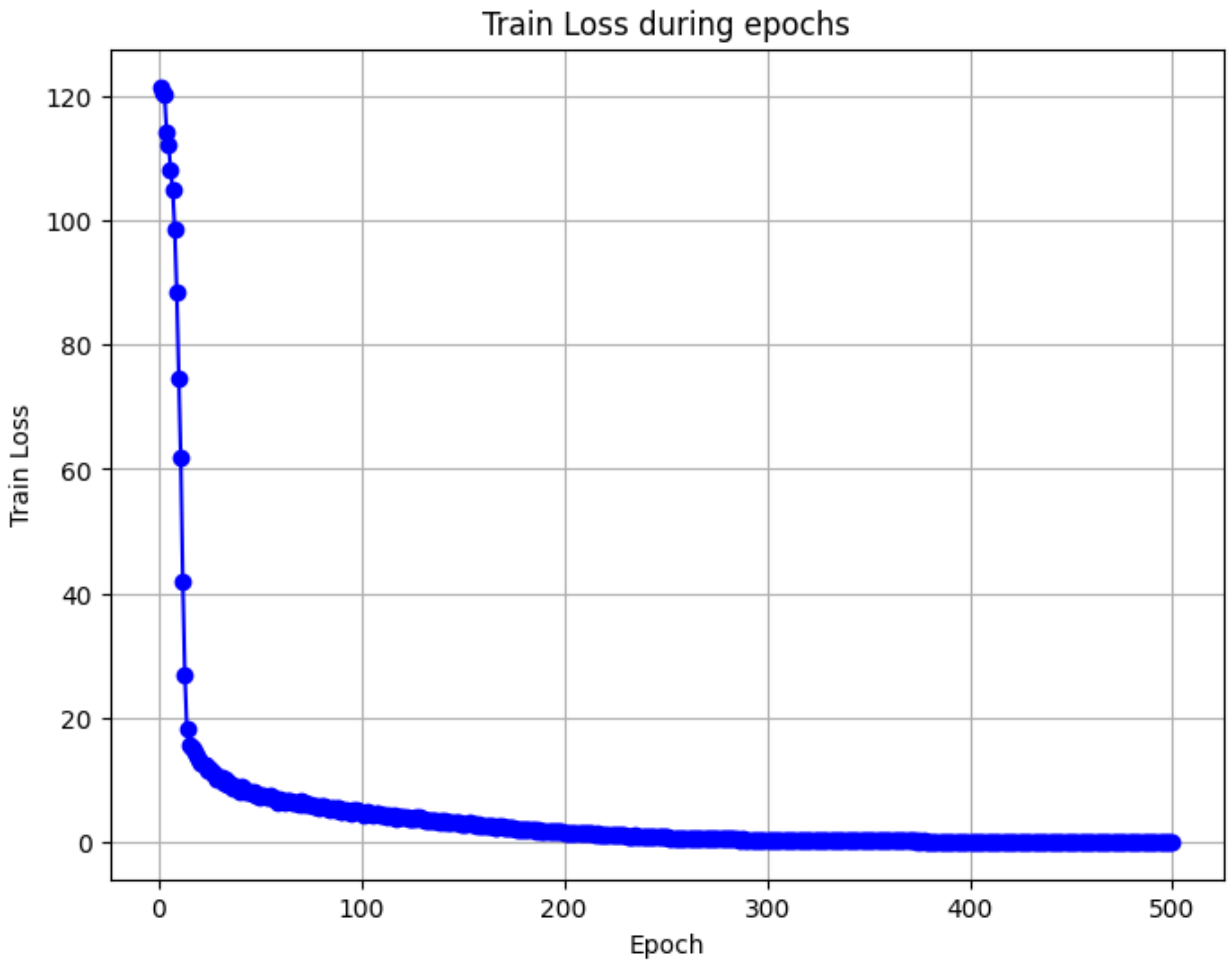
In this part we define a method for evaluating our model on the test data. This evaluation includes mean squared error, mean absolute error and R2 score.

Then, we define a method for generating a table for evaluation metrics. Using this method, we can easily display the performance of our model.

	Evaluation Metric Type	Metric
0	Mean_SQUARED_ERROR	24.129277
1	Mean_ABSOLUTE_ERROR	4.075065
2	R2	-0.629181

10) Visualization:

In this part, we generate a data frame consist of train and test loss during the given epochs. Then, we define a method for illustrating the trend of loss during training epochs using the data frame we've just created.

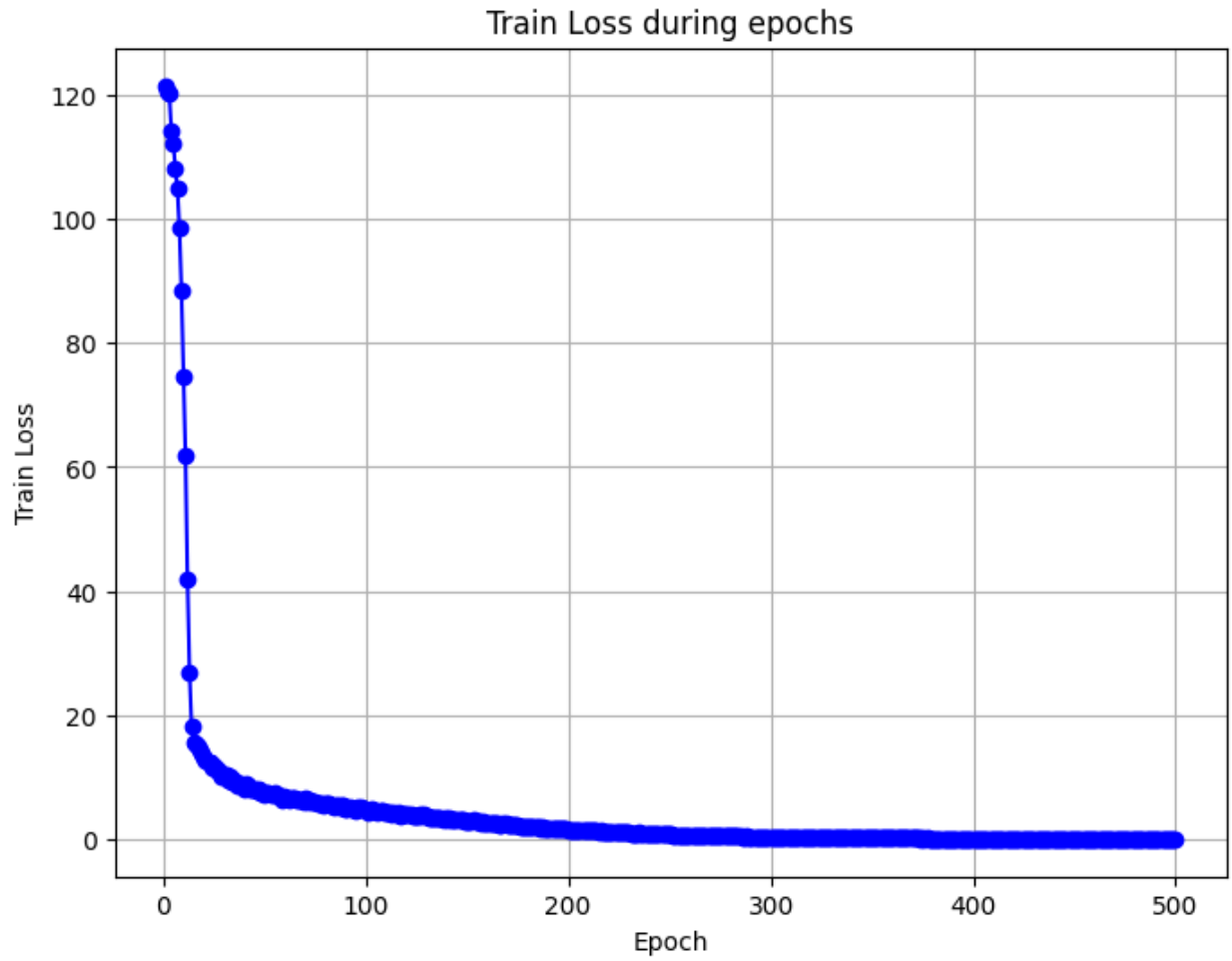


11) Enhancement:

For this part of the task we should try to improve our model to perform better on our data using methods like drop out or using leaky relu instead of relu. For this purpose, we define another model which includes drop put and leaky relu. Then we again train this model using the method ‘train_loop’ which we have defined earlier and evaluate the new model using the evaluation method again. As you can see the results with the new model are better than the previous one.

	Evaluation Metric Type	Metric
0	Mean_SQUARED_ERROR	18.373346
1	Mean_ABSOLUTE_ERROR	3.583173
2	R2	-0.240547

And again we plot the changes of loss during 500 epochs.



The End