pouya mirzaie zadeh

## Problem Description

In this practical exercise, you will work on predicting student performance using the Student Performance dataset. The dataset contains information about students' demographics, family background, and academic performance. Your task is to build an MLP model using PyTorch to predict students' performance accurately.

## Dataset Description

The Student Performance dataset contains the following columns:

- school
- sex
- age
- address
- fam size
- 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)

### Tasks

1. **Data Loading and Preprocessing**

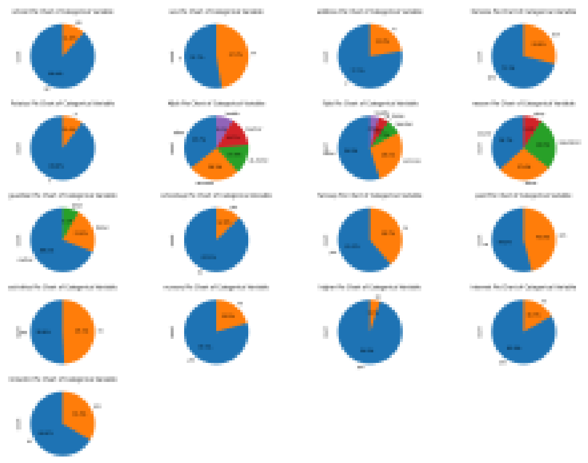
## Dataset Overview

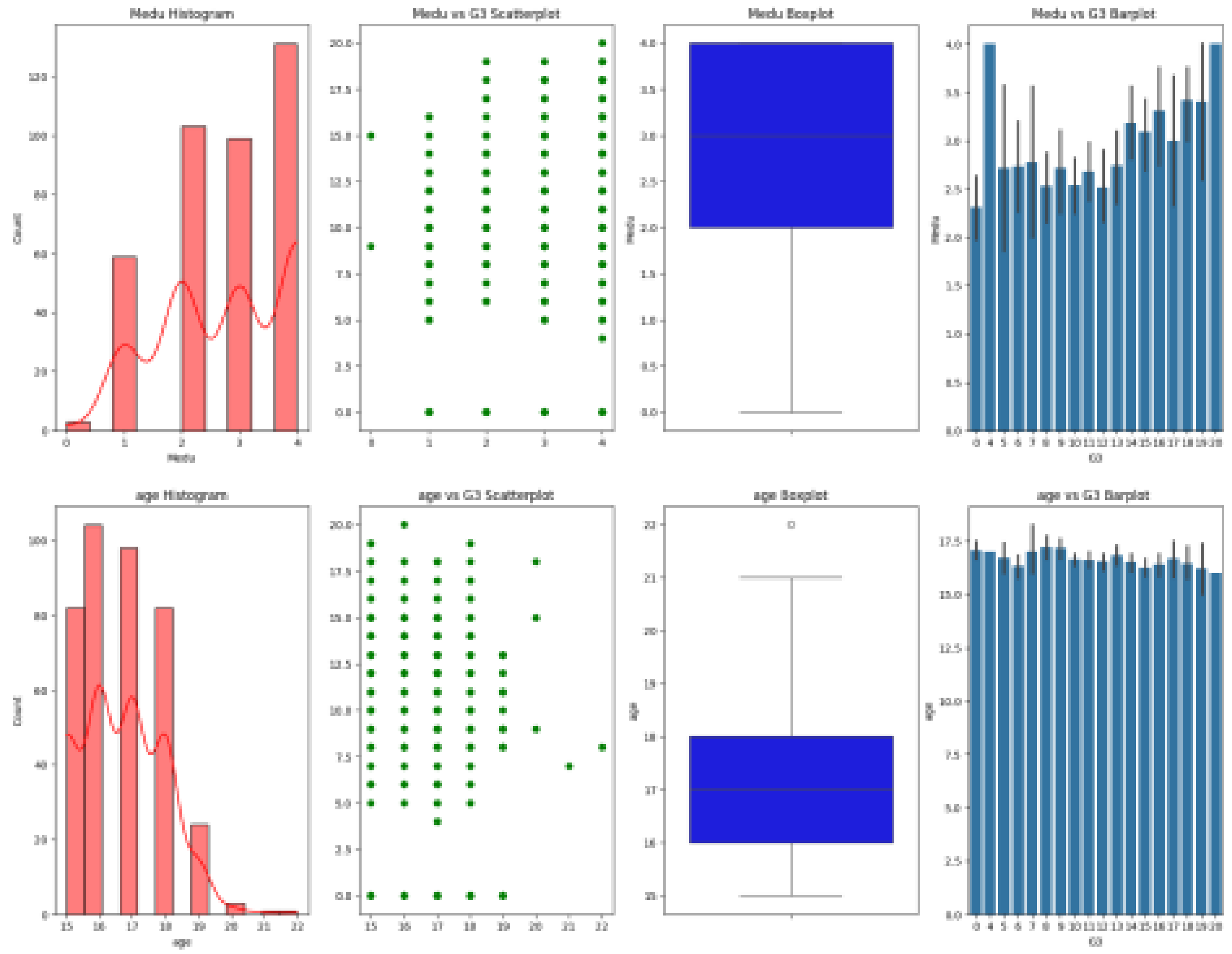
The dataset used for this project consists of 395 records, each with 33 attributes. The attributes and their types are as follows:

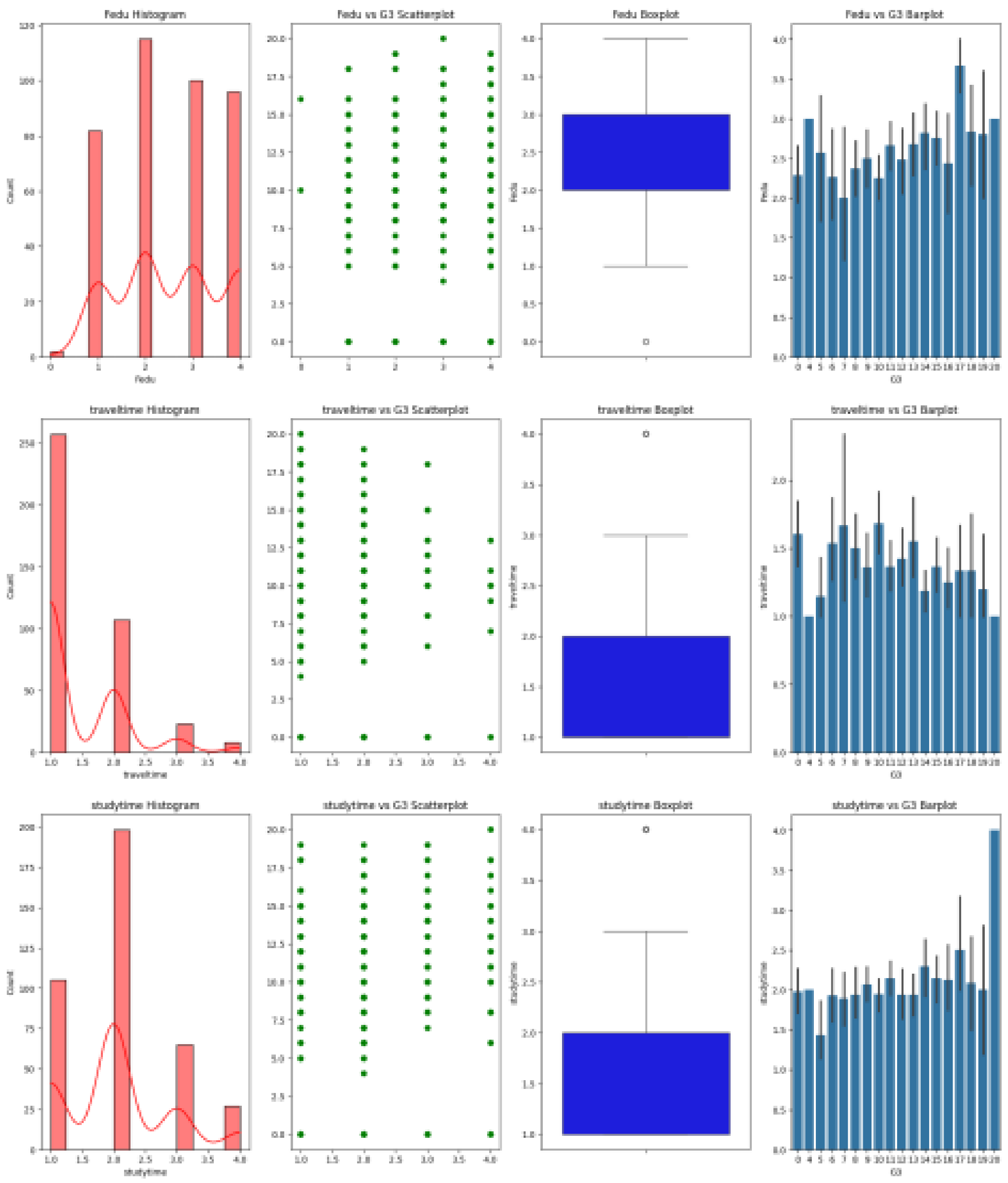
- `school`: object
- `sex`: object
- `age`: int64
- `address`: object
- `famsize`: object
- `Pstatus`: object
- `Medu`: int64
- `Fedu`: int64
- `Mjob`: object
- `Fjob`: object
- `reason`: object
- `guardian`: object
- `traveltime`: int64
- `studytime`: int64
- `failures`: int64
- `schoolsup`: object
- `famsup`: object
- `paid`: object
- `activities`: object
- `nursery`: object
- `higher`: object
- `internet`: object
- `romantic`: object
- `famrel`: int64
- `freetime`: int64
- `goout`: int64
- `Dalc`: int64
- `Walc`: int64
- `health`: int64

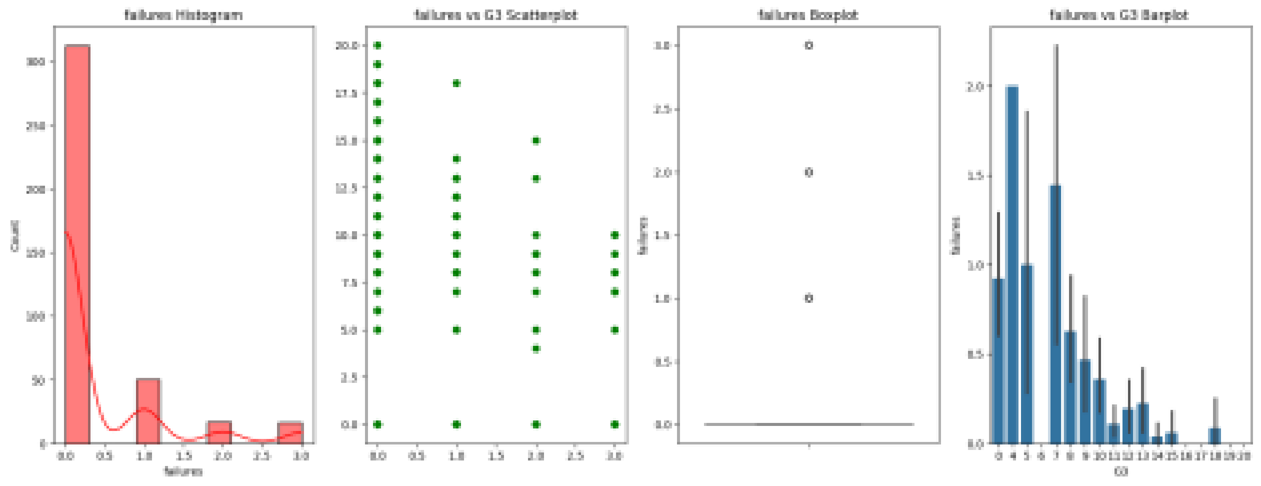
- `absences`: int64
- `G1`: int64
- `G2`: int64
- `G3`: int64

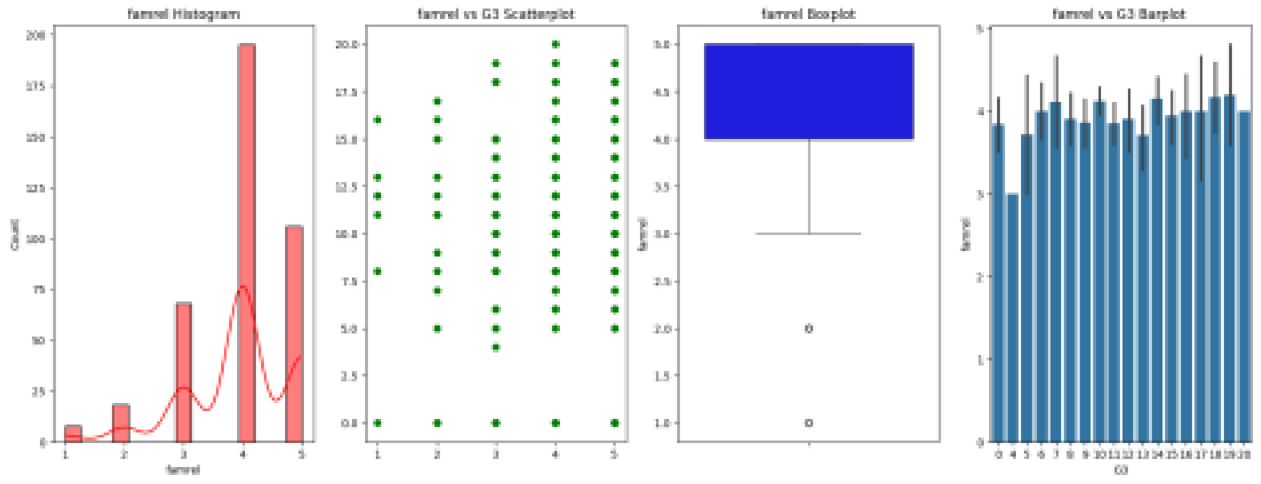
The dataset is clean, with no null values in any of the records.
This diverse set of attributes provides a comprehensive profile of each student, making it an excellent dataset for a neural network project. The variety of data types (object and int64) also adds complexity to the data preprocessing stage, which will be discussed in the next section.

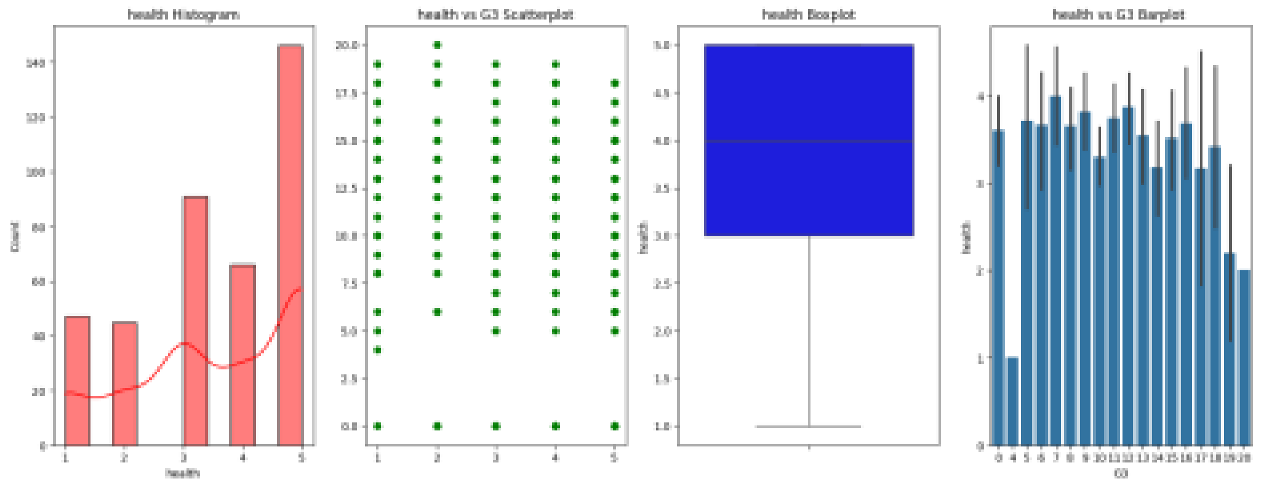


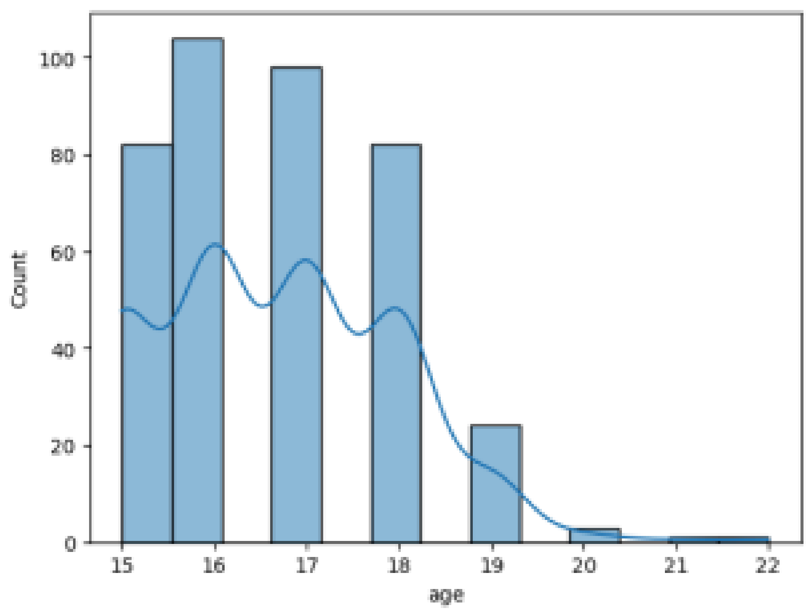


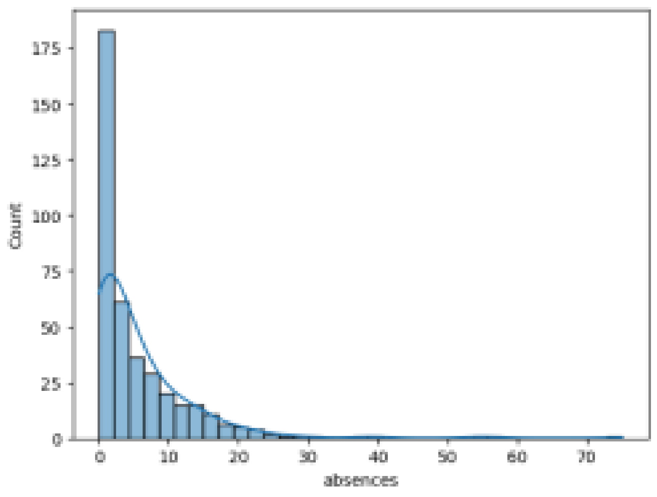

Fedu Histogram
Fedu vs G3 Scatterplot
Fedu Boxplot
Fedu vs G3 Barplot
traveltime Histogram
traveltime vs G3 Scatterplot
traveltime Boxplot
traveltime vs G3 Barplot
studytime Histogram
studytime vs G3 Scatterplot
studytime Boxplot
studytime vs G3 Barplot
Count
Fedu
traveltime
studytime
G3

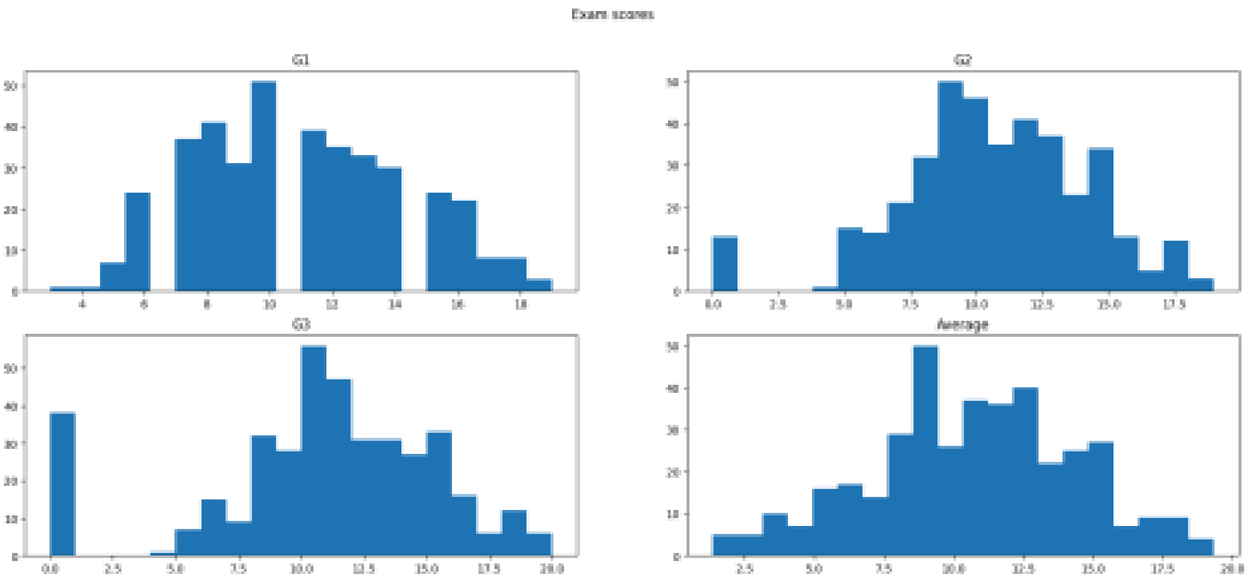

failures Histogram
Count
failures
failures vs G3 Scatterplot
failures Boxplot
failures
failures vs G3 Barplot
failures
G3

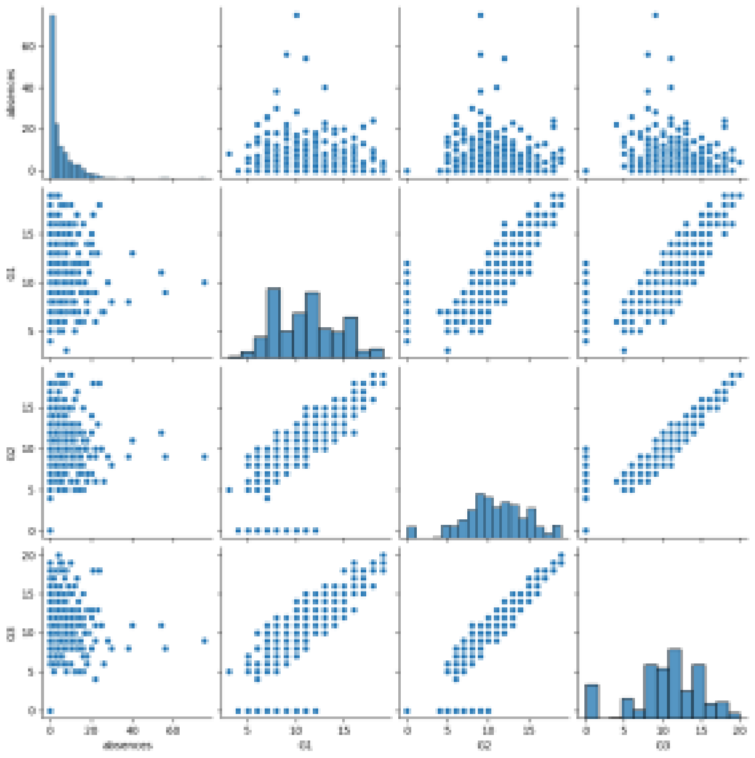

famrel Histogram
Count
famrel
famrel vs G3 Scatterplot
famrel Boxplot
famrel
famrel vs G3 Barplot
famrel
G3


health Histogram
Count
health
health vs G3 Scatterplot
health Boxplot
health
health vs G3 Barplot
health
G3


Count
age
0
20
40
60
80
100
15
16
17
18
19
20
21
22


Count
absences
0
25
50
75
100
125
150
175
0
10
20
30
40
50
60
70


Exam scores
G1
G2
G3
Average



## Data Preprocessing

### Label Encoding

Label encoding is a common preprocessing step used for transforming categorical data into a format that can be understood by machine learning algorithms. In this project, label encoding was applied to several columns in the dataset.

The following columns in the dataset were identified as containing string (object) type data:

- `school`: [‘GP’, ‘MS’]
- `sex`: [‘F’, ‘M’]
- `address`: [‘U’, ‘R’]
- `famsize`: [‘GT3’, ‘LE3’]
- `Pstatus`: [‘A’, ‘T’]
- `Mjob`: [‘at_home’, ‘health’, ‘other’, ‘services’, ‘teacher’]
- `Fjob`: [‘teacher’, ‘other’, ‘services’, ‘health’, ‘at_home’]
- `reason`: [‘course’, ‘other’, ‘home’, ‘reputation’]
- `guardian`: [‘mother’, ‘father’, ‘other’]
- `schoolsup`: [‘yes’, ‘no’]
- `famsup`: [‘no’, ‘yes’]
- `paid`: [‘no’, ‘yes’]
- `activities`: [‘no’, ‘yes’]
- `nursery`: [‘yes’, ‘no’]
- `higher`: [‘yes’, ‘no’]
- `internet`: [‘no’, ‘yes’]

- `romantic`: [‘no’, ‘yes’]

For columns where the data was binary (i.e., ‘yes’ or ‘no’), the data was replaced with ‘1’ for ‘yes’ and ‘0’ for ‘no’. This transformation allows these categorical features to be used in the subsequent stages of the machine-learning pipeline.

### Dummy Encoding

Dummy encoding is another common preprocessing step used for transforming categorical data into a format that can be understood by machine learning algorithms. In this project, dummy encoding was applied to several columns in the dataset.

The following columns in the dataset were identified as containing string (object) type data:

- `school`: [‘GP’, ‘MS’]
- `sex`: [‘F’, ‘M’]
- `address`: [‘U’, ‘R’]
- `famsize`: [‘GT3’, ‘LE3’]
- `Pstatus`: [‘A’, ‘T’]
- `Mjob`: [‘at_home’, ‘health’, ‘other’, ‘services’, ‘teacher’]
- `Fjob`: [‘teacher’, ‘other’, ‘services’, ‘health’, ‘at_home’]
- `reason`: [‘course’, ‘other’, ‘home’, ‘reputation’]
- `guardian`: [‘mother’, ‘father’, ‘other’]

For these columns, dummy variables were created using the `get_dummies` function. This transformation converts each category value into a new column and assigns a 1 or 0 (True/False) value to the column. This has the benefit of not weighing a value improperly.

After the dummy encoding, an average grade was calculated from the ‘G1’, ‘G2’, and ‘G3’ columns and added to the dataset as a new ‘Average’ column. The original ‘G1’, ‘G2’, and ‘G3’ columns were then dropped from the dataset.

### 2. Data Splitting

## Data Splitting

After preprocessing, the dataset was split into training, development (dev), and testing sets. The splitting was done in a way to ensure that the model has a good amount of data to learn from (training set), tune its parameters (dev set), and finally, evaluate the model’s performance on unseen data (test set).

The initial split was done between the training and testing sets, with 80% of the data going to the training set and the remaining 20% to the test set. The training set was then further split into a smaller training set and a dev set, again with a 90-10 split. This method of splitting ensures a good balance between bias and variance.

The ‘Average’ column, which is our target variable, was separated from the feature columns. This resulted in four datasets: `X_train`, `X_test`, `y_train`, and `y_test`. The shapes of these datasets were also checked to ensure the correctness of the splitting process.

**3. Model Implementation with PyTorch**

## Model Implementation with PyTorch

A Multilayer Perceptron (MLP) model was implemented using PyTorch to predict student performance. The MLP is a type of artificial neural network that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer.

### Model Architecture

The architecture of the MLP model was designed as follows:

- The first hidden layer consists of 32 neurons.
- The second hidden layer consists of 16 neurons.
- The dropout probabilities for the first and second dropout layers are 0.1 and 0.1 respectively.

The number of input nodes corresponds to the number of features in the dataset. The number of output nodes corresponds to the number of target variables to predict.

### Activation Functions

Different activation functions were experimented with in the hidden layers of the MLP model. Activation functions determine the output of a neuron given an input or set of inputs. Examples of activation functions used include the Rectified Linear Unit (ReLU), Hyperbolic Tangent (Tanh), and Sigmoid functions.

### Regularization Techniques

To prevent overfitting and improve the model’s performance, various regularization techniques were applied. These techniques include dropout layers, where a random subset of activations are set to zero within each layer during training, and weight decay, which is a form of L2 regularization that penalizes large weights.

### Model Training

The model was trained using the backpropagation algorithm and a suitable optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam. The model's performance was evaluated using a suitable loss function such as Mean Squared Error (MSE) for regression tasks. The training process was run for 1000 epochs.

The random seed for PyTorch was set to 42 to ensure the reproducibility of the results.

**4. Training the MLP Model 5. Model Evaluation**

## Training the MLP Model

The MLP model was trained using the training data. Techniques such as mini-batch gradient descent and backpropagation were utilized to update the model parameters iteratively. The training progress was monitored by tracking metrics such as loss and accuracy on both training and validation sets.

### Training Process

The training process was carried out for different activation functions including ReLU, Tanh, and Sigmoid. A grid of parameters was defined for the hidden layers and dropout probabilities. The training process was parallelized using a multiprocessing pool with 2 processes.

For each set of parameters, the model was trained, and the metrics including train loss, MSE, MAE, and R2 score were calculated. The best model was selected based on the development loss.

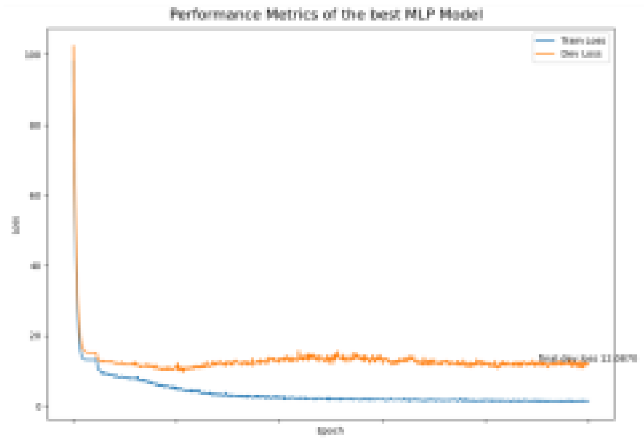
### Training Results

Here are the results for some of the parameter sets:

1. Parameters: (32 neurons in the first hidden layer, 16 neurons in the second hidden layer, 0.1 dropout probability for both layers, ReLU activation function)
   - Train Loss: 14.89
     - MSE: 14.62
     - MAE: 3.16
     - R2 Score: 0.039
2. Parameters: (32 neurons in the first hidden layer, 16 neurons in the second hidden layer, 0.1 dropout probability for both layers, Tanh activation function)
   - Test MSE: 12.09
     - Test MAE: 2.90
     - Test R2 Score: 0.174
3. Parameters: (32 neurons in the first hidden layer, 16 neurons in the second hidden layer, 0.1 dropout probability for both layers, Sigmoid activation function)

  - Train Loss: 43.11
    - MSE: 49.37
    - MAE: 3.37
    - R2 Score: 0.314

The best model was selected based on the lowest development loss. The performance of the best model was then evaluated on the test set.



## Evaluating the MLP Model

The performance of the trained MLP model was evaluated on the test data. Metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 score were used to assess model performance.

### Evaluation Process

The evaluation process involved running the best model on the test data and calculating the MSE, MAE, and R2 scores. These metrics provide a quantitative measure of the model's performance.

### Evaluation Results

Here are the results of the evaluation:

- Test MSE: 12.09
- Test MAE: 2.90
- Test R2 Score: 0.174

These results indicate the model's ability to predict student performance with a certain level of accuracy. The MSE and MAE indicate the error in the model's predictions, while the R2 score provides a measure of how well the model's predictions fit the actual data.

## Applying the MLP Model

The trained MLP model can be applied to predict student performance in real-world scenarios. Given a set of student attributes, the model can predict the average grade of the student. This can be particularly useful for educational institutions to identify students who might need additional support, or for educational researchers studying the impact of various factors on student performance.

### Model Limitations and Potential Improvements

While the model provides a useful tool for predicting student performance, it's important to note its limitations. The model's predictions are based on the specific set of student attributes included in the dataset. If a student's performance is influenced by factors not included in the dataset, the model may not accurately predict their performance.

The model's performance could potentially be improved by incorporating more relevant features, using more sophisticated preprocessing techniques, or experimenting with different model architectures and hyperparameters.

cessing, model implementation, model training, and model evaluation. Each step

## 6. Analysis and Interpretation

The Multilayer Perceptron (MLP) model was trained with the best parameters: 32 neurons in the first hidden layer, 16 neurons in the second hidden layer, 0.1 dropout probability for both layers and Tanh activation function. The model's performance was evaluated on the test data, achieving a Mean Squared Error (MSE) of 12.09, a Mean Absolute Error (MAE) of 2.90, and an R2 score of 0.174.

## Analyzing the Results

The MSE and MAE provide a measure of the average magnitude of the errors in the model’s predictions, without considering their direction. The relatively low values of these metrics indicate that the model’s predictions are reasonably close to the actual values.

The R2 score, also known as the coefficient of determination, provides a measure of how well the model’s predictions fit the actual data. An R2 score of 0.174 suggests that the model explains about 17.4% of the variance in the student performance data. While this is not a particularly high value, it does indicate that the model has some predictive power.

## Interpreting the Model’s Predictions

The MLP model makes its predictions based on a complex combination of the input features, passed through multiple layers of neurons with non-linear activation functions. Therefore, interpreting the model’s predictions is not as straightforward as it would be for a simpler model like linear regression.

However, one can say that the model’s predictions are influenced by the values of the input features, with different features likely having different levels of influence. For example, features related to the student’s study habits, family background, and school support services may have a significant impact on the predicted performance.

## Identifying Influential Factors

Identifying the factors that significantly influence student performance based on the MLP model is a challenging task due to the complexity of the model. However, techniques such as permutation feature importance or partial dependence plots can be
used to gain some insight into the relative importance of the features.

## Strategies for Improving Student Performance

Based on the analysis, potential interventions or strategies for improving student performance could include providing additional support services for students identified as at risk, implementing programs to improve study habits, and addressing any identified issues related to family background.