July 2025 CSE 472: Machine learning Sessional
Assignment 1: Data Preprocessing & Feature Engineering with a Feed Forward Neural Network Pipeline

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

December 2025

Overview

In this assignment, you will work through all major steps of preparing a dataset for machine learning, including cleaning, encoding, scaling and feature selection. You will then use the processed data to train a simple feed-forward neural network in PyTorch and evaluate its performance.

Change Log

Description	Updated By	Timestamp
Assignment declared	NTD & RZS	1 DEC, 2025, 11:30 PM

1 Introduction

Data preprocessing plays a critical role in preparing raw data for meaningful analysis by identifying and correcting errors, removing inconsistencies, and ensuring overall data quality. Feature engineering further enhances the dataset by generating informative variables that enable machine learning models to learn underlying patterns more effectively. Together, these steps significantly improve the accuracy, stability, and generalizability of machine learning systems.

In this assignment, you will explore the essential components of constructing a machine learning pipeline. Specifically, you will learn how to import a dataset into a Python-based notebook environment, preprocess its variables, handle missing values and redundancies, convert non-numerical attributes into numerical representations, apply appropriate normalization techniques, perform correlation analysis with respect to the target variable, and identify the most relevant features for model training. Finally, you will use the processed dataset to fit and evaluate a FNN model.

Please note that you are allowed to use Python library functions.

2 Dataset

In this assignment, you will use the *Medical Student Diabetes* dataset, which can be downloaded from the following link:
https://docs.google.com/spreadsheets/d/1sQgs550vWZPifjOj_HJ-iOWrbXo_MDI8Vgmm0IwJCJI/edit?gid=1996190875#gid=1996190875

2.1 Basic Description of the Dataset

The dataset is designed to facilitate an understanding of various factors associated with diabetes and to support the development of predictive models that identify individuals at risk of having diabetes.
The dataset consists of a total of 13 columns (features and target column) and 200,000 rows (individual records). These features include personal information, physical measurements, basic physiological readings, health-related indicators and a binary target variable indicating diabetes status.

- **Personal Information:** Student ID, Age, Gender
- **Physical Measurements:** Height, Weight, BMI
- **Physiological Readings:** Temperature, Heart Rate, Blood Pressure
- **Health Indicators:** Blood Type, Cholesterol, Smoking
- **Target Variable (Diabetes):** Indicates whether an individual has diabetes.

With the given dataset, you need to perform certain steps which are always crucial in dealing with any machine learning project. So we expect you to understand all these steps and implement them yourselves.

3 Required Tasks

3.1 Understanding the dataset

1. Import the dataset in a notebook environment with python library: "Pandas"
2. Show the number of attributes (columns) and number of records (rows)
3. Show the statistics of the dataset (column wise mean, standard deviation, max, min etc). You can use Matplotlib, to visualize the distribution of data.
4. Count the number of missing values in the dataset
5. Count the number of duplicate values in the dataset.

3.2 Data cleaning

1. If you find any missing values in the dataset (NaN values) replace those data with the column wise mean.
2. If you find any duplicates in the dataset, keep just one copy of the data.
3. Remember, if any row in the target column (Diabetes) is missing, you must drop that row

3.3 Creation of input and output features

You need to split the data into two parts. The "Features" variable will consist of all the columns in the dataset except the target column. And the "Labels" variable will contain only the target column.

3.4 Conversion of features into numeric values

1. You will notice that a number of columns in the dataset contains text (string type) features. For example, the target column also contains labels in the form: "Yes"/ "No". You first need to convert these columns into numeric features.
2. For doing that, you need to first convert such columns which are not numeric types, into categorical types. Then you need to perform one hot encoding on that column, which will divide that column into multiple one hot type column. To better understand this approach, follow this link: get_dummies_in_pandas. (You can use library function here)
3. Remember that, you could have performed label encoding (Label Encoding) instead of one hot encoding. However, giving strict numeric values to some labels might create a bias. For example, if you convert a column 'Blood Type' like this:

   O $\rightarrow$ 0

   A $\rightarrow$ 1

   B $\rightarrow$ 2

   This might create a bias in the model to give the blood group `B` greater value. However, if there are only 2 different values in a column, you can perform label encoding instead of one hot. This will reduce the number of new columns.

3.5 Scaling of the features

1. You will see in the dataset that, the ranges of values in a column varies significantly with the values range from a different column. This will surely hamper the training process of the ML model. To resolve this issue, we apply scaling.

2. There are two types of scaling you can perform: StandardScaling and MinMaxScaling. You need to perform both type of scaling on the dataset and verify which works well (It is expected you do this in a functional way, so whenever you prefer one scaling over another, you can simply change your preference in the argument of this scaling function).

   Standard Scaling (Z-score normalization) transforms features so that they have a mean of 0 and a standard deviation of 1.

   Min-Max Scaling (also known as normalization) scales the data to a fixed range, typically between 0 and 1.

   (You can use library function here)

3. Remember, never scale the target variable. Only scale the feature variables. Please note that, you should avoid scaling the one hot type column that you get in Section 3.4(2). They are already scaled between zero and one.

3.6 Correlation Analysis

1. At this point you have "Features" that contain numeric features and a target column. You now need to perform a correlation analysis on this processed dataset. You need to show the correlation of every column with the target column.

2. You need to show the output result (correlation of each variable with the target column) in the notebook. (You can use library function here)

3. You can also show a correlation matrix for visualization.

4. Select the top 10 columns that have the highest correlation with the target variable. For each of these columns, you can perform a 1D scatter plot to see how these variables help to understand the separation between the different classes of the target variable.

3.7 Validating the pipeline

1. At this stage, you should have a cleaned "Features" dataset containing the selected 10 features, along with the corresponding target column.

2. You may now refer to the GitHub repository to review an example of training a simple feed-forward neural network (FNN) for a classification task using PyTorch. As part of this assignment, you are expected to design and evaluate at least "three to four" different neural network architectures, each containing a minimum of "two to three" hidden layers. It should be runnable on CPU, do not make the architectures too large.

3. Since the dataset is imbalanced, you must perform a `stratified` split to create the training, validation, and test sets using a 70:15:15 ratio. During training, save the model that achieves the lowest validation loss. You may use binary cross entropy loss as the loss function during training and validation. You may use Python libraries for data split.

4. For the final selected model, evaluate its performance on the test set and report the following metrics: accuracy, precision, F1-score and AUROC (with plot). You may use Python libraries for calculating metrics.

3.8 Short Report

Prepare a short report on the architectures you have experimented on. For each architecture, mention the number of layers along with the number of neurons in each layer contains, loss functions, activation functions, etc. Also, include the training and validation loss plot for each model, and justify your choice of the final selected architecture.

4 Marking Criterion

Steps	**Points**
Understanding and Describing the dataset	15
Data cleaning	15
Creation of input and output features	5
Conversion of features into numeric values	10
Scaling of the features	10
Correlation Analysis	10
Inclusion of FNN	20
Short report	15
Interesting data visualization using plots	5 (bonus)

5 Evaluation

1. Submission deadline Sunday 11:59 PM, December 14, 2025.
2. You have to reproduce your experiments during in-lab evaluation. Keep everything ready to minimize delay.
3. You are likely to give online tasks during evaluation which will require you to modify your code.
4. You will be tested on your understanding through viva-voce. No mark will be given on code that you cannot explain.

6 Warning

1. Don't copy! We regularly use copy checkers. Do not copy codes from online resources and LLMs.
2. First time copier and copyee will receive negative marking because of dishonesty. Their default is bigger than those who will not submit.
3. Repeated occurrence will lead to severe departmental action and jeopardize your academic career. We expect fairness and honesty from you. Don't disappoint us!