Assignment 1 - Predict Medical Issues

Administrative details

Deadline: March 17th 2022, 23:59

Scoring: 2 points. After the deadline, there is a 0.25 points penalty per day, for 4 days. If you delay your submission by more than 4 days, the maximum score for the assignment is 1 point. You will personally present the assignment to a teaching assistant in the assignments evaluation weak (weak 9).

Questions: Need help? Use the *#Assignments* Teams channel.

Submission: Upload link: `https://forms.gle/b3uKzb9KH8ZWEoFKA`. The assignment should be a python Notebook that contains the code and presents the results of the experiments.

Assignment

Task According to the World Health Organization stroke is the second leading cause of death globally, responsible for approximately 11% of total deaths. We propose you a dataset to predict whether a patient is likely to get stroke based on the input parameters like gender, age, and various health and social information. And secondly, using the stroke label as feature, we want you learn a model that predicts the BMI (Body Mass Index) column.

What will you learn? The purpose of this assignment is to familiarize you with the training and evaluation process of a basic Neural Network in PyTorch (both for the classification and the regression tasks). It also introduces to you some notions necessary when working with imbalanced datasets (which is very common in practice).

Dataset Description The dataset contains anonymized data for 5110 patients. In Fig. 1 you can see the 11 columns available in the dataset.

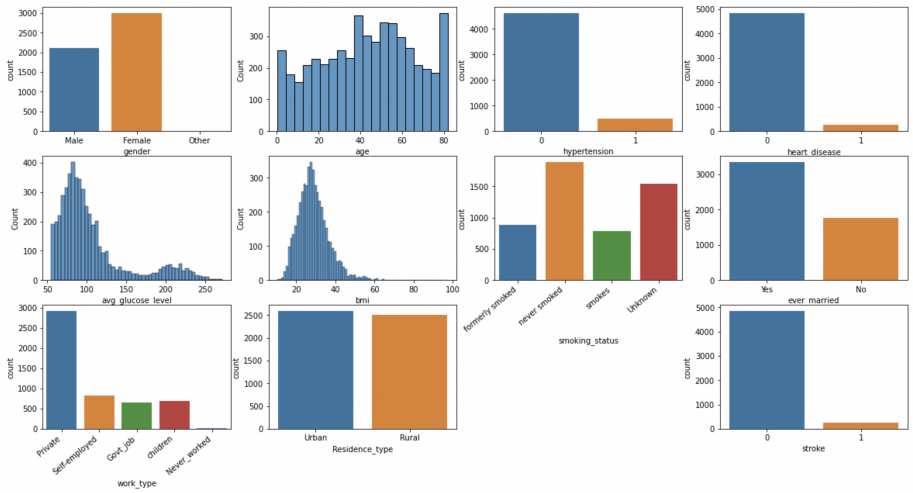

Figure 1: Original Dataset Source: Healthcare Dataset Stroke Data from Kaggle. Please use the dataset from the csv that we provided you, because we already preprocessed the data for you.

TODOs

We will guide you step by step through the necessary steps for solving each subtask, providing you also details on how to verify that things work as expected.

0. Make a new notebook in `https://colab.research.google.com`, mount your google drive and copy the `pmi—data.csv` dataset there. Here you have the detailed steps for it. Load the csv dataset in the notebook using the pandas package and apply `describe` to the pandas dataframe to see its structure.

Classification task - 1.5 points

In this part of your assignment, you should train a neural network to predict for each entry (each patient) the stroke label from the dataset.

2. (0.2 points) Datasets and Dataloaders Start by splitting the dataset between train and validation sets (you can use `train_test_split` from `sklearn.model_selection` package). Make sure you shuffle the data and use 20% of the dataset for validation. Briefly explain why do you think that those choices are considered good practices. Build next a dataset and a dataset loader for both training and validation sets (read more about it here).

3. (0.3 points) Model Build a Neural Net model class that inherits the `nn.Module` and overwrites the `__init__` and `forward` functions. Use several

linear layers followed by non-linearities (e.g.`nn.Linear` and `F.relu`). Make sure that the default `parameters` function works as expected (do not overwrite it!). This function is used by the optimizer to keep track the trainable parameters.

```
# this should print all the parameters from your model
for param in model.parameters():
    print(param)
```

4. (0.2 points) Optimizer and Loss function Initialize an optimizer of your choice (e.g. Adam, SGD, etc.) and a proper loss function (also called criterion) for binary classification (we recommend `BCEWithLogitsLoss`).

5. (0.5 points) Training and Evaluation By now you collected all the elements needed for training your PyTorch neural net model from the previous steps. Iterate through the training Dataloader and update your parameters. Make sure you use the GPU from the colab environment. Keep track of your performance for the training and validation set and plot them at the end of the training (both the loss value and the accuracy). For more details on training a model in PyTorch see Lab. 3 or this tutorial. Observe how the training and validation curves looks and .

6. (0.3 points) Balance the training data In this step we will take a closer look at your predictions. As you saw in Fig. 1, the proposed dataset is highly imbalanced for the stroke column (with way more entries on 0 rather than 1). This might lead our neural network into learning that it is better to always predict 0 all the time (since the average loss will be close to zero). First check this by showing the confusion matrix for your predictions on the validation set. Next, use the `pos_weight` parameter for the `BCEWithLogitsLoss` to say that

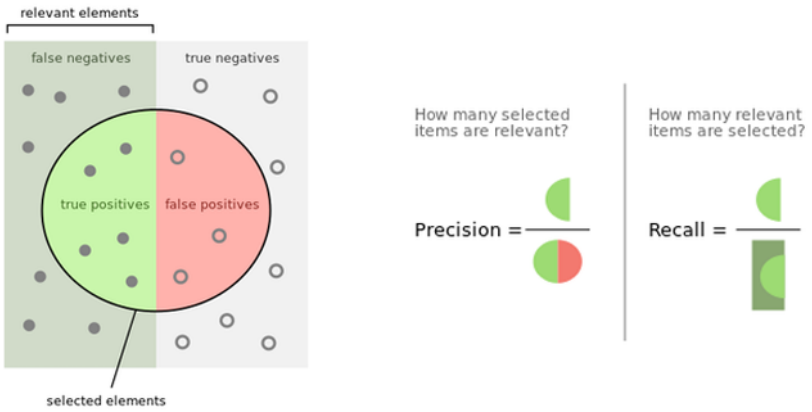

Figure 2: Precision and recall.

your data is imbalanced and by what amount. This will make your positive (1) samples weight more in the loss. We just saw that the accuracy is not a proper metric for the imbalanced case. Enrich your evaluation by computing the precision, recall and the F1 score (e.g.`precision_recall_fscore_support` from `sklearn.metrics` package). Notice that the F1 score is the harmonic mean between the precision and recall, meaning that it is largely affected if either the precision or the recall is small (Fiq. 2).

Regression task - 0.5 points

1. (0.1 points) How is a classification task different from a regression one? Briefly describe the differences from the target data type and training loss point of view.

4. (0.4 points) Redo steps 2-5 (with minor modifications) for predicting the BMI column instead of the stroke. Briefly point out what you changed in the code to adapt it and why.