Report of Machine Learning Final Project

109550027 紀竺均

GitHub_link: https://github.com/chuchunchi/ML-Final-Project
Model_link:
https:/drive.google.com/file/d/1jgrwX8ZtBEcTeN7if1GT_Voe_o7DdTvG/view?usp=share_link

a. Brief Introduction

First, I use some sklearn package to preprocess the data. Then, I use a 1D convolutional neural network implemented by PyTorch for this binary classification task.

b. Methodology

i. Data pre-process

1. Drop ‘id’ column.
2. Use OrdinalEncoder() to turn the attributes (discrete features) into ordinal integers.
3. Use SimpleImputer() to turn the missing values into ‘median’ of that feature. I have tried ‘mean’, ‘most_frequent’ as well but ‘median’ give the highest score.
4. Use sklearn.decomposition.PCA() with n_components=2 to decompose the data from 24 features to 2 features.
5. Use MinMaxScaler() to scale the features. I have tried ‘RobustScaler’ and ‘StandardScaler’ as well and ‘MinMaxScaler’ give the highest score.
6. Finally, use sklearn.train_test_split() to split train and validation data with val ratio=0.1 for training process.

ii. Model architecture

A CNN model implemented by pytorch, with a total of 2 1D_convolution layers, 3 linear layers with relu activate function, and an linear output layer with sigmoid function.
input: a tensor with shape (batch*2 features)
output: a tensor with shape (batch*1), dtype=float between 0, 1

⇨ Train the model for 10 epochs and choose the one with lowest validation loss to be my model.

iii. Hyperparameters
1. loss function: nn.BCELoss()
2. optimizer: torch.optim.Adam
3. learning rate = 5e-4

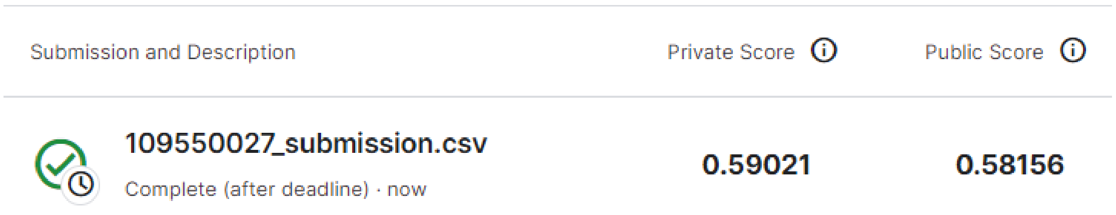
c. Comparison & Result
1. I think the key that I can beat baseline is the "PCA" decomposition method. Before I add it to my preprocess function, the accuracy can't over 0.585.

Compare between decompose or not, 3 scaler function:

	RobustScaler	StandardScaler	MinMaxScaler
PCA(n=2)	0.587	0.589	0.590
Without PCA	0.584	0.546	0.563

⇨ I guess there should be some insignificant features that affect the result. By decomposing and scaling the features using sklearn package we reduce the noise made by them.

2. Result:



Submission and Description	Private Score	Public Score
109550027_submission.csv Complete (after deadline) · now	0.59021	0.58156

d. PEP8
I use pycodestyle_magic to check if I write code under PEP8 guideline in iPython notebook.
The pycodestyle magic command:
```
%load_ext pycodestyle_magic
%pycodestyle_on
```

e. Summary
I construct a convolution neural network to solve this real-world classification problem with score 0.59021. To achieve this, I have preprocessed and decomposed the data into two features. Then, I fed the features into a CNN model which is implemented using PyTorch.

Finally I passed the output into a sigmoid function and thus get an likelihood value between 0, 1.