PSI3471

Avaliação - Parte Prática

Na parte prática da avaliação, você vai trabalhar com um projeto de uma aplicação com redes neurais para a área de saúde. O objetivo é obter um modelo de predição de uma doença baseado em dados de exames clínicos laboratoriais.

Dados disponibilizados

Os dados para treinamento e teste do modelo estão disponíveis no formato CSV e podem ser carregados utilizando a biblioteca Pandas nos `DataFrames` `data_train` e `data_test`, como mostrado a seguir:

```
import pandas as pd

data_train = pd.read_csv(
    "https://media.githubusercontent.com/media/psi3471/datasets/main/disease_prediction/diseas
).drop(columns=["Unnamed: 0"])

data_test = pd.read_csv(
    "https://media.githubusercontent.com/media/psi3471/datasets/main/disease_prediction/diseas
).drop(columns=["Unnamed: 0"])
```

```
data_train.shape
```

```
(614, 9)
```

```
data_test.shape
```

```
(154, 9)
```

```
data_train.head()
```

	0	1	2	3	4	5	6	7	8
0	6.000000	123.0	72.0	45.000000	230.000000	33.6	0.733	34.0	0.0
1	7.000000	159.0	66.0	20.536458	79.799479	30.4	0.383	36.0	1.0
2	3.845052	127.0	80.0	37.000000	210.000000	36.3	0.804	23.0	0.0
3	3.845052	105.0	64.0	41.000000	142.000000	41.5	0.173	22.0	0.0
4	3.000000	111.0	56.0	39.000000	79.799479	30.1	0.557	30.0	0.0

```
data_test.head()
```

	0	1	2	3	4	5	6	7	8
0	4.0	132.0	69.105469	20.536458	79.799479	32.9	0.302	23.0	1.0
1	9.0	145.0	80.000000	46.000000	130.000000	37.9	0.637	40.0	1.0
2	9.0	156.0	86.000000	28.000000	155.000000	34.3	1.189	42.0	1.0
3	4.0	137.0	84.000000	20.536458	79.799479	31.2	0.252	30.0	0.0
4	4.0	171.0	72.000000	20.536458	79.799479	43.6	0.479	26.0	1.0

Como mostrado, os dados consistem de 614 exemplos de treinamento e 154 para teste, cada um contendo 8 características de entrada, representadas pelas colunas de 0 a 7 e a saída desejada binária, indicando se o paciente é portador ou não da doença, representada pela coluna 8.

Em um projeto na área de ciência de dados, é essencial a exploração e tratamento dos dados para filtrar dados inadequados, preencher dados faltantes, aplicar esquemas de normalização ou selecionar colunas mais adequadas com base em análise quantitativa ou alguma informação prévia relativa à interpretação dos dados no contexto em questão. No entanto, neste projeto, **o foco será a exploração do modelo**, dado o conjunto de dados já processado. Por isso, não são fornecidos detalhes sobre o significado de cada uma das colunas da entrada. Se desejar, você pode trabalhar na exploração dos dados para melhorar o desempenho do modelo mas essa atividade não será levada em conta na avaliação.

Objetivo

O objetivo do projeto é obter um modelo baseado em uma rede neural, que receba o vetor com o conjunto de características da entrada e forneça a classificação binária indicando se o paciente é portador ou não da doença.

Para fins de avaliação do desempenho do seu modelo, você deve compará-lo com o resultado obtido com o modelo de regressão logística, que pode ser interpretado como um modelo constituído por único neurônio, alimentado por todas as entradas, com a função de ativação sigmoidal, o que faz com que sua saída esteja no intervalo de 0 a 1.

Projete o modelo, considerando as arquiteturas e as técnicas que foram vistas no curso e realize o treinamento utilizando os dados de `data_train`. Em sequida, teste o modelo fazendo inferência para os dados de `data_test`. Note que esse é um banco de dados desbalanceado e pequeno, o que dificulta o treinamento.

Leve em conta que o treinamento pode sofrer *overfitting* e use os mecanismos vistos no curso para evitá-lo. Para monitorar o *overfitting*, utilize os dados de teste para fazer validação e observe as curvas de aprendizagem para os dados de treinamento e validação. Procure salvar o modelo treinado ao final de cada época, de forma que seja possível carregar o melhor modelo com base na escolha da melhor condição de acordo com as curvas de aprendizagem de treinamento e validação.

Para implementação do modelo e treinamento da rede neural, a sugestão é que seja utilizado um *framework* de redes neurais, como o PyTorch, conforme exemplo mostrado no material de apoio. No entanto, isso não é obrigatório, podendo ser utilizada outra solução ou mesmo outra linguagem de programação, caso desejado.

No final do exercício, você deve apresentar:

1. Sua escolha para a arquitetura do modelo e os valores dos hiperparâmetros;
2. Os códigos utilizados para treinamento e teste;
3. A curva de aprendizado de treinamento e validação ao longo das épocas;
4. Usando os dados de teste, a matriz de confusão, a acurácia e o F1-score obtidos e uma interpretação desses resultados (como referência, utilize o material sobre medidas de desempenho);
5. A comparação dos resultados de 4. com os resultados obtidos utilizando um modelo de regressão logística.

Caso utilize o PyTorch, seguem algumas observações e dicas:

- Caso deseje utilizar a função custo da entropia cruzada, utilize a `BCEWithLogitsLoss` em vez da `BCELoss` pois ela é numericamente mais estável e provavelmente vai proporcionar resultados melhores. Note que a `BCEWithLogitsLoss` espera receber *logits*, que não são normalizados e podem valer de $-\infty$ a ∞. Ou seja, ao usar a `BCEWithLogitsLoss`, a função de ativação do neurônio de saída deve ser linear;
- No PyTorch, para salvar os pesos de um modelo representado pelo objeto `model` no arquivo `model01.pt`, pode-se usar a linha:

```
torch.save(model.state_dict(), "./model01.pt")
```

- E, depois, para carregar os pesos do modelo, pode ser usada a linha:

```
model.load_state_dict(torch.load("./model01.pt"))
```

Instruções para entrega

- O exercício deve ser feito **individualmente**;
- A entrega deve incluir:
  - Um vídeo de no **máximo 1m30s**, mostrando a resolução do exercício;
  - Os **códigos-fontes** dos programas, preferencialmente organizados em um Jupyter Notebook, descrevendo o experimento e mostrando como foram obtidos os resultados solicitados.
- Sobre o vídeo:
  - **Deve incluir áudio** descrevendo o experimento;
  - Gravem a tela do computador usando celular ou usando algum programa de captura de tela (por exemplo Zoom, Google Meet, ou OBS Studio);
  - No início, **deve aparecer o rosto e algum documento do aluno que gravou o vídeo** (como a carteira USP, RG, CNH, etc);
  - Tentem fazer um bom aproveitamento do tempo para apresentar os resultados solicitados, **respeitando o limite de 1m30s e não acelerem a velocidade do vídeo**;
- Sobre os códigos-fonte:
  - **Incluir o nome do aluno** no início do programa;
- Sobre o envio no Moodle:

- Podem ser enviados o arquivo de vídeo (.mkv, .mp4, .avi, etc.) ou um link para o vídeo (Youtube, Google Drive, etc);
  - No segundo caso, certifiquem-se que todos os professores/pesquisadores (magno.silva@usp.br, hae.kim@usp.br, renatocan@lps.usp.br, wesleybeccaro@usp.br) tenham acesso ao seu vídeo.
- Não se esqueçam de escrever o nome do aluno em três lugares diferentes: **no campo "comentários sobre o envio" no Moodle**, **no início do vídeo** e no **início dos códigos-fonte**.