Análise de classificação da doença de Parkinson utilizando rede MLP

1° Adriana de Oliveira Barros
IFNMG-Salinas
Salinas, Brasil
adob@aluno.ifnmg.edu.br

2° Janina Barbosa de Aguilar
IFNMG-Salinas
Salinas, Brasil
jbda3@aluno.ifnmg.edu.br

3° Leidiane Teixeira dos Reis
IFNMG-Salinas
Salinas, Brasil
ltdr@aluno.ifnmg.edu.br

I. Introdução

Esse trabalho consiste em verificar por meio de um algoritmo *Multilayer Perceptron* (MLP) se uma pessoa está ou não com a doença de Parkinson a partir de uma base de dados relacionada com a doença. A doença de *Parkinson* é uma doença degenerativa do sistema nervoso central, crônica e progressiva. É causada por uma diminuição intensa da produção de dopamina (a dopamina ajuda na realização dos movimentos voluntários do corpo de forma automática), que é um neurotransmissor [1].

Dessa forma, esse estudo utilizou uma MLP, que é uma rede neural artificial formada por camadas de entrada, ocultas e de saída, com indeterminada quantidade de neurônios, que são capazes de aprender e aprimorar o desempenho, considerando o MLP de classificação o grande diferencial seria o resultado geral em classes.

II. Materiais e Métodos

Essa seção mostra quais foram os materiais, as técnicas e os procedimentos utilizados, com o intuito de verificar o nível de confiabilidade da classificação da MLP criada para determinar se uma pessoa está com a doença *Parkinson*.

A. Notebooks Colaboratory (mlp-classifier.ipynb)

Através do Google Colaboratory (Colab) um notebook foi desenvolvido na linguagem Python. O modelo de rede MLP implementado nesse ambiente foi para classificação.

Para a realização de treinamento e teste, os dados da base, foram separados em conjuntos, de forma aleatória, sendo 75% (846) dos dados para treinamento e 25% (282) dos dados para teste.

B. Base de Dados

Para realização da classificação proposta neste trabalho, foi utilizada a base de dados *Parkinson's Disease Classification Data Set* disponível em *Kaggle*. Esta base de dados contém 756 amostras, sendo cada uma formada por 755 atributos (754 entradas e 1 saída).

C. Técnicas Utilizadas

1) MinMaxScaler: O MinMaxScaler técnica que consiste no dimensionamento de cada variável no intervalo entre zero e um ou, outra escala que pode ser especificada por meio do argumento *feature_range* [2]. Essa técnica foi utilizada para normalização dos dados, pois a partir da análise da média dos atributos da base de dados, constatou-se que os dados encontravam-se em intervalos muito variados.

2) Smote: Foi realizado o balanceamento através da técnica *Smote* que consiste em gerar dados sintéticos (não duplicados) da classe minoritária a partir de vizinhos [3], [4]. Esse procedimento foi necessário devido a classe 1 apresentar mais de 500 dados enquanto a classe 0 menos de 200 dados.

3) Principal component analysis (PCA): A PCA foi utilizada com o intuito de diminuir a dimensionalidade da base de dados, sendo essa uma técnica estatística capaz de representar dados multivariados em um subespaço de dimensão reduzido. Nessa técnica, é feito a extração dos principais componentes do conjunto de dados composto de muitas variáveis relacionadas entre si, reduzindo assim a sua dimensionalidade e mantendo ao máximo a variação do conjunto de dados [5].

4) Receiver Operating Characteristic (Curva ROC) e Area Under the Curve (Curva AUC): A curva ROC demonstra a performance entre diferentes limites de probabilidades para predição de um modelo de classificação, entre as taxas de verdadeiro positivo e falso positivo [6].A área sob a curva (AUC) pode ser usada como um resumo do desempenho do modelo e contém a taxa de falsos positivos esperada e a taxa de falsos negativos [7].

5) Validação Cruzada: A Validação cruzada é uma técnica para avaliar modelos de *Machine Learning* (ML) por meio de treinamento de vários modelos de ML em subconjuntos de dados de entrada disponíveis e a avaliação deles no subconjunto complementar dos dados [8]. Para isso foi utilizado o método K-fold, que divide os dados de entrada em k subconjuntos. É treinado o modelo de ML em todos, menos em um (k-1) dos conjuntos de dados e, em seguida, o modelo é avaliado no conjunto de dados que não foi usado para treinamento. Esse processo é repetido k vezes, com um subconjunto diferente reservado para avaliação (e excluído do treinamento) a cada vez [8].

6) Escolha do Modelo: Para a seleção do modelo da MLP, foram testadas diferentes quantidades de neurônios, com uma e duas camadas ocultas. Utilizou-se a validação cruzada, método K-fold, para seleção do melhor modelo. Para selecionar a quantidade de neurônios, iniciou-se os testes com a média da quantidade de entradas e saídas, dando o valor de 377

neurônios, em seguida foram feitos testes reduzindo essa quantidades, dividindo a média por dois, sucessivamente. Visto que, quanto menor a quantidade de neurônios e camadas ocultas mais interpretável é o modelo. Após os teste com as diferentes topologias, o melhor modelo, foi o com onze neurônios e uma camada oculta. Este com acurácia do conjunto de validação de 85.94%.

III. Resultados e Discussão

Para a obtenção dos resultados foram aplicadas à base de dados as técnicas descritas na seção II-C. O primeiro passo foi o pré-processamento, no qual, a base de dados foi normalizada utilizando a técnica MinMaxScaler, balanceada utilizando o Smote, e feita uma tentativa de redução de dimensionalidade com a técnica PCA, cujos resultados não foram significativos para uso, sendo assim descartada. O segundo passo foi a seleção do modelo, realizado por meio da validação cruzada. Após a seleção, o melhor modelo (rede com 11 neurônios e 1 camada oculta) foi treinado e testado 30 vezes, com geração aleatória de diferentes conjuntos de treinamento e teste a cada iteração. O resultados obtidos são apresentados a seguir:

- No conjunto de treinamento a acurácia média alcançada pelo modelo foi de: 98.08%, com desvio padrão de: 0.0061432.
- Na análise por classes no conjunto de treinamento fornecida pela matriz de confusão (Fig. 1), percebe-se que 405 diagnósticos da classe 0 foram identificados de maneira correta e 14 foram identificados incorretamente. Já para a classe 1, 417 diagnósticos foram identificados de maneira correta e 10 foram identificados incorretamente.

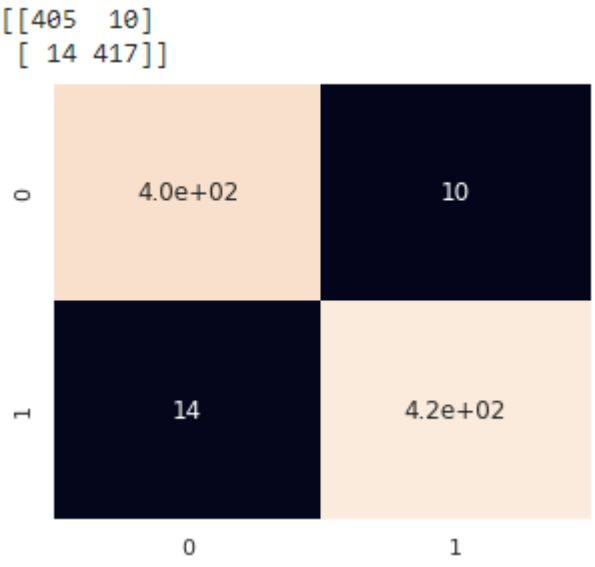

Fig. 1. Matriz de confusão - Treinamento

- No conjunto de teste a acurácia média alcançada pelo modelo foi de: 87.08%, com desvio padrão de: 0.0197017788.3.
- A análise por classes no conjunto de teste obtida pela matriz de confusão (Fig. 2), mostra que 130 diagnósticos da classe 0 foram identificados corretamente e 23 foram identificados incorretamente. Já para a classe 1, 110 diagnósticos foram identificados de maneira correta e 19 foram identificados incorretamente.

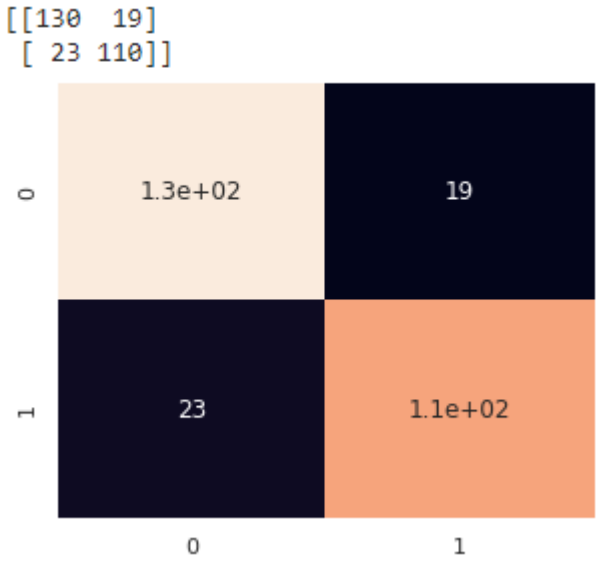

Fig. 2. Matriz de confusão - Teste

A Fig. 3 mostra a representação da curva ROC para o modelo e também apresenta o valor da área sob a curva (AUC), com valor de 0.92.

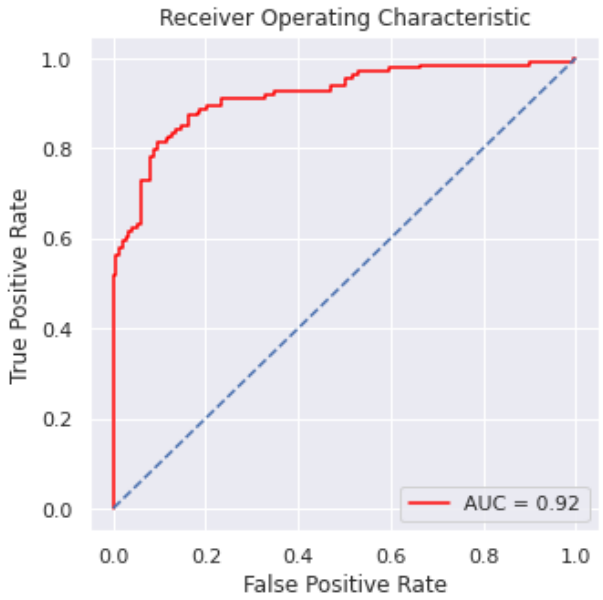

Fig. 3. Curva ROC e AUC

Considerando as técnicas empregadas e o modelo escolhido, os dados retornados apresentam um desempenho promissor,

pois obteve uma acurácia 87.08%, com desvio padrão de: 0.0197017788.3, que corroboram os valores obtidos ao selecionar o modelo. Portanto, é possível concluir que a técnica de MLP foi eficiente na classificação da doença de Parkinson para o conjunto de dados utilizado.

REFERENCES

[1] 2021. [Online]. Available: https://www.einstein.br/doencas-sintomas/parkinson

[2] Scikit-learn, "sklearn.preprocessing.minmaxscaler — scikit-learn 0.24.2 documentation," 2020. [Online]. Available: encurtador.com.br/jvADT

[3] C. Y. Wijaya, "5 smote techniques for oversampling your imbalance data," Sep 2020. [Online]. Available: https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5

[4] R. Santana, "Lidando com classes desbalanceadas – machine learning - minerando dados," Feb 2020. [Online]. Available: https://minerandodados.com.br/lidando-com-classes-desbalanceadas-machine-learning/

[5] M. Lopes, *t-SNE paralelo: Uma técnica paralela para redução de dimensionalidade de dados aplicada em Cidades Inteligentes*. [Online]. Available: encurtador.com.br/zHT12

[6] C. A. Bonfim, "Uma breve descrição e uso da curva roc para classificação em machine learning," 2020. [Online]. Available: https://pt.linkedin.com/pulse/uma-breve-descri

[7] J. Brownlee, "How to use roc curves and precision-recall curves for classification in python," Aug 2018. [Online]. Available: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

[8] Amazon, "Validação cruzada - amazon machine learning," 2021. [Online]. Available: https://docs.aws.amazon.com/pt$_br/machine - learning/latest/dg/cross - validation.html$