Análise de técnicas de aprendizado de máquina em dados de eletroencefalograma com uso de PCA

Mateus Schoffen[1], Diana Francisca Adamatti[1]

[1] Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 96201-900– Rio Grande – RS – Brazil

mateus_schoffen@furg.br, dianaada@gmail.com

Abstract. *This work has an interdisciplinary nature, addressing the areas of Neuroscience and Machine Learning and aims to analyze Machine Learning techniques applied to data collected through electroencephalograms. Data mining techniques were used that are capable of processing large amounts of data, making it easier to obtain the desired information. To perform this work we used an existing database which was preprocessed and the PCA (Principal Component Analysis) technique was applied to the data. The results show that the accuracy does not improve significantly, even running the techniques with 5 or 30 principal components.*

Resumo. *Este trabalho tem um cunho interdisciplinar, abordando as áreas da Neurociência e Aprendizado de Máquina e tem como objetivo analisar técnicas de Aprendizado de Máquina aplicadas em dados coletados através de eletroencefalograma. Foram utilizadas técnicas de mineração de dados que são capazes de fazer o processamento de grandes quantidades de dados, facilitando a obtenção das informações desejadas. Para realizar o trabalho foi utilizado uma base de dados existente ao qual foi executado o pré-processamento e aplicação da técnica de PCA (Principal Analysis Component) nos dados. Os resultados mostram que a acurácia não melhora de forma significativa, mesmo executando as técnicas com 5 ou 30 componentes principais.*

1. Introdução

Os algoritmos de aprendizado de máquina são construídos de forma a aprender e fazer previsões a partir de dados. Ao contrário dos algoritmos de programação estática, que precisam de instrução humana explícita, os algoritmos de aprendizado de máquina realizam uma análise automatizada dos dados e realizam previsões a partir de amostras dos dados [Choudhary 2017].

O campo de aprendizado de máquina atingiu um desenvolvimento impressionante recentemente, com a ajuda do rápido aumento na capacidade de armazenamento de dados e o poder de processamento dos computadores [Baştanlar 2014]. Junto com muitas outras disciplinas, os métodos de aprendizado de máquina têm sido amplamente empregados em neurociências.

Grandes conjuntos de dados estão cada vez mais difundidos nas mais diversas áreas. Para interpretar tais conjuntos de dados, são necessários métodos para reduzir sua dimensionalidade de forma interpretável, de modo que a maioria das informações nos dados seja preservada. A Análise de Componentes Principais (PCA, do inglês *Principal Analysis Component*) é uma das técnicas mais antigas e mais utilizadas para reduzir a dimensionalidade de tais conjuntos de dados, aumentando a interpretabilidade e ao mesmo tempo minimizando a perda de informação característica dos dados [Bro 2014].

A obtenção de dados de sinais cerebrais através de Eletroencefalograma (EEG) pode gerar uma quantidade enorme de dados para a realização de mineração de dados. Para melhorar a precisão e classificação da atividade de um indivíduo baseado em EEG pode-se utilizar técnicas de aprendizado de máquina [Wang 2014].

Nos dias atuais, ainda são necessários estudos na área de mineração de dados que auxiliem na classificação dos dados de EEG para, eventualmente, realizar uma previsão satisfatória sobre o estado do indivíduo. Este trabalho une PCA e algoritmos de aprendizado de máquina para examinar a eficácia de diferentes técnicas supervisionadas e não supervisionadas a fim de contribuir para futuros trabalhos nestas áreas. O estudo de caso utilizado neste trabalho busca definir um estado em que o sujeito esteja realizando uma atividade matemática ou relaxando.

Assim, o trabalho busca entender se com a aplicação da técnica de redução da dimensionalidade da base, que no caso deste trabalho foi o PCA, traz benefícios em relação a acurácia dos algoritmos de aprendizado de máquina supervisionados e não supervisionados utilizados para análise dos dados de EEG do estudo de caso analisado neste trabalho.

2. Embasamento Teórico

Interfaces cérebro-computador (BCI, do inglês *Brain Computer Interface*) são sistemas de comunicação e controle que são usados para traduzir sinais cerebrais em comandos e mensagens, capazes de controlar aplicações como movimentar próteses, digitar letras usando um teclado virtual e ligar ou desligar luzes. Os sistemas BCI desenvolvidos atualmente são ferramentas que podem ajudar os usuários a se comunicar e realizar atividades cotidianas, embora apresentem um sucesso limitado e ainda se encontrem principalmente em ambientes de pesquisa [Silveira 2013].

Nesse sentido, um sistema BCI fornece uma rota alternativa para o envio de comandos para computadores ou outros dispositivos. O Eletroencefalograma (EEG), por exemplo, é uma das técnicas mais utilizadas para entrada de dados nos sistemas BCI, consistindo em um sistema que permite o acesso à atividade cerebral por um "registro de atividade elétrica ao longo do couro cabeludo, produzido pelo anel de neurônios dentro do cérebro" [Niedermeyer and Silva 2005]. Para que ocorra a captura desses sinais EEG, pode utilizar-se de uma touca na cabeça do indivíduo (Figura 1), onde são acoplados os eletrodos, que facilmente fazem o contato e enviam os estímulos cerebrais para a máquina, computador ou prótese. Dados de EEG coletados desta forma precisam ser pré-processados e analisados para a obtenção de alguma informação ou conhecimento.

A Mineração de Dados (MD) é uma parte integral da descoberta de conhecimento em banco de dados (KDD, do inglês *knowledge Discovery in Database*) [Tan et *al*. 2006].

O processo de KDD é o processo de usar Métodos de Mineração de Dados para extrair conhecimento, usando um banco de dados junto com qualquer pré-processamento, sub-amostragem e transformação necessários da base de dados. Cinco etapas são consideradas [Fayyad et *al.* 1996]:

Figura 1. Touca utilizada para coleta dos dados.

- Seleção: este estágio consiste em criar um conjunto de dados de destino ou focar em um subconjunto de variáveis ou amostras de dados;
- Pré-processamento: esta etapa consiste na limpeza e pré-processamento de dados alvo para dados consistentes;
- Transformação: esta etapa consiste em transformar os dados usando métodos de redução ou transformação de dimensionalidade;
- Mineração de Dados: esta etapa consiste na busca de padrões de interesse em uma forma de representação particular, dependendo do objetivo de mineração de dados (então geral, previsão);
- Interpretação/Avaliação: esta etapa consiste na interpretação e avaliação de padrões adquiridos.

A técnica PCA é utilizada para o pré-processamento de dados, assim podendo comparar os resultados entre as técnicas de aprendizado de máquina. Inventada por Karl Pearson em 1901, a análise de componentes principais é um algoritmo para transformar as colunas de um conjunto de dados em um novo conjunto de recursos chamados de Componentes Principais. Ao fazer isso, uma grande parte das informações em todo o conjunto de dados é efetivamente compactada em menos colunas de recursos. Isso permite redução de dimensionalidade e capacidade de visualizar a separação de classes ou clusters, se houver. Os componentes principais apresentam propriedades importantes: cada componente principal é uma combinação linear de todas as variáveis originais, são

independentes entre si e estimados com o propósito de reter, em ordem de estimação, o máximo de informação, em termos da variação total contida nos dados [Hongyu 2016].

Na busca por artigos relacionados encontramos um estudo onde diversas técnicas de aprendizado de máquina combinadas com PCA foram aplicadas em dados de eletroencefalograma de indivíduos com epilepsia para realizar a previsão de informações [Guerreiro et *al* 2021], demonstrando em alguns casos a acurácia das técnicas.

Sendo o trabalho de Guerreiro o mais correlacionado a este artigo, o presente trabalho teve como objetivo demonstrar a acurácia de técnicas de Aprendizado de Máquina aplicadas em dados coletados através de eletroencefalograma e a utilização de PCA em um estudo de caso onde o sujeito está realizando uma atividade de matemática ou está relaxando.

3. Base de Dados Utilizada

A base de dados utilizada neste trabalho foi coletada pelo programa de Mestrado em Ciência da Informação e Dados (MIDS) da Escola de Informação da Universidade da Califórnia em Berkeley, que compartilhou publicamente o conjunto de dados coletados, usando o aparelho de BCI *MindWave* (Figura 2), juntamente com o código do software e o estímulo visual usado para coletar os dados. O conjunto de dados inclui todas as leituras dos sujeitos durante a apresentação do estímulo, bem como leituras antes do início e após o término do estímulo, tudo estando disponível de forma gratuita[1].

Figura 2. Aparelho de coleta de dados EGG Mindwave.

Durante a coleta foram apresentados dois estímulos ligeiramente diferentes para dois grupos diferentes. Os estímulos 1[2] e 2[3] estão disponíveis através do *YouTube*.

Para ambos os estímulos, um grupo de cerca de 15 pessoas visualizou os estímulos ao mesmo tempo, enquanto os dados de EEG estavam sendo coletados. Os estímulos que

[1] https://www.kaggle.com/datasets/berkeley-biosense/synchronized-brainwave-dataset.
[2] https://www.youtube.com/watch?v=zkGoPdpRvaU
[3] https://www.youtube.com/watch?v=sxqlOoBBjvc

cada pessoa visualizou estão disponíveis na base de dados como `subject-metadata.csv`. Além disso, os tempos sincronizados para ambos os estímulos foram salvos no arquivo `estimulus-timing.csv`.

Para cada participante, também foi coletado anonimamente alguns outros metadados, como por exemplo, se eles já viram ou não o vídeo exibido durante o estímulo (um anúncio do *superbowl*), sexo, se viram ou não ícones ocultos exibidos durante o exercício de contagem de cores e a cor escolhida durante o exercício de contagem de cores. Tudo isso pode ser encontrado em `subject-metadata.csv`. Também foi coletado o tempo de visualização (em `indra_time`) de todos os eventos de estímulo, tanto para a sessão 1 quanto para a sessão 2. Esses tempos estão incluídos em `estimulus-times.csv`.

A base de dados contém 105.73 MB de dados separados em 13 colunas e possui 30013 registros, para o estudo foi utilizado apenas os registros da coluna `label` em que continha os valores *relax* ou *math*, que seriam o estado de relaxamento e realizando atividade matemática respectivamente, além de utilizar a coluna `raw_data` que contém os dados dos eletrodos.

4. Metodologia Proposta

Para facilitar o entendimento da metodologia seguida por este trabalho, as atividades foram divididas em 5 etapas como demonstra a Figura 3.

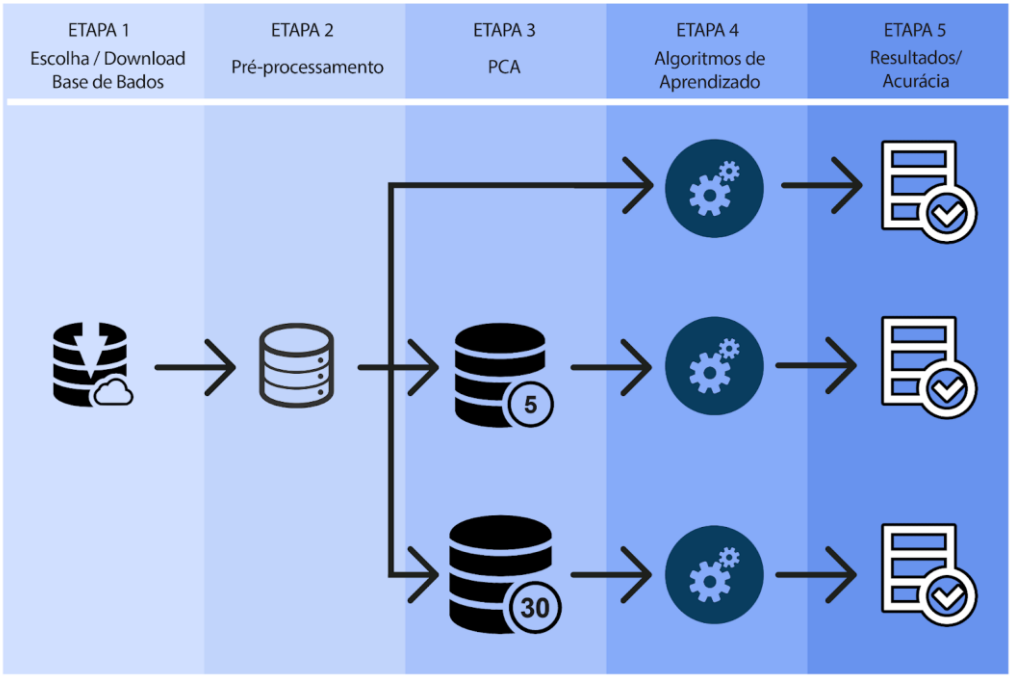
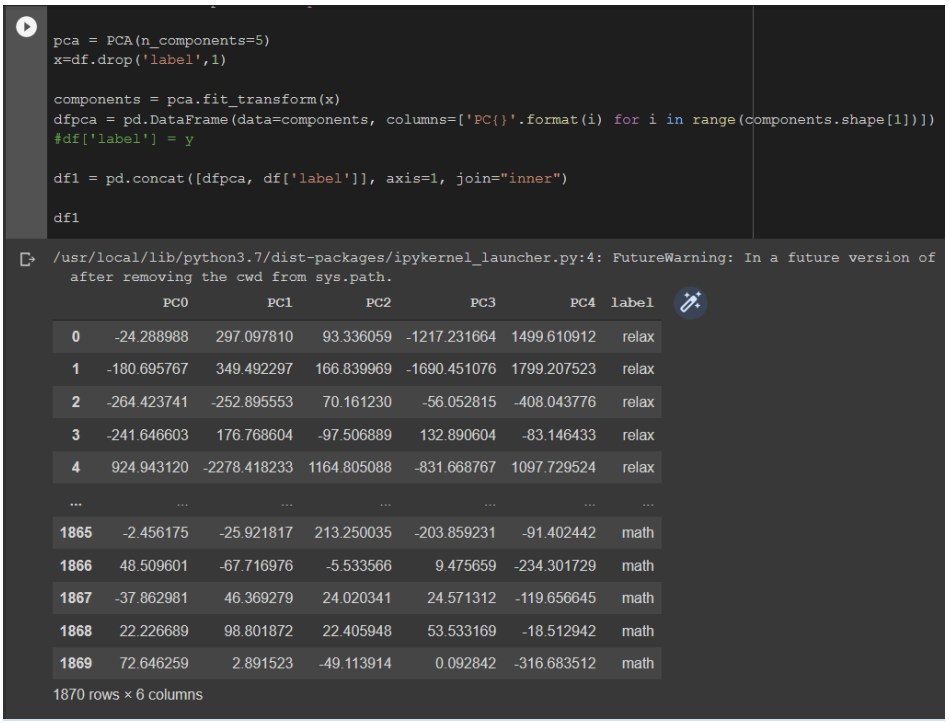


Figura 3. Etapas realizadas neste trabalho.

Na etapa 1 busca-se a base de dados para realizar os estudos, após procurar em sites de base de dados de livre acesso, escolhemos a base de dados disponível

publicamente no site *Kaggle* pois possui informações estruturadas de como foram realizadas as coletas e exemplos já aplicados à base.

Na etapa 2 foi realizado o pré-processamento dos dados, deixando apenas os dados brutos da atividade cerebral e a coluna onde indica qual estado o indivíduo se encontra, podendo este ser um estado de relaxamento ou realizando uma atividade matemática. Na terceira etapa foi aplicado a técnica de PCA na base resultando primeiramente em 5 principais componentes e na segunda vez que foi aplicado configura-se para resultar uma base com 30 principais componentes, com isso diminuindo a dimensionalidade da base de dados de 513 colunas para 6 e para 31, respectivamente, pois a última coluna sempre contém os dados do estado do indivíduo no momento da coleta (`relax, math`), como demonstra a figura 4.



```
pca = PCA(n_components=5)
x=df.drop('label',1)

components = pca.fit_transform(x)
dfpca = pd.DataFrame(data=components, columns=['PC{}'.format(i) for i in range(components.shape[1])])
#df['label'] = y

df1 = pd.concat([dfpca, df['label']], axis=1, join="inner")

df1
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: FutureWarning: In a future version of
  after removing the cwd from sys.path.
```

	PC0	PC1	PC2	PC3	PC4	label
0	-24.288988	297.097810	93.336059	-1217.231664	1499.610912	relax
1	-180.695767	349.492297	166.839969	-1690.451076	1799.207523	relax
2	-264.423741	-252.895553	70.161230	-56.052815	-408.043776	relax
3	-241.646603	176.768604	-97.506889	132.890604	-83.146433	relax
4	924.943120	-2278.418233	1164.805088	-831.668767	1097.729524	relax
...	...	...	...	...	...	...
1865	-2.456175	-25.921817	213.250035	-203.859231	-91.402442	math
1866	48.509601	-67.716976	-5.533566	9.475659	-234.301729	math
1867	-37.862981	46.369279	24.020341	24.571312	-119.656645	math
1868	22.226689	98.801872	22.405948	53.533169	-18.512942	math
1869	72.646259	2.891523	-49.113914	0.092842	-316.683512	math

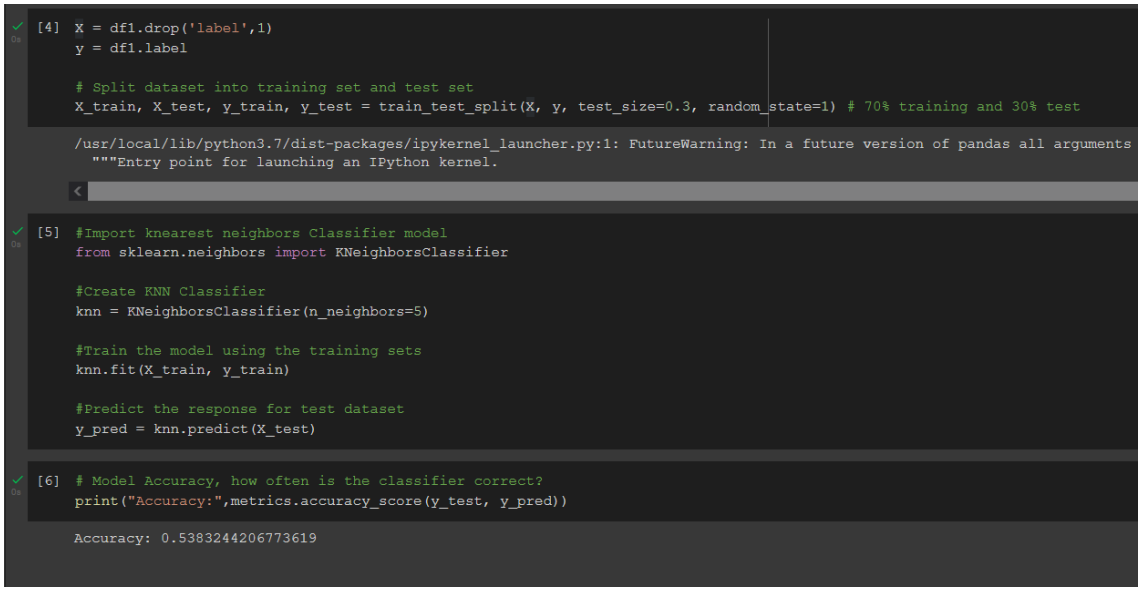
1870 rows × 6 columns

Figura 4: Colunas da base de dados após aplicação de PCA com 5 principais componentes.

Na etapa seguinte, foram aplicadas técnicas de aprendizado de máquina supervisionado e não-supervisionado tanto na base de dados original quanto na base que resultou 5 principais componentes e 30 principais componentes. Utilizando sempre 70 por cento da base como treino e 30 por cento da base como teste, como demonstra a figura 5. Os algoritmos selecionados para a pesquisa foram: KNN (*K - Nearest Neighbors*), SVM (*Support Vector Machine*), *Random Forest*, *Naive Bayes*, *Decision Tree Model* (J48), *Logistic Regression* e MLP (*Multi layer perceptron*), devido ao amplo material encontrado, suporte na internet e em fóruns de programação que descrevem em tutoriais as etapas para aplicação dos mesmos.

Todos algoritmos foram executados em *Python* através do *Google Colab* e da utilização de bibliotecas como *scikit-learn* e *pandas*.

Na última etapa, após a execução de todas as técnicas de aprendizado de máquina, foram coletadas as acurácias dos algoritmos aplicados em uma planilha para a comparação dos resultados.



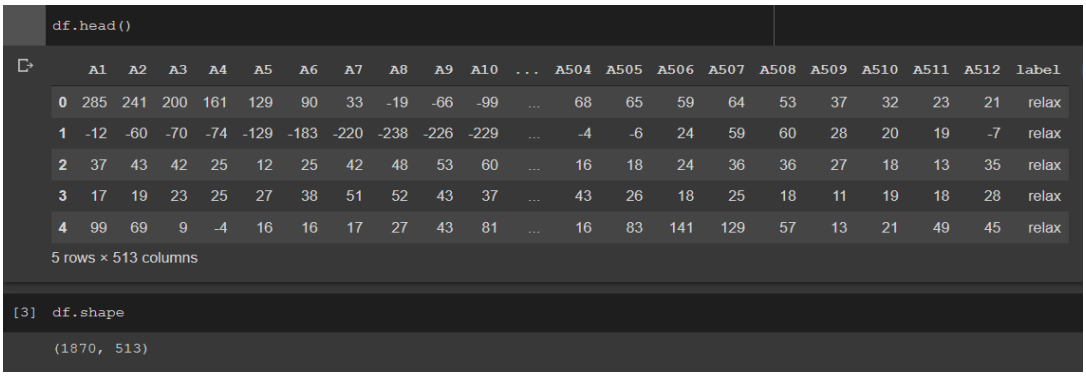
```
[4] X = df1.drop('label',1)
    y = df1.label

    # Split dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: In a future version of pandas all arguments
      """Entry point for launching an IPython kernel.

[5] #Import knearest neighbors Classifier model
    from sklearn.neighbors import KNeighborsClassifier

    #Create KNN Classifier
    knn = KNeighborsClassifier(n_neighbors=5)

    #Train the model using the training sets
    knn.fit(X_train, y_train)

    #Predict the response for test dataset
    y_pred = knn.predict(X_test)

[6] # Model Accuracy, how often is the classifier correct?
    print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

    Accuracy: 0.5383244206773619
```

Figura 5: Aplicação da técnica de aprendizado de máquina (KNN) e acurácia resultante.

5. Experimentos e Resultados

O primeiro passo para a análise dos dados foi a organização da base de dados removendo colunas não necessárias para o estudo. Após isso, a base de dados ficou com 513 colunas, sendo elas 512 colunas com dados numéricos da atividade cerebral e a última coluna a atividade que o indivíduo estava realizando no momento da coleta (para técnicas supervisionadas).

A aplicação das técnicas de aprendizado de máquina primeiramente foi feita na base de dados com os dados brutos, 513 colunas, como demonstra a figura 6.



```
df.head()
```

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	...	A504	A505	A506	A507	A508	A509	A510	A511	A512	label
0	285	241	200	161	129	90	33	-19	-66	-99	...	68	65	59	64	53	37	32	23	21	relax
1	-12	-60	-70	-74	-129	-183	-220	-238	-226	-229	...	-4	-6	24	59	60	28	20	19	-7	relax
2	37	43	42	25	12	25	42	48	53	60	...	16	18	24	36	36	27	18	13	35	relax
3	17	19	23	25	27	38	51	52	43	37	...	43	26	18	25	18	11	19	18	28	relax
4	99	69	9	-4	16	16	17	27	43	81	...	16	83	141	129	57	13	21	49	45	relax

5 rows × 513 columns

```
[3] df.shape

    (1870, 513)
```

Figura 6: Base bruta de dados

Após a aplicação das técnicas de aprendizado de máquina e a obtenção da acurácia de cada experimento, obteve-se as acurácias apresentadas na tabela 1.

Tabela 1. Acurácia das técnicas de aprendizado de máquina

TÉCNICA	BASE	PCA 5 PC	PCA 30 PC
KNN	59.18	53.83	57.04
Random Forest	55.79	59.71	61.85
SVM	49.19	49.01	53.29
Naive Bayes	47.23	54.18	50.98
Decision Tree Model (J48)	54.36	52.40	56.50
Logistic Regression	53.29	55.25	52.94
MLP	55.97	51.69	53.29

Comparando as técnicas de aprendizado supervisionadas com as não supervisionadas, a acurácia diminui nos experimentos utilizando 5 principais componentes em todas as técnicas não supervisionadas, quanto a técnicas supervisionadas tem-se o *Random Forest*, *Naive Bayes* e *Logistic Regression*, que possuem um aumento de, no mínimo, 2%.

Entre as técnicas de aprendizado de máquina não supervisionadas, percebe-se que a acurácia não atinge nem 56% em nenhuma das técnicas utilizadas, sendo a mais alta 55.97% para a técnica MLP, executada na base bruta. Quanto às técnicas supervisionadas, tem-se o *Random Forest* que atinge quase 62% de acurácia, quando executado com 30 principais componentes e o KNN, atingindo 59.19% na base bruta.

A técnica que teve menor oscilação entre a acurácia nos três experimentos foi o SVM. Levando em consideração o experimento aplicando o PCA com 5 principais componentes, o *Random Forest* foi a técnica com maior acurácia seguido por *Logistic Regression*, ambas técnicas de aprendizado de máquina supervisionada.

Contudo, para a base de dados de EGG utilizada neste estudo, e com os algoritmos testados, não se percebeu uma melhora significativa da aplicação da técnica de PCA. Mesmo o trabalho mais similar realizado a este [Guerrero et al. 2021] o qual utiliza uma base com menor quantidade de atributos, não obteve resultados tão promissores quanto o desejado. Desta forma, ainda são necessários estudos para verificar se o PCA é uma técnica que pode auxiliar na descoberta de conhecimento em sinais do tipo EEG.

6. Conclusão e Trabalhos Futuros

A correta compreensão e interpretação de dados de EEG é de suma importância para uso em conjunto com técnicas de aprendizado de máquina para futuros trabalhos que possam

impactar em tratamentos de doenças que atingem o cérebro. O EEG, apesar de ser uma técnica mais econômica e simples para se obter dados sobre a atividade cerebral, continua tendo amplo uso em estudos sobre mineração de dados, devido ao fato de serem dados com uma grande quantidade de informações que podem ser extraídas. Por isso a importância de continuar os estudos procurando melhorar as tecnologias já existentes a fim de encontrar novas aplicações para o mesmo.

Os resultados obtidos a partir deste trabalho poderão ser utilizados para estudos e comparação de novas técnicas que possam surgir a fim de realizar mineração de dados de EGG, possibilitando assim o desenvolvimento de melhores sistemas ou dispositivos para entendermos as atividades cerebrais ou realizarmos classificações.

Embora o PCA em sua forma padrão seja uma ferramenta de análise descritiva de dados amplamente utilizada e adaptável, ele também possui muitas adaptações próprias que o tornam útil para uma ampla variedade de situações e tipos de dados em vários contextos.

Neste trabalho, mesmo não aumentando significativamente a acurácia das técnicas de aprendizado de máquina, o PCA contribuiu para a redução da base original sem a perda de características dos dados e melhorando a velocidade de processamento devido a diminuição da quantidade de dados processados. Relacionado ao desempenho de processamento, não foi realizada uma análise minuciosa, pois a base utilizada não foi muito grande, mas percebeu-se que o processamento das técnicas de aprendizado de máquina entre os dados da base bruta e das bases com 5 ou 30 componentes principais foi realizada em tempos diferentes.

Para trabalhos futuros pretende-se utilizar outras bases de dados de EEG para futuras comparações e também aplicação de técnicas de balanceamento de dados e *deep learning* na base de dados atual. Trabalhos futuros podem explorar outros métodos de mineração de dados para obtenção de informações sobre a atividade cerebral e também sobre como o PCA pode ser utilizado em outros tipos de base de dados. Além disso, pode-se testar diferentes quantidades de exemplos de referências para testar a sensibilidade dos resultados da aplicação.

Este trabalho ajudou a demonstrar que para dados de EEG são necessárias mais pesquisas para compreender a sua complexidade e descobrir a melhor técnica para previsão do estado de um indivíduo levando em consideração apenas os dados obtidos da atividade cerebral.

Referências

Başţanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. *miRNomics: MicroRNA biology and computational analysis*, 105-128.

Bro, R., & Smilde, A. K. (2014). Principal component analysis. *Analytical methods*, *6*(9), 2812-2831.

Choudhary, R., & Gianey, H. K. (2017, December). Comprehensive review on supervised machine learning algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)* (pp. 37-43). IEEE

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37-37.

Guerrero, M. C., Parada, J. S., & Espitia, H. E. (2021). Principal Components Analysis of EEG Signals for Epileptic Patient Identification. *Computation*, *9*(12), 133.

Hongyu, K., Sandanielo, V. L. M., & de Oliveira Junior, G. J. (2016). Análise de componentes principais: resumo teórico, aplicação e interpretação. *E&S Engineering and science*, *5*(1), 83-90.

Niedermeyer, E., & da Silva, F. L. (Eds.). (2005). *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Silveira, J. D. Á. (2013). *Análise de sinais cerebrais utilizando árvores de decisão* (Master's thesis).

Tan, P. N., Steinbach, M., & Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*, *487*, 533.

Wang, X. W., Nie, D., & Lu, B. L. (2014). Emotional state classification from EEG data using machine learning approach. *Neurocomputing*, *129*, 94-106.