Redução da dimensionalidade em bigdata

Lucca Machado, Anderson Borba

Curso de Sistemas e Computação
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brasil
32292783@mackenzista.com.br, anderson.borba@mackenzie.br

Abstract. *Due to the technological revolution, more data is being generated exponentially on a daily basis, causing processing and storage problems. For this reason, this TCC project seeks, through PCA, Kernel PCA and LDA algorithms, to reduce the dimensionality of the data to solve this problem. The objective is to compare their dimensionality reduction capabilities in structured databases and carry out a practical analysis of processing time and, using RandomForest, report the accuracy rate of data prediction. The theoretical-methodological approach involves literary review, mathematical analysis and study of applications. Also with the aim of comparing dimensionality reduction capabilities, we will use PCA on an unstructured database consisting of 200 images. Comparisons were carried out analyzing the ability to preserve quality, processing time and storage savings. Results were shown that allowed the techniques to be compared in terms of information preservation and processing time. Experiments with datasets and evaluation metrics were carried out. The results provided valuable insights for practical application in multivariate data analysis.*

Resumo. *Devido a revolução tecnológica, mais dados estão sendo gerados no dia a dia de forma exponencial, causando problemas de processamento e de armazenamento. Por este motivo, este projeto de TCC busca por meio dos algoritmos PCA, Kernel PCA e LDA reduzir a dimensionalidade dos dados solucionar este problema. O objetivo é comparar suas capacidades de redução de dimensionalidade em bases de dados estruturada e fazer uma análise na prática do tempo de processamento e, utilizando o RandomForest, relatar a taxa de acerto da previsão dos dados. A abordagem teórico-metodológica envolve revisão literária, análise matemática e estudo de aplicações. Também com o objetivo de comparar a capacidades de redução de dimensionalidade, vamos utilizar o PCA em uma base de dados não estruturada composta por 200 imagens. As comparações foram realizadas analisando a capacidade de preservação da qualidade, tempo de processamento e economia de armazenamento. Foram mostrados resultados que permitiram comparar as técnicas em termos de preservação de informações e tempo de processamento. Experimentos com conjuntos de dados e métricas de avaliação foram realizados. Os resultados forneceram percepções valiosos para aplicação prática em análise de dados multivariados.*

Palavras-chave: Redução da dimensionalidade; PCA; Kernel PCA; LDA.

1. Introdução

Com a revolução tecnológica, permitiu o aumento constante da quantidade de dados gerados em diversas áreas, a análise desses dados tornou-se um desafio cada vez maior, pois com mais dados gerados mais espaço de armazenamento será necessário e mais processamento será preciso para fazer a análise, tornando o tempo de processamento alto, impactando entregas de projetos.

A Figura 1 contém um gráfico que pode comprovar esse fato, segundo a International Data Corporation (IDC) ocorreu um crescimento exponencial dos dados nos últimos anos e a empresa prevê que no ano de 2025 pode chegar a 175 Zettabytes de dados armazenados mundialmente.

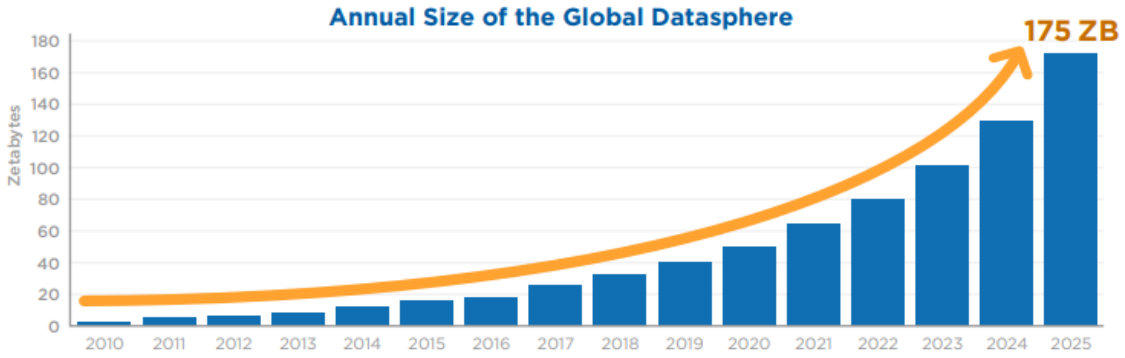

Figura 1. Crescimento exponencial de dados armazenado mundialmente de 2010 a 2025.

Fonte: IDC - The Digitization of the World (2018)

Torna-se o objetivo do nosso estudo, entender como a redução da dimensionalidade em bigdata lida com desafios de armazenamento e de processamento, mostrando na prática resultados de aumento de performance de processamento e a diminuição da armazenagem de dados.

O objetivos específicos deste trabalho é explorar as técnicas de redução de dimensionalidade em big data, destacando suas vantagens e desvantagens e mostrar, na prática, a diferença no tempo de processamento e na taxa de acerto nas previsões, comparando entre uma mesma base de dados com a dimensionalidade reduzida e a outra normal. Assim, podendo provar como elas podem ser aplicadas para lidar com grandes conjuntos de dados. Também houve uma análise feita em uma base de dados de imagens utilizando o PCA, com o objetivo de fazer a comparação de uma base sem a redução e outra base com a redução, medindo o tempo de impressão de imagens e a economia de armazenamento.

No decorrer do artigos, comentamos o referencial teórico, diferenças e vantagens dos algoritmos de redução PCA, Kernel PCA e LDA. Em seguida descrevemos os

materiais e métodos utilizados neste trabalho. Fizemos uma comparação com os resultados obtidos. E por fim, fizemos nossas considerações finais.

2. Referencial Teórico

A redução da dimensionalidade é extremamente proveitosa quando lidamos com conjuntos de dados que possuem muitos atributos. Caso a dimensionalidade não seja reduzida, pode resultar em tempos de processamento elevados. Assim, podemos usar métodos para diminuição dos atributos, fazendo o processamento se tornar mais rápido. A baixa dimensionalidade também serve para reduzir redundância nos dados. A seguir, serão descritos os algoritmos selecionados para diminuir a dimensionalidade.

2.1. Principal Component Analysis (PCA)

O PCA é uma técnica estatística amplamente utilizada para redução de dimensionalidade e extração de informações relevantes de conjuntos de dados multivariados. O PCA foi originalmente introduzido por Pearson (1901) e posteriormente desenvolvido por Hotelling (1933). O PCA tem aplicações em vários campos, como reconhecimento de padrões, processamento de sinais e análise de dados (H. Abdi, L.J. Williams, 2010).

O PCA é baseado na matriz de covariância dos dados originais e mede as relações lineares entre os atributos. Esta matriz é diagonalizada para obter os autovetores e autovalores associados. Os autovetores representam as direções dos eixos principais do espaço transformado, e os autovalores indicam a importância relativa de cada componente principal. O principal objetivo do PCA é encontrar uma transformação linear que projete os dados de entrada em um novo espaço de baixa dimensão chamado espaço de componentes principais. Essa transformação é realizada de forma que o primeiro componente principal obtenha a variação máxima possível nos dados e números componentes principais obtenha a variação restante máxima não considerada pelo primeiro componente. Dessa forma, os componentes principais são ordenados de forma decrescente de acordo com sua importância para a variância explicada pelos dados originais (H. Abdi, L.J. Williams, 2010).

Além da redução de dimensionalidade, o PCA também pode ser usado para visualização de dados, detecção de outliers, pré-processamento de dados, remoção de ruído e fusão de informações. A interpretação dos componentes principais fornece informações sobre as relações entre as variáveis originais e ajuda a identificar os principais fatores que influenciam o conjunto de dados. Existem muitas variações do PCA, incluindo o PCA incremental, que pode lidar com grandes quantidades de dados, e o PCA probabilístico, que lida com a incerteza nos dados. Além disso, o PCA pode ser estendido para análise de componentes independentes (ICA) para capturar relacionamentos não lineares entre variáveis (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

Em suma, a análise de componentes principais é uma técnica poderosa para investigar a estrutura subjacente de conjuntos de dados multivariados. É amplamente utilizado em muitos campos de pesquisa e aplicações práticas porque pode reduzir a dimensionalidade enquanto preserva informações importantes.

2.1. LDA

A análise discriminante linear (LDA) é um algoritmo de aprendizado de máquina amplamente utilizada para redução de dimensionalidade e classificação de dados. Introduzida por Fisher em 1936 e é aplicado em várias áreas, como análise de dados, processamento de imagens, biometria e reconhecimento de padrões.

O principal objetivo do LDA é encontrar uma transformação linear que projete os dados de entrada em um novo espaço de baixa dimensionalidade enquanto maximiza a separabilidade entre diferentes classes. Ao contrário da análise de componentes principais (PCA), que visa maximizar a variância total dos dados, a LDA visa maximizar a variância entre as classes e minimizar a variância dentro da classe (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

O LDA determina a melhor projeção de dados com base nas estatísticas de classe. Essas estatísticas incluem a matriz de dispersão média, a matriz de dispersão dentro da classe e a matriz de dispersão entre classes. As projeções LDA são obtidas calculando os autovetores e autovalores associados a uma combinação linear dessas matrizes de dispersão entre classes e dentro das classes. Essa combinação linear é a chave para encontrar as direções no espaço de características que melhor discriminam entre as diferentes classes (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

Um dos principais usos do LDA é a classificação de padrões. Como os dados são projetados no espaço LDA, classificadores simples, como regressão logística, podem ser aplicados para realizar tarefas de classificação. O LDA também pode ser usado como uma etapa de pré-processamento para melhorar o desempenho de outros algoritmos de classificação. O LDA é especialmente útil em cenários em que o número de amostras é pequeno (alta dimensionalidade) em relação ao número de recursos, pois quando contém relativamente poucos exemplos para aprender padrões, esses exemplos são descritos por muitas características, o LDA pode ser uma escolha eficaz para extrair informações discriminantes relevantes e reduzir a dimensionalidade do problema.

Nesses casos, o LDA pode efetivamente reduzir a dimensionalidade dos dados, preservando as informações de identificação mais relevantes (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

Vale a pena notar que o LDA assume que os dados são geralmente multivariados e que as matrizes de covariância de classe são iguais, a igualdade das matrizes de covariância de classe permite uma solução mais eficiente. No entanto, existem variações de LDA que relaxam essas suposições, como LDA regulado e LDA não paramétrico (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

Concluindo-se, a análise discriminante linear (LDA) é uma técnica poderosa para redução de dimensionalidade e classificação de dados. Sua capacidade de maximizar a separabilidade entre as classes o torna valioso em muitas áreas de pesquisa e aplicação. O uso do LDA pode melhorar o desempenho dos algoritmos de classificação e facilitar a interpretação dos dados.

2.3. Kernel PCA

Kernel Principal Component Analysis (Kernel PCA) é uma extensão da Principal Component Analysis (PCA) que permite processar dados não lineares e capturar autoestruturas mais complexas em seu conjunto de dados. A abordagem kernel PCA foi originalmente proposta por Schölkopf, Smola e Müller (1998) e tem sido aplicada em vários campos, como reconhecimento de padrões, processamento de imagens, bioinformática e análise de dados de ressonância magnética funcional (fMRI) (G. Baudat, F. Anouar, 2001).

O principal objetivo dos kernels do PCA é mapear os dados originais em um espaço de alta dimensão chamado espaço de recursos que pode capturar estruturas não lineares. A partir daí, o PCA é aplicado ao espaço de recursos para extrair componentes principais. A principal vantagem dos kernels PCA é sua capacidade de processar dados que não são linearmente separáveis no espaço original. Os kernels do PCA permitem uma extração de informações mais sofisticada, mapeando os dados em um espaço de recursos de alta dimensão onde a separabilidade pode ser alcançada. Isso é de particular relevância para tarefas de diagnóstico, como a detecção de transtorno de déficit de atenção e hiperatividade (TDAH) usando dados de fMRI (functional Magnetic Resonance Imaging) (SIDHU; ASGARIAN; GREINER; BROWN, 2012).

Além disso, os kernels do PCA podem ser combinados com métodos de classificação como Support Vector Machines (SVM) para executar tarefas de diagnóstico. Os padrões extraídos pelo kernel do PCA são usados como recursos para treinar um classificador capaz de distinguir entre indivíduos com TDAH e indivíduos saudáveis (SIDHU; ASGARIAN; GREINER; BROWN, 2012).

No entanto, é importante mencionar que os kernels do PCA podem exigir alto poder computacional devido ao aumento da dimensionalidade dos dados. Além disso, a escolha correta dos kernels e seus parâmetros é fundamental para a obtenção de resultados satisfatórios.

2.4. Diferenças entre os algoritmos PCA, LDA e Kernel PCA

O processo do PCA a extração de características, o algoritmo identifica a correlação entre atributos, e caso haja uma forte correlação o algoritmo os combina, assim reduzindo a quantidade de atributos na base de dados. O PCA contém o parâmetro n_components que especifica o número de componentes principais a serem retidos após a transformação. Escolher um valor adequado para o número de componentes permite reduzir a dimensionalidade dos dados, preservando ao mesmo tempo a maior parte da informação contida nos dados originais. A quantidade de componentes nunca deve ser maior que a quantidades de variáveis independentes (H. Abdi, L.J. Williams, 2010).

Já o LDA usa a técnica de seleção de características, o algoritmo seleciona quais atributos mais importante dentro de uma determinada base de dados e apaga os que não tem muita relevância. Além de encontrar os componentes principais, O LDA contém o parâmetro número de componentes que especifica o número de discriminantes lineares a

serem retidos após a transformação. Esse número não pode exceder o número de classes menos um (S. Raschka, 2014).

É importante destacar que o PCA é um dos principais algoritmos de aprendizagem não supervisionada, ou seja, não depende da classe para executar os processos. Diferente do LDA, que é um algoritmo de aprendizado de máquina supervisionado, usado para classificação de dados em duas ou mais classes, ou seja, ele é treinado com um conjunto de dados rotulados para encontrar uma combinação linear de características que maximiza a separação entre as classes (Hastie, T., Tibshirani, R., & Friedman, J. 2009).

Na Figura 2 podemos notar a diferenças na utilização dos autovetores, o PCA busca encontrar os componentes principais que maximizam a variação nos dados, isso é feito calculando os autovetores na matriz de covariância dos dados, no gráfico os autovetores estão direcionados onde tem maior variação. Já o LDA, busca encontrar componentes principais que maximizam a separação das classes, isso é feito calculando os autovetores da matriz de covariância entre as classes, no gráfico os autovetores estão direcionados entre a maior diferenciação entre classe.

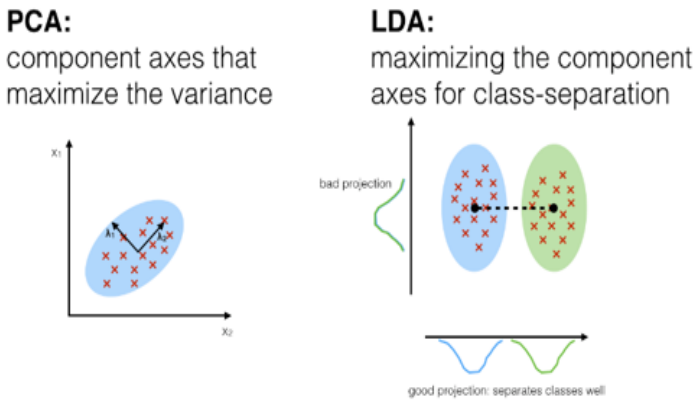

Figura 2. Diferença entre PCA e LDA

Fonte: Raschka (2014)

O diferencial do Kernel PCA está em relação de estruturas não lineares, sendo possível mapear os dados para um espaço de maior dimensionalidade, assim separando os dados onde não era possível separar utilizando algoritmos de estrutura linear, figura 3 (ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. 2021).

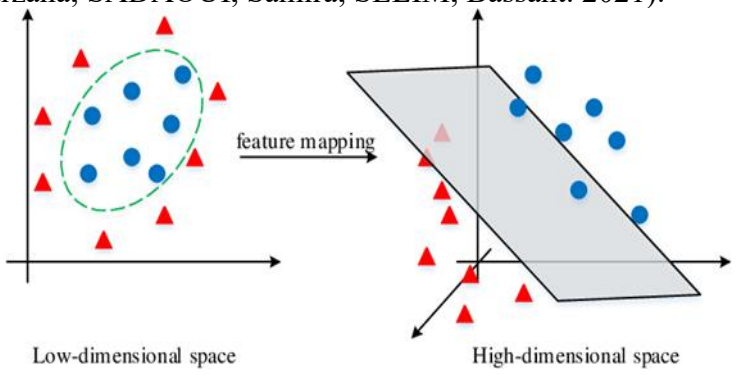

Figura 3. Exemplo de aumento da dimensão do espaço usando o Kernel PCA

Fonte: Fonte: Raschka (2014)

Portanto, o PCA é uma técnica de redução de dimensionalidade que não leva em consideração informações sobre as classes dos dados. O LDA é uma técnica supervisionada que busca encontrar projeções lineares que maximizem a separação entre as classes. O Kernel PCA é uma extensão do PCA que permite lidar com dados não lineares, mas ainda é uma técnica não supervisionada. A escolha entre esses algoritmos depende da natureza dos dados, da presença de não linearidades e do objetivo específico da análise, se é focado na variabilidade global dos dados (PCA), na separação entre classes (LDA) ou em estruturas não lineares (Kernel PCA).

2.5. Vantagem e desvantagem da redução da dimensionalidade em big data

Embora a redução de dimensionalidade ofereça muitas vantagens para análise de big data, ela também apresenta algumas desvantagens que devem ser consideradas. Como vantagem, pode gerar eficiência computacional, com menos atributos presente na base as operações de análise de dados podem ser executadas mais rapidamente. Podem melhorar a visualização devido a revelação de padrões e estruturas ocultas, projetando em um gráfico para melhor interpretação. Com a redução de atributos que a redução da dimensionalidade proporciona, podem eliminar ou mitigar informações irrelevantes ou "ruidosas", deixando a base de dados mais simplificada ocasionando maior performance (NAKRA; DUHAN, 2020).

Já como desvantagem, a redução da dimensionalidade pode ocasionar perda indesejada de informações importantes presentes nos dados originais, é importante considerar cuidadosamente o equilíbrio entre redução de dimensionalidade e preservação de informações relevantes. Escolher o método mais adequado para um determinado grande conjunto de dados pode ser uma tarefa difícil, pois cada método tem diferentes suposições e limitações. A seleção inadequada desses parâmetros pode levar a resultados abaixo do ideal. Além disso, a otimização de parâmetros pode ser computacionalmente intensiva para grandes conjuntos de dados. Resultados podem ser difíceis de interpretar em espaços dimensionais menores. As relações entre os atributos originais podem não ser fáceis de interpretar no novo espaço, dificultando a explicação dos resultados (NAKRA; DUHAN, 2020).

3. Materiais e Métodos

3.1. Hardware utilizado

O Hardware onde ocorreram o testes possui um processador de 3.80 GHz, memória RAM de 32GB de capacidade de memória, um SSD de 512GB de armazenamento com Leitura 3100MBs e Gravação 1500MBs e uma placa de vídeo RTX 3070 Ti de 8GB.

3.2. Software utilizado

O sistema operacional onde ocorreram as análises foi o Windows 11 Pro versão 22H2, como ambiente de desenvolvimento integrado foi utilizado o Jupyter Notebook 6.5.2 com a linguagem de programação Python 3.10.2.

3.3. Bases de dados utilizadas

A base de dados estruturada de classificação escolhida foi a Adult Data Set, base de dados que prevê se a renda excede US$ 50.000/ano com base nos dados do censo. Também conhecido como conjunto de dados "Renda do Censo". A base consiste em 50 mil registros com 14 atributos que definirá um atributo de classe binária.

Destes 14 atributos, 6 são atributos numéricos: age, final-weight, education-num, capital-gain, capital-loos, hour-per-week. E 8 são atributos categórico: workclass, education, marital-status, occupation, relationship, race, sex, native-country.

Fonte: Becker,Barry and Kohavi,Ronny. (1996). Adult. UCI Machine Learning Repository.

A base de dados estruturada de regressão escolhida foi a House prices, base de dados que prevê o preço de uma casa de acordo com atributos como quantidades de quartos, banheiros, andares, metros quadrados etc. A base consiste em 21.612 registros com 20 atributos que definirá um atributo de classe numérica.

Destes 20 atributos, todos são atributos numéricos: id, date, bedrooms, bathrooms, sqtt_living, sqtt_lot, floors, waterfront, view, condition, grade, sqtt_above, sqtt_basement, yr_built, yr_renovated, zipcode, lat, long, sqtt_living15, sqtt_lot15.

Fonte: Anna Montoya, DataCanary. (2016). House Prices - Advanced Regression Techniques. Kaggle.

A base de dados não estruturada escolhida foi Train2014, base de dados com imagens aleatório de alta qualidade. A base consiste 25 mil imagens com o tamanho de armazenamento 3,80 GB.

Fonte: Tsung-Yi Lin, Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., … Zitnick, C. L. (2014).

3.4. Métodos utilizados para obtenção dos resultados.

Para a base de dadas estruturados, primeiramente foi lida a base usando a biblioteca pandas para trabalhar com dataframe, após foi iniciado o tratamento de dados, no caso da classificação foi utilizado o LabelEncoder, permitindo transformar variáveis categóricas em variáveis numéricas atribuindo um número inteiro único a cada categoria, após o StandardScaler, utilizada para realizar a padronização dos recursos (features) em

um conjunto de dados. Assim transformando as variáveis categóricas em numéricas e padronizadas os recursos numéricos do data set, ajudando garantir que diferentes recursos tenham a mesma escala, e para finalizar o tratamento da base de dados, foi utilizado o train test split para separar em base de treino (85%) e base de teste (15%), escolhido empiricamente.

Com os dados separados foi aplicado o algoritmos de redução da dimensionalidade nas bases de treinamento e nas bases de teste, após foi utilizado o Random Forest, pois é um algoritmo que não precisa de grandes tratamentos de dados e traz um resultado de precisão satisfatória, para verificar o accuracy e fazer a comparação se teve perda de qualidade na redução da dimensionalidade. Já a comparação por tempo foi utilizada a biblioteca Time medindo o tempo de processamento.

A rotina de código no caso da base de classificação, para o resultado ficar mais assertivo, vão ser rodado 50 vezes a mesma rotina de código, pegando o tempo total de processamento de cada algoritmo de redução de dimensionalidade. A rotina basicamente vai ser usada o método predict do RandomForest para fazer a previsão e a biblioteca accuracy_score para calcular a taxa de acerto. Com os testes realizados pelo PCA e o kernel PCA, ocorreram testes com a variável n_components diferentes para analisar o impacto delas no tempo de processamento e nas previsões. No caso do LDA, por utilizar uma base de dados de classificação binaria, será possível apenas fazer o teste com o n_components = 1.

Para a rotina de código no caso da base de regressão, nos testes foi percebido que o tempo de processamento apresentou diferenças entre as execuções (porém as bases com as dimensões reduzidas sempre com um tempo de processamento menor do que a base sem a redução na dimensionalidade), para trazer um resultado mais assertivo, decidimos aumentar o número de testes de 50 para 100000, e conseguimos resultados mais satisfatório. A rotina basicamente foi usada o método predict e score do RandomForest para fazer a previsão e calcular a taxa de acerto, e usamos a biblioteca mean_absolute_error, para haver outra métrica de avaliação da base de dados, e ter precisão maior na comparação dos resultados por se tratar de uma classe numérica.

Na figura 4, podemos visualizar o fluxograma processo de obtenção de resultado para a base de classificação e regressão.

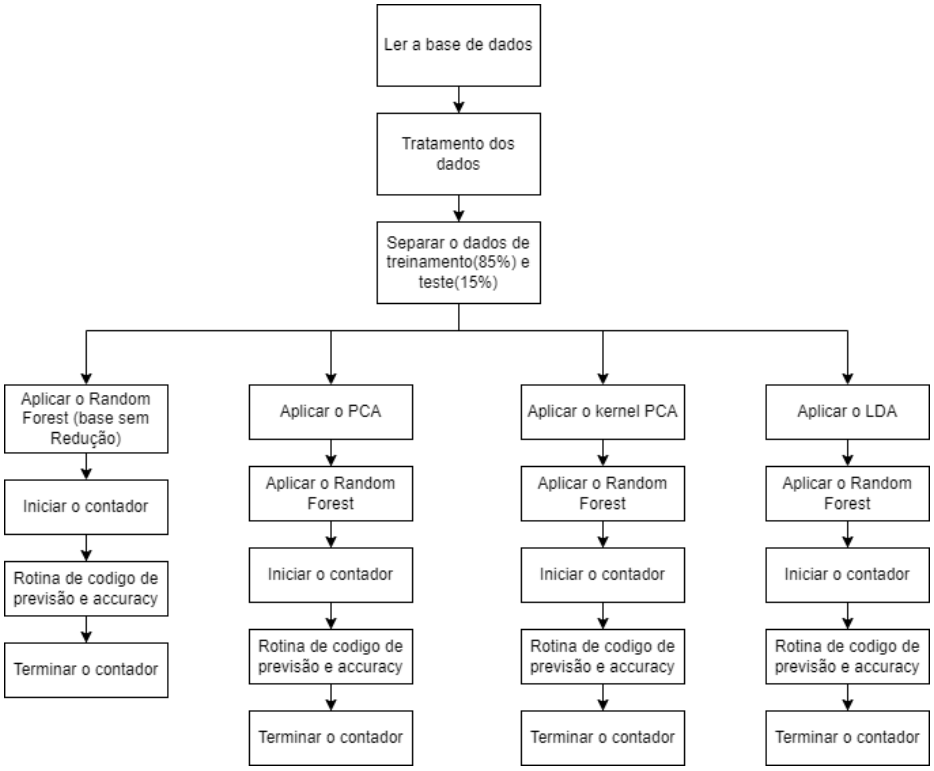

Figura 4. Fluxograma do método utilizado na base de dados estruturado.

Fonte: Elaborado pelo autor (2023)

Para a base de dados não estruturada, foi iniciada alocando em uma variável o caminho do diretório, em uma estrutura de repetição foi percorrida a base inteira até não existir imagens, foi utilizado a biblioteca OS, que ajuda a interagir com o sistema operacional, para esse processo.

Após, foram divididos os canais de cor da imagem com a biblioteca cv2 (utilizada para processamento de imagem e visão computacional) e foi aplicado o PCA com os parâmetros n_componets = 100, 50, 20 e 5 em cada uma das cores. Com as imagens com o PCA aplicado, foi utilizado o método inverse_transform (é usado para voltar do espaço de características reduzido para o espaço original) para que a imagem fica reconstruída e volta a ser com as dimensões padrão porém com a perda de informações que o houve com o PCA. Após a utilização do inverse_tranform foi utilizado novamente a biblioteca cv2 para juntar novamente as cores e foi alocada em uma lista.

Para ser inicializados os testes de performance, usamos a biblioteca time (que permite fazer a manipulação do tempo) para computar o tempo de processamento, e foi rodada a rotina de código que faz a impressão das imagens da base toda usando a biblioteca matplotlib.pyplot (que permite a criação de gráficos e visualizações de dados de alta qualidade).

Como métrica de comparação, será utilizada o tamanho em MB da base e a velocidade de processamento das imagens. E foi avaliado uma imagem individual usando

diferentes números de componentes PCA e usando a variância como parâmetro de comparação da perda de informação.

Na figura 5 podemos visualizar o fluxograma do processo de obtenção de resultado para a base de imagens.

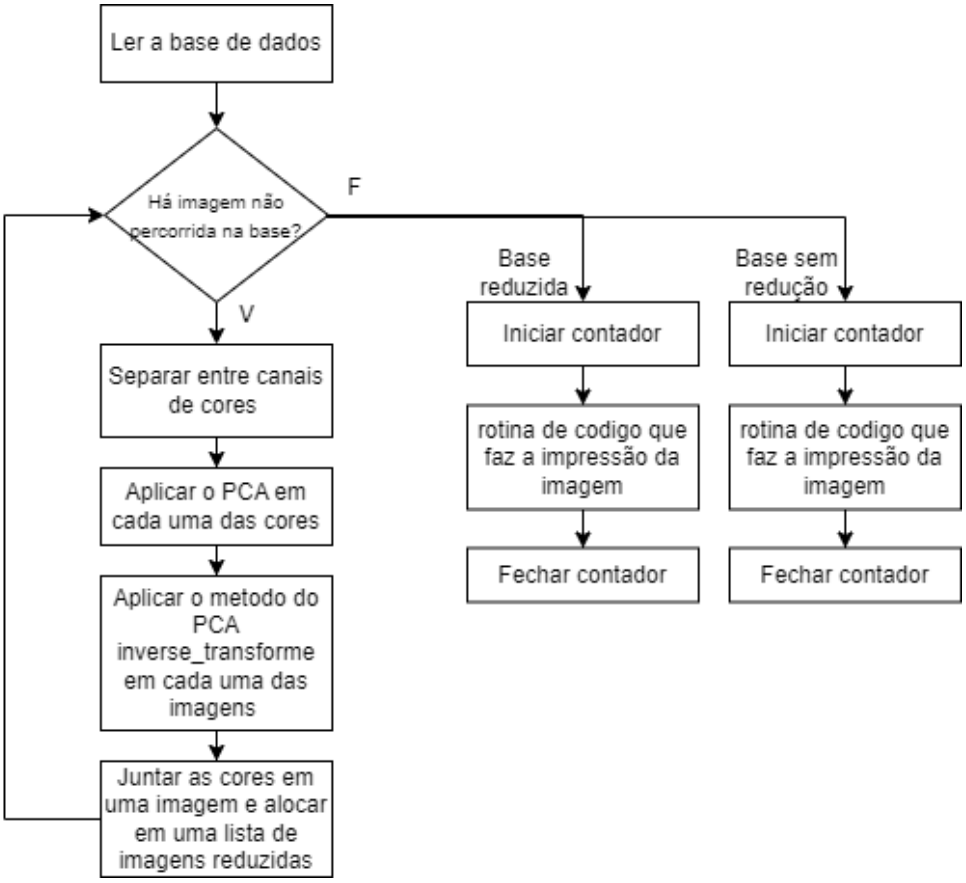

Figura 5. Fluxograma do método utilizado na base de dados não estruturado.

Fonte: Elaborado pelo autor (2023)

4. Resultados

4.1 Base de dados estruturado de classificação

As tabelas 1, 2, 3 e 4 apresentam o resultado da taxa de acerto e o tempo de execução em segundos da base de dados census de classificação binárias, na Tabela 1 apresenta os resultado da base sem a redução da dimensionalidade, a Tabela 2 apresenta os resultados da base reduzida pelo PCA com diferentes parâmetros n_components, a Tabela 3 apresenta os resultados da base reduzida pelo kernel PCA com diferentes parâmetros n_components e a Tabela 4 apresenta o resultado da base reduzida pelo LDA com somente um n_components devido a sua limitação com bases com classes binárias.

Tabela 1. Resultados obtidos da base de classificação sem redução.

Taxa de acerto	Tempo de execução
0.9404	2.563 segundos

Tabela 2. Resultados obtidos da base de classificação reduzida pelo PCA.

n_components	Taxa de acerto	Tempo de execução
14	0.9404	2.341 segundos
12	0.9404	2.303 segundos
10	0.9404	2.244 segundos
8	0.9404	2.349 segundos
6	0.9404	2.401 segundos
4	0.9404	2.474 segundos
2	0.9404	2.821 segundos

Tabela 3. Resultados obtidos da base de classificação reduzida pelo kernel PCA.

n_components	Taxa de acerto	Tempo de execução
14	0.9396	2.201 segundos
12	0.9384	2.274 segundos
10	0.9355	2.311 segundos
8	0.9312	2.383 segundos
6	0.9304	2.424 segundos
4	0.9186	2.481 segundos
2	0.8972	2.868 segundos

Tabela 4. Resultados obtidos da base de classificação reduzida pelo LDA.

n_components	Taxa de acerto	Tempo de execução
1	0.8908	2.981 segundos

Na Tabela 1 que a base sem redução teve uma boa taxa de acerto, mas pode haver oportunidade de melhoria no tempo de execução.

Na Tabela 2, foi notado que o PCA obteve uma taxa de acerto igual a base sem redução, portanto, não ocorreu perda de informação desnecessária quando reduzimos a base utilizada, deixando apenas a informação essencial. Já a questão do tempo de processamento, teve um ganho de performance de 12.5% com o n_components = 10, onde que teve mais performance segundo os teste obtidos, assim o melhor algoritmo nesse caso de classificação foi o PCA.

Na tabela 3, foi notado que o kernel PCA mostra uma variação de taxa de acerto sugerindo que diferentes configurações podem impactar a qualidade da representação reduzida. E o tempo de processamento foi aumentando conforme reduzimos o parâmetro n_components.

Na tabela 4, foi analisado que o LDA, por sua vez teve um resultado não muito satisfatório, devido a sua limitação em base de dados de classificação binaria, onde foi utilizado somente o n_components = 1, sua taxa de acerto teve um resultado pior e o tempo de execução maior em relação a base sem redução.

4.2 Base de dados estruturado de regressão

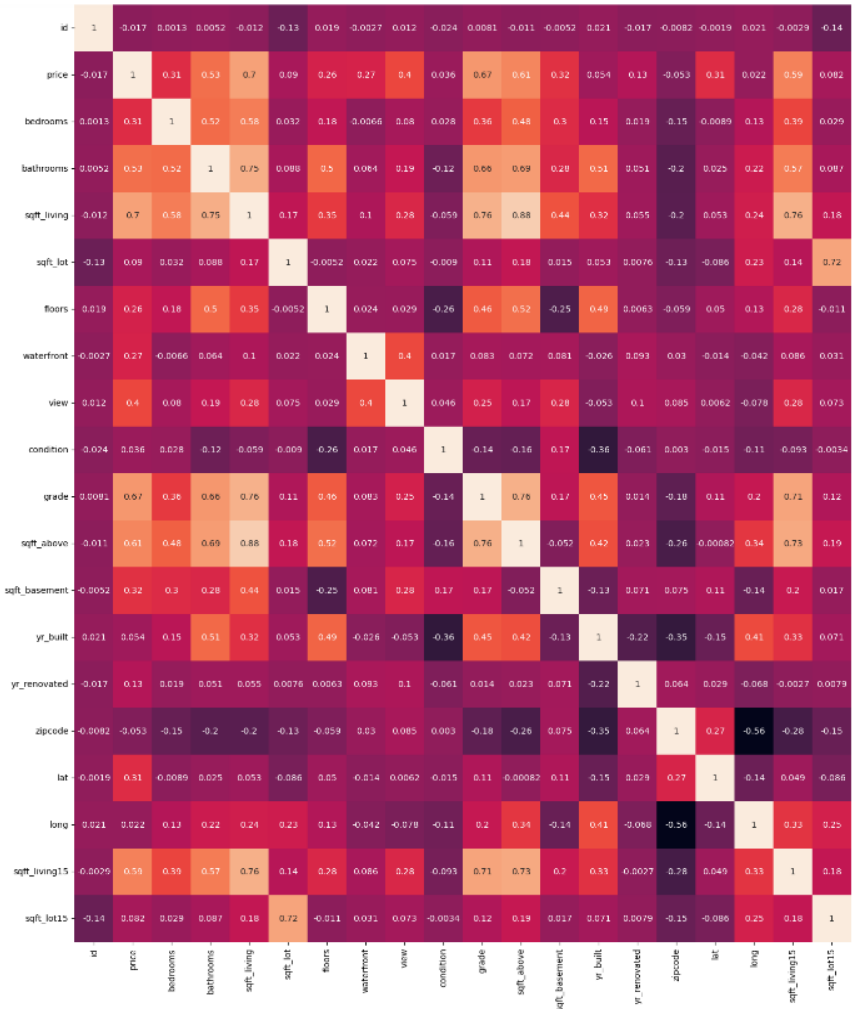

Figura 6. Gráfico de correlação criado pelo autor

Fonte: Elaborado pelo autor (2023)

Na Figura 6, apresenta um gráfico de correlação dos atributos presente na base de dados de regressão, nele basicamente contém o coeficiente de correlação que pode ir de -1 a 1. Um valor de 1 indica uma correlação positiva perfeita (à medida que o valor da

variável aumenta, o outro valor também aumenta), -1 indica uma correlação negativa perfeita (à medida que um valor da variável aumenta, o outro valor diminui), e 0 indica ausência de correlação linear. No caso da nossa base, podemos pegar a variável sqft_living e a variável sqft_living15, tem um coeficiente de correlação de 0.76, assim muito provavelmente os algoritmos de redução de dimensionalidade podem fazer a extração de características, juntando os 2 atributos devido a sua similaridade.

As tabelas 5, 6, 7 e 8 apresentam o resultado da taxa de acerto, do mean_absolute_error e o tempo de execução em segundos da base de dados house prices de regrassão, na Tabela 5 contém os resultado da base sem a redução da dimensionalidade, a Tabela 6 contém os resultados da base reduzida pelo PCA com diferentes parâmetros n_components, a Tabela 7 contém os resultados da base reduzida pelo kernel PCA com diferentes parâmetros n_components e a Tabela 8 contém os resultados da base reduzida pelo LDA com diferentes parâmetros n_components.

Tabela 5. Resultados obtidos da base de regressão sem redução.

Taxa de acerto	Erro	Tempo de execução
0.875	76620	274.181 segundos

Tabela 6. Resultados obtidos da base de regressão reduzida pelo PCA

N_components	Taxa de acerto	Erro	Tempo de execução
16	0.881	74.811	274.498 segundos
14	0.875	77.400	278.630 segundos
12	0.865	78.189	268.266 segundos
10	0.829	89.368	264.268 segundos
8	0.813	89.841	257.877 segundos
6	0.763	105.411	253.703 segundos
4	0.666	135.690	252.057 segundos
2	0.632	143.166	245.153 segundos

Tabela 7. Resultados obtidos da base de regressão reduzida pelo kernel PCA

N_components	Taxa de acerto	Erro	Tempo de execução
16	0.716	90.505	305.011 segundos
14	0.705	95.932	297.465 segundos
12	0.688	101.872	269.198 segundos
10	0.666	102.646	273.449 segundos
8	0.659	107.767	277.718 segundos
6	0.597	119.540	260.827 segundos
4	0.530	141.316	248.274 segundos
2	0.463	159.849	237.248 segundos

Tabela 8. Resultados obtidos da base de regressão reduzida pelo LDA

N_components	Taxa de acerto	Erro	Tempo de execução
16	0.881	76.323	275.234 segundos
14	0.871	77.894	288.409 segundos
12	0.852	84.612	274.185 segundos
10	0.850	84.187	271.937 segundos
8	0.823	91.019	265.443 segundos
6	0.800	97.137	261.267 segundos
4	0.781	102.447	255.100 segundos
2	0.762	108.197	253.292 segundos

Podemos analisar a Tabela 5, que a base sem redução teve um score de 0.875 e um tempo de processamento de 274.181.

Na Tabela 6, notamos que a base com o PCA aplicado conforme vai diminuindo o parâmetro de n_components vai diminuindo o score e o tempo de processamento, somente o n_components = 16 contém um score 0.06 maior e um tempo de processamento 317 milissegundos maior em relação a base sem redução.

Na Tabela 7, foi notado um desempenho muito ruim por parte do kernel PCA, perdendo muito score e um tempo de processamento elevado em relação a base sem redução. Por ter um desempenho inferior nas métricas fornecidas, sugerindo que pode não ser a melhor escolha para esta aplicação específica.

Na Tabela 8, foi notado que a base com o LDA aplicado, teve um resultado mais satisfatório, onde foi o algoritmo mais equilibrado em termos de precisão e tempo de processamento, que por sua vez teve uma perda somente de 0.113 com n_components = 2, e teve um aumento de performance de 7.7%.

Nesse cenário, todos os algoritmos quando se diminui o parâmetro n_componens tiveram perda de informação deixando o score mais baixo, porém teve um benefício de ganho de performance. Portanto, a escolha entre obter um score mais preciso ou um desempenho superior recai sobre o administrador do banco de dados.

4.3 Base de dados não estruturado

As tabelas 5, 6, 7 e 8 apresentam o resultado do tempo de impressão em segundos e o tamanho em MB da base de 200 imagens, na Tabela 9 contém os resultados da base sem redução e na Tabela 10 os resultados da base reduzida pelo PCA com o parâmetro n_components = 100, 50, 20 e 5.

Tabela 9. Resultados obtidos da base imagens sem redução.

Tamanho em MB	Tempo de impressão
31,6	26.153 segundos

Tabela 10. Resultados obtidos da base de imagens reduzida pelo PCA

N_components	Tamanho em MB	Tempo de impressão
100	4,45	20.521 segundos
50	4,21	19.842 segundos
20	3,68	17.987 segundos
5	2.80	17.130 segundos

Com a análise da Tabela 9 e da Tabela 10, podemos verificar que a base reduzida teve uma diminuição de armazenamento de 85.9%, 86.7%, 88.3% e 91.1% respectivamente para os n_components = 100, 50, 20 e 5 e um ganho de processamento de 21.6%, 24.1%, 31.2% e 34.5% respectivamente para os n_components = 100, 50, 20 e 5 no momento da impressão, mostrando a eficiência que o PCA tem com redução de imagens.

Agora analisaremos diferentes n_components PCA aplicados em uma imagem:

Figura 7. Conjunto de imagem com diferentes números de componentes PCA

Fonte: Elaborado pelo autor (2023)

Na figura 7, é apresentado um conjunto de imagens, a primeira imagem é a original sem o PCA aplicado e o restante contém números de componentes PCA aplicados de 100, 50, 20 e 5 respectivamente da esquerda para direita. A imagem, antes do PCA ser aplicado tinha as dimensões de 495 x 500, depois do PCA aplicado ficou com 495 x valor do parâmetro n_components, e após fazer a reconstrução da imagem retornou com as dimensões originais de 495 x 500, porém com a perda da qualidade.

Uma medida para diferenciar as imagens são as variâncias, no contexto do PCA refere-se à quantidade de informação contida nas principais componentes (autovetores) resultantes da redução de dimensionalidade realizada. Por se tratar de uma imagem colorida, contêm três canais de cor principais (Red, Green, Blue) podem ser chamadas de RGB, cada cor tem uma variância para medir a variabilidade ou dispersão das intensidades de cada cor na imagem colorida. Na Tabela 12 podemos visualizar a variância red, green e blue e o tamanho em Kb de diferentes n_components da Figura 7.

Tabela 11. Tamanho da imagem da imagem sem redução.

Tamanho em Kb	139

Tabela 12. Dados da imagem com diferentes n_components PCA aplicado.

N_components	Tamanho em Kb	Variância Red	Variância Green	Variância Blue
100	36	0.987	0.987	0.987
50	35,3	0.966	0.965	0.965
20	32,3	0.917	0.917	0.905
5	25	0.752	0.792	0.784

Com a análise da Tabela 12 em relação a Tabela 11, podemos verificar que o armazenamento da imagem diminui em 74.2% com o parâmetro n_components que menos perdeu informação, e conforme vai abaixando o n_components, diminui também as informações presente nas imagens e o tamanho delas.

5. Conclusão

Podemos concluir que, com base nos resultados apresentados, em relação as base de dados estruturados, o algoritmo PCA teve resultados melhores na base de classificação, não perdendo informações que faria o a precisão do Random Forest decrescer e aumentando o desempenho em 12.5% nos melhores dos seus parâmetros. Já o LDA, teve maior eficaz na base de regressão, perdendo em média de 10.29% da precisão calculada pelo Random Forest, resultando melhor precisão em comparado com o PCA e Kernel PCA que teve uma perda de informação de 22.80% e 33.94% respectivamente. O kernel PCA teve resultados satisfatórios na base de classificação, porém para haver total do seu desempenho, a base teria que ser não linear, diferentes das bases testadas.

Em relação as bases de dados não estruturados, o melhor resultado que apresentado foi com o parâmetro PCA n_components = 100, que teve uma diminuição de armazenamento de sua base de 85.9%, um aumento na velocidade da sua impressão de 21.6%, tudo isso perdendo somente 1.30% de suas informações, ou seja, quase imperceptível ao olhos humanos. Conforme ocorre a diminuição do parâmetro PCA n_components, vai diminuindo o armazenamento e aumentando a velocidade de impressão, porém a imagem vai perdendo muita qualidade em comparação a imagem n_components = 100.

Referências Bibliográficas

H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscipl. Rev.: Comput. Statist. 2 (4) (2010) 433–459. https://www.researchgate.net/publication/227644862_Principal_Component_Analysis

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830

ANOWAR, Farzana; SADAOUI, Samira; SELIM, Bassant. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). Computer Science Review. Montreal, Canada, p. 1-13. jan. 2021.
https://www.researchgate.net/publication/349483622_Conceptual_and_empirical_comparison_of_dimensionality_reduction_algorithms_PCA_KPCA_LDA_MDS_SVD_LLE_ISOMAP_LE_ICA_t-SNE

KARAMIZADEH, Sasan; ABDULLAH, Shahidan M.; MANAF, Azizah A.; ZAMANI, Mazdak; HOOMAN, Alireza. An Overview of Principal Component Analysis. Journal Of Signal And Information Processing. Cyberjaya, Malaysia, p. 173-175. abr. 2013.
https://www.scirp.org/pdf/jsip_2013101711003963.pdf

SIDHU, Gagan S.; ASGARIAN, Nasimeh; GREINER, Russell; BROWN, Matthew R. G.. Kernel Principal Component Analysis for dimensionality reduction in fMRI-based diagnosis of ADHD. Frontiers In Systems Neuroscience. Edmonton, Ab, Canada, p. 1-16. nov. 2012.
https://www.frontiersin.org/articles/10.3389/fnsys.2012.00074/full#h9

H., Telgaonkar Archana; SACHIN, Deshmukh. Dimensionality Reduction and Classification through PCA and LDA. International Journal Of Computer Applications. Aurangabad, p. 4-8. jul. 2015.
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=414722ddd809b460d5b397eaf454fbb697cfb881

MA, Yanyuan; ZHU, Liping. A Review on Dimension Reduction. International Statistical Review. [S. L.], p. 134-150. abr. 2013.
https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2012.00182.x

MAATEN, L.J.P. van Der; POSTMA, E.O.; HERIK, H.J. van Den. Dimensionality Reduction: A Comparative Review. Journal Of Machine Learning Research. Maastricht, p. 1-22. jan. 2007.
https://www.researchgate.net/publication/228657549_Dimensionality_Reduction_A_Comparative_Review

G. Baudat, F. Anouar, Kernel-based methods and function approximation, in: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), Vol. 2, IEEE, 2001, pp. 1244–1249.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction.

Reddy G, T, Kumar Reddy M, P, Lakshmanna, K, Kaluri, R, Singh Rajput, D, Srivastava, G, and Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. IEEE Access, 8, 54776-54788.

H., Telgaonkar Archana; SACHIN, Deshmukh. Dimensionality Reduction and Classification through PCA and LDA. International Journal Of Computer Applications. Aurangabad, p. 0975-8887. jul. 2015.
Belarbi, Mohammed Amin & Saïd, Mahmoudi & Belalem, Ghalem. (2017). PCA as Dimensionality Reduction for Large-Scale Image Retrieval Systems. International Journal of Ambient Computing and Intelligence. 8. 14. 10.4018/IJACI.2017100104.

NAKRA, Abhilasha; DUHAN, Manoj. Feature Extraction and Dimensionality Reduction Techniques with Their Advantages and Disadvantages for EEG-Based BCI System: A Review. Iup Journal Of Computer Sciences. [S. L.], p. 21-34. jan. 2020.