

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**APRENDIZADO NÃO SUPERVISIONADO DE  
MÉTRICAS VIA REDUÇÃO DE  
DIMENSIONALIDADE PARA  
RECONHECIMENTO DE DÍGITOS NUMÉRICOS**

**FELIPE CHURUYUKI CHINEN**

**ORIENTADOR: PROF. DR. ALEXANDRE LUÍS MAGALHÃES LEVADA**

São Carlos – SP  
17 de novembro de 2021

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**APRENDIZADO NÃO SUPERVISIONADO DE  
MÉTRICAS VIA REDUÇÃO DE  
DIMENSIONALIDADE PARA  
RECONHECIMENTO DE DÍGITOS NUMÉRICOS**

**FELIPE CHURUYUKI CHINEN**

Monografia apresentada ao Departamento de Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia da Computação, área de concentração: Reconhecimento de Padrões  
Orientador: Prof. Dr. Alexandre Luís Magalhães Levada

São Carlos – SP

17 de novembro de 2021

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente aos meus pais Cássio e Eliana, pois me apoiaram desde o primeiro dia de faculdade e continuaram me apoiando durante esses longos anos.

Gostaria de agradecer também ao Professor Doutor Alexandre Levada, que teve a paciência de me orientar nesse trabalho.

## RESUMO

CHINEN, Felipe. APRENDIZADO NÃO SUPERVISIONADO DE MÉTRICAS VIA REDUÇÃO DE DIMENSIONALIDADE PARA RECONHECIMENTO DE DÍGITOS. 48f. Trabalho de Conclusão de Curso - Engenharia de Computação, Universidade Federal de São Carlos. São Carlos. 2021.

O trabalho com reconhecimento de caracteres ocorre desde meados do século passado e até hoje é um tema muito estudado. Um dos desafios encontrados é que os caracteres a serem classificados são imagens, dessa maneira, contém várias dimensões (sendo cada *pixel*, numa imagem de nível de cinza, uma dimensão diferente). Trabalhar com conjunto de dados com muitas dimensões é um desafio, devido a problemas como a maldição da dimensionalidade. Para isso, a redução de dimensionalidade é geralmente como uma tentativa de mitigar esse problema. Neste trabalho foram utilizados diversos algoritmos de classificação supervisionada e foram utilizados os algoritmos de redução de dimensionalidade não linear ISOMAP, LLE e *Laplacian Eigenmaps* com o intuito de encontrar uma métrica (função de distância) mais adequada que a distância Euclidiana entre as imagens de entrada. Por fim, foi utilizado o teste de Wilcoxon para comparar estatisticamente os algoritmos de redução de dimensionalidade .

**Palavras-chave:** reconhecimento de padrões, aprendizado de máquina, aprendizado não supervisionado de métricas, redução de dimensionalidade.

# ABSTRACT

CHINEN, Felipe. UNSUPERVISED METRIC LEARNING VIA DIMENSIONALITY REDUCTION FOR DIGIT RECOGNITION. 48f. Trabalho de Conclusão de Curso - Engenharia de Computação, Universidade Federal de São Carlos. São Carlos. 2021.

Character recognition is a task that has occurred since the last century and is still a problem to be solved nowadays. One of the challenges is that the characters to be classified are images, therefore, they have many dimensions (each pixel is a dimension in a gray image). In this paper were used two different databases that contain handmade characters. Working with a database with many dimensions is a challenge, due to issues like the curse of dimensionality, and to try to solve this complication, dimensionality reduction is usually an attempt to mitigate the problem. In this work were used several supervised classification algorithms and were used non-linear dimensionality reduction algorithms: ISOMAP, LLE and Laplacian Eigenmaps with the purpose of learning a metric (a distance function) better suited than the Euclidean distance among the input images. Finally, the Wilcoxon test was used to compare statistically the dimensionality reduction algorithms.

**Keywords:** pattern recognition, machine learning, unsupervised metric learning, dimensionality reduction.

## LISTA DE FIGURAS

3.1	Figura dos dígitos da base de dados MNIST . . . . .	28
3.2	Figura dos dígitos da base de dados do Scikit Learn . . . . .	29
4.1	Figura que mostra a distribuição em 2d das classes quando utilizado o ISOMAP na base <i>Sklearn digits</i> . . . . .	38
4.2	Figura que mostra a distribuição em 2d das classes quando utilizado o LLE na base <i>Sklearn digits</i> . . . . .	39
4.3	Figura que mostra a distribuição em 2d das classes quando utilizado o <i>Laplacian eigenmaps</i> na base <i>Sklearn digits</i> . . . . .	40
4.4	Figura que mostra a distribuição em 2d das classes quando utilizado o ISOMAP na base MNIST . . . . .	41
4.5	Figura que mostra a distribuição em 2d das classes quando utilizado o LLE na base MNIST . . . . .	42
4.6	Figura que mostra a distribuição em 2d das classes quando utilizado o <i>Laplacian Eigenmaps</i> na base MNIST . . . . .	43

## LISTA DE TABELAS

2.1	Contém dados da classificação utilizando NCML, SLLE e LSDA . . . . .	21
2.2	Contém as informações do F1-Score da classificação do estudo 1 . . . . .	22
2.3	Contém as informações do F1-Score da classificação do estudo 2 . . . . .	22
2.4	Contém as informações do F1-Score da classificação do estudo 3 . . . . .	23
2.5	Contém as acurácias da classificação de cada conjunto . . . . .	24
2.6	Contém as sensividade, especificidade,acurácias da classificação dos músculos .	25
2.7	Contém taxa de reconhecimento da classificação dos para o PCA e o LE com o $k$ do classificador KNN variando . . . . .	25
2.8	Contém a média harmônica da sensibilidade e a utilização de todas as características da imagem . . . . .	26
2.9	Contém a média harmônica da sensibilidade e a utilização das características apenas do núcleo . . . . .	26
4.1	Contém informações sobre as acurácias de classificação utilizando o ISOMAP .	34
4.2	Contém informações sobre as acurácias de classificação utilizando o LLE . . .	34
4.3	Contém informações sobre as acurácias classificação utilizando o Laplacian Eigenmaps . . . . .	34
4.4	Contém informações sober o <i>Silhouette coefficient</i> . . . . .	35
4.5	Contém informações sobre as acurácias de classificação utilizando o ISOMAP .	35
4.6	Contém informações sobre as acurácias de classificação utilizando o LLE . . .	36
4.7	Contém informações sobre as acurácias classificação utilizando o <i>Laplacian Eigenmaps</i> . . . . .	36
4.8	Contém informações sober o <i>Silhouette coefficient</i> . . . . .	36

4.9	Contém informações sobre o teste de Wilcoxon para a base MNIST . . . . .	37
4.10	Contém informações sobre o teste de Wilcoxon para a base <i>Sklearn Digits</i> . . .	37



# GLOSSÁRIO

---

---

**ABC-PSO** – *Hybrid Artificial Bee Colony - Particle Swarm Optimization*

**AG** – *Algoritmo Genético*

**ARIMA** – *Autoregressive Integrated Moving Average*

**CCL** – *Comprometimento Cognitivo Leve*

**DA** – *Doença de Alzheimer*

**DT** – *Decision Tree*

**EEG** – *Eletroencefalograma*

**EMG** – *Electromyography*

**Ext-Isomap** – *Extended Isomap*

**FA** – *Factor Analysis*

**FLDA** – *Fisher Linear Discriminate Analysis*

**GRNN** – *Generalized Regression Neural Network*

**GRNN** – *Generalized Regression Neural Network*

**HLLE** – *Hessian Local Linear Embedding*

**ICA** – *Independent Component Analysis*

**ISOMAP** – *Isometric Feature Mapping*

**K-PCA** – *Kernel-PCA*

**KFD-Isomap** – *Kernel Fisher Discriminant Isomap*

**KIsomap** – *Kernel Isomap*

**KNN** – *K-Nearest Neighbors*

**LE** – *Laplacian Eigenmap*

**LE** – *Laplacian Eigenmaps*

**LLE** – *Locally Linear Embedding*

**MAE** – *Mean Absolute Error*

**MDS** – *Multidimensional Scaling*

**MLP** – *Multilayer Perceptron*

**MNSIT** – *Modified National Institute of Standards and Technology database*

**MRI** – *Magnetic Resonance Imaging*

**NB** – *Naive Bayes*

**OMF** – *Overlapping-detections Matched Filtering*

**PCA** – *Principal Component Analysis*

**PLSDA)** – *Partial least squares discriminant analysis*

**PSO** – *Particle Swarm Optimization*

**QDA** – *Quadratic Discriminant Analysis*

**RBF** – *Radial Basis Function*

**RF** – *Random Forest*

**RMSE** – *Root-Mean-Square Error*

**RPL-ISOMAP** – *Robust Positive semidefinite L-Isomap*

**SDDS-R10** – *Sloan Digital Sky Survey Data Release 10*

**SDSS** – *Sloan Digital Sky Survey*

**SVD** – *Singular value decomposition*

**SVM** – *Support Vector Machine*

**VMF** – *Vectorized Matched Filtering*

**WDMS** – *White Dwarf + Main Sequence*

**t-SNE** – *t-Distributed Stochastic Neighbor Embedding*

# SUMÁRIO

## GLOSSÁRIO

<b>CAPÍTULO 1 – INTRODUÇÃO</b>	<b>12</b>
1.1 Contextualização e motivação . . . . .	12
1.2 Objetivo Geral . . . . .	13
1.3 Objetivo Específico . . . . .	13
1.4 Organização do Trabalho . . . . .	14
<b>CAPÍTULO 2 – TRABALHOS RELACIONADOS</b>	<b>15</b>
2.1 ISOMAP . . . . .	15
2.2 LLE . . . . .	18
2.3 Laplacian eigenmaps . . . . .	23
<b>CAPÍTULO 3 – MATERIAIS E MÉTODOS</b>	<b>28</b>
3.1 MNIST . . . . .	28
3.2 Sklearn digits . . . . .	29
3.3 Plataforma computacional . . . . .	29
3.4 Métricas para avaliação de desempenho . . . . .	29
3.4.1 Silhouette coefficient . . . . .	29
3.4.2 Acurácia da classificação . . . . .	30
3.5 Classificação supervisionada . . . . .	31

3.5.1	Naive Bayes . . . . .	31
3.5.2	KNN . . . . .	31
3.5.3	Decision Trees . . . . .	31
3.5.4	Quadratic Discriminant Analysis . . . . .	32
3.5.5	Multilayer Perceptron . . . . .	32
3.5.6	Random Forest . . . . .	32
<b>CAPÍTULO 4 – RESULTADOS E DISCUSSÃO</b>		<b>33</b>
4.1	Sklearn digits . . . . .	33
4.2	MNIST . . . . .	35
4.3	Wilcoxon . . . . .	37
4.4	Visualização de Dados . . . . .	38
<b>CAPÍTULO 5 – CONCLUSÃO</b>		<b>44</b>
<b>REFERÊNCIAS</b>		<b>46</b>

# Capítulo 1

## INTRODUÇÃO

---

---

Neste capítulo serão apresentados a contextualização, motivação e objetivos e como está estruturado o trabalho.

### 1.1 Contextualização e motivação

Um dos primeiros sistemas de reconhecimento de caracteres data dos anos 40 e conforme os sistemas foram ficando mais robustos, em 1965 a IBM lançou sua primeira máquina que lia até números manuscritos, denominada “IBM 1287”(MORI; SUEN; YAMAMOTO, 1992). Nas décadas seguintes, sistemas de reconhecimento de caracteres foram desenvolvidos por institutos educacionais (WILKINSON, 1992). No início dos anos 2000 para preservar documentos históricos e poder armazenar estes de maneira digital, foram utilizadas técnicas de binarização(WOLF; JOLION; CHASSAING, 2002). Durante a metade de 2000, para ajudar pessoas com deficiência, foram introduzidas múltiplas aplicações. Na década atual foram utilizadas várias abordagens de aprendizado de máquina (LORIGO; GOVINDARAJU, 2006). Recentemente devido às melhorias de tecnologias, as abordagens para o reconhecimento de padrão se baseiam em aprendizagem profunda, como: Redes neurais recorrentes, redes neurais convolucionais, entre outras (MEMON et al., 2020).

O reconhecimento de caracteres foi tema escolhido para este trabalho, pois a base de dados são imagens e assim contém muitos atributos. Classificar base de dados com muitos atributos é um desafio, pois além de ter alta complexidade computacional, há a maldição de dimensionalidade, que por ter muitas dimensões, as amostras ficam esparsas no espaço aumentando a chance de *overfitting*. Dessa maneira, será utilizado no trabalho algoritmos de redução de dimensionalidade.

A redução de dimensionalidade mapeia os dados de grandes dimensões para baixas dimensões tentando manter a sua estrutura. A redução de dimensionalidade tem uma desvantagem, pois não é possível armazenar toda a informação de um espaço de alta dimensão para um espaço de baixa dimensão, dessa maneira, deve-se escolher quais informações são relevantes para se manter (GISBRECHT; HAMMER, 2015).

Tradicionalmente os algoritmos de redução utilizados eram lineares como o Principal Component Analysis (PCA), *factor analysis* e *classical scaling*, no entanto, algoritmos lineares não funcionam em dados não-lineares. Foram introduzidos então algoritmos não-lineares (Como os utilizados neste trabalho), nos quais tem a habilidade de lidar com dados não-lineares. Esses algoritmos têm vantagem pois no mundo real, os dados costumam ser não-lineares. Os algoritmos não-lineares conseguem demonstrar superioridade aos lineares em tarefas artificiais porém em conjunto de dados naturais os resultados são menos convincentes. Não está claro o porque da diferença de performance entre os algoritmos em dados artificiais e dados naturais (MAATEN et al., 2009).

Neste trabalho foi utilizado a redução de dimensionalidade, para verificar qual o efeito da redução na tarefa de classificação. Foi utilizado três algoritmos de redução de dimensionalidade o ISOMAP, *Locally Linear Embedding* (LLE) e o *Laplacian Eigenmaps* (LE). Uma questão levantada pelo trabalho é se algum desses algoritmos se destaca perante os outros ou são todos similares.

## 1.2 **Objetivo Geral**

O objetivo principal deste trabalho é utilizar algoritmos de redução de dimensionalidade não-lineares nas bases de dados para posterior classificação e verificar quais algoritmos de redução de dimensionalidade se sobressaem, na acurácia de classificação.

## 1.3 **Objetivo Específico**

Os objetivos específicos são:

- Utilizar algoritmos de redução de dimensionalidade para visualizar a distribuição das classes numa imagem 2D.
- Pesquisar sobre a utilização dos algoritmos de redução de dimensionalidade e verificar sua aplicação.

- Treinar e testar, após a redução de dimensionalidade, as bases de dados utilizando vários algoritmos de classificação.
- Calcular as acurácias obtidas para cada método utilizado.
- Analisar e comparar os resultados utilizando um teste estatístico e verificar se algum método de redução de dimensionalidade se sobressai.

## 1.4 Organização do Trabalho

Esse trabalho está organizado da seguinte maneira: No capítulo 1 é apresentado um breve histórico sobre a base de dados utilizado, em seguida, apresentado o problema, a motivação e os objetivos do trabalho. No capítulo 2 é apresentado os algoritmos principais algoritmos de aprendizado não-supervisionado de métricas seguido com trabalhos utilizando os mesmos. Ainda no capítulo 2 são apresentados de maneira sucinta os algoritmos de classificação utilizados no trabalho. No capítulo 3 é apresentado as base de dados utilizadas, junto com a plataforma computacional utilizada e a descrição das métricas de avaliação de desempenho. No capítulo 4 é discutido os resultados obtidos pelo trabalho. No capítulo 5 é feita a conclusão sobre os resultados obtidos.

# Capítulo 2

## TRABALHOS RELACIONADOS

---

---

Nesse capítulo será apresentado os trabalhos relacionados para os seguinte algoritmos: ISOMAP, LLE, e Laplacian Eigenmaps.

### 2.1 ISOMAP

O *Isometric Feature Mapping* (ISOMAP) é um algoritmo de redução de dimensionalidade não-linear. Diferente dos algoritmos de PCA (Principal Component Analysis) e MDS (Multidimensional Scaling) que utilizam a distância euclidiana, o ISOMAP utiliza as distâncias geodésicas para representar as distâncias no espaço de atributos. O algoritmo ISOMAP contém três etapas principais: a primeira, para criar um grafo, é utilizado o algoritmo de k-vizinhos mais próximos, sendo que cada peso da aresta do grafo é a distância euclidiana entre os pontos. A segunda, para estimar as distâncias geodésicas, utiliza-se o algoritmo de Dijkstra que forma uma árvore de caminhos mínimos e forma uma matriz de distância ponto-a-ponto. A terceira é a aplicação do do algoritmo MDS na matriz obtida na etapa anterior, pode-se verificar com mais detalhes esse passo no trabalho (TENENBAUM; SILVA; LANGFORD, 2000).

A seguir, é apresentada uma breve revisão de trabalhos que utilizam o algoritmo ISOMAP em aplicações de aprendizado de máquina.

No trabalho de Bu et. al (BU; CHEN; PAN, 2014), os autores utilizaram o ISOMAP para diminuir a quantidade de dimensões dos dados espectral estelar e depois classificá-los utilizando SVM(Support Vector Machine) com o intuito de analisar a performance dos algoritmos. Para a comparação, foi utilizado o algoritmo PCA para a redução de dimensionalidade e o algoritmo SVM para a classificação dos dados. Os dados utilizados para a classificação foram obtidos da *Sloan Digital Sky Survey* (SDSS), que contém 4 subclasses: tipo K1, tipo K3, tipo K5 e tipo K7.



Houve uma divisão dos dados em 3 conjuntos menores, sendo que o primeiro (DATA1) contém a razão *signal-to-noise* (SNR) menor que 5, o segundo (DATA2) entre 5 e 10 e o terceiro (DATA3) maior que 10.

Para a escolha da quantidade de dimensões  $k$ , é verificado o critério de taxa de contribuições acumuladas ( $C_k$ ) para o PCA. O valor escolhido para  $C_k$  foi 92%, dessa maneira, para que haja a garantia desse valor, foi escolhido para  $k$ , nesse experimento, o valor 10. Foi utilizado o algoritmo  $k$ -Isomap, que utiliza o algoritmo de  $k$ -vizinhos mais próximos para a geração do grafo. Os valores escolhidos para  $k$  foram 6, 8, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100. O valor escolhido para  $\lambda$  para o SVM foi o padrão do MATLAB r2011b, no entanto, foi verificado que diferentes valores para  $\lambda$  chegam ao mesmo resultado. Pode ser verificado que a performance da dupla ISOMAP e SVM foram superiores ao do PCA e SVM. Foi verificado que o valor da classificação muda de maneira significativa conforme o aumento de SNR em ambas abordagens. Além disso, pode-se notar que o resultado da classificação é afetado significativamente dependendo da dimensão escolhida para o ISOMAP

No trabalho de Du et. al (DU; ZHANG; WANG, 2009), os autores utilizaram o supervised ISOMAP para classificação das diferentes espécies de plantas. Para melhorar o processo de classificação, foi necessário um pré-processamento da imagem, no qual é feita uma segmentação para separar a folha da planta do fundo da imagem. Além disso, como o pecíolo da planta tem muita variação de tamanho e curvatura, estes também devem ser retirados.

Dessa forma foi obtida a base de dados contendo 20 espécies de plantas diferentes. Para cada espécie, contém 100 imagens de folhas diferentes, sendo que metade dessas imagens foram utilizada para treino e a outra metade para teste.

O classificador de 1-vizinho mais próximo foi utilizado e o algoritmo conseguiu classificar de maneira correta 92,25% das imagens de teste.

O trabalho de Prabhakar e Rajaguru (PRABHAKAR; RAJAGURU, 2016) utilizou diferentes algoritmos de redução de dimensionalidade nos dados de eletroencefalograma (EEG) e para classificar foi utilizado o Algoritmo Genético (AG) e o *Generalized Regression Neural Network* (GRNN). Os algoritmos de redução de dimensionalidade utilizados para comparação foram: ISOMAP, *Hessian Local Linear Embedding* (HLLE) e *Factor Analysis* (FA).

Para isso, a coleta da base de dados foram de 20 pacientes com epilepsia. Para obter as gravações, foi capturado 16 canais de eletrodos no couro cabeludo desses pacientes. Os períodos gravados dos pacientes foram: dormindo, acordados e ativos, com olhos abertos, com olhos fechados e semi-dormindo. Para facilitar a extração de características e a redução de

dimensionalidade, as gravações do EEG foram feitas em diferentes épocas e com a duração de pelo menos 35 minutos.

Todos os métodos de redução de dimensionalidade conseguiram 100% de acurácia utilizando o GRNN. Já para o algoritmo GA associado com algoritmo FA conseguiu 93,78%, quando associado com o ISOMAP conseguiu 93,645% e quando associado com o HLLE conseguiu 95,24% de acurácia.

O trabalho de Gu et. al (GU; XU; YE, 2007) propõe um novo algoritmo baseado no Isomap denominado *Kernel Fisher Discriminant Isomap* (KFD-Isomap) para extrair características para o classificador de vizinhos mais próximos fazer a tarefa de reconhecimento facial. Para aprender o mapeamento explícito de novas amostras, foi utilizado a técnica de interpolação denominada *Generalized Regression Neural Network* (GRNN). Para comparar a performance do algoritmo proposto, foram utilizados os seguintes algoritmos: Isomap, *Extended-Isomap* (Ext-Isomap), Fisherface e Eigenface.

Foram utilizadas duas bases de dados para aplicar os algoritmos, a ORL e a Yale, sendo que a primeira contém 400 imagens de 40 pessoas, sendo 36 homens e 4 mulheres e a segunda contém 165 imagens de 15 pessoas. Em ambas as bases de dados há variação na iluminação e em expressões faciais. Houve um pré-processamento das imagens, no qual foi retirado o fundo das imagens, mantendo apenas a face do indivíduo.

Os experimentos foram repetidos 400 vezes devido a aleatoriedade do conjunto de dados e foi computado a média dos resultados. O algoritmo proposto teve a menor taxa de erro(0,75%) contra 1,25% do Ext-Isomap, 3,5% do ISOMAP, 1,0% do Fisherface e 3,0% do Eigenface, na base de dados da ORL. Para a base de dados da Yale, o algoritmo proposto também teve a menor taxa de erro(8,48%) contra 13,94% do Ext-Isomap, 26,06% do ISOMAP, 9,7% do Fisherface e 30.30% do Eigenface.

O trabalho de Wang et. al (WANG et al., 2015) propõe a utilização do PCA e ISOMAP como algoritmos de redução de dimensionalidade e a utilização do *Particle Swarm Optimization* (PSO) para melhorar a acurácia de classificação do algoritmo SVM.

A base de dados utilizada no experimento é a *Sloan Digital Sky Survey Data Release 10* (SDSS-DR10) que contém  $2 \times 10^6$  fits por arquivo, sendo que cada fit representa um espectro celestial. Os autores desse artigo tem o propósito de classificar um específico tipo de binário denominado: *White Dwarf + Main Sequence* (WDMS). Foram obtidos um total de 6000 WDMS spectra e 6000 spectra aleatórios do conjunto de dados, sendo que 75% dos dados foram para o conjunto de treino e 25% para o conjunto de teste.

A variação das dimensões para o PCA e para o ISOMAP foram de 2 até 50, sendo que a melhor dimensão encontrada para o ISOMAP foi 20 e sua acurácia foi 84.27% e para o PCA foi 10 dimensões e acurácia de 83,25%

No trabalho Park et. al (PARK et al., 2012) propõe a utilização do algoritmo ISOMAP em *Magnetic Resonance Imaging* (MRI) do cérebro em pacientes normais, comprometimento cognitivo leve (CCL) e com doença de Alzheimer(DA), para reduzir a dimensionalidade e assim classificar o paciente. Foram obtidos as imagens do MRI da base de dados ADNI.

Dessa base, foram retirados a imagem do MRI de 75 pacientes. 25 dos pacientes com DA tinham idade entre 66 e 82 e eram 12 homens e 13 mulheres.25 dos pacientes com CCL tinham a idade entre 60 e 90 anos sendo 14 homens e 11 mulheres .25 dos pacientes normais tinham idade entre 71 e 90 sendo 14 homens e 11 mulheres.

Houve um agrupamento prévio do conjunto de dados,pareando os diferentes tipos de pacientes da seguinte forma: pacientes com DA e normais, pacientes com CCL e normais e o último DA e MCI.

Foi utilizado o algoritmo classificador k-vizinhos mais próximos para fazer a classificação e para comparação foi utilizado o algoritmo SVM usando um kernel linear de redução de dimensionalidade. Para compara a performance do ISOMAP, foi utilizado uma abordagem que utiliza o volume do hipocampo dos pacientes.

Para o grupo de pacientes com DA e normais, o ISOMAP foi superior, conseguindo 84% de acurácia contra 76% do hipocampo e 76% do SVM. Para o grupo de pacientes com CCL e normais o ISOMAP também se apresentou superior com 76% de acurácia contra 60% do hipocampo e 75% do SVM. Para o grupo com DA e CCL o SVM foi superior com 70% de acurácia contra 56% do ISOMAP e 54% do hipocampo

## 2.2 LLE

O algoritmo denominado *Locally Linear Embedding* (LLE) é um algoritmo de redução de dimensionalidade que se difere do ISOMAP por ser um método local, ou seja, utiliza apenas informações sobre o ponto e seus vizinhos mais próximos.

O funcionamento do algoritmo pode ser dividido em 3 passos, o primeiro é utilizar o algoritmo de k-vizinhos mais próximos, obtendo os vizinhos  $\vec{X}_j$  do ponto  $\vec{X}_i$ , sendo  $\vec{X}_i$  é a amostra em questão e  $\vec{X}_j$  são os seus k vizinhos escolhidos. O segundo, deve-se minimizar a seguinte

equação:

$$\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij} \vec{X}_j|^2 \quad (2.1)$$

Dessa equação, deve-se obter os pesos  $W_{ij}$ , utilizando os vetores  $\vec{X}_j$  como combinações lineares para formar o ponto  $\vec{X}_i$ . Duas restrições são necessárias nesse passo:  $W_{ij} = 0$  quando o  $\vec{X}_j$  não for vizinho de  $\vec{X}_i$  e  $\sum_j W_{ij} = 1$

O terceiro passo é mapear o vetor  $\vec{X}$  para o vetor  $\vec{Y}$  numa dimensão menor, minimizando a seguinte função:

$$\phi(Y) = \sum_i |\vec{Y}_i - \sum_j W_{ij} \vec{Y}_j|^2 \quad (2.2)$$

Nessa equação, diferente da 2.2 os pesos  $W_{ij}$  são fixados e são otimizado os valores de  $\vec{Y}$ . Nesse passo são necessárias as seguintes restrições:  $\sum_i \vec{Y}_i = 0$  e que  $\frac{1}{n} Y^T Y = I$ , sendo  $I$  a matriz identidade (ROWEIS; SAUL, 2000). A seguir são apresentados trabalhos, nos quais foram utilizados o algoritmo de LLE no tema de aprendizado de máquina.

No trabalho Huang et. al (HUANG et al., 2012) propõe a utilização dos algoritmos LLE, *mean-LLE* e *mean spectra* para a redução de dimensionalidade e os algoritmos *Partial least squares discriminant analysis* (PLSDA) e SVM para a classificação.

Foram obtidos dados de 580 maçãs, que foram divididas em 240 que estariam maduras e mantidas geladas e 340 foram mantidas em 95% de umidade relativa e 20°C para acelerar o amadurecimento. De cada maçã, foram obtidas imagens de dispersão hiperespectral.

Os valores ótimos para o LLE foram  $K = 9$  e  $d = 49$  e para o algoritmo *mean-LLE* foram  $K = 12$  e  $d = 44$ . Os resultados obtidos para LLE em conjunto com o PLSDA foi 80.4% de acurácia, o *mean-LLE* com o PLSDA obteve 76.2% e o *mean spectra* com o PLSDA obteve 73%. Para o LLE com o SVM a acurácia foi de 82.5%, para o *mean-LLE* com o SVM obteve 79.4% e o *mean spectra* com o SVM obteve 78.3%.

No trabalho de Wang e Ding (WANG; DING, 2012) é proposto um novo algoritmo baseado no LLE denominado KFDDA-LLE, que utiliza o LLE para diminuir a dimensionalidade e então, utiliza o *kernel Fisher Discriminant Analysis* para encontrar a melhor direção para a projeção.

Os dados utilizados foram obtidos da base de dados *ORL face*. A base de dados contém 10 imagens diferentes de 40 pessoas. As imagens estão em escala de cinza e estão normalizadas da maneira que os olhos estão alinhados na mesma posição e as imagens foram redimensionadas para 32x32.

No experimento, foram escolhidas de maneira aleatória 5 imagens por indivíduo, enquanto

as outras 5 imagens foram para o conjunto de teste. Foram realizadas 10 réplicas do experimento, para reduzir a flutuação dos resultados causados pela aleatoriedade e foi retirada a média desses experimentos. O classificador de vizinhos mais próximos foi o algoritmo utilizado para a classificação.

Os resultados obtidos do experimento foram: *Eigenface* obteve taxa de reconhecimento de 86,63%, *Fisherface* obteve 93,02%, LLE obteve 89,74% e o algoritmo proposto KFDDA-LLE obteve 97,86%.

No trabalho Rajaguru e Prabhakar (RAJAGURU; PRABHAKAR, 2018) utiliza dados de EEG para classificar epilepsia. O algoritmo funciona da seguinte maneira: os dados de EEG tem suas dimensões reduzidas pelos algoritmos LLE e HLLE e a partir de então é utilizado o algoritmo *Hybrid Artificial Bee Colony - Particle Swarm Optimization* (ABC-PSO) para a classificação.

Para obter os dados do EEG foi colocado no couro cabeludo dos pacientes com epilepsia, 16 canais de eletrodos. Cada sessão foi dividida em épocas que teve a duração de mais de 50 minutos cada.

O acurácia obtida com o algoritmo LLE junto com o ABC-PSO foi de: 95,52% na primeira época, 95,72% na segunda época, 95,2% na terceira época e 95,48% na média. Para o algoritmo HLLE junto com o ABC-PSO as acurácias obtidas foram: 97,29% para a primeira época, 97,29% para a segunda época, 97,6% para a terceira e 97,39% na média

No trabalho de Yu et. al (YU et al., 2020) foi utilizado o algoritmo LLE para a redução de dimensionalidade nos indicadores que afetam o preço de ações. Para a predição do valor do preço da ação foi utilizado o *Back Propagation neural network* (BP).

Foram obtidos dados da ação de *Pingtan Development* durante o período de 1 de janeiro de 2016 até 8 de agosto de 2017. Foram selecionados um total de 14 indicadores. Esses indicadores foram pré-processados utilizando o método Z-score.

Os parâmetros escolhidos para o algoritmo LLE foi  $k = 100$  e  $d = 4$ . Para a predição, foram selecionados o período entre 1 de janeiro de 2016 até 28 de julho de 2017 como amostras de treino e 29 de julho de 2017 até 8 de agosto de 2017 como amostras de testes.

O modelo criado foi comparado com BP, PCA-BP, *autoregressive integrated moving average* (ARIMA). As métricas utilizadas para o resultado foram o valor *mean absolute error* (MAE) e *root-mean-square error* (RMSE).

O resultado final foi: BP obteve RMSE 0,7923 e MAE 0.7040, LLE-BP obteve RMSE

0,1619 e MAE 0,1483, PCA-BP obteve RMSE 0,7688 e MAE 0,5121 e ARIMA obteve RMSE 0,4911 e MSE 0,3945

No trabalho de He et. al (HE et al., 2019) os autores propõe um novo algoritmo de classificação baseado no LLE denominado *Nonlinear Manifold Classification* (NMCL). Para o teste deste algoritmo, foram utilizados 7 base de dados diferentes sendo eles: base de dados de doenças hepáticas, base de dados de diabétes, base de dados de meningite bacteriana aguda, base de dados de câncer de mama, distrofia muscular, base de dados de massa mamográfica e base de dados de bebês com nascimento com peso muito baixo. Os algoritmos utilizados como comparação foram: *Supervised LLE* (SLLE) *Local Sensitive Discriminant Analysis* (LSDA)

Os resultados obtidos podem ser verificados na tabela abaixo:

**Tabela 2.1: Contém dados da classificação utilizando NCML, SLLE e LSDA**

Dataset	SLLE(%)	LSDA(%)	NMCL(%)
<i>Liver</i>	83,23	86,25	88,3
<i>Diabetes</i>	73,96	79,41	82,73
<i>Meningitis</i>	72,98	73,42	75,87
<i>Breast cancer</i>	71,55	74,98	75,25
<i>Dystrophy</i>	83,72	71,35	74,54
<i>VLBW infant</i>	74,67	76,52	78,62

Fonte: (HE et al., 2019)

Em Anowar et. al (ANOWAR; SADAQUI; SELIM, 2021), três casos de estudo foram realizados para verificar a performance de classificação após utilizar os algoritmos de redução de dimensionalidade, são eles: PCA, *Linear discriminant analysis* (LDA), MDS, *Singular value decomposition* (SVD), ISOMAP, *Laplacian eigenmap* (LE), Isomap, LLE, *Independent component analysis* (ICA), t-Distributed Stochastic Neighbor Embedding (t-SNE). É utilizado o SVM com kernel *Radial Basis Function* (RBF) como o algoritmo de classificação.

No primeiro caso de estudo, foi utilizado a base de dados ECG200 que contém com dados de 96 dimensões. A performance de cada algoritmo pode ser verificada na tabela abaixo:

O segundo caso de estudo foi feito utilizando a base de dados Arcene juntando dados de espectrometria de massa. Os dados obtidos foram os seguintes:

O terceiro caso de estudo foi uma base de dados multi-classe denominado *Human Activity Recognition Using Smartphones* do repositório da UCI. Os resultados obtidos estão a seguir:

**Tabela 2.2: Contém as informações do F1-Score da classificação do estudo 1**

Dataset	Dimensão	F1-Score
Original	96	0,8966
PCA	65	0,9205
LDA	1	0,9885
MDS	70	0,9168
KPCA	65	0,9303
SVD	55	0,9205
LLE	70	0,9163
ISOMAP	81	0,9022
LE	71	0,9181
ICA	26	0,9271
t-SNE	2	0,8928

Fonte: (ANOWAR; SADAQUI; SELIM, 2021)

**Tabela 2.3: Contém as informações do F1-Score da classificação do estudo 2**

Dataset	Dimensão	F1-Score
Original	10000	0,630
PCA	200	0,8019
LDA	1	0,8181
MDS	186	0,8118
KPCA	200	0,7162
SVD	76	0,8481
LLE	102	0,7409
ISOMAP	57	0,6765
LE	96	0,8482
ICA	70	0,8036
t-SNE	2	0,7997

Fonte: (ANOWAR; SADAQUI; SELIM, 2021)

**Tabela 2.4: Contém as informações do F1-Score da classificação do estudo 3**

Dataset	Dimensão	F1-Score
Original	561	0,9505
PCA	300	0,9557
LDA	5	0,9848
MDS	478	0,9250
KPCA	300	0,9679
SVD	243	0,9562
LLE	530	0,9527
ISOMAP	480	0,9505
LE	200	0,9536
ICA	226	0,9575
t-SNE	2	0.6786

Fonte: (ANOWAR; SADAQUI; SELIM, 2021)

## 2.3 Laplacian eigenmaps

O algoritmo *Laplacian eigenmaps* (LE) é um algoritmo de redução de dimensionalidade não-linear que tenta manter as propriedades locais, de forma similar ao LLE. O funcionamento do algoritmo é descrito a seguir.

Primeiro, utiliza-se o algoritmo dos k-vizinhos mais próximos para criar um grafo. Em seguida, é calculado a matriz de adjacências  $W$  utilizando a seguinte fórmula:

$$W_{ij} = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{t}\right) \quad (2.3)$$

Então, deve-se computar a matriz diagonal  $D$ , somando as linhas da matriz de adjacências da seguinte forma:

$$d_i = \sum W_{ij} \quad (2.4)$$

Em seguida, deve-se computar a matriz laplaciana:  $L = D - W$  e selecionar os  $d$  menores autovetores associados aos menores autovalores não-nulos e definir a nova matriz  $Y$  (BELKIN; NIYOGI, 2001). A seguir, são apresentados trabalhos que utilizaram o algoritmo Laplacian eigenmaps (LE).

Em Ying et. al (YING et al., 2009) os autores propõe a utilização de uma combinação de *Local Binary Pattern* (LBP) e LE para o reconhecimento de expressão facial.

Nesse experimento foram retiradas expressões faciais da base de dados denominada *Japanese Female Facial Expression* (JAFFE) que contém 213 imagens de 10 japonesas. Há 21



imagens diferentes por pessoa com 7 expressões faciais diferentes (Bravo, desgosto, medo, felicidade, neutro, triste e surpreso). Sendo que para o experimento, foram selecionadas 3 cópias de cada expressão facial, totalizando 210 imagens.

Para a classificação, foi utilizado o algoritmo SVM em todos os casos, sendo que como base de comparação foi utilizado LBP+PCA e LBP+LDA. Os resultados foram o seguinte:

**Tabela 2.5: Contém as acurácias da classificação de cada conjunto**

Expressão	LBP+PCA(%)	LBP+LDA(%)	LBP+LE(%)
Bravo	90	73,33	90
Desgosto	56,67	70	60
Medo	70	56,67	66,67
Felicidade	80	73,33	83,33
Neutro	50	66,67	70
Triste	56,67	46,67	50
Surpreso	76,67	63,33	73,33
Total	68,57	64,29	70,48

Fonte: (YING et al., 2009)

Em Leon-Medina et. al (LEON-MEDINA et al., 2020) os autores propõe a comparação de 4 tipos diferentes de algoritmos de redução de dimensionalidade para reduzir dados obtidos por um nariz eletrônico. O classificador utilizado foi o KNN para  $k = 1$ .

A base de dados contém 3600 medições de seis compostos orgânicos voláteis, e foi medido por 16 sensores de gases. As acurácias obtidas foram: 0.9833 para *Kernel PCA*, 0.9789 para ISOMAP, 0.9389 para *Laplacian Eigenmaps* e 0.9122 para LLE

Em Lasghari e Demircan (LASHGARI; DEMIRCAN, 2017) foi proposto a utilização de o algoritmo LE para os dados de *Electromyography* (EMG) para então classificar foram utilizados KNN, *Fisher Linear Discriminate Analysis* (FLDA) e o classificador Bayesiano.

Nesse estudo foi obtido dados de *Electromyography* (EMG) de 10 músculos diferentes de 10 pessoas em 4 velocidades diferentes (2m/s, 3m/s, 4m/s e 5m/s), sendo todos os indivíduos homens em média de 71kg, 1.77m e 30 anos.

O melhor resultado obtido para os músculos *vastus medialis* foi utilizando  $k = 20$  dimensões para o LE e o classificador Bayesiano que conseguiu  $97,87 \pm 0,69$  de sensividade,  $88.37 \pm 0.79$  de especificidade e  $97.07 \pm 0.29$ .

O resultado para os outros músculos para o classificador bayesiano com  $k = 20$  para a dimensão do LE:

**Tabela 2.6: Contém as sensibilidade, especificidade,acurácias da classificação dos músculos**

Músculo	Sensividade(%)	Especificidade(%)	Acurácia(%)
<i>soleus</i>	98,10 ± 0,34	89,10 ± 0,33	97,01 ± 0,52
<i>gastrocnemius medialis</i>	98,98 ± 0,62	90,61 ± 0,44	96,61 ± 0,42
<i>gastrocnemius lateralis</i>	98,02 ± 0,56	88,55 ± 0,16	96,18 ± 0,82
<i>vastus medialis</i>	97,87 ± 0,69	88,37 ± 0,79	97,07 ± 0,29
<i>vastus lateralis</i>	97,32 ± 0,21	86,92 ± 0,91	98,54 ± 1,01
<i>rectus femoris</i>	98,03 ± 0,61	89,15 ± 0,56	98,32 ± 0,54
<i>biceps femoris-long head</i>	99,05 ± 0,01	91,65 ± 0,23	98,65 ± 0,02
<i>gluteus medius</i>	98,45 ± 0,84	92,74 ± 0,84	97,70 ± 0,84
<i>gluteus maximus</i>	98,71 ± 0,24	92,70 ± 0,24	96,70 ± 0,12
<i>tibialis anterior</i>	98,70 ± 0,64	93,31 ± 0,78	93,32 ± 0,14

Fonte: (LASHGARI; DEMIRCAN, 2017)

Em Raducanu e Dornaika (RADUCANU; DORNAIKA, 2010), os autores propõem a utilização do LE para o aprender a estrutura da expressão facial e a utilização do classificador KNN para o reconhecimento de expressão facial.

A base de dados obtida foi a da *CMU facial expression database* que contém 20 pessoas que expressam 5 tipos de expressão, sendo elas, surpresa, tristeza, alegria, desgosto e raiva. O algoritmo LDA foi utilizado para refinar os dados de representação para o reconhecimento. Foi utilizado 90% das amostras para teste e 10% para teste.

Foi utilizado o algoritmo PCA como base de comparação para o LE. As acurácias obtidas podem ser verificadas na seguinte tabela:

**Tabela 2.7: Contém taxa de reconhecimento da classificação dos para o PCA e o LE com o  $k$  do classificador KNN variando**

LE/PCA	K-NN=1	K-NN=3	K-NN=5
5	91,42/91,42	88,57/88,57	91,42/91,42
10	97,14/94,285	97,14/91,42	97,14/91,42
15	91,42/85,71	91,42/85,71	91,42/85,71
20	88,57/68,57	88,57/65,71	88,57/68,57

Fonte: (RADUCANU; DORNAIKA, 2010)

Em Plissiti e Nikou (PLISSITI; NIKOU, 2012) fizeram um *framework* para a classificação de células cervicais normais e anormais.

A base de dados dos autores contém 917 imagens, cada uma contendo uma célula e estão distribuídas de maneira não uniforme em 7 classes, três consideradas normais e quatro conside-

radas anormais.

A média harmônica da sensibilidade foi utilizada como forma de avaliar os resultados. Os algoritmos utilizados para a redução de dimensionalidade foram: PCA, *Kernel-PCA* (K-PCA) polinomial, K-PCA gaussiano, ISOMAP, LE e foram utilizados os classificadores *Spectral Clustering* e *Fuzzy C-means*.

Os resultados estão representados nas tabelas abaixo:

**Tabela 2.8: Contém a média harmônica da sensibilidade e a utilização de todas as características da imagem**

Algoritmos	Spectral Clustering	Fuzzy C-Means
Sem redução de dimensionalidade	74,21	72,89
PCA	74,25	74,23
K-PCA(polinomial)	85,78	74,24
K-PCA(gaussiano)	84,44	90,42
ISOMAP	76,61	75,02
LLE	86,97	81,69
Laplacian Eigenmaps	80,84	85,31

Fonte: (PLISSITI; NIKOU, 2012)

**Tabela 2.9: Contém a média harmônica da sensibilidade e a utilização das características apenas do núcleo**

Algoritmos	Spectral Clustering	Fuzzy C-Means
Sem redução de dimensionalidade	73,59	71,98
PCA	83,38	71,99
K-PCA(polinomial)	88,53	85,41
K-PCA(gaussiano)	87,52	90,58
ISOMAP	88,77	75,08
LLE	86,45	87,17
Laplacian Eigenmaps	87,52	87,20

Fonte: (PLISSITI; NIKOU, 2012)

No trabalho Karamatou et. al (DJIMA et al., 2015) os autores propõem a utilização do algoritmo LE para a redução de dimensionalidade e a utilização do algoritmo de classificação *Vectorized Matched Filtering* (VMF) em imagens da retina para prever anomalias.

Como comparação de classificação, foi utilizado o algoritmo *Overlapping-detections Matched Filtering* (OMF). Os resultados obtidos foram: Apenas OMF com 73% de acerto em detecção de anomalias, apenas VMF obteve 82%, LE com OMF obteve 50% e LE em conjunto

com VMF obteve 95%.

# Capítulo 3

## MATERIAIS E MÉTODOS

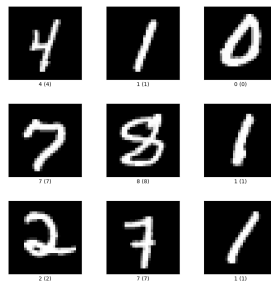
---

---

Nesse capítulo será discutido os materiais e métodos utilizados no trabalho.

### 3.1 MNIST

A base de dados MNIST (*Modified National Institute of Standards and Technology database*) contém imagens de números escritos a mão. Os dados obtidos nela são um subconjunto da base de dados NIST, nos quais as imagens tiveram seus tamanhos normalizados e foram centralizadas. A base contém 60 mil exemplos para treino e 10 mil exemplos para teste. A base de dados é utilizada em larga escala para treinar e testar algoritmos de aprendizado de máquina ou ainda para treinamento de sistemas de processamento de imagens. As imagens tem dimensões 28x28, ou seja, contém 784 atributos e contém números de 0 a 9, ou seja, contém 10 classes para a classificação.



**FIGURA 3.1 – Figura dos dígitos da base de dados MNIST**

Fonte: <https://www.tensorflow.org/datasets/catalog/mnist>

## 3.2 Sklearn digits

Essa base de dados é nativa da biblioteca *Scikit Learn*. Nela contém dígitos manuscritos em escala de cinza de imagens de tamanho 8x8, dessa forma há 64 atributos e os números da base de dados vão de 0 a 9, contendo 10 classes para a classificação. Nessa base, contém 1797 dígitos com aproximadamente 180 amostras por classe.

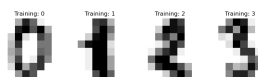


FIGURA 3.2 – Figura dos dígitos da base de dados do Scikit Learn

Fonte:

[https://scikit-learn.org/stable/auto\\_examples/classification/plot\\_digits\\_classification.html](https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html)

## 3.3 Plataforma computacional

Para a execução dos experimentos foi utilizado um computador com um Ryzen 5 2600 3.4GHz 6-Cores com 16 gigabytes de memória ram e uma placa gráfica GeForce GTX 1050 Ti de 4 gigabytes de RAM. O sistema operacional utilizado foi o Ubuntu 20.04. Foi utilizado a linguagem de programação Python e as bibliotecas utilizadas foram numpy, scipy e scikit learn.

## 3.4 Métricas para avaliação de desempenho

Nesta seção, serão discutidas as métricas quantitativas para a avaliação do desempenho dos métodos de classificação após a etapa de redução de dimensionalidade. Para isso, foram escolhidas duas medidas: o *Silhouette coefficient* e a acurácia da classificação, obtida a partir da matriz de confusão.

### 3.4.1 Silhouette coefficient

O *Silhouette coefficient* é um método que indica quão bom foram os agrupamentos criados (ROUSSEUW, 1987). A seguir, é apresentada uma breve explicação do porque esse índice é relevante na análise de agrupamentos. Seja  $C_i$  o  $i$ -ésimo agrupamento. Para cada ponto  $i \in C_i$  seja  $a(i)$  a média das distâncias entre  $i$  e todos os outros pontos do mesmo agrupamento  $C_i$ :

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j) \quad (3.1)$$

onde  $d(i, j)$  é a distância entre os pontos  $i$  e  $j$  dentro do agrupamento  $C_i$ . Então, a média das distâncias entre  $i$  e todos os outros pontos em  $C$  é a medida de dissimilaridade do ponto  $i$  ao agrupamento  $C$ . A menor média das distâncias entre o ponto  $i$  e todos os outros pontos no qual  $i$  não é membro pode ser definido como o valor de  $b(i)$ .

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \quad (3.2)$$

O valor de silhueta do ponto  $i$  é dado por:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1 \quad (3.3)$$

e

$$s(i) = 0, \text{ if } |C_i| = 1 \quad (3.4)$$

Combinando ambas definições, temos:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases} \quad (3.5)$$

É possível verificar que quando  $a(i) \ll b(i)$  o valor  $s(i) = 1$ , demonstra que o agrupamento foi bom. Quando  $b(i) \ll a(i)$  o valor  $s(i) = -1$ , significando que a atribuição poderia ser feito com outro grupo e quando  $b(i) = a(i)$  tendo o valor  $s(i) = 0$  significa que há sobreposição de grupos.

### 3.4.2 Acurácia da classificação

A matriz de confusão é uma matriz que contém informações de quais eram as classes originais e quais foram as predições feitas por um algoritmo. Seja  $a_{ij}$  um elemento da matriz, as linhas  $i$  indicam quais eram as classes originais da base de dados, enquanto as colunas  $j$  são os valores preditos. Dessa forma, se dividirmos a diagonal principal pela soma dos elementos da

sua respectiva linha, podemos obter a acurácia para cada classe do processo de classificação. Além disso, a acurácia total pode ser calculada somando todos os elementos da diagonal principal dividindo pela soma de todos os elementos da matriz.

## 3.5 Classificação supervisionada

A tarefa de classificação supervisionada tenta determinar a classe de um objeto utilizando suas características. Para essa tarefa então são criados os classificadores, que conforme os dados de treinamento criam modelos para tentar de maneira eficaz rotular objetos com características ainda não vistas. Deste modo, a classificação supervisionada é o ato de criar modelos aprendendo características de objetos já rotulados (Por seres humanos), a fim de conseguir prever um objeto com características ainda não vistas (LOOG, 2018). A seguir, segue uma breve descrição de cada algoritmo de classificação utilizado nesse trabalho.

### 3.5.1 Naive Bayes

O algoritmo de *Naive Bayes* utiliza a probabilidade de ocorrência de eventos para predição de um novo dado. O treinamento é o cálculo da probabilidade de cada evento/atributo ocorrer em cada classe. A predição então ocorre calculando a probabilidade da ocorrência de cada evento/atributo de um novo dado para cada classe, e a classe com maior probabilidade de ocorrência é atribuída ao dado (ZHANG, 2004).

### 3.5.2 KNN

O algoritmo de KNN utiliza os vizinhos mais próximos para definir a sua classe. O treinamento é basicamente a posição no espaço das amostras de treino. A classificação é feita calculando as distância entre o novo dado e cada amostra de treino e verificando os k vizinhos mais próximos. A classe majoritária entre os vizinhos mais próximos é atribuída ao novo dado (LOOG, 2018).

### 3.5.3 Decision Trees

O algoritmo *decision tree* utiliza uma estrutura de árvore para classificar. No treinamento (A criação da árvore), cada atributo é verificado e é criado um novo nó de maneira de melhor separar os dados entre as diferentes classes das amostras, então é repetido esse passo até que os



nós folha (Que não contém filhos) sejam criados e sejam pertencentes a apenas uma classe. A classificação de um novo dado ocorre comparando os atributos conforme percorre pelos nós da árvore, até chegar num nó folha, que o classifica (HART; STORK; DUDA, 2000).

### 3.5.4 Quadratic Discriminant Analysis

O algoritmo *Quadratic Discriminant Analysis* assume que os dados estão distribuídos de maneira normal multivariada. Então no treinamento são calculadas as médias e covariância dos dados para cada classe. Na classificação, calcula-se a probabilidade do novo dado e a classe com maior probabilidade é atribuída ao novo dado (GHOJOGH; CROWLEY, 2019).

### 3.5.5 Multilayer Perceptron

O algoritmo *Multilayer Perceptron* é a abordagem de redes neurais. Redes neurais podem ser colocadas interligadas para criar a estrutura do *Multilayer Perceptron*. A cada camada da rede neural contém a entrada, pesos para cada entrada e em seguida, uma camada de ativação por neurônio que pode ser ativada conforme a entrada e seus pesos. No treinamento, são inicializados os dados de entrada na rede neural e então verifica-se a saída da última camada da rede neural, caso o valor seja discrepante com a função de perda é utilizado um método denominado *Backward propagation* que modifica os parâmetros das camadas das redes neurais, ajustando para que haja a convergência. A classificação é feita adicionando os atributos do novo dado como entrada da rede neural que então o classifica (HART; STORK; DUDA, 2000).

### 3.5.6 Random Forest

O algoritmo *Random Forest* consiste em criar múltiplas *decision trees* para aumentar a acurácia do método. No treinamento, no primeiro passo é criada uma nova base de dados, escolhendo aleatoriamente dados da base de dados original. O segundo passo consiste em criar uma *decision tree* como já descrito anteriormente na seção 3.5.3. Esses passos são repetidos uma quantidade arbitrária de vezes até construir a *Random Forest*. Para classificar, utiliza-se todas as árvores criadas colocando o novo dado como entrada, cada classificação é contabilizada e a classe majoritária atribui a classe para o novo dado (KULKARNI; SINHA, 2012).

# Capítulo 4

## RESULTADOS E DISCUSSÃO

---

---

Nessa seção é apresentado os resultados experimentais do trabalho, da execução dos algoritmos apresentados anteriormente com diferentes parâmetros de entrada.

Na primeira etapa, utiliza-se os algoritmos de redução de dimensionalidade nos dados (ISOMAP, LLE e *Laplacian Eigenmaps*), diminuindo a dimensionalidade dos dados para as dimensões  $d$  variando de 2 até 10. No segundo passo, são executadas as classificações utilizando os algoritmos de classificação já discutidos, sendo que a quantidade de amostras foram 50% para treino e 50% para teste. Após a classificação, são obtidas as acurácias entre os algoritmos de forma de ser possível a discussão e a comparação dos dados. Essas etapas foram feitas tanto para a base de dados *Sklearn digits* quanto para a base de dados MNIST, no entanto, para a base MNIST foi amostrado apenas 10% dos dados da sua base, para diminuir a quantidade de amostras de treino e teste. Além disso, é computado também o *Silhouette coefficient* para verificar quão bom foi a separação das classes após a redução de dimensionalidade. Foi utilizado o valores de  $k = 20$  para o valor de k-vizinhos mais próximos para os algoritmos de redução de dimensionalidade.

### 4.1 Sklearn digits

Nessa seção, serão discutidos os dados obtidos pelo processo definido anteriormente para a base de dados *Sklearn digits*.

A tabela 4.1 mostra a acurácia de classificação pelos algoritmos descritos anteriormente quando algoritmo de redução de dimensionalidade é o ISOMAP. É notável que a média de acurácia cresce conforme aumenta-se a quantidade de dimensões, sendo que a maior acurácia foi atingida pelo MLP com 10 dimensões, sendo ela 95,6%.

**Tabela 4.1: Contém informações sobre as acurácias de classificação utilizando o ISOMAP**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,736	0,849	0,913	0,925	0,934	0,942	0,949	0,95	0,948
NB	0,680	0,783	0,827	0,852	0,835	0,839	0,882	0,881	0,894
DT	0,651	0,765	0,869	0,856	0,855	0,874	0,889	0,871	0,881
QDA	0,701	0,807	0,882	0,895	0,909	0,909	0,93	0,931	0,935
MLP	0,753	0,818	0,899	0,916	0,928	0,933	0,939	0,941	0,956
RF	0,731	0,825	0,892	0,903	0,908	0,924	0,933	0,924	0,935
Média	0,709	0,808	0,880	0,891	0,895	0,904	0,92	0,916	0,925

**Tabela 4.2: Contém informações sobre as acurácias de classificação utilizando o LLE**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,595	0,715	0,765	0,786	0,829	0,839	0,838	0,841	0,857
NB	0,382	0,492	0,586	0,571	0,649	0,731	0,751	0,749	0,765
DT	0,555	0,661	0,686	0,722	0,752	0,761	0,775	0,789	0,788
QDA	0,567	0,664	0,728	0,785	0,839	0,843	0,86	0,882	0,898
MLP	0,436	0,566	0,638	0,657	0,699	0,709	0,732	0,746	0,754
RF	0,597	0,712	0,773	0,799	0,842	0,853	0,853	0,861	0,863
Média	0,522	0,635	0,696	0,720	0,769	0,789	0,801	0,811	0,821

A tabela 4.2 mostra a acurácia de classificação quando o algoritmo de redução de dimensionalidade utilizado é o LLE. Pode-se notar que a média de acurácias do modelo utilizando o LLE são as menores, atingindo a maior acurácia pelo algoritmo QDA com o seu valor sendo 89,8% com 10 dimensões.

**Tabela 4.3: Contém informações sobre as acurácias de classificação utilizando o Laplacian Eigenmaps**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,829	0,899	0,924	0,931	0,941	0,943	0,939	0,939	0,94
NB	0,685	0,738	0,794	0,824	0,843	0,847	0,877	0,888	0,887
DT	0,786	0,879	0,906	0,921	0,928	0,934	0,931	0,941	0,94
QDA	0,708	0,859	0,877	0,911	0,929	0,936	0,939	0,934	0,939
MLP	0,099	0,091	0,507	0,582	0,680	0,764	0,772	0,786	0,889
RF	0,814	0,899	0,922	0,934	0,944	0,939	0,949	0,943	0,948
Média	0,653	0,728	0,822	0,850	0,878	0,894	0,901	0,905	0,924

A tabela 4.3 mostra as acurácias de classificação quando o algoritmo de redução de dimensionalidade utilizado foi o *Laplacian Eigenmaps*. Nela, pode-se notar que a média das acurácias também cresce conforme o número de dimensões, sendo que o seu maior valor é obtido

quando há 8 dimensões e o algoritmo de classificação é o RF, pontuando 94,9% de acurácia.

**Tabela 4.4: Contém informações sobre o *Silhouette coefficient***

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
ISOMAP	0,172	0,286	0,305	0,310	0,297	0,292	0,292	0,284	0,29
LLE	-0,038	0,046	0,082	0,072	0,093	0,087	0,094	0,076	0,11
LE	0,268	0,3371	0,412	0,464	0,507	0,512	0,489	0,476	0,501

A tabela 4.4 mostra quais foram as pontuações do *Silhouette coefficient* para cada dimensão escolhida. Nessa tabela, pode-se notar que os menores valores pontuados foram os do algoritmo LLE tendo o seu menor valor com dimensão igual a 2. Além disso, os maiores valores da tabela são os do algoritmo LE, pontuando 0,512 para  $D = 7$ .

## 4.2 MNIST

Nessa seção serão mostrados os resultados obtidos pelo processo descrito anteriormente para a base de dados MNIST.

**Tabela 4.5: Contém informações sobre as acurácias de classificação utilizando o ISOMAP**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,451	0,596	0,721	0,797	0,821	0,843	0,862	0,863	0,873
NB	0,465	0,529	0,634	0,702	0,725	0,764	0,791	0,801	0,811
DT	0,377	0,508	0,634	0,712	0,74	0,753	0,761	0,771	0,785
QDA	0,479	0,585	0,716	0,785	0,815	0,842	0,858	0,857	0,867
MLP	0,482	0,616	0,736	0,787	0,813	0,842	0,86	0,862	0,871
RF	0,424	0,594	0,728	0,799	0,823	0,853	0,868	0,872	0,883
Média	0,446	0,571	0,695	0,764	0,789	0,816	0,833	0,838	0,848

Na tabela 4.5 mostra os dados obtidos das acurácias na base MNIST quando o algoritmo de redução de dimensionalidade aplicado é o ISOMAP. Pode-se notar um grande crescimento de acurácia para  $d = 2$  até  $d = 4$ , e depois o crescimento diminui. Além disso, o maior valor de acurácia é para a técnica de classificação RF, obtendo 88,3% com  $d = 10$ .

Na tabela 4.6 mostra as acurácias obtidas das classificações quando o algoritmo de redução de dimensionalidade aplicado é o LLE. Pode-se notar que as acurácias utilizando esse método são mais baixas que os vistos na tabela 4.5, sendo a maior acurácia obtida pelo método RF com  $d = 10$ , sendo o seu valor igual a 76,6%.

**Tabela 4.6: Contém informações sobre as acurácias de classificação utilizando o LLE**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,448	0,538	0,586	0,643	0,648	0,675	0,694	0,731	0,751
NB	0,269	0,285	0,367	0,498	0,528	0,524	0,516	0,558	0,578
DT	0,382	0,485	0,568	0,586	0,601	0,605	0,626	0,659	0,671
QDA	0,34	0,319	0,395	0,507	0,506	0,524	0,553	0,593	0,607
MLP	0,119	0,119	0,456	0,531	0,563	0,618	0,613	0,627	0,656
RF	0,44	0,557	0,636	0,665	0,688	0,703	0,718	0,757	0,766
Média	0,333	0,384	0,501	0,571	0,589	0,608	0,62	0,654	0,671

**Tabela 4.7: Contém informações sobre as acurácias classificação utilizando o Laplacian Eigenmaps**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
KNN	0,693	0,735	0,762	0,813	0,831	0,843	0,846	0,845	0,853
NB	0,631	0,637	0,66	0,714	0,749	0,751	0,765	0,755	0,758
DT	0,598	0,657	0,677	0,761	0,773	0,775	0,795	0,797	0,799
QDA	0,614	0,638	0,642	0,708	0,752	0,753	0,75	0,752	0,757
MLP	0,119	0,119	0,593	0,679	0,713	0,704	0,735	0,735	0,78
RF	0,678	0,733	0,756	0,82	0,839	0,848	0,865	0,865	0,862
Média	0,555	0,586	0,682	0,749	0,776	0,779	0,793	0,792	0,802

Na tabela 4.7 contém as acurácias obtidas pelas classificações quando o método de redução de dimensionalidade aplicado é o LE. Nessa base de dados, o classificador com maior acurácia também foi o RF, empatando em acurácia de 86,5% para  $d = 8$  e  $d = 9$

**Tabela 4.8: Contém informações sobre o Silhouette coefficient**

Algoritmo	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
ISOMAP	-0,066	-0,027	0,0541	0,0656	0,0749	0,099	0,103	0,098	0,1
LLE	-0,289	-0,305	-0,156	-0,0748	-0,071	-0,054	-0,056	-0,033	-0,028
LE	0,132	0,137	0,0807	0,0866	0,107	0,093	0,101	0,0947	0,0905

Os valores da *Silhouette coefficient* ficaram em maior parte em torno de 0,1 para o ISOMAP e LE e valores negativos próximos de 0 para o LLE para  $d > 4$ . Os piores valores obtidos para o agrupamento foram os do LLE para dimensão de 1 a 3.

## 4.3 Wilcoxon

Para comparar o desempenho dos métodos de modo estatístico, foi aplicado o teste não paramétrico de Wilcoxon, utilizando como base as tabelas de acurácia de classificação das bases de dados digits e MNIST. Para isso, foi concatenado cada linha da tabela, formando um vetor de acurácias para cada tabela. Em seguida, o teste Wilcoxon foi utilizado para cada dupla de vetores (ISOMAP vs LLE, ISOMAP vs LE, LLE vs LE). Os resultados obtidos estão na tabela:

**Tabela 4.9: Contém informações sobre o teste de Wilcoxon para a base MNIST**

Algoritmo	Soma do rank	p-value
ISOMAP vs LLE	5,0	$2,1506^{-10}$
ISOMAP vs LE	690,0	0,8214
LE vs LLE	0,0	$3,4980^{-10}$

Após aplicar o teste com um nível de significância  $\alpha = 0.05$ , é possível concluir que o método ISOMAP é estatisticamente superior ao método LLE e que o método *Laplacian Eigenmaps* também é estatisticamente superior ao LLE. Porém, não é possível concluir que existam diferenças entre os métodos ISOMAP e *Laplacian Eigenmaps* na base de dados MNIST. Sendo assim, a recomendação consiste em utilizar o algoritmos ISOMAP ou o algoritmo *Laplacian Eigenmaps* nessa base.

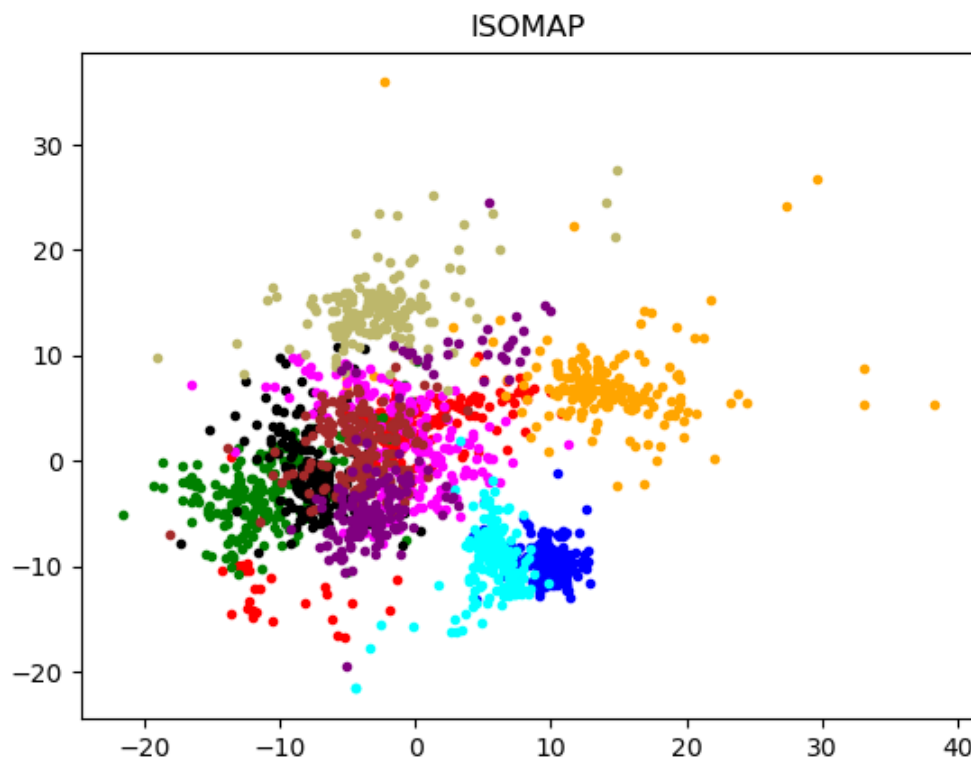
**Tabela 4.10: Contém informações sobre o teste de Wilcoxon para a base Sklearn Digits**

Algoritmo	Soma do rank	p-value
ISOMAP vs LLE	0,0	$1,621^{-10}$
ISOMAP vs LE	597,5	0,2118
LE vs LLE	140,0	$2,128^{-7}$

Assim como ocorreu no teste anterior, o método ISOMAP é estatisticamente superior ao método LLE e o método *Laplacian Eigenmaps* também é estatisticamente superior ao LLE. Como a hipótese foi rejeitada entre a comparação do ISOMAP e *Laplacian Eigenmaps*, não é possível afirmar que existam diferenças entre os métodos na base *Sklearn digits*. Dessa forma, recomenda-se utilizar os algoritmos ISOMAP ou o algoritmo *Laplacian Eigenmaps* nessa base.

## 4.4 Visualização de Dados

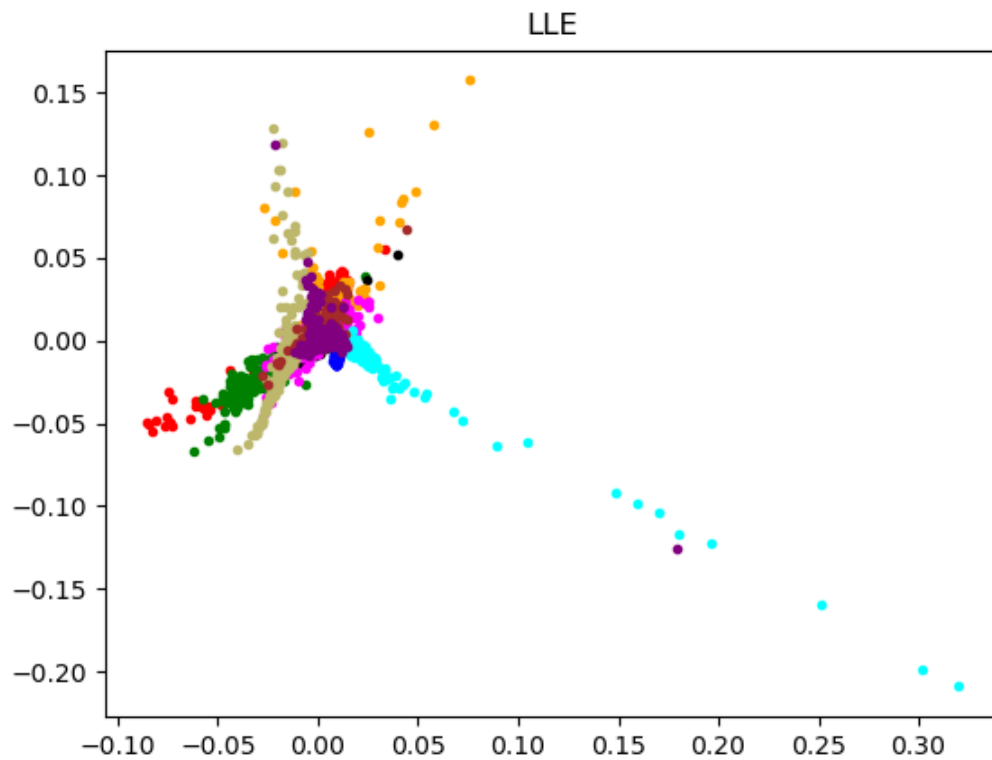
Nessa seção será apresentado a distribuição dos dados das classes numa imagem, sendo que cada cor indica uma classe diferente. Foi utilizado a redução de dimensionalidade para  $d = 2$  para os algoritmos ISOMAP, LLE e *Laplacian Eigenmaps*, para conseguir traçar o gráfico num plano cartesiano.



**FIGURA 4.1** – Figura que mostra a distribuição em 2d das classes quando utilizado o ISOMAP na base *Sklearn digits*

Fonte: O Autor

É possível verificar na imagem 4.1 que há certa divisão das classes, no entanto, a sobreposição ainda é grande entre muitas classes. Visualmente, parece que foi o algoritmo que melhor separou as diferentes classes.

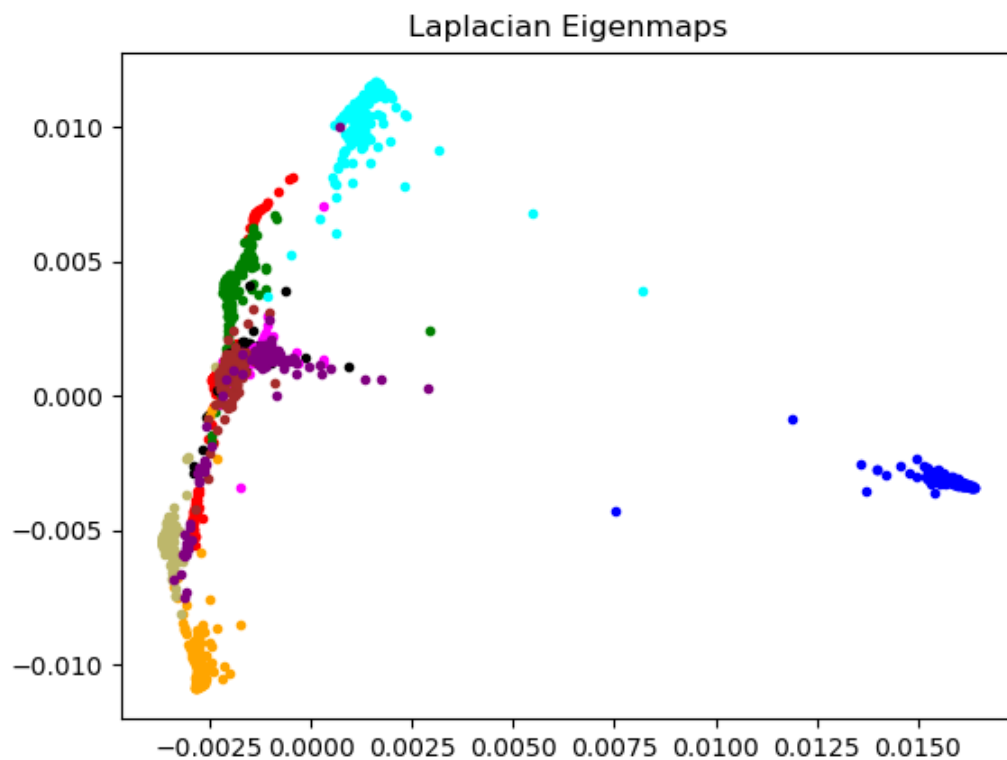


**FIGURA 4.2** – Figura que mostra a distribuição em 2d das classes quando utilizado o LLE na base *Sklearn digits*

Fonte: O Autor

Na figura 4.2 nota-se que não há separações entre classes, todas as classes permanecem aglomeradas perto da região do (0,0).

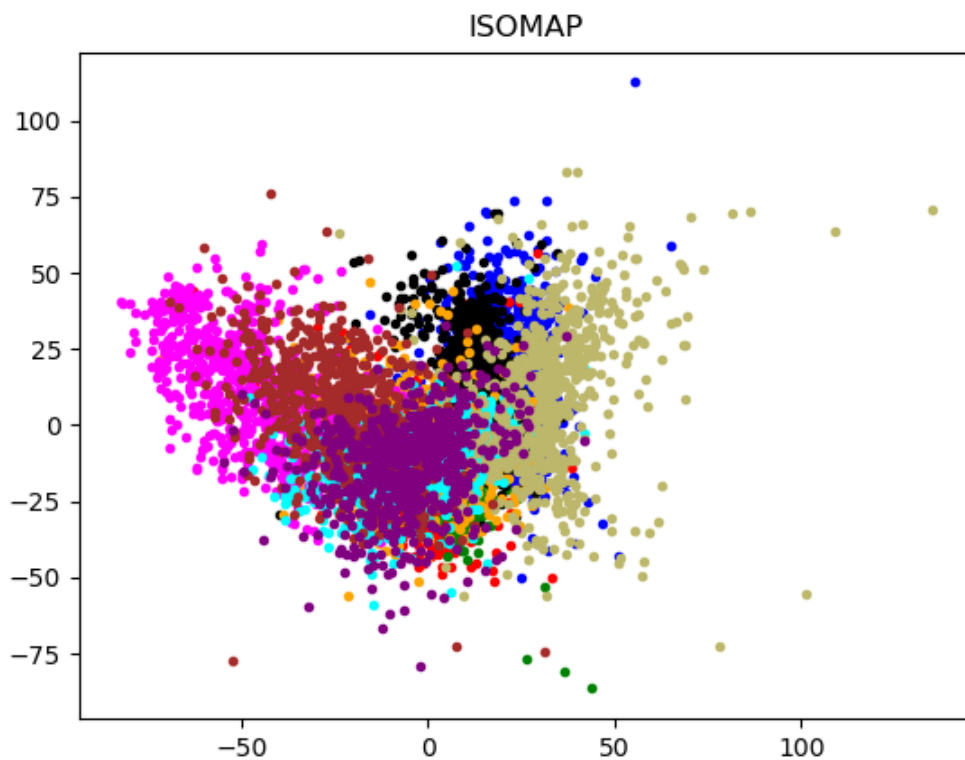




**FIGURA 4.3** – Figura que mostra a distribuição em 2d das classes quando utilizado o *Laplacian eigenmaps* na base *Sklearn digits*

Fonte: O Autor

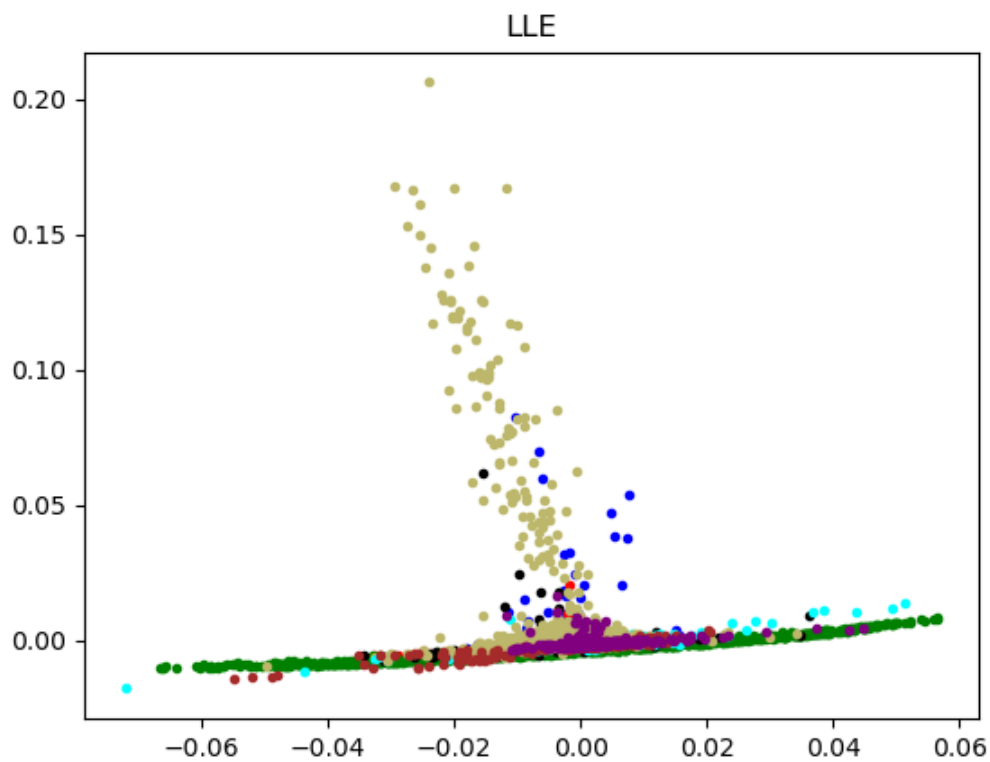
É possível notar na figura 4.2 que há uma separação entre algumas classes, no entanto, a maioria das classes ainda permanecem aglomeradas.



**FIGURA 4.4** – Figura que mostra a distribuição em 2d das classes quando utilizado o ISOMAP na base MNIST

Fonte: O Autor

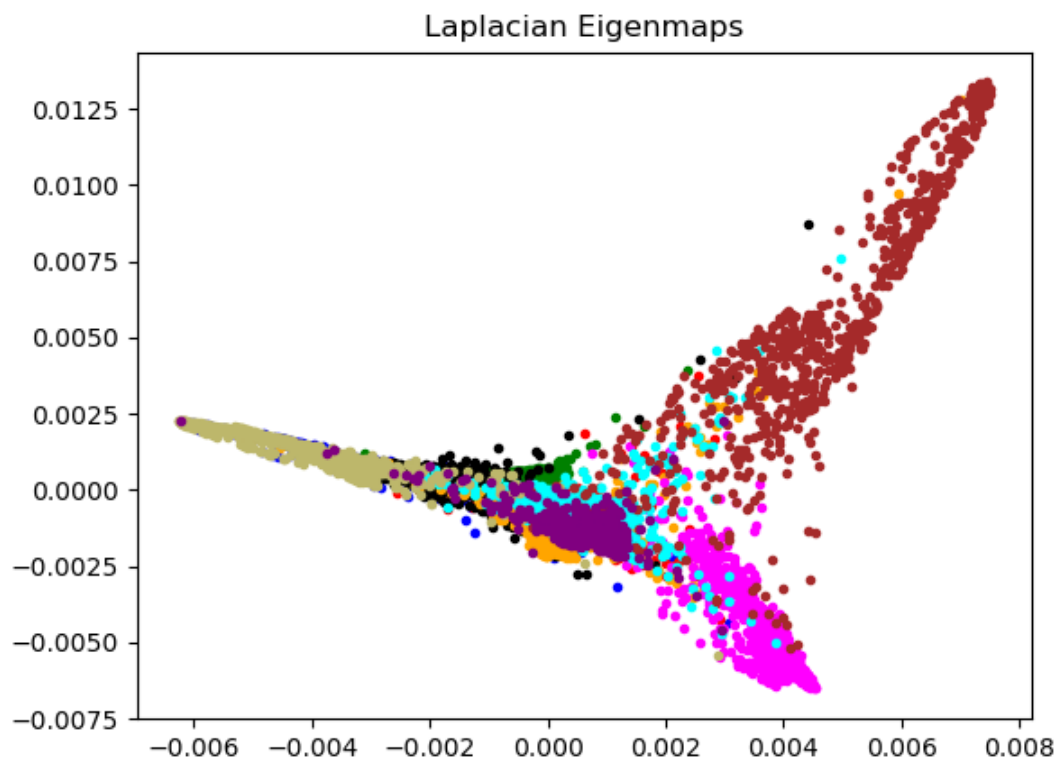
É possível notar na figura 4.4, que o padrão de separação ocorrido na base *Sklearn digits* ocorre nessa base também, há certa separação entre algumas classes, mas há sobreposição entre a maioria das classes.



**FIGURA 4.5** – Figura que mostra a distribuição em 2d das classes quando utilizado o LLE na base MNIST

Fonte: O Autor

Na figura 4.5, é possível notar que a maioria das classes ficam aglomeradas na região do 0 no eixo y.



**FIGURA 4.6** – Figura que mostra a distribuição em 2d das classes quando utilizado o *Laplacian Eigenmaps* na base MNIST

Fonte: O Autor

Na figura 4.6 é possível verificar que há a separação de algumas classes, no entanto, a maioria continuam aglomeradas.

# Capítulo 5

## CONCLUSÃO

---

---

A classificação é uma tarefa muito utilizada no aprendizado de máquina. Encontrar meios para aumentar a acurácia na classificação é uma tarefa árdua. Assim, bases de dados com muitos atributos são difíceis de classificar devido a maldição da dimensionalidade. Para isso, a utilização de algoritmos de redução de dimensionalidade é um bom caminho para tratar desse problema. A seguir escrevo sobre a estratégia utilizada pelo trabalho e sobre a conclusão de seus resultados.

Neste trabalho, o objetivo principal foi realizar uma comparação objetiva entre diversos métodos de redução de dimensionalidade não linear para o aprendizado não supervisionado de métricas em problemas de reconhecimento de dígitos manuscritos. A pergunta fundamental que motivou o desenvolvimento da pesquisa foi: há diferenças significativas na classificação de tais padrões ao utilizar os algoritmos ISOMAP, LLE e *Laplacian Eigenmaps* como extratores de características? Foram propostos experimentos computacionais com diversos classificadores supervisionados de modo a obter uma conclusão acerca do problema em questão.

Foram feitas a redução de dimensionalidade e a classificação e os melhores resultados obtidos na base *Sklearn digits* foi pelos algoritmos ISOMAP com 10 dimensões e classificador *Multilayer Perceptron*, obtendo 95,2% de acurácia, a maior acurácia obtida pelo algoritmo *Laplacian Eigenmaps* foi com 8 dimensões e classificador RF obtendo 94,9% de acurácia e por fim, a maior acurácia obtida pelo LLE foi com 10 dimensões e utilizando o classificador QDA obtendo 89,8% de acurácia.

Também foi utilizada a base MNIST para os testes. O maior valor de acurácia obtido foi utilizando o algoritmo ISOMAP com 10 dimensões em conjunto com o algoritmo RF obtendo 88,3%, a maior acurácia obtida pelo *Laplacian Eigenmaps* foi com 8 ou 9 dimensões (houve empate) em conjunto com o algoritmo RF obtendo 86,5%, e por último a maior acurácia obtida

com o LLE foi com 10 dimensões em conjunto com RF obtendo 76,6% de acurácia.

O *Silhouette coefficient* foi utilizado na base *Sklearn digits* para medir a qualidade dos agrupamentos após a redução de dimensionalidade para a dimensão de 2 a 10. Foi verificado que o algoritmo *Laplacian Eigenmaps* obteve o melhor valor de 0,501 para 10 dimensões, seguido pelo ISOMAP que obteve 0,310 para 5 dimensões e em último lugar o LLE que obteve 0,094 com 8 dimensões.

O *Silhouette coefficient* foi utilizado também na base MNIST e na maioria dos casos o valor ficava muito próximo a 0, indicando que os agrupamentos não tem uma região bem definida, havendo muita sobreposição entre as classes.

Por fim, foi utilizado o algoritmo de Wilcoxon para a verificação estatística das acurácias obtidas. Nele, foi possível verificar que o ISOMAP é superior ao LLE e o *Laplacian Eigenmaps* é superior que o LLE, porém nada pode se afirmar da comparação do *Laplacian Eigenmaps* e o ISOMAP. Esse resultado foi obtido tanto nas acurácias da base do *Sklearn digits* quanto na base MNIST

## REFERÊNCIAS

---

---

- ANOWAR, F.; SADAQUI, S.; SELIM, B. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, Elsevier, v. 40, p. 100378, 2021.
- BELKIN, M.; NIYOGI, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Nips*. [S.l.: s.n.], 2001. v. 14, n. 14, p. 585–591.
- BU, Y.; CHEN, F.; PAN, J. Stellar spectral subclasses classification based on isomap and svm. *New Astronomy*, Elsevier, v. 28, p. 35–43, 2014.
- DJIMA, K. A. Y. et al. Detection of anomaly in human retina using laplacian eigenmaps and vectorized matched filtering. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2015: Image Processing*. [S.l.], 2015. v. 9413, p. 94132F.
- DU, M.; ZHANG, S.; WANG, H. Supervised isomap for plant leaf image classification. In: HUANG, D.-S. et al. (Ed.). *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 627–634. ISBN 978-3-642-04020-7.
- GHOJOGH, B.; CROWLEY, M. Linear and quadratic discriminant analysis: Tutorial. *arXiv preprint arXiv:1906.02590*, 2019.
- GISBRECHT, A.; HAMMER, B. Data visualization by nonlinear dimensionality reduction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 5, n. 2, p. 51–73, 2015.
- GU, R.-j.; XU, W.-b.; YE, B. Imd-isomap for data visualization and classification. In: *2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID)*. [S.l.: s.n.], 2007. p. 148–151.
- HART, P. E.; STORK, D. G.; DUDA, R. O. *Pattern classification*. [S.l.]: Wiley Hoboken, 2000.
- HE, P. et al. Nonlinear manifold classification based on lle. In: BHATIA, S. K. et al. (Ed.). *Advances in Computer Communication and Computational Sciences*. Singapore: Springer Singapore, 2019. p. 227–234. ISBN 978-981-13-0341-8.
- HUANG, M. et al. Analysis of hyperspectral scattering images using locally linear embedding algorithm for apple mealiness classification. *Computers and Electronics in Agriculture*, v. 89, p. 175–181, 2012. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169912002220>.

- KULKARNI, V. Y.; SINHA, P. K. Pruning of random forest classifiers: A survey and future directions. In: IEEE. *2012 International Conference on Data Science & Engineering (ICDSE)*. [S.l.], 2012. p. 64–68.
- LASHGARI, E.; DEMIRCAN, E. Electromyography pattern classification with laplacian eigenmaps in human running. *International Journal of Electronics and Communication Engineering*, v. 11, n. 4, p. 412–417, 2017.
- LEON-MEDINA, J. X. et al. Application of manifold learning algorithms to improve the classification performance of an electronic nose. In: *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. [S.l.: s.n.], 2020. p. 1–6.
- LOOG, M. Supervised classification: Quite a brief overview. *Machine Learning Techniques for Space Weather*, Elsevier, p. 113–145, 2018.
- LORIGO, L.; GOVINDARAJU, V. Offline arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 5, p. 712–724, 2006.
- MAATEN, L. V. D. et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, v. 10, n. 66-71, p. 13, 2009.
- MEMON, J. et al. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, IEEE, v. 8, p. 142642–142668, 2020.
- MORI, S.; SUEN, C.; YAMAMOTO, K. Historical review of ocr research and development. *Proceedings of the IEEE*, v. 80, n. 7, p. 1029–1058, 1992.
- PARK, H. et al. Isomap induced manifold embedding and its application to alzheimer’s disease and mild cognitive impairment. *Neuroscience Letters*, Elsevier, v. 513, n. 2, p. 141–145, 2012.
- PLISSITI, M. E.; NIKOU, C. Cervical cell classification based exclusively on nucleus features. In: SPRINGER. *International Conference Image Analysis and Recognition*. [S.l.], 2012. p. 483–490.
- PRABHAKAR, S. K.; RAJAGURU, H. Factor analysis, hessian local linear embedding and isomap for epilepsy classification from eeg. In: *2016 Electrical Engineering Conference (EECon)*. [S.l.: s.n.], 2016. p. 19–24.
- RADUCANU, B.; DORNAIKA, F. Dynamic facial expression recognition using laplacian eigenmaps-based manifold learning. In: *2010 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2010. p. 156–161.
- RAJAGURU, H.; PRABHAKAR, S. K. Performance analysis of local linear embedding (lle) and hessian lle with hybrid abc-pso for epilepsy classification from eeg signals. In: *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. [S.l.: s.n.], 2018. p. 1084–1088.
- ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, Elsevier, v. 20, p. 53–65, 1987.
- ROWEIS, S. T.; SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, American Association for the Advancement of Science, v. 290, n. 5500, p. 2323–2326, 2000.



- TENENBAUM, J. B.; SILVA, V. D.; LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, American Association for the Advancement of Science, v. 290, n. 5500, p. 2319–2323, 2000.
- WANG, G.; DING, G. Face recognition using kfda-lle. In: HUANG, D.-S. et al. (Ed.). *Advanced Intelligent Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 420–425. ISBN 978-3-642-24728-6.
- WANG, W.-y. et al. Automatic classification for wdms with isomap and svm. In: *2015 IEEE International Conference on Information and Automation*. [S.l.: s.n.], 2015. p. 1409–1413.
- WILKINSON, R. A. *The first census optical character recognition system conference*. [S.l.]: US Department of Commerce, National Institute of Standards and Technology, 1992. v. 4912.
- WOLF, C.; JOLION, J.-M.; CHASSAING, F. Text localization, enhancement and binarization in multimedia documents. In: IEEE. *Object recognition supported by user interaction for service robots*. [S.l.], 2002. v. 2, p. 1037–1040.
- YING, Z. et al. Facial expression recognition with local binary pattern and laplacian eigenmaps. In: SPRINGER. *International Conference on Intelligent Computing*. [S.l.], 2009. p. 228–235.
- YU, Z. et al. Stock price forecasting based on lle-bp neural network model. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 553, p. 124197, 2020.
- ZHANG, H. The optimality of naive bayes. *AA*, v. 1, n. 2, p. 3, 2004.