



Logotipo, nome da empresa

Descrição gerada automaticamente

MACHINE LEARNING II - TCC

EFEITOS DO PRÉ-PROCESSAMENTO DE DADOS PARA MÁQUINAS DE APRENDIZADO

Sumário

[1. Introdução 3](#_Toc143436515)

[2. Conceitos básicos 3](#_Toc143436516)

[2.1. Técnicas de pré-processamento 3](#_Toc143436517)

[2.2. Algoritmos de Aprendizagem de Máquina 3](#_Toc143436518)

[3. Metodologia dos experimentos 3](#_Toc143436519)

[3.1. Bancos de dados 3](#_Toc143436520)

[3.2. Métricas 5](#_Toc143436521)

[4. Resultados 7](#_Toc143436522)

[4.1. Dados sem pré-processamento 8](#_Toc143436523)

[4.1.1. KNN 8](#_Toc143436524)

[4.1.2. DT 8](#_Toc143436525)

[4.1.3. MLP 8](#_Toc143436526)

[4.1.4. NB 8](#_Toc143436527)

[4.1.5. SVM 8](#_Toc143436528)

[4.2. Resultados após pré-processamento 8](#_Toc143436529)

[4.2.1. KNN 8](#_Toc143436530)

[4.2.2. DT 9](#_Toc143436531)

[4.2.3. MLP 9](#_Toc143436532)

[4.2.4. NB 9](#_Toc143436533)

[4.2.5. SVM 9](#_Toc143436534)

[5. Conclusões 9](#_Toc143436535)

[6. Referências 9](#_Toc143436536)

# Introdução

O projeto em questão visa avaliar os efeitos da utilização de técnicas de pré-processamento de dados aos resultados de diferentes máquinas de aprendizado, antes e após sua aplicação.

Os dados utilizados para esse estudo são oriundos do portal OpenML, relativos ao artigo "*Letter Recognition Using Holland-style Adaptive Classifiers",* de Peter Frey e David Slate, publicado em 1991 (Machine Learning, 6, 161-182). Tais observações foram analisadas pelas máquinas de aprendizado k-Nearest Neighbour (KNN), Decision Tree (DT), Multilayer Perceptron (MLP), Naive Bayes (NB) e Support Vector Machine (SVM) com respectivos resultados comparados entre si.

# Conceitos básicos

# Técnicas de pré-processamento

XXXXX

# Algoritmos de Aprendizagem de Máquina

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

KNN: O KNN é muito utilizado em problemas de classificação, e felizmente é um dos algoritmos de machine learning mais fáceis de se compreender. Em resumo, o KNN tenta classificar cada amostra de um conjunto de dados avaliando sua distância em relação aos vizinhos mais próximos.

DT: Uma árvore de decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Isto é, pode ser usado para prever categorias discretas (sim ou não, por exemplo) e para prever valores numéricos (o valor do lucro em reais).

MLP: O perceptron multi-camadas ou (MLP – MultiLayer Perceptron), é um algoritmo de redes neurais artificiais que tem diversas aplicações. Por isso, esse artigo trata do assunto, descrevendo o algoritmo e visando facilitar o aprendizado do mesmo. O perceptron multi-camadas ou (MLP – MultiLayer Perceptron), é um algoritmo de redes neurais artificiais que tem diversas aplicações. Por isso, esse artigo trata do assunto, descrevendo o algoritmo e visando facilitar o aprendizado do mesmo.

Algoritmos KNN, Decision Tree, MLP e SVM foram utilizados para análise e geração

dos modelos. Tais algoritmos foram selecionados devido aos seus modelos

matemáticos serem de certa forma assimilares, trazendo uma perspectivas diferentes

para os modelos, como por exemplo, análise por clusterização, análise de variância,

# Metodologia dos experimentos

# Bancos de dados

Seguindo os requisitos estabelecidos pela UFPE (Universidade Federal de Pernambuco) para esse projeto, o *Dataset* abaixo foi selecionado:

* Descrição: Identificação de letras maiúsculas do alfabeto inglês, com base em avaliação de pixels ativos de uma imagem, ou seja, os pixels preenchidos que formam as letras analisadas;
* Características: 17 *features*, 26 classes e 20000 instâncias. As classes são compostas pelas letras do alfabeto em inglês, de A até Z.
* URL: <https://www.openml.org/search?type=data&status=active&id=6>.

Abaixo estão inclusos os atributos do *Dataset* utilizado, contendo o nome da *feature* entre parêntesis seguida da sua respectiva descrição:

1. (x-box) A posição horizontal, contando os pixels da borda esquerda da imagem, do centro da menor caixa retangular que pode ser desenhada contendo os pixels “ativos” dentro da caixa;
2. (y-box) A posição vertical, contando os pixels da parte de baixo da imagem até a caixa acima;
3. (width) A largura, em pixels, da caixa;
4. (high) A altura, em pixels, da caixa;
5. (onpix) O número total de pixels ativos na imagem analisada;
6. (x-bar) A posição horizontal média de todos os pixels ativos em relação ao centro da caixa dividida pela largura da caixa. Esta característica tem um valor negativo quando a distribuição dos pixels da letra em questão é desbalanceada, como no caso da representação da letra “L”;
7. (y-bar) A posição vertical média de todos os pixels ativos em relação ao centro da caixa dividida pela altura da caixa;
8. (x2bar) O valor médio, ao quadrado, das distâncias de pixel horizontais conforme medido no tópico 6 (x-bar). Este atributo terá um valor maior para imagens cujos pixels estejam mais separados na direção horizontal como seria o caso das letras W ou M;
9. (y2bar) O valor médio, ao quadrado, das distâncias de pixel vertical conforme medido no tópico 7 (y-bar);
10. (xybar) O produto médio das distâncias horizontal e vertical para cada pixel ativo, conforme medido nos tópicos 6 e 7. Este atributo tem um valor positivo para linhas diagonais que vão da parte inferior esquerda para a parte superior direita e um valor negativo para linhas diagonais da parte superior esquerda para a parte inferior direita;
11. (x2ybr) O valor médio da distância horizontal ao quadrado vezes a distância vertical para cada pixel ativo. Isso mede a correlação da variação horizontal com a posição vertical;
12. (xy2br) O valor médio da distância vertical ao quadrado vezes a distância horizontal para cada pixel ativo. Isso mede a correlação da variação vertical com a posição horizontal;
13. (x-ege) O número médio de arestas (um pixel ativo imediatamente à direita de um pixel inativo ou do limite da imagem) encontrado ao fazer varreduras sistemáticas da esquerda para a direita em todas as posições verticais dentro da caixa. Esta medida permite distinguir letras como "W" ou "M" e letras como “T” ou "L";
14. (xegvy) A soma das posições verticais das arestas encontradas conforme medido no tópico 13 (x-ege). Esse recurso dará um valor maior se houver mais arestas na parte superior da caixa, como na letra "Y";
15. (y-ege) O número médio de arestas (um pixel ativo imediatamente acima de um pixel inativo ou do limite da imagem) encontrado ao fazer varreduras sistemáticas da imagem de baixo para cima nas posições horizontais gerais dentro da caixa.
16. (yegvx) A soma das posições horizontais das arestas encontradas, conforme medido no tópico 15 (y-ege);
17. (class) Identificação da letra do alfabeto correspondente às medições acima citadas.

# Métricas

XXXXX Requisitos

area under the ROC curve, f1-score, accuracy, matriz de confusão média

Levando em consideração os pontos acima, foi realizada a técnica de K-fold com

valores 3, 5 e 10 para o dataset de treinamento e teste para todos os modelos

treinados e estudados.

Utilizada técnicas de mini-batch para o cálculo entre as diversas máquinas para

melhor resultado performance durante o processamento dos modelos e

interatividade do Grid Search CV via Google Colab.

**3.2. Métricas.**

Pela hipótese de separação de grãos, e devido a análise “impacto versus efeito”, foi

utilizado como métrica para medição, e critério de seleção entre os modelos, a

precisão média. De todos os dados classificados como positivos, quantos realmente

são positivos, assumindo a premissa que o modelo trabalhará na divisão de grãos,

daremos maior importância para o TP(“True Positive”).

Foi notada uma forte correlação de dados com formação de clusters entre suas

variáveis e a separação das classes, algumas classes apresentam características muito

próximas deixando seus agrupamentos similares a de outras classes, porém em um

hiperplano pode-se ter um entendimento diferente, assim veremos nos resultados dos

modelos aplicados e seus valores de “p”.

Devido a quantidade de features presentes no dataset, foram utilizados “plots” de

features aleatórias em segundo plano (2D) para fins de representatividade, nota-se que

em diversos outros “plots” foram obtidos resultados similares durante a preparação e

interpretação dos dados.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Para trabalhar nos modelos propostos foi realizada uma análise de significância

dos dados e verificada a necessidade de normalização ou padronização dos

mesmos, devido à objetividade do projeto proposto, não foram realizadas

“normalização” ou “padronização” dos dados, porém segue ressalva conforme

analisado:

Conforme visto existe um alto índice de outliers entre as features 40 à 150, sendo as

mesmas normalizadas naturalmente a partir da feature 151 adiante.

Como o problema proposto é “classificação” com variáveis independentes numéricas,

pode-se esperar que o resultado possa ser afetado devido a falta de normalização dos

dados ou remoção dos outliers. Partiu-se da hipótese que os outliers não afetam a

previsibilidade dos dados e os modelos propostos, porém sabendo que o modelo pode ser

otimizado por essas técnicas

É sugerido uma análise de relevância das features para o modelo, a técnica ANOVA é a escolha inicial para entender a significância que as features tem para a variável de saída.

Segunda a configuração do modelo, onde temos uma multi-classificação de saída e as variáveis de entradas são todas numéricas, realizar essa análise por variância dentro dos grupos é viável. Devido a limitações e escopo do projeto, não será realizado durante os testes nenhuma técnica de seleção de features. Partimos das hipóteses que todas contribuem para o resultado do modelo, porém, sabendo que o modelo pode ser otimizado por essa técnica.

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

Levando em consideração os pontos acima, foi realizada a técnica de K-fold com

valores 3, 5 e 10 para o dataset de treinamento e teste para todos os modelos

treinados e estudados.

Utilizada técnicas de mini-batch para o cálculo entre as diversas máquinas para

melhor resultado performance durante o processamento dos modelos e

interatividade do Grid Search CV via Google Colab.

# Resultados

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# Dados sem pré-processamento

XXXXX REQUISITOS

# KNN

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# DT

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# MLP

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# NB

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# SVM

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# Resultados após pré-processamento

XXXXX REQUISITOS

# KNN

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# DT

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# MLP

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# NB

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# SVM

XXXXX REQUISITOS

KNN, DT, MLP, NB e SVM

# Conclusões

XXXXX

# Referências

Dados sobre banco de dados:

* URL do *Dataset* escolhido:
  + <https://www.openml.org/search?type=data&status=active&id=6>
* Informações adicionais sobre a base de dados:
  + https://link.springer.com/article/10.1007/BF00114162

GITHUB:

* Folder do projeto:
  + https://github.com/DannGirotto/TCC\_ML2
* URL do arquivo do *Dataset:*
  + https://github.com/DannGirotto/TCC\_ML2/blob/main/PROJETO/DATASET/dataset\_6\_letter.arff
* URL do projeto em Python:
  + https://github.com/DannGirotto/TCC\_ML2/blob/main/PROJETO/Python/TCC.py

Logotipo

Descrição gerada automaticamente

Logotipo, nome da empresa

Descrição gerada automaticamente