

Utilização de Métodos de Aprendizado de Máquina na Classificação de Imagens para Reconhecimento de Lixo Orgânico ou Reciclável

Antônio Marcos Machado, Bárbara Boechat, Bruce Williss

Universidade Federal de São João del-Rei

Abstract

O objetivo deste estudo é apresentar e utilizar cinco algoritmos de aprendizado de máquina supervisionados conhecidos para tarefa de classificação de imagens: SVM, Árvores de Decisão, Naive-Bayes, KNN e Redes Neurais Convolucionais baseadas em TensorFlow. Esses algoritmos foram induzidos a fazer previsões em um conjunto de dados de tamanho médio, cujos dados são baseados em resíduos orgânicos e resíduos recicláveis. No final, estaremos comparando seu desempenho com base na precisão.

Palavras-chave: aprendizado supervisionado, classificação de imagens, árvores de decisão, classificador id3, naïve bayes, bayes, svm, k nearest neighbors, tensor flow, comparação aprendizado supervisionado.

1 Introdução

Aprendizado de Máquina (do inglês: *Machine Learning*) é um subcampo da Engenharia e da Ciência da Computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial [Hosch, 2021]. Em 1959, Arthur Samuel definiu aprendizado de máquina como o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados". O aprendizado automático explora o estudo e construção de algoritmos que podem aprender de seus erros e fazer previsões sobre dados.

O aprendizado de máquina compreende realizar tarefas que são tipicamente classificadas em três grupos: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

Neste trabalho, o foco principal é apresentar, descrever, analisar e comparar a performance de cinco diferentes algoritmos de aprendizado supervisionado que terão seu foco em atuar sobre a tarefa de classificação de imagens. Os algoritmos em questão são: K-Nearest Neighbors (KNN), Árvores de Decisão (Decision Trees), Naive-Bayes, Redes Neurais Convolucionais - baseadas na biblioteca de classificação de imagens do TensorFlow - e Máquina de Vetor de Suporte (SVM).

Para induzir esses algoritmos à classificação, foi escolhido um dataset de tamanho médio que foi obtido do repositório público da *Kaggle*. Esse dataset armazena imagens que são divididas em dois conjuntos: lixo orgânico e lixo reciclável. Inicialmente, tratando-se de um problema de classificação de imagens, é fundamental que o conjunto de dados utilizado passe por um pré-processamento que padronize o formato e algumas de suas características. Esse é um procedimento clássico e fundamental quando se trata de classificação de imagens sendo conhecido como "aumento de dados" (do inglês: *data augmentation*), utilizado para facilitar o trabalho dos classificadores e alcançar uma melhor performance. Esse processo é melhor detalhado na seção 2.

Após o pré-processamento, esses algoritmos serão induzidos, separadamente, à classificação do dataset e, ao final, vamos analisar cada algoritmo quanto à sua capacidade de analisar e realizar previsões sobre esse conjunto e estabelecer um comparativo quanto a performance de cada um por meio de algumas métricas de desempenho aplicáveis à algoritmos de aprendizado supervisionado.

2 Fundamentação Teórica

2.1 Classificação de Imagens

A classificação de imagens é o processo de categorizar e rotular vetores em uma imagem (grupos de pixels) com base em regras específicas, as quais podem se basear em uma ou mais características espectrais, ou texturais. A classificação pode ser feita através de dois métodos gerais os "supervisionados" e "não supervisionados".

Classificação Não Supervisionada

O método não supervisionado é totalmente automatizado sem o uso de dados de treinamento. Utilizando um algoritmo adequado, as características especificadas de uma imagem são detectadas sistematicamente durante o estágio de processamento da imagem. Estes métodos descrevem um modelo de 'agrupamento de imagens' ou 'reconhecimento de padrões', os algoritmos 'ISODATA' e 'K-mean' são comumente utilizados.

Classificação Supervisionada

O método supervisionado é o processo de selecionar amostras visualmente nas imagens dos dados de treinamento e atribuí-las a categorias pré-selecionadas dependendo da base de

dados utilizada, essas categorias podem ser, por exemplo, estradas, edifícios, corpo d'água, vegetação e entre outras, de modo a criar medidas estatísticas a serem aplicadas à imagem inteira.

'Máxima verossimilhança' e 'distância mínima' são dois métodos comuns para categorizar a imagem inteira usando os dados de treinamento. Por exemplo, a classificação de 'probabilidade máxima' usa as características estatísticas dos dados, em que os valores de média e desvio padrão de cada índice espectral e textural da imagem são calculados primeiro. Então, considerando uma distribuição normal para os pixels em cada classe e usando algumas estatísticas clássicas e relações probabilísticas, a probabilidade de cada pixel pertencer a classes individuais é calculada. Finalmente, os pixels são rotulados para uma classe de recursos que mostram a maior probabilidade.

2.2 Processamento de Imagens

O banco de imagens necessitou da etapa de pré-processamento antes da etapa seguinte de extração de dados e classificação. Assim, foram realizados três tratamentos diferentes, o redimensionamento, o histograma de equalização e a representação das imagens em escalas de cinza, a seguir em 1, 2 e 3 mais detalhes sobre os métodos.

1. **Redimensionamento:** Imagens com resoluções elevadas podem causar lentidão no processado caso sejam usadas em seus estados originais, então foi necessário realizar um redimensionamento para que as imagens passem a ter resolução de 150x150;
2. **Histograma de Equalização:** O histograma é utilizado para fornecer a ideia geral de como é uma imagem, então, a partir disso é possível tentar equalizar uma imagem que aparentemente esteja escura, ou seja, imagem que tenha pixels com valores baixos no histograma, ou que seus pixels tenham valores elevados no histograma, indicando uma imagem muito clara. É ideal equalizar os valores dos pixels com o objetivo de uniformemente espalhar seus níveis e obter como resultado uma melhora no contraste da imagem;
3. **Escalas de Cinza:** A conversão da imagem colorida em escala de cinza costuma ser um requisito importante no processamento de imagens e tem como objetivo usar o número limitado de níveis de cinza para preservar a maior quantidade possível do contraste da imagem original. A intensidade do nível de cinza está normalmente no intervalo $[0, 2^b - 1]$, tal que b é o número de bits por pixel.

2.3 Support Vector Machine

Uma máquina de vetores de suporte a Support Vector Machine (SVM) é um conceito na ciência da computação para um conjunto de métodos de aprendizado supervisionado que analisam os dados e reconhecem padrões, usado para classificação e análise de regressão. O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico.

Dado um conjunto de exemplos de treinamento, sendo cada um marcado como pertencente a categorias diferentes, o algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos às categorias. Esse modelo representa essas classificações como pontos no espaço, os quais são mapeados para que os exemplos de cada categoria sejam divididos por um espaço tão amplo quanto possível. Os novos exemplos são então mapeados no mesmo espaço e preditos como pertencentes a uma categoria com base no lado do espaço em que eles foram colocados.

Em outras palavras, a Support Vector Machine encontra uma linha de separação (hiperplano) entre dados de duas classes, a qual busca maximizar a distância entre os pontos mais próximos em relação às distintas classes. A distância entre o hiperplano e o primeiro ponto de cada classe é chamada margem. Em primeiro lugar a SVM coloca a classificação das classes, definindo as respectivas classes de cada ponto, e na sequência maximiza a margem. Assim, são primeiramente classificadas as classes corretamente e depois em função dessa restrição são definidas as distâncias entre as margens.

2.4 Árvore de Decisão

O aprendizado de árvore de decisão é uma abordagem de modelagem preditiva usada em estatística, mineração de dados e aprendizado de máquina. Neste método é utilizada uma árvore de decisão como modelo preditivo para realizar observações sobre um item representado pelos ramos e para conclusões sobre as folhas, que representam o valor alvo do item.

A árvore em que a variável de destino pode assumir um conjunto discreto de valores é chamada árvore de classificação e nessas estruturas, as folhas representam rótulos de classe e os ramos representam conjunções de recursos que levam a esses rótulos de classe. Já as árvores em que a variável de destino pode assumir valores contínuos são chamadas árvores de regressão. As árvores de decisão estão entre os algoritmos de aprendizado de máquina mais populares devido ao seu fácil entendimento e simplicidade. As árvores de decisão usadas na mineração de dados são de dois tipos principais:

1. **Análise da árvore de classificação:** Resultado previsto é a classe (discreta) à qual os dados pertencem.
2. **Análise da árvore de regressão:** Resultado previsto pode ser considerado um número real (por exemplo, o preço de uma casa ou o tempo de permanência de um paciente em um hospital).

Uma árvore é construída dividindo o conjunto de origem, constituindo o nó raiz da árvore, em subconjuntos que constituem os filhos sucessores. A divisão é baseada em um conjunto de regras de divisão baseadas em características de classificação. Este processo é repetido em cada subconjunto derivado de uma maneira recursiva chamada particionamento recursivo. A recursão é concluída quando o subconjunto em um nó tem todos os mesmos valores da variável de destino ou quando a divisão não agrega mais valor às previsões. Este processo de indução top-down de árvores de decisão é um exemplo de um algoritmo guloso, e é de longe a estratégia mais comum para aprender árvores de decisão a partir de dados.

Neste trabalho é adotada a árvore de classificação e pretende criar um modelo que preveja o valor de uma variável de destino com base em várias variáveis de entrada. Dessa forma, utilizando a árvore de decisão como uma representação simples para classificar exemplos e cada elemento do domínio da classificação é denominado classe.

2.5 K Nearest Neighbors

O algoritmo de k vizinhos mais próximos (KNN) é um método de classificação não paramétrico desenvolvido por Evelyn Fix e Joseph Hodges em 1951, o qual acabou sendo expandido por Thomas Cover posteriormente. Ele é utilizado para classificação e regressão e em ambos os casos, a entrada consiste nos k exemplos de treinamento mais próximos no conjunto de dados. Já a saída depende de como o KNN é usado, seja classificação ou regressão.

A saída é uma associação de classe ao utilizar o KNN para classificação, assim, um objeto é classificado através dos votos de seus vizinhos, com o objeto sendo atribuído à classe mais comum entre seus k vizinhos mais próximos tal que k é um número inteiro positivo pequeno. Se $k = 1$, então o objeto é simplesmente atribuído à classe daquele único vizinho mais próximo. Neste algoritmo a função é aproximada apenas localmente e todos os cálculos são adiados até a avaliação da função, ele depende da distância para classificação, se os recursos representam unidades físicas diferentes ou vêm em escalas muito diferentes, então a normalização dos dados de treinamento pode melhorar drasticamente a precisão. Já na regressão KNN, a saída é o valor da propriedade do objeto, sendo que este valor é a média dos valores dos k vizinhos mais próximos.

O KNN é um algoritmo sensível à estrutura local dos dados, então, os vizinhos são obtidos de um conjunto de objetos para os quais a classe (classificação) ou o valor da propriedade (regressão) é conhecido. Dessa maneira, isso pode ser considerado o conjunto de treinamento para o algoritmo, embora nenhuma etapa de treinamento explícita seja necessária.

Por fim, tanto para classificação quanto para regressão, é preciso destacar uma técnica útil para atribuir pesos às contribuições dos vizinhos, de maneira que os vizinhos mais próximos consigam contribuir mais para a média do que aqueles mais distantes. Um exemplo seria um esquema de ponderação comum que consiste em dar a cada vizinho um peso de $1/d$, onde d é a distância ao vizinho. Neste trabalho foi utilizado o algoritmo KNN básico da biblioteca SciKit Learn.

2.6 Artificial Neural Networks

As redes neurais artificiais são modelos computacionais inspirados pelo sistema nervoso central de um animal, considerando particularmente exame das estruturas do cérebro, especificamente examinando os neurônios. A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado aos seus pesos, o treino. Dessa forma, esses modelos conseguem realizar tanto o aprendizado de máquina como o reconhecimento de padrões.

Essas redes são geralmente apresentadas como sistemas de "neurônios interconectados, que podem computar valores de entradas", simulando assim o comportamento de redes neurais biológicas. Dessa forma, para ilustrar o conceito um exemplo seria uma rede neural para o reconhecimento de escrita manual, esta rede é definida por um conjunto de neurônios de entrada que podem ser ativados pelos pixels de uma imagem de entrada. Os dados adquiridos pela ativação dos neurônios em seguida são repassados, ponderados e transformados por uma função determinada pelo arquiteto da rede, para outros neurônios. Este processo se repete até que um neurônio de saída seja ativado, o que determina que um caractere foi lido, então, a aprendizagem ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente, dessa maneira, os paradigmas existentes são a aprendizagem supervisionada e a não supervisionada. Respectivamente, o primeiro acontece quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada, o segundo quando não existe um agente externo indicando a resposta desejada para os padrões de entrada. Também há um terceiro paradigma a aprendizagem por reforço, a qual acontece quando há avaliação extensa da resposta fornecida pela rede.

Redes Neurais Convolucionais

Redes neurais convolucionais (CNNs) pertencem a uma categoria de algoritmos baseados em redes neurais artificiais que utilizam a convolução em pelo menos uma de suas camadas. As CNNs provaram ser eficientes em diversas tarefas de reconhecimento de imagens e vídeos, sistemas de recomendação e processamento de linguagem natural, tornando-se o novo padrão em visão computacional e são fáceis de treinar quando existe um abundante número de amostras rotuladas que representam as diferentes classes-alvo.

Algumas das vantagens consistem em:

1. Sua capacidade de extrair características relevantes através de aprendizado de transformações, os kernels;
2. Dependência de menor número de parâmetros de ajustes do que redes totalmente conectadas com o mesmo número de camadas ocultas. Como cada unidade deu uma camada não é conectada com todas as unidades da camada seguinte, há menos pesos para serem atualizados, o que facilita o treinamento.

As CNNs são formadas por sequências de camadas e cada uma destas possui uma função específica na propagação do sinal de entrada. A figura 1 ilustra a arquitetura de uma LeNete suas três principais camadas: convolucionais, de pooling e totalmente conectadas. As camadas convolucionais são responsáveis por extrair atributos dos volumes de entradas. As camadas de pooling são responsáveis por reduzir a dimensionalidade do volume resultante após as camadas convolucionais e ajudam a tornar a representação invariante a pequenas translações na entrada.

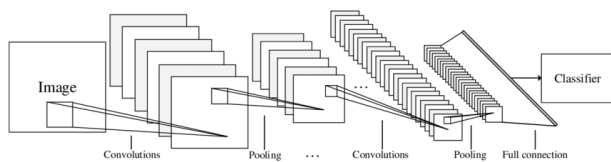


Figure 1: Arquitetura LeNete

3 Trabalhos Relacionados

Trabalhos que utilizam a técnica de reconhecimento de imagens são comumente encontrados na literatura e utilizam diferentes técnicas de tratamento de imagens e diferentes algoritmos de aprendizado de máquina, como, por exemplo, nos artigos de [Montanari, 2015] e [Maia, 2016].

O primeiro intitulado de Detecção e classificação de objetos em imagens para rastreamento de veículos, traz a contribuição de um método que propõe um sistema de visão computacional para rastrear veículos usando sistemas de detecção e classificação de segmentos em imagens que emprega a técnica *bag-of-features* que é uma adaptação do método *bag-of-words*. Porém, na adaptação a etapa de agrupamento é feita através de *features* retiradas das imagens.

Já o segundo trabalho de [GHELLERE, 2015], com o título de Detecção de Objetos em Imagens por Meio de Combinação de Descritores Locais e Classificadores, também trabalha com o reconhecimento de imagens, contudo, este trabalho traz cinco abordagens de aprendizado supervisionado, utilizando os algoritmos: Multilayer Perceptron, FURIA, Random Forest, Support Vector Machines e o KNN.

Os algoritmos de reconhecimento de imagens também são utilizados para reconhecimento fácil, como no artigo de [Maia, 2016] intitulado Detecção e Reconhecimento Facial por Meio de Aprendizado de Máquina, em que a primeira etapa consiste em construir um banco de faces através da extração de características faciais para construção de um banco de faces e também da extração de atributos para utilização do treinamento e teste de uma rede neural.

Apesar das grandes contribuições para o avanço do reconhecimento facial, é preciso destacar que estes algoritmos podem ter suas classificações influenciadas pelo viés algorítmico, como destacado por [Joy Buolamwini, 2018] em seu artigo, o qual aponta preconceito de gênero e racial em sistemas comerciais de inteligência artificial, a autora examinou softwares de reconhecimento facial e encontrou uma taxa de erro de 0,8% para homens de pele clara e 34,7% para mulheres de pele escura.

4 Desenvolvimento

Neste trabalho foi escolhida a linguagem de programação Python e suas bibliotecas Scikitlearn e TensorFlow, a base de dados escolhida foi retirada do Kaggle e é chamada de Waste Classification DataSet. O conjunto de dados é dividido em 85% dados de treinamento e 15% dados de teste, contando com 22564 imagens de treinamento e 2513 imagens de teste.

4.1 Organização do Conjunto de Treino

Do conjunto de testes, **1.401** imagens são do tipo lixo orgânico e **1.112** imagens são do tipo lixo reciclável. Do

conjunto de treinamento, **12.565** imagens são do tipo lixo orgânico e **9.999** são do tipo lixo reciclável.

4.2 Hardware Utilizado

- Processador: Intel(R) Core(TM) i7-8665U CPU @ 1.90GHZ
- Placa de Vídeo: Intel(R) UHD Graphics 620
- Memória Ram: 32gb

4.3 Aplicação dos Algoritmos de Processamento

As imagens foram tratadas utilizando os métodos citados em 2.2 e com auxílio das bibliotecas da linguagem Python. Ao aplicar os algoritmos que implementam os métodos de pré-processamento é possível verificar mudanças como as demonstradas pelas figuras 2 e 3 em que a primeira é a imagem original que conta com todas as suas cores e suas matrizes para cada canal das cores RGB, enquanto a segunda demonstra a única matriz na qual a original foi convertida após passar pelo processo de *GrayScaling*.

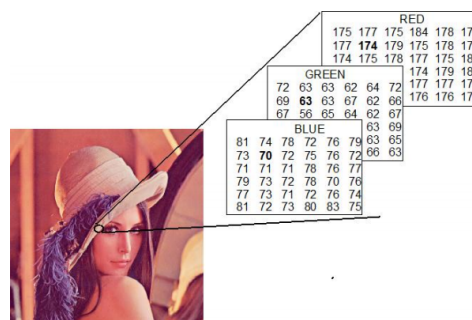


Figure 2: Imagem Colorida



Figure 3: Imagem em GrayScale

Já na figura 4 no lado esquerdo está a imagem original já em *GrayScale* antes da normalização dos níveis de seus pixels. E no lado direito, a imagem já foi equalizada e conta agora com uma maior nitidez ou pouca disparidade nos níveis dos pixels.



Figure 4: Resultado do Histograma de Equalização

4.4 Execução dos Algoritmos

Após essa etapa descrita em 4.3, os algoritmos de Redes Neurais Convolucionais utilizando TensorFlow, SVM, KNN, Naive Bayes e Decision Tree foram executados para a base de dados devidamente pré-processada.

Os seguintes parâmetros base foram utilizados nos respectivos classificadores para execução dos testes:

- **K-Nearest Neighbors:**
Número de vizinhos (n_neighbors): 10,
Uso de processadores/núcleos (n_jobs) = -1,
- **Decision Trees:**
Critério: entropy,
Grau de importância (splitter): best,
Limite de profundidade (max_depth): none.
- **Redes Neurais Convolucionais:**
Modelo: Sequencial,
MaxPooling (pool_size): (2,2),
Kernel (kernel_size): (3,3),
Otimizador: Adam(lr=0.0001),
Perca: binary_crossentropy.
Parâmetros (fitting):
validation_split=0.3,
epochs=30,
batch_size=25,
steps_per_epoch=100,
verbose=2.
- **Naive-Bayes:**
Cálculo de Estabilidade (var_smoothing):
np.logspace(0,-9, num=100),
Verbosidade (verbose) = 1,
Scoring = accuracy.
- **Support Vector Machine (SVM):**
Kernel: rbf,
Outros parâmetros (SVC):
C=1.0,
degree=3,
gamma='scale',
coef0=0.0,
shrinking=True,
probability=False,
tol=0.001,

```
cache_size=200,
class_weight=None,
verbose=False,
max_iter=-1,
decision_function_shape='ovr',
break_ties=False,
random_state=None.
```

5 Validação: Cross-Validation Score

Agora que os algoritmos conseguem realizar previsões com base nos parâmetros de execução especificados, é necessário analisar seu desempenho. Para otimizar esse feito, uma das técnicas utilizadas em algoritmos de classificação é o método da Validação Cruzada, uma técnica estatística que consiste em dividir aleatoriamente o conjunto de dados em k-grupos (chamados folds) de tamanhos aproximadamente iguais.

Para reduzir a variabilidade, várias rodadas de validação cruzada são realizadas com diferentes subconjuntos dos mesmos dados. Combinando os resultados de validação dessas rodadas obtém-se uma estimativa precisa do desempenho preditivo de cada modelo. O primeiro grupo é mantido para testes e os k-1 grupos restantes serão utilizados para treinamento.

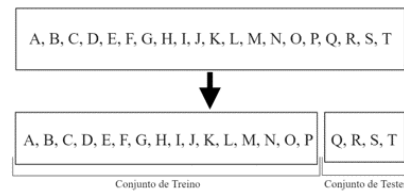


Figure 5: Exemplo de particionamento de um conjunto de 20 pontos de dados em 2 subconjunto: um para treino e o outro para testes.

Esse processo é repetido k vezes e a cada vez um grupo/fold diferente é usado para validação. A medida de desempenho relatada pela validação cruzada será, então, a média dos valores calculados nas iterações[26]. Tipicamente o valor de k é 5 ou 10. Para todos os classificadores, o valor de k utilizado é 10, pois o tempo computacional é otimizado para os conjuntos de dados tratado e atribui-se uma característica importante que é o chamado *bias reduzido*, que trata a ocorrência dos algoritmos darem preferência de uma hipótese sobre outra. Todos os dados analisados são testados apenas uma vez e usados no treinamento k-1 vezes.

6 Análise de resultados

Os resultados são descritos nas tabelas 1 e 2, nelas estão os valores de acurácia com e sem a validação cruzada, essa métrica foi escolhida neste trabalho para avaliar a eficiência dos algoritmos citados anteriormente em 4.4 na classificação das imagens de lixo orgânico ou reciclável.

A acurácia se faz importante, pois é preciso confiar nos resultados. Afinal, eles são a base de uma decisão final para alguma aplicação. Essa métrica é dada pelo cálculo da fração dos positivos verdadeiros e dos falsos positivos sobre o número total de atributos atribuídos.

Vem da fórmula:

$$acc = \frac{\text{Número de Predições Corretas}}{\text{Total de Predições Feitas}} \quad (1)$$

A acurácia é a proximidade de um resultado com seu valor de referência real. Dessa forma, quanto maior a acurácia, que vai de 0 a 100%, mais próximo da referência ou valor real é o resultado encontrado. O conceito de acuracidade é muitas vezes confundido com a ideia de precisão, mas seu significado é um pouco mais complexo. A precisão é o grau de variação resultante de um conjunto de medições realizadas, como visto. Com isso, quanto mais preciso, menor é a variabilidade entre os valores encontrados.

Sem Cross-Validation

A tabela abaixo foi obtida ao executar os algoritmos sem muitos ajustes no pré-processamento padronizado do dataset, sem a execução da validação cruzada e também sem muitos de ajustes no fitting do classificadores.

| | CNN | SVM | KNN | NB | DT |
|----|-------|--------|--------|--------|--------|
| 1ª | 50,0% | 60,9% | 59,3% | 57,75% | 57,99% |
| 2ª | 50,7% | 59,95% | 59,6% | 58,15% | 60,55% |
| 3ª | 50,0% | 59,75% | 58,9% | 57,49% | 60,3% |
| M | 50,2% | 60,2% | 59,26% | 57,79% | 59,61% |

Table 1: Tabela de Resultados Preliminares. **Glossário:** CNN = Convolutional Neural Networks, SVM = Support Vector Machine, KNN = K-Nearest Neighbors, NB = Naive Bayes, DT = Decision Tress

Com Cross-Validation

Após a validação dos algoritmos com a validação cruzada, uma melhor taxa de acerto dos algoritmos foi obtida, contudo a melhora não foi expressiva, é possível perceber através dos dados na tabela 2 que a maior taxa de melhora foi encontrada no algoritmo KNN tendo acertado 59,3% na execução sem a validação cruzada e 61,9% com a validação cruzada, obtendo assim a variação de acerto de 2,6%. Contudo, o algoritmo SVM obteve a maior taxa de acerto dentre todas as execuções realizadas tendo acertado 68,9%. O algoritmo CNN não obteve resultados utilizando a validação cruzada já que sua execução de forma geral demanda bastante tempo e processamento, assim, não tornou-se prioridade em detrimento às outras execuções planejadas para este trabalho.

| | CNN | SVM | KNN | NB | DT |
|----|-----|--------|--------|--------|--------|
| 1ª | 0% | 68,9% | 61,55% | 60,49% | 59,3% |
| 2ª | 0% | 67,65% | 61,9% | 61,20% | 60,55% |
| 3ª | 0% | 67,75% | 60,9% | 60,9% | 60,3% |
| M | 0% | 67,8% | 61,45% | 60,32% | 59,6% |

Table 2: Tabela de Resultados Final. **Glossário:** CNN = Convolutional Neural Networks, SVM = Support Vector Machine, KNN = K-Nearest Neighbors, NB = Naive Bayes, DT = Decision Tress

Tempos de Treinamento

A tabela 3 concentra a informação sobre a quantidade de tempo que cada algoritmo demorou para realizar a etapa de treinamento, é possível perceber que o algoritmo SVM realizou o treino no pior tempo e a melhor taxa de acerto, contudo, ainda assim, contribuindo para que fosse inviável realizar muitas execuções. O algoritmo KNN é uma exceção aos dados colhidos, pois este não realiza a etapa de treino, então sua execução foi medida totalmente, porém essa característica confere vantagem ao seu resultado apresentado.

| | CNN | SVM | KNN | NB | DT |
|-------|-----|----------|------|------|--------|
| 1ª | 0 | 15201,54 | 0,35 | 2,59 | 567,92 |
| 2ª | 0 | 15131,54 | 0,30 | 3,00 | 550,92 |
| 3ª | 0 | 15005,54 | 0,29 | 2,66 | 570,92 |
| Média | 0 | 15112,87 | 0,31 | 2,75 | 563,25 |

Table 3: Tabela de tempos de execução (em segundos). **Glossário:** CNN = Convolutional Neural Networks, SVM = Support Vector Machine, KNN = K-Nearest Neighbors, NB = Naive Bayes, DT = Decision Tress

7 Conclusão

Apesar de um tempo de treinamento muito longo, nota-se que o SVM oferece uma taxa de acertos muito superior em relação aos outros classificadores. Isso pode ser explicado pelo fato de que o algoritmo do SVM é conhecido por seu truque de kernel para lidar com espaços de entrada não lineares e também por utilizar menos memória porque usa um subconjunto de pontos de treinamento na fase de **decisão**. Além disso, o SVM funciona bem com uma clara margem de separação e com grande espaço dimensional.

Os demais algoritmos se equiparam com taxas de acerto parecidas, mas muito abaixo do esperado. Contudo, sentimos a necessidade de fornecer um melhor ajuste nos parâmetros dos classificadores para obter melhores resultados, porém a execução do dataset tornava a reexecução sucessiva com ajustes inviável. Como planos futuros, para contornar esse problema, pretendemos otimizar o tempo de execução e fornecer melhores ajustes de parâmetros para obter uma melhor análise.

References

- [cro, 2020] Cross-validation: evaluating estimator performance. 2020.
- [Filho, 2018] Luiz César Medeiros Filho. Machine learning: o que a tecnologia tem a ver com ecm? 2018.
- [GHELLERE, 2015] JHONATTAN SALVADOR GHELLERE. Detecção de objetos em imagens por meio da combinação de descritores locais e classificadores. *UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR*, page 90, 2015.
- [González, 2019] Mariana González. O que é acurácia? entenda o conceito e sua importância. <https://blog.idwall.co/o-que-e-acuracia/>, 2019.

- [Hardesty, 2018] Larry Hardesty. Study finds gender and skin-type bias in commercial artificial-intelligence systems. *MIT News Office*, 2018.
- [Hosch, 2021] William L. Hosch. machine learning. *Neomind*, 2021.
- [Joy Buolamwini, 2018] Timnit Gebru Joy Buolamwini. Gender shades: Intersectional accuracy disparities in commercial gender classification. *Proceedings of Machine Learning Research* 81:1–15, page 15, 2018.
- [Maia, 2016] Hugo Leite Florenço Maia. Detecção e reconhecimento facial por meio de aprendizado de máquina. *UNIVERSIDADE DE BRASÍLIA - FACULDADE DE TECNOLOGIA: CURSO DE GRADUAÇÃO EM ENGENHARIA DE REDES DE COMUNICAÇÃO*, page 50, 2016.
- [Montanari, 2015] Raphael Montanari. Detecção e classificação de objetos em imagens para rastreamento de veículos. *Instituto de Ciências Matemáticas e de Computação - ICMC-USP*, page 77, 2015.
- [SciKitLearn, 2021] SciKitLearn. Machine learning in python. <https://scikit-learn.org/stable/>, 2021.
- [TensorFlow, 2021] TensorFlow. A principal biblioteca de código aberto para desenvolver e criar modelos de ML. <https://www.tensorflow.org/hub>, 2021.