Comparação de indutores para modelos de predição de classifição de imagens

Lucas Mrowskovsy Paim*

2019

Resumo

Este trabalho tem por objetivo comparar os indutores apresentados durante a disciplina de Machine Learning do programa de especialização em Inteligência Artificial Aplicada da Pontifícia Universidade Católica do Paraná, para predição de classificação de imagens, utilizando principalmente a bilioteca scikit-learn e a linguagem Python, assim como a demonstração de técnicas de combinação de indicadores e seleção dinâmica.

Tem por objetivo também a comparação dos resultados utilizando técnicas de extração de características manuais, handcrafted, e CNN deep-learning.

Para a extração de características de imagens foram utilizados as bibliotecas: opencv-python, skimage e keras

//Falta a conclusão

Palavras-chave: Naive Bayes. Árvores de decisão. Redes Neurais. k-Vizinhos Mais Próximos. Máquina de Vetores de Suporte.

Abstract

This paper aims to compare the inductors presented during the Machine Learning discipline of the Pontifical Catholic University of Paraná specialization program in Applied Artificial Intelligence, to predict image classification using mainly the scikit-learn library and the Python language as well as the demonstration of techniques of combination of indicators and dynamic inductor selection.

It also aims to compare the results using handcrafted and CNN deep-learning feature extraction techniques.

*Programa de Pós Graduação em Inteligência Aritificial, Curitiba - PR, 80215-901; Especializando em Inteligência Artificial Aplicada na PUCPR; Tecnólogo em Sistemas para Internet pela Universidade Positivo; E-mail: lucasmpaim1@gmail.com

For the extraction of image characteristics we used the libraries: opencv-python, skimage and keras.

Keywords: Naive Bayes. Decision Tree. Neural Network. k-Nearest Neighbors. Support Vector Machine.

1 Introdução

A utilização de classificadores de imagens é utilizada nas mais diversas áreas do conhecimento, que vão desde o auxílio de diagnóstico de câncer de pele Soares (2008), sistemas de segmentação e classificação de imagens via satélite Erhal et al. (1991), OCR's (Optical Character Recognition) para reconhecimento de textos e dígitos Ribas et al. (2012) etc.

Os mesmos indutores também podem ser utilizados para outras finalidades que não sejam classificação de imagem, como: detecção de emoções em processamento de textos Dosciatti, Ferreira e Paraiso (2013), regressão linear etc.

Segundo Coppin (2004), fornecer a capacidade de visão a sistemas computacionais, assim como a robôs, agentes é algo, sem dúvida, extremamente desejável.

As técnicas de extração de caracterísicas de imagens handcrafted que foram utilizadas são:

- LBP: Local Binnary Pattern: Segundo, Rosebrock (2015), esta é uma técnica de extração de texturas de uma imagem, em que cada pixel é comparado com seus vizinhos a partir de uma máscara e com isso é gerado uma sequência binária que toma o lugar daquele pixel.
- HOG: Histogram of Oriented Gradients: Segundo Mallick (2016) é uma técnica de extrai a distribuição (Historgramas) de direções de gradientes.
- Histograma de Cores RGB.

Visto o exposto o estudo desse trabalho se torna importante para a área de classifição de imagens, Podendo ser aplicado em diversas áreas, tendo por objetivo criar modelos de predição para classifição de imagens comparando vários indutores e os combinando para gerar maiores assertividades.

2 Método

A base utilizada para este trabalho, conta com apenas mil imagens, separadas em dez classes com cem imagens cada, sendo assim, é uma base completamente balanceada.

Todos os indutores, foram treinados utilizando a mesma base de imagens, em um primeiro momento ela é separada em 70% em treinamento e 30% em testes, técnica esta chamada de holdout (percentage-split), após esta separação ainda foi separado mais 30% da base de treinamento e criada uma base de validação, para evitar assim o chamado overfitting, que segundo Sarle (2016), é um dos maiores problemas da aprendizagem de máquina, isto indica que o indutor está perdendo sua capacidade de generalização, isto é,

segundo Lima, Pinheiro e Santos (2014) a capacidade do indutor responder adequadamente a dados que não fazem parte do conjunto de treinamento.

Para a comparação dos indutores será avaliado, com base em holdout e também em validação cruzada que segundo, Britto (2019b), consiste em dividir a base de dados em N grupos, em que um desses grupos é selecionado para teste e isto é repetido N vezes, garantindo assim que todas as instâncias serão utilizadas como teste ao menos uma vez.

Devido ao não suporte dos métodos de overfitting para alguns indicadores, este trabalho utiliza, quando disponível, o método:

```
.partial_fit(X, y, [classes])
```

Este método segundo learn (2019), é ideal para treinamento incremental, sendo assim, é possível extender a lógica do early stopping para alguns indicadores que não suportam tal método nativamente, como o Naive Bayes.

Como o método incremental, tende a adicionar ruído na base de dados, foi utilizado o método de IIR (Infinite Impulse Response), que segundo Grout (2008) é um filtro recursivo em que a saída do filtro é calculada usando as entradas atuais e anteriores e as saídas anteriores, fazendo assim com que o ruído seja minimizado, este filtro foi aplicado no tratamento de erro tanto da base de validação quanto da base de teste.

Sua fórmula consiste em:

$$y_t = \sum_{i=0}^{M} b_j x_(t-i) - \sum_{j=1}^{L} \alpha_j y_(t-j) \tag{1}$$

O resultado da aplicação desta técnica pode ser consultada na Figura 1, onde é possível observar que o gráfico foi suavizado, de forma que, quando ambas as bases forem confrontadas será possível detectar mais fácilmente onde está ocorrendo o overfitting.

Sendo assim podemos definir a taxa de aprendizado como sendo:

$$r = y_{t-1} - y_t \tag{2}$$

Onde y_{t-1} se refere ao erro na base de validação na iteração anterior e y_t se refere a iteração atual de treinamento, desta forma quando obtemos um valor para $r < 10^{-3}$ que é a taxa mínima de aprendizagem e isso ocorre por N vezes seguidas, significa que não há um ganho no treinamento e este pode ser interrompido.

2.1 Naive Bayes

Segundo, Britto (2019a), o teorema de Bayes consiste em uma abordagem probabilística para aprendizagem, calculando assim a probabilidade de uma característica X de um vetor de prever determinada classe.

A técnica de Naive Bayes, assume que todas as caractéristicas no vetor são independentes, por isso, é chamado de Naive (do inglês Ingênuo), pois determinadas caracterísicas podem estar relacionadas entre si, o que faz com que essa técnica tenha um resultado inferior a ténica de Redes Bayesianas que consideram que os atributos podem ou não ser relacionados, isto é, a característica X pode ou não estar diretamente relacionada a característica Y.

2.1.1 Resultados do Treinamento

A matriz de confusão para o esta técnica usando houldout e a base de características usando deeplearning pode ser consultada em Figura 3, assim como sua taxa de aprendizagem Figura 2

```
Acurácia Naive Bayes Validação : 0.8928571428571429
Acurácia Naive Bayes Teste     : 0.9266666666666666
```

Para o método usando as características extraídas de forma hand crafting, a curva de erro na aprendizagem pode ser consultado na Figura 4, assim como sua matriz de confução na Figura 5.

```
Acurácia Naive Bayes Validação : 0.7785714285714286
Acurácia Naive Bayes Teste     : 0.8533333333333334
```

2.2 Árvores de decisão

A técnica da árvore de decisão utiliza a teoria do ganho de informação (entropia) , para decidir seus novos vértices até que consiga realizar a classificação das instâncias.

A entropia pode ser definida pela fórmula:

$$S = -p_+ log_2 p_+ - p_- log_2 p_- \tag{3}$$

Onde p_+ é a proporção de elementos positivos em S e p_- a de elementos negativos.

As árvores geradas para a este trabalho podem ser consultadas em: Figura 6, Figura 8

2.2.1 Resultados do Treinamento

DEEP

```
Acurácia Decision Tree Teste     : 0.8466666666666667
```

HAND-CRAFT

```
Acurácia Decision Tree Teste     : 0.67
```

2.3 Redes Neurais

2.3.1 Resultados do Treinamento

DEEP

```
Acurácia Neural Network - adam Validação : 0.9357142857142857
Acurácia Neural Network - adam Teste     : 0.92
```

HAND-CRAFT

```
Acurácia Neural Network - adam Validação: 0.10714285714285714
Acurácia Neural Network - adam Teste : 0.1
```

2.4 KNN

A técnica de k-nearest neighbors, k-vizinhos mais próximos, consiste em agrupar intâncias similares, e quando novas intâncias são inseridas para a classificação, elas são classificados pelos k-vizinhos mais próximos.

Uma grande desvantagem desta abordagem que dispensa treinamento é que ela é mais lenta, uma vez que realiza todos os cálculos necessários a cada nova classificação

2.4.1 Resultados

DEEP

```
Acurácia KNN Teste : 0.96
```

HAND-CRAFT

```
Acurácia KNN Teste : 0.39666666666666667
```

2.5 SVM

A técnica de SVM consiste em "encontrar"novas dimensões para classificar os atríbutos

2.5.1 Resultado do Treinamento

DEEP

```
Acurácia SVM Teste : 0.9833333333333333
```

HAND-CRAFT

```
Acurácia SVM Teste : 0.7533333333333333
```

3 Considerações finais

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

> Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl

> hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Referências

BRITTO, A. *Aprendizagem Bayesiana*. 2019. Naive-bayes-britto. Disponível em: <http://www.ppgia.pucpr.br/~alceu/am/4%20-%20Naive%20Bayes%20and%20Bayesian%20Networks/NaiveBayes.pdf>. Acesso em: 25 set 2019. Citado na página 3.

BRITTO, A. *Conceitos Básicos*. 2019. Basic-concepts. Disponível em: <http://www.ppgia.pucpr.br/~alceu/am/1%20-%20Introduction%20to%20ML/Conceitos%20B%e1sicos.pdf>. Acesso em: 22 set 2019. Citado na página 3.

COPPIN, B. *Inteligência Artificial*. 1. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 2004. Acesso em: 22 set 2019. Citado na página 2.

DOSCIATTI, M. M.; FERREIRA, L. P. C.; PARAISO, E. C. *Identificando Emoções em Textos em Português do Brasil usando Máquina de Vetores de Suporte em Solução Multiclasse*. 2013. PPGIA. Disponível em: <http://www.ppgia.pucpr.br/~paraiso/mineracaodeemocoes/recursos/emocoesENIAC2013.pdf>. Acesso em: 21 set 2019. Citado na página 2.

ERHAL, G. J. et al. *Um Sistema de Segmentação e Classificação de Imagens De Satélite*. 1991. INPE. Disponível em: <http://sibgrapi.sid.inpe.br/col/sid.inpe.br/sibgrapi/2012/09.28.17.13/doc/Um%20sistema%20de%20segmentacao%20e%20classificacao.pdf>. Acesso em: 21 set 2019. Citado na página 2.

GROUT, I. *Learn more about Infinite Impulse Response*. 2008. Iir-concepts. Disponível em: <https://www.sciencedirect.com/topics/computer-science/infinite-impulse-response>. Acesso em: 24 set 2019. Citado na página 3.

LEARN scikit. *Incremental learning*. 2019. Incremental-learning. Disponível em: <https://scikit-learn.org/0.18/modules/scaling_strategies.html#incremental-learning>. Acesso em: 24 set 2019. Citado na página 3.

LIMA, I.; PINHEIRO, C. A. M.; SANTOS, F. A. O. *Inteligência Artificial*. 1. ed. Rio de Janeiro: CAMPUS, 2014. Acesso em: 22 set 2019. Citado na página 3.

MALLICK, S. *Histogram of Oriented Gradients*. 2016. Learnopencv. Disponível em: <https://www.learnopencv.com/histogram-of-oriented-gradients/>. Acesso em: 22 set 2019. Citado na página 2.

RIBAS, F. C. et al. *Handwritten digit segmentation: a comparative study*. 2012. UFPR. Disponível em: <http://www.inf.ufpr.br/lesoliveira/download/IJDAR2012.pdf>. Acesso em: 22 set 2019. Citado na página 2.

ROSEBROCK, A. *Local Binary Patterns with Python & OpenCV*. 2015. Pyimagesearch. Disponível em: <https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>. Acesso em: 22 set 2019. Citado na página 2.

SARLE, W. S. *Stopped Training and Other Remedies for Overfitting*. 2016. Overfitting. Disponível em: <https://s3.amazonaws.com/academia.edu.documents/31153602/1329.pdf>. Acesso em: 22 set 2019. Citado na página 2.

SOARES, H. B. *Análise e Classificação de Imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte*. 2008. UFRN. Disponível em: <https://repositorio.ufrn.br/jspui/bitstream/123456789/15118/1/HelianaBS_TESE.pdf>. Acesso em: 21 set 2019. Citado na página 2.

APÊNDICE A – Nullam elementum urna vel imperdiet sodales elit ipsum pharetra ligula ac pretium ante justo a nulla curabitur tristique arcu eu metus

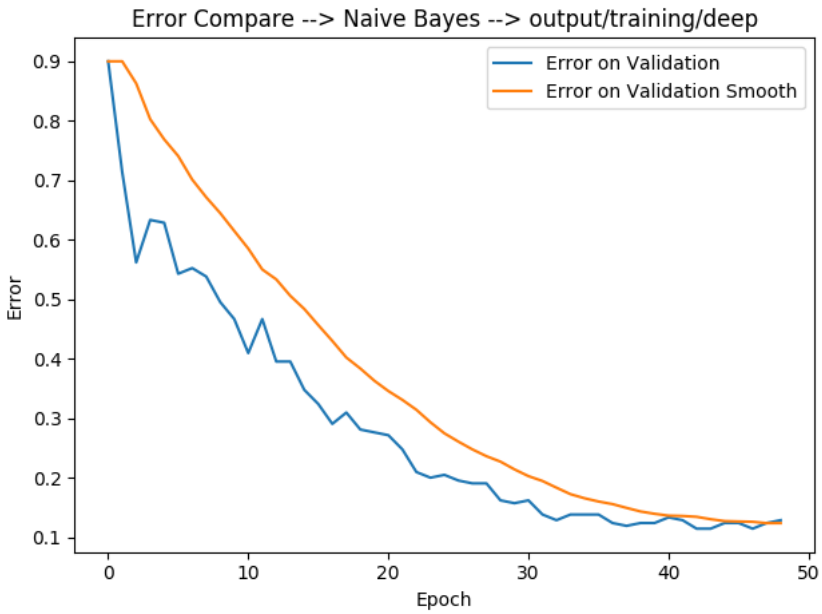
Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Agradecimentos

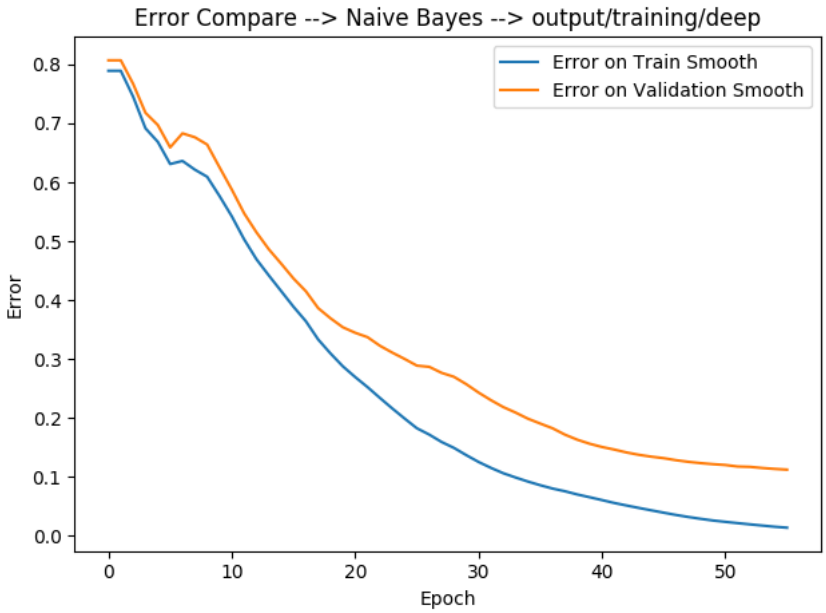
Texto sucinto aprovado pelo periódico em que será publicado. Último elemento pós-textual.

Figura 1 – Comparação do Erro com e sem IIR, usando Holdout



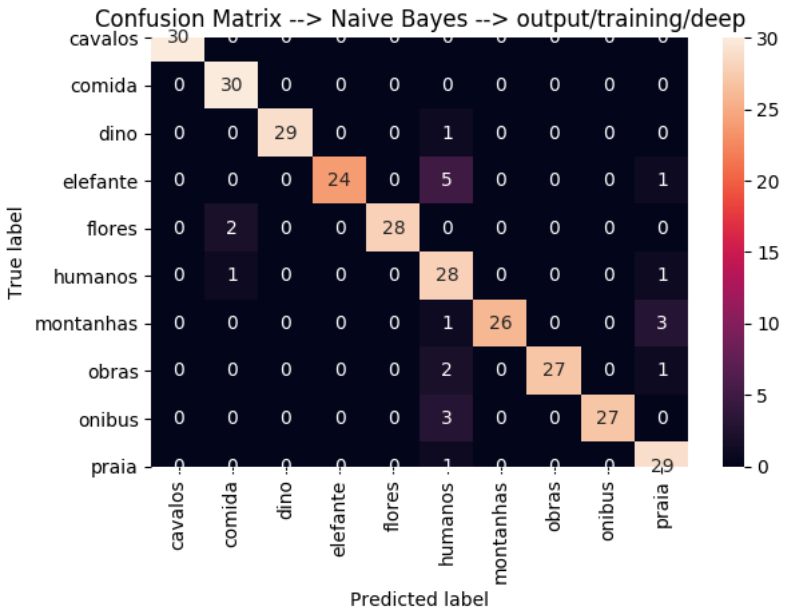
Fonte: O Autor

Figura 2 – Gráfico de Margem de erro usando Holdout



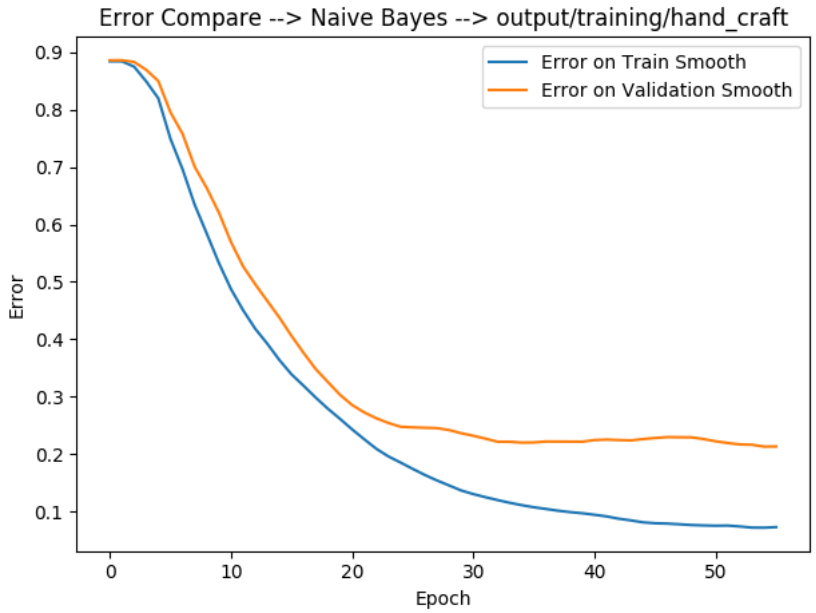
Fonte: O Autor

Figura 3 – Matrix de confusão Naive Bayes, usando Holdout



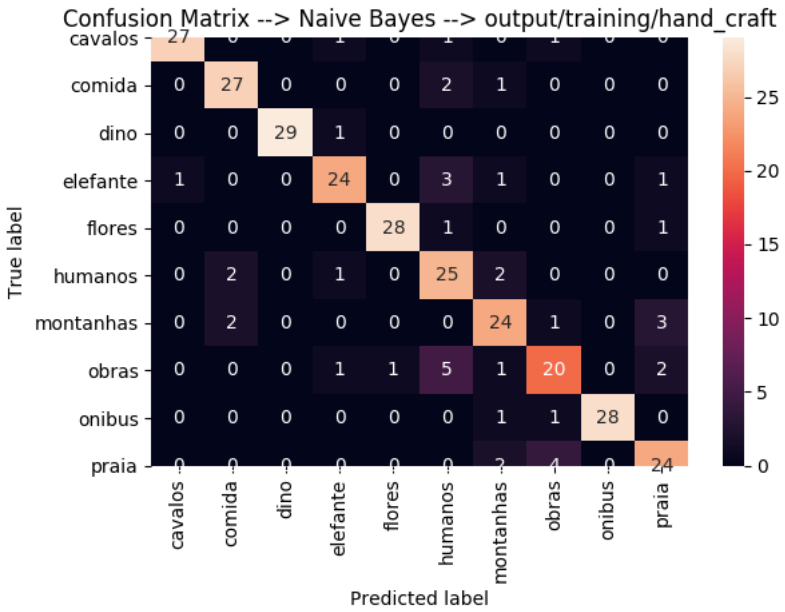
Fonte: O Autor

Figura 4 – Gráfico de treinamento Naive Bayes, usando Holdout



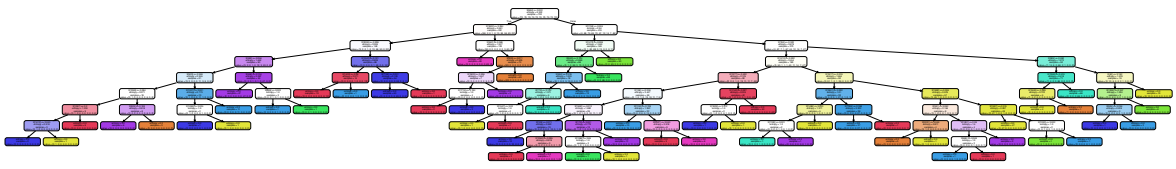
Fonte: O Autor

Figura 5 – Matrix de confusão Naive Bayes, usando Holdout



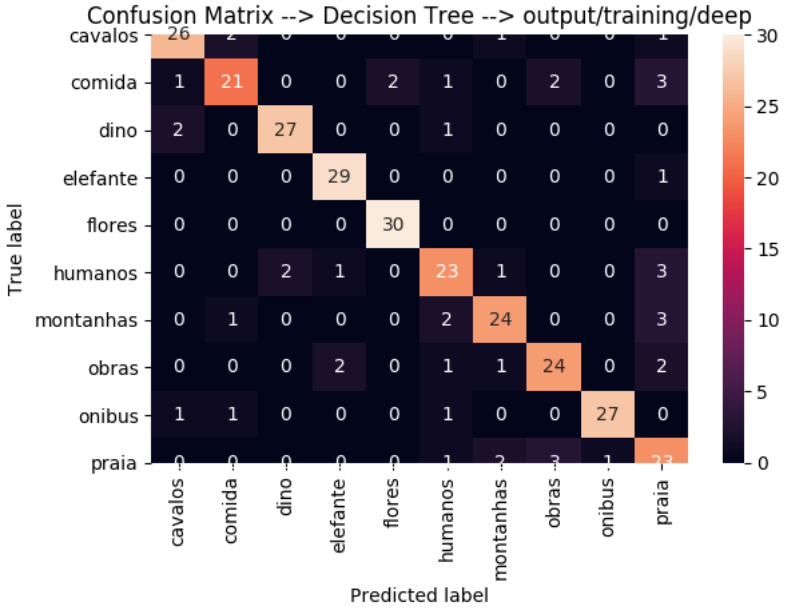
Fonte: O Autor

Figura 6 – Decision Tree, usando Holdout e Deep Learning



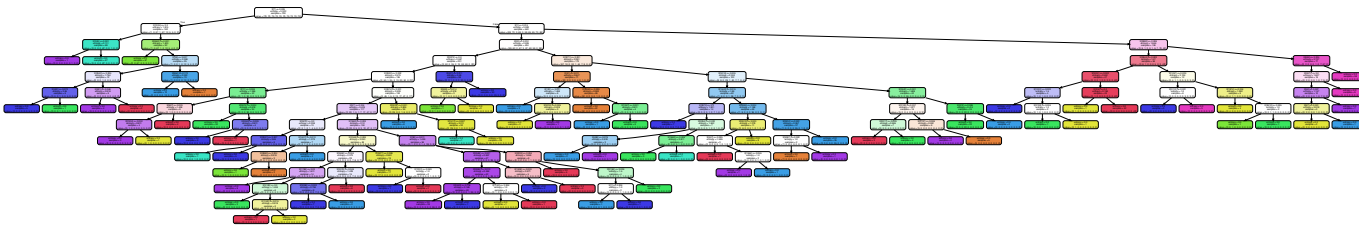
Fonte: O Autor

Figura 7 – Matrix de confusão Decision Tree, usando Holdout e Deep Learning



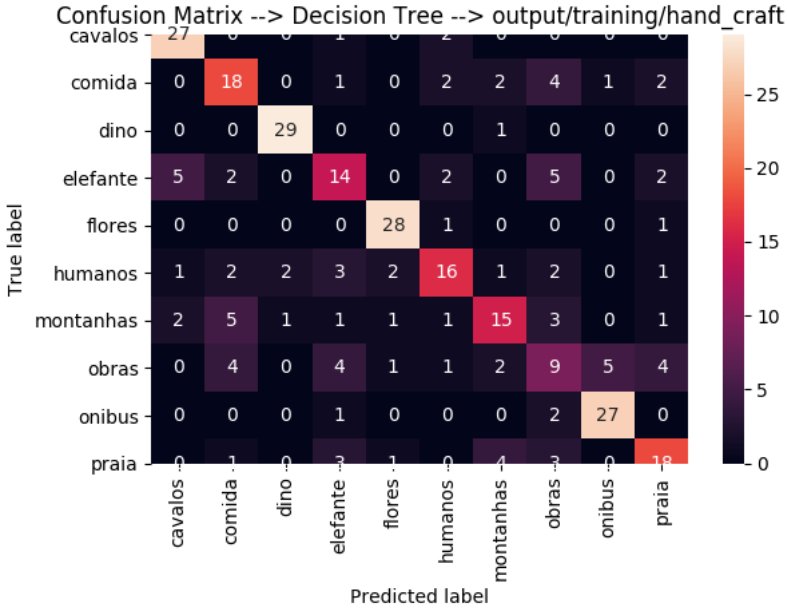
Fonte: O Autor

Figura 8 – Árvore de Desição, usando Holdout e Hand Craft



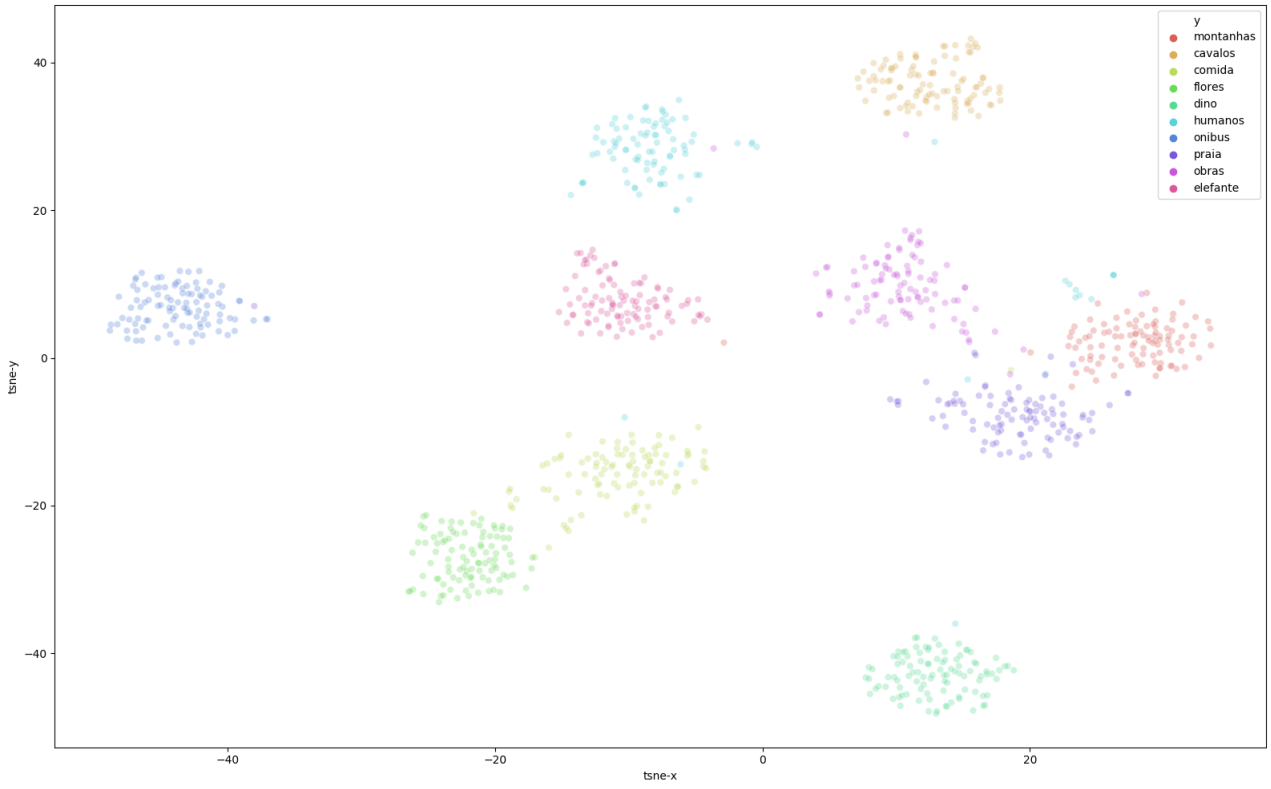
Fonte: O Autor

Figura 9 – Matrix de confusão Decision Tree, usando Holdout e Hand Craft



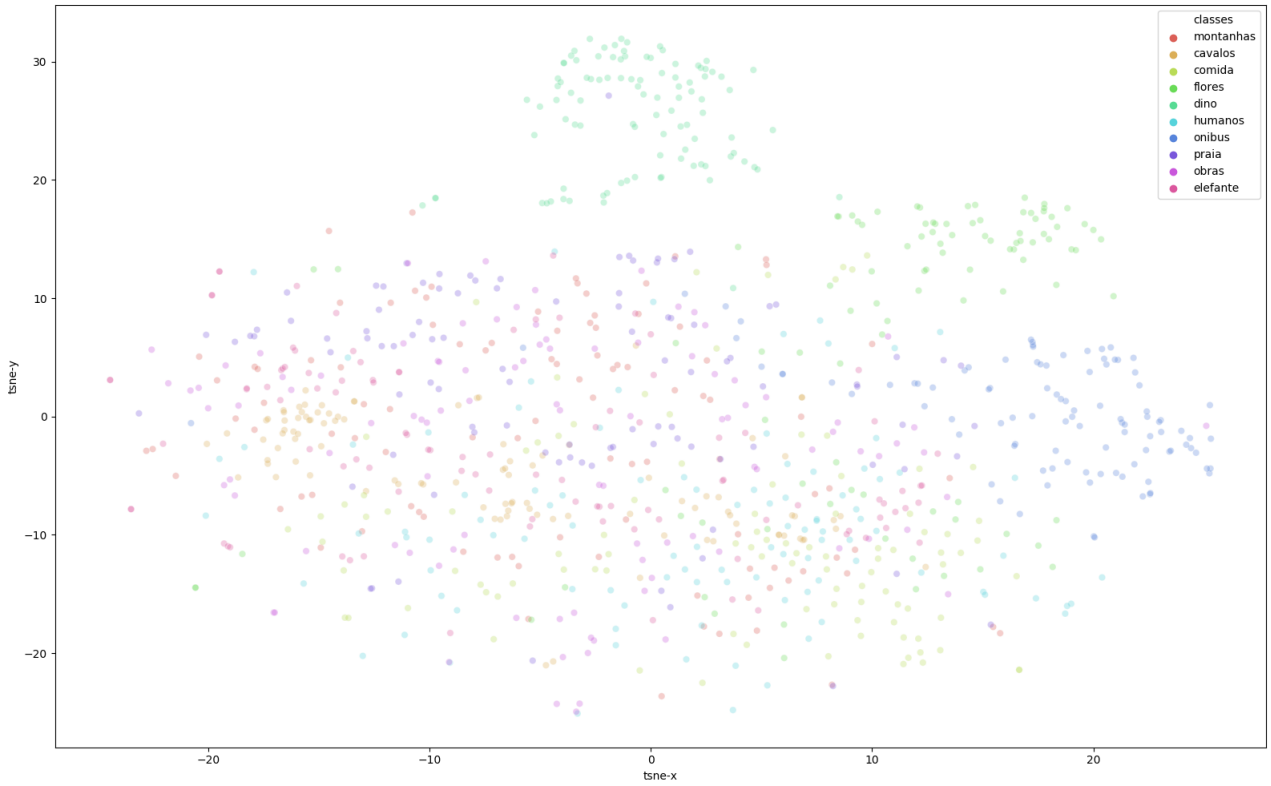
Fonte: O Autor

Figura 10 – TSNE Base Deep Learning



Fonte: O Autor

Figura 11 – TSNE Base Hand Crafting



Fonte: O Autor