Classificação de Gênero de Nomes com Técnicas de Aprendizado de Máquina Supervisionado

Camila Manara Ribeiro
Departamento de Computação
UFSCAR - Universidade Federal de São Carlos
São Carlos, Brasil
camilamanara@estudante.ufscar.br

Júlia Aparecida Sousa de Oliveira
Departamento de Computação
UFSCAR - Universidade Federal de São Carlos
São Carlos, Brasil
julia.sousa@estudante.ufscar.br

Luciana Oliveira de Souza Gomes
Departamento de Computação
UFSCAR - Universidade Federal de São Carlos
São Carlos, Brasil
luciana.souza@estudante.ufscar.br

Rafael Vinicius Polato Passador
Departamento de Computação
UFSCAR - Universidade Federal de São Carlos
São Carlos, Brasil
rafaelpassador@estudante.ufscar.br

***Resumo*—Esse artigo tem por objetivo realizar a classificação de gênero de nomes utilizando aprendizado de máquina em Python. Utilizamos um conjunto de dados com 147270 instâncias divididas entre nome, gênero, contador de frequência de ocorrências e probabilidade de gênero. Por ser um córpus desbalanceado, foi utilizada a técnica *random oversampling* para ajustar a distribuição de classe do conjunto de dados, resultando na duplicação de dados da classe minoritária, sendo essa a de nomes considerados do gênero masculino. Criou-se dois modelos compultacionais. Primeiramente foi feito o treinamento utilizando-se os nomes como *features*, através da utilização do calculo de frequência do TF-IDF, que apresentou resultados moderados. Paralelamente, realizou-se um treinamento utilizando features gramaticais, sendo elas o sufixo e o prefixo de cada nome do conjunto de dados. Como classificador, usou-se o algoritimo de baseado nos classificadores de Bayes, Naive Bayes. Por fim, encontrou-se um modelo com melhores resultados baseados nas métricas avaliativas acurácia e $F1$, em que o modelo composto pela combinação de sufixo e prefixo dos nomes como features apresentou 74,62% de acurácia e 76,85% de $F1$ em uma amostra de 500 nomes.**

***Index Terms*—Aprendizado de Máquina. *Features*.Classificação. Gêneros e Nomes.**

I. Introdução

Diversos trabalhos na literatura utilizam com êxito técnicas de Processamento de Linguaguem Natural (PLN), alinhados com algoritimos de aprendizado de máquina (AM) para elaborar modelos linguístico-compultacionais de classificação de tarefas.

Na língua inglesa, diversos estudos (DE CHOUDHURY et al., 2013; EICHSTAEDT et al., 2018; COPPERSMITH et al., 2018) comprovam a efetividade de técnicas de PLN para a classificação automática de textos em língua natural que denotam depressão. Entre estes trabalhos há uma grande variedade de abordagens em relação a escopo, plataformas analisadas, padrão-ouro (gold standard), engenharia de features e modelos usados, mas em geral eles comprovam que informações relevantes sobre a saúde mental de usuários de mídias sociais podem ser captadas a partir do conteúdo em texto produzido pelos mesmos.

Analogamente, diversos trabalhos também utilizam da análise textual para classificação de gênero. Tang et al. (2011) elaboram um modelo a partir de uma lista de nomes oriundos do facebook para classificação de gênero a partir do Multinomial Naive Bayes, enquanto Tripathi et al. (2011) e Hu et al. (2021) utilizam do SVM para prever gêneros em seus respectivos córpus.

O presente trabalho tem como objetivo fazer a implementação de algoritmos de classificação aplicados ao caso de identificação de gênero a partir de nomes disponibilizados como entrada na análise. Dessa forma, selecionamos diversas features a serem utilizadas para classificação através do algorítimo de aprendizado de máquina Naive Bayes, que serão explicitadas com mais detalhes posteriormente Para avaliação de qualidade, terão destaques os indicadores de acurácia e *F1-score*, antes e após balanceamento da base de dados utilizada.

Na Seção 2, serão apresentados os métodos utilizados no desenvolvimento do trabalho, explicando em detalhes os conceitos aplicados, linguagem de programação, bibliotecas/pacotes específicos. Na Seção 3, serão mostradas as informações referentes aos dados utilizados como características da base, instâncias e atributos. Na Seção 4, traz-se os resultados obtidos dos indicadores de acurácia e *F1-score* para considerando a base em análise – sendo que esses são discutidos por fim na Seção 5. A Seção 6, trata-se de todas as referências listadas que serviram de bibliografia para os autores do artigo.

II. Descrição dos métodos utilizados

Nessa seção, serão descritos os conceitos, métodos e ferramentas utilizadas para a realização do pré-processamento dos dados (a fim de tornar a base balanceada) e os algoritmos que foram necessários na realização da análise a partir da

classificação obtida. A classificação foi realizada através da extração de features, TF-IDF e *features* gramaticais como prefixos e sufixos com o algoritimo tradicional de aprendizado de máquina *Naive Bayes* Os principais conceitos utilizados para avaliação foram a acurácia e o *F1-Score*, sendo feita a comparação dos resultados a partir do tratamento dos dados.

Dessarte, em um primeiro modelo, transformamos toda lista do conjunto de nomes presente no dataset em vetores e, calculamos o TF-IDF de cada nome, uma medida estatística que tem o intuito de indicar a importância de uma palavra de um documento em relação ao dataset, ou seja, calcula-se um valor baseado na sua incidência e admite-se isso como *feature*. Paralelamente, um segundo modelo foi criado a partir da extração de sufixos, prefixos e a combinação de ambos, utilizando as letras dos nomes como *features* para o classificador.

O algoritmo *Naive Bayes* é um classificador probabilístico baseado no "Teorema de Bayes". Atualmente, o algoritmo se tornou popular na área de Aprendizado de Máquina (Machine Learning) para categorizar textos baseado na frequência das palavras usadas, e assim pode ser usado, exemplificando, para identificar se determinado e-mail é um SPAM ou sobre qual assunto se refere determinado texto. Em nosso contexto, classifica-se qual gênero é atribuído a um nome, baseando-se nos atributos supracitados.

Destarte, a acurácia também é vista como precisão geral do modelo e deve ser medida a partir do número de acertos em relação ao número total de testes, podendo ser apresentada em percentual para melhor interpretação. Essa medida é feita pela fórmula 1 a seguir, onde VP representa verdadeiros positivos e VN, verdadeiros negativos.

$$Acurcia = \frac{VP + VN}{Total} \tag{1}$$

Enquanto isso, o *F1-score* é uma métrica que combina precisão e recall para indicar a qualidade do modelo, até em cenários com conjunto de dados que não são proporcionais. A precisão é a métrica dos resultados efetivamente verdadeiros em relação ao total, e o *recall* é a frequência em que exemplos da classe são encontrados pelo classificador. O máximo do *score* é 1, que indicaria um modelo considerado perfeito (LEAL,2017). Para obter esse indicador deve-se calcular a partir da fórmula 2, a seguir.

$$F1 = \frac{2 * preciso * recall}{preciso + recall} \tag{2}$$

Como a base de dados escolhida apresentou desbalanceamento, utilizou-se da estratégia de *random oversampling* para evitar a tendência de produzir o modelo que favorecesse a classe majoritária (com maior ocorrência), fazendo com que o grupo minoritário não tenha a sua representatividade por erro de generalização (CASTRO, 2011). Ressalta-se, no entanto, que essa técnica tem suas limitações, tais como a replicação aleatoria de instancias que acarretam em resultados descontínuos na experimentação e a não avaliação de quais dados são mais relevantes para serem replicados, podendo gerar em *overlifting* em algum casos.

O ambiente utilizado para o desenvolvimento foi o *Google Colab*, sendo o *notebook* disponibilizado no *GitHub* em modo público e a linguagem de programação escolhida foi *Python*, pelo conhecimento prévio dos autores. As bibliotecas utilizadas foram *nltk* para manipulação dos dados em formato de texto, *pandas* e *numpy* para análises dos dados.

Todos os recursos e códigos produzidos nesta pesquisa, bem como o vídeo de apresentação, estão disponíveis em https://github.com/RafaelPassador/Classificacao-de-Genero-por-Nomes

III. Descrição dos dados utilizados

A base de dados utilizada foi obtida no site do *Machine Learning Repository*, do *Center for Machine Learning and Intelligent Systems* e pode ser encontrada pelo nome *"Gender by Name Data Set"*[1]. O *dataset* foi desenvolvido por Arun Rao da UC Berkeley e consta com 147270 instâncias.

Os atributos dos dados são os primeiros nomes associados aos gêneros, números de ocorrências e probabilidades. Tem como características o formato de texto, sendo 147270 instâncias e não possui *missing values* – tendo como possibilidade de tarefas de classificação e *clustering*.

O *dataset* combina contagens de quatro bases diferentes como os nomes femininos e masculinos em períodos específicos, então tem a probabilidade calculado a partir do agregado. As bases utilizadas são: i) Estados Unidos: *Baby Names from Social Security Card Applications – Nacional Data* (1880 a 2019); ii) Reino Unido: *Baby names in England and Wales Statistical bulletins* (2011 a 2018); iii) Canadá: *British Columbia 100 Years of Popular Baby Names* (1918 a 2018) e Austrália: *Popular Baby Namers, Attorney-General's Departament* (1944 a 2019).

A figura 1 explicita a distribuição dos dados conforme os rótulos de interesse nesse estudo, masculino e feminino. Dessa forma, é possível observar que há um balanceamento notável dos dados, em que há consideravelmente mais nomes da classe feminino do que dos masculinos.

Figura 1. Distribuição de Classes

[1]Disponível em: https://archive.ics.uci.edu/ml/datasets/Gender+by+Name

Dessa forma, após a utilização do *Random Oversampling*, a figura 2 retrata o novo conjunto de dados, com os nomes da classe minoritária, nomes masculinos, duplicados.

Figura 2. Distribuição de Classes Balanceadas

Por fim, a figura 3 explicita a função de *Random Oversampling* utilizada.

```
def oversampler(dataset):
    classes = dataset.Gender.value_counts().to_dict()
    maior = max(classes.values())
    classes_lista = []
    for key in classes:
        classes_lista.append(dataset[dataset['Gender'] == key])
    classes_exemplo = []
    for i in range(1,len(classes_lista)):
        classes_exemplo.append(classes_lista[i].sample(maior, replace=True))
    dataset_prov = pd.concat(classes_exemplo)
    final_dataset = pd.concat([dataset_prov,classes_lista[0]], axis=0)
    final_dataset = final_dataset.reset_index(drop=True)
    return final_dataset
```

Figura 3. Função de *Random Oversampling*

IV. Apresentação dos resultados

Nessa seção, apresentamos os resultados obtidos através do treinamento dos dois modelos compultacionais criados a partir das features e do classificador *Naive Bayes*.

A priori, analisamos o modelo contruido a partir do TF-IDF. Calculando o valor de incidência e utilizando-o como feature para o MultinomialNaiveBayes, transformamos os valores como *feature* para o classificador. Separamos os dados entre treinamento e teste, com os dados de treinamento apresenta-se os dados do "padrão ouro"e treina-se seu modelo, combinando a entrada com a saída esperada. Já com os dados de teste, estima-se o quão bem o modelo foi treinado. Utilizamos a divisão padrão da biblioteca sklearn, sendo 25% dos dados para treino e 75% dos dados para teste.

Nesse ínterim, obteve-se resultados moderados, com 50,12% de acurácia e 65,11% de $F1$ com os dados balanceados, estando, assim, dentro das expectativas considerando a reincidência dos nomes no conjunto de dados.

Não obstante, o segundo modelo constou com dois tamanhos distintos de amostras e três combinações de features gramaticais. Treinou-se modelos com amostras de 500 e 50.000 nomes, rearranjados a partir da extração de sufixos, prefixos e prefixos juntamente de sufixos. A partir disso, foi possível analisar em experimentos práticos que, em amostragens menores, os classificadores desempenham melhores tarefas, entrave que pode acarretar no enviesamento de resultados, uma vez que, amostras pequenas nem sempre representam o resultado real do conjunto em si.

Diante dessa perspectiva, nota-se que, em ambas configurações, o classificador obteve melhores resultados quando combinamos o prefixo e sufixo dos nomes para prever o seu gênero, obtendo 74,62% de acurácia e 76,85% de $F1$, no teste com 500 nomes e 69,78% de acurácia e 74,32% de $F1$ na amostra e 50.000 nomes. Em contrapartida, o classificador combinado com a extração de prefixos resultou nos piores índices de classificação entre todos os testes realizados, evidenciando que esse fator não corrobora para discernir o gênero entre os nomes.

A tabela 1 e a tabela 2 resumem as configurações que foram utilizadas, bem como seus respectivos resultados. Dessa forma, é possível analisar o impacto da perfomance da aplicação do *Random Oversampling* na base de dados, considerando como métricas avaliativas os valores de acurácia e $F1$.

Tabela I
Resultados de Acurácia e de $F1$ para cada Feature antes do Random Oversampling (R.O.S)

Configuração	Valor de Acurácia	Valor de $F1$
1 - MULTINaiveBayes+TF-IDF	56,14%	06.23%
2 - NaiveBayes+Sufix500	71,14%	69,98%
3 - NaiveBayes+PrefixSufix500	**74,62%**	**71,10%**
4 - NaiveBayes+Prefix500	35,21%	42,04%
5 - NaiveBayes+Sufix50000	29,96%	44,77%
6 - NaiveBayes+PrefixSufix50000	19,27%	35,80%
7 - NaiveBayes+Prefix50000	01,03%	03,01%

Tabela II
Resultados de Acurácia e de $F1$ para cada Feature antes do Random Oversampling (R.O.S)

Configuração	Valor de Acurácia	Valor de $F1$
1 - MULTINaiveBayes+TF-IDF	50,12%	65,11%
2 - NaiveBayes+Sufix500	**74,62%**	75,93%
3 - NaiveBayes+PrefixSufix500	**74,62%**	**76,85%**
4 - NaiveBayes+Prefix500	55,30%	57,51%
5 - NaiveBayes+Sufix50000	68,72%	73,67%
6 - NaiveBayes+PrefixSufix50000	69,78%	74,32%
7 - NaiveBayes+Prefix50000	12,83%	22,06%

Dessa forma, observa-se que contam com amostras de treinamento menores, no caso as que possuíam 500 nomes para rotulação (configurações 1, 2 e 3) , apresentam melhores resultados em relação às maiores, com 50.000 nomes (configurações 5, 6 e 7). Em relação as features, o TF-IDF obteve o pior desempenho em ambas tarefas, indicando que o valor de incidência dos nomes presentes no *corpus* não são um fator determinante para classificação.

Em outra perspectiva, utilizar *features* textuais, como a extração de sufixos e prefixos, obtiveram os melhores índices de acurácia e $F1$. Diante disso, reitera-se que o sufixo dos nomes são as *features* que mais corroboraram para a classificação de gênero. Nesse contexto, a figura 3 explicita as features mais

relevantes na configuração de melhor desempenho apresentada nesse trabalho, enquanto a figura 4 explicita a matriz de confusão dessa classificação (NaiveBayes+PrefixSufix500).

```
Most Informative Features
                sufixo2 = 'os'              1 : 0      =     33.9 : 1.0
                sufixo2 = 'ob'              1 : 0      =     28.3 : 1.0
                sufixo2 = 'yk'              1 : 0      =     23.1 : 1.0
                sufixo2 = 'io'              1 : 0      =     19.4 : 1.0
                sufixo2 = 'ib'              1 : 0      =     18.7 : 1.0
                sufixo2 = 'ik'              1 : 0      =     18.5 : 1.0
                sufixo2 = 'jr'              1 : 0      =     18.4 : 1.0
                sufixo2 = 'up'              1 : 0      =     17.8 : 1.0
                sufixo2 = 'ck'              1 : 0      =     17.6 : 1.0
                sufixo2 = 'rd'              1 : 0      =     16.8 : 1.0
                sufixo2 = 'ff'              1 : 0      =     16.5 : 1.0
                sufixo2 = 'bb'              1 : 0      =     16.4 : 1.0
                sufixo2 = 'av'              1 : 0      =     15.3 : 1.0
                sufixo2 = 'eq'              1 : 0      =     14.4 : 1.0
                sufixo2 = 'if'              1 : 0      =     14.4 : 1.0
```

Figura 4. Features Mais Relevantes no Modelo aiveBayes+Prefix500

Através matriz de confusão é possivel observar detalhadamente todos os aspectos da classificação, salientando-se os verdadeiros positivos e verdadeiros negativos, aqueles em que o modelo previu corretamente a classe e, também, os falsos positivos e falsos negativos, em que o classificador não previu a classe correta. Nessa perspectiva, sua importância consiste no contraponto com a acurácia, a qual leva em consideração o numero total de acertos, mas desconsidera os erros, sendo especialmente prejudicial em dados desbalanceados (vide tabela 1).

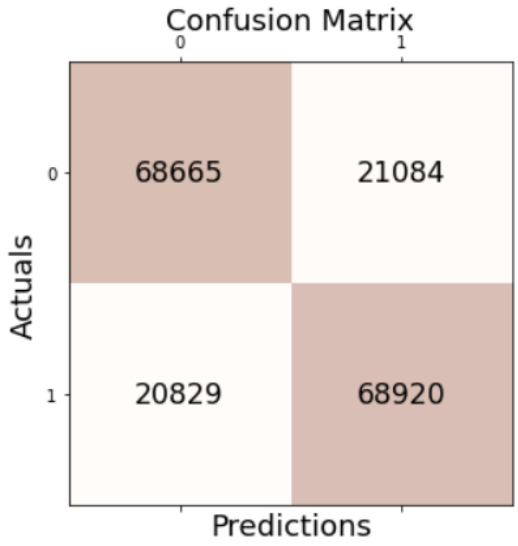

Figura 5. Matriz de Confusão do Modelo aiveBayes+Prefix500

V. Discussão

Nesta seção apresentamos uma breve discussão sobre os resultados obtidos e sobre os prós e contras dos métodos vistos nas seções anteriores.

O primeiro ponto a se destacar é a necessidade de balanceamento dos dados devido ao paradoxo da accurácia. Dados desbalanceados nem sempre mostram-se prejudiciais, como quando a classe minoritária é semelhante e pode se unir à outras classes para obter um número de instâncias equilibrado quando comparado à classe majoritária. Mas, quando as classes são opostas, como em nosso caso, os dados devem ser tratados de maneira a equilibrar a base de dados. Há diversas técnicas para o balanceamento de dados, algumas mais interessantes não eram viáveis para o nosso dataset, como a coleta de mais dados e como a técnica SMOTE que observa os dados obtidos e cria dados com características intermediárias baseadas nas instâncias já existentes.

Tendo esse fato em vista, recorremos a técnica da baleanceamento Random Oversampling que cria cópias de dados já existentes na classe minoritária e que pode ser boa quando a classe minoritária não tem uma variação quantitativa muito grande nos seus parâmetros. Por outro lado, o algoritmo também favorece o overfitting, quando observamos que o modelo é muito bom em classificar casos específicos. Outro ponto de atenção observado é que utilizando o Oversampling aleatório, a cada vez que executarmos nossa aplicação nomes diferentes serão duplicados e diferentes níveis de accuráracia e score podem ser observados.

Desta forma, para um dataset de nomes associados à gêneros, recomenda-se que sejam feitas coletas de dados balanceadas, pois métodos artificais de balanceamento podem levar à resultados inesperados.

Outro ponto de discussão se deve aos níveis de accuracia e F1-Score encontrados para cada feature que mostram o quão confiáveis são as classificaçõe obtidas em nosso modelo e quais features são mais importantes quando classificamos nomes da lingua inglêsa. Estes níveis são observados nas Tabelas I e II, apresentadas anteriormente. Foram aplicadas 4 features diferentes aos dados e tanto para os dados desbalanceados, quanto para os dados balanceados, diferenciando também o tamanho da base de dados utilizada. Observam-se como features mais confiáveis aquelas que classificam os nomes à partir do Prefixo com Sufixo e à partir do Sufixo, respectivamente. Para uma base de dados menor, com 500 instâncias os resultados se mostraram melhores, sobretudo quando observamos os dados desbalanceados. A feature com menor desempenho é a classificação por prefixo, que chegou a apresentar apenas 1,03% de accurácia na classificação desbalanceada de 50000 instâncias e 12,83% nas 50000 balanceadas. Este resultado valida a hipótese proposta por intuição de que ao observar as letras iniciais de um nome, pouco ou nada se conclui sobre o seu gênero relacionado, afinal, é mais comum diferenciar o gênero associado à um nome observando suas letras finais.

Ao comparar os resultados obtidos pela classificação do dataset menor (500) com a classificação do dataset maior (50000), o menor obtém maior accurácia e f1-score. Para os dados desbalanceados (Tabela I) isso se deve ao fato de que, para o menor dataset há a possibilidade de a diferença entre as classses observadas der menor. Enquanto para a classificação após Oversamplig, com os dados balanceados, observamos uma diferença menor entre os resultados para cada dataset,

mas ainda sim o datasetmenor se mostra mais confiável. Neste caso, podemos interpretar que a maior incidência de duplicatas pode diminuir a confiabilidade do modelo de classificação.

REFERÊNCIAS

[1] A., Tripathi; M., Faruqui. Gender prediction of Indian names. In: IEEE Technology Students' Symposium. IEEE, 2011. pags. 137-141.

[2] Base Gender by Name Data Set: banco de dados preparado por Arun Rao. In: Machine Learning Repository. Disponível: https://archive.ics.uci.edu/ml/datasets/Gender+by+Name. Acesso: 25 out.21.

[3] C.L. Castro, A.P, Braga. Aprendizado Supervisionado com Conjuntos de Dados Desbalanceados. Revista Controle Automação, vol.22 n.5, set/out 2011, pags 411-466. Disponível em: https://doi.org/10.1590/S0103-17592011000500002. Acesso em: 10 nov. 21.

[4] C., Tang et al. What's in a name: A study of names, gender inference, and gender behavior in facebook. In: International Conference on Database Systems for Advanced Applications. Springer, Berlin, Heidelberg, 2011. pags. 344-356.

[5] G, Coppersmith; R Leary.; P. Crutchley; A, Alex Fine. Natural Language In: Processing of Social Media as Screening for Suicide Risk. Biomedical Informatics Insights. v. 10, p. 1-11, 2018.

[6] J.C., Eichstaed; R.J., Smith; R.M., Merchant; L. H., Ungar; P. Crutchley; D., Preotiuc-Pietro; D.A., Asch; H.A,, Schwartz. Facebook language predicts depression in medical records. PNAS, v. 115, n. 44, out/18.

[7] M., De Choudhur; S., Counts; E., Horvitz. Social media as a measurement tool of depression in populations. In: Proceedings of the 5th Annual ACM Web Science Conference, p. 47-56, 2013.

[8] N., Bérubé et al. Wiki-Gendersort: Automatic gender detection using first names in Wikipedia. 2020.

[9] R. S, Leal. Métricas Comuns em Machine Learning: como analisar a qualidade de chat bots inteligentes – métricas. Disponível em: https://medium.com/as-m%C3%A1quinas-que-pensam/m%C3%A9tricas-comuns-em-machine-learning-como-analisar-a-qualidade-de-chat-bots-inteligentes-m%C3%A9tricas-1ba580d7cc96. Acesso em: 12 nov. 21.

[10] Y., Hu et al. What's in a name?–gender classification of names with character based machine learning models. Data Mining and Knowledge Discovery, pags. 1-27, 2021.