Comparison of Classifiers for the Cross-system Personalization problem in Social Networks

Comparação de Classificadores para o problema de *Cross-system Personalization* em Redes Sociais

João Marcos Mareto Calado[1], **Jefferson Oliveira Andrade**[1], **Karin Satie Komati**[1]

[1]Programa de Pós-graduação em Computação Aplicada
Campus Serra do Instituto Federal do Espírito Santo (IFES)
Rodovia ES-010 – Km 6,5 – Manguinhos CEP: 29173-087 – Serra – ES

{joao.calado, jefferson.andrade, kkomati}@ifes.edu.br

Abstract. *Virtual social networks open up the possibility of using users' public data to improve customization or anticipation of services, individualized marketing, or research of alumni of an educational institution to monitor their career in the market or their further studies. Crossing data of the same user between different social networks increase the knowledge about the user. However, different profiles of the same user on distinct social networks, often have inconsistencies, which makes cross-identification difficult. This article uses the machine learning approach to the problem of linking user profiles in different social networks and for this purpose comparatively studies eight classification algorithms, logistic regression, LDA, KNN, Decision Tree, Naïve Bayes, SVM, AdaBoost, and XGBoost, the one that obtains the best performance to link user profiles in different databases. It was possible to replicate, and overcome in some cases, the accuracy results reported in the literature. In the best scenario, Adaboost and XGBoost achieved an accuracy of 0.96.*
Keywords: *Cross-system Personalization, Social Networks, Classifiers.*

Resumo. *As redes sociais virtuais abrem a possibilidade de usar os dados públicos dos usuários para melhorar a customização ou antecipação de serviços por empresas, marketing individualizado, ou pesquisa de egressos de uma instituição de ensino para acompanhamento de sua carreira no mercado ou da continuação dos seus estudos. Cruzar dados do mesmo usuário entre diferentes redes sociais amplia o conhecimento acerca do usuário. Entretanto, diferentes perfis de um mesmo usuário em redes sociais distintas, frequentemente apresentam inconsistências entre campos, tais como não apresentarem o mesmo nome ou o mesmo endereço, o que torna a identificação cruzada difícil. Este artigo usa a abordagem de aprendizado de máquina para o problema de ligação de perfis de usuários em diferentes redes sociais e para tal estuda comparativamente oito algoritmos de classificação, regressão logística, LDA, KNN, Árvore de decisão, Naïve Bayes, SVM, AdaBoost e XGBoost. Conseguiu-se replicar, e superar em alguns casos, os resultados de acurácia relatados na literatura, com acurácia de 0,96 no melhor caso com AdaBoost e XGBoost.*
Palavras-chave: *Identificação em sistemas cruzados, Redes sociais, Classificadores.*

1. Introdução

Atualmente as pessoas passam um tempo considerável nas redes sociais, criando um ambiente virtual onde elas se conectam à amigos, compartilham informações e expandem os laços sociais. Ter a capacidade de ligar os perfis entre diversas bases de dados poderia levar a um maior entendimento sobre o comportamento e costume dos usuários, permitindo a melhora na provisão e customização de serviços, além de recomendações melhores [Carmagnola and Cena 2009].

Este é um problema que possui diferentes termos, tais como, a identificação de usuário em sistemas cruzados (em inglês *Cross-system Personalisation*) [Carmagnola and Cena 2009] [Esfandyari et al. 2018], ou como *Social Link Identification* [Zhang et al. 2019] [Shu et al. 2017] ou como *Disambiguate Identity References* [Rowe 2009] [Digiampietri et al. 2015], dentre outros.

O problema de identificação de usuário em sistemas cruzados é ilustrado na Figura 1, que apresenta os diferentes campos correspondentes entre dois perfis da mesma pessoa em dois *sites* diferentes: do Twitter e da plataforma Lattes. Na figura é possível verificar que existem alguns dados (não todos) em comum em posições diferentes, e provavelmente com nomes de campos diferentes entre os sistemas[1].

Figura 1. Dois perfis da mesma pessoa em *sites* diferentes, evidenciando informações comuns entre os diferentes perfis.

O problema de identificação de usuário em sistemas cruzados é definido formalmente da seguinte forma: dados dois perfis P^{s_1} e P^{s_2} de duas redes sociais diferentes s_1 e s_2, determinar se estes perfis pertencem à mesma pessoa. Isto corresponde a aprender

[1] A figura é meramente ilustrativa, assim, poderiam ser outras páginas que não a do Lattes e a do Twitter.

uma função de identificação $f(P^{s_1}, P^{s_2})$, tal que:

$$f(P^{s_1}, P^{s_2}) = \begin{cases} 1 & \text{se } P^{s_1} \text{ e } P^{s_2} \text{ pertencem à mesma pessoa} \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

Neste tipo de problema, a extração de características se diferencia da extração de características de um texto ou de uma imagem, em que o vetor de características de uma única amostra é repassado diretamente ao classificador. Neste caso, os perfis devem ser "pareados" através de métricas de distância ou similaridade, sendo criado um vetor com estas métricas a partir de dois perfis. É este vetor de similaridade/distância entre os campos de dois perfis que será a entrada para algoritmos de aprendizado de máquina, tanto no treinamento quanto na classificação.

Cada par de atributos correspondentes, um de cada perfil de entrada, passa por um processo de extração de características. Por exemplo, usar uma métrica de comparação de texto como a cadeia mais longa em comum dos nomes de usuário dos dois perfis, de cada rede social. O mesmo indivíduo pode usar diferentes nomes de usuários em redes diferentes, assim em uma rede ele pode se chamar "Fulano" e na outra rede, pode-se chamar "Fulano92", o resultado da comparação é de 6, pois são 6 caracteres ("Fulano") que formam a cadeia mais longa em comum. Também deve-se ter pelo menos uma outra métrica para demonstrar o quanto os dois nomes são diferentes, que poderia ter o valor 0,25, indicando que são 2 caracteres diferentes dividido pelo maior comprimento dos dois nomes (que é 8).

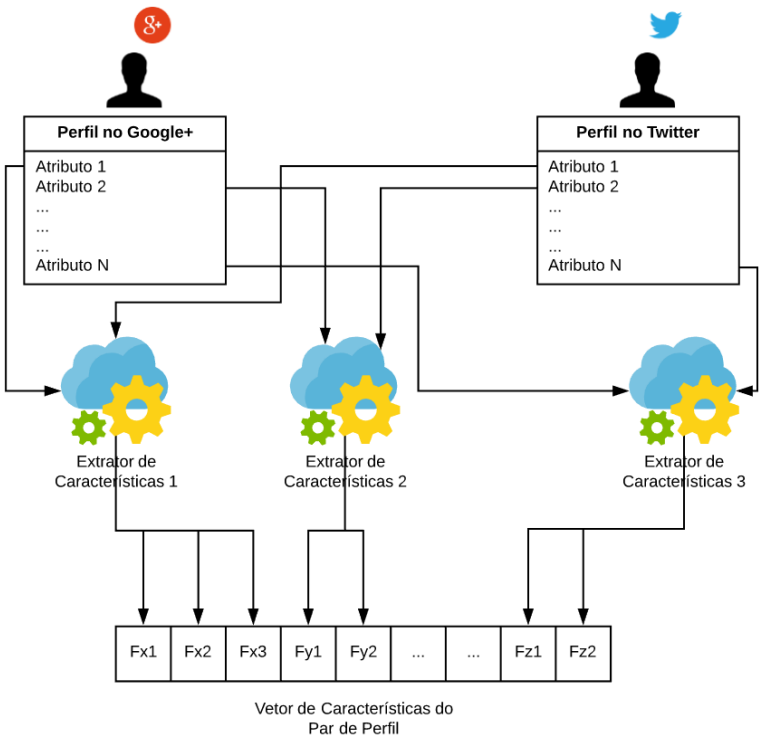

Figura 2. Arquitetura da Extração de Características dos Perfis.

A Figura 1 ilustra um exemplo em que dois perfis, um do Google+ e outro do Twitter, possuem um conjunto de atributos: "Atributo 1", "Atributo 2" até "Atributo N".

A quantidade de atributos em cada perfil pode ser diferente, de modo que é possível que alguns atributos não sejam usados para a extração de características. Para cada par de atributos selecionados, é calculada a distância/similaridade entre os valores, gerando uma ou mais posições do vetor de características. Na Figura 1, por exemplo, o "Extrator de Características 1" extrai 3 características do par "Atributo 1" do Google+ e "Atributo 1" do Twitter, enquanto o "Extrator de Características 2" extrai 2.

Na abordagem de aprendizado de máquina, na fase de treinamento, cada vetor de características transformado terá um rótulo "positivo" ou "negativo", indicando respectivamente se o par de perfis correspondem à mesma pessoa (instância "positiva") ou não (instância "negativo"). Na fase de testes, o modelo do classificador responderá positivo ou negativo.

Nesse sentido, a proposta deste artigo é comparar, entre 8 (oito) algoritmos de classificação, regressão logística, LDA (do inglês *Linear Discriminant Analysis*), KNN (do inglês *k-nearest neighbors*, em português k-Vizinhos mais próximos), árvore de decisão, Naïve Bayes, SVM (do inglês *Support Vector Machine*), AdaBoost e XGBoost, aquele que obtém o melhor desempenho na tarefa de ligação de perfis de usuários em diferentes bases de dados. Será usado um conjunto de características a partir do perfil dos usuários proposto por Esfandyari e colegas [Esfandyari et al. 2018], que após extraídas, serão usadas como conjunto de entrada para os classificadores, além da mesma base de dados denominada "GT dataset", que contém 10.571 pares de perfis corretamente rotulados de usuários do *Google+* e *Twitter*.

A diferença é que o trabalho de Esfandyari e colegas [Esfandyari et al. 2018] comparava os resultados de dois classificadores: Floresta Aleatória e MLP (*Multilayer Perceptron*), e este trabalho é uma contribuição incremental e faz a comparação usando outros 8 (oito) algoritmos de classificação. Esta escolha se deveu ao fato de ter sido o único trabalho, dentre os citados nos trabalhos correlatos, que disponibilizou publicamente a base de dados, o que seria possível comparação de resultados.

O restante deste trabalho está organizado da forma a seguir, na Seção 2 são apresentados os trabalhos correlatos. Na Seção 3 estão descritos detalhadamente, a base de dados, o processo de extração de características a partir dos perfis da base de dados e os algoritmos de classificação. Na Seção 4 são apresentados os resultados encontrados neste experimento e por fim, na Seção 5 são apresentadas as considerações finais deste trabalho.

2. Trabalhos Correlatos

As redes sociais vêm sendo objeto de estudos desde bem antes do advento das redes sociais digitais, como visto em Wasserman e Galaskiewicz [Wasserman and Galaskiewicz 1994]. O autor ainda cita questões como influência, popularidade, interação entre pessoas, e mesmo propagação de doenças já eram estudados como fenômenos sociais desde antes do surgimento da internet comercial nos EUA. Ainda no ano de 1997 foi lançado o primeiro site do tipo rede social chamado sixdegrees.com, conforme informado no trabalho de Boyd e Ellison [Boyd and Ellison 2007].

Porém, foi somente a partir de 2003 que os maiores sites em termos de quantidade de usuários começaram a ser lançados, como Linkedin e MySpace em 2003, Youtube em 2005 e Facebook aberto a todos em 2006. Dados de outubro de 2018 mostram que as

maiores redes sociais são o Facebook com 2,6 bilhões de usuários em suas plataformas; o YouTube com 1,9 bilhões e o Instagram com 1 bilhão de pessoas [Valente 2018].

O trabalho de Veldman [Veldman 2009] já havia demonstrado que os usuários costumam criar contas em diversas redes sociais online e passar bastante tempo nestas atividades. Esfandyari [Esfandyari et al. 2018] ainda complementa dizendo que as pessoas gastam parte de suas vidas sociais na web, criando um ambiente onde podem encontrar amigos, compartilhar e criar informações e se engajarem uns com os outros.

A identificação de que dois perfis em diferentes redes sociais pertencem à mesma pessoa no mundo real é um problema difícil dada a não estruturação das informações, além da falta de garantia na veracidade das informações preenchidas [Esfandyari et al. 2018]. São elencados dois motivos pelos quais existe esse desafio [Shu et al. 2017]: o primeiro é que embora usuários tenham contas em diferentes redes, a informação de uma mesma pessoa no mundo real pode ser diversa entre as redes, e o segundo motivo é que as informações da identidade dos usuários é ruidosa, incompleta e altamente não estruturada.

A forma como as soluções estão divididas varia de autor para autor e, apesar de serem similares, possuem suas diferenças. O trabalho conduzido por Shu et al. [Shu et al. 2017] detalha a extração de característica dos usuários em 3 formas a saber: (i) características de perfil, (ii) características de conteúdo e (iii) características de rede de amizades. Esfandyari et al. [Esfandyari et al. 2018] por sua vez, fez um estudo das pesquisas para identificação de um mesmo usuário em diferentes sites de redes sociais e agrupou as abordagens de seguinte forma: (i) baseada em nome do usuário, (ii) baseada em atributos de perfil e, (iii) baseada em conteúdo e rede de amizades. Já Deng e colegas [Deng et al. 2019] divide as soluções em (i) baseada em atributos de perfil, (ii) baseada em estrutura de redes de relacionamentos, (iii) baseada na geração de dados, como posts.

Neste trabalho optou-se pela divisão de Esfandyari et al. [Esfandyari et al. 2018] e nas próximas subseções serão detalhadas as abordagens baseadas em nome de usuário e baseadas em atributos de perfil, que serão utilizadas nesse trabalho.

2.1. Identificação Baseada em Nome

Na identificação baseada em nome de usuário, as soluções levam em consideração apenas o nome de usuário, consequentemente os diferentes métodos dependem apenas das similaridades e diferenças extraídas das *strings* que compõem os nomes.

O trabalho de Zafarani e Liu [Zafarani and Liu 2009] demonstrou a possibilidade de identificar perfis correspondentes em doze comunidades, utilizando nomes de usuários e um motor de busca, no caso, o Google. A técnica começa pesquisando pelo nome de um determinado usuário no Google tentando encontrar um conjunto de palavras-chave que podem representar possíveis nomes de usuários nas redes sociais alvo. Então este conjunto é estendido, adicionando ou removendo prefixos e sufixos comuns de seus membros. O resultado no Google+ teve uma precisão de 0,66.

Posteriormente, os autores [Zafarani and Liu 2013] propuseram uma metodologia denominada "Modeling Behavior for Identifying Users" (MOBIUS) baseada nos padrões comportamentais dos usuários quando estes selecionam seus nomes de usuários, que podem ser cadeias de caracteres alfanuméricas ou *e-mails*. É importante que um mesmo in-

divíduo pode ter diferentes contas de *e-mails* e associar cada conta de rede social com um *e-mail* diferente. O trabalho demonstrou que o ambiente, a personalidade e as limitações humanas resultam em escolhas de nomes redundantes, e que a identificação por nomes é possível.

Perito e colegas [Perito et al. 2011] estimaram o quão único é um nome utilizando teoria de modelo de linguagem e Cadeias de Markov. Para cada nome, o classificador checa todos os possíveis nomes numa lista de similaridades. Foram feitos testes em 3 bases de dados, a INRIA, Google e MySpace, e os autores concluem que a grande maioria dos usuários pode ser identificada exclusivamente pelo nome do usuário em uma base de dados de 1 bilhão de usuários. Infelizmente, os autores não disponibilizaram a base de dados de forma pública e *online* para uso.

No trabalho de Li e colegas [Li et al. 2017], é mostrada uma solução para composição de uma série de características extraídas dos nomes de usuários. As bases de dados foram construídas pelos autores (não disponível para *download*), uma base com perfis do Facebook-Twitter com 67.826 perfis, outra de Facebook-Foursquare com 288.480 perfis e Foursquare-Twitter com 102.315 perfis. O trabalho utilizou sete classificadores: Gaussian Naïve Bayes, Bernoulli Naïve Bayes, regressão logística, regressão logística com validação cruzada embarcada, SVM, árvore de decisão e floresta aleatória, via algoritmos da biblioteca scikit-learn com parametrização padrão [Pedregosa et al. 2011]. O experimento mostra que a solução proposta apresentou medida-F1 atingindo $96{,}24\,\%$, $92{,}49\,\%$, e $90{,}68\,\%$ em três conjuntos de dados reais diferentes, respectivamente.

2.2. Identificação Baseada em Perfil

A identificação baseada em atributos de perfil leva em consideração o conjunto de atributos do perfil de usuários disponíveis publicamente nos sites de redes sociais, além do nome do usuário. As abordagens empregadas podem ser categorizadas em baseadas em distância e baseadas em frequência [Shu et al. 2017].

Os trabalhos envolvendo ligação de perfis de redes sociais lidam basicamente com dados que são cadeia de caracteres ou nominais, que se baseiam principalmente na avaliação da semelhança por similaridade de strings, seja baseada na distância ou padrões de frequência de repetição. Para os métodos baseados em distância ou similaridade entre os componentes dos perfis a serem analisados, a "distância" pode ser medida por meio de métricas como *Jaro-Winkler distance*, *Jaccard similarity* e *Levenshtein distance* e informam a distância entre diferentes valores para os atributos de determinada entidade. Essa distância é então utilizada como indicativo da semelhança entre os valores. Para os métodos que consideram a frequência, ao invés de calcular a distância entre os valores de cada atributo dos perfis envolvidos no processo de identificação, deve-se investigar o padrão de frequência. Neste modelo, o texto é separado utilizando a técnica *bag of words* e/ou *TF-IDF* (abreviação do inglês *Term Trequency-Inverse Document Frequency*, que significa frequência do termo–inverso da frequência nos documentos) [Cohen et al. 2003].

Alguns autores, no entanto, atacaram este problema de uma forma um pouco diferente, como Carmagnola e Cena [Carmagnola and Cena 2009], que definiram um conjunto de propriedades dos perfis e fatores de importância específicos para cada proprie-

dade. Assim, o algoritmo proposto por eles, compara os atributos de perfil levando em consideração o fator de importância. Apesar do teste ter sido feito numa base de dados com 80 usuários, obteve relativo sucesso, com 59 de um total de 64 casos sendo corretamente identificados, e 2 de um total de 16 casos sendo incorretamente marcados como não identificados.

Vosecky e colegas [Vosecky et al. 2009] usaram uma abordagem baseada no estabelecimento de valores que indiquem pesos para os campos dos perfis envolvidos na comparação. Dessa forma, é possível controlar a influência que cada propriedade tem no processo de classificação de similaridade dos perfis, e para comparar os atributos, foram utilizadas técnicas como *exact*, *partial* e *fuzzy match*. A base de dados envolveu mil usuários do Facebook e do StudiVZ, e ao final do processo, os autores conseguiram uma taxa de sucesso de $83\,\%$.

3. Materiais e Métodos

Nesta seção serão descritos a base de dados e o processo de extração de características dos perfis. Além disso, serão descritos quais foram os hiper-parâmetros usados nos classificadores e a descrição da métrica de avaliação dos resultados.

3.1. A Base de Dados "GT dataset"

Para o experimento, foi utilizada uma base de dados pública, disponível no portal do Laboratório de Protocolo de Redes e Tecnologias (NPTLab) da Universidade de Milão[2] [Esfandyari et al. 2018]. A base de dados, denominada "GT dataset", contém pares de perfis corretamente rotulados de usuários do *Google+* e *Twitter*.

Esfandyari e colegas elaboraram a base de dados da seguinte forma, após uma raspagem de dados nas redes sociais, a base inicial continha 10.571 pares de perfis corretamente rotulados de usuários do *Google+* e *Twitter*. A etapa de limpeza de dados removeu registros cujos atributos possuem valor nulo ou cadeia de caracteres vazia e após esta etapa, sobraram 1.981 registros válidos.

Esta base de dados disponibiliza 15 atributos em cada registro, sendo 6 do perfil do *Google+*, 8 do perfil do *Twitter* e um único atributo de identificação no registro da base de dados. Os atributos são listados na Tabela 1.

Apesar da base de dados conter 15 atributos, apenas os atributos em comum, ou seja, que existem na rede social *Google+* e *Twitter* serão utilizados para a extração de características e consequente associação de perfis. Os atributos que serão pareados são:

- concatenação do *G_Firstname* e *G_Lastname* com um espaço em branco entre eles pareado com o campo *T_Fullname*;
- *G_Displayname* pareado com *T_ScreenName*;
- *G_Location* pareado com *T_Location*;
- *G_aboutme* pareado com *T_Description*.

[2]No momento da pesquisa, o site do Laboratório de Protocolo de Redes e Tecnologias da Universidade de Milão passou por uma alteração e os *links* de *download* da base de dados deixaram de funcionar, porém ainda é possível acessar os arquivos diretamente pela URL http://nptlab.di.unimi.it/wp-content/uploads/datasetUserIdentification.zip

Tabela 1. Atributos da base GT Dataset

Nome do atributo	Descrição
_id	Atributo identificador do registro na base de dados
Gid	Atributo identificador do perfil no Google+
G_Firstname	Atributo que representa o primeiro nome do usuário no Google+
G_Lastname	Atributo que representa o último nome do usuário no Google+
G_Displayname	Atributo que representa o nome de usuário no Google+
G_Location	Atributo que representa a localização do usuário no Google+
G_aboutme	Atributo que contém uma descrição a respeito do usuário no Google+
Tid	Atributo identificador do usuário no Twitter
T_Fullname	Atributo que representa o nome completo do usuário no Twitter
T_ScreenName	Atributo que representa o nome de usuário no Twitter
T_Location	Atributo que representa a localização do usuário no Twitter
T_Description	Atributo que contém uma descrição a respeito do usuário no Twitter
T_Time_Zone	Atributo que representa a zona de horário do usuário no Twitter
T_StatusText	Atributo que representa um texto breve a respeito do estado do usuário no Twitter
T_Language	Atributo que representa a língua do usuário no Twitter

Assim, os três atributos de identificadores não serão usados, *_id*, *Gid* e *Tid*. Nem os atributos específicos do *Twitter*: *T_Time_Zone*, *T_StatusText* e *T_Language*.

Como todas as instâncias da base "GT dataset" são positivas, isto é, instâncias cujos pares de perfis foram corretamente identificados como pertencentes ao mesmo indivíduo, as instâncias negativas foram criadas pelo método descrito em [Esfandyari et al. 2018]. A técnica envolve a criação aleatória de pares P_i^{s1}, P_j^{s2} tal que P_i^{s1} seja o perfil do usuário *i* na rede social *s1* de uma instância positiva, e o P_j^{s2} seja o perfil do usuário *j* na rede social *s2* de uma outra instância positiva, sendo que $i \neq j$.

Os autores da base elaboraram três conjuntos de treinos, cada um com um nível de dificuldade diferente para os classificadores. Todos os conjuntos de treino são balanceados com 50% das instâncias positivas e 50% das instâncias negativas, diferenciando na forma como as negativas foram selecionadas.

- Treino 1: as instâncias negativas são selecionadas de forma randômica. O tamanho deste conjunto de dados é de 3.500 registros;
- Treino 2: a construção de instâncias negativas teve como objetivo obter um nível de dificuldade maior, 50% das instâncias negativas foram obtidas de forma aleatória e 50% são construídas para que cada par negativo tenha valores similares em ao menos um atributo. O tamanho deste conjunto de dados é de 3.540 registros;
- Treino 3: foi elaborado a fim de se obter um conjunto de treino mais difícil. Nele, todas as instâncias negativas são construídas de forma que cada par negativo tenha valores similares em ao menos um atributo. O tamanho deste conjunto de dados é de 3.550 registros.

Também elaboraram dois conjuntos de testes:

- Teste 1: inclui 50% de instâncias positivas e 50% das instâncias negativas construídas de forma aleatória. Contém 870 registros;
- Teste 2: inclui instâncias positivas que possuem ao menos 1 atributo diferente, enquanto todas as instâncias negativas são construídas de forma a ter valores iguais

em ao menos um atributo, assim como no conjunto de treino 2. Contém 663 registros.

Estes conjuntos de treino e teste, que estão disponíveis para *download*[3], é que serão utilizados para a extração de características e posterior análise por meio de algoritmos classificadores.

3.2. Extração de Características

A etapa de extração de características tem como entrada dois perfis pareados e tem como saída um único vetor de características. Conforme descrito sobre a "GT dataset", um perfil é da rede social *Google+* e outro perfil é do *Twitter*. Cada perfil possui vários atributos. O que se deseja é verificar se há dois perfis que correspondem à mesma pessoa.

Para os atributos que são correspondentes, cada par de atributo, um de cada perfil de entrada passa por um processo de extração de características. Para cada par de atributos selecionados, é calculada a similaridade/distância entre os valores, gerando uma ou mais posições do vetor de característica.

Na etapa de extração de características, foram empregadas as seguintes medidas de similaridade/distância:

- *Exact Match* (EM): comparação exata dos dois valores de entrada;
- *Longest Common Substring* (LCS): a cadeia mais longa em comum. Em geral, este valor é normalizado, dividindo-se pela média do tamanhos das duas *strings* de entrada;
- *Longest Common Sub-Sequence* (LCSS): medida parecida com a LCS, porém de forma que a sequência não precise ser contígua. Novamente o valor de retorno é normalizado pela média do tamanho das duas strings originais;
- *Levenshtein Distance* (LD): o algoritmo de Levenshtein calcula o número mínimo de operações de edição que são necessárias para modificar uma *string* de forma à obter outra *string*;
- *Jaccard Similarity* (JS): é o cálculo do tamanho da interseção de termos (por exemplo, palavras) dividida pelo tamanho da união dos conjuntos dos termos das entradas e;
- *Cosine Similarity* (CS) *with* TF-IDF *weights*: Esta é uma técnica bem conhecida de recuperação de informação, que mede a similaridade entre dois conjuntos de textos. Primeiramente são calculados os pesos TF-IDF e posteriormente esses pesos servem de entrada para a similaridade de cosseno. Os termos TF e IDF vêm do inglês e significam respectivamente frequência do termo e inverso da frequência nos documentos. Enquanto TF mede o número de vezes que um termo (palavra) aparece em cada texto, IDF tenta dar importância aos termos com base na frequência em que aparecem nos textos de entrada e é calculado com base no número de textos e no número de textos que contém o termo a ser pesquisado. Após os pesos serem calculados para cada termo dos dois textos, eles são multiplicados e os resultados são armazenados em dois vetores, um vetor dedicado a cada texto de entrada. A similaridade entre os vetores então é dada pelo produtos desses dois vetores, medindo o cosseno do ângulo entre eles no espaço vetorial [Tata and Patel 2007].

[3]O *link* para *download* é http://nptlab.di.unimi.it/wp-content/uploads/GoogleTwitterTrainTest.zip

As métricas foram aplicadas da seguinte forma, resultando em um vetor com 14 características (Figura 3.2):

- 5 características, usando EM, LCS, LCSS, LD e JS ao par de campos *G_Displayname* e *T_ScreenName*;
- 5 características, usando EM, LCS, LCSS, LD e JS ao par de campos *T_Fullname* e concatenação dos campos *G_Firstname* e *G_Lastname* com um espaço em branco;
- 3 características, usando as métricas EM, LCS, e JS entre os campos *G_Location* e *T_Location*;
- 1 característica pela métrica CS entre os campos *G_Aboutme* e *T_Description* dos registros.

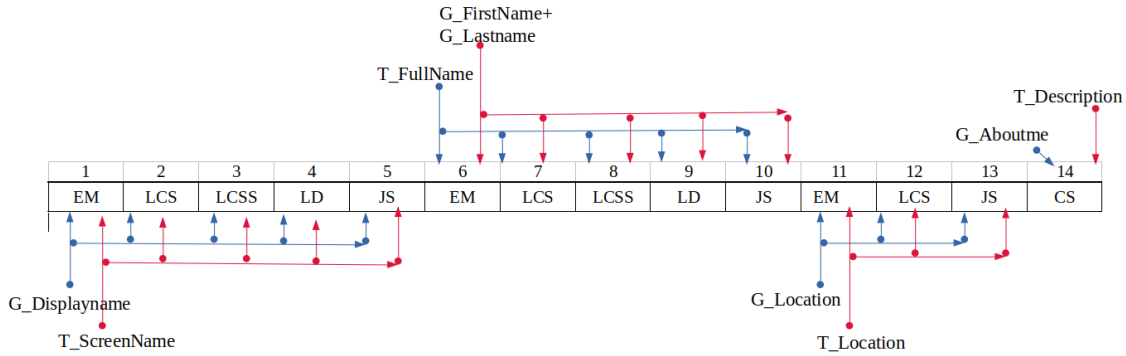

Figura 3. Vetor de características com 14 posições.

Na fase de treinamento, cada vetor de características transformado terá um rótulo "positivo" ou "negativo", indicando respectivamente se o par de perfis é da mesma pessoa ("positivo") ou não ("negativo"). Na fase de testes, deve-se entregar um vetor transformado, cujo modelo do classificador responderá positivo ou negativo.

3.3. Classificadores

Os modelos preditivos clássicos podem ser baseados em distância, probabilístico, otimização e procura [Faceli et al. 2000]. Assim, foram selecionados pelo menos um de cada tipo, o KNN que é baseado em distância [Cover and Hart 1967], o SVM baseado em otimização [Boser et al. 1992], a árvore de decisão baseado em procura [Safavian and Landgrebe 1991] e três baseados em modelo probabilístico, o Naïve Bayes, a Regressão Logística e o LDA [Schütze et al. 1995, Li and Jain 1998]. Além disso, foram selecionados dois métodos do tipo *ensemble*, o AdaBoost [Freund et al. 1999] e o XGBoost [Chen and Guestrin 2016].

Todo o código foi desenvolvido na linguagem Python 3.6 e com exceção do classificador XGBoost, todos os classificadores utilizados neste experimento são provenientes do pacote scikit-learn[4].A seguir há uma breve descrição dos valores utilizados nos parâmetros em cada algoritmo classificador. Não houve etapa de redução de dimensionalidade.

Baseado em distância, KNN efetua a classificação a partir de uma votação por maioria simples dos vizinhos mais próximos de cada ponto. Classificar um ponto, utilizando o KNN, pode ser resumido em três passos, segundo Neto et al. [Neto et al. 2017],

[4]https://scikit-learn.org/stable/

que são: (i) o cálculo da distância entre o exemplo que não é conhecido, com os demais exemplos do conjunto de treinamento, (ii) a identificação dos K vizinhos mais próximos e (iii) a utilização do rótulo da classe de vizinhos mais próximos para determinar o rótulo do exemplo desconhecido, usando um sistema de votação. Nos experimentos, foram usados os seguintes parâmetros: 5 vizinhos ($K = 5$); todos os 5 vizinhos tem o mesmo poder de voto, isto é, não contém pesos diferentes, independente da distância; a distância usada foi a de Minkowski com potência 2 (que é o mesmo que a distância euclidiana).

Baseado em otimização, SVM constrói hiperplanos para separar diferentes classes no espaço. Esse algoritmo tem como vantagem ser eficaz em dados com muitas dimensões, mesmo que haja um número de instâncias menor do que o total de dimensões [Ceravolo et al. 2019]. O SVM tradicional é um classificador binário, ou seja, classifica apenas duas classes. Para implementar um classificador multi-classes, podem ser empregadas a abordagem Um-Contra-Um (ovo, do inglês *One-vs-One*) e Um-Contra-Todos (ovr, do inglês *One Verse Rest*) [Pal 2008]. A opção usada foi a 'ovr', nesta abordagem são construídos classificadores binários que distinguem entre uma determinada classe entre as demais. Neste caso, se existem N classes, são construídos N classificadores binários, onde cada classificador é treinado para distinguir uma entre as outras $N - 1$ classes. Para novas instâncias não rotuladas, é aplicada a abordagem do "o-vencedor-ganha", onde o classificador com maior margem que separa os hiperplanos, é o vencedor.

Baseado em teoria probabilística, o Naïve-Bayes, trabalha com a ideia de independência de atributos. Sendo considerado como ingênuo (*Naïve* em inglês), desconsidera a associação entre os atributos e os analisa como condicionalmente independentes. Naïve-Bayes oferece bons resultados quando se tem disponível um conjunto de treinamento médio ou grande. Já o Processo Gaussiano é um método que utiliza distribuições de probabilidade para estimar a classe de um ponto através de inferência Bayesiana [Rasmussen 2003].

Na Regressão logística o parâmetro C controla o *trade-off* entre um limite de decisão suave e a classificação correta dos dados de treinamento. Aumentar o valor de C pode levar ao *overfitting*. O valor utilizado foi de 1. O algoritmo de classificação foi o algoritmo Broyden–Fletcher–Goldfarb–Shanno com memória limitada como resolvedor. Além disso, o número máximo de iterações que o algoritmo resolvedor poderá executar para convergir para a solução da classificação foi definido como 100 e o erro de critério de parada da convergência é de $0{,}0001$. O LDA usa o método *Singular Value Decomposition* para realizar a classificação.

Baseada em busca, a árvore de decisão classifica os dados com base em regras inferidas a partir de seus atributos, tendo como vantagem facilitar o entendimento do modelo, bem como dos atributos mais relevantes. De modo geral, no método de árvore de decisão, o problema é representado como uma árvore, onde cada nó de entrada é analisado e dividido [Rodrigues et al. 2018]. Não houve limitação quanto à profundidade da árvore, também não houve limitação do número máximo de características a serem consideradas no processo de *split*, nem o número máximo de nós folhas. O número mínimo de amostras em um nó folha foi de 1 e o número mínimo de amostras necessárias para separar um nó foi de 2.

A técnica é chamada de *ensemble* quando um conjunto de classificadores é trei-

nado individualmente mas as decisões são tomadas de forma combinada. Métodos *ensemble* tendem a apresentar um menor *overfitting*. O algoritmo AdaBoost é um método *ensemble* em série, o classificador base foi Árvore de Decisão com profundidade máxima igual 1. Nesta implementação, tem-se no máximo 50 árvores, sendo que todas as árvores têm pesos iguais. Foi definido que cada árvore de decisão tem o mesmo peso no resultado da classificação.

XGBoost é uma técnica que tem se destacado pelo bom desempenho em competições envolvendo classificação de dados. É um algoritmo do tipo comitê que utiliza a técnica de *boosting* e *gradient-descendent* na criação das árvores de forma a minimizar o erro na direção do gradiente. Nesta implementação, o modelo foi baseado em árvore com profundidade máxima de 3, usando no máximo 100 árvores (parâmetros padrões da implementação do scikit-learn).

3.4. Métrica de Avaliação

No campo da aprendizagem de máquina, uma matriz de confusão, também conhecida como matriz de erro, é um *layout* de tabela específico que permite a visualização do desempenho de um algoritmo de aprendizado supervisionado [Stehman 1997]. Cada linha da matriz representa as instâncias em uma classe prevista, enquanto cada coluna representa as instâncias em uma classe real [Powers 2011].

O exemplo de uma matriz de confusão é mostrado na Tabela 2, onde foi utilizado como exemplo, a classificação binária entre POSITIVO e NEGATIVO, conceito presente neste trabalho. Onde positivo é quando os dois perfis pareados são da mesma pessoa e negativo quando não são.

Tabela 2. Exemplo de uma matriz de confusão

		REAIS	
		POSITIVO	NEGATIVO
PREDITAS	POSITIVO	VP	FP
	NEGATIVO	FN	VN

Os valores desejados estão na diagonal principal da matriz, onde a predição é realizada de forma correta para ambos os casos: se a instância for positiva e classificada como positiva, é contada como um Verdadeiro Positivo (VP) e se a instância for negativa e classificada como negativa, ela será contada como um Verdadeiro Negativo (VN). Os erros estão fora da diagonal principal: se a instância for positiva e for predita como negativa, é contada como um Falso Negativo (FN) e; se a instância for negativa, mas for predita como positiva, é contada como Falso Positivo (FP).

A métrica acurácia é indicada quando o *dataset* está balanceado, onde o número de classes positivas e negativas são iguais. Esta métrica foi selecionada, pois é a usada pelo trabalho de Esfandyari e colegas [Esfandyari et al. 2018], o que possibilita a comparação de resultados. É dada pela Equação 2.

$$acurácia = \frac{VP + VN}{VP + FP + FN + VN} \tag{2}$$

Precisão (em inglês *precision*) é a taxa de observações corretamente preditas como verdadeiras em relação ao total de observações preditas como verdadeiras. Neste trabalho, essa métrica responde à pergunta de quantas associações de perfis ditas como verdadeira, quantas foram corretas? Altas taxas de precisão estão relacionadas a baixas taxas de falsos positivos. É dada pela Equação 3.

$$precisão = \frac{VP}{VP + FP} \tag{3}$$

A sensibilidade, também conhecido em algumas literaturas como revocação (em inglês *recall*), consiste na proporção de verdadeiros positivos frente à todos os casos positivos existentes, isto é, a soma da quantidade de verdadeiros positivos, divididos pela soma dos verdadeiros positivos mais os falsos negativos, conforme mostrado na Equação 4.

$$sensibilidade = \frac{VP}{VP + FN} \tag{4}$$

4. Experimentos e Resultados

Neste artigo, testa-se o desempenho dos classificadores usando a extração de características baseada em similaridade de atributos de perfil do tipo *string*, na resolução do problema de associação de perfis em redes sociais da base "GT Dataset".

Para comparação dos resultados, foram utilizadas as métricas acurácia, precisão, sensibilidade e o tempo de execução dos testes. Todo o código gerado para o experimento está disponível em um repositório no GitHub[5]. Os experimentos foram executados em um computador com Sistema Operacional Windows 10 de 64 bits, processador Ryzen 7 1700 e 8 gigabytes de memória RAM.

Neste trabalho, seguimos a metodologia proposta por Esfandyari e colegas [Esfandyari et al. 2018], de forma que fosse possível comparar os resultados. Desse modo, cada um dos três conjuntos de treino foi utilizado para treinar os classificadores. Em seguida, cada modelo de classificador foi testado em dois conjuntos de testes diferentes, gerando ao final, 48 experimentos. Foi usada a validação cruzada com k-*fold*, com $k = 10$. Os resultados estão resumidos na Tabela 3, com o tempo de execução e o valor da acurácia de cada experimento. Em todas as tabelas de resultado, foram usadas duas casas decimais de modo a facilitar a leitura. Os valores em negrito são os melhores valores da métrica avaliada (considerando diferença de $0{,}1$) de um determinado Teste/Treino.

Como esperado, os modelos testados nos conjuntos de dados com o menor nível de dificuldade (Treino 1/Teste 1) obtiveram os melhores resultados. Resultado compatível com a metodologia proposta. Os melhores resultados de acurácia foram do AdaBoost e do XGBoost, resultado esperado pois métodos *ensemble*, em geral, têm melhores resultados que modelos simples. Com relação ao tempo de execução, o AdaBoost leva praticamente o dobro do tempo de resposta que o XGBoost. E os classificadores KNN e SVM apresentam tempo de resposta bem maior que todos os outros e também os piores valores de acurácia.

[5] `https://github.com/joaomarcosmareto/reabtic`

A Tabela 4 apresenta as métricas de precisão e sensibilidade e assim como na tabela anterior, os melhores classificadores foram AdaBoost e XGBoost, em todos os casos de treino e teste. Os valores em negrito marcam os melhores resultados. Os resultados da regressão logística e do LDA também apresentam bons resultados. No geral, os piores resultados são dos classificadores KNN e SVM. É interessante observar que, dentre os métodos clássicos, os baseados em teoria probabilística, a regressão logística e o LDA, são melhores que os outros.

Tabela 3. Comparativo de Performance dos Classificadores, onde T (ms) é o tempo e Acc é a acurácia.

	Treino 1				Treino 2				Treino 3			
	Teste 1		Teste 2		Teste 1		Teste 2		Teste 1		Teste 2	
Classificadores	T	Acc	T	Acc	T	Acc	T	Acc	T	Acc	T	Acc
Regressão Logística	2	0,94	2	0,82	2	0,92	1	0,88	2	**0,92**	1	0,88
LDA	2	0,90	1	**0,84**	2	0,90	1	0,86	2	0,89	1	0,85
Naïve Bayes	2	0,94	2	0,81	3	0,87	2	0,83	2	0,86	1	0,81
KNN	35	0,90	27	0,78	34	0,85	30	0,81	34	0,76	26	0,71
Árvore de decisão	2	0,93	1	0,82	2	0,89	2	0,88	2	0,83	1	0,84
SVM	23	0,89	21	0,76	28	0,84	29	0,79	28	0,83	21	0,77
AdaBoost	13	**0,96**	12	**0,84**	13	**0,93**	11	**0,90**	12	**0,93**	12	**0,91**
XGBoost	7	**0,96**	6	**0,85**	7	**0,94**	5	**0,91**	8	**0,93**	6	**0,91**

Tabela 4. Comparativo de Performance dos Classificadores, onde P é Precisão e S é Sensibilidade.

	Treino 1				Treino 2				Treino 3			
	Teste 1		Teste 2		Teste 1		Teste 2		Teste 1		Teste 2	
Classificadores	P	S	P	S	P	S	P	S	P	S	P	S
Regressão Logística	0,95	0,94	0,82	0,81	**0,93**	0,92	0,88	0,89	**0,93**	**0,91**	0,88	0,89
LDA	0,91	0,90	**0,85**	**0,85**	0,92	0,90	0,87	0,87	0,91	0,89	0,86	0,86
Naïve Bayes	0,94	0,94	0,81	0,81	0,90	0,87	0,85	0,84	0,89	0,86	0,85	0,83
KNN	0,91	0,90	0,78	0,78	0,87	0,85	0,83	0,83	0,77	0,76	0,71	0,71
Árvore de decisão	0,93	0,93	0,83	0,80	0,89	0,89	0,88	0,89	0,83	0,83	0,84	0,84
SVM	0,91	0,89	0,76	0,76	0,88	0,84	0,83	0,81	0,87	0,83	0,82	0,80
AdaBoost	**0,96**	**0,96**	**0,85**	**0,83**	**0,93**	**0,93**	**0,90**	**0,91**	**0,93**	**0,93**	**0,91**	**0,92**
XGBoost	**0,97**	**0,96**	**0,86**	**0,83**	**0,94**	**0,94**	**0,91**	**0,92**	**0,93**	**0,93**	**0,91**	**0,92**

Os resultados obtidos neste trabalho foram comparados com os resultados obtidos por Esfandyari e colegas [Esfandyari et al. 2018] e apresentados nas Tabelas 5, 6 e 7. Em termos de acurácia, o melhor resultado do XGBoost foi no cenário Treino 3 e Teste 1, apresentando valor de 0,93 de acurácia contra 0,90 do trabalho de Esfandyari e colegas [Esfandyari et al. 2018]. Não é perceptível a diferença de acurácias nos outros cenários. Praticamente não há diferenças em questão da métrica de precisão entre os trabalhos. Em termos de sensibilidade, os resultados deste trabalho, AdaBoost e XGBoost, foram melhores no cenário Treino 3 e Teste 1 e foram piores no cenário Treino 1 Teste 2.

Tabela 5. Comparação de Acurácia entre este trabalho e o de Esfandyari e colegas [Esfandyari et al. 2018].

		Treino 1		Treino 2		Treino 3	
	Classificadores	Teste 1	Teste 2	Teste 1	Teste 2	Teste 1	Teste 2
[Esfandyari et al. 2018]	MLP	0,96	0,84	0,92	0,91	0,90	0,91
	Random Forest	0,95	0,85	0,93	0,91	0,90	0,92
Este trabalho	AdaBoost	0,96	0,84	0,93	0,90	0,93	0,91
	XGBoost	0,96	0,85	0,94	0,91	0,93	0,91

Tabela 6. Comparação de Precisão entre este trabalho e o de Esfandyari e colegas [Esfandyari et al. 2018].

		Treino 1		Treino 2		Treino 3	
	Classificadores	Teste 1	Teste 2	Teste 1	Teste 2	Teste 1	Teste 2
[Esfandyari et al. 2018]	MLP	0,96	0,85	0,92	0,91	0,90	0,91
	Random Forest	0,96	0,85	0,94	0,91	0,90	0,93
Este trabalho	AdaBoost	0,96	0,85	0,93	0,90	0,93	0,91
	XGBoost	0,97	0,86	0,94	0,91	0,93	0,91

Tabela 7. Comparação de Sensibilidade entre este trabalho e o de Esfandyari e colegas [Esfandyari et al. 2018].

		Treino 1		Treino 2		Treino 3	
	Classificadores	Teste 1	Teste 2	Teste 1	Teste 2	Teste 1	Teste 2
[Esfandyari et al. 2018]	MLP	0,96	0,85	0,92	0,91	0,90	0,91
	Random Forest	0,95	0,85	0,93	0,91	0,90	0,92
Este trabalho	AdaBoost	0,96	0,83	0,93	0,91	0,93	0,92
	XGBoost	0,96	0,83	0,94	0,92	0,93	0,92

5. Considerações Finais

Neste trabalho, foi feito um estudo sobre o tema de identificação cruzada entre diferentes redes sociais baseada na abordagem de em atributos de perfil, comparando 8 classificadores sob uma base de dados publicamente disponível. Os resultados mostraram-se promissores, sendo iguais ou, em alguns casos, ligeiramente superiores ao trabalho de referência, levando em consideração apenas a acurácia como métrica de avaliação. Além disso, foi observado o resultado do tempo de execução, que a depender do classificador utilizado, pode ser aproximadamente 2 vezes menor.

Até onde o conhecimento dos autores alcança, o trabalho de Esfandyari e colegas foi o que demonstrou o método com o melhor resultado na tarefa de identificação de perfis de um mesmo indivíduo e foi o único que disponibilizou a base de dados usada de forma pública. Desta forma foi possível testar mais classificadores e comparar com os resultados do trabalho base.

Apesar de ser uma rede social válida, o Google+ foi desativado em janeiro de 2019 para usuários domésticos. Assim, é necessário um estudo em redes sociais mais atuais e com mais números de usuários. Vislumbramos como trabalho futuro, a elaboração de uma base dados própria contendo dados reais. A aplicação é para o acompanhamento

de egressos da instituição de ensino. De acordo com o SINAES[6], a política institucional deve garantir mecanismo de acompanhamento de egressos e a atualização sistemática de informações a respeito da continuidade na vida acadêmica ou da inserção profissional. Assim, planeja-se fazer a coleta de dados do Linkedin, tendo em vista ser a maior rede social voltada ao mercado profissional [Exame 2014] e dado o grau de seriedade de preenchimento das informações [Herrman 2019]. Seriam buscadas as informações sobre a continuidade dos estudos e quais os empregos que tiveram após a formatura na instituição. Outros trabalhos futuros, podem ser o uso de outras formas de extração de características, outras métricas de comparação e usar outros classificadores, inclusive *Deep Learning*.

Referências

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230.

Carmagnola, F. and Cena, F. (2009). User identification for cross-system personalisation. *Information Sciences*, 179(1):16–32.

Ceravolo, I., Brasil, A. A., and Komati, K. (2019). Classifying readers with dyslexia from eye movements using machine learning and wavelets. In *ENIAC 2019*.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Cohen, W. W., Ravikumar, P., Fienberg, S. E., et al. (2003). A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 2003, pages 73–78.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Deng, K., Xing, L., Zheng, L., Wu, H., Xie, P., and Gao, F. (2019). A user identification algorithm based on user behavior analysis in social networks. *IEEE Access*, 7:47114–47123.

Digiampietri, L., Linden, R., and Barbosa, L. (2015). Desambiguação de nomes em redes sociais acadêmicas: Um estudo de caso usando dblp. In *Anais do IV Brazilian Workshop on Social Network Analysis and Mining*. SBC.

Esfandyari, A., Zignani, M., Gaito, S., and Rossi, G. P. (2018). User identification across online social networks in practice: Pitfalls and solutions. *Journal of Information Science*, 44(3):377–391.

Exame (2014). Estudo do linkedin analisa relação de jovens com marcas. Disponível em: https://exame.com/marketing/estudo-do-linkedin-analisa-relacao-de-jovens-com-marcas/, Acesso em 06 set. 2020.

[6]http://download.inep.gov.br/educacao_superior/avaliacao_institucional/instrumentos/2017/IES_recredenciamento.pdf

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. (2000). *Inteligência Artificial*. LTC, Rio de Janeiro.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Herrman, J. (2019). Redes em crise, exceto uma: por que ninguém fala sobre o linkedin? Disponível em: `https://exame.com/negocios/por-que-ninguem-fala-sobre-o-linkedin/`, Acesso em 06 set. 2020.

Li, Y., Peng, Y., Ji, W., Zhang, Z., and Xu, Q. (2017). User identification based on display names across online social networks. *IEEE Access*, 5:17342–17353.

Li, Y. H. and Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, 41(8):537–546.

Neto, W. B. d. R., Jr., J. M. P. d. M., and Souza, R. V. L. (2017). Análise de dados obtidos através de um sistema de telemetria automotivo utilizando k-nn. *XIV Encontro Nacional de Inteligência Artificial e Computacional*, pages 960–971.

Pal, M. (2008). Multiclass approaches for support vector machine based land cover classification. *arXiv preprint arXiv:0802.2411*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Perito, D., Castelluccia, C., Kaafar, M. A., and Manils, P. (2011). How unique and traceable are usernames? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–17. Springer.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

Rodrigues, D. S. et al. (2018). A comparative analysis of loan requests classification algorithms in a peer-to-peer lending platform. In *Proceedings of the XIV Brazilian Symposium on Information Systems*, page 42. ACM.

Rowe, M. (2009). Applying semantic social graphs to disambiguate identity references. In *European Semantic Web Conference*, pages 461–475. Springer.

Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.

Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237.

Shu, K., Wang, S., Tang, J., Zafarani, R., and Liu, H. (2017). User identity linkage across online social networks: A review. *ACM SIGKDD Explorations Newsletter*, 18(2):5–17.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89.

Tata, S. and Patel, J. M. (2007). Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM Sigmod Record*, 36(2):7–12.

Valente, J. (2018). Facebook chega a 2,6 bilhões de usuários no mundo com suas plataformas. Disponível em: `http://agenciabrasil.ebc.com.br/geral/noticia/2018-10/facebook-chega-26-bilhoes-de-usuarios-no-mundo-com-suas-plataformas`, Acesso em 22 jun. 2019.

Veldman, I. (2009). *Matching Profiles from Social Network Sites*. University of Twente, Netherlands.

Vosecky, J., Hong, D., and Shen, V. Y. (2009). User identification across multiple social networks. In *2009 first international conference on networked digital technologies*, pages 360–365. IEEE.

Wasserman, S. and Galaskiewicz, J. (1994). *Advances in social network analysis*. Sage, Califórnia, CA, sage publications edition.

Zafarani, R. and Liu, H. (2009). Connecting corresponding identities across communities. In *Third International AAAI Conference on Weblogs and Social Media*, pages 354–357.

Zafarani, R. and Liu, H. (2013). Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 41–49. ACM.

Zhang, Y., Fu, J., Yang, C., and Xiao, C. (2019). A local expansion propagation algorithm for social link identification. *Knowledge and Information Systems*, 60(1):545–568.