Universidade FUMEC
Faculdade de Ciências Empresariais
Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento

Aplicação de técnicas de aprendizado de máquina aplicadas na integração de dados

José Ricardo Gonçalves

Belo Horizonte
2021

José Ricardo Gonçalves

Aplicação de técnicas de aprendizado de máquina aplicadas na integração de dados

Dissertação de mestrado apresentado ao Programa de Mestrado Profissional em Sistemas de Informação e Gestão do Conhecimento, da Universidade FUMEC, como parte dos requisitos para a obtenção do título de Mestre em Sistemas de Informação e Gestão do Conhecimento.

Área de concentração: Gestão de Sistemas de Informação e do Conhecimento.

Linha de pesquisa: Tecnologia e Sistemas de Informação.

Orientador: Prof. Dr. Fernando Silva Parreiras

Belo Horizonte

2021

Dados Internacionais de Catalogação na Publicação (CIP)

G635a Gonçalves, José Ricardo, 1993-

Aplicação de técnicas de aprendizado de máquina aplicadas na integração de dados / José Ricardo Gonçalves. - Belo Horizonte, 2021.

69 f. : il.

Orientador: Fernando Silva Parreiras

Dissertação (Mestrado em Sistemas de Informação e Gestão do Conhecimento), Universidade FUMEC, Faculdade de Ciências Empresariais, Belo Horizonte, 2021.

1. Aprendizado do computador. 2. Integração de dados (Computação). 3. Redes neurais (Computação). 4. Inteligência artificial. I. Título. II. Parreiras, Fernando Silva. III. Universidade FUMEC, Faculdade de Ciências Empresariais.

CDU: 004.8

Ficha catalográfica elaborada pela Biblioteca Universitária-FUMEC

Dissertação intitulada "**Aplicação de técnicas de aprendizado de máquina aplicadas na integração de dados**" de autoria de **José Ricardo Gonçalves**, aprovada pela banca examinadora constituída pelos seguintes professores**:**

Prof. Dr. Fernando Silva Parreiras – Universidade FUMEC
(Orientador)

Prof. Dr. José Maurício Costa – Universidade FUMEC
(Examinador Interno)

Prof. Dr. Eric de Paula Ferreira – UEMG
(Examinador Externo)

Prof. Dr. Fernando Silva Parreiras
Coordenador do Programa de Pós-Graduação em Sistemas de Informação e Gestão do Conhecimento da Universidade FUMEC

Belo Horizonte, 7 de julho de 2021.

Fernando Silva Parreiras

Eric de Paula Ferreira

José Maurício Costa

ABARIS

REQUESTED

TITLE	**Assinatura de ata e contra-capas Universidade**
FILE NAME	**cf941edd-ab4f-455e-b8a2-c05a561353ed.pdf**
RESQUEST ID	**signature_request_eb2c20b1-4038-4e65-a73a-6d0fb**
RESQUESTED BY	**Karem Estefani Oliveira De Paula**
STATUS	**Completed**

Professor (fernando.parreiras@fumec.br)

SENDED	05/10/2021 20:58:47UTC±0	SIGNED	05/10/2021 20:58:56UTC±0 187.111.30.10

Professor (jose.costa@fumec.br)

SENDED	06/10/2021 00:26:07UTC±0	SIGNED	06/10/2021 00:26:40UTC±0 179.189.93.156

Professor (eric.p.f@gmail.com)

SENDED	07/10/2021 11:40:36UTC±0	SIGNED	07/10/2021 11:40:59UTC±0 131.161.13.17

COMPLETED

07/10/2021
11:40:59 UTC±0
The document has been completed.

Resumo

A representação da informação pode mudar de uma circunstância para outra. A identificação de estruturas lógicas que representam os mesmos conceitos é uma tarefa fundamental na integração de dados. O processo de integração de dados consiste em encontrar o mapeamento entre os esquemas de entrada e produzir como saída um sistema integrado. A excussão da tarefa de integração de banco de dados pode ser feita de forma manual, tornando o trabalho demorado e propenso a erros. Existem métodos propostos na literatura com o objetivo de automatizar essa tarefa utilizando os paradigmas do aprendizado de máquina, podendo tratar a tarefa de integração de dados de forma autônoma ou semiautônoma. Este trabalho analisa a aplicação de técnicas de aprendizado de máquina para a realização da tarefa de integração de dados. Nos experimentos, a utilização de mais *matchers* apresentou resultados melhores, em comparação com um menor número de *matchers*, onde os resultados foram 7% melhores, utilizando a medida-f.

Palavras-chave: Integração de dados; Árvore de decisão; Aprendizado de máquina; Florestas de decisão aleatória.

Lista de Figuras

Lista de Tabelas

Sumário

1 INTRODUÇÃO

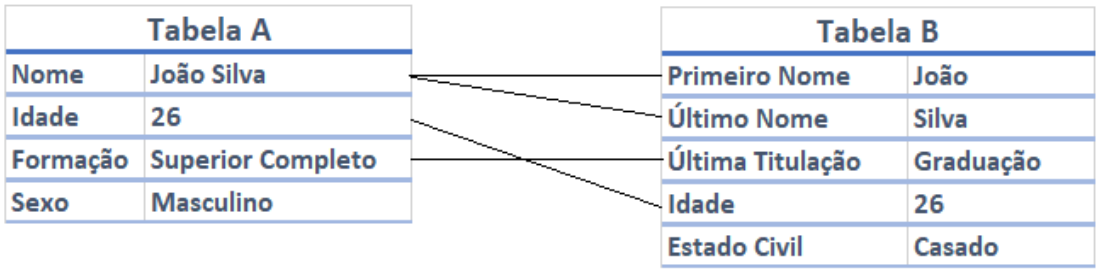
A representação da informação pode variar em ambientes distintos. Os bancos de dados são heterogêneos, pois eles armazenam modelos de dados com suas particularidades, representam os dados de várias formas, executados em vários tipos de plataformas de hardware e costumam ser gerenciados por softwares variados (Scopim, 2003).

A identificação de estruturas lógicas que representam os mesmos conceitos é uma tarefa fundamental na integração de dados (Silva et al., 2017). Com essa variedade de representação da mesma informação a necessidade de encontrar as correspondências entre os esquemas tem se tornado relevante. A tarefa de encontrar estas correspondências é minuciosa (Doan et al., 2012), pois consiste no mapeamento entre os esquemas de entrada e produzir como saída um sistema integrado.

De forma típica a integração de dados é executada de forma manual, em que uma interface gráfica pode auxiliar na execução da tarefa. Esse trabalho manual de integração de esquema pode ser demorado, propenso a erros e complexo (Rahm e Bernstein, 2001).

A Figura 1 ilustra dois esquemas em que estão armazenadas as informações de um usuário. Os esquemas apresentados apresenta a estrutura de chave-valor. Os esquemas apresentam as informações de um usuário fictício, "João Silva". As correspondências encontradas entre os bancos estão indicadas pelas linhas, que ligam as duas tabelas. Entre as tabelas, existem as correspondências que possuem as mesmas formas de representação (Rahm e Bernstein, 2001), como "Idade". Existem correspondências que possuem representação variada, mas possuem a mesma semântica (Rahm e Bernstein, 2001), como "Formação"na Tabela A e "Última Titulação"na Tabela B. Outro caso complexo é quando um atributo de um esquema é representado por dois ou mais atributos em outro esquema (Rahm e Bernstein, 2001), como é o caso do atributo "Nome"na Tabela A, que corresponde a "Primeiro Nome"e "Último Nome"na Tabela B. Embora estejam descrevendo um usuário, existem casos em que os esquemas não possuem correspondências entre todos os atributos (Silva et al., 2017), como é o exemplo de "Sexo"na Tabela A e "Estado Civil"na Tabela B.

Um método para integrar dados tem a tarefa de retornar um mapeamento entre os atributos correspondentes entre dois esquemas fornecidos como entrada. No exemplo, para o caso da Tabela A e da Tabela B, o mapeamento é: "Nome"na Tabela A corresponde à "Primeiro Nome"e "Último Nome"na Tabela B; "Idade"na Tabela A corresponde à "Idade"na Tabela B; "Formação"na Tabela A corresponde à "Última Titulação"na Tabela B. Um exemplo de utilização desse mapeamento de esquemas pode ser em aplicações como geração de esquemas globais, reescrita de consultas em fontes heterogêneas e eliminação



Figuras 1 – Exemplo Casamento de Esquemas.

de dados duplicados (Doan et al., 2012).

Para realizar essa tarefa os métodos utilizam *matchers* nos esquemas de entrada, que retornam uma matriz de semelhança entre cada par de elementos. Existem classificações para agrupar os *matchers* em grupos distintos (Rahm e Bernstein, 2001). Existem classificações baseadas em: **nível de instância**, **nível de esquema**, **nível híbrido**, **nível de instância** e **nível auxiliar**.

1.1 Problema de Pesquisa

Ter as informações centradas facilita o seu uso para o desenvolvimento de pesquisas e processos dentro de uma instituição. Como nas instituições existem milhares de dados segregados, a integração realizada de forma manual se torna ineficiente (Rahm e Bernstein, 2001). A utilização de métodos para a automatização do processo de integração de dados se torna fundamental, deixando-o ágil e assertivo (Li et al., 2005).

Um dos paradigmas utilizados para a realização dessa automatização é o aprendizado de máquina. Os métodos podem apresentar grau de imperfeição (Gal, 2006), por isso comparar métodos de integração de dados para cada cenário estudado é de suma importância. Esse trabalho apresenta o paradigma de aprendizado de máquina utilizado na integração de dados: Redes neurais artificiais. O aprendizado de máquina é uma área da Inteligência Artificial, cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado e a construção de sistemas capazes de adquirir conhecimento de forma automática e tomar uma decisão (Monard e Baranauskas, 2003a; Marsland, 2015; Alpaydin, 2020).

Para realizar de forma autônoma a tarefa de integração de dados, podemos utilizar algumas informações, como: similaridade de sentença, metadados, padrões, tipos de dados, dentre outras informações (Rahm e Bernstein, 2001).

Nessas circunstâncias, levantamos a seguinte questão de pesquisa: **Qual é a acurácia das técnicas que utilizam aprendizado de máquina na integração de dados?**

1.2 Objetivos

Com o objetivo de responder à questão de pesquisa, este trabalho visa analisar a aplicação técnicas de aprendizado de máquina para a realização da tarefa de integração de dados. As técnicas serão aplicadas a bancos de dados e os resultados comparados. Ao final, espera-se obter uma comparação entre as técnicas utilizadas. Para atingir o objetivo primário, traçou-se três objetivos específicos.

- **Objetivo específico 1**: Estabelecer uma linha de base para as técnicas de integração de dados.
- **Objetivo específico 2**: Identificar as técnicas de aprendizado de máquina.
- **Objetivo específico 3**: Comparar técnicas de aprendizado de máquina no processo de integração de dados.

1.3 Motivação

A possibilidade de integrar bases de dados de forma autônoma, ágil e dinâmica é de essencial importância para as instituições. Aumentando a eficiência de projetos e processos.

1.4 Adequação do Projeto de Mestrado e a Linha de Pesquisa

A pesquisa está adepta ao Programa de Mestrado em Sistemas de Informação e Gestão do Conhecimento da Universidade FUMEC, pois propõe a aplicação de técnicas para a integração de dados. O foco desta proposta está no campo Sistemas de Informação, em conformidade com o programa de pós-graduação da FUMEC.

1.5 Estrutura da Pesquisa

Esta dissertação apresenta-se estruturado em 6 Capítulos. O Capítulo 1 apresenta a Introdução, o Capítulo 2 apresenta a Fundamentação Teórica, o qual encontra-se subdividido em três assuntos: Integração de Dados, Aprendizado de Máquina e Utilização de Aprendizado de Máquina para Integração de Dados. O Capítulo 3 apresenta os Trabalhos Relacionados ao estudo. O Capítulo 4 apresenta os Procedimentos Metodológicos seguidos pela pesquisa. O Capítulo 5 apresenta os Resultados e Discussões. E por fim, o Capítulo 6 apresenta a conclusão da pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

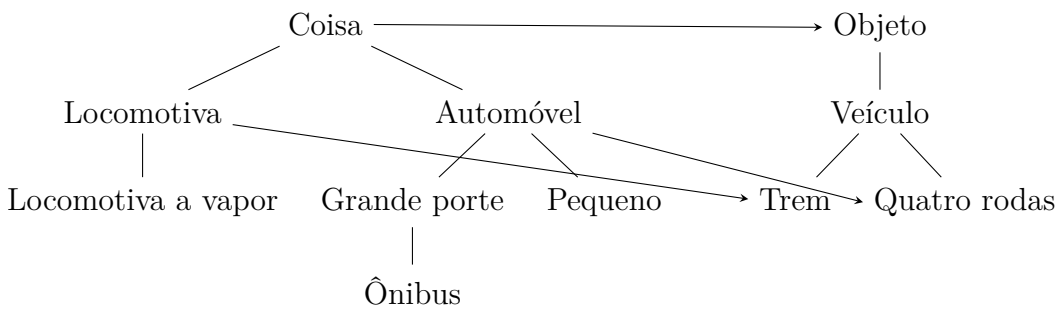
2.1 Integração de Dados

Os avanços na tecnologia, em destaque na área de tecnologia da informação, permite o acesso e analise de um volume elevado de dados e transformá-los em informação. O uso da informação vem ganhando seu espaço no dia-a-dia das instituições, estudos científicos, das notícias, e a informação varia de acordo com a finalidade (Becker, 2015).

A tarefa de integrar dados é executável com velocidade e escalabilidade graças aos avanços tecnológicos em relação as ferramentas computacionais e de arquitetura de dados, as quais permitem armazenar os dados coletados em bancos de dados e disponibilizá-los ao usuário em caso de necessidade (Silva e Campos, 2015).

Devido a diversidade de informação e as formas de se fazer o armazenamento dos dados, houve a necessidade de realizar a integração de dados, ou seja, concentrá-las em uma fonte de dados (Do e Rahm, 2002; Lenzerini, 2002). Proporcionando facilidade e geração de informação a partir dos dados de forma ágil e com a mínima ocorrência de erros (Massmann et al., 2011). Os cenários observados para a aplicação da integração de dados são: integração de dados orientada para a Web, comércio eletrônico, integração de esquemas, evolução de esquemas e migração, evolução de aplicativos, *data warehouse*, *design* de banco de dados, criação e gerenciamento de sites e desenvolvimento baseado em componentes (Lenzerini, 2002).

Para isto a integração de dados visa identificar correspondências semânticas entre estruturas ou modelos de metadados, conforme exemplificado pela Figura 2, nela tem-se duas representações da informação e as prováveis correspondências entre seus elementos. Apesar de possuir estudos e revisões na literatura, esta tarefa se mantém morosa (Conrad et al., 1997; Shvaiko e Euzenat, 2011; Otero-Cerdeira et al., 2015; Ardjani et al., 2015).



Figuras 2 – Exemplo de integração entre esquemas. As linhas representam as prováveis correspondências entra as duas formas de representação da informação.

Integração de dados é uma tarefa básica, mas crítica, que visa construir mapeamentos de correspondência para elementos que representam os mesmos conceitos (Lenzerini, 2002; Kalfoglou e Schorlemmer, 2003). Integrar os dados e manter as correspondências de forma correta pode ser complexo e exige tempo, sobretudo se realizada em esquemas volumosos (Rahm e Bernstein, 2001).

A integração de dados é realizada no momento em que os dados são combinados de forma concreta ou virtual das fontes de dados heterogêneas e distribuídas. Os esquemas de dados serão combinados de forma engenhosa usando um esquema de destino ou esquemas intermediários que serão combinados no final de forma unificada (Widom, 1995; Batista, 2003; Mukkala et al., 2015).

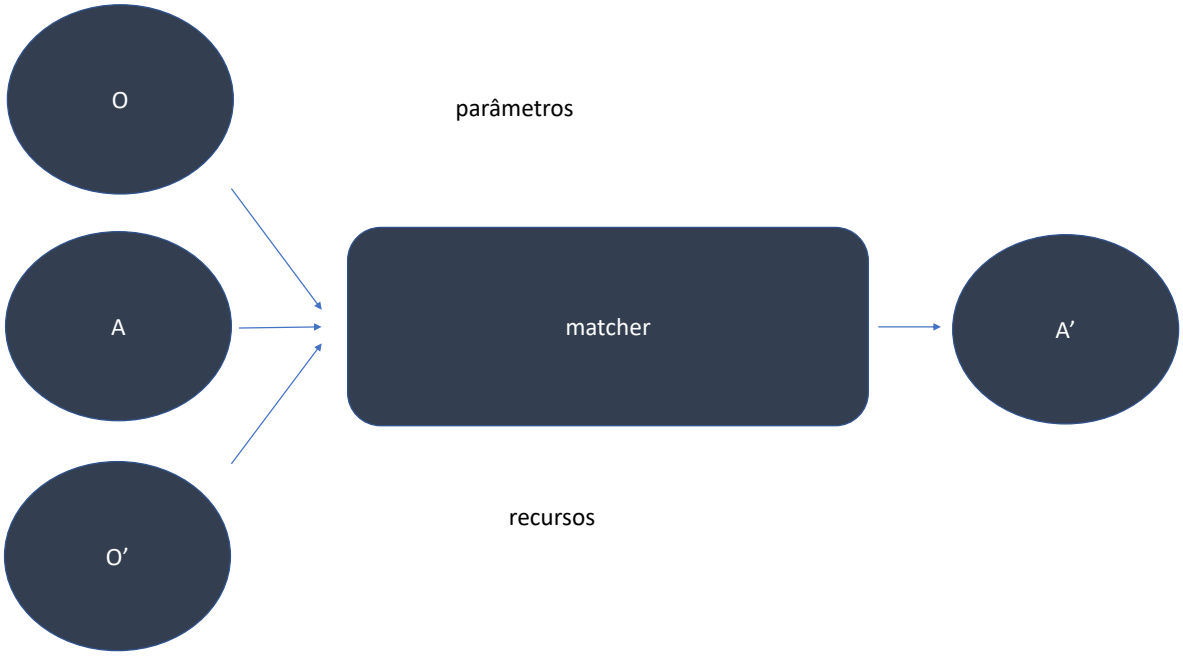
A Integração de dados é realizada de forma manual, autônoma ou semiautônoma com a ajuda de técnicas computacionais. Devido a dimensão extensa dos bancos de dados, fazer a integração de dados de forma manual tem se tornado inviável, estudos na área de sistema de informação buscam desenvolver ou descobrir as melhores técnicas para a integração de dados (Rahm e Bernstein, 2001).

Para encontrar as correspondências o usuário conta com a assistência de interfaces gráficas, que facilitam o processo (Munroe e Papakonstantiou, 2000; Ledesma, 2008). No momento em que o número de fontes de dados e o volume aumenta, a tarefa de encontrar correspondências se torna demorada e propensa a erros (Wick et al., 2008; Rodrigues et al., 2013). Métodos autônomos disponíveis tentam encontrar correspondências semânticas, estruturais ou linguísticas entre os elementos de esquema das fontes de dados, mas não há um método exclusivo que demonstre bons resultados em todos os cenários (Mukkala et al., 2015).

A estratégia utilizada para realizar a integração de dados faz uso das informações disponíveis nos esquemas como: similaridade de nomes, descrições, valores, tipos de dados, estrutura, padrões, dentre outros e então usam funções conhecidas como *matchers* para encontrar o valor de similaridade entre as correspondências detectadas (Rodrigues et al., 2013). Os métodos abordados nesse trabalho pegam as matrizes de similaridade geradas pelos *matchers* e utilizam técnicas de aprendizado de máquina para realizar essa combinação entre as matrizes de similaridade.

2.1.1 Matchers

Para realizar a integração de dados, foi desenvolvida uma forma de identificar as correspondências entre os esquemas de entrada e promover um resultado final, que recebeu o nome de *matcher*. O *matcher* é uma função que faz a combinação de métodos para mapear as correspondências dos dados, a partir de parâmetros e recursos fornecidos, entre os esquemas propostos e retornar um valor de similaridade, os quais serão armazenados



Figuras 3 – Exemplo de um matcher. A é o alinhamento de entrada, no qual O e O' são os dados em que o matcher busca correspondência a partir de parâmetros e recursos fornecidos, gerando o alinhamento A'. Adaptado de Euzenat et al. (2007)

.

em uma matriz (Rodrigues et al., 2013). Pode-se observar como esse processo acontece na Figura 3. Na figura, A é o alinhamento de entrada, no qual O e O' são os dados em que o matcher busca correspondência a partir de parâmetros e recursos fornecidos, gerando o alinhamento A'.

Matcher é uma operação de manipulação de esquemas que recebe dois esquemas, com n e m elementos, como entrada e retorna uma matriz n por m, de similaridade entre os elementos gravados nos dois esquemas (Madhavan et al., 2001). A matriz possui nm valores, em que cada entrada a_{ij} possui o valor de similaridade a entre o elemento i de um esquema e o elemento j do outro esquema (Lipschutz e Lipson, 2009). O valor dado por um *matcher* deve refletir a similaridade entre o par de elementos analisado segundo o critério referido, normalmente esse valor estará normalizado entre [0, 1] sendo 0 o valor que denota similaridade nula e 1 denotando similaridade máxima (Rodrigues et al., 2013).

Uma matriz de similaridade entre os elementos de dois esquemas será exemplificada na Tabela 1. Um dos esquemas possui os elementos "PO", "Custumer", "Fname"e "ShipAddress"e o outro esquema possui os elementos "PurchaseOrder", "Product", "BillTo", "Name"e "Adress".

Na matriz, as correspondências verídicas encontradas são: entre elementos que possuem representação dissemelhante, mas tem a mesma semântica, como "PurchaseOrder"e "PO", em que o *matcher* encontrou similaridade 0,80. Entre elementos cuja semântica

Matcher	PO	Customer	Fname	ShipAddress
PurchaseOrder	0.80	0.30	0.28	0.16
Product	0.50	0.20	0.12	0.22
BillTo	0.60	0.33	0.12	0.20
Name	0.30	0.65	1.00	0.25
Address	0.30	0.22	0.20	1.00

Tabelas 1 – Tabela com exemplos de similaridades resultantes da execução de um *matcher*. De Rodrigues et al. (2013)

corresponde de forma parcial, como "Name", que contém o nome integral, e "FName", que contém parte do nome, em que o *matcher* encontrou similaridade 1,00. Entre elementos cujo a semântica será análoga, mas representam conceitos dissemelhantes, como "Address"e "ShipAddress", em que o *matcher* encontrou similaridade 1,00.

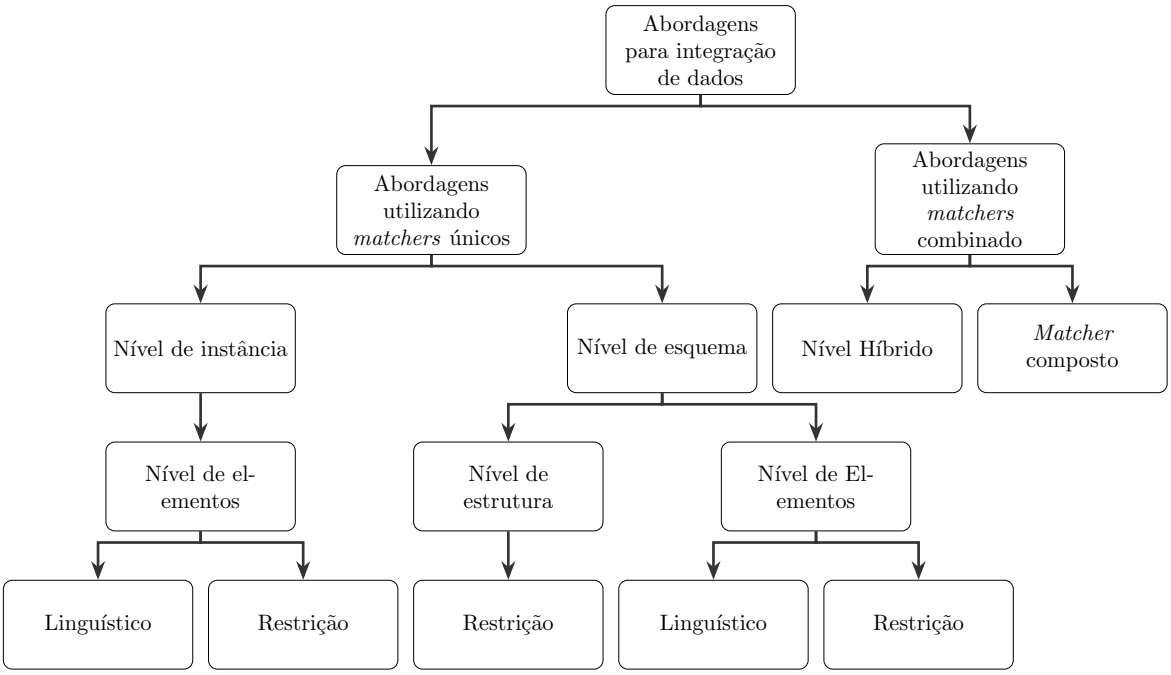
A tarefa de mapear e encontrar correspondências entre os esquemas é complexa, uma vez que os bancos de dados não se apresentam da mesma forma, ou seja, são heterogêneos e estruturados de acordo com a necessidade de cada instituição. Como solução e redução de erros para este problema o *matcher* realiza sua tarefa sob os seguintes pilares: **i)** informações fornecidas pelas fontes de dados, **ii)** instâncias e informações auxiliares e **iii)** fontes externas.

Existem categorias para a classificação dos *matchers* Rahm e Bernstein (2001); Alwan et al. (2017). A Figura 4 mostra as classificações das abordagens de integração de dados abordadas em Alwan et al. (2017) . Existem classificações baseadas em: **nível de instância**, que consideram o conteúdo dos dados; **nível de esquema**, que consideram informações do nível de esquemas; **nível híbrido**, que combinam as informações de entrada no nível do esquema e no nível de instância e **nível auxiliar** que combina esquemas existentes com fontes externas (Alwan et al., 2017; Rahm e Bernstein, 2001).

2.1.1.1 *Matchers* em nível de esquema

Os *matchers* de nível de esquema consideram informações de esquema. As informações disponíveis incluem as propriedades usuais dos elementos do esquema, como nome, descrição, tipo de dados, tipos de relacionamento, restrições e estrutura do esquema. A obtenção das informações de esquemas são feitas a partir do nível **linguístico**, de **restrição** e **estrutura** (Rahm e Bernstein, 2001; Shvaiko e Euzenat, 2005; Tagarelli, 2011; Alwan et al., 2017).

- **Nível linguístico:** As informações serão obtidas a partir dos nomes dos atributos e textos disponíveis.
- **Nível restrições:** As informações serão obtidas a partir do tipo do dado.



Figuras 4 – Classificação das abordagens de integração de dados. Adaptado de Alwan et al. (2017)

- **Nível estrutura:** As informações serão obtidas a partir da estrutura interna e externa dos atributos.

2.1.1.2 Matchers em nível de instância

Segundo Rahm e Bernstein (2001) os dados no nível da instância podem fornecer informações essenciais sobre o conteúdo e o significado dos elementos do esquema, pois a forma como os *matchers* em nível de instância obtém as informações podem ser consideradas eficientes e confiáveis. Eles examinam o conteúdo advindo das fontes envolvidas e, como examinam os dados de forma extensiva das fontes envolvidas são de custo elevado (Rahm e Bernstein, 2001; Shvaiko e Euzenat, 2005; Tagarelli, 2011).

2.1.1.3 Matchers em nível híbrido

Os *matchers* em nível híbrido fazem a combinação das informações extraídas pelos *matchers* em nível de esquema e nível de instância, as correspondências alcançadas são em sua maioria assertivas em relação as alcançadas nas outras categorias (Alwan et al., 2017).

2.1.1.4 Matchers em nível auxiliar

Os *matchers* em nível híbrido utilizam as informações alcançadas em esquemas existentes como os *matchers* em nível híbrido, mas ele também utiliza informação externas,

sendo esta condição que o diferencia do nível híbrido (Alwan et al., 2017).

2.1.2 Matchers utilizados

2.1.2.1 Lin Matcher

O algoritmo de Lin et al. (1998) é um *matcher* em nível de esquema e linguístico. O autor sugere três intuições para encontrar o nível de similaridade semântica entre duas palavras w e v:

- **Primeira intuição:** A similaridade entre as palavras w e v está relacionada à suas características. Quanto mais características em comum existem entre as palavras w e v mais similares elas são.
- **Segunda intuição:** Quanto menos características em comum existem entre as palavras w e v menos similares elas são.
- **Terceira intuição:** O nível máximo de semelhança é quando as palavras w e v são idênticas, possuindo as mesmas características.

Definindo $F(w)$ como o conjunto com todas as características da palavra w e $F(v)$ como o conjunto com todas as características da palavra v. As características em comum das palavras w e v é definida como $F(w) \cap F(v)$. A semelhança entre as palavras w e v é definida como:

$$Sim(v, w) = \frac{2 * I(F(w) \cap F(v))}{I(F(w)) + I(F(v))}$$

$I(S)$ é a quantidade de informações contidas no conjunto de características S. O valor de máxima semelhança entre as palavras é 1. O valor de mínima semelhança entre as palavras é 0.

2.1.2.2 Wup matcher

O algoritmo de Wu e Palmer (1994) é um *matcher* em nível de esquema e linguístico. Os autores propuseram um artigo sobre a tradução de verbos da língua inglesa para a língua chinesa. Os autores apresentam uma métrica escalonada para mensurar a similaridade entre as palavras w e v. A métrica de similaridade mede a profundidade das características em comum entre as palavras em estudo.

Em que $F(w)$ é o conjunto com todas as características da palavra w e $F(v)$ é o conjunto com todas as características da palavra v. As características em comum das palavras w e v é definida como $F(w) \cap F(v)$.

A semelhança entre as palavras w e v é definida como:

$$Sim(v, w) = \frac{2 * depth(F(w) \cap F(v))}{depth(F(w)) + depth(F(v))}$$

$depth(S)$ mede a profundidade das características S com base na taxonomia *WordNet*.

2.1.2.3 Resnik matcher

O algoritmo de Resnik (1995) é um *matcher* em nível de esquema e linguístico. Os autores se basearam no pressuposto de que quanto mais provável um conceito, menos informações ele carrega Euzenat et al. (2007). No algoritmo proposto cada conjunto de características da palavra v possui uma probabilidade de ocorrência $\pi(v)$ na base de dados utilizada. A probabilidade de ocorrência de uma característica é a soma das ocorrências da característica dividida pelo número total de características presentes na base.

Definimos $F(w)$ como o conjunto com todas as características da palavra w e $F(v)$ como o conjunto com todas as características da palavra v. As características em comum das palavras w e v é definida como $F(w) \cap F(v)$.

A semelhança semântica proposta entre as palavras w e v é uma função da características mais provável que é comum a ambos os termos. Uma definição é:

$$Sim(v, w) = (-log(\pi(F(w) \cap F(v))))$$

$F(w) \cap F(v)$ são as características em comum entre as palavras v e w, e $\pi(a)$ é a probabilidade de ocorrência da característica a

2.1.2.4 Schutze matcher

O algoritmo de Schütze (1998) é um *matcher* em nível de esquema e linguístico. No algoritmo os autores calculam a similaridade entre as palavras w e v usando os vetores das duas palavras. O vetor da palavra é composto por todos os vizinhos próximos da palavra, baseada na coocorrência de segunda ordem: em que a vizinhança é definida por palavras que coocorrem na base de dados utilizada. Ou seja, os vizinhos próximos são todas as palavras que ocorrem simultaneamente com w em uma frase ou em um contexto amplo.

A semelhança entre as palavras w e v é calculada pelo cosseno entre dois vetores:

$$Sim(v, w) = corr(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2 \sum_{i=1}^{N} w_i^2}}$$

Em que, N é a dimensão do espaço vetorial.

2.1.2.5 Tata matcher

O algoritmo de Tata e Patel (2007) é um *matcher* que faz uso do cosseno para encontrar a similaridade semântica entre as palavras a partir do ângulo formado pelos

dois vetores gerados para as *strings*, ou seja, o ângulo cosseno é a similaridade entre as *strings*. O vetor tf.idf é uma técnica popular utilizada para cálculo do comprimento do vetor, em que o tamanho é calculado pelo número de *tokens*, cada *token* é uma palavra.

A similaridade é calculada pelo cosseno do ângulo, ou seja, é o produto escalar dos dois vetores tf.idf. Para encontrar a similaridade basta encontrar a função distribuição acumulada do produto escalar. O vetor tf.idf pode ser entendido como $(X_1X_2...X_n)$ variáveis aleatórias, o produto é dado por

$$Y = \sum_{i=1}^{n} u_i X_i,$$

Em que, u é o vetor de consulta.

2.1.2.6 Li matcher

O algoritmo de Li et al. (2003) é um *matcher* que faz uso do método de comprimento do caminho, ou seja, ele verifica a partir de um nível hierárquico de palavras qual é a palavra mais próxima da palavra referência, a que apresentar um caminho mais curto, entende-se que é a mais similar, além disso os conceitos de profundidade e densidade também são levados em consideração para melhores resultados.

A semelhança de $s(w_1, w_2)$ entre w_1 e w_2 ser uma função dos atributos comprimento do caminho, profundidade e densidade local é calculada pela seguinte equação

$$s(w_1, w_2) = f(l, h, d)$$

Em que, l é o tamanho do caminho mais curto entre as palavras w_1 e w_2, h é a profundidade e d a densidade semântica de w_1 e w_2.

A reescrita da equação pode ser com as três funções independentes, ficando da seguinte forma

$$s(w_1, w_2) = f(f_1(l), f_2(h), f_3(d))$$

$f_1(l)$ é definido como

$$f_1(l) = e^{-\alpha l},$$

$f_2(h)$ é definido como

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}},$$

$f_3(d)$ é definido como

$$f_3(wsim) = \frac{e^{\lambda wsim(w_1,w_2)} - e^{-\lambda wsim(w_1,w_2)}}{e^{\lambda wsim(w_1,w_2)} - e^{-\lambda wsim(w_1,w_2)}}.$$

2.1.2.7 Zhu matcher

O algoritmo de Zhu e Iglesias (2016) é um *matcher* que faz uso da abordagem de corpus, em que ela mede a similaridade semântica entre conceitos com base nas informações obtidas a partir de grandes textos, como por exemplo, Wikipédia. Neste caso o significado da palavra não será levado em consideração, mas o número de ocorrências da palavra.

O *wpath* é o método de comprimento do caminho ponderado, ele faz a busca da codificação da estrutura da taxonomia conceitual alinhada com o método estatístico de conceito (IC). O método *wpath* é dado por

$$sim_{wpath}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j) * k^{IC(c_{lcs})}},$$

Em que $k \in (0, 1]$ e $k = 1$ significa que IC não tem contribuição no comprimento do caminho mais curto.

2.2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um dos métodos utilizados para construir algoritmos computacionais que possam extrair padrões a partir dos dados fornecidos (Goodfellow et al., 2016; Alpaydin, 2020). Algoritmos estes, que serão cada vez treinados, acurados e desenvolvidos na busca por melhores resultados (dos Santos, 2005; De Souto et al., 2003). AM é multidisciplinar e tem em sua base conceitos como: aplicação de inteligência artificial, probabilidade e estatística, complexidade computacional, teoria da informação, psicologia, neurociências e filosofia (dos Santos, 2005).

Ao utilizar aprendizado de máquina, pode-se observar ganhos nas seguintes áreas:

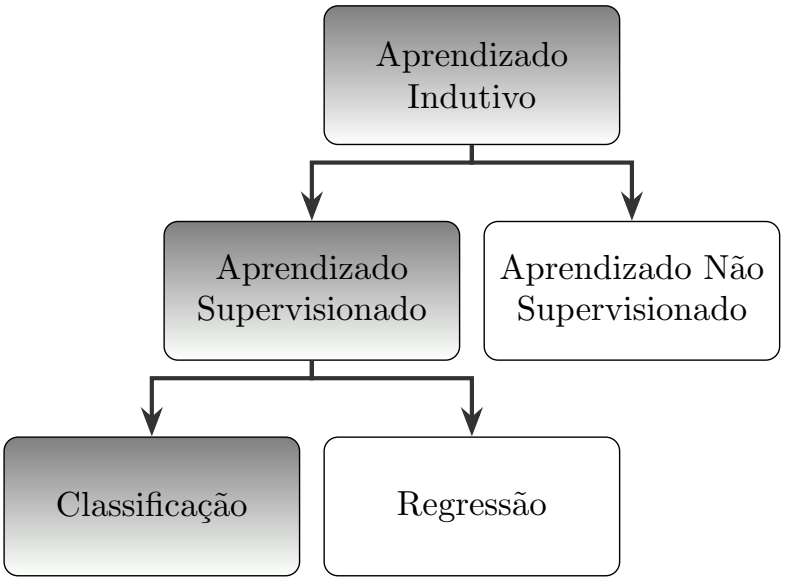
- No auxílio do diagnóstico de doenças (Wong e Bressler, 2016);
- Na comparação de sequências (DNA, RNA e proteínas) (Bittencourt, 2005);
- Em reconhecimento de fraudes (Addo et al., 2018);
- No reconhecimento facial (Zhao et al., 2015);
- No mercado de ações (Vargas et al., 2017);
- *Natural Language Processing - NLP* (Falci et al., 2019).

O aprendizado indutivo, a base do aprendizado de máquina, é o aprendizado a partir de exemplos gerados, buscando inferências lógicas dentro de um conjunto específico

de exemplos (Michie et al., 1994; Monard e Baranauskas, 2003a; Kotsiantis et al., 2007). O aprendizado indutivo divide-se em aprendizado supervisionado e não-supervisionado.

No aprendizado supervisionado a variável de resposta será conhecida, ou seja, sabe-se a resposta desejada. No caso do aprendizado não-supervisionado a variável de resposta não será conhecida, ou seja, não é de antemão conhecido o resultado esperado. Por consequência um permite comparar os resultados encontrados com os dados originais e o outro não, facilitando a análise do desempenho dos modelos propostos.

No aprendizado supervisionado, se a variável de resposta é discreta o problema é tratado como de classificação (Krishna e Murty, 1999; Xu e Tian, 2015). Caso a variável seja contínua o problema é tratado como de regressão (Michie et al., 1994; Monard e Baranauskas, 2003a; Kotsiantis et al., 2007). Na Figura 5 é apresentada a hierarquia de aprendizado de máquina, estão destacados os nós que levam ao aprendizado supervisionado do tipo classificação. O aprendizado não-supervisionado, apresenta como objetivo primário analisar os exemplos fornecidos e tentar determinar alguma forma de agrupá-los, formando assim, os agrupamentos ou *clusters* nome pelo qual também são conhecidos (Cheeseman et al., 1988).



Figuras 5 – Hierarquia do aprendizado. Adaptado de Monard e Baranauskas (2003a)

2.2.1 Paradigmas do aprendizado de máquina

Existem paradigmas de aprendizado de máquina, como: Simbólico, Estatístico, Baseado em Exemplos, Conexionista e Genético (Monard e Baranauskas, 2003b).

- **Simbólico -** Os sistemas que utilizam paradigma simbólico buscam aprender analisando representações simbólicas de um conceito a partir da análise de exemplos e

contraexemplos desse conceito (Horst, 1999; Monard e Baranauskas, 2003b).

- **Estatístico -** Os sistemas que utilizam paradigma estatísticos buscam valores apropriados para os parâmetros do modelo a partir dos dados. Esses sistemas assumem que os valores estão distribuídos de acordo com a distribuição normal e usam os dados fornecidos para encontrar a combinação linear dos valores que fornece a aproximação em sua maioria de forma assertiva sobre o conjunto de dados (Horst, 1999; Monard e Baranauskas, 2003b).

- **Baseados em Exemplos -** Os sistemas que utilizam paradigma baseados em exemplos buscam em exemplos anteriores casos similares em que a variável de resposta será conhecida e assume que o caso proposto possui a mesma variável de resposta (Horst, 1999; Monard e Baranauskas, 2003b).

- **Conexionista -** Os sistemas que utilizam paradigma conexionistas serão inspirados no sistema nervoso dos seres humanos, e sua representação envolve unidades conectadas (Horst, 1999; Monard e Baranauskas, 2003b).

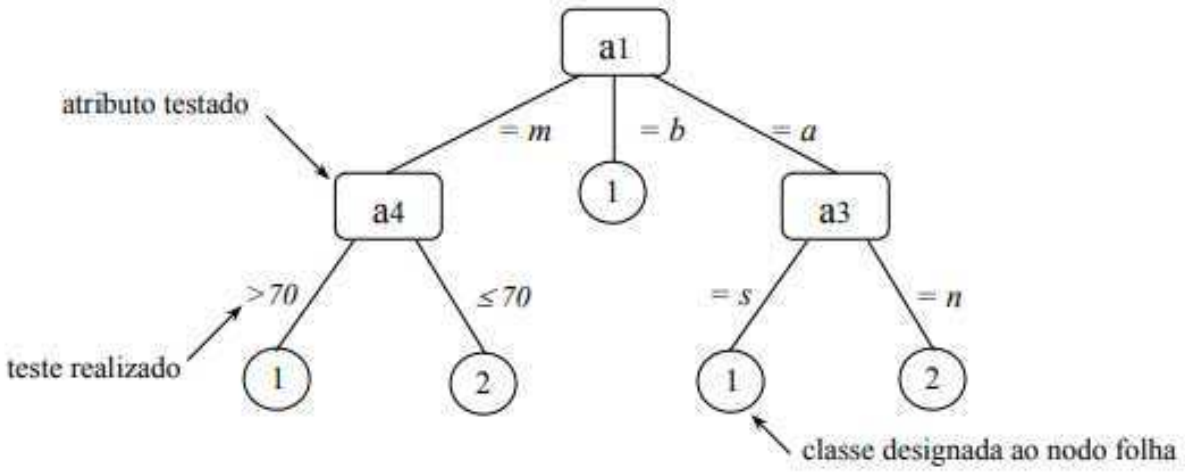
- **Genético -** O paradigma genético deriva-se do modelo evolucionário de aprendizado (Holland, 1986). Um classificador genético consiste de uma população de elementos de classificação, em que cada indivíduo da população compete a tarefa de predição. Será estabelecida uma função de avaliação para ranquear as predições realizadas. Os indivíduos que estão na parte superior do ranque serão proliferados nas próximas gerações, os indivíduos com os piores ranques serão descartados. Para a geração de indivíduos serão utilizados os operadores genéticos de reprodução, cruzamento, mutação e inversão (Horst, 1999; Monard e Baranauskas, 2003b).

2.2.2 Técnicas de aprendizado de máquina

2.2.2.1 Árvore de decisão

Árvore de decisão é um método de aprendizado de máquina, no qual o seu modelo é representado por nós e folhas, de forma análoga a uma árvore, mas com sentido invertido (Safavian e Landgrebe, 1991; Friedl e Brodley, 1997; Monard e Baranauskas, 2003b; Han et al., 2011). Os nós internos da árvore, incluindo o nó raiz, recebem o nome de nós de decisão, esses nós realizam um teste sobre uma variável independente e os resultados desses testes formam os ramos da árvore. Os nós da extremidade da árvore serão nomeados de nós folha e eles apresentam as classificações finais da árvore (Garcia, 2003).

A partir da Figura 6 é possível observar o esquema de uma árvore que apresenta três nós de decisão e cinco nós folha. As arestas apresentam o teste realizado por cada nó de decisão. A classe de saída dos nós folha seguem apresentadas dentro dos círculos.



Figuras 6 – Exemplo árvore de decisão. Adaptado de Garcia (2003)

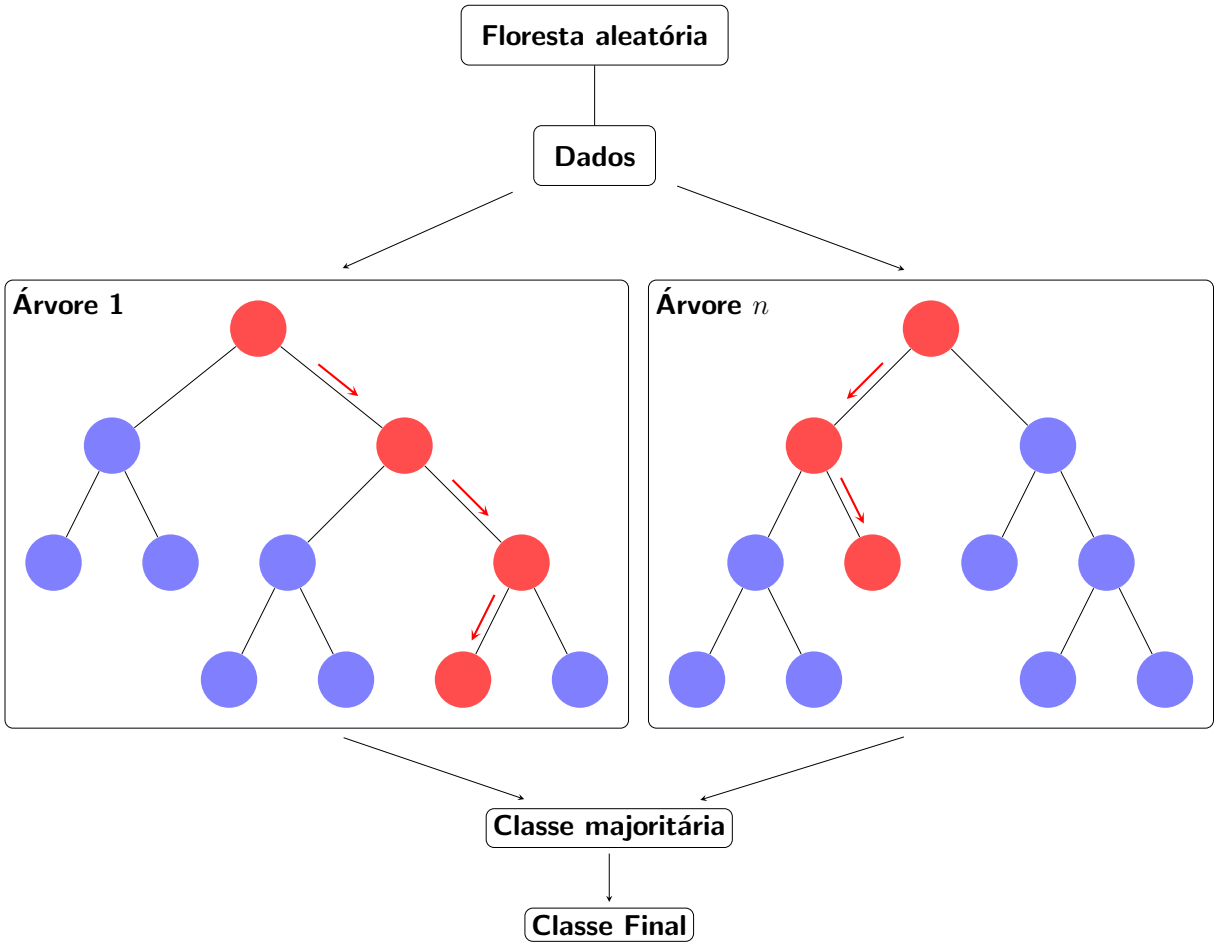
2.2.2.2 Floresta aleatória

Uma floresta aleatória é um conjunto de árvores de decisão, em que a classificação final será feita de acordo com a votação de todas as árvores da floresta. O método consiste em construir n árvores a partir de uma amostragem dos dados. Para a criação das amostras é utilizado o *bootstrap* (Rubin, 1981; Chernick et al., 2011), que é um método de reamostragem dos dados com ou sem repetição, ou seja, a seleção de um dado pode ocorrer mais de uma vez para a amostra. A seleção dos dados para compor a amostra acontece de forma aleatória, elas serão utilizadas para construir as n árvores de decisão (Breiman, 2001).

As árvores selecionam atributos desses elementos afim de classificá-los. Cada etapa desse processo de classificação representa um nó. Os elementos com classificação serão armazenados em uma árvore e os outros repassam pelo processo de seleção de seus atributos e classificação, resultando em um segundo nó. Esse processo se repete enquanto todos os elementos não classificados recebem uma classificação (de Alvarenga Júnior, 2018).

A decisão tomada em cada nó vem da contagem de votos realizada no conjunto de componentes preditoras, vence a classe com número superior de votos acumulados, ou seja, ao alcançar a maioridade dos votos. Espera-se dentro de cada árvore homogeneidade dos elementos, ou seja, elementos análogos, implicando em um desempenho satisfatório na decisão do algoritmo (de Alvarenga Júnior, 2018).

A Figura 7 apresenta o esquema de uma floresta aleatória. Em que se observa as árvores de decisão e suas classificações finais. Por fim, será apresentado o resultado final da floresta aleatória, voto majoritário de todas as árvores.



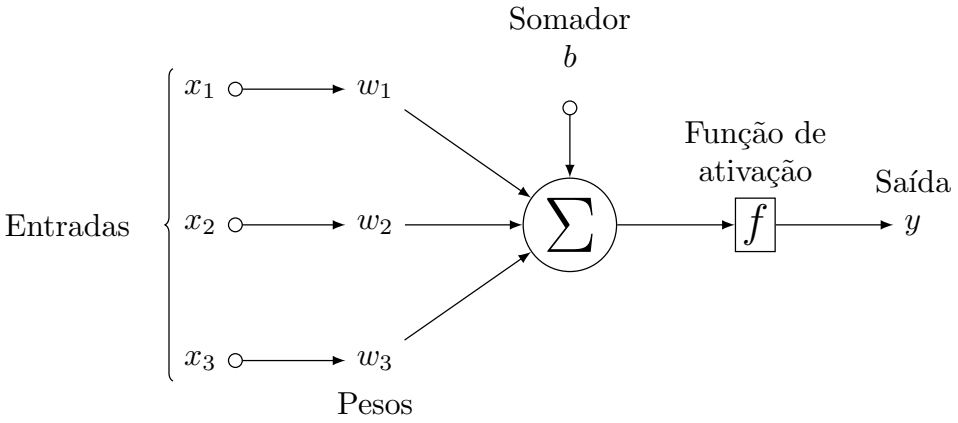
Figuras 7 – Exemplo de uma floresta aleatória

2.2.2.3 Redes neurais artificiais

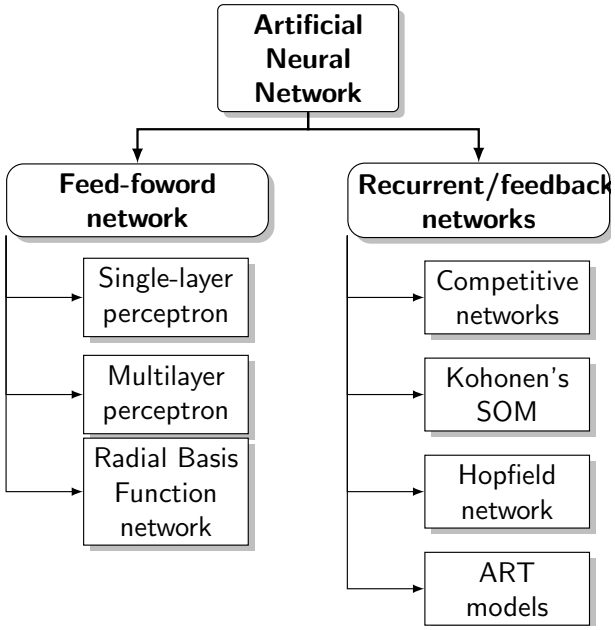
Redes neurais artificiais (RNA), frequentemente também conhecidas como redes neurais Khoudja et al. (2018), são técnicas computacionais, capazes de simular a forma de aprendizado do cérebro humano, ou seja, adquirir conhecimento baseado na experiência LeCun et al. (2015). Essa técnica permite que o algoritmo aprenda progressivamente com a experiência Barreto (2002). As RNA's possuem uma capacidade de resolver problemas e tarefas com alta complexidade, portanto são uma opção para a resolução do problema de integração de dados Khoudja et al. (2018).

Uma rede neural é composta por neurônios, que são as unidades fundamentais para a formação de uma RNA LeCun et al. (2015). A Figura 8 apresenta um modelo genérico de um neurônio.

Na Figura 8 percebe-se que cada sinal de entrada x_i é multiplicado pelo peso w_i. O *somador* é responsável pela combinação dos sinais de entrada ponderados, onde realiza-se a soma de todos os sinais. Por fim, a *função de ativação* é responsável por definir a ativação do neurônio, tipicamente a saída será dada com os valores 0 ou 1 Iyoda et al. (2000). Os neurônios são modelados como dispositivos de entrada-saída, conectados entre



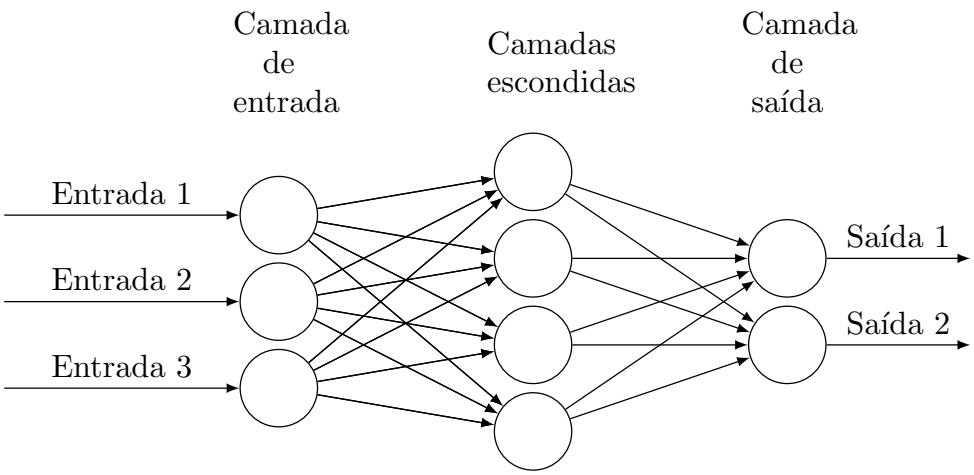
Figuras 8 – Exemplo de um neurônio artificial



Figuras 9 – Tipos de RNA. Adaptado de Gardner e Dorling (1998)

si em uma rede Noriega (2005). Os neurônios ao receber os dados de entrada, realizam o processamento dos mesmos e imputa-se os seus resultados em nova camada de neurônios, os quais efetuam o processo gerando novas saídas. A forma como esse processo ocorre é conhecido como propagação para frente (*feedfoword*) Rauber (2005). Os tipos de RNA seguem ilustrados na Figura 9.

Multilayer Perceptron (MLP) é uma das técnicas de RNA Gardner e Dorling (1998). A MLP consiste em um sistema de neurônios interconectados, ilustrado na Figura 10. O MLP é um modelo que representa o mapeamento não linear entre um vetor de entrada e um vetor de saída Gardner e Dorling (1998). Nesse processo os neurônios ficam dispostos em camadas, cada neurônio de uma camada possui conexão com todos os neurônios da camada posterior e da camada anterior, mas não possui conexões com neurônios da sua mesma camada Noriega (2005). Camadas escondidas, são as camadas



Figuras 10 – Esquema de uma MLP

que não possuem conexão com os dados de entrada ou a saída da rede Rauber (2005).

2.3 Utilização de aprendizado de máquina para integração de dados

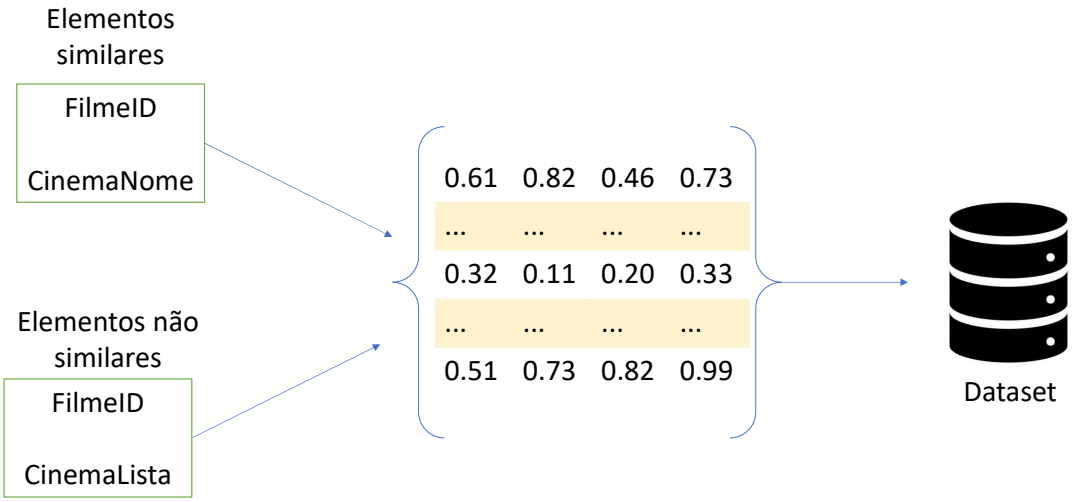
Uma das abordagens utilizadas para a integração de dados é o uso do aprendizado de máquina como paradigma principal, podendo tratar a tarefa de integração de dados de forma autônoma ou semiautônoma (Aumueller et al., 2005). A facilidade com que essa abordagem se adequa aos tipos de domínios tem trazido bons resultados (Silva et al., 2017). Dentre os métodos de aprendizado de máquina existentes na literatura para aplicar na integração de dados, escolheu-se o *neural network based schema matching* (NNSM) que utiliza uma RNA para a realização da tarefa de integração de dados, que apresentou bons resultados para base de dados relacional.

2.4 Utilização de rede neural artificial na integração de dados

Um dos métodos de aprendizado de máquina que utiliza RNA's para a integração de dados é o NNSM. O NNSM (Zhang et al., 2014) utiliza uma MLP para realizar a tarefa de integração de dados e funciona em duas fases.

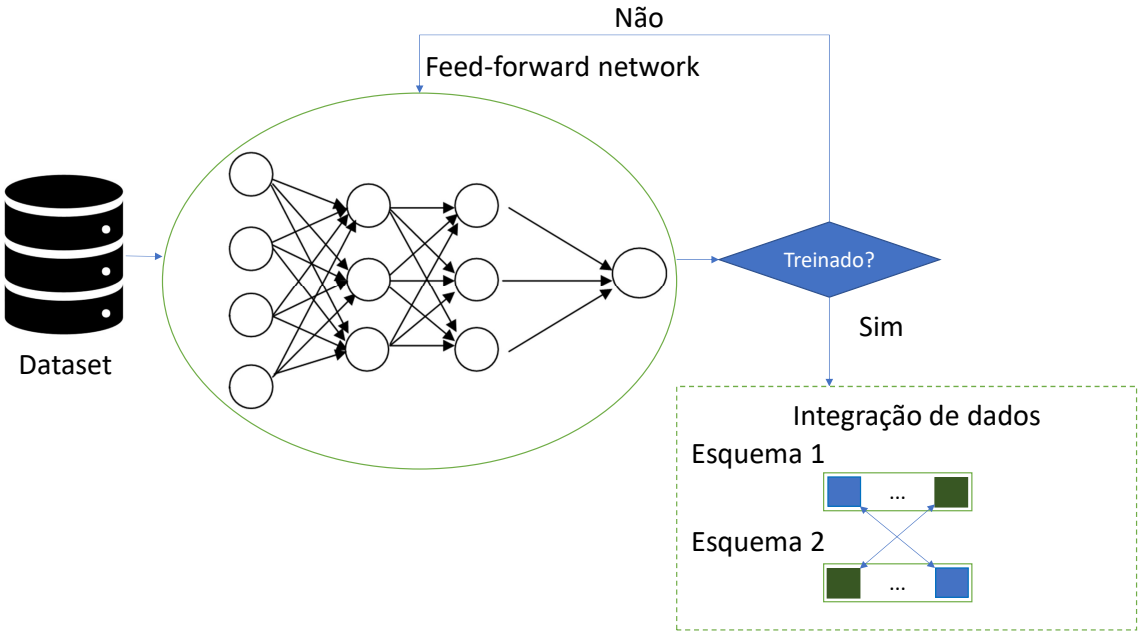
Na primeira fase, gerada uma matriz de similaridade do produto cartesiano das colunas das bases existentes utilizando o *Lin Matcher*, o *Wup Matcher*, o *Resnik Matcher* e o *Schutze Matcher*, que foram descritos nas seções anteriores. Criando um *dataset* com as correspondências existentes e outro com as não correspondentes, conforme apresentado pela Figura 11.

Na segunda fase, os dois *datasets*, um com as correspondências existentes e outro com as não correspondências, são unidos e armazenadas em um *dataset*. Uma parte dos



Figuras 11 – Criação do dataset. Adaptado de Zhang et al. (2014)

dados é selecionada de forma aleatória para treino e a outra parte será utilizada para teste. Então uma MLP é treinada, com 2 camadas escondidas, para encontrar futuras correspondências existentes, conforme apresentado pela Figura 12.



Figuras 12 – Treino da MLP. Adaptado de Zhang et al. (2014)

3 Trabalhos Relacionados

Apresenta-se neste capítulo uma visão geral das técnicas mais populares existentes na literatura para o problema de integração de dados que foram consideradas significativas para o desenvolvimento do trabalho.

Os autores Do e Rahm (2002) propõem uma forma de encontrar as correspondências semânticas na integração de dados. O estudo abordado propõe o uso do esquema de correspondências nomeado como *COMA*, que é composto pela combinação, via uma heurística fixa, do resultado de *matchers*, seguido de uma determinação da veracidade dos valores resultante. A heurística pode: Retornar o maior valor de similaridade encontrado pelos *matchers* utilizados; retornar à similaridade média encontrada pelos *matchers* utilizados; retornar à similaridade média ponderada encontrada pelos *matchers* utilizados, em que cada *matcher* recebe um peso; retornar o menor valor de similaridade encontrado pelos *matchers* utilizados. Para avaliar a veracidade de uma correspondência o *COMA* utiliza os critérios *Threshold*, *MaxN* e *MaxDelta*. O critério *Threshold* seleciona as correspondências que possuem valor superior ao valor estabelecido. O critério *MaxN* seleciona as s correspondências com maior similaridade. E o critério *MaxDelta* seleciona a correspondência com máxima similaridade e retorna todas as correspondências que possuem valor superior a um valor de tolerância, calculado de acordo que possui a correspondência com máxima similaridade selecionada.

Integrar dados de forma autônoma, assertiva e com o mínimo de intervenção humana no processo é o que Rodrigues et al. (2013) busca realizar neste estudo, com uma técnica chamada *Active Learning Matching* (*ALMa*). Segundo os estudos levantados por ele, uma boa parte dos métodos atuais utiliza-se de heurísticas para realizar a integração de dados e perante essa realidade ele propõe o uso de aprendizado ativo para realizar a tarefa de integração de dados e em um segundo momento fazer uma comparação dos resultados encontrados entre os métodos levantados e escolhidos por ele. O uso de aprendizado ativo em sua pesquisa alcançou valor médio da medida-f 0.64, demonstrando desempenho superior em relação aos outros métodos utilizados como linha de base. O *ALMa* funciona em quatro etapas: *Seleção*, *Treinamento*, *Eleição do Comitê* e *Votação*. Na *Seleção* realiza-se a seleção de pares para a rotulação humana. Na etapa de *Treinamento* realiza-se o treinamento de árvores de decisão com bases nos dados rotulados na etapa de *Seleção* e nos dados de similaridade retornados pelos *matchers* utilizados. Na etapa da *Eleição do Comitê* realiza-se a decisão de quais árvores de decisão treinadas apresentaram resultados confiáveis. Por fim na etapa de *Votação* realiza-se a avaliação por cada árvore selecionada na etapa anterior, para decidir se a correspondência é verdadeira ou falsa. Para correspondências com nível de confiança inferior ao valor mínimo aceitável realiza-se o ciclo

novamente, em que o critério de parada do algoritmo pode ser o número de interações com o usuário ou assim que o nível de confiança mínimo seja encontrado.

Os autores Zhang et al. (2014) propõem a utilização de um método de correspondência de esquema baseado em rede neural, também conhecido como *Neural Network Based Schema Matching* (NNSM). O método proposto visa combinar *matchers*, visto que a similaridade semântica calculada por cada *matcher* depende de aspectos individuais de informações sobre esquemas, que não são suficientes para encontrar correspondências de elementos entre esquemas. Para cada par de instâncias utiliza-se quatro *matchers* para encontrar o nível de similaridade entre o par. Com essa matriz de similaridade gerada pelos *matchers* realiza-se o treinamento de um *multilayer perceptron*. Após o treinamento a *multilayer perceptron* é executada e seus resultados analisados. O uso de *matchers* e da *multilayer perceptron* em sua pesquisa alcançou valor de medida-f $0,92$, para os cenários estudados, demonstrando desempenho superior em relação aos outros métodos utilizados como linha de base.

Fazer a integração de dados é de suma importância e relevância para as instituições, no entanto é uma tarefa trabalhosa. Mukkala et al. (2015) propõe o levantamento de métodos disponíveis na literatura para encontrar as correspondências entre os dados armazenados nos esquemas de entrada e então explorar, explicar e comparar a disponibilidade de código acessível, aplicabilidade, funcionalidade, desempenho, prós e contras no uso de cada um, afim de salientar o usuário sobre o assunto.

Alwan et al. (2017) menciona em seu estudo a importância da integração de dados e para que ela funcione de forma viável, ou seja, retorne as melhores correspondências deve-se atentar a forma como elas serão realizadas, uma vez que as fontes de dados de entradas são heterogêneas e podem estar estruturadas de várias formas. Com isso ele explora e compara os esquemas de banco dados como também as instâncias, promovendo um debate e conclusão sobre os métodos disponíveis na literatura.

Para Silva et al. (2017) a tarefa de integrar dados é a combinação de esquemas de entrada a partir de correspondências encontradas entre os dados existentes nos esquemas e geração de uma fonte de dados final. A integração de dados tem sido utilizada de forma acentuada no comércio eletrônico. Silva et al. (2017) faz a seleção de métodos disponíveis na literatura para fazer a integração de dados como: COMA, ALMa e RF4SM, para verificar se o uso de informações de instância melhora os resultados, no caso, foi concluído que sem elas alcançou-se resultados satisfatórios para o estudo.

Neste outro estudo, Rodrigues et al. (2018) aborda de forma detalhada o casamento de esquemas, tarefa esta desafiadora devido a forma como os esquemas de entrada apresentam-se, tornando por vezes a busca por correspondências trabalhosas. Como proposta ele utiliza métodos de *matchers* baseados em aprendizado de máquina e para facilitar o treino automático ele utilizou a técnica de *bootstrap*. Com a ajuda de um especialista

ele faz as avaliações finais, alcançando bons resultados, com a medida-f podendo chegar a 0.83.

4 METODOLOGIA

4.1 Natureza da pesquisa

O estudo exposto refere-se a uma pesquisa de natureza aplicada, pois possui o objetivo de gerar conhecimentos para a aplicação prática, voltada para a solução de problemas (Silveira e Córdova, 2009). O resultado desta pesquisa deve ser uma avaliação das técnicas aplicadas a um banco de dados criado.

Esse estudo é classificado como uma pesquisa experimental que, segundo Gil (2008), é um tipo de pesquisa que consiste em determinar um objeto de estudo, selecionar as variáveis capazes de influenciá-lo, definir formas de controle e observação dos impactos que essas variáveis podem ter no objeto.

Em relação ao objetivo, classifica-se o estudo proposto como pesquisa descritiva que, para Silveira e Córdova (2009), é um tipo de estudo realizado com a pretensão de descrever os fatos e fenômenos de uma realidade estabelecida.

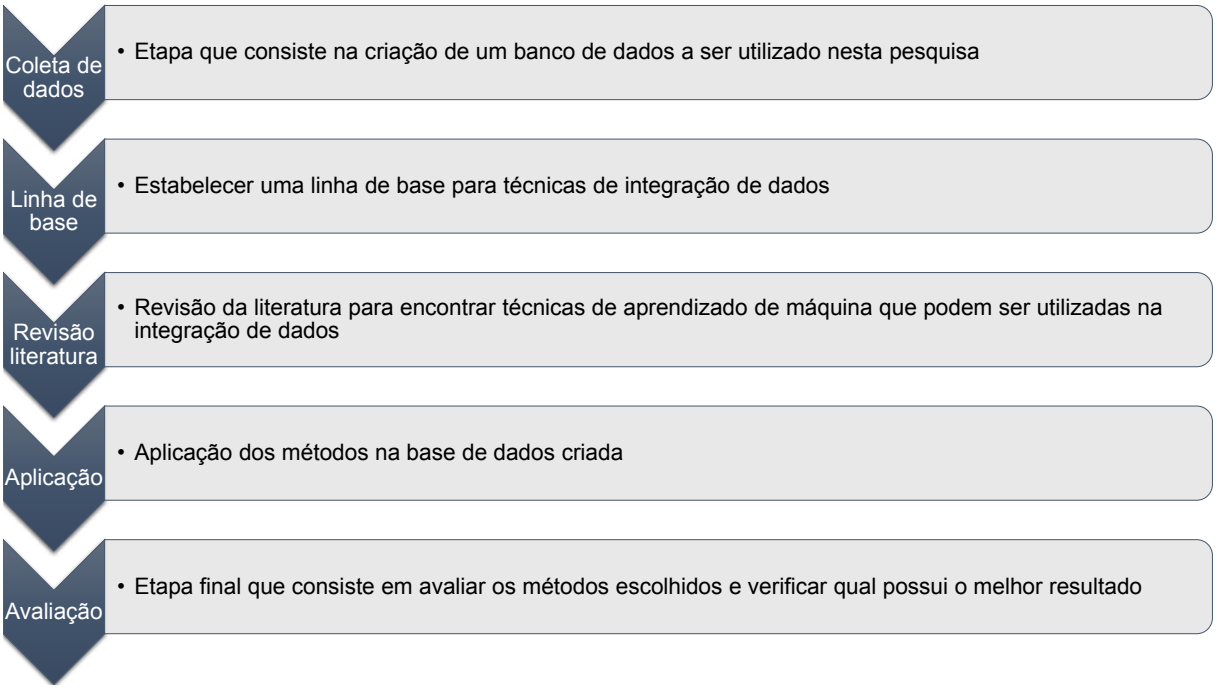
4.2 Proposta de pesquisa

O estudo tem como objetivo analisar a aplicação de técnicas para a realização da tarefa de integração de dados. O fluxo do desenvolvimento do trabalho segue apresentado na Figura 13. Para verificar quais os métodos se adequam aos dados disponibilizados, realizou-se a análise manual do banco de dados para a geração das correspondências existentes, a criação de uma linha de base, a criação de proposta, derivada da linha de base, e a avaliação dessa proposta em relação a linha de base.

Para atingir o objetivo base do trabalho, divide-se o estudo em duas etapas. A primeira etapa consiste na definição da linha de base. A linha de base estabelecida, foi a do autor Zhang et al. (2014), que apresentou resultados, com medida-f $0,92$, respondendo ao objetivo específico 1 da pesquisa. A segunda etapa consiste na comparação da linha de base com outros métodos. Ela será utilizada para comparação com a utilização de variações da MLP proposta pelo autor para a linha de base, respondendo os objetivos específicos 2 e 3.

4.2.1 Primeira etapa

A primeira etapa dedica-se a responder o **objetivo 1:** estabelecer uma linha de base para as técnicas de integração de dados. Para a definição dos métodos que serão utilizados como linha de base realizou-se uma revisão da literatura, em que se utilizou o



Figuras 13 – Fluxo de desenvolvimento da pesquisa.

artigo de Shvaiko e Euzenat (2005) como fonte primária de informação. Utilizando técnica de *Snowball*, descrita por Goodman (1961) como uma amostragem não probabilista, em que se procura por outros artigos que o citaram.

Como linha de base utilizou-se o método *NNSM* (Zhang et al., 2014), levantado na revisão bibliográfica, em que sua escolha se deu por apresentar bons resultados em suas aplicações (Zhang et al., 2014).

4.2.2 Segunda etapa

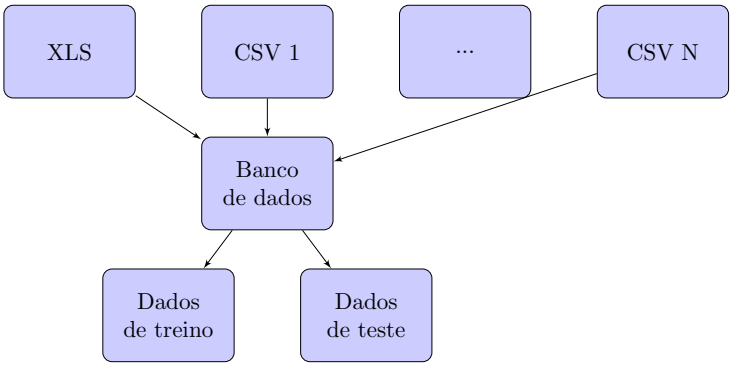
A segunda etapa dedica-se a responder os objetivos 2 e 3. Em que o **objetivo 2**: visa identificar os *matchers* e o **objetivo 3**: visa comparar os métodos de aprendizado de máquina no processo de integração de dados.

Para a identificação desses *matchers* realiza-se uma revisão da literatura utilizando a mesma estratégia do *Snowball*, utilizando como artigo seminal o trabalho de Schütze et al. (2008). Selecionou-se o *matcher* que calcula a similaridade pelo cosseno, levantado na revisão bibliográfica, em que sua escolha se deu por ser um *matcher* que compara a similaridade entre textos (Schütze et al., 2008).

4.3 Banco de Dados

4.3.1 Estrutura do banco de dados

A etapa de desenvolvimento do banco de dados visa determinar quais conjuntos de dados serão alinhados. O fluxo de desenvolvimento do banco de dados segue apresentado na Figura 14. A entrada será feita pelas tabelas de dados existentes, em que os dados de cada base serão recebidos em um arquivo CSV e um arquivo XLS contendo os metadados e no caso de ser o treinamento do modelo o mapeamento correto das bases de entrada.



Figuras 14 – Fluxo de desenvolvimento do banco de dados.

Os arquivos CSV de entrada devem possuir o formato conforme segue exemplificado na Tabela 2, contendo na primeira linha o nome das n colunas armazenadas e nas outras linhas apresentam-se os m dados. O CSV deve ter padrão de separação com vírgula (,), separador de decimal ponto (.) e codificação *latin*.

Nome 1	Nome 2	Nome 3	Nome 4	Nome 5
Dado1.a	Dado1.b	Dado1.c	Dado1.d	Dado1.e
Dado2.a	Dado2.b	Dado2.c	Dado2.d	Dado2.e
Dado3.a	Dado3.b	Dado3.c	Dado3.d	Dado3.e
Dado4.a	Dado4.b	Dado4.c	Dado4.d	Dado4.e
Dado5.a	Dado5.b	Dado5.c	Dado5.d	Dado5.e

Tabelas 2 – Tabela com exemplos de CSV de entrada.

No arquivo XLS estarão disponíveis os metadados das bases de dados, conforme será exemplificado na Tabela 3, e no caso do treinamento do modelo também estará disponível o mapeamento correto das bases de entrada.

Cada linha do XLS apresenta os atributos: **Nome da tabela** apresentando o nome do CSV de entrada; **Nome da coluna** apresentando o nome da coluna da tabela de entrada; **Tipo da coluna** apresentando o tipo da coluna em questão, podendo ser: *Numérico* em que os valores são numéricos, podendo conter casas decimais; *Data* em que se apresenta a data, podendo conter a hora; *Caracter* para caracteres. A coluna **Código**

Nome da tabela	Nome da coluna	Tipo da coluna	Código da coluna
ASC_Facility	Facility ID	Caracter	1
ASC_Facility	City	Caracter	4
HCAHPS - Hospital	Facility Name	Caracter	2
HCAHPS - Hospital	City	Caracter	4
VA_IPF	Facility ID	Caracter	1
VA_IPF	End Date	Date	30

Tabelas 3 – Tabela com exemplos de CSV de entrada.

da coluna apresenta o código para encontrar os mapeamentos existentes, isto é, colunas com código igual representam uma correspondência correta.

4.3.2 Bases de dados

As bases de dados utilizadas na pesquisa são bases de dados hospitalares públicas. É possível encontrá-las em https://data.cms.gov/provider-data/. Das bases de dados disponíveis selecionou-se 80 bases de dados para a pesquisa proposta, foram removidas 4 bases de dados do estudo, sendo 3 delas por não condizerem com o descritivo que a acompanhava e 1 por estar corrompida, restando 76 bases de dados para o estudo. Cada base de dados está salva como um arquivo CSV, conforme descrito em 4.3. O número de linhas e colunas podem variar para cada uma delas.

As bases de dados apresentam um total 1232 variáveis, no entanto foram realizadas 197 codificações diferentes, ou seja, existem casos com variáveis representando informação semelhante, conforme as condições apresentadas e descritas em 2.1.1. A Tabela 4 contém o nome e o descritivo de cada uma das bases de dados utilizadas.

Tabelas 4 – Bases de dados utilizadas na pesquisa

Nome da Base de Dados	**Descritivo**
ASC CCN pr18q2 19q1	Resultados em nível de hospital para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatória de Prestadores de Serviços de Saúde e Sistemas para Centros Cirúrgicos Ambulatoriais.
ASC Facility	Resultados em nível de unidade de saúde para medidas do Programa de Relatórios de Qualidade de Centro Cirúrgico Ambulatorial.

Continua

ASC National	Resultados em nível nacional para medidas do Programa de Relatórios de Qualidade de Centro Cirúrgico Ambulatorial.
ASC NATIONAL pr18q2 19q1	Resultados de nível nacional para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatória de Prestadores de Serviços de Saúde e Sistemas para Centros Cirúrgicos Ambulatoriais.
ASC State	Resultados em nível estadual para medidas do Programa de Relatórios de Qualidade do Centro Cirúrgico Ambulatorial.
ASC STATE pr18q2 19q1	Resultados em nível estadual para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatorial de Provedores e Sistemas de Saúde para centros cirúrgicos ambulatoriais.
CEBP IQR 10 17 2018	Compra baseada em episódios clínicos em nível hospitalar.
CEBP IQR 6 decimals 10 17 2018	Aquisição baseada em episódios clínicos por nível de instalação exibida em até 6 casas decimais.
CEBP IQR Breakdowns by Claim Ty	Discriminação de compra baseada em episódios clínicos por tipo de reclamação.
CEBP IQR national 10 17 2018	Compra baseada em episódios clínicos em nível nacional.
CEBP IQR state 10 17 2018	Compra baseada em episódios clínicos em nível estadual.
CJR PY3 Quality Reporting July 2019 Production File	Taxa de complicações para pacientes com substituição de quadril/joelho e pontuação média linear de roll-up do HCAHPS.
CMS PSI 6 decimal file	CMS PSI-90 e medidas de componentes por instalação exibidas em 6 decimais.
Complications and Deaths - Hospital	Resultados em nível hospitalar para complicações cirúrgicas e medidas de mortalidade.
Complications and Deaths - Nacional	Resultados em nível nacional para complicações cirúrgicas e medidas de mortalidade.
Complications and Deaths - State	Resultados em nível estadual para complicações cirúrgicas e medidas de mortalidade.

Continua

DoD TE October 2019 Production	Dados em nível de hospital do Departamento de Defesa para medidas de atendimento oportunas e eficazes.
DoD TRISS Final File October 2019	Dados em nível de hospital do Departamento de Defesa para pesquisas de satisfação de pacientes internos da TRICARE.
Footnote Crosswalk	Procure na tabela o texto do resumo da nota de rodapé.
footnotes deliver 18q2 19q1	Procure na tabela o texto do resumo da nota de rodapé para arquivos OAS.
FY2017 Distribution of Net Change in Base Op DRG Payment Amt 2018-11-30	Distribuição da variação líquida no valor de pagamento do grupo relacionado ao diagnóstico operacional de base.
FY2017 Net Change in Base Op DRG Payment Amt 2018-11-30	Alteração líquida no valor de pagamento do grupo relacionado ao diagnóstico operacional de base.
FY2017 Percent Change in Medicare Payments 2018-12-03	Alteração percentual no valor de pagamento do grupo relacionado ao diagnóstico operacional de base.
FY2017 Value Based Incentive Payment Amount 2018-11-30	Montante de pagamento de incentivo baseado em valor.
HCAHPS - Hospital	Resultados em nível de hospital para a Avaliação de Consumidores de Hospitais de Provedores e Sistemas de Saúde.
HCAHPS - National	Resultados de nível nacional para a Avaliação de Consumidores de Hospitais de Provedores e Sistemas de Saúde.
HCAHPS - State	Resultados em nível estadual para a Avaliação de Consumidores de Hospitais de Provedores e Sistemas de Saúde.
Healthcare Associated Infections - Hospital	Resultados em nível de hospital para medidas de infecções associadas a cuidados de saúde.
Healthcare Associated Infections - National	Resultados a nível nacional para medidas de infecções associadas aos cuidados de saúde.
Healthcare Associated Infections - State	Resultados em nível de estado para medidas de infecções associadas à saúde.

Continua

HOPD CCN pr18q2 19q1	Resultados em nível de hospital para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatorial de Prestadores de Serviços de Saúde e Sistemas para departamentos ambulatoriais de hospitais.
HOPD NATIONAL pr18q2 19q1	Resultados de nível nacional para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatorial de Prestadores de Serviços de Saúde e Sistemas para departamentos ambulatoriais de hospitais.
HOPD STATE pr18q2 19q1	Resultados em nível estadual para a Avaliação do Consumidor de Cirurgia Ambulatorial e Ambulatorial de Provedores de Saúde e Sistemas para departamentos ambulatoriais de hospitais.
Hospital General Information	Informações gerais sobre hospitais no conjunto de dados.
HOSPITAL ANNUAL QUALITYMEASURE PCH OCM Hospital	Resultados em nível de hospital para medidas de cuidados oncológicos do Programa de Relatórios de Qualidade de Hospital de Câncer isento de PPS.
HOSPITAL QUARTERLY HAC DOMAIN HOSPITAL	Resultados em nível de hospital para medidas do Programa de Redução de Condições Adquiridas no Hospital.
HOSPITAL QUARTERLY MSPB 6 DECIMALS	Gastos com Medicare por Beneficiário por instalação exibidos em 6 casas decimais.
HOSPITAL QUARTERLY QUALITYMEASURE PCH HAI HOSPITAL	Resultados em nível de hospital para medidas de infecções associadas à assistência médica do Programa de Relatórios de Qualidade de Hospital de Câncer isento de PPS.
HOSPITAL QUARTERLY QUALITYMEASURE PCH HCAHPS HOSPITAL	Resultados em nível de hospital para Programa de Relatório de Qualidade de Hospital de Câncer isento de PPS para as medidas de domínio de experiência do paciente.

Continua

HOSPITAL QUARTERLY QUALITYMEASURE PCH HCAHPS NATIONAL	Resultados de nível nacional para o Programa de Relatórios de Qualidade do Hospital de Câncer isento de PPS para as medidas de domínio de experiência do paciente.
HOSPITAL QUARTERLY QUALITYMEASURE PCH HCAHPS STATE	Resultados em nível estadual para o Programa de Relatórios de Qualidade do Hospital de Câncer isento de PPS para as medidas de domínio de experiência do paciente.
HOSPITAL QUARTERLY QUALITYMEASURE RRP HOSPITAL	Resultados em nível de hospital para medidas do Programa de Redução de Readmissões em hospitais.
HOSPITAL QUARTERLY QUALITYMEASURE RRP HOSPITAL	Resultados em nível de hospital para medidas do Programa de Redução de Readmissões em hospitais.
hvbp clinical care 11 09 2018	Resultados em nível de hospital sobre medidas de domínio de resultado para compras com base em valor hospitalar.
hvbp efficiency 11 09 2018	Resultados em nível de hospital sobre medidas de domínio de eficiência para compras baseadas em valor hospitalar.
hvbp tps 11 09 2018	Pontuação total de desempenho em nível de hospital para compras com base em valor hospitalar.
IPFQR QualityMeasures Facility	Resultados a nível de hospital para medidas do Programa de Relatórios de Qualidade de Instalações Psiquiátricas Internas.
IPFQR QualityMeasures National	Resultados a nível nacional para medidas do Programa de Relatórios de Qualidade de Instalações Psiquiátricas Internas.
IPFQR QualityMeasures State	Resultados em nível estadual para medidas do Programa de Relatórios de Qualidade de Instalações Psiquiátricas Internas.
Measure Dates	Datas de coleta atuais para todas as medidas no Hospital Compare.
Medicare Hospital Spending by Claim	Desdobramentos de despesas do Medicare por beneficiário por tipo de sinistro.

Continua

Medicare Hospital Spending Per Patient - Hospital	Gastos com Medicare em nível de hospital por beneficiário.
Medicare Hospital Spending Per Patient - National	Gastos com Medicare em nível nacional por beneficiário.
Medicare Hospital Spending Per Patient - State	Gastos com Medicare em nível estadual por beneficiário.
Outpatient Imaging Efficiency - Hospital	Resultados em nível de hospital para medidas do uso de imagens médicas.
Outpatient Imaging Efficiency - National	Resultados em nível nacional para medidas do uso de imagens médicas.
Outpatient Imaging Efficiency - State	Resultados em nível estadual para medidas do uso de imagens médicas.
Outpatient Procedures - Volume	Volume de procedimentos cirúrgicos ambulatoriais de hospitais.
Payment - National	Resultados a nível nacional para medidas de pagamento.
Payment - State	Resultados em nível estadual para medidas de pagamento.
Payment and Value of Care - Hospital	Resultados em nível de hospital para medidas de pagamento e exibições de valor de atendimento associadas a medidas de mortalidade em 30 dias.
Readmissions and Deaths - COPD - VA	Dados de nível hospitalar da Veterans Health Administration para mortalidade por DPOC e medidas de readmissão.
Structural Measures - Hospital	Resultados em nível de hospital para medidas estruturais.
Timely and Effective Care - Hospital	Resultados em nível de hospital para medidas de processo de atendimento.
Timely and Effective Care - National	Resultados a nível nacional para medidas do Processo de Cuidado.
Timely and Effective Care - State	Resultados em nível estadual para medidas de processo de atendimento.
Unplanned Hospital Visits - Hospital	Resultados em nível de hospital para medidas de readmissão de 30 dias e dias de retorno ao hospital.

Continua

Unplanned Hospital Visits - National	Resultados a nível nacional para medidas de readmissão de 30 dias e dias de retorno ao hospital.
Unplanned Hospital Visits - State	Resultados em nível estadual para medidas de readmissão de 30 dias e dias de retorno ao hospital.
VA IPF	Dados em nível de hospital da Veterans Health Administration para medidas de saúde comportamental.
VA PSI	Dados de nível hospitalar da Veterans Health Administration para indicadores de segurança do paciente.
VA TE	Dados em nível de hospital da Veterans Health Administration para medidas de cuidado oportunas e eficazes.
Value of Care - National	Resultados em nível nacional para valores de exibição de cuidados associados a medidas de mortalidade em 30 dias
Veterans Health Administration Measure Dates	As datas de coleta atuais da Veterans Health Administration para todas as medidas no Hospital Compare
Veterans Health Administration Provider Level Data	Informações gerais sobre hospitais da Veterans Health Administration

4.4 Ferramenta de análise

Desenvolveu-se este estudo em um Notebook Acer com processador Intel Core i7-3612QM, com seis núcleos, 8gb de memória RAM de uso próprio do pesquisador. Utilizou-se os seguintes softwares: o sistema operacional Windows 10 Pro e o Microsoft Office 365. O tratamento dos dados e a execução dos algoritmos foram realizados por meio da linguagem de programação Python.

4.4.1 Python

Criado por Van Rossum e Drake Jr (1995), Python é uma linguagem de programação de alto nível e difundida devido ao seu ecossistema amplo e ativo de pacotes de

terceiros VanderPlas (2016). Como em outras linguagens, Python apresenta bibliotecas com foco na aplicabilidade em estudos que envolvem a análise de dados VanderPlas (2016). O algoritmo gasta em torno de 12h para sua execução completa. Para este estudo fez-se uso das seguintes bibliotecas:

- Pandas e Numpy: utilizadas para manipulação e tratamento de dados;
- Sklearn e Tensorflow: utilizadas para elaboração de modelos de aprendizado de máquina;
- Sematch: utilizada para os cálculos de similaridade de palavras;
- Matplotlib: utilizada para elaboração de gráficos;
- Keras: utilizada para elaboração de modelos redes neurais artificiais.

4.5 Unidade de análise

Os *Centers for Medicare & Medicaid Services* (CMS) criaram o *Hospital Compare*, um site que oferece aos seus usuários informações sobre a qualidade de atendimento oferecida por hospitais aos seus pacientes. Desta forma, um potencial paciente pode escolher qual o hospital indicado para atendê-lo. Eles oferecem informações sobre aproximadamente 4.000 hospitais diferentes. Para que o hospital possa fazer parte do site, ele passa por um processo de certificação realizado pelo Medicare, em caso de aprovação o hospital passa a ter suas informações apresentadas no site (for Medicare et al., 2016).

O Hospital Compare oferece essa informação na forma de comparação, ou seja, o usuário acessa o site e escolhe 2 (dois) hospitais que tem interesse em utilizar o serviço e o site retorna um comparativo entre eles para o setor específico que o usuário deseja saber, como por exemplo o setor de atendimento de emergência. Munido de informação o usuário pode fazer a uma escolha com embasamento (for Medicare et al., 2016).

O comparativo e outras informações são encontrados acessando o endereço `www.medicare.gov/hospitalcompare`. A atualização do site ocorre trimestralmente e as informações sobre os hospitais, as bases de dados, dentre outras informações são encontradas nele (for Medicare et al., 2016).

4.6 Classificação

Nesse momento do estudo deve-se fazer a validação dos modelos, conhecida como validação cruzada, ou seja, com a aplicação dessa técnica espera-se estimar o desempenho do modelo treinado (de Alvarenga Júnior, 2018).

Esta técnica consiste em dividir a base de dados em duas partes, ou seja, dos dados disponíveis para o estudo parte será destinada ao treinamento do modelo e parte para validação/teste do modelo. Os dados utilizados para treinamento do modelo correspondem a 80% de todos os dados e os dados utilizados para testes correspondem a 20% de todos os dados disponíveis (de Alvarenga Júnior, 2018).

Uma das formas de realizar a divisão dos dados em **treino** e **teste** é por meio da técnica de amostragem aleatória, em que todas as observações tem probabilidade similar de sorteio para entrar em qualquer uma das bases de dados (Ishwaran et al., 2008; de Alvarenga Júnior, 2018). Os dados utilizados no treinamento não serão utilizados nos testes (de Alvarenga Júnior, 2018).

4.7 Métricas de avaliação

As métricas de validação do modelo proposto, serão dadas pela: acurácia, precisão, revocação e medida-F, métricas estas muitos utilizadas no contexto da integração de dados (Rahm e Bernstein, 2001; Silva et al., 2017; Rodrigues et al., 2018). As equações que retornam essas medidas de avaliação do modelo são apresentadas a seguir:

- **Acurácia**
  A **acurácia**, apresentada pela equação 4.1, das correspondências presentes, quantas o modelo classificou corretamente.

$$Acurácia = \frac{Número\ de\ correspondências\ e\ não\ correspondências\ corretas\ encontradas}{Número\ total\ de\ correspondências\ e\ não\ correspondências} \tag{4.1}$$

- **Revocação**
  A **revocação**, apresentada pela equação 4.2, é dada pela fração das correspondências corretas que serão recuperadas.

$$Revocação = \frac{Número\ de\ correspondências\ corretas}{Número\ de\ correspondências\ existentes} \tag{4.2}$$

- **Precisão**
  A **precisão**, apresentada pela equação 4.3, é dada pela razão das correspondências recuperadas que estão corretas.

$$\text{Precisão} = \frac{Número\ de\ correspondências\ corretas}{Número\ de\ correspondências\ encontradas} \tag{4.3}$$

- **Medida-F**
  E a **medida-f**, apresentada pela equação 4.4, é a média harmônica entre *revocação* e *precisão*.

$$\textit{Medida-F} = \frac{2 \times (\textit{Precisão} \times \textit{Revocação})}{\text{Precisão} + \text{Revocação}} \tag{4.4}$$

5 Resultados e discussões

Esse capítulo apresenta uma avaliação experimental sobre o uso de RNA's para a integração de dados. Sendo o objetivo principal verificar experimentalmente se a utilização de um maior número de *matchers* influenciam nos resultados.

Para a realização desse experimento realizamos uma extrapolação do método NNSM de Zhang et al. (2014). Como Zhang et al. (2014) não especifica todos os hiperparâmetros utilizados para a criação de uma MLP, foram criadas cinco versões distintas, para a avaliação dos resultados. Em que, é analisado para cada MLP criada seus resultados para a base de dados especificada no Seção 4.3

Nas seções seguintes, são apresentadas as configurações utilizadas para cada extrapolação do método utilizado, uma análise sobre os *matchers* que foram utilizados, os resultados dos experimentos e por fim, as considerações finais sobre os resultados obtidos.

5.1 Configuração do método NNSM

Os experimentos foram realizados utilizando cinco extrapolações distintas do método NNSM de Zhang et al. (2014). As configurações utilizadas e o treinamento foram os mesmos para todas as extrapolações criadas. As configurações utilizadas para a montagem das extrapolações criadas foram: Para a função de otimização foi selecionada a função *Adam* de Kingma e Ba (2014), função responsável pela atualização dos pesos da RNA. Segundo Kingma e Ba (2014) a função de otimização *Adam* é "um algoritmo para otimização baseada em gradiente de primeira ordem de funções objetivo estocásticas, com base em estimativas adaptativas de momentos de ordem inferior". A função de perda utilizada foi a *binary cross-entropy*, função que calcula a diferença entre os dados de teste e os dados de validação. As métricas avaliadas foram as descritas no Seção 4.7. Para todos os treinamentos foram utilizadas 150 épocas e tamanhos do lote 50, que define o número de amostras que serão propagadas pela RNA.

A função padrão de perda *binary cross-entropy* é definida como:

$$J_bce = -\frac{1}{M}\sum_{m=1}^{M}[y_m * log(h_\theta(x_m)) + (1 - y_m) * log(1 - (h_\theta(x_m))] \tag{5.1}$$

Em que M é o número de exemplos de treinamento, y_m é a variável de resultado do exemplo de treinamento m, x_m é entrada para o exemplo de treinamento m e h_θ é modelo com pesos da RNA θ.

5.1.1 Primeira MLP

A primeira MLP criada, nomeada de *rede_1*, possui as camadas de entrada e saída e duas camadas escondidas. A camada de entrada possui o número de neurônios igual ao número de *matchers* utilizado, possuindo três saídas, a função de ativação foi a $ReLU$, definida como $f(x) = max(0, x)$ LeCun et al. (2015). A primeira camada escondida possui seis neurônios, e a função de ativação $ReLU$. A segunda camada escondida possui três neurônios, e a função de ativação $ReLU$. A camada de saída possui um neurônio, e a função de ativação *sigmoid*, definida como $f(x) = (1 + exp(-x))^{-1}$.

5.1.2 Segunda MLP

A segunda MLP criada, nomeada de *rede_2*, possui as camadas de entrada e saída e três camadas escondidas. A camada de entrada possui o número de neurônios igual ao número de *matchers* utilizado, possuindo três saídas, a função de ativação foi a $ReLU$. A primeira camada escondida possui seis neurônios, e a função de ativação $ReLU$. A segunda camada escondida possui doze neurônios, e a função de ativação $ReLU$. A terceira camada escondida possui seis neurônios, e a função de ativação $ReLU$. A camada de saída possui um neurônio, e a função de ativação *sigmoid*.

5.1.3 Terceira MLP

A terceira MLP criada, nomeada de *rede_3*, possui as camadas de entrada e saída e quatro camadas escondidas. A camada de entrada possui o número de neurônios igual ao número de *matchers* utilizado, possuindo três saídas, a função de ativação foi a $ReLU$. A primeira camada escondida possui seis neurônios, e a função de ativação $ReLU$. A segunda camada escondida é uma camada de *Dropout*, com hiperparâmetro aleatório de $0,2$. *Dropout* é uma técnica para resolver problemas com *overfitting*, a ideia é descartar aleatoriamente, com a probabilidade passada via hiperparâmetro, unidades e suas conexões da rede neural durante o treinamento Srivastava et al. (2014). A terceira camada escondida possui três neurônios, e a função de ativação $ReLU$. A quarta camada escondida é uma camada de *Dropout*, com hiperparâmetro aleatório de $0,2$. A camada de saída possui um neurônio, e a função de ativação *sigmoid*.

5.1.4 Quarta MLP

A quarta MLP criada, nomeada de *rede_4*, possui as camadas de entrada e saída e seis camadas escondidas. A camada de entrada possui o número de neurônios igual ao número de *matchers* utilizado, possuindo três saídas, a função de ativação foi a $ReLU$. A primeira camada escondida possui seis neurônios, e a função de ativação $ReLU$. A segunda camada escondida é uma camada de *Dropout*, com hiperparâmetro aleatório de $0,2$. A

terceira camada escondida possui doze neurônios, e a função de ativação *ReLU*. A quarta camada escondida é uma camada de *Dropout*, com hiperparâmetro aleatório de $0,2$. A quinta camada escondida possui seis neurônios, e a função de ativação *ReLU*. A sexta camada escondida é uma camada de *Dropout*, com hiperparâmetro aleatório de $0,2$. A camada de saída possui um neurônio, e a função de ativação *sigmoid*.

5.1.5 Quinta MLP

A quinta MLP criada, nomeada de *rede_5*, possui as camadas de entrada e saída e uma camada escondida. A camada de entrada possui o número de neurônios igual ao número de *matchers* utilizado, possuindo três saídas, a função de ativação foi a *ReLU*. A camada escondida possui seis neurônios, e a função de ativação *ReLU*. A camada de saída possui um neurônio, e a função de ativação *sigmoid*.

5.2 Matchers Utilizados

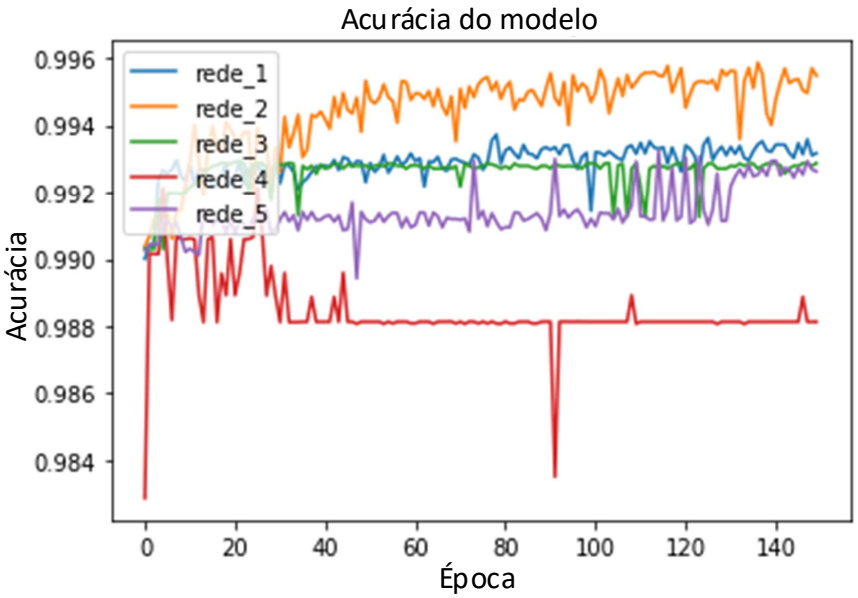
Os experimentos foram realizados utilizando duas configurações distintas de *matchers*: A primeira, nomeada de *linha de base*, utilizando os *matchers* previamente utilizados em Zhang et al. (2014). Utilizando o **Lin Matcher**, descrito na subseção 2.1.2.1, o **Wup matcher**, descrito na subseção 2.1.2.2, o **Resnik matcher**, descrito na subseção 2.1.2.3 e o **Schutze matcher** descrito na subseção 2.1.2.4.

A segunda configuração, nomeada de *proposta*, utiliza os mesmos *matchers* que são usados na primeira configuração, além dos *matchers* de **Zhu Matcher**, descrito na subseção 2.1.2.7, o **Tata Matcher**, descrito na subseção 2.1.2.5 e o **Li Matcher**, descrito na subseção 2.1.2.6. Também foram utilizadas duas outras técnicas: a primeira propõe o uso de um indicador do tipo da primeira e da segunda coluna em questão; a segunda proposta sugere um contador do número de palavras e letras da primeira e da segunda coluna em questão.

5.3 Resultados dos experimentos

Para analisar a eficiência das MLP's e das configurações de *matchers* criadas, foram executados experimentos no banco de dados, especificado na seção 4.3, nas MLP's criadas *rede_1*, *rede_2*, *rede_3*, *rede_4* e *rede_5*. Os resultados foram analisados em nível de acurácia, precisão, revocação e medida-F, comparando a *linha de base* com a *proposta*.

O limiar utilizado foi $0,5$, a base selecionada é desbalanceda, a base de treino possui 469.680 não correspondências e 10.176 correspondências, totalizando $2,12\%$ de correspondências corretas. A base de teste possui 201.377 não correspondências e 4.276 correspondências, totalizando $2,08\%$ de correspondências corretas.



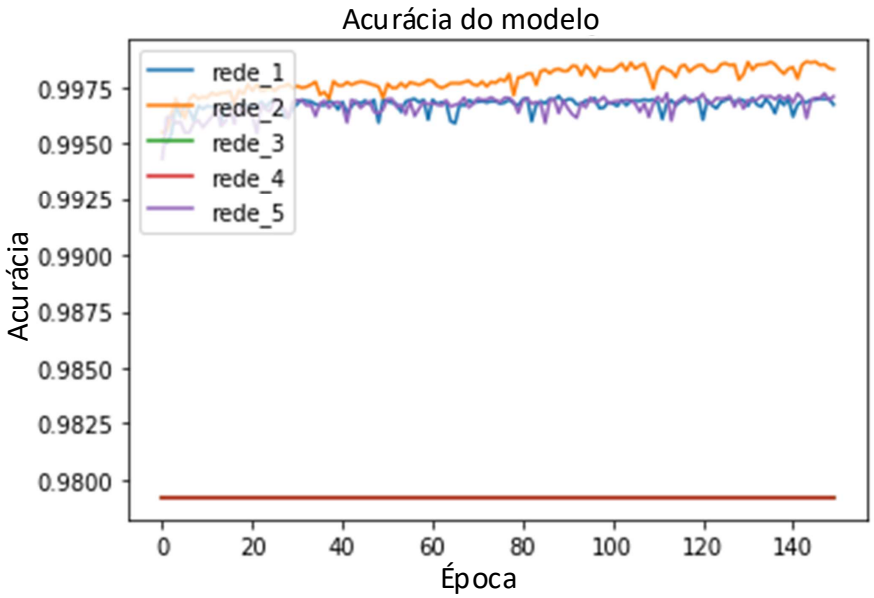
Figuras 15 – Acurácia nos experimentos com as configurações da *linha de base*.

As Figuras 15 e 16 apresentam as evoluções da métrica acurácia durante as 150 épocas de treinamento para as cinco MLP criadas *rede*_1, *rede*_2, *rede*_3, *rede*_4 e *rede*_5, utilizando na Figura 15 as configurações do primeiro experimento *linha de base* e na Figura 16 as configurações do segundo experimento *proposta*.

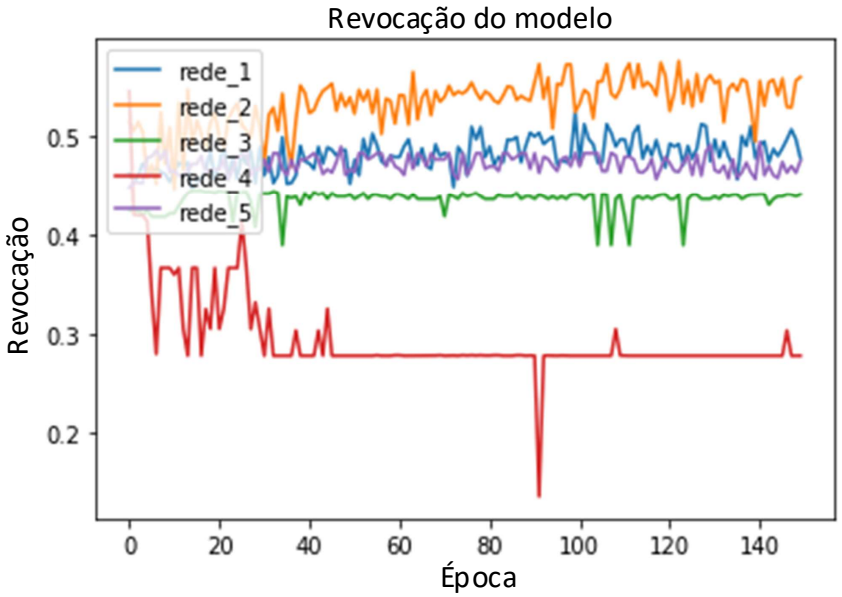
Nos resultados observamos que para a configuração de *linha de base* a *rede*_2, consegue acurácia de $0,996$. Para a configuração da *proposta* a *rede*_2, consegue acurácia de $0,998$, com um ganho de 2% na acurácia. O ganho da configuração da *proposta* em relação a *linha de base* se deu ao fato de encontrar uma ótima solução com o menor número de épocas. Os resultados formam dentro do esperado, como a base é desbalanceada, se o modelo respondesse como resultado apenas como correspondência falsa, a acurácia seria em torno de 98%, fazendo-se necessário a análise das métricas precisão, revocação e medida-F para uma avaliação acurada dos resultados encontrados.

As Figuras 17 e 18 apresentam as evoluções da métrica revocação durante as 150 épocas de treinamento para as cinco MLP criadas *rede*_1, *rede*_2, *rede*_3, *rede*_4 e *rede*_5, utilizando na Figura 17 as configurações do primeiro experimento *linha de base* e na Figura 18 as configurações do segundo experimento *proposta*.

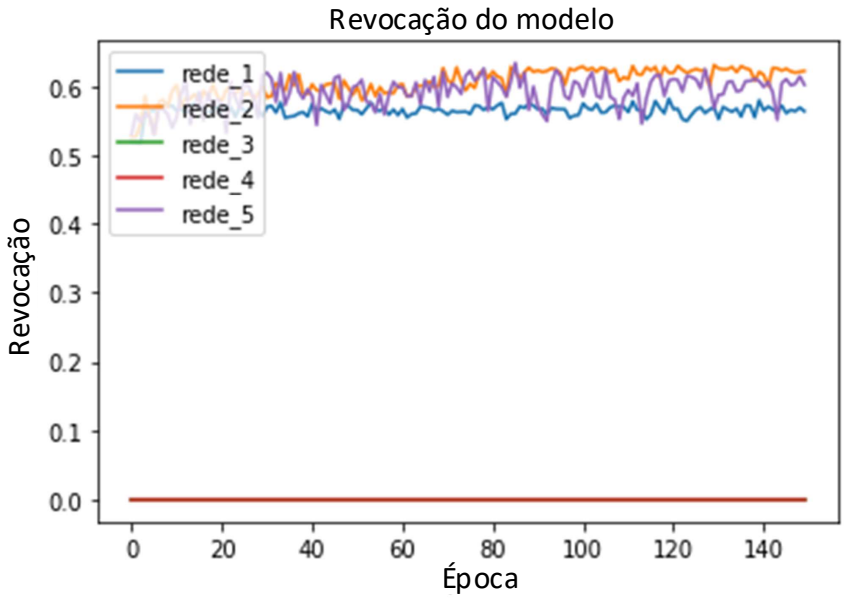
Nos resultados observamos que para a configuração de *linha de base* a *rede*_2, consegue revocação de $0,56$. Para a configuração da *proposta* a *rede*_2, consegue revocação de $0,61$, com um ganho de $0,05$ na revocação. A baixa revocação se deve ao fato do desbalanceamento da base, nesse cenário de desbalanceamento a configuração da *proposta*



Figuras 16 – Acurácia nos experimentos com as configurações da *proposta*.



Figuras 17 – Revocação nos experimentos com as configurações da *linha de base*.



Figuras 18 – Revocação nos experimentos com as configurações da *proposta*.

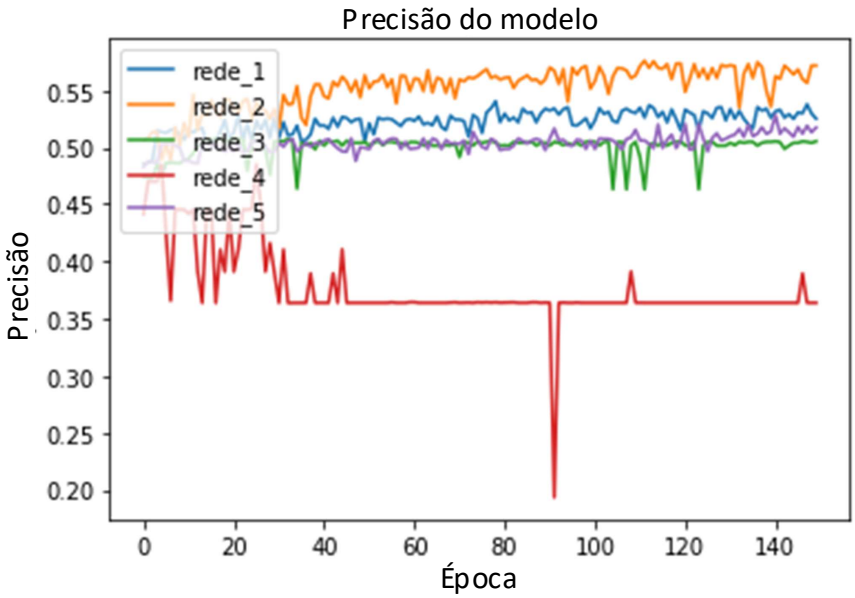
apresentou um ganho de 9%. Esses resultados indicam que 61% das correspondências existentes seriam descobertas de forma automática.

As Figuras 19 e 20 apresentam as evoluções da métrica precisão durante as 150 épocas de treinamento para as cinco MLP criadas *rede_1*, *rede_2*, *rede_3*, *rede_4* e *rede_5*, utilizando na Figura 19 as configurações do primeiro experimento *linha de base* e na Figura 20 as configurações do segundo experimento denominado *proposta*.

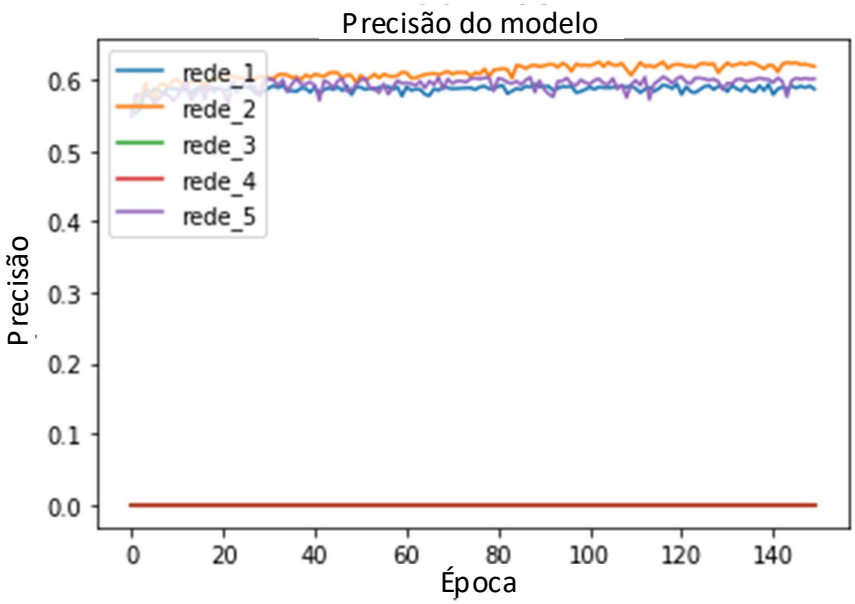
Nos resultados observa-se que para a configuração de *linha de base* a *rede_2*, consegue precisão de $0,57$. Para a configuração da *proposta* a *rede_2*, consegue precisão de $0,61$, com um ganho de $0,04$ na precisão. O que representa um ganho de 7%, considerando-se este um ganho importante para tarefas complexas como a de integração de dados.

As Figuras 21 e 22 apresentam as evoluções da métrica medida-f durante as 150 épocas de treinamento, para as cinco MLP criadas *rede_1*, *rede_2*, *rede_3*, *rede_4* e *rede_5*, utilizando na Figura 21 as configurações do primeiro experimento, *linha de base* e na Figura 22 as configurações do segundo experimento, *proposta*.

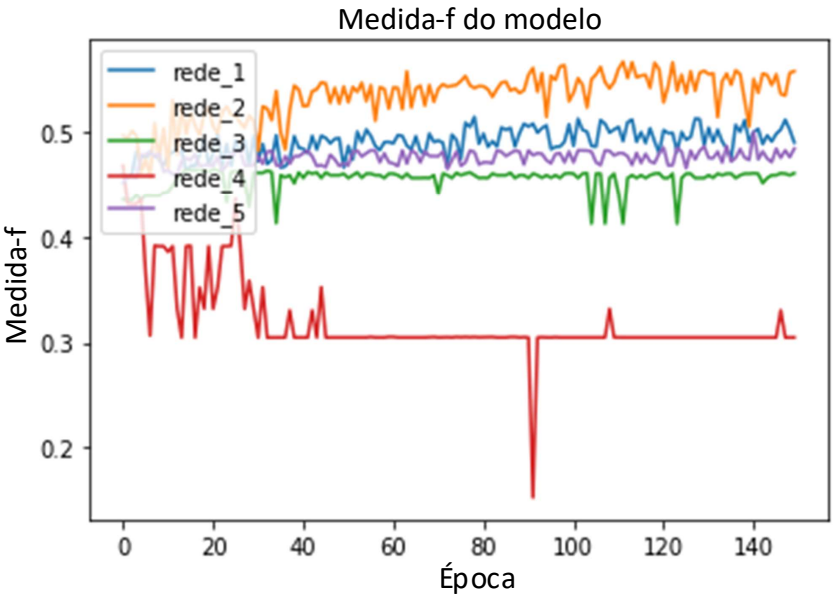
Observa-se nos resultados que para a configuração de *linha de base* a *rede_2*, consegue medida-f de $0,57$. Para a configuração denominado *proposta* a *rede_2*, consegue medida-f de $0,61$, com um ganho de $0,04$ na medida-f. O que representa um ganho de 7%, podendo considerá-lo um ganho importante para tarefas complexas como a de integração de dados.



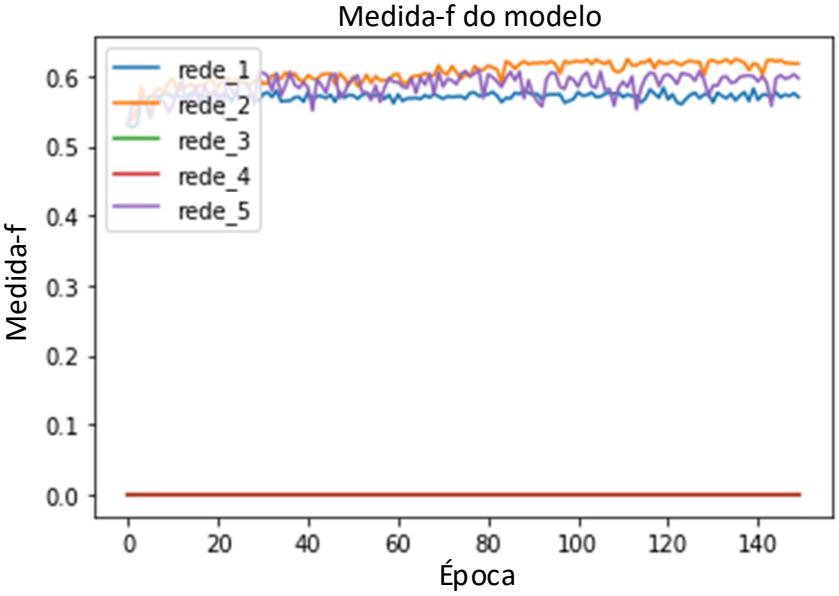
Figuras 19 – Precisão nos experimentos com as configurações da *linha de base*.



Figuras 20 – Precisão nos experimentos com as configurações denominada *proposta*.



Figuras 21 – Medida-F dos experimentos com as configurações da *linha de base*.



Figuras 22 – Medida-F nos experimentos com as configurações da *proposta*.

Analisando os resultados, o que se destaca é o baixo valor de medida-f, tanto para a configuração de *linha de base* como para a configuração da *proposta* em comparação com os resultados apresentados pelos autores Zhang et al. (2014), utilizados na configuração de *linha de base*, em que apresenta medida-f de 0.92 em seu artigo, a expectativa era que os resultados estivessem semelhantes. Uma das explicações para esse fato, está no desbalanceamento da base de dados utilizada. Os autores não citam a taxa de correspondências verdadeiras e falsas existentes na base de dados utilizada por eles.

Um dos objetivos do experimento foi analisar o desempenho de técnicas de aprendizado de máquina, com diferentes configurações, utilizando para isso uma extrapolação na MLP apresentada em Zhang et al. (2014) e o uso de diferentes *matchers*. Em nossos resultados, pode-se verificar que a utilização de um maior número de *matchers* apresentou resultados cerca de 7% melhores em comparação com a utilização de um menor número de *matchers*. A *rede_2* apresentou os melhores resultados em comparação com as outras redes criadas, mostrando que o número de neurônios por camada e o número de camadas utilizadas ajusta-se de acordo com o cenário em estudo.

6 Conclusão

O objetivo do estudo foi analisar a aplicação de técnicas de aprendizado de máquina para a realização da tarefa de integração de dados. Além disso queríamos esclarecer o assunto de integração de dados e como a utilização de técnicas de aprendizado de máquina podem auxiliar nesse processo. Assim, a partir deste conhecimento gerado, será possível traçar estratégias para integrar dados relacionais de fontes desconhecidas de forma rápida e assertiva. Este estudo mostrou que as técnicas de aprendizado de máquina apresentam resultados satisfatórios para a tarefa de integração de dados.

Para atingir o objetivo específico 1 "Estabelecer uma linha de base para as técnicas de integração de dados."utilizamos o trabalho do autor Zhang et al. (2014), que apresentou resultados com medida-f $0,92$. Para atingir o objetivo específico 2 "Identificar as técnicas de aprendizado de máquina."realizamos uma extrapolação, criando cinco MLP's distintas, com base na MLP criada pelo trabalho Zhang et al. (2014), para a realização da tarefa de integração de dados. Para atingir o objetivo específico 3 "Comparar as técnicas de aprendizado de máquina no processo de integração de dados."realizamos o experimento com duas configurações e cinco MLP's, comparando os resultados obtidos.

Após a execução das técnicas com nossa base de dados de hospitais, analisando os resultados, podemos verificar que a versão do modelo criado com os melhores resultados, possui as camadas de entrada e saída e três camadas escondidas, utiliza como entrada o **Lin Matcher**, o **Wup matcher**, o **Resnik matcher**, o **Schutze matcher**, o **Zhu Matcher**, o **Tata Matcher**, o **Li Matcher**, o indicador do tipo da primeira e da segunda coluna em questão e o número de palavras e letras da primeira e da segunda coluna em questão. Apresentando resultados de acurácia de $0,998\%$ e medida-f de $0,61\%$. Com isso verifica-se que aparentemente quanto maior o número de *matchers* utilizados melhores serão os resultados, mas como a melhora dos resultados não foram tão expressivas, pode existir um número ideal de *matchers* a ser utilizado de acordo com o contexto.

Durante nosso estudo, identificamos uma limitação para esse tipo de abordagem. A criação de um banco de dados marcado para o treinamento da RNA é uma tarefa complexa, uma vez que é necessário um número significativo de registros previamente marcados para a realização de tarefas em futuras correspondências. Para o treinamento das MLP's utilizados nesse trabalho a base de treinamento possui o mesmo contexto da base selecionada para teste, no caso da utilização de uma MLP pré-treinada os resultados podem ser discrepantes.

Como trabalhos futuros sugere-se o estudo de RNA's previamente treinadas em contextos diferentes dos testes para avaliação da pertinência de sua utilização, o que po-

deria diminuir significativamente o esforço para o treinamento das RNA. E o estudo do tratamento prévio da base de dados para o treino do modelo, em caso de base desbalanceada.

Referências

Addo, P. M., Guegan, D., and Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Alwan, A. A., Nordin, A., Alzeber, M., and Abualkishik, A. Z. (2017). A survey of schema matching research using database schemas and instances. *International Journal of Advanced Computer Science and Applications*, 8(10).

Ardjani, F., Bouchiha, D., and Malki, M. (2015). Ontology-alignment techniques: survey and analysis. *International Journal of Modern Education and Computer Science*, 7(11):67.

Aumueller, D., Do, H.-H., Massmann, S., and Rahm, E. (2005). Schema and ontology matching with coma++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 906–908.

Barreto, J. M. (2002). Introdução as redes neurais artificiais. *V Escola Regional de Informática. Sociedade Brasileira de Computaçao, Regional Sul, Santa Maria, Florianópolis, Maringá*, pages 5–10.

Batista, M. d. C. M. (2003). Otimização de acesso em um sistema de integração de dados através do uso de caching e materialização de dados. Master's thesis, Universidade Federal de Pernambuco.

Becker, J. L. (2015). *Estatística básica: transformando dados em informação*. Bookman editora.

Bittencourt, V. G. (2005). Aplicação de técnicas de aprendizado de máquina no reconhecimento de classes estruturais de proteínas. Master's thesis, Universidade Federal do Rio Grande do Norte.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. (1988). Autoclass: A bayesian classification system. In *Machine learning proceedings 1988*, pages 54–64. Elsevier.

Chernick, M. R., González-Manteiga, W., Crujeiras, R. M., and Barrios, E. B. (2011). Bootstrap methods.

Conrad, S., Höding, M., Saake, G., Schmitt, I., and Türker, C. (1997). Schema integration with integrity constraints. In *British National Conference on Databases*, pages 200–214. Springer.

de Alvarenga Júnior, W. J. (2018). *Métodos de otimização hiperparamétrica: um estudo comparativo utilizando árvores de decisão e florestas aleatórias na classificação binária*. PhD thesis, Universidade Federal de Minas Gerais.

De Souto, M., Lorena, A., Delbem, A., and de Carvalho, A. (2003). Técnicas de aprendizado de máquina para problemas de biologia molecular. *Sociedade Brasileira de Computaçao*.

Do, H.-H. and Rahm, E. (2002). Coma: a system for flexible combination of schema matching approaches. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 610–621. VLDB Endowment.

Doan, A., Halevy, A., and Ives, Z. (2012). *Principles of data integration*. Elsevier.

dos Santos, C. N. (2005). *Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro*. PhD thesis, Instituto Militar de Engenharia.

Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.

Falci, D. H. M., Soares, M. A. C., Brandão, W. C., and Parreiras, F. S. (2019). Using recurrent neural networks for semantic role labeling in portuguese. In *EPIA Conference on Artificial Intelligence*, pages 682–694. Springer.

for Medicare, C., Services, M., et al. (2016). Hospital compare downloadable database data dictionary.

Friedl, M. A. and Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, 61(3):399–409.

Gal, A. (2006). Why is schema matching tough and what can we do about it? *ACM Sigmod Record*, 35(4):2–5.

Garcia, S. C. (2003). *O uso de árvores de decisão na descoberta de conhecimento na área da saúde*. PhD thesis, Universidade Federal do Rio Grande do Sul.

Gardner, M. W. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

Gil, A. C. (2008). *Métodos e técnicas de pesquisa social*. 6. ed. Ediitora Atlas SA.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*. MIT press Cambridge.

Goodman, L. A. (1961). Snowball sampling. *The annals of mathematical statistics*, pages 148–170.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Holland, J. H. (1986). The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. *Machine learning, an artificial intelligence approach*, 2:593–623.

Horst, P. S. (1999). *Avaliaçao do conhecimento adquirido por algoritmos de aprendizado de máquina utilizando exemplos*. PhD thesis, Universidade de São Paulo.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., Lauer, M. S., et al. (2008). Random survival forests. *The annals of applied statistics*, 2(3):841–860.

Iyoda, E. M. et al. (2000). *Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas*. PhD thesis, unicamp.

Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31.

Khoudja, M. A., Fareh, M., and Bouarfa, H. (2018). A new supervised learning based ontology matching approach using neural networks. In *International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning*, pages 542–551. Springer.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Krishna, K. and Murty, M. N. (1999). Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Ledesma, R. (2008). Software de análisis de correspondencias múltiples: una revisión comparativa. *Metodología de encuestas*, 10(1):59–75.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246.

Li, Y., Bandar, Z. A., and McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on knowledge and data engineering*, 15(4):871–882.

Li, Y., Liu, D.-B., and Zhang, W.-M. (2005). Schema matching using neural network. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 743–746. IEEE.

Lin, D. et al. (1998). An information-theoretic definition of similarity. In *Icml*, pages 296–304.

Lipschutz, S. and Lipson, M. (2009). *Algebra Linear: Coleção Schaum*. Bookman Editora.

Madhavan, J., Bernstein, P. A., and Rahm, E. (2001). Generic schema matching with cupid. In *vldb*, volume 1, pages 49–58.

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

Massmann, S., Raunich, S., Aumüller, D., Arnold, P., and Rahm, E. (2011). Evolution of the coma match system. In *Proceedings of the 6th International Conference on Ontology Matching-Volume 814*, pages 49–60. CEUR-WS. org.

Michie, D., Spiegelhalter, D. J., Taylor, C., et al. (1994). Machine learning. *Neural and Statistical Classification*, 13(1994):1–298.

Monard, M. C. and Baranauskas, J. A. (2003a). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.

Monard, M. C. and Baranauskas, J. A. (2003b). Indução de regras e árvores de decisão. *Sistemas Inteligentes-Fundamentos e Aplicações*, 1:115–139.

Mukkala, L., Arvo, J., Lehtonen, T., Knuutila, T., et al. (2015). *Current state of ontology matching. A survey of ontology and schema matching*. PhD thesis, University of Turku, Technology Research Center.

Munroe, K. D. and Papakonstantiou, Y. (2000). Bbq: A visual interface for integrated browsing and querying of xml. In *Working Conference on Visual Database Systems*, pages 277–296. Springer.

Noriega, L. (2005). Multilayer perceptron tutorial. *School of Computing. Staffordshire University*.

Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971.

Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350.

Rauber, T. W. (2005). Redes neurais artificiais. *Universidade Federal do Espírito Santo*, page 29.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Rodrigues, D. d. A. et al. (2013). *Casamento de esquemas de banco de dados aplicando aprendizado ativo*. PhD thesis, Universidade Federal do Amazonas.

Rodrigues, D. d. A. et al. (2018). *A Study on Machine Learning Techniques for the Schema Matching Networks Problem*. PhD thesis, Universidade Federal do Amazonas.

Rubin, D. B. (1981). The bayesian bootstrap. *The annals of statistics*, pages 130–134.

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. (1998). Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Scopim, K. d. S. (2003). *J-schemas integrator*. PhD thesis, Universidade Federal do Paraná.

Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. In *Journal on data semantics IV*, pages 146–171. Springer.

Shvaiko, P. and Euzenat, J. (2011). Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176.

Silva, I. and Campos, F. (2015). New perspectives using big data: a study of bibliometric 2000-2012. In *Anais da 11a Conferência Internacional sobre Sistemas de Informação e Gestão de Tecnologia, São Paulo, SP*.

Silva, L. d. R. et al. (2017). *Um estudo sobre o uso de informações de instâncias para o casamento de esquemas no domínio de comércio eletrônico*. PhD thesis, Universidade Federal do Amazonas.

Silveira, D. T. and Córdova, F. P. (2009). Unidade 2–a pesquisa científica. *Métodos de pesquisa*, 1.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tagarelli, A. (2011). *XML Data Mining: Models, Methods, and Applications: Models, Methods, and Applications*. IGI Global.

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

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. "O'Reilly Media, Inc.".

Vargas, M. R., De Lima, B. S., and Evsukoff, A. G. (2017). Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 60–65. IEEE.

Wick, M. L., Rohanimanesh, K., Schultz, K., and McCallum, A. (2008). A unified approach for schema matching, coreference and canonicalization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 722–730.

Widom, J. (1995). Special issue on materialized views and data warehousing. *IEEE Bulletin on Data Engineering*, 18(2).

Wong, T. Y. and Bressler, N. M. (2016). Artificial intelligence with deep learning technology looks into diabetic retinopathy screening. *Jama*, 316(22):2366–2367.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

Zhang, J., Li, J., Wang, S., and Bian, J. (2014). A neural network based schema matching method for web service matching. In *2014 IEEE International Conference on Services Computing*, pages 448–455. IEEE.

Zhao, X., Shi, X., and Zhang, S. (2015). Facial expression recognition via deep learning. *IETE technical review*, 32(5):347–355.

Zhu, G. and Iglesias, C. A. (2016). Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):72–85.