

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CAMPUS MONTE CARMELO
Bacharel em Sistemas de Informações

Deep Learning para Avaliação de Risco de Crédito Financeiro

Janayna Moura Fernandes

Monte Carmelo - MG

2019

Janayna Moura Fernandes

Deep Learning para Avaliação de Risco de Crédito Financeiro

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Uberlândia (UFU) *Campus* Monte Carmelo como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informações.

Universidade Federal de Uberlândia - UFU

Campus Monte Carmelo

Orientador: Prof. Dr. Murillo Guimarães Carneiro

Monte Carmelo - MG

15 de julho de 2019

Janayna Moura Fernandes

Deep Learning para Avaliação de Risco de Crédito Financeiro/ Janayna Moura
Fernandes

. – Monte Carmelo - MG, 15 de julho de 2019-
56 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Murillo Guimarães Carneiro

Trabalho de Conclusão de Curso (TCC) – Universidade Federal de Uberlândia -
UFU

Campus Monte Carmelo, 15 de julho de 2019.

1. Mineração de Dados. 2. Crédito. I. Orientador Prof. Dr. Murillo G. Carneiro.
II. Universidade Federal de Uberlândia. III. Faculdade de Computação. IV. Título:
Aprendizado de Máquina na Avaliação de Risco de Crédito Financeiro

Janayna Moura Fernandes

Deep Learning para Avaliação de Risco de Crédito Financeiro

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Uberlândia (UFU) *Campus* Monte Carmelo como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informações.

Versão original.

Trabalho aprovado em 11 de Julho de 2019, Monte Carmelo - MG:

Prof. Dr. Murillo Guimarães Carneiro
Orientador

Professora Dra. Fernanda Maria da Cunha Santos
Convidada

Professor Dr. Rodrigo Sanches Miani
Convidado

Monte Carmelo - MG
15 de julho de 2019

Resumo

Bancos e organizações financeiras que lidam com o empréstimo de crédito têm como chave para o sucesso os modelos de avaliação de crédito para classificar seus clientes em perspectiva como prováveis adimplentes ou inadimplentes. A avaliação do risco de crédito é frequentemente avaliada por um gerente a partir de sua experiência intuitiva. No entanto, é possível tomar essas decisões substanciais com mais precisão através do suporte de modelos desenvolvidos a partir do aprendizado de máquina e da mineração de dados, usando fundamentalmente a sua tarefa de classificação. Diversas técnicas de mineração de dados, como árvores de decisão, florestas aleatórias, máquinas de vetores de suporte, redes neurais, já foram usadas para classificar os clientes em perspectiva. O objetivo neste trabalho é a avaliação de algoritmos classificatórios consolidadas na literatura e a sugestão da construção de um modelo de redes neurais profundas aplicados para o problema de avaliação de risco de crédito. Os experimentos foram conduzidos em três conjuntos de dados disponíveis publicamente no UCI Machine Learning Repository. Foi comparada a acurácia média das técnicas. Os resultados obtidos mostram que o classificador florestas aleatórias se sobressaiu às demais técnicas por apresentar a maior acurácia média e por ser a mais estável para as três bases de dados estudadas. A técnica de aprendizado profundo apresentou um bom resultado mas não obteve resultado superior a da mencionada anteriormente.

Palavras-chaves: Análise de Dados, Mineração de dados, Crédito, Análise de risco de crédito, Sistemas de suporte de decisão, Deep Learning, Deep Neural Networks.

Abstract

Banks and financial organizations are dealing with credit lending as the key to success. The credit assessment models rank their prospective clients as likely defaulters or not. Credit risk assessment is often evaluated by a manager from his or her intuitive experience. However, it is possible to make these substantial decisions more accurately through the support of the models developed from machine learning and data mining, using essentially the classification task. Several data mining techniques, such as decision trees, random forests, support vector machines, neural networks, have already been used to classify prospective customers. The objective of this work is the evaluation of consolidated classificatory algorithms in the literature and the suggestion of the construction of a deep neural networks model applied to the problem of credit risk assessment. The experiments were conducted on three publicly available data sets in the UCI Machine Learning Repository. The average accuracy of the techniques was compared. The results obtained show that the random forest classifier excels to the other techniques because it presents the highest average accuracy and is the most stable for the three databases studied. The technique of deep learning presented a good result but did not obtain a result superior to the one mentioned previously.

Keywords: Data Analysis, Data Mining, Credit, Credit Risk Analysis, Decision Support Systems, Deep Learning, Deep Neural Networks.

Lista de ilustrações

Figura 1 – As etapas do KDD definidas por Fayyad (1996)	20
Figura 2 – O processo de descoberta de conhecimento em bancos de dados (KDD).	20
Figura 3 – Relação entre dados, informação e conhecimento.	22
Figura 4 – Exemplo de uma Árvore de Decisão	25
Figura 5 – Transformação da Máquina de Vetores de Suporte	26
Figura 6 – Exemplo de uma Rede Neural Artificial	27
Figura 7 – Representação de neurônios se comunicando através da sinapse	30
Figura 8 – Uma RNR simples. Representada em ciclo (esquerda) e como desenrolada (direita) em t passos de tempo.	32
Figura 9 – Uma RNC em três dimensões. Adaptado pela autora a partir de (DOSHI, 2018)	33
Figura 10 – Ilustração da Validação Cruzada Aninhada	44

Lista de tabelas

Tabela 1 – Etapas do KDD.	19
Tabela 2 – Dados, informação e conhecimento em espaço computacional.	22
Tabela 3 – Matriz de confusão para um problema de duas classes	29
Tabela 4 – Atributos no conjunto de dados Australiano	41
Tabela 5 – Atributos no conjunto de dados Alemão	41
Tabela 6 – Atributos no conjunto de dados Taiwanês	42
Tabela 7 – Breve descrição dos conjuntos de dados da análise de risco de crédito em termos do número de itens de dados (<i>#Inst.</i>), número de atributos (<i>#Attr.</i>) e número de classes (<i>#Classes</i>).	46
Tabela 8 – Resultados obtidos com diferentes parâmetros da DNN em termos de acurácia média(AM) e desvio padrão (DP) para os três conjuntos de dados em estudo	47
Tabela 9 – Resultados dos classificadores em termos de acurácia média (AM) e desvio padrão (DP) para os três conjuntos de dados em estudo.	47

Lista de abreviaturas e siglas

AD	Classificador árvores de decisão
AG	Algoritmos genéticos
AM	Aprendizado de máquina
AUC	Área sob a curva
DBN	Deep Belief Networks
DL	Aprendizado Profundo
DNN	Redes Neurais Profundas
FA	Classificador florestas aleatórias
IA	Inteligência artificial
KDD	Descoberta de conhecimento em bases de dados
<i>k</i>-NN	Classificador <i>k</i> vizinhos mais próximos
MD	Mineração de dados
NB	Classificador ingênuo de Bayes
RL	Classificador regressão logística
RNA	Redes neurais artificiais
RNC	Redes Neurais Convolucionais
RNR	Redes Neurais Recorrentes
SVM	Classificador máquina de vetores de suporte
VC	Validação cruzada

Sumário

1	INTRODUÇÃO	11
1.1	Justificativas	11
1.2	Objetivos	12
1.3	Hipótese	13
1.4	Organização do documento	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Aprendizado de Máquina	14
2.2	Descoberta de Conhecimento em Bases de Dados (KDD)	17
2.2.1	Dados, Informação e Conhecimento	21
2.2.2	Mineração de Dados	22
2.2.3	Classificação de Dados	23
2.2.3.1	Árvores de Decisão	24
2.2.3.2	Florestas Aleatórias	25
2.2.3.3	Máquina de Vetores de Suporte	25
2.2.3.4	Redes Neurais Artificiais	26
2.2.3.5	Regressão Logística	27
2.2.4	Validação e Medidas de Desempenho de um Classificador	27
2.3	Aprendizado Profundo	29
2.3.1	Redes Neurais Profundas (DNN)	31
2.3.2	Deep Belief Networks (DBN)	31
2.3.3	Redes Neurais Recorrentes (RNR)	32
2.3.4	Redes Neurais Convolucionais (RNC)	33
3	TRABALHOS RELACIONADOS	35
3.1	Aprendizado de máquina para problemas de Finanças	35
3.2	Deep learning para problemas de Finanças	38
4	DESENVOLVIMENTO	40
4.1	Conjuntos de Dados para Avaliação	40
4.1.1	Dados de Crédito Australiano	40
4.1.2	Dados de Crédito Alemão	40
4.1.3	Dados de Crédito Taiwanês	42
4.2	Configuração da DNN	42
4.2.1	Batch size	42
4.2.2	Número de épocas	43

4.2.3	Taxa de aprendizado e Otimizador	43
4.2.4	Função de Ativação	43
4.2.5	Modo de inicialização	43
4.3	Ambiente Computacional	44
5	RESULTADOS E DISCUSSÃO	46
5.1	Apresentação dos dados coletados	46
5.2	Seleção de parâmetros da DNN	46
5.3	Análise dos dados	47
6	CONCLUSÕES	49
	REFERÊNCIAS	51

1 Introdução

Um grande número de instituições tem percebido a importância de suas bases de dados (WEI et al., 2015). Essas bases de dados, que geralmente cobrem transações realizadas ao longo de vários anos, podem levar a uma melhor compreensão do perfil de seus clientes e, portanto, vir a corroborar com a qualidade na oferta de novos produtos ou serviços. Nos ambientes de decisão financeira não é diferente, já que são utilizadas quantidades substanciais de informações oriundas das mais variadas fontes em um mercado em constante mudança (SILVA, 2008).

O aumento do volume de dados armazenado e gerenciado pelas empresas tem se elevado de tal forma que o tratamento dessas grandes bases de dados supera a habilidade humana de entender e de eficientemente lidar com esses dados. O que leva a uma necessidade crítica de ferramentas e técnicas que sejam capazes de realizar automaticamente análises eficazes e eficientes dos dados para apoiar empresas e indivíduos no planejamento estratégico (ZHANG; ZHOU, 2004). O processo geral que possui este fim é geralmente referenciado na literatura como Knowledge Discovery in Databases (KDD): Descoberta de Conhecimento em Bases de Dados.

As razões para as dificuldades da exploração e análise dos dados armazenados são o grande volume dos dados a serem examinados e a natureza dos relacionamentos em si que não são óbvios. Uma ferramenta de descoberta de conhecimento é necessária para auxiliar o tomador de decisão em relação aos pedidos de empréstimo. O processo KDD fornece uma variedade de métodos úteis para descobrir tais relacionamentos em dados históricos, enquanto garante que os relacionamentos descobertos possam ser generalizados para os dados novos (futuros) (TAN et al., 2009). Ele pode ser descrito em três grandes fases: o pré-processamento dos dados, a mineração de dados (MD) e o pós-processamento de dados.

1.1 Justificativas

O KDD pode ajudar a extrair relações dos dados históricos e, assim, melhores decisões podem ser tomadas. Para a realização da fase de mineração de dados técnicas de aprendizado de máquina (AM) podem ser utilizadas.

Neste contexto, como um ramo do AM, são as técnicas de aprendizado profundo (DL). As DL têm se mostrado eficiente em diversos domínios (visão computacional, reconhecimento e geração de voz, processamento de linguagem natural) (LECUN et al., 2015), impulsionadas pelo aumento das capacidades de computação modernas, pela ampla

disponibilidade de dados e pelos avanços nas estruturas de codificação. As redes neurais profundas (DNN) são agora onipresentes (ALBERICIO et al., 2016).

Por este ângulo, para garantir o sucesso de instituições comerciais e bancárias faz-se necessária uma avaliação automática, usando técnicas rápidas e adaptativas como DL, onde a probabilidade de inadimplência pode ser calculada a partir de um conjunto de dados históricos em um período de tempo razoável.

O objetivo deste trabalho é usar dados históricos de pedidos de empréstimo de crédito para prever se um candidato poderá ou não pagar a dívida contraída. A avaliação de risco de crédito é basicamente um problema de classificação e de previsão (KRUPPA et al., 2013) que estuda o rendimento e a sua segurança. A decisão de oferecer ou não crédito a um cliente deve ser bastante criteriosa. Dar crédito ao cliente em perspectiva é determinado pelas suas características relacionadas, como idade, renda, solvência e escolaridade e de características do crédito como tipo de crédito, vencimento, valor do empréstimo e outras características inerentes às operações financeiras (CHEN; HUANG, 2003).

A finalidade dos modelos de avaliação de risco de crédito é classificar os clientes como adimplentes (aceitar) ou como inadimplentes (rejeitar) (SILVA, 2008; HAND; HENLEY, 1997). O conhecimento obtido através do KDD, no final, pode ser usado pelos gerentes de crédito para ajudá-los a rejeitar ou aceitar candidatos. Ao utilizar destas técnicas, haveria menos risco para as empresas financeiras ao prever quais clientes serão bem-sucedidos em seus pagamentos. Consequentemente, mais pessoas poderiam ter acesso a empréstimos de crédito.

Na literatura existem muitas técnicas de classificação, como árvores de decisão (AD), máquina de vetores de suporte (SVM), redes neurais artificiais (RNA), florestas aleatórias (FA), regressão Logística (RL) e outras (TAN et al., 2009). O enfoque desta pesquisa é aplicar um conjunto consistente de técnicas de acordo com a literatura e também investigar a aplicação de DNN no contexto da análise de crédito, o que representa uma contribuição para a literatura.

1.2 Objetivos

O presente trabalho tem como objetivo principal o de empregar um conjunto de técnicas de classificação para problemas de avaliação de risco de crédito, em especial as redes neurais profundas. Mais especificamente:

- Preparar os dados de três bases de dados referentes ao histórico de tomadores de crédito disponíveis publicamente no repositório de dados UCI Machine Learning;

- Selecionar e aplicar um conjunto de técnicas de classificação para as bases de dados consideradas;
- Desenvolver uma rede neural profunda para o problema de análise de crédito;
- Codificar um ambiente computacional para a análise de desempenho de todas as técnicas sob estudo.

1.3 Hipótese

A hipótese investigada é se o método de DNN pode ser eficiente para avaliar o risco de inadimplência de tomadores de crédito, quando comparada com técnicas do estado-da-arte para o problema.

1.4 Organização do documento

O restante deste trabalho está organizado em outros cinco capítulos:

- Capítulo 2 - Este capítulo é composto pela revisão sistemática bibliográfica, toda fundamentação teórica necessária ao entendimento do trabalho: A inteligência artificial, o processo do KDD e mais detalhadamente as técnicas AM e DL.
- Capítulo 3 - Esta seção é composta pelos trabalhos relacionados.
- Capítulo 4 - Este capítulo é composto pela apresentação dos conjuntos de dados utilizados, as configurações das técnicas de mineração de dados e dos métodos para a validação dos resultados obtidos.
- Capítulo 5 - Nesta parte do trabalho são apresentados os resultados obtidos até o fechamento desta pesquisa.
- Capítulo 6 - Este capítulo é composto pelas conclusões que foram possíveis deduzir a respeito do assunto e é composta também pelos trabalhos futuros.

2 Fundamentação Teórica

Com os avanços na tecnologia de computadores, temos a capacidade de armazenar e processar grandes quantidades de dados, bem como acessá-los de locais fisicamente distantes em uma rede de computadores. Mas os dados armazenados apenas se tornam úteis quando analisados e transformados em informações que podemos usar, por exemplo, para tomadas de decisões (ALPAYDIN, 2014).

2.1 Aprendizado de Máquina

Aprendizagem é o processo de converter experiência em perícia ou conhecimento (SHALEV-SHWARTZ; BEN-DAVID, 2014). Os algoritmos de aprendizado de máquina usam das capacidades de processamento dos computadores para generalizar conhecimento automaticamente a partir de dados armazenados (SHALEV-SHWARTZ; BEN-DAVID, 2014). A entrada para um algoritmo de aprendizado é o conjunto de dados, que representa a experiência, a saída é alguma opinião, o modelo aprendido ou deduzido (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Segundo Alpaydin (2014) podemos não saber os detalhes do processo subjacente à geração de dados, por exemplo, o comportamento do consumidor, mas sabemos que isso não é completamente aleatório. As pessoas não vão a supermercados e compram coisas ao acaso. Quando compram manteiga, compram pães; eles compram sorvete no verão e casacos no inverno. Existem certos padrões nos dados.

Tais padrões podem ajudar a entender o processo, ou pode-se usar esses padrões para fazer previsões, supondo que o futuro, pelo menos no futuro próximo, não seja muito diferente do passado quando os dados da amostra foram coletados, então pode-se esperar que as previsões futuras estejam certas (ALPAYDIN, 2014). O modelo pode ser preditivo para fazer previsões no futuro (prever resultados), ou descritivo para compreender a essência dos padrões encontrados nos dados, ou ambos (HORNICK et al., 2010).

O componente da experiência, ou treinamento, em aprendizado de máquina geralmente se refere a dados de uma amostra aleatória (SHALEV-SHWARTZ; BEN-DAVID, 2014). Essa descrição do aprendizado de máquina destaca sua estreita relação com as estatísticas. De fato, há muito em comum entre as duas disciplinas, em termos de metas e técnicas utilizadas. Se um médico apresentar a hipótese de que há uma correlação entre tabagismo e doença cardíaca, é papel do estatístico visualizar amostras de pacientes e verificar a validade dessa hipótese (teste de hipóteses). Em contraste, o aprendizado de máquina tem como objetivo usar os dados coletados de amostras de pacientes para

obter uma descrição das causas das doenças cardíacas. A esperança é que as técnicas automatizadas possam ser capazes de descobrir padrões significativos (ou hipóteses) que possam ter sido perdidos pelo observador humano (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Na elaboração de algoritmos de AM normalmente utiliza-se de teorias da estatística e de modelos matemáticos porque a tarefa principal é inferir conhecimento a partir de uma amostra. Ainda, é importante que os algoritmos sejam eficientes na fase da construção do modelo (fase de treinamento), bem como para armazenar e processar a enorme quantidade de dados que geralmente tem-se (ALPAYDIN, 2014). Em certas aplicações, sua complexidade de espaço e tempo, pode ser tão importante quanto sua precisão preditiva.

As áreas de aplicação de AM são abundantes: os bancos financeiros analisam seus dados passados para criar modelos para uso em aplicações de crédito, detecção de fraudes e em mercado de ações. Na indústria, os modelos de aprendizado são usados para otimização, controle e solução de problemas. Na medicina, programas de aprendizado são usados para diagnóstico clínico. Nas telecomunicações, os padrões de chamadas são analisados para otimização da rede e maximização da qualidade e oferta de serviço. Na ciência, grandes quantidades de dados em física, astronomia e biologia só podem ser analisadas com rapidez suficiente pelos computadores. Em ciência computacional: reconhecimento de áudios e imagens, robótica, visão computacional, classificação de documentos (ALPAYDIN, 2014; LANGLEY; SIMON, 1995).

Para executar a tarefa de classificação, o algoritmo de aprendizado recebe um conjunto de exemplos de treinamento rotulados (com sua classificação correta) (BLUM et al., 2016). Esses dados de treinamento podem ser uma coleção de transações sobre concessão de crédito, que contém características da transação e do cliente, cada uma rotulada como “sim” (aprovado para a concessão de crédito) ou “não” (reprovado para a concessão de crédito), ou uma coleção de pacientes, cada qual rotulada por se responderam bem ou não ao tratamento médico fornecido. O algoritmo então pretende usar os exemplos de treinamento para produzir uma regra de classificação que funcionará adequadamente sobre novos dados, por exemplo, clientes futuros. Uma característica fundamental do AM e do KDD de forma geral é que seu objetivo é a generalização: usar um conjunto de dados armazenados para ter um bom desempenho em dados futuros (BLUM et al., 2016).

O aprendizado de máquina consiste em técnicas de aprendizado supervisionado, não-supervisionado, ou semi-supervisionado, tal como detalhado a seguir:

- Aprendizado de máquina supervisionado:

É fornecido ao algoritmo um conjunto de registros para treinamento. Cada registro é descrito por um vetor de características (atributos) e pelo seu rótulo de classe (variável

alvo/objetivo)(RUSSELL; NORVIG, 2016). Este rótulo de classe normalmente é especificado por um especialista no domínio, como no caso de análise de dados financeiros poderia ser o gerente do banco ou o analista de crédito.

O objetivo do algoritmo é, a partir de análise dos dados de treinamento que possuem a variável objetivo definidas em classes conhecidas, treinar-se um modelo e assim, posteriormente, ser aplicado para determinar corretamente a classe de outros dados de entrada do mesmo domínio. Isso exige que o algoritmo de aprendizado possa generalizar a partir dos dados de treinamento para situações desconhecidas com certa confiança (MUHAMMAD; YAN, 2015).

Para rótulos de classe discretos, essa tarefa é conhecida como classificação e para valores contínuos como regressão (MCCUE, 2014). Exemplos de técnicas usadas para tratar o aprendizado de máquina supervisionado são: RNA, SVM, AD, classificador ingênuo de Bayes (NB), FA e k-vizinhos mais próximos (k-NN) (KOTSIANTIS et al., 2007).

- Aprendizado de máquina não supervisionado:

Neste tipo de aprendizado não existe o rótulo da classe. São analisados os registros e o objetivo é tentar identificar os relacionamentos subjacentes existentes entre um conjunto de objetos caracterizados por um conjunto de atributos e ao encontrar as regularidades na entrada definir como eles podem ser reunidos de tal forma que eles gerem os chamados clusters ou agrupamentos com base em uma medida de similaridade entre os atributos (HASTIE et al., 2009b).

Existe uma estrutura para o espaço de entrada de modo que certos padrões ocorrem mais frequentemente do que outros. Nas estatísticas, isso é chamado de estimativa de densidade (HASTIE et al., 2009b). Após a determinação dos agrupamentos, geralmente, faz-se necessário uma investigação e estudo sobre cada agrupamento com o intuito de tentar entendê-lo e explicá-lo no ambiente do problema que está sendo analisado (HASTIE et al., 2009b).

O aprendizado não supervisionado oferece a possibilidade de explorar a estrutura de dados sem orientação na forma de informações de classe e pode frequentemente revelar características não esperadas ou conhecidas anteriormente (HATZIARGYRIOU; VOUMVOULAKIS, 2016).

Normalmente observa-se a utilização desse tipo de aprendizado quando algo impossibilita ou torna de alto custo a definição da variável objetivo (HASTIE et al., 2009b). São vistos como algoritmos descritivos, ou seja, descrevem de forma clara as propriedades dos dados encontrados. Tarefas não supervisionadas podem ser agrupamento, regras de associação, extração de características, redução de dimensionalidade e quantização vetorial (HASTIE et al., 2009b). Exemplos de técnicas usadas para tratar o aprendizado de má-

quina não-supervisionado são: k-means, agrupamento hierárquico aglomerativo e DBSCAN (KYAN et al., 2014).

- Aprendizado de máquina semi-supervisionado:

Por fim, o aprendizado de máquina semi-supervisionado, uma área mais recente, consiste em utilizar algoritmos que aprendem a partir de dados rotulados e, também, de não rotulados durante o treinamento. Os métodos de aprendizado semi-supervisionado usam dados não rotulados para modificar ou priorizar hipóteses obtidas apenas de dados rotulados (ZHU, 2005).

Segundo Sanches (2003, p. 50):

A grande motivação para esse tipo de aprendizado se dá pelo fato de exemplos não rotulados existirem em abundância e exemplos rotulados serem geralmente escassos. Além disso, a rotulação de exemplos pode ser custosa, como nos casos de indexação de vídeo, categorização de textos e diagnósticos médicos, entre outros. [...] A ideia do aprendizado semi-supervisionado é então utilizar os exemplos rotulados para se obter informações sobre o problema e utilizá-las para guiar o processo de aprendizado a partir dos exemplos não rotulados (Bruce, 2001). Aprendizado semi-supervisionado pode ser utilizado tanto em tarefas de classificação como em tarefas de agrupamento (Basu, Banerjee e Mooney, 2002).

2.2 Descoberta de Conhecimento em Bases de Dados (KDD)

O método tradicional de transformar dados em conhecimento depende da análise e interpretação manuais. Por exemplo, no setor financeiro, é comum que os especialistas analisem periodicamente as tendências atuais e as mudanças nos dados de crédito, digamos, mensalmente. Os especialistas, em seguida, fornecem um relatório detalhando a análise para a instituição financeira; este relatório torna-se a base para futuras tomadas de decisão e planejamento para a gestão de crédito. Seja ciência, marketing, finanças, saúde ou qualquer outro campo, a abordagem clássica da análise de dados depende fundamentalmente de um ou mais analistas que se familiarizem profundamente com os dados, usuários, produtos e ou serviços (GALINDO; TAMAYO, 2000).

Segundo Fayyad et al. (1996) essa forma de análise manual em um conjunto de dados é vagarosa, onerosa e bastante subjetiva. Na verdade, à medida que os volumes de dados crescem drasticamente, esse tipo de análise manual de dados tem se tornado completamente impraticável em muitos domínios. Diante disso, é fundamental o uso de técnicas computacionais que permitam a extração automática ou semi-automática de novos conhecimentos a partir de grandes bases de dados (FRAWLEY et al., 1992). Desta forma as empresas podem obter vantagem competitiva, aumentar a eficiência e fornecer serviços mais valiosos aos clientes. É, também, fundamental usar criteriosamente o tempo dos especialistas e minimizar o esforço geral do processo (BRODLEY et al., 2012).

KDD é uma área multidisciplinar que prosperou e continua a prosperar a partir da intercessão de campos de pesquisa, como aprendizado de máquina, reconhecimento de padrões, bancos de dados, estatísticas, inteligência artificial (IA), visualização de dados (FAYYAD et al., 1996). Em nível teórico, o KDD contempla o desenvolvimento de métodos e técnicas para entender os dados. E tem por objetivo o de encontrar conhecimento a partir de um conjunto de dados para ser utilizado em um processo decisório (LINOFF; BERRY, 2011). Portanto, um requisito importante é que esse conhecimento descoberto seja compreensível a humanos, além de útil e interessante aos usuários finais do processo (FRAWLEY et al., 1992). O KDD também enfatiza as propriedades de escala e robustez dos algoritmos de modelagem para grandes conjuntos de dados ruidosos (FAYYAD et al., 1996b). No cerne do processo está a aplicação de métodos específicos de mineração de dados para a descoberta e extração de padrões (FAYYAD et al., 1996).

De acordo com a literatura, o KDD se refere ao processo geral de descoberta de conhecimento útil a partir de dados, e a mineração de dados refere-se a uma etapa específica desse processo (FAYYAD et al., 1996; CIOS et al., 2007; KALAVATHY et al., 2007).

O termo processo indica que o KDD integra várias etapas, que abrangem a preparação de dados, a busca de padrões, a avaliação do conhecimento e o refinamento, todos repetidos em várias iterações (FAYYAD et al., 1996a; KALAVATHY et al., 2007). As etapas do processo de KDD que englobam a preparação de dados, seleção de dados, limpeza de dados, transformação dos dados e avaliação e interpretação dos resultados da mineração, são imprescindíveis para assegurar que conhecimento útil e válido seja derivado dos dados. Segundo Fayyad et al. (1996a): a aplicação direta de métodos de mineração de dados pode ser uma atividade precipitada, levando facilmente à descoberta de padrões sem sentido e inválidos, logo para a análise dos dados faz-se necessário o pré-processamento e pós-processamento dos dados.

O processo geral do KDD proposto por Fayyad (1996) consiste em nove etapas como é mostrado na Figura 1. Segundo Fayyad (1996) o processo do KDD pode envolver uma iteração significativa e pode conter repetições entre quaisquer duas etapas. Uma explicação sobre cada etapa do processo pode ser encontrada na Tabela 1.

Já em Olaru e Wehenkel (1999) esta divisão é composta por cinco etapas: seleção de dados, transformação de dados, mineração de dados, interpretação e validação de resultados e por fim incorporação do conhecimento descoberto.

E em Tan et al. (2009) considera-se a divisão do processo KDD em três grandes etapas (pré-processamento, mineração de dados e pós-processamento) como é mostrado na Figura 2, que também apresenta ferramentas que podem ser utilizadas nas fases de pré-processamento e pós-processamento.

Tabela 1 – Etapas do KDD.

Etapas	Explicação
Compreensão do domínio da aplicação	Desenvolver uma compreensão do domínio da aplicação, o conhecimento prévio relevante e os objetivos dos usuário finais (partes interessadas).
Seleção de dados	Selecionar um conjunto de dados ou focar em um subconjunto de variáveis (amostra).
Limpeza e pré-processamento de dados	Inclui operações como remoção de ruído e manuseio de dados ausentes. Os dados armazenados são muitas vezes errados, incompletos e inconsistentes, talvez devido a erros de operação ou falhas na implementação do sistema. Esses dados de baixa qualidade precisam ser limpos antes da mineração de dados.
Redução e projeção de dados	Encontrar recursos úteis para representar os dados (dependendo do objetivo da tarefa) e inclui, caso se faça necessário, o uso de técnicas de redução ou transformação de dimensionalidade.
Escolha da tarefa de mineração de dados	Decidir sobre o objetivo do KDD, por exemplo, sumarização, classificação, regressão, agrupamento e assim por diante.
Definição do(s) algoritmo(s) de mineração de dados	Selecionar método(s) a ser(em) usado(s) para pesquisar padrões em dados, como decidir qual modelo e parâmetros são os mais apropriados.
Mineração de dados	Essa etapa gera padrões em uma forma representacional específica, como regras de classificação, árvores de decisão, modelos de regressão, tendências etc.
Avaliação dos padrões minerados	Interpretar os padrões descobertos, bem como a possível visualização dos padrões e modelos extraídos. Possivelmente retornando a qualquer uma das etapas de 1 a 7 para mais iterações.
Utilização do conhecimento descoberto	Incorporar o conhecimento descoberto para tomadas de decisão, ou simplesmente documentando-o e relatando-o às partes interessadas. Esse processo também inclui verificar e resolver possíveis conflitos com conhecimento previamente considerado.

Fonte: Adaptado pela autora a partir de Fayyad (1996)

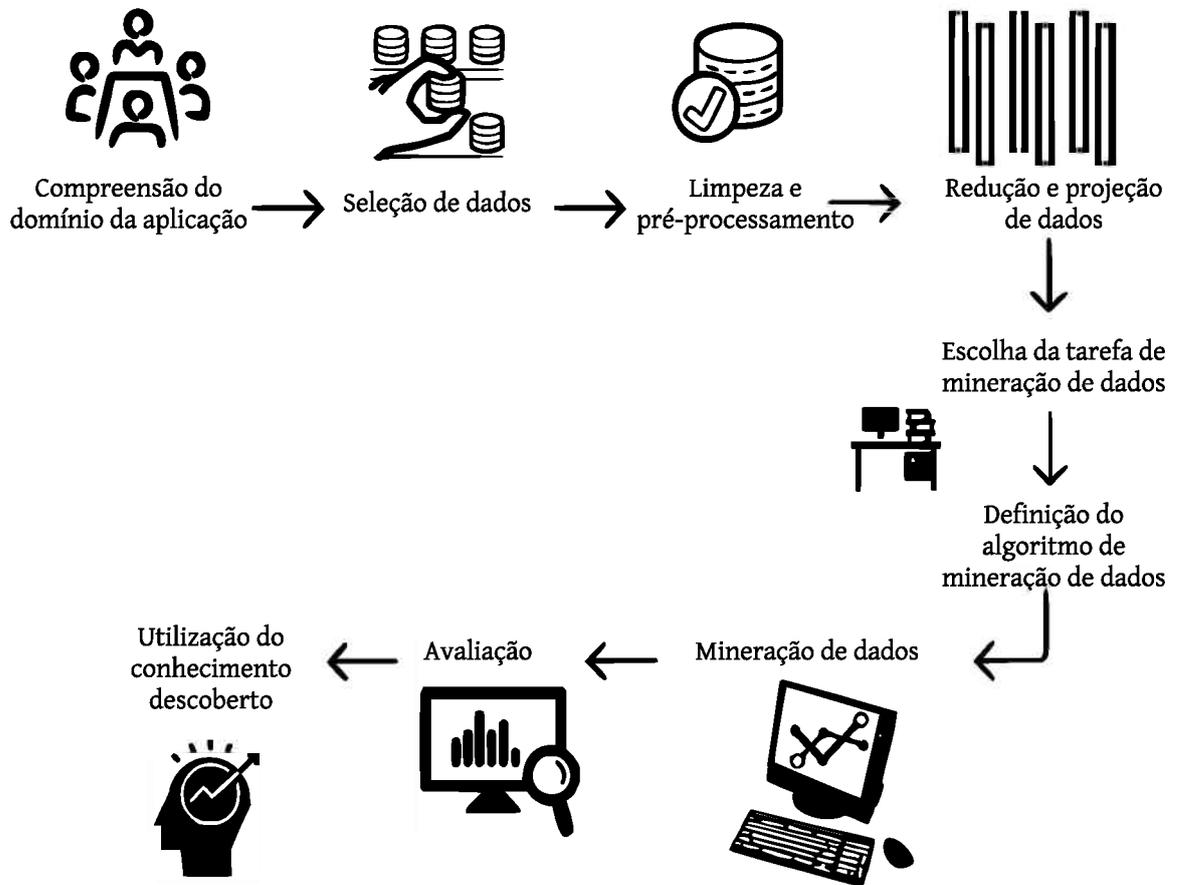


Figura 1 – As etapas do KDD definidas por Fayyad (1996)

Fonte: Adaptado pela autora a partir de Fayyad (1996)

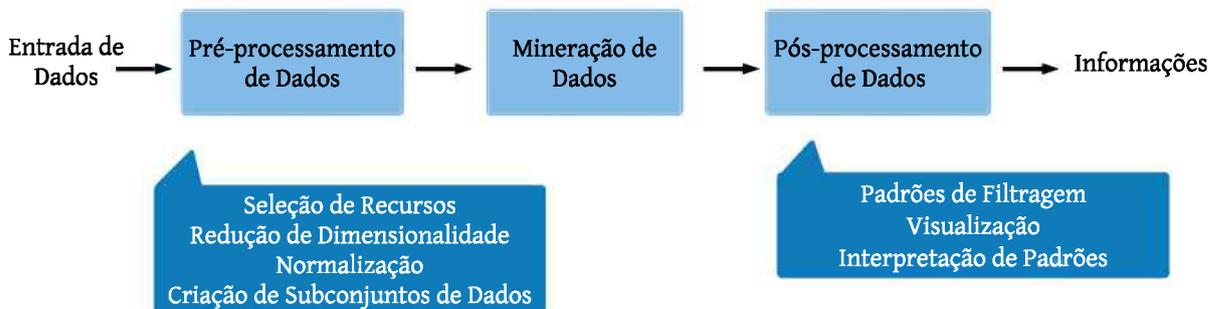


Figura 2 – O processo de descoberta de conhecimento em bancos de dados (KDD).

Fonte: Tan et al. (2009)

Diante disso é possível perceber que os limites precisos das etapas do KDD não são fáceis de declarar por estarem tão intimamente relacionados e assim podem ser descritos em modos variantes.

2.2.1 Dados, Informação e Conhecimento

Como pode-se ler dados, aprender informações e adquirir conhecimento, deve-se diferenciá-los no espaço perceptivo e cognitivo (CHEN et al., 2009). Segundo Angeloni (2003), os dados são um conjunto de fatos (por exemplo, instâncias em um banco de dados). Os dados são elementos brutos, os registros soltos, aleatórios, sem significado, constituem a matéria-prima da informação (ZINS, 2007).

Informação pode ser definida como dados contextualizados, analisados (SETZER, 1999): é qualquer estruturação ou organização desses dados. É a base para o conhecimento ou é um componente do conhecimento.

Numa definição muito simples, conhecimento traduz-se de dados e informações organizados, processados e transformados para obter entedimento quando se aplicam a determinado problema ou atividade (ZINS, 2007).

O conhecimento geralmente se divide em dois tipos: explícito, por exemplo, diretrizes escritas, prontamente disponíveis e que podem ser facilmente passadas a outras pessoas, e implícitas, internalizadas, obtidas pela memória, experiência ou intuição de que o indivíduo pode não estar consciente de que possui, e muito mais difícil de formalizar (BELLINGER et al., 2004).

Apenas o conhecimento prévio permitirá uma compreensão contextual da própria informação, e a informação, por sua vez, conterà características que modificarão esse conhecimento (BOISOT; CANALS, 2004).

Chen et al. (2009) propôs definições de dados, informação e conhecimento em espaço computacional como são mostradas na Tabela 2.

A relação entre dados e conhecimento tem sido descrita variadamente como pirâmide ou cadeia (Figura 3) (BELLINGER et al., 2004).

Assim é possível perceber como dados, informação e conhecimento estão estreitamente conectados porém são conceitos distintos.

Tabela 2 – Dados, informação e conhecimento em espaço computacional.

Categoria	Definição
Dados	Representações de modelos e atributos de entidades reais ou simuladas.
Informação	Dados que representam os resultados de um processo computacional, como a análise estatística, para atribuir significados aos dados ou as transcrições de alguns significados atribuídos por seres humanos.
Conhecimento	Dados que representam os resultados de um processo cognitivo simulado por computador, como percepção, aprendizagem, associação e raciocínio, ou as transcrições de alguns conhecimentos adquiridos por seres humanos.

Fonte: Adaptado pela autora a partir de Chen et al. (2009)



Figura 3 – Relação entre dados, informação e conhecimento.

Fonte: Adaptado pela autora a partir de Bellinger et al. (2004)

2.2.2 Mineração de Dados

A MD é uma etapa não trivial do processo KDD, na qual ocorre a aplicação de algoritmos específicos com um objetivo geral de prever resultados e descobrir relacionamentos implícitos em grandes bases de dados que, caso contrário podem permanecer desconhecidos (FRAWLEY et al., 1992).

A MD denota um campo multidisciplinar de pesquisa e desenvolvimento de algoritmos e ambientes de software para apoiar o KDD no contexto de problemas da vida real, onde muitas vezes demasiadas quantidades de dados estão disponíveis para serem minerados (OLARU; WEHENKEL, 1999). E depende muito de técnicas conhecidas de aprendizado de máquina, reconhecimento de padrões e estatísticas para analisar, descobrir padrões ocultos, associações, anomalias e/ou estrutura em dados (CHEN et al., 1996).

A escolha da tarefa deve ser feita de acordo com os objetivos pré-estabelecidos. As possíveis tarefas de um algoritmo de MD são geralmente divididas, em alto nível, em duas categorias principais (TAN et al., 2009):

- Tarefas Preditivas:

Tarefas de previsão envolvem o uso de variáveis na base de dados para inferir valores desconhecidos ou futuros de outras instâncias (FAYYAD, 1996). As duas principais tarefas preditivas são classificação e regressão. A classificação é uma tarefa de aprendizado supervisionado em que dado um objeto e seus atributos de entrada, a saída de classificação é uma das possíveis classes categóricas pré-definidas (AGRAWAL et al., 1993), por exemplo, prever, baseado em dados históricos, se o cliente será adimplente ou inadimplente. Na regressão, a variável alvo constitui-se em um valor contínuo, ou seja a informação de saída é um valor numérico contínuo ou um vetor de tais valores em vez de uma classe discreta, citando caso análogo: prever a probabilidade de um cliente em perspectiva ser adimplente (TAN et al., 2009).

- Tarefas Descritivas

Tarefas de descrição se concentram em encontrar padrões que descrevam os dados e possam ser interpretados por humanos (FAYYAD, 1996). De acordo com Tan et al. (2009) em tarefas descritivas o objetivo é derivar padrões, como correlações, tendências, grupos, trajetórias e anomalias que condensem os relacionamentos implícitos nos dados. Algumas das tarefas de descrição são agrupamento, regras de associação e sumarização.

2.2.3 Classificação de Dados

A classificação de dados é um dos pilares da aprendizagem de máquina (SRIVASTAVA et al., 1999). A classificação é a tarefa de mapear um conjunto de atributos x no seu rótulo de classe y , os dados de entrada de um algoritmo classificatório são um conjunto de instâncias cada uma delas denotada pela tupla (x,y) , sendo que x denota atributos e y classe. Assim o número de classes é conhecido e o sistema sabe a classe correta durante o treinamento. Se y é discreto caracteriza a classificação, se y é contínuo caracteriza a

regressão. O modelo de classificação é útil tanto para a modelagem preditiva quanto para a descritiva.

Muitos tipos de técnicas foram construídas para classificação, como RNA, NB, AD, métodos estatísticos e métodos genéticos (KOTSIANTIS et al., 2007).

2.2.3.1 Árvores de Decisão

Os métodos de AD são muito popular entre os pesquisadores por causa de sua simplicidade, inteligibilidade e facilidade de implementação. A sua construção é rápida e produz resultados tão bons ou melhores que outros algoritmos mais complexos de classificação (EREMIA et al., 2016). Uma AD é um método preditivo supervisionado que pode ser usado tanto para a tarefa de classificação quanto de regressão (BARROS et al., 2012).

Uma AD pode ser vista como uma estratégia de divisão e conquista para classificação de objetos (EREMIA et al., 2016). Ela é geralmente construída de cima para baixo, partindo do nó raiz e, continuamente, o conjunto é particionado de acordo com suas características (QUINLAN, 1986), resultando em um fluxograma com estrutura de árvore contendo nós e arestas (TAN et al., 2009):

- Um nó raiz onde começa a árvore, contém condição de teste e não possui arestas chegando.
- Nós internos que denotam um teste em um ou mais atributos, cada nó interno possui uma aresta chegando e duas saindo, cada ramificação representa um resultado do teste.
- Cada nó folha representa uma decisão para um valor do atributo classe, possui uma aresta chegando e nenhuma saindo.

O conhecimento em uma AD é representado por cada nó, que ao ser testado conduz a busca para um nó filho, até chegar em um nó folha. A Figura 4 mostra um exemplo de uma AD para uma aplicação de concessão de crédito (para fins ilustrativos). A árvore é avaliada de cima para baixo, onde o nó raiz é um teste no atributo “Casa Própria”. Neste caso se o cliente possui uma então ele não será inadimplente caso contrário é direcionado para o nó-interno à direita, que é um teste no atributo “Estado Civil”, caso seja casado então não será inadimplente, caso contrário (Solteiro ou Divorciado) é direcionado para o nó interno “Renda Anual”. Dependendo do resultado do terceiro teste a decisão é tomada através do valor correspondente ao nó terminal, determinando a aprovação ou não de crédito para o cliente. Para a classificação de cada registro, os nós da árvore são testados e um caminho é percorrido até chegar a um nó folha que contém o rótulo a ser atribuído.

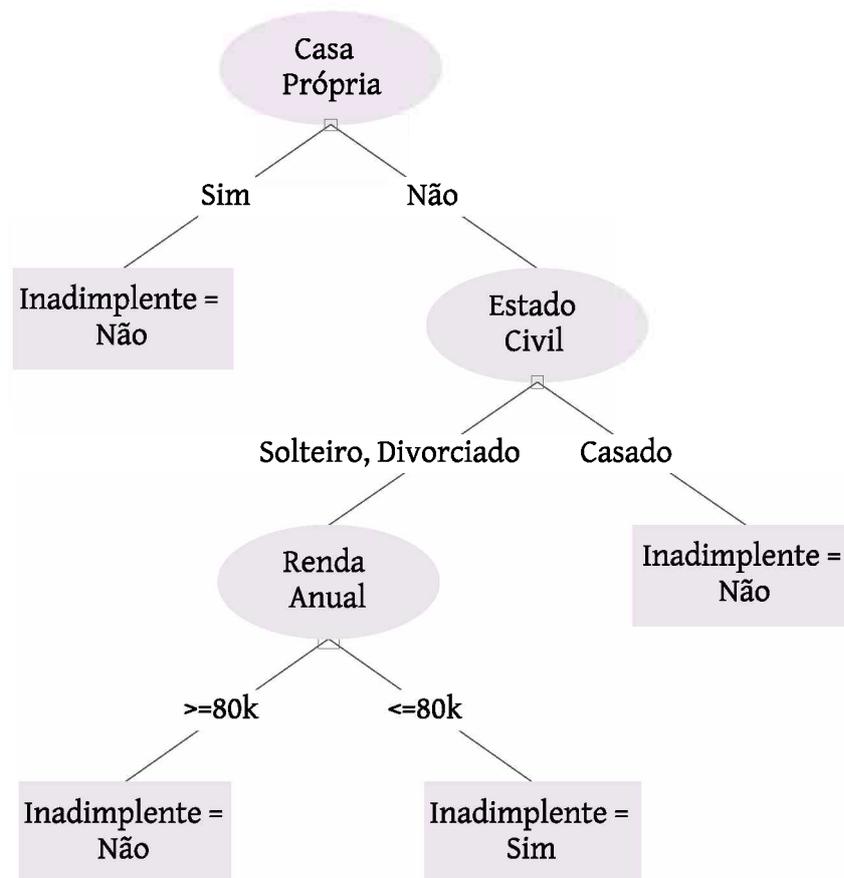


Figura 4 – Exemplo de uma Árvore de Decisão

Fonte: Tan et al. (2009)

2.2.3.2 Florestas Aleatórias

FA é um algoritmo que consiste em um conjunto (ou comitê) de árvores de decisão. Cada uma das árvores é construída usando uma reamostragem dos dados e, em cada divisão, o conjunto de variáveis utilizado é um subconjunto aleatório das variáveis, o que resulta em baixa correlação das árvores individuais (BREIMAN, 2001). Esta técnica também pode ser aplicada tanto para a classificação quanto para a regressão. Para a regressão, ajusta-se a mesma árvore de regressão muitas vezes para as versões de reamostragem dos dados de treinamento, e calcula-se a média do resultado. Para classificação, a classe prevista foi escolhida pela maioria entre o comitê das árvores de decisão (HASTIE et al., 2009a).

2.2.3.3 Máquina de Vetores de Suporte

SVM constitui uma técnica fundamentada na Teoria de Aprendizagem Estatístico (Vapnik, 1999). Ela tem mostrado bom desempenho em várias aplicações reais, como bioinformática, mineração de texto, reconhecimento de faces, e processamento de imagem (WANG, 2005). Essa estratégia foi introduzida por Vapnik e colaboradores e é capaz

de representar funções complexas não lineares (SUYKENS; VANDEWALLE, 1999). A SVM encontra um hiperplano com máxima margem de separação traçando uma linha nos objetos da borda o que é conhecido como vetores de suporte. Para padrões não-linearmente separáveis, os dados de entrada são transformados em um vetor de características multidimensional, para então ser possível separar as características linearmente no espaço, este mapeamento é realizado através de uma representação de kernel (HASTIE et al., 2009a). Essa transformação em um vetor de características multidimensional é mostrada na figura 5.

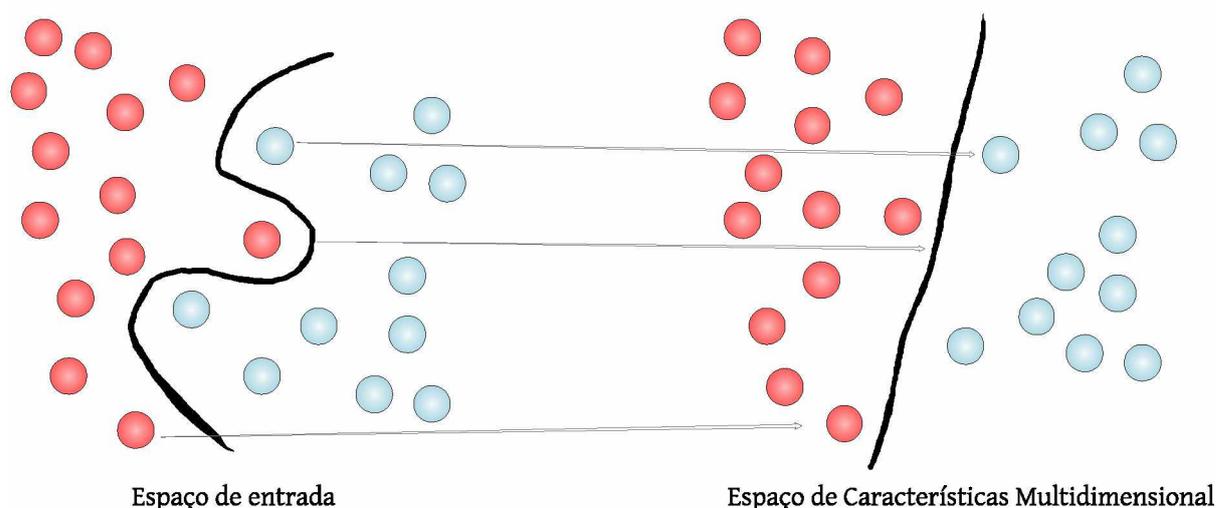


Figura 5 – Transformação da Máquina de Vetores de Suporte

Fonte: Adaptado pela autora a partir de Hastie et al. (2009a)

2.2.3.4 Redes Neurais Artificiais

RNA são modelos computacionais biologicamente inspiradas no modelo do sistema nervoso, que “aprendem” e podem ser utilizadas para classificação e regressão. O aprendizado ocorre quando a RNA atinge uma solução generalizada para dado problema. Existem duas categorias principais de RNA: redes acíclicas ou de alimentação direta, que representam uma função de seu estado atual; e redes cíclicas ou recorrentes que utilizam suas saídas para realimentar suas entradas. A RNA é composta por camadas de unidades ou “neurônios” interconectados, que calculam determinadas funções matemáticas. Basicamente, essas camadas são a de entrada, as ocultas e a de saída. Neurônios que se conectam com os dados de entrada compõem a camada de entrada, a camada final constitui a camada de saída, a camada intermediária, composta por neurônios que não pertencem a camada de entrada nem a de saída, é conhecida como oculta (KOVÁCS, 2002). Uma RNA pode possuir uma ou múltiplas camadas ocultas.

A Figura 6 é uma elucidação de uma RNA de duas camadas ocultas. Em RNA o processo usual para a solução do problema consiste em uma fase de aprendizagem, em que

um conjunto de dados históricos é apresentado para a rede, que extrai as características necessárias para modelar os dados fornecidos como uma função não linear, e o aprendizado ocorre através do ajuste de pesos. O resultado é um eficiente método de aprendizado, com aplicações difundidas em muitos campos (HASTIE et al., 2009a).

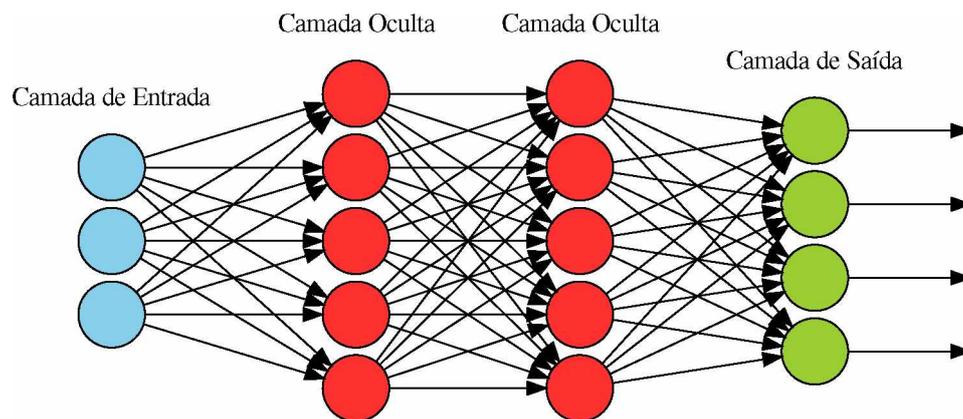


Figura 6 – Exemplo de uma Rede Neural Artificial

Fonte: Adaptado pela autora a partir de Tsai e Wu (2008)

2.2.3.5 Regressão Logística

RL é um método estatístico preditivo válido para problemas em que o número de classes seja dicotômico. Busca estimar a probabilidade da variável alvo assumir uma determinada classe baseado nas outras instâncias já mapeadas, os resultados da análise ficam contidos no intervalo de zero a um. Sua classificação é feita por um hiperplano, que assume que os dados são linearmente separáveis para bons resultados. Essa técnica tem como benefício a possibilidade de saber o valor da probabilidade da classe predita. É bastante utilizada em aplicações bioestatísticas onde respostas dicotômicas ocorrem com bastante frequência. Por exemplo, os pacientes sobrevivem ou morrem, têm doenças cardíacas ou não, ou uma condição está presente ou ausente (HASTIE et al., 2009a).

2.2.4 Validação e Medidas de Desempenho de um Classificador

Depois de criar um modelo preditivo, os profissionais de crédito usarão esse modelo de classificação para prever quais clientes estão propensos a serem inadimplentes. Assim são essenciais técnicas para avaliar o desempenho dos classificadores já que fornecem uma avaliação imparcial de erro e generalização (LINOFF; BERRY, 2011).

No método *holdout* os dados originais são particionados em dois conjuntos disjuntos: treinamento ($t1$) e teste ($t2$). Normalmente utiliza-se 2/3 dos dados para treinamento e o restante para teste (TAN et al., 2009). Um modelo de classificação é então induzido a partir do conjunto $t1$ e seu desempenho é avaliado no conjunto $t2$. Limitações:

- O modelo induzido pode não ser tão bom quanto quando todos os dados são usados para treinamento.
- O modelo é altamente dependente da composição dos conjuntos $t1$ e $t2$, assim a avaliação pode ter uma alta variância: quanto menor o conjunto $t1$ maior a variância do modelo, por outro lado se o conjunto $t1$ for muito grande então a precisão estimada calculada a partir do conjunto menor é menos confiável.
- Uma classe que esteja representada em excesso em um subconjunto ($t1$ ou $t2$) estará representada de menos na outra e vice versa.

No método de *validação cruzada* (VC), cada registro é usado o mesmo número de vezes para treinamento e pelo menos uma vez para teste. Define-se dois subconjuntos: treinamento e teste. Em seguida são trocados os papéis dos subconjuntos, isto é treinamento passa a ser teste e vice versa. Esta é a abordagem validação cruzada de duas pastas. O método de validação cruzada de k pastas segmenta os dados em k partições de tamanho igual. Os conjuntos podem ser estratificados¹ antes de executar a VC. Durante cada execução, uma das partições é escolhida para teste, enquanto que as outras são usadas para treinamento. Este procedimento é repetido k vezes de modo que cada partição seja usada para teste exatamente uma vez (TAN et al., 2009).

A acurácia é a métrica de avaliação mais utilizada de um classificador. A acurácia mede a frequência com que o classificador faz a previsão correta e calcula a proporção de casos classificados corretamente para o número total de casos, (ALPAYDIN, 2014), o quão acurado é o modelo, é obtido através da tabela de confusão. A matriz de confusão mostra o número de classificações corretas versus as classificações preditas para cada classe, como é mostrado na Tabela 3.

Na matriz, o valor “*Positivo*” refere-se ao que o classificador classificou como pertencente à classe (inadimplente); “*Negativo*” ao que o classificador classificou como não pertencente à classe; “*Falso*” no que o classificador categorizou erroneamente; e “*Verdadeiro*” no que ele acertou.

É importante falar sobre matriz de confusão e as medidas que podemos derivar dela, e também sobre outras métricas de desempenho para casos específicos, por exemplo F1 quando os problemas são desbalanceados.

¹ A estratificação garante a representatividade das classes

Tabela 3 – Matriz de confusão para um problema de duas classes

		Classes Preditas	
		Inadimplente	Adimplente
Classes Reais	Inadimplente	Verdadeiro Positivo	Falso Negativo
	Adimplente	Falso Positivo	Verdadeiro Negativo

Além da acurácia, também é possível, a partir da matriz de confusão a realização de outros cálculos, como a *sensitividade* (*sens*), ou seja, a capacidade do algoritmo categorizar os registros classificados corretamente como inadimplentes dentre todas as que realmente foram inadimplentes. A *especificidade* (*esp*) é a capacidade do algoritmo de categorizar os registros adimplentes. E ainda a *precisão* (*pre*), que avalia o quanto o modelo acerta. A seguir é mostrado como calcular cada uma dessas medidas.

$$Acu = \frac{VP + VN}{VP + VN + FP + FN}$$

$$Sens = \frac{VP}{VP + FN}$$

$$Esp = \frac{VN}{VN + FP}$$

$$Pre = \frac{VP}{VP + FP}$$

Caso o conjunto de dados esteja desbalanceado é interessante o uso da medida *F1*. Ela é uma medida harmônica entre a precisão e a sensibilidade. Em geral, quanto maior a medida *F1*, melhor.

$$F1 = \frac{2 \cdot \textit{precisão} \cdot \textit{sensitividade}}{\textit{precisão} + \textit{sensitividade}}$$

2.3 Aprendizado Profundo

Aprendizado profundo (DL) é o subconjunto das técnicas de aprendizado de máquina em que os modelos são baseados em redes neurais com múltiplas camadas (TOULOUPAS, 2018). Para alguns autores DL significa que as camadas ocultas da rede neural são a partir de duas (ZHANG, 2014), e para outros autores para ser classificado como um trabalho de DL o uso de redes neurais deve conter um número relativamente grande de camadas ocultas (mais de cinco) (LIAO, 2017). As redes neurais profundas têm alcançado resultados empíricos superiores para vários domínios de problemas em comparação com

outros modelos de aprendizado de máquina (TOULOUPAS, 2018). De modo geral, DL é sobre aprender múltiplos níveis de representação e abstração para ajudar a analisar e a entender dados como som, vídeo, imagens, texto e assim por diante. Seu sucesso pode ser atribuído à sua capacidade de aprender modelos complexos, exibindo ao mesmo tempo uma boa generalização

A arquitetura hierárquica de múltiplas camadas é uma inspiração da biologia e da cognição humana. O cérebro humano é conectado por sinapses e neurônios, que formam uma rede profunda camada por camada, onde uma unidade, em rede profunda, representa o neurônio e cada conexão dentro da rede representa a sinapse. Os pesos da borda representam a relação entre dois neurônios conectados ou a força entre eles (ZHANG, 2014). A Figura 7 ilustra essa comunicação entre neurônios. A informação de um neurônio flui para outro através de um pequeno espaço chamado sinapse. Na sinapse, os sinais elétricos são traduzidos em sinais químicos para atravessar a lacuna. Uma vez do outro lado, o sinal se torna elétrico novamente. No cérebro, um neurônio emissor pode se conectar a vários neurônios receptores, e um neurônio receptor pode se conectar a vários neurônios emissores (SALINAS; SEJNOWSKI, 2001).

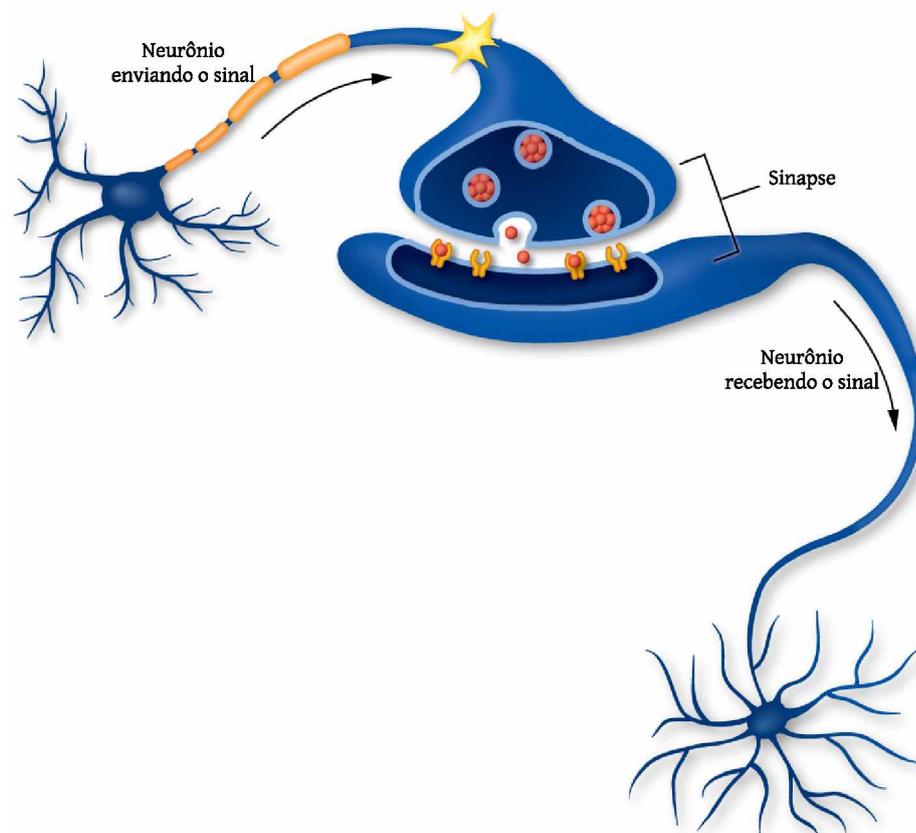


Figura 7 – Representação de neurônios se comunicando através da sinapse

Fonte: <https://learn.genetics.utah.edu/content/neuroscience/neurons>

2.3.1 Redes Neurais Profundas (DNN)

Uma DNN é um perceptron multicamada convencional com muitas (geralmente mais de duas) camadas ocultas (ZHANG, 2014). As DNN utilizam métodos de aprendizagem de representação com múltiplos níveis de camadas, obtidos pela construção de módulos não lineares. Cada módulo opera sobre uma entrada, começando com a entrada de dados, gerando uma representação que é então passada para a próxima camada, uma representação em um nível mais alto e um pouco mais abstrata (LECUN et al., 2015). Assim, as DNN "aprendem" por meio de composições de funções e ajustam seus próprios parâmetros diretamente a partir dos dados e das saídas da camada anterior.

2.3.2 Deep Belief Networks (DBN)

Máquinas restritas de Boltzmann (RBM) são modelos probabilísticos que são usados como recursos não-supervisionados não-lineares, consistindo de um conjunto de unidades ocultas binárias h , um conjunto de unidades visíveis (binárias ou reais) v e uma matriz de pesos w associada às conexões entre as duas camadas (LEE et al., 2009). A função de probabilidade conjunta para unidades visíveis e ocultas é definida como:

$$P(v, h) = \frac{1}{Z} \exp(-E(v, h))$$

Onde Z é a função de partição e $E(v, h)$ é a função de energia. Para um RBM com unidades visíveis de valor real, a função de energia é definida como:

$$E(v, h) = \sum_{s=1}^{n^v} b_s^v v_s - \sum_{t=1}^{n^h} b_t^h h_t - \sum_{s=1}^{n^v} \sum_{t=1}^{n^h} v_s h_t w_{st}$$

Onde v_s e h_t são as s^{th} e t^{th} unidades de v e h , b_s^v e b_t^h são os bias associados à unidade v_s e h_t , e w_{st} é o peso associado à conexão entre v_s e h_t (SIHAG; DUTTA, 2015).

A Deep Belief Network, ou DBN, em termos de estrutura, é idêntica a um perceptron multicamadas (MLP), mas difere quanto ao método de treinamento. O método é conhecido como pré-treinamento em camada. Essencialmente, a DBN é treinada em duas camadas de cada vez, e essas duas camadas são tratadas como um RBM. Em toda a rede, a camada oculta de um RBM atua como a camada de entrada do adjacente. Portanto, o primeiro RBM é treinado e suas saídas são usadas como entradas para o próximo RBM. Este procedimento é repetido até que a camada de saída seja atingida. Após esse processo de treinamento, a DBN é capaz de reconhecer os padrões inerentes aos dados. Em outras palavras, é um sofisticado extrator de características multicamadas. O aspecto único desse tipo de rede é que cada camada acaba aprendendo toda a estrutura de entrada. Em outros tipos de redes profundas, as camadas geralmente aprendem padrões progressivamente complexos, para o reconhecimento facial, as camadas iniciais poderiam detectar bordas e

camadas posteriores as combinariam para formar características faciais. Por outro lado, a DBN aprende os padrões ocultos globalmente, como uma câmera colocando lentamente uma imagem em foco.

Por fim, a DBN requer um conjunto de rótulos para aplicar aos padrões resultantes. Como etapa final, a DBN é aperfeiçoada (fine tuning) com o aprendizado supervisionado e um pequeno conjunto de exemplos rotulados, assim o processo de treinamento não requer muitos dados rotulados. Depois de fazer pequenos ajustes nos pesos e bias, a rede alcançará um leve aumento na precisão. A rede resultante é normalmente muito precisa (ZHANG, 2014).

2.3.3 Redes Neurais Recorrentes (RNR)

Uma RNR é uma forma de redes neurais adequada para processar sequências de dados. Ela é cíclica o que permite que as informações persistam e isso fornece a memória do modelo. A Figura 8 ilustra um RNR básica, ela pode ser ilustrada como um ciclo e como uma rede desenrolada (SKREDE, 2017).

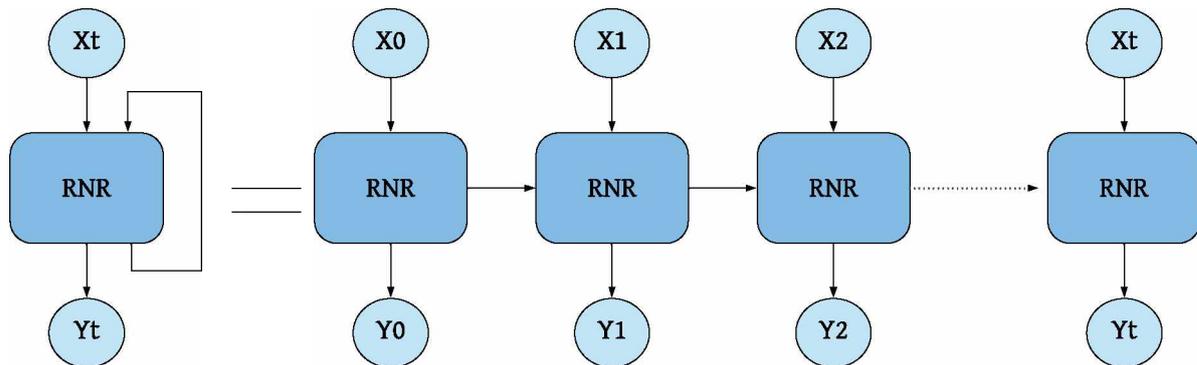


Figura 8 – Uma RNR simples. Representada em ciclo (esquerda) e como desenrolada (direita) em t passos de tempo.

Em cada etapa de tempo, a RNR recebe como entrada e entrada externa a partir da última etapa de tempo e duas saídas são criadas. Uma saída é passada para a camada de saída ou para a próxima camada oculta, se houver. A outra saída é o estado da RNR, que é passada para a próxima etapa de tempo (SKREDE, 2017).

Exemplos de modelos são LSTM e GRU. Um bloco LSTM é composto de quatro componentes principais: um estado de célula, uma porta de entrada, uma porta de saída e uma porta de esquecimento. Os portões atuam como reguladores do fluxo de informações que passam pelas conexões do LSTM e podem, assim, controlar o estado da célula. Essas portas permitem que o estado da célula se lembre das informações em intervalos de tempo arbitrários (DOSHI, 2018).

2.3.4 Redes Neurais Convolucionais (RNC)

RNC é uma das arquiteturas de aprendizagem profunda mais conhecidas e amplamente utilizadas, inspiradas pelo mecanismo natural de percepção visual dos seres vivos. RNC são projetadas para processar dados com topologia tipo grade, por exemplo, uma imagem colorida composta de três vetores bidimensionais que contêm intensidades de pixel nos três canais de cores. Nos últimos anos, as RNC tiveram um grande sucesso em tarefas de visão computacional (CAO, 2018).

A arquitetura da RNC possui neurônios dispostos em três dimensões: largura, altura e profundidade. As RNC são usadas principalmente com imagens como entrada, por exemplo, em uma imagem que tem dimensões $32 \times 32 \times 3$ os neurônios em uma camada se conectarão a uma região personalizada da camada anterior. Além disso, a camada de saída final tem dimensões $1 \times 1 \times d$, onde d é o número de classes. As dimensões são reduzidas de 3072 para d . A saída é um único vetor de distribuições de classe (ZHANG, 2016).

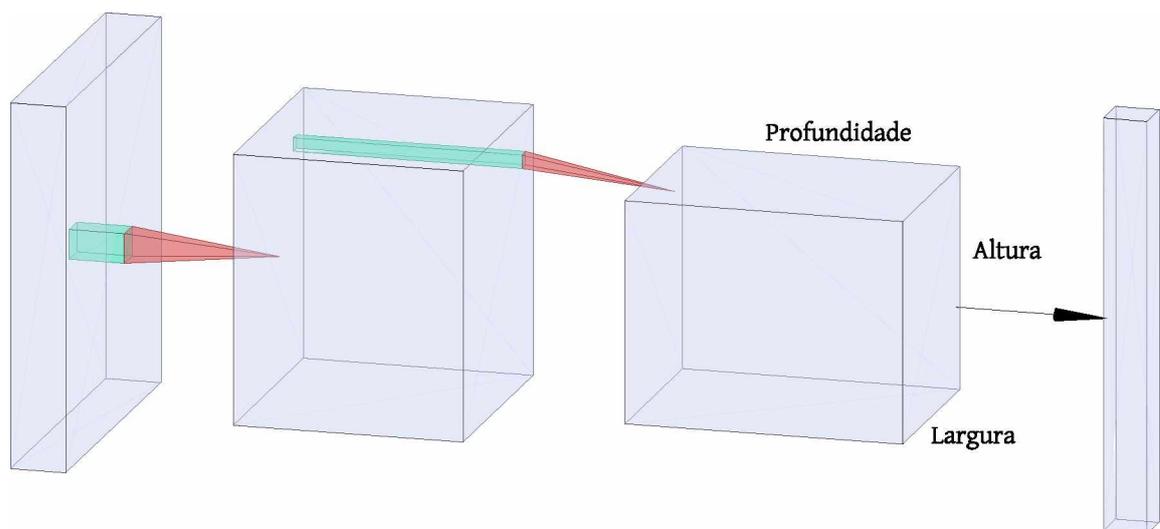


Figura 9 – Uma RNC em três dimensões. Adaptado pela autora a partir de (DOSHI, 2018)

RNC possuem quatro principais tipos de camadas:

- Camadas convolucionais;
- Camadas não lineares;
- Camadas de agrupamento; e
- Camadas totalmente conectadas.

Cada camada transforma a entrada na saída através de uma função diferente. Em uma rede neural convolucional simples, a entrada é passada através de uma série de camadas

convolucionais, camadas não lineares com funções ReLU (Unidades Lineares Retificadas), camadas de agrupamento e camadas totalmente conectadas. Há quatro ideias-chave por trás das RNC que aproveitam as propriedades dos sinais naturais: conexões locais, pesos compartilhados, agrupamento e o uso de várias camadas (CAO, 2018).

RNC são construídas sobre a ideia de conectividade local. Usa-se convoluções sobre a camada de entrada para calcular a saída em uma RNC, isso significa que um neurônio na camada de saída é conectado apenas a um conjunto de neurônios na camada de entrada, resultando em conexões locais. A camada de convolução extrai padrões específicos dos dados criando um mapa de características. Ter várias camadas de convolução permite que a rede aprenda filtros de alto nível com relativamente menos parâmetros (DOSHI, 2018). Todas as unidades em um mapa de recursos compartilham o mesmo banco de filtros. Mapas de características diferentes em uma camada convolucional usam diferentes bancos de filtros (CAO, 2018). O mapa de características produzido pela camada de convolução é passado através de ReLU, uma função de ativação não linear comumente usada que substitui todo o valor negativo no mapa de característica por zero (DOSHI, 2018). As camadas de agrupamento visam reduzir a dimensão espacial da representação e controlar o super ajuste reduzindo a quantidade de parâmetros. As camadas totalmente conectadas são normalmente usadas como as últimas camadas da rede para melhor resumir as informações transmitidas pelas camadas de nível inferior no mapeamento final das classes (CAO, 2018). Famosas arquiteturas de RNC são AlexNet, VGGNet, ResNet and DenseNet.

3 Trabalhos Relacionados

No campo de Tecnologia da Informação, são muitos os trabalhos que se voltam para mitigar os problemas de Finanças. O estado da arte conduzido pela análise automática de crédito leva à estatística (HAND; HENLEY, 1997), à inteligência artificial (BAHRAM-MIRZAEI, 2010) e ao aprendizado de máquina (ALPAYDIN, 2014; LIN et al., 2012). Para fins didáticos, este capítulo é dividido em duas seções: a primeira de propósito geral a qual discute abordagens de aprendizado de máquina para problemas de finanças; e a segunda de propósito específico, a qual considera a aplicação de métodos de aprendizado profundo em tal contexto.

3.1 Aprendizado de máquina para problemas de Finanças

Em (YEH; LIEN, 2009) os autores analisaram o caso de inadimplência de cartão de crédito de clientes taiwaneses comparando a acurácia de seis métodos de mineração de dados (análise discriminante, RL, NB, k -NN, RNA e AD). Os dados foram divididos em dois grupos aleatórios, um para treinar o modelo e outro para validá-lo e utilizaram a área sob a curva (AUC) para avaliar os resultados. Segundo os autores do ponto de vista da gestão de risco, a probabilidade de inadimplência é mais importante do que o resultado binário adimplentes e inadimplentes, para tanto este estudo propôs o “*Sorting Smoothing Method*” para deduzi-la. Entre as seis técnicas de mineração de dados, a RNA foi a única que pôde estimar com precisão a probabilidade real de inadimplência.

Os autores Malekipirbazari e Aksakalli (2015) propuseram em seu trabalho um método de classificação baseado na técnica floresta aleatória para prever o status do tomador de crédito em dados de uma plataforma de empréstimo social, a Lending Club, que são empréstimos entre pares na qual credores e tomadores de empréstimos podem fazer negócios sem a ajuda de intermediários institucionais, como os bancos. Os autores utilizaram o método de validação cruzada de cinco pastas. Conduziram, ainda, o processo de definição dos melhores parâmetros da floresta aleatória em uma tarefa de duas etapas: (1) determinar o tamanho ótimo da FA e (2) determinar os valores ótimos de m e d , onde m representa o número de divisão de características e d a profundidade máxima da FA. Foi estabelecido como tamanho da FA 80 com $m = 5$ e $d = 25$. Os autores comparam a FA com três técnicas de classificação alternativas k -NN, SVM e RL. E concluem afirmando que seus resultados computacionais indicam que a FA superou os outros classificadores mostrando-se a mais adequada para prever corretamente os tomadores de crédito.

Os autores apresentaram neste trabalho (KRUPPA et al., 2013) uma estrutura geral para estimar riscos de tomadores de crédito por meio de métodos de aprendizado

de máquina. Utilizando as técnicas florestas aleatórias, k-NN e bagged k-vizinhos mais próximos (bNN) e regressão logística otimizada para um grande conjunto de dados de históricos de pagamento de créditos parcelados de curto prazo. O tamanho da FA foi igual a 500 e o número de divisão de características $m = 5$, estendendo sua própria implementação de floresta aleatória: *Random Jungle*, um pacote em C++. Para a análise, dividiram os dados aleatoriamente em 2/3 para treinamento e 1/3 para teste. Utilizaram a área sob a curva (AUC) para avaliar os resultados. Completaram que a FA atinge resultados substancialmente melhores do que os demais modelos classificadores.

Em (HUANG et al., 2007), os autores utilizaram três estratégias para construir modelos híbridos baseados em SVM para avaliar o risco de crédito do candidato: (1) usando busca em grid para otimizar parâmetros do modelo, (2) usando busca em grid para otimizar parâmetros do modelo e usando F-score para selecionar as características de entrada, e (3) usando o algoritmo genético (AG) para otimizar simultaneamente os parâmetros do modelo e as características de entrada. Dois conjuntos de dados de crédito no banco de dados da UCI foram utilizados (alemão e australiano). O SVM é comparado com RNA, AG e AD. O classificador SVM alcançou uma acurácia tão boa quanto à RNA e com menos características. Além disso, combinando algoritmos genéticos, a estratégia híbrida GA-SVM proposta, pôde executar simultaneamente a tarefa de seleção de atributos e a otimização dos parâmetros do modelo.

Foi realizado o teste não paramétrico de Friedman - um teste para as amostras dependentes de k - e não foram encontradas diferenças significativas entre a acurácia dessas três estratégias tanto para ambos os conjuntos de dados. Foi realizado também testes de ranking não paramétricos de Wilcoxon em que não foram encontradas diferenças significativas para a base alemã quanto à quantidade de características selecionadas. No entanto, na base australiana, SVM + AG tem um número significativamente menor de características selecionadas. Os resultados acima revelam que uma estratégia baseada no AG é uma alternativa aceitável para otimizar ambos o subconjunto de características e os parâmetros do modelo para a análise de crédito.

Para o conjunto de dados australiano, não houve diferenças significativas na acurácia entre a SVM (SVM + AG), AG, BPN e C4.5 com base no teste de Friedman. Para o conjunto de dados alemão, no entanto, encontraram que SVM, AG e BPN não possuíam diferenças significativas, mas que o modelo C4.5 era significativamente inferior às outras três abordagens baseado nos testes de ranking não-paramétricos de Wilcoxon. No que diz respeito ao tempo de execução, no entanto o tempo de CPU do C4.5 é bem menor em comparação com as demais abordagens.

Em (KHASHMAN, 2010) o autor utilizou da base de dados de crédito de um banco alemão fornecido pelo UCI para implementar da técnica de redes neurais o algoritmo BPN, usando uma única camada oculta (com o número de nós ocultos variando com o modelo

de rede neural) e um único nó na camada de saída, tais configurações visam minimizar os custos computacionais e de tempo. A camada de saída tem um único neurônio: “0” para aceitar ou “1” para rejeitar um pedido de crédito.

Foram treinados e implementados três modelos de rede neural usando nove esquemas de aprendizado. O sistema consistiu em duas fases: uma fase de pré-processamento de dados em que cada valor numérico dos atributos do candidato era normalizado separadamente. A saída dessa fase fornecia valores numéricos normalizados, que eram usados na segunda fase: avaliar os atributos do candidato e decidir se deveria aceitar ou rejeitar o candidato. Os autores encontraram o número ideal de nós ocultos h , após experimentos envolvendo o ajuste de nós de 1 para 50: RNA-1 com $h = 18$, RNA-2 com $h = 23$ e RNA-3 com $h = 27$. Se o resultado de saída da BPN era maior ou igual a 0.5, o caso apresentado era atribuído a uma classe (bom, aceitar); caso contrário, era atribuído à outra classe (ruim, rejeitar).

Os esquemas de aprendizagem diferiram na proporção de dados de treinamento para validação. Por exemplo, o primeiro esquema de aprendizado (LS1) usa uma proporção de (100: 900); Ou seja, os primeiros 100 casos de solicitação de crédito são usados para treinar a rede, enquanto os 900 casos restantes para testar ou validar a classificação da rede após a conclusão do treinamento. Como critério de avaliação foram definidos o valor de erro mínimo exigido de 0,008 e o número máximo de iterações igual a 25.000, uma vez que um número maior de iterações levaria ao aumento dos custos computacionais.

Apenas três implementações satisfizeram os critérios de avaliação. Em primeiro lugar, o modelo neural RNA-2 no esquema de aprendizagem LS4 (taxa de aprendizado: 400: 600) que satisfaz ambos os critérios ao requerer o menor número de iterações (18.652 iterações) e ao convergir com o valor de erro definido pelos autores (0,008). Em segundo lugar, a implementação usando o modelo neural RNA-1 no esquema de aprendizagem LS5 (taxa de aprendizado: 500: 500) também satisfaz ambos os critérios embora com um número maior de iterações (19.236 iterações). Em terceiro lugar, a implementação usando o modelo neural RNA-3 no esquema de aprendizagem LS6 (taxa de aprendizagem: 600: 400) requerendo 25.000 iterações para convergir para um valor de erro de 0,008531. As restantes 27 implementações não satisfizeram os critérios. Ao considerar os custos computacionais, o modelo neural RNA-2, sob o esquema de aprendizagem LS4 (400: 600) é mais eficiente.

O autor Tsai (2009) teve como objetivo a comparação de cinco métodos de seleção de atributos conhecidos pela literatura de avaliação de risco de crédito, que são teste-t, matriz de correlação, stepwise regression, análise de componentes principais e análise fatorial. Para examinar o seu desempenho de previsão são usadas como modelo BPN baseado em apenas uma camada oculta, com quatro números diferentes de nós ocultos (8, 16, 32 e 64) e épocas de aprendizado (50, 100, 200 e 400). Para fornecer uma conclusão confiável o autor utilizou cinco conjuntos de dados (crédito alemão, crédito japonês, bankruptcy dataset, e conjuntos de dados da competição UC). O método de validação cruzada com

cinco pastas foi usado. Como medidas de desempenho foram avaliadas a acurácia e as taxas de erro tipo I e tipo II. Este trabalho baseou-se em três etapas. Na primeira não foi aplicado nenhum método de seleção de atributos. Na segunda etapa foram aplicados os cinco métodos de seleção de atributos. Na terceira etapa foi avaliado o desempenho dos modelos. Foi realizado um ranqueamento de acordo com os resultados e para acurácia o melhor foi a análise fatorial e em segundo o teste t. Para o erro Tipo I, o primeiro lugar foi o teste t e o segundo o stepwise. Matriz de correlação e stepwise foram o primeiro e segundo métodos, respectivamente, para o erro do Tipo II. O autor conclui que, com relação aos resultados experimentais, os métodos de seleção de recursos certamente aumentam o desempenho da predição. Em suma, o teste t é o melhor método de seleção de recursos para fornecer maior acurácia e menores erros de Tipo I e II por ser mais estável do que outros.

Neste artigo (TSAI; WU, 2008), os autores compararam o desempenho de classificadores únicos com o de classificadores múltiplos e com de classificadores múltiplos diversificados, para tanto utilizaram de RNA. Três conjuntos de dados financeiros foram escolhidos para os experimentos, que são o crédito australiano, o crédito alemão e o crédito japonês. Para a construção do modelo da rede neural utilizaram a BPN de três camadas ocultas. O número de nós nas camadas ocultas variou de 8 a 32 e foram utilizadas quatro diferentes épocas de aprendizado (50, 100, 200 e 300). Para avaliação compararam a acurácia e as taxas de erros I e II. Para a criação dos classificadores únicos cada conjunto de dados foi dividido em dados de treinamento e teste aleatoriamente, na proporção de 70-30%. Para a criação dos classificadores múltiplos (ensemble) as saídas de várias redes neurais treinadas diferentes são combinadas para produzir uma saída final. Então, a saída que recebe a maioria dos votos é selecionada como decisão final de classificação. O classificador único teve de acurácia, considerando as épocas e números de nós ocultos, como os melhores classificadores de RNA 97,32% (100:24), 78,97% (200:32) e 87,94% (200:16), respectivamente, nas bases de crédito australiano, alemão e japonês. Comparando estes resultados, os classificadores múltiplos só têm melhor desempenho para o conjunto de crédito alemão. Os múltiplos classificadores diversificados possuem um desempenho pior em todos os conjuntos de dados. No entanto, comparando as taxas de erros Tipo I e Tipo II, não há um método melhor exato.

3.2 Deep learning para problemas de Finanças

Os autores Deng et al. (2016) introduziram uma nova estrutura RDNN (Recurrent Deep Neural Network) para sensoriamento simultâneo de ambiente e tomada de decisão recorrente para negociação na avaliação financeira. O modelo é a combinação de técnicas de aprendizado profundo e de aprendizado por reforço (RL). A parte de DL detecta automaticamente a condição de mercado dinâmico para o aprendizado de atributos

informativos. Em seguida, o módulo de RL interage com representações profundas e toma decisões comerciais. Os conceitos de aprendizagem fuzzy são introduzidos para reduzir a incerteza dos dados de entrada e para melhorar ainda mais a robustez para a sumarização do mercado. O modelo é validado com dados de mercados de ações. Este é o primeiro artigo a implementar DNN e a demonstrar a eficácia de projetar um sistema de negociação em tempo real.

Neste outro artigo, os autores Chen et al. (2016) desenvolveram uma estrutura de DL baseada em redes neurais convolucionais (RNC) para analisar os dados de séries temporais de negociação. O sistema proposto é implementado e validado nos conjuntos de dados históricos do Taiwan Stock Index Futures. Os dados são transformados em dados de imagens bidimensionais como entrada para a RNC através de diferentes métodos de visualização. Os resultados experimentais mostram que o sistema de aprendizagem proposto pode entender informações mais complexas e úteis do que as tradicionais negociações baseadas em regras como a tendência do preço das ações, suas características e previsibilidade.

Sun e Vasarhelyi (2018) demonstram a eficácia do DL na previsão de inadimplência em cartões de crédito. Eles usaram dados reais ligados a 711.397 titulares de cartões de crédito em um banco do Brasil. Desenvolveram uma DNN para prever inadimplência com base nas informações pessoais dos clientes e das características de gastos. Eles compararam o desempenho preditivo da DNN com os modelos tradicionais de RNA, LR, NB e AD. Utilizaram CV para validação e busca em grid para otimização de hiperparâmetros. A DNN têm um melhor desempenho preditivo geral comparado com os outros modelos. A aplicação bem sucedida da aprendizagem profunda implica que a inteligência artificial tem um grande potencial para apoiar e automatizar a avaliação do risco de crédito para instituições financeiras e agências de crédito.

4 Desenvolvimento

Este capítulo tem os seguintes objetivos: 1) apresentar e descrever os conjuntos de dados em estudo; 2) apresentar o método empregado neste trabalho para a construção das técnicas mencionados nos capítulos anteriores, onde serão apresentados os passos necessários para a criação dos modelos, para a seleção de parâmetros e validação dos resultados.

4.1 Conjuntos de Dados para Avaliação

Aqui, é discutido em detalhes os conjuntos de dados de avaliação de crédito, disponíveis publicamente no repositório de dados da UCI Machine Learning (ASUNCION; NEWMAN, 2007), bem como as etapas de preparação e pré-processamento realizadas em cada uma delas. A fim de fornecer uma conclusão mais abrangente são utilizados três conjuntos de dados de crédito, o australiano, o alemão e o taiwanês.

4.1.1 Dados de Crédito Australiano

Este conjunto de dados lida com aplicações de cartão de crédito. Todos os nomes e valores de atributos foram alterados para símbolos sem sentido para proteger a confidencialidade dos dados. Ele contém 14 atributos, onde seis são atributos contínuos e oito são atributos categóricos.

Havia também alguns valores ausentes, dentre eles 37 casos (5%) tinham um ou mais valores ausentes. Os valores ausentes foram substituídos pelo moda do atributo, se fossem atributos categóricos e se fossem atributos numéricos, os valores ausentes foram substituídos pela média do atributo. Os tipos dos atributos são mostrados na Tabela 4.

4.1.2 Dados de Crédito Alemão

O segundo conjunto de dados foi fornecido pelo Prof Hofmann em Hamburgo. Os dados originais têm uma mistura de 20 atributos categóricos e numéricos, conforme apresentado na Tabela 5, e documentam várias informações financeiras e demográficas sobre os clientes. Para algoritmos que não lidam com atributos categóricos, a Universidade de Strathclyde produziu uma versão numérica desse conjunto de dados que também está disponível na UCI, o que aumenta a dimensão para 24 valores numéricos de entrada. As instâncias são rotuladas como 1 (boa) e 2 (ruim).

Tabela 4 – Atributos no conjunto de dados Australiano

Atributo	Descrição	Tipo
1	X1	Categórico
2	X2	Numérico
3	X3	Numérico
4	X4	Categórico
5	X5	Categórico
6	X6	Categórico
7	X7	Numérico
8	X8	Categórico
9	X9	Categórico
10	X10	Numérico
11	X11	Categórico
12	X12	Categórico
13	X13	Numérico
14	X14	Numérico

Tabela 5 – Atributos no conjunto de dados Alemão

Atributo	Descrição	Tipo
1	Status da conta corrente existente	Categórico
2	Duração em meses	Numérico
3	Histórico de crédito	Categórico
4	Propósito	Categórico
5	Conta de crédito	Numérico
6	Conta poupança/títulos	Categórico
7	Emprego atual desde	Categórico
8	Taxa de parcelamento em porcentagem da renda disponível	Numérico
9	Status pessoal e sexo	Categórico
10	Outros devedores/fiadores	Categórico
11	Residência atual desde	Numérico
12	Propriedade	Categórico
13	Idade	Numérico
14	Outros planos de parcelamento	Categórico
15	Habitação	Categórico
16	Número de créditos existentes neste banco	Numérico
17	Trabalho	Categórico
18	Número de pessoas responsáveis de fornecer sustento	Numérico
19	Telefone (sim/não)	Categórico
20	Trabalhador estrangeiro	Categórico

4.1.3 Dados de Crédito Taiwanês

Este conjunto de dados refere-se a um emissor de cartão de crédito em Taiwan e os alvos são de detentores de cartão de crédito. Contém 23 atributos, sendo três atributos categóricos e todos os demais contínuos e contém uma variável binária, pagamento ausente (Sim = 1, Não = 0), como variável de rótulo. A descrição dos atributos é mostrada na Tabela 6.

Tabela 6 – Atributos no conjunto de dados Taiwanês

Atributo	Descrição	Tipo
1	Saldo Limite	Numérico
2	Sexo	Categórico
3	Educação	Categórico
4	Estado civil	Categórico
5	Idade	Numérico
6	O status de reembolso em setembro, 2005	Numérico
7	O status de reembolso em agosto, 2005	Numérico
8	O status de reembolso em julho, 2005	Numérico
9	O status de reembolso em junho, 2005	Numérico
10	O status de reembolso em maio, 2005	Numérico
11	O status de reembolso em abril, 2005	Numérico
12	Quantia no extrato de conta em setembro, 2005	Numérico
13	Quantia no extrato de conta em agosto, 2005	Numérico
14	Quantia no extrato de conta em julho, 2005	Numérico
15	Quantia no extrato de conta em junho, 2005	Numérico
16	Quantia no extrato de conta em maio, 2005	Numérico
17	Quantia no extrato de conta em abril, 2005	Numérico
18	Montante pago em setembro, 2005	Numérico
19	Montante pago em agosto, 2005	Numérico
20	Montante pago em julho, 2005	Numérico
21	Montante pago em junho, 2005	Numérico
22	Montante pago em maio, 2005	Numérico
23	Montante pago em abril, 2005	Numérico

4.2 Configuração da DNN

Em seguida são brevemente descritos os parâmetros utilizados da rede neural profunda desenvolvida neste trabalho.

4.2.1 Batch size

O Batch determina quantos exemplos de treinamento é apresentado no modelo antes que o peso e os bias sejam atualizados, ou seja utiliza-se um conjunto de exemplos de entrada, chamado de Batch. Quanto maior o tamanho do Batch, mais espaço de memória

no sistema é necessário (CANZIANI et al., 2016). Mas ele deve ser grande o suficiente para fornecer uma representação satisfatória do conjunto de treinamento em cada iteração.

4.2.2 Número de épocas

O número de épocas é o número de vezes que todo o conjunto de dados é passado através da rede neural durante o treinamento. Atualizar os pesos e outros parâmetros com apenas uma época não é suficiente, já que levaria ao subajuste (underfitting). À medida que o número de épocas aumenta, mais vezes o peso e outros parâmetros são atualizados na rede neural, e assim a precisão do treinamento e a precisão da validação aumentarão. No entanto, quando o número de épocas atinge um certo ponto, a acurácia da validação começa a diminuir, enquanto a acurácia do treinamento ainda está aumentando. Isso significa que o modelo está super ajustando (overfitting). Assim, o número ideal de épocas é o ponto em que a acurácia da validação atinge seu valor mais alto (SINHA et al., 2010). O número de épocas no nosso modelo DNN é 500.

4.2.3 Taxa de aprendizado e Otimizador

A taxa de aprendizado define a rapidez com que uma rede atualiza seus parâmetros (ZHANG, 2016). Em vez de usar uma taxa de aprendizado constante para atualizar os parâmetros (por exemplo, pesos de rede) para cada época de treinamento, foi utilizada uma taxa de aprendizado adaptável, chamado Otimizador Adadelta, que permite a especificação de diferentes taxas de aprendizado por camada. Em vez de acumular a soma dos gradientes quadrados ao longo do tempo, o Adadelta restringe-se a uma janela de gradientes anteriores que são acumulados com algum tamanho fixo. Isso garante que o aprendizado continue a progredir mesmo depois de muitas iterações de atualizações terem sido feitas.

4.2.4 Função de Ativação

A função de ativação é usada para introduzir a não linearidade na DNN. É a transformação não linear executada sobre os dados de entrada, e a saída transformada será então enviada para a próxima camada como dados de entrada. Nosso modelo usa a função de ativação Relu nas três camadas ocultas. A função de ativação Sigmóide é aplicada à camada de saída, pois é uma previsão binária (LIN et al., 2013).

4.2.5 Modo de inicialização

O modo de inicialização determina a maneira de definir os pesos aleatórios iniciais das camadas da DNN (LIN et al., 2013). Foi usado o método de Inicialização Normal para inicializar os pesos da rede e, em seguida, propagamos o peso por toda a rede. Em cada

neurônio, os pesos e os dados de entrada são multiplicados, agregados e transmitidos por meio da função de ativação.

4.3 Ambiente Computacional

A seguir, é descrita a configuração experimental em que os estudos foram conduzidos, bem como a etapa de seleção de parâmetros para os classificadores. Cada experimento é conduzido através de um processo de validação cruzada estratificada de 10 pastas em em um total de três execuções, tomando os conjuntos de dados aleatoriamente a cada vez. Com relação à seleção de parâmetros, o método de busca em grid é executado sobre cada partição de treinamento, fazendo uma validação cruzada estratificada de três pastas. Esse procedimento de validação cruzada aninhada garante um aprendizado imparcial, pois os dados do teste estão fora do processo de aprendizado. Portanto, a validação cruzada interna é usada para a seleção de parâmetros, enquanto a validação cruzada externa para avaliar o desempenho preditivo. A Figura 10 ilustra este procedimento.

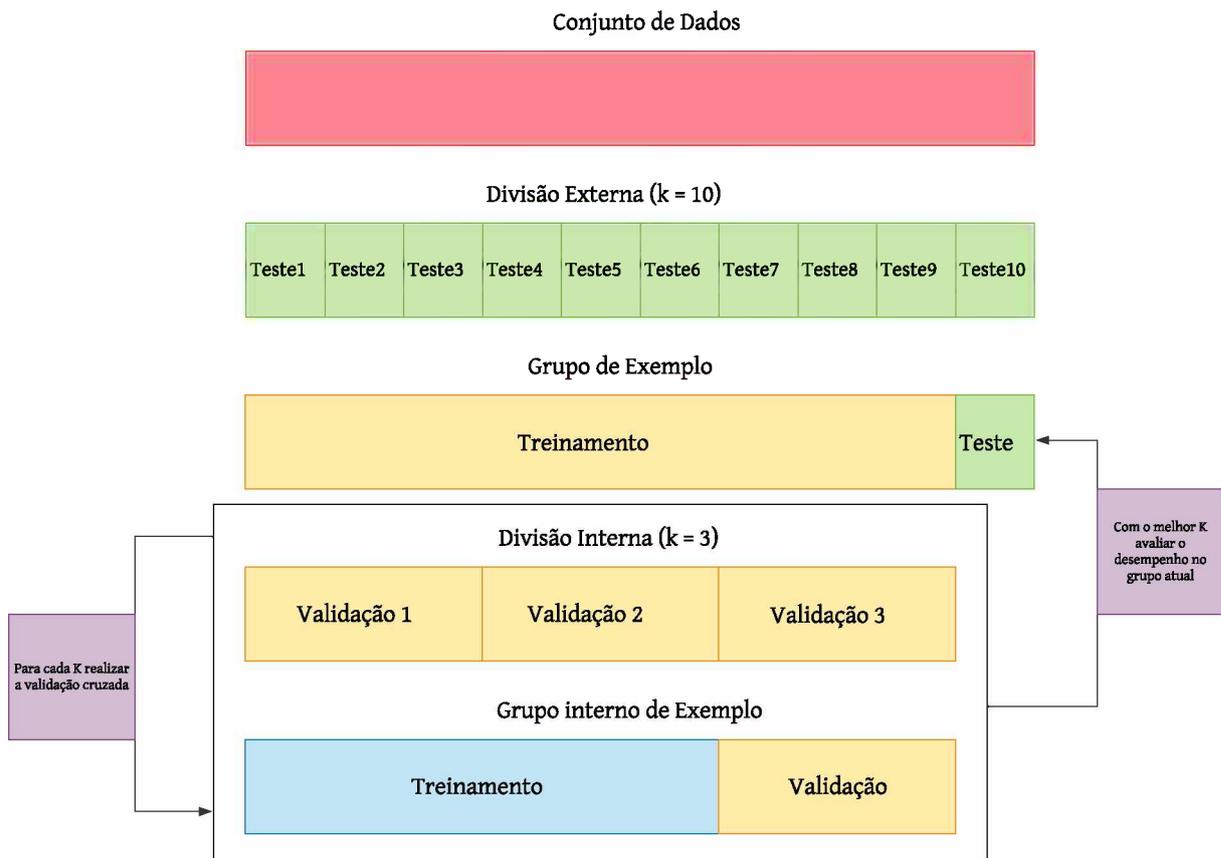


Figura 10 – Ilustração da Validação Cruzada Aninhada

Um dos componentes mais importantes da nossa configuração é o uso de validação cruzada aninhada para o treinamento e teste. Essa abordagem reduz a probabilidade de overfitting e fornece uma melhor aproximação do erro real do classificador. Neste processo,

os dados formam três conjuntos distintos: treinamento, validação e teste. O conjunto de treinamento é usado para treinar a os modelos, o conjunto de validação é usado para determinar os parâmetros, a serem introduzidos no próximo parágrafo. Quando os modelos são deduzidos, o conjunto de teste é usado para avaliar seus desempenhos.

A seleção de parâmetros é um ponto muito importante no processo de treinamento dos modelos. A busca em grid é adotada aqui para encontrar a melhor combinação de parâmetros baseada em um espaço de configurações de parâmetros pré-definidos. Neste trabalho, esse espaço está relacionado aos seguintes parâmetros para cada técnica:

- AD tem dois parâmetros, o número mínimo de amostras necessárias para dividir um nó interno $m_{split} \in 2, 5, 10, (1 * n)/100$ e o número mínimo de amostras necessárias para estar em um nó de folha $m_{leaf} \in 2, 5, 10, (1 * n)/100$, com n denotando o número de itens de dados de treinamento;
- FA tem um parâmetro, o número de árvores na floresta $t \in 2^1, 2^2, \dots, 2^{10}$;
- RL tem dois parâmetros, a norma usada na penalização $p \in l1, l2$ e a força de regularização $C \in 2^2, 2^4, \dots, 2^{14}$;
- MLP tem dois parâmetros, a taxa de aprendizado inicial $\alpha \in 0.01, 0.05, 0.1, 0.2, 0.3$ e o número de neurônios na camada oculta $n_h \in 10, 20, 50, 100, 500, 1000$.
- SVM tem dois parâmetros, o coeficiente do kernel $\gamma \in 2^4, 2^3, \dots, 2^{-10}$ e o parâmetro de penalidade $C \in 2^{12}, 2^{11}, \dots, 2^{-2}$.

5 Resultados e Discussão

Neste trabalho analisamos dados financeiros como entrada para classificar as instâncias no futuro. As categorias da instância são conceder e não conceder o crédito. Nós utilizamos seis técnicas de classificação: AD, RL, MLP, FA e SVM e DNN. Para selecionar o melhor estimador foi adotado o método de pesquisa em grid pois as técnicas mencionadas possuem diferentes parâmetros que devem ser pré-definidos para a realização do treino. Foi utilizada a validação cruzada de 10 pastas estratificada para avaliação de desempenho preditivo e dentro desse processo, também é realizada a validação cruzada de 3 pastas estratificada para seleção de parâmetros (validação cruzada aninhada). A seguir são apresentados, em termos de acurácia, os resultados obtidos.

5.1 Apresentação dos dados coletados

A Tabela 7 resume os três conjuntos de dados de análise de risco de crédito em estudo, que foram discutidos anteriormente.

Tabela 7 – Breve descrição dos conjuntos de dados da análise de risco de crédito em termos do número de itens de dados (*#Inst.*), número de atributos (*#Attr.*) e número de classes (*#Classes*).

Nome	#Inst.	#Attr.	#Classes [Distribuição]
Crédito australiano	690	14	2 [307/383]
Crédito alemão	1000	20	2 [700/300]
Crédito taiwanês	30000	23	2 [6636/23364]

5.2 Seleção de parâmetros da DNN

A Tabela 8 apresenta os resultados obtidos com diferentes parâmetros testados na DNN. O melhor resultado vem da rede de três camadas ocultas, totalizando cinco camadas totalmente conectadas, dez neurônios, e, como mencionado anteriormente, o tamanho de batch foi configurado igual a vinte, o otimizador utilizado o Adadelta, modo inicial escolhido foi o normal, função de ativação Relu (exceto a camada de saída, Sigmóide) e número de épocas igual a 500. Ao utilizar o número de neurônios igual a 50 obtivemos os piores escores nos três conjuntos de dados, independentemente se a quantidade de camadas ocultas foram duas ou três.

Tabela 8 – Resultados obtidos com diferentes parâmetros da DNN em termos de acurácia média (AM) e desvio padrão (DP) para os três conjuntos de dados em estudo

Camadas	Neurônios	Crédito Australiano		Crédito Alemão		Crédito Taiwanês	
		AM	DP+/-	AM	DP+/-	AM	DP+/-
2	10	84.65	3.42	76.40	3.01	78.13	3.90
	20	85.07	3.95	75.30	3.85	77.88	1.23
	50	83.46	5.10	72.59	2.11	72.29	1.67
3	10	85.07	2.38	75.90	4.30	78.54	6.08
	20	83.91	3.75	73.30	4.08	78.13	5.77
	50	81.87	3.61	70.09	3.33	78.20	5.42

5.3 Análise dos dados

Já a Tabela 9 apresenta a acurácia média dos seis classificadores sobre os três conjuntos de dados. Como apontado na literatura, o FA alcançou desempenho consistente para os três conjuntos de dados, retornando os melhores resultados para duas das três bases de dados. A RL, que é outro algoritmo bem recomendado para avaliação de crédito, também apresentou bom desempenho. No entanto, a AD e a SVM foram claramente superadas pela FA e pela RL, com SVM apresentando a maior complexidade de tempo entre todas as técnicas em termos de aprendizagem. A MLP demonstrou desempenho inconsistente, uma vez que alcançou o melhor resultado para o conjunto de dados da Alemanha, mas os piores resultados para conjuntos de dados da Austrália e de Taiwan.

Tabela 9 – Resultados dos classificadores em termos de acurácia média (AM) e desvio padrão (DP) para os três conjuntos de dados em estudo.

Algs	Crédito Australiano		Crédito Alemão		Crédito Taiwanês	
	AM	DP+/-	AM	DP+/-	AM	DP+/-
AD	84.48	3.60	72.03	3.92	77.46	0.62
RL	86.47	4.34	76.66	4.23	81.03	0.42
MLP	71.88	7.48	78.03	4.63	69.95	1.43
FA	87.16	4.52	76.83	3.66	81.88	0.43
SVM	72.03	3.92	76.46	3.51	77.88	0.24
DNN	85.10	2.40	75.90	4.30	78.50	0.60

A FA sobressai às demais técnicas, e uma explicação para isso é devido a sua habilidade de lidar com dados desbalanceados e ruidosos, já que ela introduz certos mecanismos para reduzir a influência do ruído e dos atributos redundantes. Por exemplo, para reduzir a influência dos atributos redundantes apresenta a seleção aleatória de atributos, para reduzir a variação e a influência do ruído combina sub-conjuntos de treinamento gerados aleatoriamente (WANG et al., 2012). Em particular, Lessmann et al. (2015) recomendam FA como referência para este problema.

Quanto à MLP, esta mostrou-se instável já que apresentou o melhor resultado para a base Alemã e o pior para as outras duas bases. Sugere-se, na literatura, que o uso de modelos de rede neural para o problema de avaliação de risco de crédito podem não ser os mais precisos (WEST, 2000) exigindo mais habilidade de modelagem para desenvolver topologias de MLP e desenvolvimento de métodos de treinamento superiores. Nesse sentido, sugere-se também o experimento de MLP em conjunto com outras técnicas de classificação.

A predominância da classe adimplentes nos exemplos de crédito é uma possível explicação para a confusão nos modelos de redes neurais, sendo mais difícil classificar clientes ruins de candidatos a crédito, devido à complexidade do risco de crédito. Segundo Yu et al. (2009) as MLP têm pior desempenho na classificação de risco de crédito do que em outras tarefas de classificação. Uma razão importante pode ser que exista um problema de overfitting ao aplicar a RNA na tarefa de classificação de risco de crédito.

A DNN não superou os métodos tradicionais de aprendizado de máquina, isso pode ter sido causado pelas limitações de números de exemplos. O desempenho da DNN poderia apresentar acurácia mais satisfatória através do possível ajuste dos parâmetros de treinamento, bem como com a alteração da quantidade de camadas ocultas.

É fácil de visualizar que o desempenho dos modelos no conjunto de dados alemão é pior do que os outros conjuntos de dados. Uma possível explicação é em termos de atributos coletados já que a base de crédito de Taiwan utiliza o histórico da conta/dívida nos últimos meses e a base Alemã refere-se a coleta de informações antes de conceder o empréstimo.

Por outro lado, os resultados para a base Australiana são em geral melhores do que para as outras bases, contudo, fica difícil explicar a razão disso já que a descrição dos atributos dessa base foram omitidas pelos pesquisadores que a disponibilizaram.

6 Conclusões

Este documento apresentou uma investigação sobre a seleção e uso de algoritmos supervisionados a fim de serem empregados para o problema de avaliação de risco de crédito. Foi codificado um ambiente computacional para a análise de desempenho de todas as técnicas sob estudo sob o ambiente Python. Os algoritmos de AD, MLP, RL, FA, SVM e DNN foram usados para criar modelos preditivos, para prever e classificar a aplicação do empréstimo como boa ou ruim usando três conjuntos de dados no domínio de crédito do repositório da UCI. Também a busca em grid foi empregada para selecionar uma configuração apropriada de parâmetros para os modelos. A validação cruzada estratificada aninhada foi utilizada para garantir um estimador livre de qualquer vício sobre os dados de teste e para fornecer uma análise comparativa confiável. Depois de aplicar técnicas de classificação, foi constatado que o melhor algoritmo para classificação de tomadores de crédito é o algoritmo de FA. O algoritmo de FA foi considerado o melhor, porque tem maior acurácia média entre as bases de dados utilizadas, como mostrado na Tabela 9. Tal resultado está de acordo com vários outros trabalhos na literatura sobre o assunto.

Além disso, adicionamos ao estado-da-arte o desenvolvimento e avaliação de uma arquitetura de rede neural profunda no contexto de finanças. Este estudo experimental em dados reais indica que as técnicas de mineração de dados de classificação são eficientes e práticas para ajudar as organizações de crédito a tomar a decisão certa de aprovar ou rejeitar as demandas de um cliente com base em suas características e histórico de pagamentos anteriores. E também é possível concluir que as técnicas de DL são pesquisas promissoras na área. Já que apesar de a mesma não ter alcançado os melhores resultados, o número de modelos de aprendizado profundo e de configurações de parâmetros a serem explorados são grandes, e acredita-se que melhores resultados poderiam ser alcançados com tal exploração.

Para pesquisas futuras sugere-se algumas modificações na arquitetura e parâmetros da DNN considerada com o objetivo de melhorar seu desempenho preditivo, além da investigação de técnicas para extração ou representação do conhecimento em redes neurais profundas.

Vários autores apontam as dificuldades de entender os mecanismos de inferência das redes neurais. E na literatura o problema de explicar as predições da rede neural tem sido abordado por técnicas que extraem regras simbólicas das redes treinadas. Essas técnicas de extração de regras de redes neurais tentam abrir a “caixa preta” da rede neural e representar o seu mecanismo de inferência a partir de regras simbólicas. Baesens et al. (2003) usaram a extração de regras de RNA para avaliar o risco de crédito com três

conjuntos de dados e afirmaram que sua abordagem foi capaz de ajudar a esclarecer as decisões das redes neurais.

Referências

- AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Database mining: A performance perspective. *IEEE transactions on knowledge and data engineering*, IEEE, v. 5, n. 6, p. 914–925, 1993. Citado na página 23.
- ALBERICIO, J. et al. Cnvlutin: Ineffectual-neuron-free deep neural network computing. *ACM SIGARCH Computer Architecture News*, ACM, v. 44, n. 3, p. 1–13, 2016. Citado na página 12.
- ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2014. Citado 4 vezes nas páginas 14, 15, 28 e 35.
- ANGELONI, M. T. Elementos intervenientes na tomada de decisão. *Ci. Inf*, SciELO Brasil, v. 32, n. 1, p. 17–22, 2003. Citado na página 21.
- ASUNCION, A.; NEWMAN, D. *UCI machine learning repository*. 2007. Citado na página 40.
- BAESENS, B. et al. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science*, INFORMS, v. 49, n. 3, p. 312–329, 2003. Citado na página 49.
- BAHRAMMIRZAEI, A. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, Springer, v. 19, n. 8, p. 1165–1195, 2010. Citado na página 35.
- BARROS, R. C. et al. A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 42, n. 3, p. 291–312, 2012. Citado na página 24.
- BELLINGER, G.; CASTRO, D.; MILLS, A. Data, information, knowledge, and wisdom. 2004. Citado 2 vezes nas páginas 21 e 22.
- BLUM, A.; HOPCROFT, J.; KANNAN, R. Foundations of data science. *Vorabversion eines Lehrbuchs*, 2016. Citado na página 15.
- BOISOT, M.; CANALS, A. Data, information and knowledge: have we got it right? *Journal of evolutionary economics*, Springer, v. 14, n. 1, p. 43–67, 2004. Citado na página 21.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 25.
- BRODLEY, C. E. et al. Challenges and opportunities in applied machine learning. *AI Magazine*, v. 33, n. 1, p. 11–24, 2012. Citado na página 17.
- CANZIANI, A.; PASZKE, A.; CULURCIELLO, E. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. Citado na página 43.

- CAO, Y. *Deep learning based RGB-D vision tasks*. Tese (Doutorado), 2018. Citado 2 vezes nas páginas 33 e 34.
- Chen, J. et al. Financial time-series data analysis using deep convolutional neural networks. In: *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. [S.l.: s.n.], 2016. p. 87–92. Citado na página 39.
- CHEN, M. et al. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, IEEE, v. 29, n. 1, 2009. Citado 2 vezes nas páginas 21 e 22.
- CHEN, M.-C.; HUANG, S.-H. Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications*, Elsevier, v. 24, n. 4, p. 433–441, 2003. Citado na página 12.
- CHEN, M.-S.; HAN, J.; YU, P. S. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, IEEE, v. 8, n. 6, p. 866–883, 1996. Citado na página 23.
- CIOS, K. J. et al. *Data mining: a knowledge discovery approach*. [S.l.]: Springer Science & Business Media, 2007. Citado na página 18.
- DENG, Y. et al. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, IEEE, v. 28, n. 3, p. 653–664, 2016. Citado na página 38.
- DOSHI, C. *A deep learning approach to state estimation from videos*. Tese (Doutorado), 01 2018. Citado 4 vezes nas páginas 6, 32, 33 e 34.
- EREMIA, M.; LIU, C.-C.; EDRIS, A.-A. Decision trees. In: _____. *Advanced Solutions in Power Systems:HVDC, FACTS, and Artificial Intelligence*. Wiley-IEEE Press, 2016. p. 1072–. ISBN 9781119175391. Disponível em: <<https://ieeexplore-ieee-org.ez34.periodicos.capes.gov.br/xpl/articleDetails.jsp?arnumber=7656752>>. Citado na página 24.
- FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37, 1996. Citado na página 18.
- FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, ACM, v. 39, n. 11, p. 27–34, 1996. Citado na página 18.
- FAYYAD, U. M. Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications*, IEEE Educational Activities Department, v. 11, n. 5, p. 20–25, 1996. Citado 5 vezes nas páginas 6, 18, 19, 20 e 23.
- FAYYAD, U. M. et al. Knowledge discovery and data mining: Towards a unifying framework. In: *KDD*. [S.l.: s.n.], 1996. v. 96, p. 82–88. Citado 2 vezes nas páginas 17 e 18.
- FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. Knowledge discovery in databases: An overview. *AI magazine*, v. 13, n. 3, p. 57, 1992. Citado 3 vezes nas páginas 17, 18 e 22.
- GALINDO, J.; TAMAYO, P. Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. *Computational Economics*, Springer, v. 15, n. 1-2, p. 107–143, 2000. Citado na página 17.

- HAND, D. J.; HENLEY, W. E. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, Wiley Online Library, v. 160, n. 3, p. 523–541, 1997. Citado 2 vezes nas páginas 12 e 35.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: data mining, inference, and prediction*, Springer Series in Statistics. [S.l.]: Springer New York, 2009. Citado 3 vezes nas páginas 25, 26 e 27.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Unsupervised learning. In: *The elements of statistical learning*. [S.l.]: Springer, 2009. p. 485–585. Citado na página 16.
- HATZIARGYRIOU, N.; VOUMVOULAKIS, M. Unsupervised learning and hybrid methods. *Advanced Solutions in Power Systems: HVDC, FACTS, and Artificial Intelligence: HVDC, FACTS, and Artificial Intelligence*, Wiley Online Library, p. 985–1032, 2016. Citado na página 16.
- HORNICK, M. F.; MARCADÉ, E.; VENKAYALA, S. *Java data mining: strategy, standard, and practice: a practical guide for architecture, design, and implementation*. [S.l.]: Elsevier, 2010. Citado na página 14.
- HUANG, C.-L.; CHEN, M.-C.; WANG, C.-J. Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, Elsevier, v. 33, n. 4, p. 847–856, 2007. Citado na página 36.
- KALAVATHY, R.; SURESH, R.; AKHILA, R. Kdd and data mining. In: IET. *Information and Communication Technology in Electrical Sciences (ICTES 2007)*, 2007. *ICTES. IET-UK International Conference on*. [S.l.], 2007. p. 1105–1110. Citado na página 18.
- KHASHMAN, A. Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications*, Elsevier, v. 37, n. 9, p. 6233–6239, 2010. Citado na página 36.
- KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, v. 160, p. 3–24, 2007. Citado 2 vezes nas páginas 16 e 24.
- KOVÁCS, Z. L. *Redes neurais artificiais*. [S.l.]: Editora Livraria da Física, 2002. Citado na página 26.
- KRUPPA, J. et al. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, Elsevier, v. 40, n. 13, p. 5125–5131, 2013. Citado 2 vezes nas páginas 12 e 35.
- KYAN, M. et al. *Unsupervised Learning: A Dynamic Approach*. [S.l.]: John Wiley & Sons, 2014. Citado na página 17.
- LANGLEY, P.; SIMON, H. A. Applications of machine learning and rule induction. *Communications of the ACM*, ACM, v. 38, n. 11, p. 54–64, 1995. Citado na página 15.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Citado 2 vezes nas páginas 11 e 31.

- LEE, H. et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ACM. *Proceedings of the 26th annual international conference on machine learning*. [S.l.], 2009. p. 609–616. Citado na página 31.
- LESSMANN, S. et al. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, Elsevier, v. 247, n. 1, p. 124–136, 2015. Citado na página 47.
- LIAO, Z. *Methods for Understanding and Improving Deep Learning Classification Models*. Tese (Doutorado), 2017. Citado na página 29.
- LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. Citado na página 43.
- LIN, W.-Y.; HU, Y.-H.; TSAI, C.-F. Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 42, n. 4, p. 421–436, 2012. Citado na página 35.
- LINOFF, G. S.; BERRY, M. J. *Data mining techniques: for marketing, sales, and customer relationship management*. [S.l.]: John Wiley & Sons, 2011. Citado 2 vezes nas páginas 18 e 27.
- MALEKIPIRBAZARI, M.; AKSAKALLI, V. Risk assessment in social lending via random forests. *Expert Systems with Applications*, Elsevier, v. 42, n. 10, p. 4621–4631, 2015. Citado na página 35.
- MCCUE, C. *Data mining and predictive analysis: Intelligence gathering and crime analysis*. [S.l.]: Butterworth-Heinemann, 2014. Citado na página 16.
- MUHAMMAD, I.; YAN, Z. Supervised machine learning approaches: A survey. *ICTACT Journal on Soft Computing*, v. 5, n. 3, 2015. Citado na página 16.
- OLARU, C.; WEHENKEL, L. Data mining. *IEEE Computer Applications in Power*, v. 12, n. 3, p. 19–25, Jul 1999. ISSN 0895-0156. Citado 2 vezes nas páginas 18 e 23.
- QUINLAN, J. R. Induction of decision trees. *Machine learning*, Springer, v. 1, n. 1, p. 81–106, 1986. Citado na página 24.
- RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Malaysia; Pearson Education Limited,, 2016. Citado na página 16.
- SALINAS, E.; SEJNOWSKI, T. J. Correlated neuronal activity and the flow of neural information. *Nature reviews neuroscience*, Nature Publishing Group, v. 2, n. 8, p. 539, 2001. Citado na página 30.
- SANCHES, M. K. *Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados*. Tese (Doutorado) — Universidade de São Paulo, 2003. Citado na página 17.
- SETZER, V. W. Dado, informação, conhecimento e competência. *DataGramZero Revista de Ciência da Informação*, n. 0, 1999. Citado na página 21.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding machine learning: From theory to algorithms*. [S.l.]: Cambridge university press, 2014. Citado 2 vezes nas páginas 14 e 15.

- SIHAG, S.; DUTTA, P. K. Faster method for deep belief network based object classification using dwt. *arXiv preprint arXiv:1511.06276*, 2015. Citado na página 31.
- SILVA, J. P. da. *Gestão e análise de risco de crédito*. [S.l.]: Editora Atlas SA, 2008. Citado 2 vezes nas páginas 11 e 12.
- SINHA, S. et al. Epoch determination for neural network by self-organized map (som). *Computational Geosciences*, Springer, v. 14, n. 1, p. 199–206, 2010. Citado na página 43.
- SKREDE, O. S. L. *Inter-/Intra-session Recurrent Neural Network for Session-based Recommender Systems*. Dissertação (Mestrado) — NTNU, 2017. Citado na página 32.
- SRIVASTAVA, A. et al. Parallel formulations of decision-tree classification algorithms. In: *High Performance Data Mining*. [S.l.]: Springer, 1999. p. 237–261. Citado na página 23.
- SUN, T.; VASARHELYI, M. A. Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management*, Wiley Online Library, v. 25, n. 4, p. 174–189, 2018. Citado na página 39.
- SUYKENS, J. A.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural processing letters*, Springer, v. 9, n. 3, p. 293–300, 1999. Citado na página 26.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introdução ao datamining: mineração de dados*. [S.l.]: Ciência Moderna, 2009. Citado 8 vezes nas páginas 11, 12, 18, 20, 23, 24, 25 e 28.
- TOULoupAS, G. *Deep Learning for Analyzing Atmospheric Clouds*. Dissertação (Mestrado) — ETH Zurich, 2018. Citado 2 vezes nas páginas 29 e 30.
- TSAI, C.-F. Feature selection in bankruptcy prediction. *Knowledge-Based Systems*, Elsevier, v. 22, n. 2, p. 120–127, 2009. Citado na página 37.
- TSAI, C.-F.; WU, J.-W. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert systems with applications*, Elsevier, v. 34, n. 4, p. 2639–2649, 2008. Citado 2 vezes nas páginas 27 e 38.
- Vapnik, V. N. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, v. 10, n. 5, p. 988–999, Sep. 1999. ISSN 1045-9227. Citado na página 25.
- WANG, G. et al. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, Elsevier, v. 26, p. 61–68, 2012. Citado na página 47.
- WANG, L. *Support vector machines: theory and applications*. [S.l.]: Springer Science & Business Media, 2005. v. 177. Citado na página 25.
- WEI, G.; YINGJIE, S.; MU, Y. X. Commercial bank credit risk evaluation method based on decision tree algorithm. In: IEEE. *Measuring Technology and Mechatronics Automation (ICMTMA), 2015 Seventh International Conference on*. [S.l.], 2015. p. 285–288. Citado na página 11.
- WEST, D. Neural network credit scoring models. *Computers & Operations Research*, Elsevier, v. 27, n. 11-12, p. 1131–1152, 2000. Citado na página 48.

- YEH, I.-C.; LIEN, C.-h. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, Elsevier, v. 36, n. 2, p. 2473–2480, 2009. Citado na página 35.
- YU, L.; WANG, S.; CAO, J. A modified least squares support vector machine classifier with application to credit risk analysis. *International Journal of Information Technology & Decision Making*, World Scientific, v. 8, n. 04, p. 697–710, 2009. Citado na página 48.
- ZHANG, B. *Deep learning with application to hashing*. Tese (Doutorado), 2014. Citado 4 vezes nas páginas 29, 30, 31 e 32.
- ZHANG, D.; ZHOU, L. Discovering golden nuggets: data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 34, n. 4, p. 513–522, 2004. Citado na página 11.
- ZHANG, J. *Deep learning for multi-label scene classification*. Tese (Doutorado), 2016. Citado 2 vezes nas páginas 33 e 43.
- ZHU, X. Semi-supervised learning literature survey. Citeseer, 2005. Citado na página 17.
- ZINS, C. Conceptual approaches for defining data, information, and knowledge. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 58, n. 4, p. 479–493, 2007. Citado na página 21.