

# Predição de Pagamento de Cartão de Crédito com Uso de Árvore de Decisão e Rede Neural.

Augusto S. L. Carvalho\*, Jês J. F. Cerqueira\*\*, Eduardo F. S. Filho\*\*\*

Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal da Bahia

\*Secretaria de Educação do Distrito Federal (e-mail: augusto.carvalho@edu.se.df.gov.br)

\*\* (e-mail: jes@ufba.br).

\*\*\* (e-mail: eduardo.simas@ufba.br)

---

**Abstract:** In this article, an analysis is made of the use of two methodologies for credit card payment prediction classification, verifying a payment probability. Among the various existing Data Mining techniques, Decision Trees and Neural Networks were used, observing the most appropriate method for this type of problem, in which users are classified based on 23 available existing attributes. For analysis, first of all, it were used datas without pre-processing and then with pre-processing, using the following techniques: Clustering, Principal Component Analysis and Standardization, checking which is the best fit for the proposed credit granting problem.

**Resumo:** Neste artigo, faz-se uma análise do uso de duas metodologias para classificação de predição de pagamento com cartão de crédito, verificando uma probabilidade de pagamento. Entre as diversas técnicas de Mineração de Dados existentes, foram usadas as Árvores de Decisão e as Redes Neurais, observando o método mais adequado para esse tipo de problema, em que usuários são classificados com base em 23 espécies disponíveis. Para análise, primeiro, uso de dados sem pré-processamento e, em seguida, com pré-processamento, por meio das seguintes técnicas: Análise de Componentes Principais e Normalização, verificando qual é a melhor adequação ao problema de concessão de crédito proposto.

**Keywords:** Attributes, Classification, Credit Card, Decision Tree, Neural Networks.

**Palavras-chaves:** Árvore de Decisão, Atributos, Cartão de Crédito, Classificação, Rede Neural.

---

## 1. INTRODUÇÃO

Neste trabalho tem por objetivo fazer a análise do uso de duas metodologias para a classificação de predição de pagamento de cartão de crédito. Para isso, foram utilizadas, entre as diversas técnicas de Mineração de Dados existentes, a Árvores de Decisão e a Rede Neural. Além disso, foi realizada uma revisão bibliográfica sobre o assunto, foram apresentados os materiais e métodos necessários para o desenvolvimento da pesquisa e os resultados obtidos com suas devidas análises diante dos experimentos com e sem processamento dos dados utilizados para a classificação.

Com isso, pretende-se contribuir com a comunidade científica comparando o uso de duas metodologias de classificação de créditos bancários para clientes, classificando-os como adimplentes ou inadimplentes, utilizando Árvores de Decisão e Redes Neurais e destacando a importância do pré-processamento dos dados para obtenção de resultados mais eficientes. Para isso, será feito o experimento com e sem pré-processamentos dos dados, verificando e comparando os resultados e o grau de confiança obtidos.

O cartão de crédito é um instrumento que permite que o usuário pague compras e serviços sem precisar estar com

dinheiro no momento da transação, deixando para efetuar o pagamento em um momento futuro, na data do vencimento (GOVERNO FEDERAL, 2012). Para adquirir um cartão, as corretoras fazem uma análise de crédito do usuário, levando em consideração algumas variáveis, as quais classificam o usuário como bom pagador ou não. Esse artigo tem como base a implementação dessa análise, utilizando Mineração de Dados.

A Mineração de Dados é uma metodologia para melhorar a qualidade e a eficiência das decisões, com estratégias baseadas nos dados históricos de um determinado problema que se quer trabalhar (PREZEPIORSKI et al, 2005).

Predição de pagamento de cartão de crédito é tipicamente um problema de detecção de erro na concessão de crédito e o objetivo da análise é mapear um espaço de atributos de entrada para um espaço de uma classe de saída, atribuindo uma classificação (ALESKEROV et al, 1997).

As informações dadas do banco de dados encontram-se balanceadas, ou seja, sem dados faltantes ou em excesso, logo este experimento faz uma comparação entre o desempenho da Árvores de Decisão e da Rede Neural, sem pré-processamento e com pré-processamento dos dados, aplicando a técnica de Análise de Componentes Principais (PCA); em seguida, efetua

a normalização dos dados e realiza comparações dos resultados obtidos em todas as etapas.

No confronto entre as duas técnicas de mineração de dados (Árvores de Decisão e Rede Neural), observando o comportamento de ambas e analisando a necessidade de se pré-processar a informação, serão inseridas, como o diferencial deste trabalho, as técnicas estatística de pré-processamento de dados, verificando a sua performance nos resultados e a melhor aplicação de pré-processamento para este tipo de informação.

## 1.1 Apresentação

Neste artigo, será feito um referencial teórico sobre os modelos de mineração de dados: Árvore de Decisão e Rede Neural; a técnica de pré-processamento Normalização e Análise de Componentes Principais e; o banco de dados utilizado no experimento.

Em seguida, será explicada a metodologia utilizada no experimento, comparando os resultados com o modelo de Árvore de Decisão e Rede Neural sem e com pré-processamento.

Por fim, na conclusão, será realizada uma análise geral dos modelos implementados, relatando as suas aplicabilidades e eventuais riscos para as empresas na concessão do crédito.

## 2. REFERENCIAL TEÓRICO

A seguir, nas subseções, será apresentado o referencial teórico com informações relevantes para esta pesquisa científica na predição de pagamento de cartão de crédito com uso de Árvore de Decisão e Rede Neural, destacando os dois modelos de pré-processamentos utilizados (Normalização e Análise de Componentes Principais – PCA).

### 2.1 Árvores de Decisão

As Árvores de Decisão são modelos estatísticos que empregam um treinamento supervisionado para classificação e previsão de dados. Esses modelos utilizam a estratégia de dividir para conquistar: um problema complexo é decomposto em subproblemas mais simples, sendo essa técnica aplicada a cada subproblema (DE PAULA, 2002). Esse é um dos métodos mais populares, pois constrói uma árvore binária, dividindo os registros em cada nó de acordo com a função de um campo de entrada (M. ZEKIC-SUSAC et al., 2004).

Uma Árvore de Decisão é normalmente construída recursivamente de maneira descendente. Se um conjunto de instâncias rotuladas for relativamente puro, a árvore terá apenas uma folha, sendo regida pela classe que ocorre com maior frequência nesse conjunto de dados (SOLTAN & MOHAMMAD, 2004).

A capacidade de discriminação de uma árvore vem da divisão do espaço definido pelos atributos em subespaços, que são associados a uma classe. A Árvore de Decisão leva em consideração os atributos mais relevantes, que são compreensíveis para as pessoas, apresentando-os em ordem de importância (DE PAULA, 2002), logo, as Árvores de Decisão

podem ser representadas como conjuntos de regras do tipo se-então (if-then) seguindo o trajeto do nó raiz até uma folha da árvore.

### 2.1.1 Algoritmo da Árvore de Decisão

O algoritmo usado na Árvore de Decisão será o J48, o qual classifica uma nova instância a partir dos valores do conjunto de atributos do treinamento fornecido. No momento em que o algoritmo se depara com o conjunto em treinamento, ele reconhece o atributo responsável por categorizar as várias instâncias, atribuindo valores sem ambiguidade ao ramo de interesse (PANIGRAHIA, 2018) na Árvore de Decisão. No entanto, esse algoritmo é bastante suscetível aos ruídos (HANG, 2018) e à falta de dados, mas, mesmo assim, obtém-se resultados de alto desempenho com baixo custo computacional. (VIEIRA et al., 2018).

Além disso, o algoritmo J48 é a recodificação de um outro algoritmo, o C4.5, que é escrito em C, sendo recodificado para a linguagem JAVA. O J48 tem a finalidade de gerar uma Árvore de Decisão baseada em um conjunto de dados de treinamento, sendo esse modelo usado para classificar as instâncias no conjunto de teste. A principal característica do J48 é apresentar o melhor resultado na montagem de Árvores de Decisão, a partir de um conjunto de dados de treinamento, com a abordagem de dividir-para-conquistar.

O J48, também, tem a propriedade de lidar tanto com atributos contínuos e discretos, com valores categóricos e valores ausentes. O tratamento dos atributos contínuos leva em consideração todos os valores do conjunto de treinamento, ordenando de forma crescente e considerando todos os valores dos dados de treinamento. Isso é favorável para a redução da entropia, sendo o cálculo para o ganho realizados da seguinte forma (ARAUJO, 2018):

### 2.1.2 Razão do ganho da Árvore de Decisão

Para se efetuar o cálculo de ganho na Árvore de Decisão necessita-se reduzir a entropia, que são impurezas de um determinado atributo, buscando a homogeneidade dos dados e estimando a diferença do valor da informação esperada pelo valor real.

Redução de Entropia, com  $info(S)$  identificando a classe no conjunto de treinamento:

$$info(S) = - \sum_{i=1}^h \left[ \left[ \frac{freq(C_i, S)}{|S|} \right] \log_2 \left[ \frac{freq(C_i, S)}{|S|} \right] \right] \quad (1)$$

$|S|$  é o número de casos no conjunto de treinamento;  $C_i$  é as variáveis do problema;  $k$  é o número de classes e  $freq(C_i, S)$  é o número de casos em  $C_i$ .

Redução de Entropia, calculando o valor da informação esperada,  $Info_x(S)$ , em que  $S$  é o conjunto de exemplo de treino para uma variável  $X$ .

$$info_x(S) = - \sum_{i=1}^n \left[ \left[ \frac{S_i}{|S|} \right] info(S_i) \right] \quad (2)$$

Logo o ganho da Árvore de Decisão será:

$$\text{Ganho}(X) = \text{info}(S) - \text{info}(S_i) \quad (3)$$

Por fim, o algoritmo gera uma razão do ganho (RG) da informação do atributo, normalizado pela medida de informação.

$$RG = \frac{\text{Ganho}}{\sum_{i=1}^h \left[ \frac{|S_i|}{|S|} \log_2 \left[ \frac{|S_i|}{|S|} \right] \right]} \quad (4)$$

## 2.2 Redes Neurais

As Redes Neurais artificiais são sistemas de processamento de informações distribuídas, compostos por elementos computacionais que interagem por meio de conexões contendo pesos distintos com componentes de processamento altamente interconectados, levando em consideração técnicas neurobiológicas (BAESENS et al, 2003). Com base na arquitetura do cérebro humano, as Redes Neurais possuem uma característica de aprender padrões complexos de dados e generalizar a informação aprendida (ZHANG, 1998).

Na figura 1, é apresentado o modelo de um neurônio que forma a base para o projeto de Redes Neurais Artificiais, podendo destacar os seguintes componentes: um conjunto de sinapses caracterizado por um peso, tendo um sinal  $X_i$  na entrada da sinapse conectado a um neurônio  $k$ ; em seguida, é multiplicado pelo peso sináptico  $W_{kj}$ . Na etapa seguinte, tem-se um somador ou integrador para somar os sinais de entrada, ponderado pelas respectivas sinapses do neurônio. Por fim, uma função de ativação para restringir a amplitude da saída do neurônio.

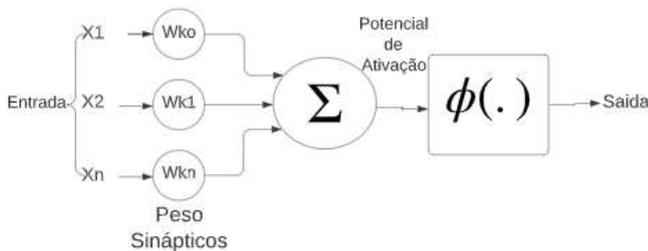


Fig. 1 Modelo de Um Neurônio Artificial (Fonte: Próprio autor).

Cada elemento não linear na Rede Neural é chamado de nó e é densamente interconectado por meio de conexões diretas. Os nós possuem características análogas a uma função de dois estágios. Primeiramente, calcula-se a soma dos sinais de entrada, atribuindo pesos aos sinais e, em seguida, aplica-se à soma dos sinais uma função de saída, chamada de função de ativação (LIMA et al, 2009).

O modelo neural da figura 1 inclui um viés representado por  $b_k$ , tendo a função de aumentar ou diminuir a entrada da função de ativação, dependendo se ele é positivo ou negativo, respectivamente, gerando a equação em um neurônio  $k$ :

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (5)$$

$$y_k = \phi(u_k + b_k) \quad (6)$$

Onde  $x_j$  são os sinais de entrada;  $w_{kj}$  são os pesos sinápticos do neurônio  $k$ ;  $u_k$  é a saída do combinador devido aos sinais de entrada;  $b_k$  o viés;  $\phi(\cdot)$  é a função de ativação; e  $y_k$ , o sinal de saída do neurônio.

Modelos de redes lineares têm a propriedade em soluções mais simples e fechadas como um problema de regressão, minimizando a diferença quadrática entre valores previstos, mas esse modelo pode não ser preciso para aplicação em mercados financeiros, devido às variáveis não lineares dos dados por conta das oscilações e instabilidade do mercado econômico (LIMA et al, 2009), que também abrange os consumidores, fator que poderá ser analisado durante a realização desse experimento.

Nesse contexto, a Neste trabalho foi utilizada uma rede do tipo *multilayerperceptron* (MLP), pois suas camadas intermediárias possibilitam a aproximação de funções contínuas lineares e não lineares das entradas, sendo algoritmo de treinamento implementado o *backpropagation*.

Esse método consiste em um aprendizado supervisionado, em que são fornecidos os valores de entrada e saída. Ele é realizado em duas etapas: a *forward* e a *backward*. Na etapa *forward*, a rede calcula os valores da saída a partir dos dados de entrada. Na fase seguinte, a *backward*, são utilizados os pesos associados a cada conexão de acordo com as diferenças entre as saídas obtidas e as saídas desejadas, conforme discutido anteriormente.

## 2.3 Normalização

A distorção entre valores do banco de dados pode variar bastante comprometendo o processamento da Árvore de Decisão e da Rede Neural, tendenciado a análise, pois considera valores maiores de atributos com mais relevância, já que possui uma maior variação em torno da média (COELHO, 2011).

A Normalização de dados age reescalando os dados por atributos, sendo uma boa escolha quando se sabe que a distribuição dos dados não é normal/gaussiana ou quando não se sabe qual é o tipo de distribuição dos dados (COELHO, 2011). A função utilizada por esse método no software Matlab (NORMALIZE, 2019) para normalização é:

$$\text{Valor Novo} = \frac{(\text{Valor Atual} - \text{Média})}{\text{Desvio Padrão}} \quad (7)$$

## 2.4 Análise de Componentes Principais - PCA

A Análise de Componentes Principais (PCA) é um método estatístico de múltiplas variáveis, que considera a transformação linear ótima, reconhecendo relações entre características extraídas dos dados e encontrando autovalores e autovetores da matriz de covariância. Tal método pode reduzir o custo computacional com a redução dimensional dos atributos dos dados em análise, ou seja, é uma técnica que transforma diversos atributos com uma certa correlação em um único atributo. Esse método deve ser aplicado apenas nos casos em que os dados analisados possuem muitos atributos, como nesse experimento que são 23, uma vez que no PCA qual

sempre haverá perda de informações com a “junção” de atributos correlacionados (COELHO, 2019). Sua aplicação será para observar o comportamento da Árvore de Decisão e da Rede Neural diante da redução dos atributos, verificando o impacto dessa técnica.

### 2.5 Banco de Dados

O Banco de Dados utilizado no problema foi o repositório do site UCI, *Machine Learning Repository* (DEFAULT, 2016), com informações de cartões de crédito de clientes de Taiwan, na China. O conjunto de dados é multivariado, possuindo 30.000 instâncias, onde 23.364 são de adimplentes e 6.636 são de inadimplentes, além de atributos inteiros, reais e binário, sendo, no total, 24 atributos, dos quais 23 são características de usuários, como mostrado na Tabela 1 e um atributo de classificação entre 0 (adimplente) e 1 (inadimplente). Não há pré-processamento aplicado aos dados utilizado neste experimento, e o mesmo, foi inserido no repositório no dia 26 de janeiro de 2016.

Esse banco de dados foi selecionado por se tratar de procedimentos financeiros, recorrentes no dia a dia do consumidor, podendo ser útil para empresas que pretendem discutir e/ou usar técnicas de mineração de dados para tratar informações de seus clientes ou para estudantes que queiram entender o processamento de informações com Árvore de Decisão e Rede Neural, utilizando um banco de dados usual no cotidiano dos consumidores, tornando a contextualização do conteúdo mais dinâmico para discernimento das questões abordadas.

**Tabela 1. Relação das instâncias do problema.**

Variáveis	Referência
X1	Montante do crédito concedido
X2	Sexo (1 = masculino; 2 = feminino)
X3	Educação (1 = pós-graduação; 2 = universidade; 3 = ensino médio; 4 = outros)
X4	Estado civil (1 = casado; 2 = solteiro; 3 = outros)
X5	Idade (ano)
X6 – X11	Histórico de pagamentos passados
X12 – X17	Valor do extrato da fatura
X18 – X23	Valor do pagamento anterior (dólar)

### 3. MATERIAIS E MÉTODOS

Nas subseções, serão apresentados os materiais e métodos necessários para o desenvolvimento desta pesquisa.

Este experimento, inicialmente, utilizará métodos de mineração de dados (Árvore de Decisão e Rede Neural). Os dados da seção 2.5, serão processados de três maneiras, como mostrado na figura 2: inicialmente sem o pré-processamento; em seguida, utilizando a Árvore de Decisão e a Rede Neural, coletando os dados da eficácia desta aplicação; e, por fim, os dados, serão processados, usando primeiro a técnica de pré-processamento PCA e, posteriormente, a Normalização, também coletando resultados e verificando a eficácia dessas

técnicas, comparando os seis resultados obtidos nos experimentos.

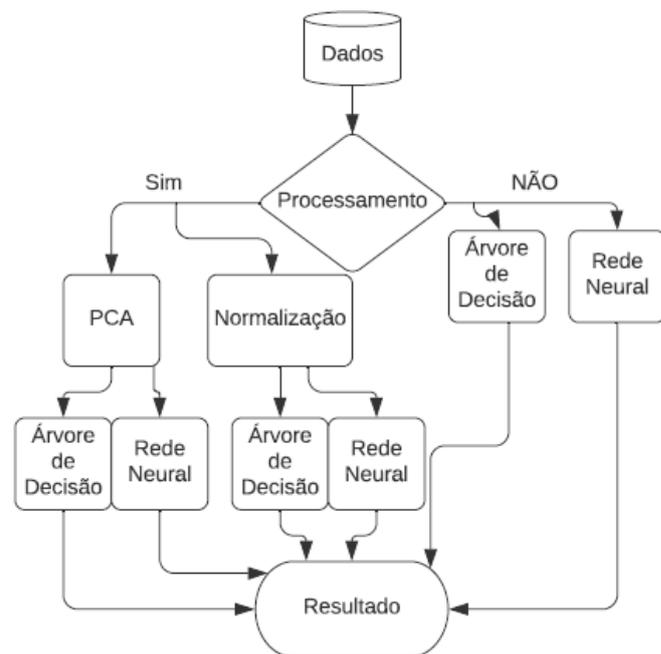


Fig. 2 Fluxograma do processo de análise de dados (Fonte: autoria própria)

No fluxograma do processo é possível observar que existirão dois resultados para a etapa sem pré-processamento (um para Árvore de Decisão e outro para a Rede Neural), além de quatro resultados para a etapa com pré-processamento (dois com o uso da função de Normalização do Matlab e dois com o uso da função PCA do Matlab), sendo que cada análise será processada também pela Árvore de Decisão e pela Rede Neural

#### 3.1 Árvore de Decisão

Na implementação da técnica Árvore de Decisão foi utilizado o software computacional WEKA (*Waikato Environment for Knowledge Analysis*), por sua praticidade de utilização e por ser um software domínio público (WEKA, 2019). O software WEKA é formado por um conjunto de diversas técnicas para resolver problemas concretos de Mineração de Dados. Ele é implementado em linguagem Java e foi desenvolvido na Universidade de Waikato, na Nova Zelândia, em 1999.

Para esse trabalho, na técnica de Árvore de Decisão, foi utilizado o Banco de Danos apresentado no item 2.5, empregando na execução o algoritmo de classificação J48 (C45 release 8) composto por informações de 23.364 adimplentes e 6.636 inadimplentes, considerando-se para cada uma das informações as 24 características apresentadas.

#### 3.2 Rede Neural

A Rede Neural utilizada foi a *Perceptron*, elaborada com o uso da ferramenta MATLAB (*MATrixLABoratory*), que se utiliza

de uma rede de *feed-forward* de duas camadas, dividindo-as em conjuntos de treinamento, validação e teste, com neurônios *sigmóides* ocultos e um *softmax* de saída. A rede foi treinada com *backpropagation* de gradiente conjugado em escala (NEURAL, 2019) e, após o treinamento, foi avaliado o desempenho usando entropia cruzada, analisando a porcentagem de erro de classificação incorreta e as matrizes de confusão para cada etapa, seja sem ou com pré-processamento de dados.

Na ferramenta *MATLAB*, para a programação da Normalização e Análise de Componentes Principais, foi usada a função *normalize* e a função *pca*, respectivamente.

#### 4. RESULTADOS E DISCUSSÃO

A seguir, nas subseções, serão apresentados os resultados obtidos com suas devidas discussões diante dos experimentos realizados.

##### 4.1 Sem pré-processamento dos dados

Inicialmente, serão apresentados os dados sem pré-processamento verificando os devidos resultados e discussões.

###### 4.1.1 Árvore de Decisão

Na implementação da técnica Árvore de Decisão, foi utilizada a base de dados composta por informações de 23.364 clientes adimplentes e 6.636 inadimplentes, considerando-se para cada uma das árvores 23 características apresentadas na seção 2.5 deste artigo. Observado o desempenho é possível analisar que as decisões tomadas ao longo da árvore passam pelo histórico de pagamento, como elemento de escolha entre a classificação de adimplência e de inadimplência.

A Árvore de Decisão gerada após o treino obteve uma classificação incorreta em torno de 16,6%, dado que será confrontado com o da Rede Neural, observando qual possui uma melhor aplicação para esse tipo de problema.

###### 4.1.2 Rede Neural

O treinamento foi realizado por meio de uma rede de múltiplas camadas, usando o algoritmo de *backpropagation* padrão, com: 15% para teste, que é para o ajuste da rede durante o treinamento, ocorrendo de acordo com seu erro; 15% para validação, que são usados para medir a generalização da rede e interromper o treinamento quando a generalização parar de melhorar; e 70% para treino, que fornece uma medida independente do desempenho da rede durante e após o treinamento.

A quantidade de neurônios utilizados foram: 2, 6, 10, 20 e 40 para testes e comparações na camada intermediária, como mostrando na Figura 4.

O treinamento para dois neurônios na camada oculta foi interrompido na iteração (época) 55, que acontece quando o erro de validação aumentou por seis iterações; já para seis neurônios ocorreu a interrupção na época 90; e no último teste, com 40 neurônios, parou na época 78.

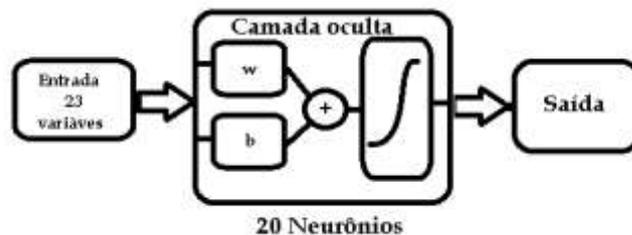


Fig. 3 Composição da Rede *Perceptron* com 20 neurônios na camada oculta. (Fonte: autoria própria)

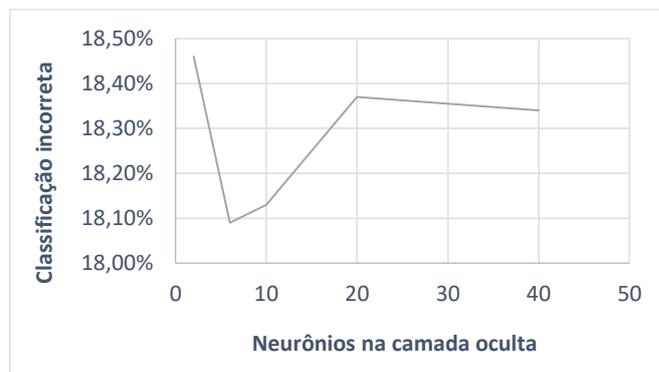


Fig. 4 Instâncias classificadas incorretamente por número de neurônios na camada oculta.

No experimento, observou-se que, à medida que aumenta o número de neurônios, melhora a performance da rede, porém, esse valor possui um limiar próximo de 6, haja vista que com 10 neurônios o erro aumentou. Foram realizados, também, outros testes entre 6 e 40, que não tiveram desempenho satisfatório, em relação com o que tinha sido coletado e, por esse motivo, não foi adicionado à tabela. Esse experimento não teve o intuito de procurar o número de neurônios ideal para a camada oculta, mas apenas ver o comportamento da rede diante do problema proposto.

###### 4.1.3. Matriz de Confusão

A matriz de confusão é uma tabela binária, que possui duas possibilidades, verdadeiro ou falso; no entanto, existem classificações que podem ser feitas erroneamente. A Tabela 2 e 3 mostram as frequências de classificação para cada classe do modelo, podendo ser: Verdadeiro Positivo, quando no conjunto real que se está avaliando, a classe foi prevista corretamente; Falso Positivo, quando a classe que estamos buscando prever foi prevista incorretamente, gerando uma classificação errada para classe; Verdadeiro Negativo, que é quando a classe que não estamos buscando prever a classificação prevista foi corretamente; ou Falso Negativo, que ocorre quando a classe que não estamos buscando prever foi incorretamente classificada.

Nos dois softwares foi obtida a Matriz de Confusão. Além disso, é possível observar que a Árvore de Decisão obteve um resultado mais expressivo em relação à Rede *Perceptron* com seis neurônios na camada oculta, que foi a melhor classificada entre as cinco redes analisadas, levando vantagem também

quando são feitas classificações equivocadas, como o Falso Positivo e o Falso Negativo.

**Tabela 2. Matriz de confusão da Árvore de Decisão.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	2823	3813
Adimplentes	1167	22197

Quando analisadas as instâncias classificadas incorretamente, observa-se que a Árvore de Decisão leva uma vantagem de 1,49 pontos percentuais em relação à Rede Neural com seis neurônios ocultos, classificando com melhor precisão os dados de adimplentes.

**Tabela 3. Matriz de confusão da Rede Neural.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	2440	4190
Adimplentes	1190	22174

Na comparação entre as matrizes de confusão, observa-se que a Arvore de Decisão obteve um melhor resultado em relação à Rede Neural, classificando com maior precisão os adimplentes que são adimplentes e os inadimplentes que são inadimplentes.

#### 4.2 Com pré-processamento dos dados

Abaixo serão apresentados os dados com pré-processamento, verificando os devidos resultados e discussões.

##### 4.2.1 PCA

Aplicando a função PCA do Matlab é possível gerar um novo dado. Os dados correlacionados foram reorganizados pelo próprio *Matlab*, resultando apenas 13 atributos, em função da disposição espacial dos dados, sendo observada a porcentagem de variação dos dados no espaço das variáveis. A classificação será analisada com a Árvore de Decisão e a Rede Neural.

##### 4.2.1.1 Árvore de Decisão

Com o uso da Árvore de Decisão, observou que o erro de classificação cresceu com o uso do PCA chegando a 21,5 %, observado no *WEKA*. Na Matriz de Confusão é possível verificar que os valores minoritários (inadimplentes) foram os que mais sofreram redução percentual com o uso dessa ferramenta.

Analisando a Tabela 4, verifica-se que apenas 2804 de 6.636 inadimplentes foram classificados corretamente, valor baixo comparado ao experimento anterior. Por outro lado, os valores majoritários obtiveram vantagem com o uso do PCA, observando que existiu um acréscimo de 22.197 para 22.358 instâncias classificadas corretamente como adimplentes.

**Tabela 4. Matriz de confusão da Árvore de Decisão com PCA.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	2804	3831
Adimplentes	1006	22358

##### 4.2.1.2 Rede Neural

A Rede Neural foi gerada e a classificação obtida foi pior que a sem pré-processamento, obtendo um erro de 22,0%, observado no *Matlab*.

Na Matriz de confusão da Rede Neural, Tabela 5, observa-se que apenas a classe de adimplentes foram classificados com melhor precisão, ou seja, existindo uma penalização da classe minoritária (inadimplente), contendo apenas 1 inadimplente classificado corretamente como inadimplente, o mesmo pode ser observado na Árvore de Decisão que também foi prejudicada com o uso do PCA e os valores majoritários (adimplentes) obtiveram um melhor rendimento em ambos os métodos propostos.

**Tabela 5. Matriz de confusão da Rede Neural com PCA.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	1	6635
Adimplentes	11	23353

##### 4.2.2 Normalização

Aplicando a função *normalize* do *Matlab* é possível gerar novos dados, com os valores numéricos da tabela normalizados, seguindo a Fórmula 7. Para os novos dados gerados serão realizadas análises de classificação com Árvore de Decisão e Rede Neural.

##### 4.2.2.1 Árvore de Decisão

Com o uso da Árvore de Decisão, gerando os dados cinco vezes no *WEKA* e comparando-os com o valor médio, observa-se que o erro de classificação com a normalização foi de 12,6%, menor em 4 pontos percentuais, que a classificação dos dados sem pré-processamento.

**Tabela 6. Matriz de confusão da Árvore de Decisão com a Normalização.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	3463	3172
Adimplentes	599	22765

Foi observado que a performance da Árvore de Decisão melhorou bastante, uma vez que os valores de adimplentes classificados erroneamente como inadimplentes diminuíram em relação aos dados sem pré-processamento, de 1.167 para

599, sendo um retorno considerável, já que com o uso do PCA esse valor de classificação foi o que sofreu a maior degradação.

#### 4.2.2.2 Rede Neural

Com o uso da Rede *Perceptron*, foram feitos ajustes na camada oculta, assim como no índice 4.1.2, mudando crescentemente e verificando a melhor quantidade de neurônios que se adequam à rede. O melhor nível de erro de classificação alcançado foi de 18.07%, com 15 neurônios na camada oculta.

Na Matriz de confusão da Rede Neural, Tabela 7, ocorreram poucas alterações em relação à análise realizada nos dados sem pré-processamento, sendo os dados bem próximos à distribuição realizada no item 4.1.3; porém, observa-se que na Rede Neural o número de Falso positivo e de Falso negativo foi aproximadamente 5,3 pontos percentuais maior que a classificação realizada pela Árvore de Decisão, com o uso da normalização.

**Tabela 7. Matriz de confusão da Rede Neural com a Normalização.**

Classificado→	Inadimplentes	Adimplentes
Inadimplentes	2371	4206
Adimplentes	1155	22209

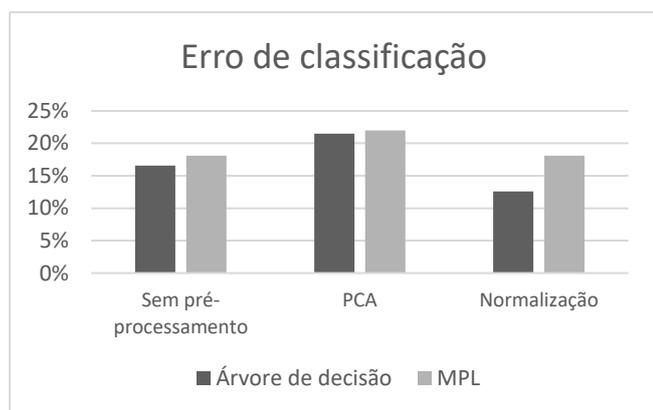


Fig. 5 Erro de classificação dos experimentos sem e com pré-processamento.

Comparando os seis resultados obtidos no gráfico de barras gerados no Excel (Figura 5), observa-se que o experimento com pré-processamento, utilizando a Normalização no modelo de Árvore de Decisão, obteve o resultado mais satisfatório dentre os seis resultados testados.

Em outros modelos para análise de risco de crédito implementados como uso de Rede Neural (SAMI, 2020), o modelo intitulado CS-MC4 chegou a ter acurácia de 96,97%. Por outro lado, modelos comparativos entre Árvore de Decisão e Rede Neural (GOLBAYANI,2020), obtiveram comportamentos parecidos com o deste experimento, com Árvore de Decisão obtendo melhor desempenho quando comparado à MPL.

## 5. CONCLUSÕES

Nos três experimentos, verificou-se uma taxa de erro acima de 13% na Árvore de Decisão e acima de 18% nos três experimentos com a Rede Neural. Como consequência do erro dessa análise, transações de concessão de crédito podem ser suspensas injustamente afetando a expectativa do cliente ou, então, as operadoras de cartão podem conceder crédito a um cliente com tradições de inadimplência. Logo, essas técnicas, devem ser aliadas à experiência do analista de crédito.

Para análise, neste experimento, foi aplicada a Análise Principal de Componentes, no intuito de verificar o comportamento dos dados com a junção de atributos correlacionados, mas não houve resultados significativos para os dados em observação. Apenas foi verificado que os dados majoritários (adimplentes) prevaleceram na análise da matriz de confusão, em detrimento dos dados minoritários (inadimplentes), mostrando que o PCA não é uma boa técnica para esse tipo de problema, já que os atributos correlacionados absorvidos conteriam informações relevantes para os dados minoritários de classificação.

Com os dados pré-processado, utilizando a normalização, os dados foram reescalados e os atributos que possuíam valores elevados, como crédito concedido, histórico de pagamentos, valor do extrato da fatura e valor do pagamento anterior, sofreram ajustes significativos, ajudando no desempenho da análise dos dados, tanto na Árvore de Decisão, que obteve 12,6% de classificação incorreta, quanto na Rede Neural, com 18,07%. O melhor desempenho ocorreu quando foram comparados os três experimentos realizados, com destaque para a Árvore de Decisão que alcançou seu melhor desempenho. Isso demonstra a importância de um pré-processamento dos dados, tornando-os mensuráveis para a devida classificação.

## AGRADECIMENTOS

Agradeço aos membros do SBA pela divulgação e incentivo ao meu estudo.

## REFERÊNCIAS

- Aleskerov, E., Freisleben, B. and Rao B., "CARDWATCH: a neural network based database mining system for credit card fraud detection," Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFER), New York City, NY, USA, 1997, pp. 220-226, doi: 10.1109/CIFER.1997.618940.
- Araujo Vieira, E.M. de, Neves, N., de Oliveira, A.C., de Moraes, R. e do Nascimento, J. 2018. Avaliação da performance do algoritmo J48 para construção de modelos baseados em árvores de decisão. Revista Brasileira de Computação Aplicada. 10, 2 (jul. 2018), 80-90. DOI:<https://doi.org/10.5335/rbca.v10i2.8078>.
- Baesens, Bart; Setiono, Rudy; Mues, Christophe; Vanthienen, Jan. Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. Management Science, [S. l.], v. 49, n. 3, p. 1683-1688, 1 mar. 2003.

- Baranauskas, José. O que é Clustering. [S. l.], 2018. Disponível em: <https://pt.scribd.com/document/375885355/AM-I-Clustering>. Acesso em: 28 nov. 2019.
- Cluster Analysis. [S. l.], 2019. Disponível em: <https://www.mathworks.com/help/stats/cluster-analysis.html>. Acesso em: 4 dez. 2019.
- Coelho, Caique. Um guia completo para o pré-processamento de dados em machine learning. [S. l.], 16 jun. 2019. Disponível em: <https://medium.com/@caiquecoelho/um-guia-completo-para-o-pr%C3%A9-processamento-de-dados-em-machine-learning-f860fbadabel>. Acesso em: 4 dez. 2019.
- De Paula, Mauricio. Introdução automática a árvore de decisão. 2002. 85 f. Dissertação (Mestrado em ciência da computação) - Aluno do Mestrado, [S. l.], 2002. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/82465/189971.pdf?sequence=1&isAllowed=y>. Acesso em: 14 nov. 2019.
- Default of credit card clients. Machine Learning Repository; Department of Information Management, Chung Hua University, Taiwan; Data Set. In: UCI: Repositório de Aprendizado de Máquina. Taiwan, 2016. Disponível em: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#>. Acesso em: 1 nov. 2019.
- Golbayani, P; Florescu I; Chatterjee, R. A comparative study of forecasting Corporate Credit Ratings using Neural Networks, Support Vector Machines, and Decision Trees. The North American Journal of Economics and Finance, p. 101251, 2020.
- Governo Federal. Banco Central. Cartão de Crédito: Utilize de Forma Consciente, Brasília DF: Banco Central, 2012.
- Hong, Haoyuan; Liu, Junzhi; Bui, Dieu Tien; Pradhan, Biswajeet; Acharya, Tri Dev; Pham, Binh Thai; Zhu, A-Xing; Chen, Wei; Ahmad, Baharin Bin; Landslide susceptibility mapping using J48 Decision Tree with AdaBoost, Bagging and Rotation Forest ensembles in the Guangchang area (China); CATENA, ISSN: 0341-8162, Vol: 163, Page: 399-413, 2018
- Librelotto, Rubert; Mozzaquatro, Patricia. análise dos algoritmos de mineração j48 e priori aplicados na detecção de indicadores da qualidade de vida e saúde. Revista Interdisciplinar de ensino, pesquisa e extensão, [s. l.], v. 1, 2013.
- Lima, Fabiano; PEREIRA, Luiz; FILHO, Antnio; KIMURA, Herbert. Aplicação de redes neurais na análise e na concessão de crédito ao consumidor. Revista de Administração - RAUSP, [s. l.], v. 44, p. 34-45, 2009
- M. Zekic-Susac, N. Sarlija and M. Bencic, "Small business credit scoring: a comparison of logistic regression, neural network, and decision tree models," 26th International Conference on Information Technology Interfaces, 2004., Cavtat, 2004, pp. 265-270 Vol.1.
- Neural Net Pattern Recognition. In: Neural Net Pattern Recognition. [S. l.], 2019. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/neuralnet-patternrecognition-app.html>. Acesso em: 22 nov. 2019.
- Normalize. [S. l.], 2019. Disponível em: <https://www.mathworks.com/help/matlab/ref/double.normalize.html>. Acesso em: 4 dez. 2019.
- Panigrahi, Ranjit; BORAH, Samarjeet. Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets. International Conference on Computational Intelligence and Data Science (ICCIDS 2018), [s. l.], p. 323-333, 2018.
- Prezepiorski, Lemos; ARNS, Steiner; TERESINHA, Maria; NIEVOLA, Júlio. Análise de crédito bancário por meio de redes neurais e árvores de decisão: uma aplicação simples de data mining. Revista de Administração - RAUSP, [s. l.], ano 3, v. 40, p. 225-234, 2005.
- Sami Ben Jabeur, Amir Sadaoui, Asma Sghaier & Riadh Aloui (2020) Machine learning models and cost-sensitive decision trees for bond rating prediction, Journal of the Operational Research Society, 71:8, 1161-1179, DOI: 10.1080/01605682.2019.1581405
- Soltan, A & Mohammadi, M. (2012). A hybrid model using decision tree and neural network for credit scoring problem. Management Science Letters, 2(5), 1683-1688.
- Vieira, Elamara; NEVES, Nívea; OLIVEIRA, Ana; MORAES, Ronei; NASCIMENTO, João. Avaliação da performance do algoritmo J48 para construção de modelos baseados em árvores de decisão. Revista Brasileira de Computação Aplicada, [s. l.], v. 10, ed. 2, p. 80-90, 2018.
- Weka. [S. l.], 2019. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: 13 nov. 2019.
- Zhang, G.; Patuwo, B.E.; HU, M.Y. Forecasting with artificial neural networks: the state of the art. International Journal of Forecasting, Kent (Ohio), v.14, n.1, p.35-62, 1998.
- Zhou, L., & Yau, S. Efficient association rule mining among both frequent and infrequent items. Computers & Mathematics with Applications., 2007, 54, 737-749.