UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Marcos Fonseca Alcure

Uso de estatística e métodos de aprendizado de máquinas na classificação de risco de clientes em operações de crédito imobiliário

São Carlos

2021

Marcos Fonseca Alcure

Uso de estatística e métodos de aprendizado de máquinas na classificação de risco de clientes em operações de crédito imobiliário

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas à Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciências de Dados.

Área de concentração: Ciências de Dados

Orientador: Prof. Dr.Afonso Paiva

São Carlos

2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

ALCURE , MARCOS FONSECA
A356u Uso de estatística e métodos de aprendizado de máquinas na classificação de risco de clientes em operações de crédito imobiliário / MARCOS FONSECA ALCURE ; orientador Afonso Paiva . -- São Carlos, 2021.
34 p.

Trabalho de conclusão de curso (MBA em Ciência de Dados) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. credit score . 2. Imobiliário. I. Paiva , Afonso , orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Marcos Fonseca Alcure

Uso de estatística e métodos de aprendizado de máquinas na classificação de risco de clientes em operações de crédito imobiliário

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas à Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciências de Dados.

Data de defesa: 22 de janeiro de 2022

Comissão Julgadora:

Prof. Dr.Afonso Paiva
Orientadora

Professor
Convidado1

Professor
Convidado2

São Carlos
2021

AGRADECIMENTOS

Em primeiro lugar, a Deus, por ter permitido que eu tivesse saúde determinação para não desanimar durante a realização deste trabalho.A minha esposa e filho, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho. Aos meus pais que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória. Ao meu orientador professor Dr. Afonso Paiva, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional e finalmente a instituição de ensino Universidade de São Paulo (USP), essencial no meu processo de formação profissional, pela dedicação, e por tudo o que aprendi ao longo do curso.

RESUMO

Alcure, M. F. **Uso de estatística e métodos de aprendizado de máquinas na classificação de risco de clientes em operações de crédito imobiliário** . 2021. 49p. Monografia (MBA em Ciências de Dados) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2021.

O mercado imobiliário trabalha com concessão de crédito para futuros clientes e é extremamente importante ser capaz de traçar um perfil desses clientes para que seja possível avaliar o risco do crédito a ser concedido. Em muitas empresas ainda existe uma deficiência na metodologia adotada para essa tomada de decisão. Com a melhoria de todo o ecossistema que envolve as análises de informações, ou seja, a possibilidade de se armazenar e se trabalhar com um grande quantidade de dados em conjunto com o desenvolvimento de áreas de estudo como a de cientista de dados, e com a disponibilização de plataformas como a linguagem de programação Python, a qual contém diversos pacotes desenvolvidos para esse fim, torna-se possível, dessa forma, que construtoras, mesmo que de pequeno e médio porte, possam ter acesso a análise de tomadas de decisões mais embasadas e muito mais robustas. O trabalho em questão analisa o impacto na tomada de decisão para risco de crédito (bom pagador e mau pagador) de uma carteira de clientes. Foram utilizados modelos de aprendizado de máquina para classificação de clientes como, bom pagador e mau pagador. Para tal utilizou-se os modelos Multilayer Perceptron e a Regressão Logistica. O primeiro, com um custo operacional um pouco mais elevado. Por fim, avaliou-se qual seria o modelo que melhor se ajustaria à carteira de clientes, conseguindo realizar a melhor classificação. Como metodologia, foi realizado um estudo exploratório na base de dados e posteriormente realizado alguns tratamentos necessários à base para a utilização da mesma no processo de machine-learning. Para avaliação final foram utilizadas as principais métricas de classificação: Acurácia, área sob a curva ROC (Auc) , Matriz de confusão, precision, reacall e f1. Os resultados obtidos ajudam a reforçar a importância da utilização de aprendizado de máquina no processo de tomada de decisão para classificação de risco. Conseguiu-se obter uma acurácia de 80% na classificação quando usado o melhor modelo encontrado, o Random Forest. Sem a utilização de nenhum modelo tem-se uma acurácia de 50% (bom ou mau pagador). Esse aumento na acurácia da classificação, indica que nesse caso colocar em operação esse modelo de "machine learning"poderá trazer um ganho financeiro expressivo para essa construtora.

Palavras-chave: crédito, Random Fprest, Multilayer Perceptron, Regressão Logistica, Aurácia, machine learning

ABSTRACT

Alcure, M. F. **Use of statistics and machine learning methods in the risk classification of customers in real estate credit operations**. 2021. 49p. Monografia (MBA em Ciências de Dados) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2021.

The real estate market works with granting credit to future customers, it is important to be able to draw a profile of these customers so that it is possible to assess the risk of the credit to be granted. In many companies there is still a deficiency in the methodology adopted for this decision making. With the improvement of the entire ecosystem that involves the analysis of information, that is, the possibility of storing and working with a large amount of data together with the development of areas of study such as the data scientist, and with the availability of platforms such as the Python programming language, which contains several packages developed for this purpose, thus making it possible for construction companies, even small and medium-sized ones, to have access to more informed and much more robust decision-making analysis. This work analyzes the impact on decision making for credit risk (good and bad debtors) of a customer portfolio. Machine learning models were used to classify customers as good and bad payers. For this, the Multilayer Perceptron and Logistic Regression models were used, the first with a slightly higher operational cost. Finally, it was evaluated which model would best fit the customer portfolio, achieving the best classification. As a methodology, an exploratory study was carried out in the database and later performed some treatments necessary for the use of machine-learning process. For the final evaluation, the main classification metrics were used: Accuracy, area under the ROC curve (Auc), Confusion matrix, precision, reacall and f1. The results obtained help to reinforce the importance of using machine learning in the decision-making process for risk classification. It was possible to obtain an accuracy of 80% in the classification when we used the best model found, the Random Forest. Without the use of any model, there is an accuracy of 50% (good or bad payers). This increase in the accuracy of the classification indicates that, in this case, putting this "machine learning" model into operation could bring a significant financial gain to this construction company.

Keywords:credit, Random Fprest, Multilayer Perceptron, Logistic Regression, Auracia, machine learning

SUMÁRIO

1 INTRODUÇÃO

O grau de risco que um determinado cliente tem ao contratar um financiamento junto a uma construtora, para um determinado empreendimento, torna-se objeto de interesse para tomadas de decisões tais como: determinar valor compatível de juros junto ao cliente de forma a assegurar ou pelo menos minimizar o risco da operação, ou até mesmo orientá-los para outras opções de empreendimentos.

O entendimento do risco associado a uma operação de crédito, com o uso do aprendizado de máquina é uma metodologia que vem sendo usada cada vez mais por fintechs e empresas de pequeno e médio porte. Algumas técnicas estatísticas tem sido amplamente aplicadas para a construção dos modelos de pontuação de crédito, como Análise Discriminante Linear (LDA) (REICHERT; CHO; WAGNER, 1983; KARELS; PRAKASH, 1987), Análise de Regressão Logística (IRA) (THOMAS, 2000; WEST, 2000), Linhas de Regressão Adaptativa Multivariada (MARS) (FRIEDMAN, 1991). No entanto, o problema com a aplicação dessas técnicas estatísticas à pontuação de crédito resulta que, em algumas suposições, como as premissas de normalidade multivariada para variáveis independentes, são frequentemente violadas na realidade, o que torna essas técnicas teoricamente inválidas para amostras finitas (HUANG et al., 2004). Nos últimos anos, muitos estudos tem demonstrado que técnicas de IA, como Rede Neural Artificial (ANN) (DESAI; CROOK; JR, 1996), árvore de decisão (DT) (MAKOWSKI, 1985; HUNG; CHEN, 2009) , e Máquina vetorial de suporte (SVM)(HUANG; CHEN; WANG, 2007; BAESENS et al., 2003; SCHEBESCH; STECKING, 2005) podem ser usadas como métodos alternativos para pontuação de crédito (WANG et al., 2012).

Em contraste com as técnicas estatísticas, as técnicas de IA não assumem certas distribuições de dados. Essas técnicas extraem automaticamente o conhecimento de amostras de treinamento. Deve-se ressaltar que técnicas de IA trata-se de modelo complexo, e deve ser abordado com pleno entendimento das suas peculiaridades. Dentre elas destacam-se as influências das limitações dos modelos, das escolhas dos atributos dos credores e do tamanho do banco de dados, bem como uma boa adequação dos dados ao modelo a ser utilizado. Um exemplo é a necessidade de se fazer ajustes em dados desbalanceados e a utilização de técnicas de PCA (do inglês, Principal Component Analysis) para reduzir atributos redundantes ou também desnecessários ao modelo e, dessa forma, melhorar o desempenho computacional.

Zhang et al. (2007) utilizaram técnicas de AI e apresenta a acurácia da classificação de risco de clientes em três diferentes modelos: Redes Neurais (BP), Programação Genética (GP) e máquinas de vetoriais de suporte (SVM). Para isso foram usados dois grandes bancos de dados de clientes: o primeiro, o Credit German e o segundo, o Australia credit data.

Além dos modelos acima, os autores apresentaram um modelo com uma melhor acurácia nas classificações entre, bons e maus pagadores, construindo um modelo combinado (CM). Para cada base de dados, foram gerados oito grupos de estudo distintos e analisado a acurácia de cada modelo para cada um desses grupos e, finalmente, para cada modelo foi calculado uma média ponderada referente à resposta de cada um, em cada um desses oito grupos. Todos os modelos propostos no trabalho se apresentaram como bons candidatos para classificação de análise do credit score. Foram obtidos graus de assertividades não inferiores a 88,5% referente ao banco de dados australiano e não inferiores 79,77% no banco de dados de clientes alemães. Os autores entendem que o fato do banco de dados alemão ser desbalanceado, tendo clientes muito bons, chegando a 70% , pode ser um dos motivos pelo seu baixo desempenho quando comparado ao resultado obtido pelo banco de dados australiano. Notamos que nesse estudo os autores não se preocuparam em observar a influência do tipo de erro II (um cliente com crédito ruim é mal classificado como um cliente com bom crédito) na resposta de cada modelo, que se pautou apenas no aspecto da acurácia, sabe-se que modelos que apresentam um alto grau de erro tipo II são responsáveis por absorver clientes potencialmente custosos ao sistema. Outra questão que chama a atenção seria que, apesar de citar o desbalanceio dos dados provenientes do German Credit Data Set, onde 70% dos clientes são bons pagadores, não foi realizado nenhum tratamento específico a esses dados afim de mitigar o seu impacto no resultado. Acreditamos que uma abordagem que leve em consideração esse desbalanceamento seja capaz de gerar resultados mais precisos.

No trabalho de Wang et al. (2012), os autores utilizaram os mesmos bancos de dados (German Credit Data e Australian Data set) do trabalho anterior com uma abordagem diferente ao problema. Primeiramente, para minimizar a influência da variabilidade do conjunto de treinamento, foi realizado a validação cruzada nos conjuntos de dados de crédito. Em relação ao método adotado foi feito uma análise do uso da metodologia de árvore de decisão agregando mais duas etapas, *bagging* e *random subspace*, uma vez que de modo geral a análise de crédito apenas com a árvore de decisão não se obtém bons resultados pois se resulta em um modelo que sofre grande influência dos ruídos do conjunto de dados e da redundância dos atributos. O processo de *bagging* tem por objetivo reduzir a influência dos ruídos, enquanto que o *random subspace* minimiza as perdas de precisão do modelo devido a redundância dos atributos. Os resultados obtidos com emprego dessa metodologia, conseguiram fazer com que o método de árvore de decisão passasse a ser mais uma opção para tratar problemas de classificação para análise de crédito, independentemente da ordem adotadas para as etapas agregadas que possuem influência na acurácia do resultado final. O menor valor de acurácia para a metodologia aplicada ao Australian Credit Dataset ficou em 88.01% , um pouco abaixo do apresentado no trabalho de Zhang et al. (2007). O erro do tipo II também se apresentou abaixo dos modelos usuais em 7.52%. Para o resultado do banco de dados do German Credit a mesma realidade

se repetiu, apresentando o ganho de desempenho e classificando essa metodologia como mais uma possível ferramenta para análise de crédito. Várias direções futuras de pesquisa também surgem de acordo com este estudo. Em primeiro lugar, grandes conjuntos de dados para experimentos e aplicações, particularmente com maior exploração de estruturas de dados de classificação de crédito, devem ser coletados para validar ainda mais as conclusões do estudo.

Djeundje et al. (2021) introduziram uma forma diferente de tratar a modelagem de análise de crédito através de atributos totalmente não usuais, para esse tipo de análise ao realizar uma série de modelagens com a metodologia de regressão logística, utilizando-se apenas de informação do cliente (tomador do crédito) relacionadas ao fluxo de e-mail, tais como: quantidades de e-mails enviados entre meia-noite e 6 horas da manhã, média de palavras usadas no título dos últimos dois mil e-mails, quantidade de e-mails enviado em certo dia da semana, dentre outros. Vários desses atributos conseguiram um p-value correspondente a 5% , demonstrando a representatividade desse atributo na resposta do modelo como altamente relevante. Esse trabalho deixa claro o grande grau de liberdade que temos ao se abordar um problema de credit score.

Diversos métodos encontram-se disponíveis hoje, porém, apesar do intenso estudo da pontuação de crédito, não há consenso sobre a técnica de classificação mais adequada para utilização. Importante notar que ao tratar de análise de credit score não se pode focar apenas no modelo a ser utilizado, mas sim no processo como um todo. Nos casos apresentados acima, isso aparece de forma clara, com a inserção de mais duas etapas na metodologia de árvore de decisão, aumentando a assertividade na resposta do modelo, assim como a abordagem distinta apresentado no trabalho de Djeundje et al. (2021) na escolha de diferentes atributos, pode-se mudar de forma expressiva os resultados, independente do modelo utilizado.

O objetivo desse trabalho visa analisar o perfil de um cliente através de atributos associados (idade, sexo, escolaridade, renda, profissão, etc..) utilizando métodos de aprendizado de máquinas, tais como: regressão logística, redes neurais e árvores de classificação, com o intuito de classifica-lo de acordo com a sua predisposição para inadimplência em relação à pagamentos futuros, ou seja, seu grau de risco.

1.1 Estrutura do trabalho

O presente trabalho está estruturado da seguinte forma: Capitulo 1 Introdução, Capítulo 2 Metodologia, onde é apresentado todo ferramental teórico para o desenvolvimento do trabalho. O Capítulo 3 contém os resultados obtidos. Finalmente, Capítulo 4, onde é apresentada a conclusão e uma visão geral para trabalhos futuros.

2 METODOLOGIA

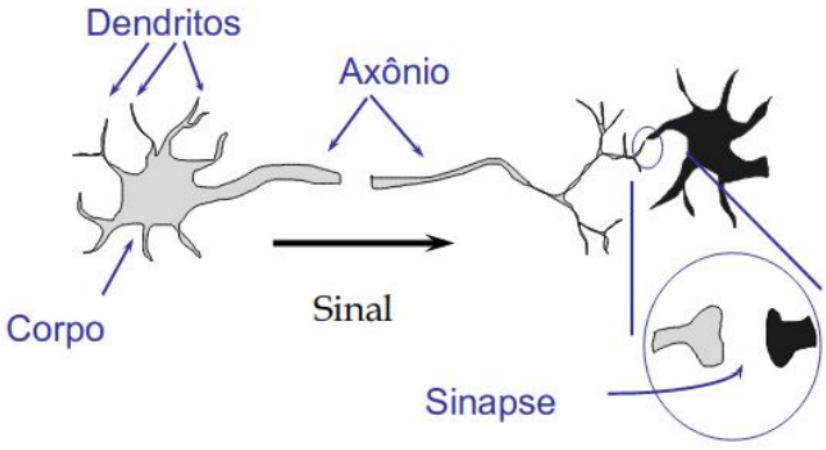
A análise de classificação de risco de crédito dos clientes, será feita através de um processo de classificação utilizando-se do algorítimo de Rede Neural - *Multilayer Perceptrons* (MLP).Esse mesmo banco de dados também será submetido a uma análise computacionalmente menos custosa, no caso a Regressão Logística. E posteriormente se buscará identificar, através de análise comparativa de diversos modelos, qual apresenta o melhor resultado de classificação.

2.1 Modelos de Aprendizado de Máquina:

2.1.1 Redes Neurais Multilayer Perceptron (MLP):

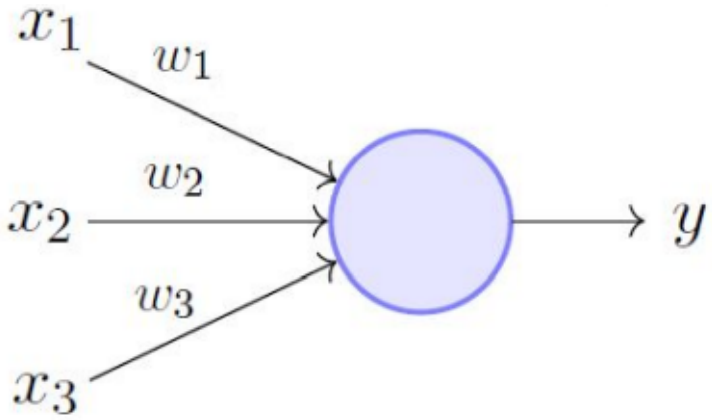
Antes de prosseguirmos para explicar as Redes Neurais Multilayer Perceptron é importante entendermos o conceito do Perceptron. O Perceptron é baseado no mesmo princípio de uma rede de neurônios biológicos, ele é conhecida como uma Rede de Neurônio Artificial (RNA), essa semelhança fica clara ao observarmos a Figura 1 e Figura 2.

Figura 1 – Modelo de um neurônio biológico.



Fonte: André Ponce de Leon F de Carvalho

Figura 2 – Modelo de um Perceptron.



Fonte: Perceptron Model (Minsky-Papert em 1996). Os valores (x_1, x_2 e x_3) representam os atributos de entrada, (w_1, w_2 e w_3) o peso determinado pelo modelo Perceptron associado a cada atributo e como saída temos a variável alvo, representada pelo y.

Cada conexão de entrada (x_1, x_2 e x_3) no Perceptron está associada um peso (w_1, w_2 e w_3), gerando uma soma ponderada. De forma que teremos a equação 2.1

$$u = \sum_{i=1}^{N} x_i w_i \,. \tag{2.1}$$

Posteriormente aplicasse uma função degrau f, dada pela equação 2.2:

$$f(u) = \begin{cases} +1, & \text{se} \quad u \geq \theta \\ -1, & \text{se} \quad u < \theta \end{cases} \,, \tag{2.2}$$

A função degrau tem por objetivo fazer a separação para uma classificação binária, por exemplo valores acima de θ classifica-se na classe positiva e valores menores e iguais a θ na classe negativa. A classe negativa e positiva, entende-se por qualquer classificação binária tais como estar ou não doente, quente ou frio.

A rede Perceptron busca ajustar os pesos de forma que a classificação final, seja compatível. Primeiramente os pesos são colocados de forma aleatórias, ao rodar a rede neural (épocas) os pesos vão se ajustando, através de uma equação de erro, e alterando a equação linear (2.1) de forma que em algum momento consiga-se a melhor divisão entre as classes. A Figura 3 representa essa separação de classes.

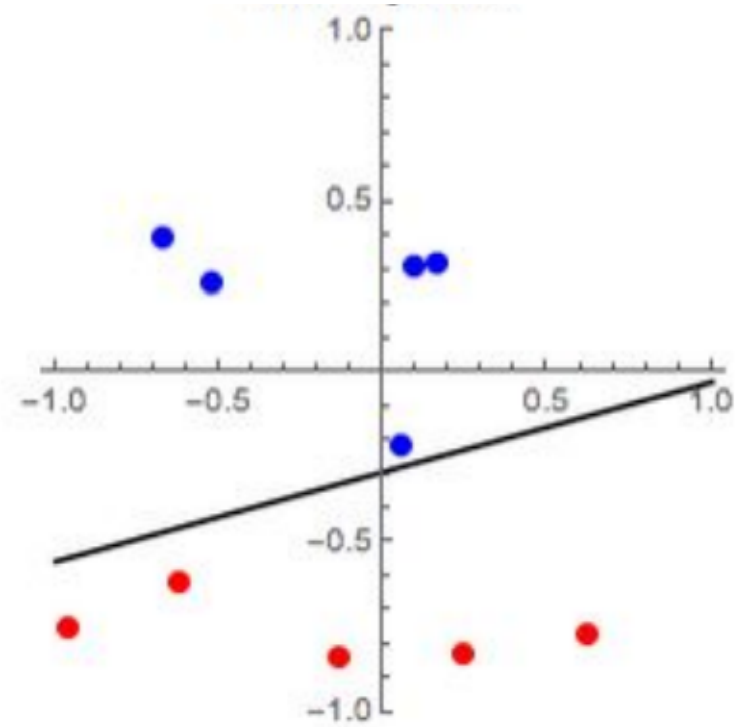
Observe que para que o Perceptron tenha uma boa resposta os dados precisam ser linearmente separáveis, de modo que o algoritmo perceptron consiga corrigir para um conjunto consistentes de pesos.

Com apenas um neurônio não se pode fazer muita coisa, mas podemos combiná-los em uma estrutura em camadas, cada uma com número diferente de neurônios, formando uma rede neural denominada Perceptron Multicamadas (“Multi Layer Perceptron — MLP”). O vetor de valores de entrada x passa pela camada inicial, cujos valores de saída

são ligados às entradas da camada seguinte, e assim por diante, até a rede fornecer como resultado os valores de saída da última camada. Pode-se arranjar a rede em várias camadas, tornando-a profunda e capaz de aprender relações cada vez mais complexas. A Figura 4 apresenta a estrutura de uma MLP com duas camadas escondidas ou intermediárias.

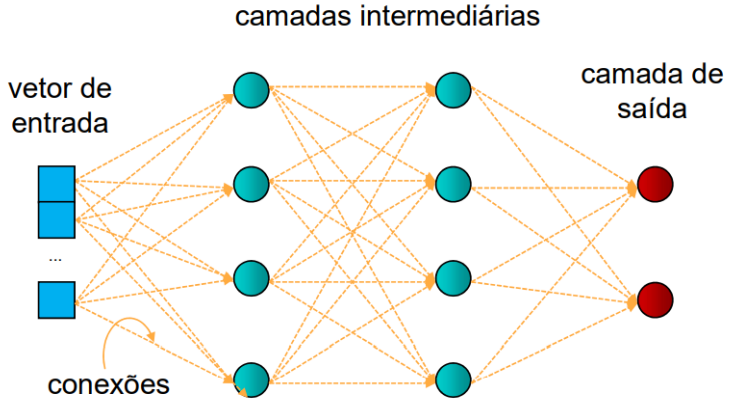
A fórmula para o MLP é a mesma do Perceptron simples, sendo que, no MLP, para neurônios cuja camada anterior não é a camada de entradas, as entradas são as saídas dos

Figura 3 – Figura 3: Classificação dada pelo Perceptron: os dados são separados em duas classes, uma positiva (pontos azuis) e negativa (pontos vermelhos).



Fonte: Prof Leonardo Mauro P. Moraes

Figura 4 – Multi Layer Perceptron com duas camadas (representado pelas duas colunas com os círculos verdes) com classificação binária representadas pelos círculos vermelhos.



Fonte: André Ponce de Leon F de Carvalho

neurônios anteriores. Conforme equação 2.3:

$$u = \sum_{i=1}^{N} x_i w_{ij} + b_j \tag{2.3}$$

Na equação **??** , as entradas x_i podem ser os sinais submetidos a rede ou a saída de neurônios que são entradas de outros neurônios. O bias b está associado aos neurônios, isto é, cada neurônio tem um bias e gera um potencial de ativação u que corresponde a soma ponderada das entradas pelos devidos pesos. Após o cálculo da soma ponderada, ele é somado ao bias.

Funções de Ativação

A Rede MLP utiliza-se nas camadas ocultas funções de ativações não-lineares o que possibilita a transformação de um sistema não-linearmente separável em um problema linearmente esperável. A aplicação dessas funções nas camadas ocultas, possibilita a alteração da dimensão do problema, realização de transformações não-lineares sucessivas e a composição de funções. Essas operações realizadas nas camadas ocultas simplificam o problema para a camada de saída e possibilitam a classificação antes impossível na rede Perceptron. Abaixo algumas funções de ativação:

Figura 5 – Função de ativação utilizadas nas camadas ocultas do modelo Perceptron.

Nome	Figura	Função	Derivada
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Fonte: André Ponce de Leon F de Carvalho

Treinamento de uma rede Multi Layer Perceptron

Para que uma rede dessas funcione, é preciso treiná-la. É como ensinar a uma criança o beabá. O treinamento de uma rede MLP insere-se no contexto de aprendizado de máquina supervisionado, em que cada amostra de dados utilizada apresenta um rótulo informando a que classificação ela se encaixa.

O treinamento da rede se dar por uma técnica chamada de *backpropagation* que consiste em calcular do erro ocorrido na camada de saída da rede neural, recalcular o valor dos pesos do vetor w da camada última camada de neurônios e assim proceder

para as camadas anteriores, de trás para a frente, ou seja, atualizar todos os pesos w das camadas a partir da última até atingir a camada de entrada da rede, para isso realizando a retropropagação o erro obtido pela rede. Em outras palavras, calcula-se o erro entre o que a rede achou que era e o que de fato deveria ser, então recalculamos o valor de todos os pesos, começando da última camada e indo até a primeira, sempre tendo em vista diminuir esse erro. Conforme o algoritmo abaixo:

1. Inicializar todos os pesos e bias da rede (w) com pequenos valores aleatórios;
2. Fornecer dados de entrada à rede e calcular o valor da função de erro obtida, ao comparar com o valor de saída esperado;
3. Na tentativa de minimizar o valor da função de erro, calculam-se os valores dos gradientes para cada peso da rede. Como queremos caminhar com os pesos na direção de maior decréscimo da função de erro, basta tomarmos o sentido contrário ao do gradiente;
4. Uma vez que temos o vetor gradiente calculado, atualizamos cada peso de modo iterativo, sempre recalculando os gradientes em cada passo de iteração, até o erro diminuir e chegar abaixo de algum limiar preestabelecido, ou o número de iterações atingir um valor máximo, quando enfim o algoritmo termina e a rede está treinada.

Assim, a fórmula geral de atualização dos pesos (w) na iteração é dada por:

$$w \leftarrow w - \eta(\frac{dE}{dw}) \tag{2.4}$$

A atualização do bias é análoga a dos pesos, sendo que, é calculado um novo gradiente para o bias a ser atualizado. A fórmula geral de atualização dos bias (b) na iteração é dada por:

$$b \leftarrow b - \eta(\frac{dE}{db}) \tag{2.5}$$

O valor do peso na iteração atual será o valor do peso na iteração anterior, corrigido de valor proporcional ao gradiente. O sinal negativo indica que estamos indo na direção contrária à do gradiente, conforme mencionado. O parâmetro n representa a taxa de aprendizado da rede neural, controlando a tamanho do passo que tomamos na correção do peso.

O conceito-chave da equação anterior é o cálculo dos termos das equações (2.4) e (2.5) consistindo em computar as derivadas parciais da função de erro E em relação a cada peso do vetor w e ao bias.

Sendo y a saída esperada e $\hat{y}$ a saída obtida pela rede, definimos a função de erro como sendo:

$$E(y, \hat{y}) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \,. \tag{2.6}$$

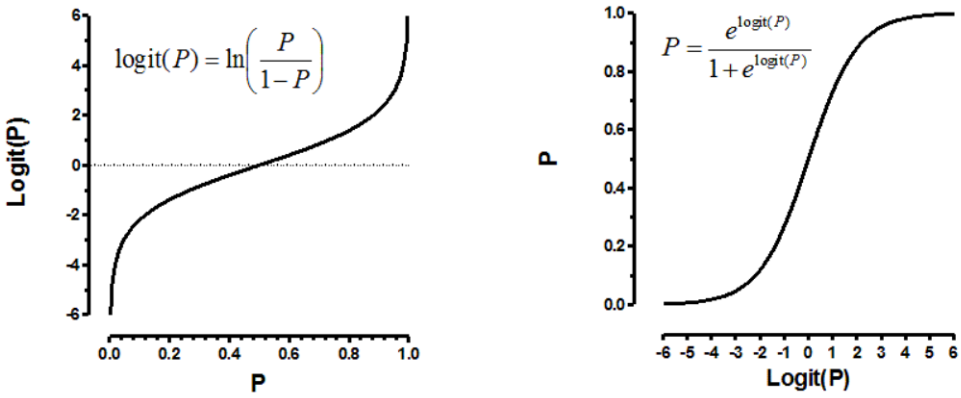
2.1.2 Regressão Logística

A regressão logística envolve o cálculo da função *logit* representativa do logaritmo neperiano da razão entre a probabilidade de se pertencer a um grupo e a probabilidade de se pertencer a outro grupo, conforme a equação 2.7:

$$logit[\Pi(x_1, ...x_n)] = \ln(\frac{\Pi(x_1, ...x_n)}{1 - \Pi(x_1, ...x_n)}), \tag{2.7}$$

onde $\Pi(x_1, \ldots, x_n)$ representa a probabilidade de um evento dado seus valores de x_n. Na regressão logística estima-se uma relação linear entre várias variáveis explicativas (x_n) e a razão entre probabilidades de uma observação pertencer a um e a outro grupo. Em particular, as premissas das regressões logísticas envolvem a linearidade do relacionamento na equação e a ausência de interação entre as variáveis independentes.

Figura 6 – Funcão *logit*.



Fonte: estatisite.com.br

2.2 Pré-processamento

De forma geral os bancos de dados disponíveis precisam passar por um processo de análise e tratamento dos dados antes de serem submetidos a qualquer processo de aprendizado de maquinas. É nessa fase do pipeline que trataremos:

- **Existência de dados nulos:** a existência de dados nulos cria uma descontinuidade nos parâmetros, o que em alguns casos impossibilita aplicação de técnicas de aprendizado de máquina. Para esse trabalho trataremos essa questão utilizando o `KNNinputer` implementado no Python.

- **Valores redundantes e errôneos:** analisamos redundância removendo atributos constantes e linhas duplicadas. Valores errôneos que possam ser corrigidos, devem ser tratados. Os que não puderem ser corrigidos são removidos.

- **Valores de parâmetros em escalas diferentes:** banco de dados com parâmetros em escalas diferentes, afetam de forma direta soluções de classificação de aprendizado de máquina em algoritmos cuja solução envolva o cálculo da distância, tal como a função Euclidiana. Nesse trabalho verificaremos a necessidade da aplicação da normalização dos dados pelo MaxminScaler uma ferramenta da biblioteca do Sklearn.

- **Existência de outliers:** os outliers são dados não representativos do modelos. Podem ser afirmações que tiveram algum erro na digitação. A existência dessa informação errônea no banco de dados, pode enviesar negativamente todo o resultado de uma análise. Para esse trabalho utilizaremos do método estatístico usando a Normal uni-variada na busca de outliers.

- **Valores desbalanceados:** ao se trabalhar com um banco de dados desbalanceado existe uma tendência natural da classe preponderante de cada atributo se tornar mais representativa nos resultados. Nesse trabalho faremos a análise da divisão de classes por atributos e caso necessário será realizada a técnica do SMOTE implementada na biblioteca do Sklearn.

- **Codificação de variáveis categóricas:** variáveis categóricas, geralmente expressas por meio de texto, não são utilizáveis diretamente por parte dos modelos estatísticos e computacionais. Para o trabalho em questão será utilizada a função *get_dummies* implementada no Sklearn. A vantagem desse método é que ele aprende as categorias a partir do conjunto, fornece parâmetros para tratar com categorias desconhecidas e permite transformação inversa.

- **Análise Exploratória dos dados:** essa parte do trabalho tem como objetivo ter uma visão geral dos dados, ser a pedra fundamental, um primeiro passo para a análise de dados. Devemos destacar e chamar a atenção para aspectos e padrões que não podem se confirmar inferencialmente. Iremos trabalhar em três frentes, observando as medidas de posição, medidas de dispersão e associação dos dados.

2.3 Modelos adotados

Para a análise serão utilizados os dois modelos apresentados nas seções anteriores, MLP e a Regressão Logística. Esses modelos estão implementados na biblioteca do Python. Além disso será realizada a busca pelo modelo otimizado, o que melhor consegue realizar a classificação das duas classes (bom pagador e mau pagador).

Nas redes neurais MLP temos que ajustar os hiper-parâmetros tais como o número de camadas ocultas, número de neurônios por camada o tipo de função de ativação por camada, etc). Nesse trabalho serão adotadas as seguintes estratégias para a adoção dos hiper-parâmetros.

Número de camadas ocultas

Como estratégia de redução de custo operacional, o modelo será iniciado com uma ou duas camadas ocultas o que na maioria das vezes resolvem bem os problemas.

Número de neurônios por camada oculta

Em relação a quantidade de neurônios por camada iremos começar com uma quantidade maior de neurônios do que realmente precisa e ir diminuindo essa metodologia e conhecida como “calça-curta”.

Funções de ativação

Será adotada a função de ativação ReLU nas camadas ocultas pois é mais rápida para calcular do que as outras funções de ativação e o Gradiente Descendente não fica tão preso em platôs graça ao fato que ela não satura para grandes valores de entrada.

2.4 Métricas de avaliação dos modelos

Para esse trabalho vamos adotar as seguintes métricas de avaliação dos modelos já muito bem testadas e altamente consolidadas:

Acurácia

Proporção de exemplos classificados corretamente em um conjunto com n objetos,varia entre 0 e 1 e valores próximos de 1 são melhores. A da acurácia é definida conforme equação 2.8:

$$Ac(f) = 1 - \frac{1}{n}\sum_{i}^{n} I(y_i \neq f(x_i)) \tag{2.8}$$

Matriz de confusão

As colunas da matriz A representam classes preditas. Cada coeficiente a_{ij} denota o número de exemplos da classe c_i classificados como pertencentes à classe c_j. A diagonal da matriz A são os acertos do classificador e elementos fora da diagonal são os erros cometidos. A Figura 7 mostra um exemplo de uma matriz de confusão.

Figura 7 – Exemplo de Matriz de Confusão: dados Iris. Os labels das colunas representam a classe real e os labels nas linhas a classe predita. Os valores em azul a quantidade de acerto e em vermelho a de erros.

Clase Predita	Classe Correcta		
	Setosa	Versicolor	Virgínica
Setosa	15	0	0
Versicolor	0	14	1
Virginica	0	1	4

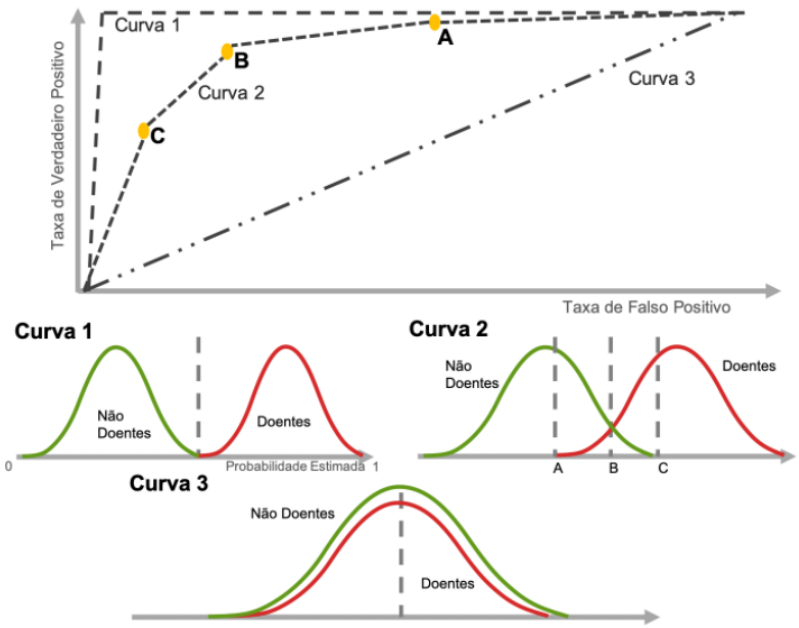
Fonte: Francisco A. Rodrigues

Curva ROC

A curva ROC apresentado na figura 8 é um gráfico simples mas robusto, que permite estudar a variação da sensibilidade (taxa de verdadeiro positivo) e especificidade (taxa de falso positivo), para diferentes pontos de corte na probabilidade estimada (*thresholds*).

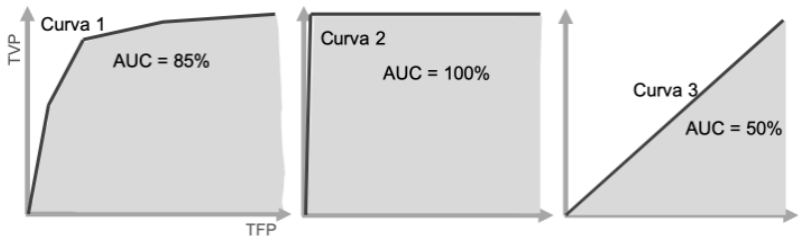
A área sob a curva, apresentado na figura 9, (Area Under the Curve - AUC) é uma medida que facilita a comparação entre duas curvas ROC. O valor da AUC varia no intervalo $[0, 1]$.

Figura 8 – Curva ROC representa num gráfico a relação entre o resultado da taxa de falso positivo e a taxa de verdadeiro positivo.



Fonte: Francisco A. Rodrigues

Figura 9 – Representação de valores de AUC calculada pela área abaixo da curva ROC. Quanto mais próximo de 100% melhor a resposta do modelo.



Fonte: Francisco A. Rodrigues

3 RESULTADOS

3.1 Contexto dos dados utilizados:

O conjunto de dados original, contém 1000 entradas com 20 atributos categorial/simbólico. Neste conjunto de dados, cada entrada representa uma pessoa que recebe um crédito por um banco. Cada pessoa é classificada como baixo ou alto em relação ao risco de tomada de crédito de acordo com o conjunto de atributos. Os atributos selecionados são:

- “Idade” (numérica);
- “Sexo” (texto: masculino, feminino);
- “Trabalho” (numérico: 0 - não qualificado e não residente, 1 - não qualificado e residente, 2 - qualificado, 3 - altamente qualificado);
- “Moradia” (texto: próprio, alugado ou casa parente);
- “Ativos” (texto - pouco, moderado, bastante rico, rico);
- “Conta corrente” (numérica);
- “Valor do financiamento” (numérico);
- “Duração” (numérica, em mês);
- “Propósito” (texto: carro, móveis/equipamentos, rádio/TV, eletrodomésticos, reparos, educação, negócios, férias/outros).

Com o objetivo de trazer a base de dados para a realidade da proposta do trabalho, foi realizada uma adequação no atributo “Propósito”. De forma que o objetivo dos empréstimos remetesse a financiamento de apartamentos ou casas em determinados *empreendimentos* (EMP), conforme apresentado abaixo:

- Propósito (texto: EMP 1, radio/TV:EMP 2, furniture/equipment:EMP 3, business:EMP 4, education:EMP 5, repairs:EMP 6, domestic appliances:EMP 7, vacation/others:EMP 8).

A tabela 1 já apresenta a base de dados com essa modificação.

Tabela 1 – Estrutura dos dados utilizados

	Idade	Sexo	Profissao	Moradia	Ativos	Conta_corrente	Valor_financiamento	Duracao_mes	Proposito	Risco
0	67	homem	2	propria	NaN	baixo	1169	6	EMP_2	baixo
1	22	mulher	2	propria	baixo	moderado	5951	48	EMP_2	alto
2	49	homem	1	propria	baixo	NaN	2096	12	EMP_4	baixo
3	45	homem	2	casa_parente	baixo	baixo	7882	42	EMP_3	baixo
4	53	homem	2	casa_parente	baixo	baixo	4870	24	EMP_1	alto

Fonte: German Credit Data - Adaptado

Tabela 2 – Distribuição dos dados

	count	mean	std	min	25%	50%	90%	max
Idade	1000.0	35.546	11.375469	19.0	27.0	33.0	52.0	75.0
Profissao	1000.0	1.904	0.653614	0.0	2.0	2.0	3.0	3.0
Valor_financiamento	1000.0	3271.258	2822.736876	250.0	1365.5	2319.5	7179.4	18424.0
Duracao_mes	1000.0	20.903	12.058814	4.0	12.0	18.0	36.0	72.0

Fonte: Autor

3.2 Preparação dos dados

Nesse momento o objetivo é ter uma visão macro da qualidade dos dados, para isso foi observado a distribuição dos dados conforme apresentado na tabela 2, os tipos de dados (floats, inteiros, objetos) a existência de dados duplicados ou nulos.

Distribuição dos dados

Pela distribuição dos dados, não é possível identificar nenhuma anormalidade em relação aos atributos, “Idade”, “Profissão”, “Valor financiamento” e “Duração”. Entenda como anormalidade valores impossíveis de pertencerem ao atributo em questão, tal como idades negativas ou valores muito acima da idade média de vida esperada.

Tipos de dados e dados nulos

Aqui existe um ponto de atenção, observamos que para os atributos “Ativos” e “Conta corrente” existe uma grande quantidade de dados faltantes conforme pode ser visto pela tabela 3, respectivamente 183 e 394. Essa base de dados precisará ser tratada para que possamos utiliza-la em algum tipo de aprendizado de máquina. Mas adiante apresentaremos a metodologia adotada na solução desse problema, por enquanto estamos apenas fazendo o mapeamento e a identificação. Em relação aos tipos de dados, esses encontram-se em conformidade.

Tabela 3 – Exemplo de dados e dados nulos.

#	Column	Non-Null Count	Dtype
0	Idade	1000 non-null	int64
1	Sexo	1000 non-null	object
2	Profissao	1000 non-null	int64
3	Moradia	1000 non-null	object
4	Ativos	817 non-null	object
5	Conta_corrente	606 non-null	object
6	Valor_financiamento	1000 non-null	int64
7	Duracao_mes	1000 non-null	int64
8	Proposito	1000 non-null	object
9	Risco	1000 non-null	object

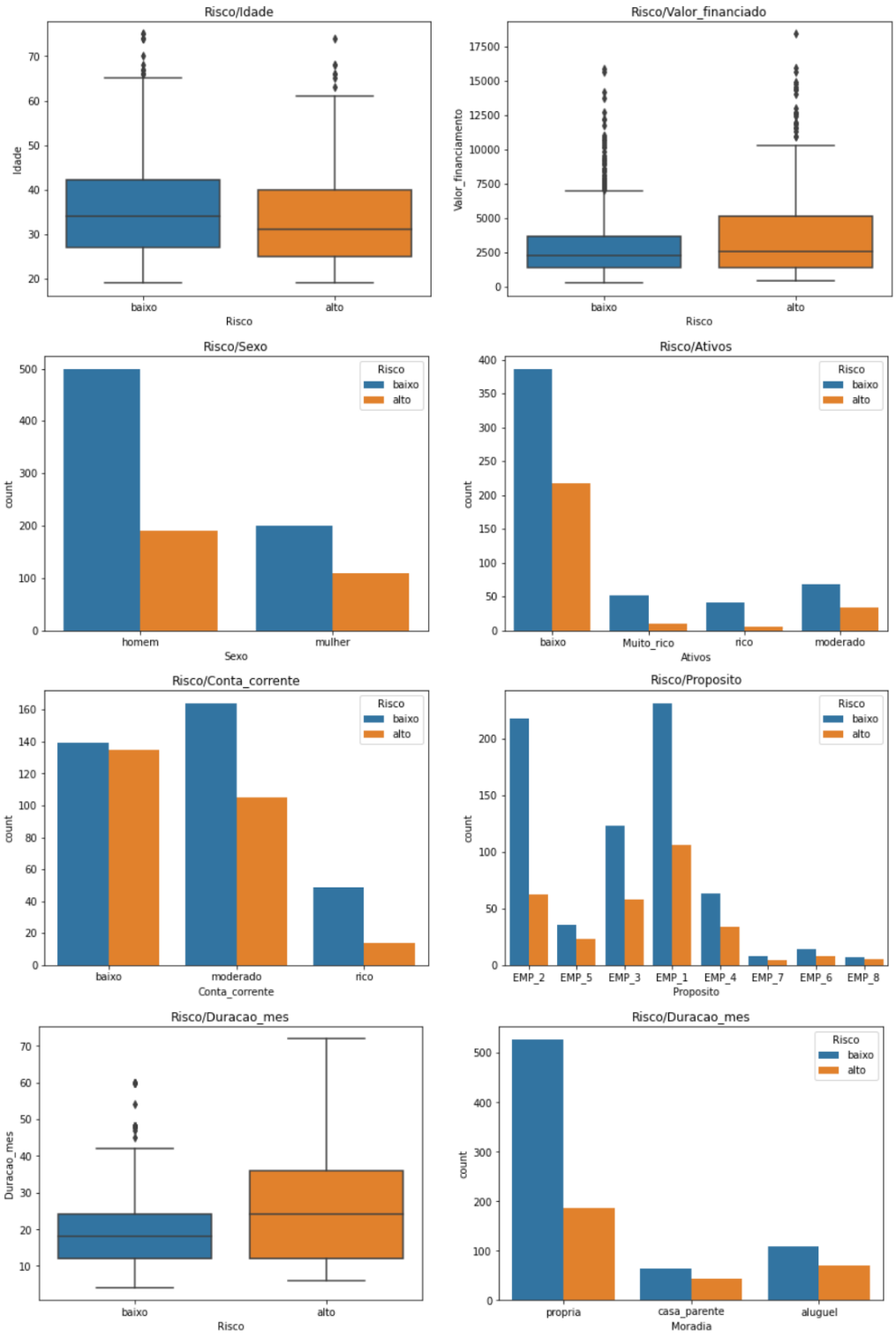
Fonte: Autor

Aprofundando a visão dos dados

A figura 10 apresenta a primeira visão da relação entre os atributos e a classificação do risco (baixo e alto), para isso foram geradas analises gráficas apresentadas , do risco com a idade, “Valor financiado, “Sexo”, “Ativo”, “Propósito” e “Duração do mês”.

Em relação a idade observa-se que os mais jovens apresentam um risco de crédito maior, enquanto que para os mais idosos o risco de credito e menor. Essa observação da diferença de risco a depender da idade, fez com que preferíssemos trabalhar com o atributo de idade, os agrupando, gerando grupos de idades e não com idades isoladas conforme o banco de dados originais. Posteriormente será apresentado a metodologia utilizada na separação dos grupos.

Figura 10 – Comportamento do padrão de risco para os diversos atributos.



Fonte: Autor

Observa-se também a influência do valor financiado 75% do risco baixo encontrasse em empréstimos abaixo de 3750 mil reais tendo como valor máximo 7500 mil reais. Da mesma forma vemos valores acima de 3500 mil reais e 10.000 mil reais mais propensos a um risco alto.

Em relação ao atributo ”Sexo”, existe uma tendência do sexo feminino ser pertencente a um risco maior, o risco alto aparece para aproximadamente 28% dos homens enquanto para as mulheres esse número sobe para 33%

Para os atributos: “Ativos", “Conta corrente"e “Moradia"esses também seguem uma ordem de risco diferente a depender da classe, essa observação inicial nos levou a tratar no nosso modelo final esses atributos como variáveis ordinais, exatamente pelo fato de entender que existe uma relação de forma ordenada entre o risco e as classes. Posteriormente será apresentado a definição essa ordenação.

Em relação a duração do empréstimo, pode-se observar pela 10 que quanto mais duradouro for o empréstimo tem-se um maior risco associado.

Tratando os atributos não numéricos do banco de dados

Para utilização desse banco de dados nas análises de classificação, faz-se necessário transformar as variáveis não numéricas em numéricas. Para isso precisamos definir se serão ordinais, ou seja, dotadas de uma ordem lógica ou qualitativa nominal onde não existe uma ordenação.

Conforme foi analisado, ficou sugestivo que os atributos “Moradia”, “Ativos” e “Conta corrente” pudessem pertencer a uma classificação de variável ordinais. Para poder chegar a essa conclusão e conseguir ordenar essas classes foi realizado o cálculo da porcentagem de risco alto e baixo para cada atributo. A tabela 4 apresenta esses valores.

Tabela 4 – Tabelas utilizadas para fazer a classificação ordinal dos atributos (moradia, ativo e conta corrente). Para tal utilizou-se os valores associados aos riscos: alto e baixo.

Risco	alto	baixo
Moradia		
aluguel	38.636364	61.363636
casa_parente	37.373737	62.626263
propria	25.428571	74.571429

Risco	alto	baixo
Ativos		
Muito_rico	17.741935	82.258065
baixo	34.804754	65.195246
moderado	32.000000	68.000000
rico	12.500000	87.500000

Risco	alto	baixo
Conta_corrente		
baixo	49.070632	50.929368
moderado	36.363636	63.636364
rico	22.222222	77.777778

Fonte: Autor

Os atributos “Moradia”, “Ativos” e “Conta corrente” foram substituídos pelos valores numéricos conforme esquema abaixo. Observe que existe uma ordem onde os menores valores representam menor grau de risco de tomada de crédito para o atributo em questão. Vamos tomar por exemplo a “Moradia”, dentro dela tem-se três classes (aluguel, casa parente e própria). Ao analisarmos a tabela acima vemos que o risco de pegar crédito para quem tem casa própria, alugada ou mora em casa de parente e de 25% , 38% e 37%. Logo maior risco para quem mora de aluguel e menor para quem tem casa própria. Sendo assim a ordenação segue da seguinte forma:

- **“Moradia”:** própria - 1 / casa parente - 2 / aluguel - 3;
- **“Ativos”:** rico - 1 / muito rico - 2 / moderado - 3 / baixo - 4;
- **“Conta corrente”:** rico - 1 / moderado - 2 / baixo - 3.

Os atributos qualitativos nominais também foram transformados para a forma numérica, porem nesse caso utilizando-se a função *get_dummies* implementada no Python. No banco de dados identificamos os seguintes atributos como qualitativo nominal, sexo, proposito e risco. Essa função gera dados numéricos apenas com a capacidade de classificar o atributo e não de ordena-los. Na tabela 5 temos um exemplo de como fica essa transformação no atributo "Sexo".

Tabela 5 – Atributos (Homem e mulher) transformado em forma numérica.

Sexo_homem	Sexo_mulher
1	0
0	1
1	0
1	0
1	0

Fonte: Autor

Criando grupos definidos por faixa etária

Ao aprofundarmos a visão dos dados na section anterior, notamos que existe uma relação entre a idade e o risco, no intuito de tentar explorar mais essa relação foram criados grupos com diferentes faixas etárias. De modo que pessoas entre 19 e 28 anos fazem parte do grupo denominado JovensI, entre 29 e 38 anos sao os JovensII, entre 39 e 48 anos AdultosI, entre 49 e 58 anos AdultosII, entre 59 e 68 anos SeniorsI e acima de 68 anos SeniorsII. Dessa forma vamos substituir a coluna de atributo idade pela coluna contendo os grupos de faixa etária. A tabela 6 apresenta como fica a divisão em termos quantitativos nesses grupos.

Tabela 6 – Os seis grupos originados do agrupamento das idades, com suas respectivas quantidades.

JovensI	363
JovensII	320
AdultosI	170
AdultosII	72
SeniorsI	41
SeniorsII	7

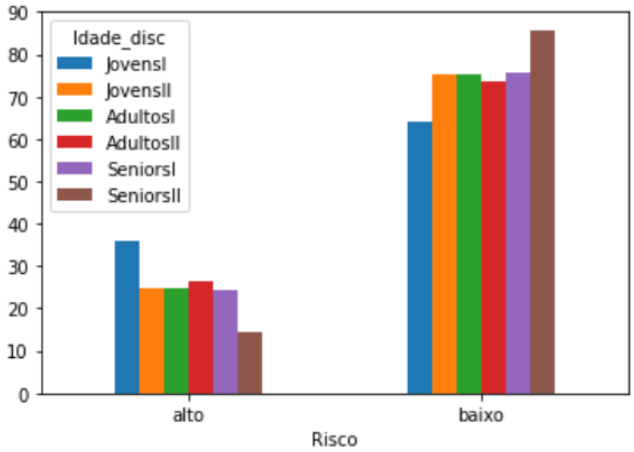
Fonte: Autor

Importante notar que como precisamos trabalhar com valores numéricos nas simulações, essas classes contendo os grupos etários, foram transformadas em números da

mesma forma que fizemos para os atributos “Sexo”, “Propósito” e “Risco”, utilizando novamente a função *get_dummies*.

A sugestão antes observada no gráfico de boxplot, de que os Jovens representavam um maior risco e os mais idosos um menor risco a tomada de credito, consegue ser visualizada de forma mais objetiva após a realização do agrupamento das faixas etárias. Conforme pode ser visto pela figura 11 onde o eixo da ordenada representa a porcentagem de cada evento.

Figura 11 – No eixo das ordenadas temos o valor da porcentagem de cada grupo com risco baixo ou alto de inadimplência.No eixo das abcissas a separação das classes de risco (alto e baixo).



Fonte: Autor

Tratando Outliers

Conforme pode ser analisado no gráfico de boxplot existe uma grande quantidade de outliers nos valores financiados, dessa forma decidiu-se por tratar esses valores antes de rodar o moledo de classificação. A metodologia adotada foi a remoção através das duas extremidades do conjunto de dados, para isso utilizou-se um intervalo de confiança de 95% descartando valores que ficaram fora do range estipulado entre a média mais três desvios padrões e menos três desvios padrões. Após essa limpeza os dados passaram de um conjunto com 1000 instâncias para 975.

Tratando as escalas dos valores

Após a transformação de todas as variáveis para numéricas, temos que nos assegurar de que não exista uma disparidade entre os valores. Conforme pode ser visto pela amostra do banco de dados na tabela 7, temos valores indo até oito mil vezes maior que o outro. Para tratar essa questão usamos a ferramenta *MaxminScaler* implementada no Python, com um intervalo $[0, 1]$. Após essa etapa todos os valores passarão a existir no espaço entre $[0, 1]$ não ocorrendo mais disparidade de valores.

Tabela 7 – Os valores de atributos antes de serem normalizados, apresentando muita disparidade.

	Profissao	Moradia	Ativos	Conta_corrente	Valor_financiamento	Duracao_mes	Sexo_homem	Sexo_mulher
0	2	1	NaN	3.0	1169	6	1	0
1	2	1	4.0	2.0	5951	48	0	1
2	1	1	4.0	NaN	2096	12	1	0
3	2	2	4.0	3.0	7882	42	1	0
4	2	2	4.0	3.0	4870	24	1	0
...	...	...	...	...	...	...	...	...
1253	2	3	3.0	2.0	1264	15	1	0
1254	2	1	4.0	2.0	8386	30	1	0
1255	3	3	4.0	NaN	4844	48	1	0

Fonte: Autor

Tabela 8 – Os valores de atributos após passar pela normalização, sem disparidade entre os valores

	Profissao	Moradia	Ativos	Conta_corrente	Valor_financiamento	Duracao_mes	Sexo_homem	Sexo_mulher
0	0.666667	0.0	0.500000	1.0	0.081041	0.029412	1.0	0.0
1	0.666667	0.0	1.000000	0.5	0.502734	0.647059	0.0	1.0
2	0.333333	0.0	1.000000	1.0	0.162787	0.117647	1.0	0.0
3	0.666667	0.5	1.000000	1.0	0.673016	0.558824	1.0	0.0
4	0.666667	0.5	1.000000	1.0	0.407407	0.294118	1.0	0.0
...	...	...	...	...	...	...	...	...
1253	0.666667	1.0	0.666667	0.5	0.089418	0.161765	1.0	0.0
1254	0.666667	0.0	1.000000	0.5	0.717460	0.382353	1.0	0.0
1255	1.000000	1.0	1.000000	1.0	0.405115	0.647059	1.0	0.0
1256	0.666667	0.0	1.000000	1.0	0.703616	0.470588	1.0	0.0

Fonte: Autor

Tratando os valores nulos

Por final antes de rodar o modelo, temos que tratar dos valores nulos. Para esse trabalho foi decidido trabalhar com a implementação do `KNNinputer` uma ferramenta do Sklearn, onde os valores faltantes de cada amostra são imputados usando o valor médio de n vizinhos mais próximos encontrados no conjunto de treinamento. Para esse caso foi utilizado número de vizinhos $k = 2$.

Tratando os valores desbalanceados

Existe um desbalanceio nas classes do classificador, a quantidade de valores com risco baixo é bem maior do que com risco alto de uma tem-se 692 casos relatados como de

baixo risco e 283 como de alto risco. Para que se possa rodar algum algoritmo nessa base de dados, tem-se que fazer um balanceamento entre essas classes. No trabalho em questão, foi utilizado o método SMOTE da biblioteca Scikit-learn.

Hiper-parâmetros

Os hiper-parâmetros adotados para os três modelos estão apresentados no apêndice.

3.3 Classificador através do modelo MLP e Regressão Logística

Para que seja possível a avaliação da acurácia do modelo de classificação, o banco de dados foi dividido em dois grupos. O grupo de treinamento e o grupo de teste.Para o grupo de treinamento utilizou-se 60% do banco de dados e e os 40% restantes para o teste.

3.4 Resultado de comparação de principais modelos

Na Figura 12 tem-se o resultado das principais métricas para os modelos mais usuais de classificação para a solução do nosso problema. Interessante observar que a escolha do modelo adequado , assim como de seus hiperparametros é de suma importância no resultado obtido, basta observar a acurácia de cada modelo, saímos de 0.58 no pior modelo SVM-linear Kernel para 0.78 no melhor caso o Random Forest. Isso significa, sair de algo totalmente ineficaz no sentido operacional, para um modelo que de fato traria um resultado importante se aplicado numa tomada de decisão. Sendo assim a busca pelo melhor modelo faz-se necessária em qualquer projeto de machine learning uma vez que as variações de resultado são expressivas.

Figura 12 – Resultado da simulação entre diversos modelos

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Random Forest Classifier	0.7800	0.8571	0.7572	0.7992	0.7756	0.5600
Extra Trees Classifier	0.7748	0.8318	0.7779	0.7741	0.7754	0.5496
Light Gradient Boosting Machine	0.7696	0.8521	0.7697	0.7702	0.7693	0.5393
Gradient Boosting Classifier	0.7676	0.8555	0.7509	0.7767	0.7628	0.5351
Ada Boost Classifier	0.7222	0.8046	0.7387	0.7149	0.7255	0.4445
Decision Tree Classifier	0.7211	0.7212	0.7284	0.7178	0.7222	0.4424
K Neighbors Classifier	0.6807	0.7379	0.7677	0.6557	0.7063	0.3623
Linear Discriminant Analysis	0.6385	0.6867	0.6391	0.6408	0.6374	0.2774
Ridge Classifier	0.6375	0.0000	0.6370	0.6397	0.6359	0.2753
Logistic Regression	0.6365	0.6859	0.6474	0.6362	0.6389	0.2733
Naive Bayes	0.5868	0.6343	0.7759	0.5615	0.6507	0.1746
SVM - Linear Kernel	0.5859	0.0000	0.5869	0.5537	0.5361	0.1739

Fonte: Autor

3.5 Resultado pela visão das métricas

Todas as métricas foram obtidas com crossfolder constituídos de 10 folders, os valores das métricas mencionadas abaixo representam uma média desses valores.

Ao avaliar as métricas da Figura 13 referente aos três modelos apresentados acima, observa-se como o modelo de regressão linear obtém uma solução pobre, com valores em sua grande maioria menores que 0.7 (lembrando que as métricas variam no intervalo $[0, 1]$, onde 0 seria a menor precisão para a métrica e 1 a maior, somente no caso do AUC que esse range varia entre $[0.5, 1]$). Como trabalhamos com um caso binário como alvo (bom pagador $\times$ mau pagador), não teríamos muito ganho se decidíssemos por adotar o modelo de regressão logística como ferramenta de decisão uma vez que aleatoriamente já temos uma acurácia média de 0.5. O modelo de regressão linear gerou uma solução quase que linear para a divisão das classes e acabou nao conseguindo identificar a não linearidade do problema, sendo assim considerado um modelo ineficaz nesse caso.

O modelo MLP, apresentou um pouco mais de flexibilidade na fronteira de decisão entre as classes deixando o modelo menos rígido e conseguindo uma acuracidade média de 0.716 que já é considerado um bom resultado. A área AUC na Figura 14 abaixo da curva ROU apresentou um bom valor médio de 0.778 que representa um bom resultado. A métrica do Recall que nos daria a informação da taxa de acerto dos maus pagadores está em torno de 0,744, ou seja, existe uma boa chance ao se usar esse modelo de se conseguiríamos identificar um mau pagador e agir de forma a proteger a empresa. Em relação a identificação dos bons pagadores a taxa é um pouco menor 70% (precision), valores baixos de recall nos leva a erroneamente descartar um bom pagador ao identifica-lo como um risco, nesse caso a o valor de 70% é o limiar o que pode -se considerar um bom resultado para essa métrica. A métrica f1 leva em consideração tanto a identificação errônea de um mau pagador como de um bom pagador, ela junta a percepção obtida no Recall e no precision. No caso do MLP o f1-score ficou em 73% mais uma vez um bom resultado. O modelo MLP poderia ser um bom modelo a ser adotado numa empresa para tentar minimizar as perdas causada pelos maus pagadores, se o mesmo for empregado em uma empresa de médio a grande porte esse valor o ganho financeiro para a empresa pode ser considerável.

O modelo que apresentou uma melhor resposta foi o Random Forest, ele conseguiu criar subregioes bem específicas, sem que ocorresse o overfiting. Obteve-se uma acurácia média de 78% uma AUC média de 85% , recall de 75% , precision de aproximadamente 80%, um bom ganho em relação ao MLP, o que permite aceitar os bons pagadores com maior segurança. O f1-score teve um ganho de 4% , em relação ao MPL indo para 77%. A adoção desse seria sem dúvida de grande benéfica para a empresa, isso nos mostra como a

utilização de técnicas de aprendizado de máquina pode de fato trazer grandes benefícios na tomada de decisão, nesse caso saímos de uma probabilidade natural de 50% para aproximadamente 80%.

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7010	0.7781	0.7143	0.7000	0.7071	0.4019	0.4020
1	0.7938	0.8720	0.8163	0.7843	0.8000	0.5874	0.5879
2	0.6082	0.7003	0.7083	0.5862	0.6415	0.2181	0.2228
3	0.7526	0.8308	0.7500	0.7500	0.7500	0.5051	0.5051
4	0.7938	0.8444	0.7917	0.7917	0.7917	0.5876	0.5876
5	0.6804	0.7504	0.7292	0.6604	0.6931	0.3614	0.3634
6	0.7113	0.7487	0.7708	0.6852	0.7255	0.4234	0.4266
7	0.6907	0.7190	0.7292	0.6731	0.7000	0.3819	0.3832
8	0.6979	0.7522	0.7292	0.6863	0.7071	0.3958	0.3966
9	0.7292	0.7891	0.7083	0.7391	0.7234	0.4583	0.4587
Mean	0.7159	0.7785	0.7447	0.7056	0.7239	0.4321	0.4334
SD	0.0526	0.0529	0.0351	0.0591	0.0447	0.1049	0.1037

(a) Métricas - MLP

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6392	0.6718	0.5918	0.6591	0.6237	0.2790	0.2805
1	0.6907	0.7241	0.6735	0.7021	0.6875	0.3816	0.3820
2	0.5670	0.6267	0.6667	0.5517	0.6038	0.1358	0.1387
3	0.6186	0.7113	0.6250	0.6122	0.6186	0.2372	0.2372
4	0.6289	0.6662	0.5000	0.6667	0.5714	0.2558	0.2640
5	0.5773	0.6084	0.6458	0.5636	0.6019	0.1558	0.1574
6	0.6082	0.6909	0.6875	0.5893	0.6346	0.2177	0.2208
7	0.6598	0.7249	0.7292	0.6364	0.6796	0.3205	0.3239
8	0.6458	0.6793	0.6250	0.6522	0.6383	0.2917	0.2919
9	0.7292	0.7556	0.7292	0.7292	0.7292	0.4583	0.4583
Mean	0.6365	0.6859	0.6474	0.6362	0.6389	0.2733	0.2755
SD	0.0466	0.0432	0.0647	0.0547	0.0447	0.0927	0.0920

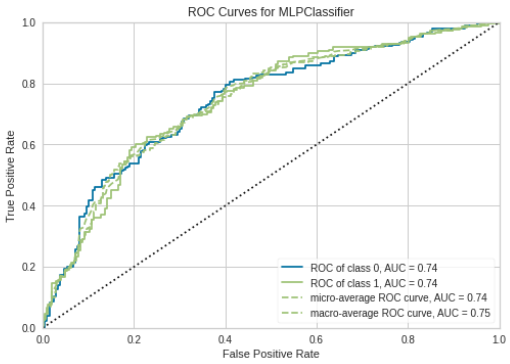
(b) Métricas - Regressão Linear

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7629	0.8476	0.7551	0.7708	0.7629	0.5258	0.5259
1	0.8557	0.9175	0.7755	0.9268	0.8444	0.7118	0.7217
2	0.6907	0.7821	0.7708	0.6607	0.7115	0.3824	0.3877
3	0.8144	0.8701	0.7500	0.8571	0.8000	0.6284	0.6332
4	0.8247	0.8973	0.7708	0.8605	0.8132	0.6491	0.6526
5	0.7113	0.8276	0.7500	0.6923	0.7200	0.4231	0.4246
6	0.7732	0.8718	0.6875	0.8250	0.7500	0.5456	0.5532
7	0.7835	0.8335	0.7917	0.7755	0.7835	0.5671	0.5672
8	0.7708	0.8557	0.7500	0.7826	0.7660	0.5417	0.5421
9	0.8125	0.8674	0.7708	0.8409	0.8043	0.6250	0.6272
Mean	0.7800	0.8571	0.7572	0.7992	0.7756	0.5600	0.5635
SD	0.0481	0.0359	0.0266	0.0762	0.0397	0.0959	0.0970

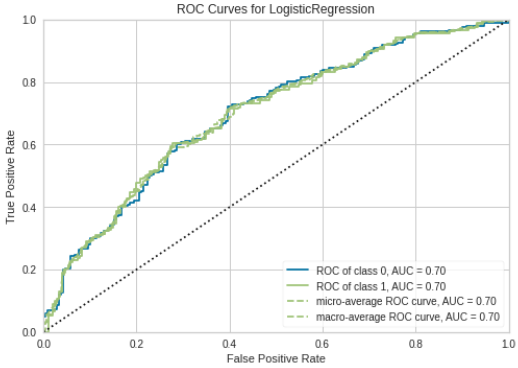
(c) Métricas - Random Forest

Figura 13 – Tabelas com resultado das principais métricas para os modelos estudados, tabela(a) - MLP, tabela(b)- Regressão Logística, Tabela (c)- Random Forest. Cada linha da tabela representa um resultado de uma simulação realizada pelo método do crossfolder, nas duas últimas linhas de cada tabela tem-se a média desses valores e o desvio padrão.

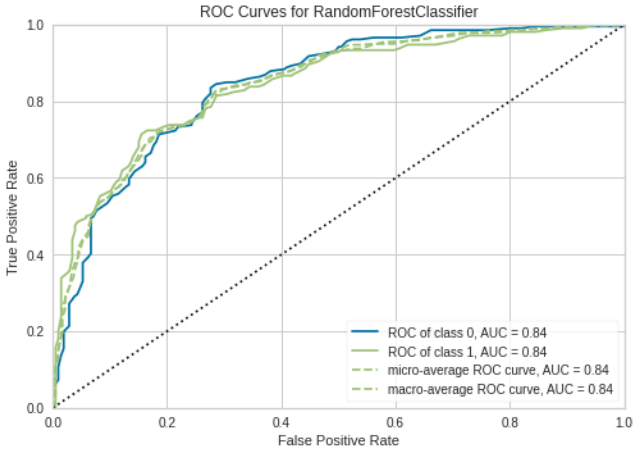
A figura 14 apresenta a curva ROC para os três modelos:



(a) ROC - MLP



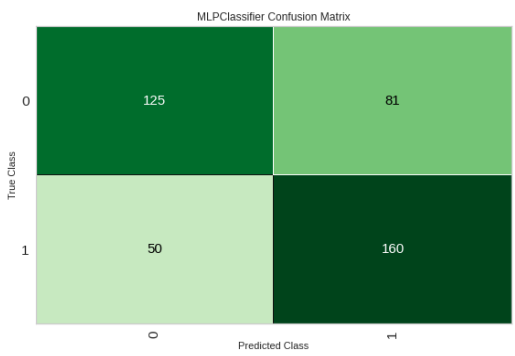
(b) ROC - Regressão Linear



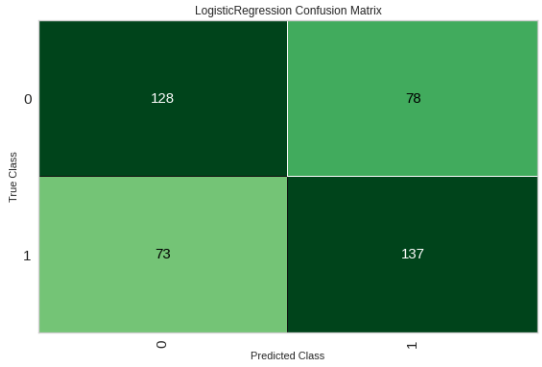
(c) ROC - Random Forest

Figura 14 – Resultado de curvas ROC para as três simulações, tabela(a)- MLP, tabela(b)- Regressão Logistica, tabela(c)- Random Forest. Tem-se quatros curvas ROCs para cada simulação, separadas da seguinte forma. A curva azul representa a classe 0, a verde a classe 1, as outras duas pontilhadas são a média micro e macro.

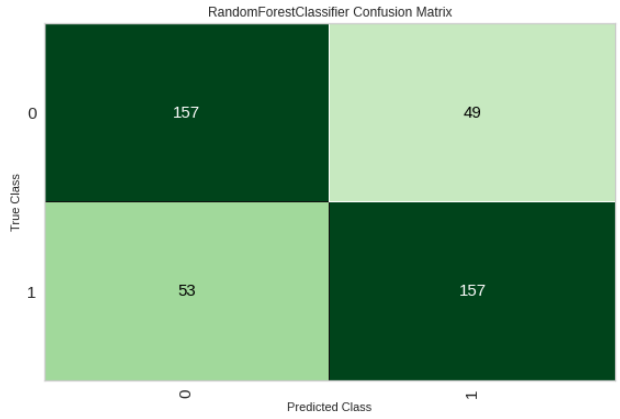
Figura 15 apresenta a matriz de confusão para os três modelos estudados. Na diagonal principal temos as classificações corretas,ou seja, o verdadeiro positivo e o verdadeiro negativo. Observa-se como o melhor modelo Random Forest esse número de acertos cresce em relação aos outros, tanto na identificação do mau pagador ($x = 1$, $y = 1$) como do bom pagador ($x = 0$, $y = 0$). Importante ter em mente que ao usar qualquer modelo de aprendizado de máquina como ferramenta para tomada de decisão, existe ainda uma imprecisão associada a esse modelo. No caso desse trabalho, ao analisar o melhor modelo, observa-se que o mesmo identificou erroneamente 25% dos bons pagadores como mau pagadores e 23% dos maus pagadores como bons pagadores. Levanto essa questão para que se tenha em mente que devesse tentar trabalhar com um ecossistema de informações o maior possível para tomada de decisões e não apenas confiar cegamente num único modelo.



(a) Matriz de Confusão - MLP



(b) Matriz de Confusão - Regressão Linear



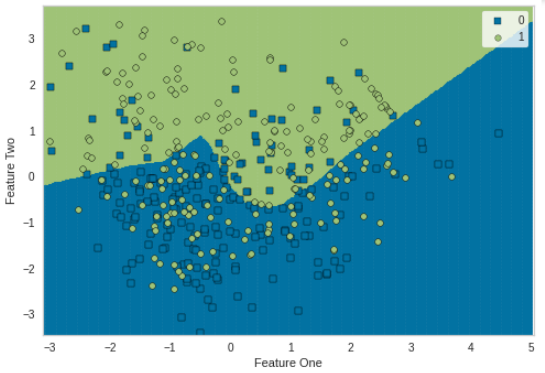
(c) Matriz de Confusão - Random Forest

Figura 15 – representa matriz de confusão dos modelos estudados, tabela(a)-MLP, tabela(b)- Regressão Linear, tabela(c)-Random Forest, na diagonal (verde escuro) tem-se as classificações corretas e na outra diagonal (verde claro) os erros de classificação.

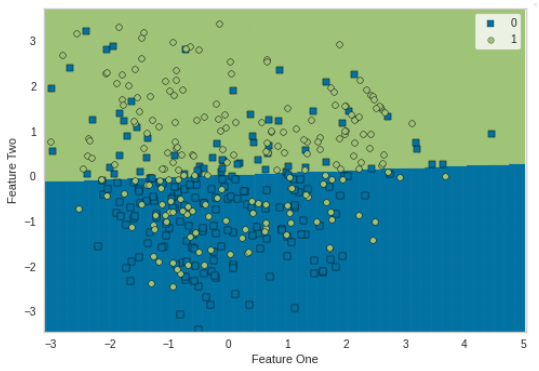
Na figura 16 temos as regiões de decisão que são os limites entre uma classe e outra. A depender de como é feita essa compartimentação conseguiremos ter um melhor ou pior resultado na classificação. A cor verde representa a região do mau pagador, enquanto que a cor azul do bom pagador. Os pontos azuis e verdes são as classificações reais. Observe que existem pontos azuis na área verde e verde na área azul, significa que o limite criado pelo modelo não foi capaz de separar esses pontos de forma correta.

Observa como o modelo de regressão linear criou essa divisão de classe, quase que linear, o que deixou o modelo pobre e com muitos pontos classificados de forma errônea, provavelmente por se tratar de um problema não linear. A divisão criada pelo modelo MLP, já saiu um pouco dessa linearidade, fazendo com que a classificação fosse mais precisa. Agora observe como o modelo Random Forest organizou de forma muito mais detalhada a divisão das classes, conseguindo uma acurácia muito superior. A preocupação seria a possibilidade desse modelo tão detalhado ter criado um overfiting, ou seja, um resultado que respondesse bem para o conjunto de dados do treinamento mas ruim para um conjunto de dados novo. Mas esse não foi o caso, uma vez que a acurácia obtida foi

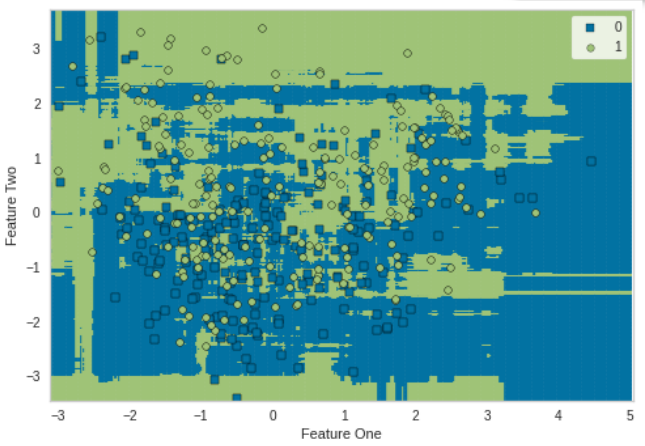
em cima do conjunto de dados de teste, nunca visto antes pelo modelo.



(a) Fronteira de decisão - MLP



(b) Fronteira de decisão - Regressão Linear



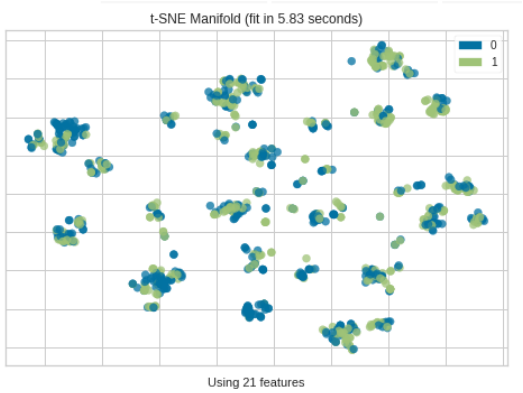
(c) Fronteira de decisão - Random Forest

Figura 16 – Idetificação da fronteira de decisão para cada modelo estudado, (a)-MLP, (b)-Regressão Logistica, (c)-Random Forest, na cor azul a região da classe 0 e na cor verde da classe 1. Um ponto verde na classe azul identifica um falso positivo, e um ponto azul na região verde um falso negativo.

A Figura 17 abaixo é uma representação 2D da divisão das classes

Aqui consegue-se observar como as classes mau pagador e bom pagador encontram-se bem misturadas, o que reforça a necessidade de um modelo que utilize um grande número de compartimentação para realizar uma boa predição. Consegue-se então entender o porquê de um modelo de seleção de limites simples como o Regressao Logistica apresentou resultados tao ruins e por outro lado o porquê do modelo mais complexo do Random Forest, conseguiu uma acurácia de aproximadamente 80%.

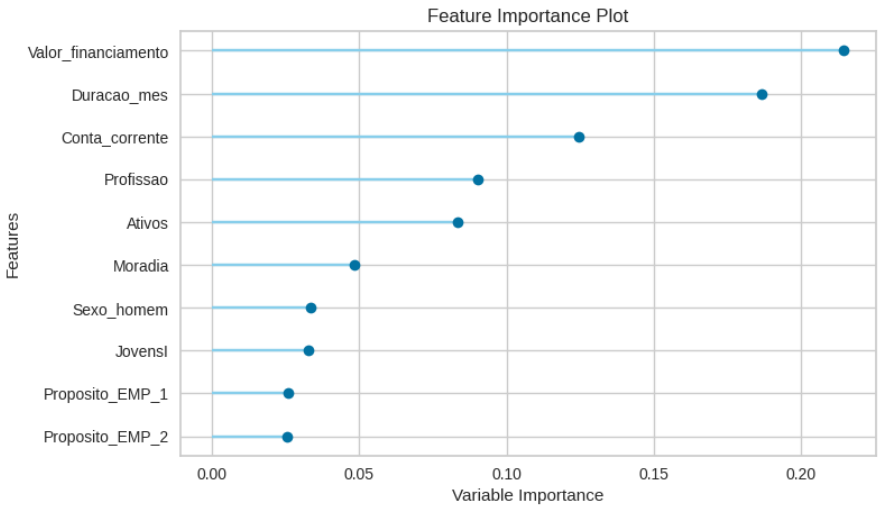
Figura 17 – Representação gráfica em duas dimensões da divisão das classes para o modelo Random Forest. Os pontos azuis representam a classe 0 e os pontos verdes a classe 1.



Fonte: Autor

A Figura 18 apresenta a importância dos atributos, trabalhar com um dataset com muitos atributos acaba sendo muito custoso, computacionalmente e a depender da organização dos dados da empresa pode gerar custo de mão de obra para a coleta dessa informação. Logo a identificação dos atributos que realmente conseguem trazer informações relevante para o modelo, torna-se importante ao pensar em colocar o modelo em produção. No caso desse trabalho, observa-se que os três parâmetros mais importantes são valor financiado, duração do financiamento e valor em conta corrente.

Figura 18 – Importância dos atributos - Modelo Random Forest



Fonte: Autor

4 CONCLUSÃO

A tarefa de classificação através da utilização de aprendizado de máquina é uma tarefa complexa que envolve várias etapas. Para que um modelo possa rodar de forma a estimar a melhor classe associada a uma série de atributos é preciso que se analise o banco de dados por diferentes perspectivas, seja a busca por dados faltantes, instancias repetidas, a necessidade ou não da normalização dos dados, a existência de outliers ou desbalanceamento dos dados. Cada etapa dessa pode ser adotado uma estratégia diferente, o que torna a análise de dados fascinante, uma vez que não existe um caminho único nem uma reposta exata.

No trabalho em questão foi apresentado como os vários desafios relatados acima podem ser tratados. Observe que um outro analista poderia, com os mesmos dados, analisa-los de outra forma e certamente chegaria à resultados diferentes. Um modelo que poderia ser colocado em produção, o Random Forest, com uma acurácia de aproximadamente 80%. Identificou-se também como a escolha de um modelo pobre pode levar à resultados que não sustentariam a implementação de um modelo de aprendizado de máquina. Identificou-se da mesma forma, a importância da seleção dos atributos, que não se faz necessário carregar todos quando for colocar o modelo em produção. Outro ponto destacado, ao observarmos a matriz de confusão, resulta que, por melhor que seja o modelo, existe um erro associado, seja do tipo I ou tipo II, e deve-se tentar trabalhar com informações de outras fontes e não apenas do modelo adotado para a tomada de decisão. Por fim, no caso estudado usando Random Forest, observamos que empresas de médio ou grande porte obteriam um lucro expressivo, ao gerir melhor os maus pagadores.

O trabalho tem por objetivo unicamente a classificação do cliente, como bom ou mau pagador. Uma proposta de trabalho futuro, seria a de mensurar o risco de cada cliente, e não apenas identifica-lo como bom ou mau pagador, de forma que se possa oferecer condições diferenciadas de crédito para cada um. Pode-se também buscar identificar clusters por empreendimentos, onde o perfil do cliente esteja associado a esse cluster, com isso diminuindo o risco de se obter um mau pagador. Acredito que, na base de dados atual, alguns maus pagadores podem existir apenas por estarem financiando empreendimentos que não são compatíveis a eles. A criação desses clusters tentaria solucionar essa questão.

REFERÊNCIAS

BAESENS, B. et al. Benchmarking state-of-the-art classification algorithms for credit scoring. **Journal of the operational research society**, Springer, v. 54, n. 6, p. 627–635, 2003.

DESAI, V. S.; CROOK, J. N.; JR, G. A. O. A comparison of neural networks and linear scoring models in the credit union environment. **European journal of operational research**, Elsevier, v. 95, n. 1, p. 24–37, 1996.

DJEUNDJE, V. B. et al. Enhancing credit scoring with alternative data. **Expert Systems with Applications**, Elsevier, v. 163, p. 113766, 2021.

FRIEDMAN, J. H. Multivariate adaptive regression splines. **The annals of statistics**, JSTOR, p. 1–67, 1991.

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.

HUANG, Z. et al. Credit rating analysis with support vector machines and neural networks: a market comparative study. **Decision support systems**, Elsevier, v. 37, n. 4, p. 543–558, 2004.

HUNG, C.; CHEN, J.-H. A selective ensemble based on expected probabilities for bankruptcy prediction. **Expert systems with applications**, Elsevier, v. 36, n. 3, p. 5297–5303, 2009.

KARELS, G. V.; PRAKASH, A. J. Multivariate normality and forecasting of business bankruptcy. **Journal of Business Finance & Accounting**, Wiley Online Library, v. 14, n. 4, p. 573–593, 1987.

MAKOWSKI, P. Credit scoring branches out. **Credit World**, v. 75, n. 1, p. 30–37, 1985.

REICHERT, A. K.; CHO, C.-C.; WAGNER, G. M. An examination of the conceptual issues involved in developing credit-scoring models. **Journal of Business & Economic Statistics**, Taylor & Francis, v. 1, n. 2, p. 101–114, 1983.

SCHEBESCH, K. B.; STECKING, R. Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. **Journal of the operational research society**, Taylor & Francis, v. 56, n. 9, p. 1082–1088, 2005.

THOMAS, L. C. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. **International journal of forecasting**, Elsevier, v. 16, n. 2, p. 149–172, 2000.

WANG, G. et al. Two credit scoring models based on dual strategy ensemble trees. **Knowledge-Based Systems**, Elsevier, v. 26, p. 61–68, 2012.

WEST, D. Neural network credit scoring models. **Computers & operations research**, Elsevier, v. 27, n. 11-12, p. 1131–1152, 2000.

ZHANG, D. et al. A comparison study of credit scoring models. In: IEEE. **Third International Conference on Natural Computation (ICNC 2007)**. [S.l.], 2007. v. 1, p. 15–18.

5 APENDICIES

`MLPClassifier` - hiper-parâmetros:

- `activation= relu`;
- `alpha=0.0001`;
- `batch size= auto`;
- `beta 2=0.999`;
- `early stopping=False`;
- `epsilon=1e-08`;
- `hidden layer sizes=(100,)`;
- `learning rate=constant`;
- `learning rate init=0.001`;
- `max fun=15000`;
- `max iter=500`;
- `momentum=0.9`;
- `n iter no change=10`;
- `nesterovs momentum=True`;
- `power t=0.5`;
- `random state=42`;
- `shuffle=True`;
- `solver='adam'`;
- `validation fraction=0.1`;

Para a regressão logística foi utilizada a função `LogisticRegression` disponível no Sklearn, com os seguintes hiper-parâmetros:

- `C=1.0`;

- class_weight=None;
- dual=False;
- fit intercept=True;
- intercept scaling=1;
- l1 ratio=None;
- max_iter=1000;
- multi_class=auto;
- penalty='l2';
- random$_state = 42$; solver=lbfgs;
- tol=0.0001;
- verbose=0;
- warm_start=False;

Para a Random Forest foi empregada a função `RandomForest` do Sklearn, com os seguintes hiper-parâmetros:

- bootstrap=True;
- ccp alpha=0.0;
- criterion=gini;
- min impurity decrease=0.0;
- intercept scaling=1;
- min samples leaf=1;
- min samples split=2;
- multi class=auto;
- penalty='l2';
- random state=42;
- n estimators=100;
- n jobs=-1;

- `random state=42`;
- `warm start=False`;