USO DE INTELIGÊNCIA ARTIFICIAL PARA CLASSIFICAÇÃO DO PADRÃO DE COMPRAS DE CLIENTES BANCÁRIOS

Amanda Isabela de Campos

Relatório referente à uma das avaliações da
disciplina COC786 – Inteligência Computacional

Professor: Alexandre Evsukoff

Rio de Janeiro
Outubro de 2020

1. Introdução:

1.1. Descrição do Problema

O presente trabalho tem como objetivo aplicar algoritmos de inteligência artificial em um conjunto de dados conhecido como *Bank Marketing Data Set* [1], dessa forma é possível prever ou classificar, a partir dos atributos, se um determinado cliente compra ou não o depósito a prazo ofertado por ligações. O depósito a prazo é um produto bancário como um investimento, onde o cliente compra uma parcela de algum fundo e a instituição de crédito o restitui com juro depois de um tempo acordado. O intuito dessa análise de dados é melhorar as campanhas e estratégias de marketing e encontrar os dados que indicam o sucesso do telemarketing bancário.

O conjunto de dados adotado para o presente trabalho está disponível no repositório da Universidade da Califórnia em Irvine (UCI) com o nome *Bank Marketing Data Set*, composto por 17 atributos para 11162 registros, com uma variável de saída que indica se o cliente efetuou um depósito a prazo. Sendo uma variável binária que tem valores de 'sim' ou 'não'. Os dados foram coletados de um banco português no período de 2008 a 2013. Esse conjunto de dados já foi adotado por Moro *et al.* [2] e Moro *et al.* [3] em 2014 para análises de classificadores com modelos de regressão logística, árvores de decisão, redes neurais e máquinas de vetores de suporte.

A proposta deste trabalho engloba ampliar as análises de Moro *et al*. e propor outros algoritmos de inteligência artificial para a classificação dos dados utilizando a técnica de validação cruzada para dividir o conjunto de dados para modelos de regressão logística, SVC (*C-Support Vector Classification*), KNN (*K Nearest Neighbor*), árvores de decisão, gradiente boosting, florestas aleatórias e Naive Bayes, e por fim comparar a acurácia dos modelos com os resultados existentes. A avaliação dos classificadores é feita a partir de análises das matrizes de confusão e da área sob a curva ROC (do inglês *receiver operating characteristic*), denominada AUC.

1.2. Pesquisa Bibliográfica

Segundo Evsukoff [4] o objetivo da classificação supervisionada, ou simplesmente classificação, é o desenvolvimento de um classificador, isto é, um modelo capaz de predizer a classe correta de um registro a partir dos valores das variáveis de entrada. Atualmente, existem diversos algoritmos de inteligência artificial que realizam a classificação supervisionada com alta confiabilidade para determinados dados.

O trabalho de Moro *et al*. [2] analisou o conjunto de dados *Bank Marketing Data* com os métodos de regressão logística, árvores de decisão, redes neurais e máquinas de vetores de suporte

obteve resultados de AUC = 0,8 com redes neurais, o que confirmou que o método é confiável e uma ferramenta valiosa para os gerentes de campanhas de telemarketing. Também foi possível detectar qual atributo tem maior influência no sucesso de vendas e propôs melhoras no método de vendas.

1.3. Apresentação da tecnologia

A análise e caracterização dos dados, bem como a aplicação dos modelos de inteligência artificial serão implementadas com a linguagem de programação Python, por ser: (i) uma linguagem de programação simples, livre e aberta; (ii) a linguagem mais usada atualmente e (iii) composta de várias bibliotecas desenvolvidas e em constante atualização já implementadas para aplicações de inteligência artificial. No presente trabalho adotou-se as bibliotecas Numpy [5] (para cálculos numéricos e operações com matrizes), Pandas [6] (para manipulação de dataset em formato de tabelas e planilhas), Matplotlib [7] (para a geração de gráficos e visualização dos dados), Seaborn [8] (também para a geração de gráficos, baseado no matplotlib porém mais voltado para estatística) e ScikitLearn [9] (biblioteca de inteligência artificial, com modelos já implementados de classificação, regressão, etc).

2. Caracterização / Visualização de Dados:

A Tab. 1 apresenta o nome, o tipo e a descrição de cada uma das 16 variáveis de entrada e a Tab. 2 apresenta estas características para a variável de saída da base de dados em estudo. Observa-se que das variáveis de entrada, sete são variáveis numéricas e oito são variáveis categóricas. Portanto a técnica de *one-hot-encoding* que transforma os atributos categóricos em numéricos (com cada variável como binária e em uma coluna) deverá ser aplicada antes da utilização dos modelos de classificação, uma vez que estes por padrão recebem como dados de entrada atributos numéricos. Com o pré-processamento foi observado que o *dataset* não possui valores ausentes, ou seja, os atributos de todas as linhas estão preenchidos.

Tabela 1. Descrição das variáveis de entrada

Índice	Variável de Entrada	Tipo de Variável	Descrição
1	Age	Numérica	Idade
2	Job	Categórica	Trabalho: "admin.", "desconhecido", "desempregado", "gerente", "empregada doméstica", "empresário", "estudante", "colarinho azul", "autônomo", "aposentado", "técnico", "serviços".
3	Marital	Categórica	Estado Civil: 'divorciado', 'casado', 'solteiro', 'desconhecido'; nota: 'divorciado' significa divorciado ou viúvo
4	Education	Categórica	Escolaridade: "desconhecido", "secundário", "primário", "terciário".

5	Default	Categórica	Incumprimento tem crédito em falta?: 'não', 'sim'
6	Balance	Numérica	Saldo: saldo médio anual, em euros
7	Housing	Categórica	Habitação, tem empréstimo de habitação?: 'não', 'sim'
8	Loan	Categórica	Empréstimo: tem empréstimo pessoal? 'não', 'sim'
9	Contact	Categórica	Contato: tipo de contato: "desconhecido", "telefone", "celular"
10	Day	Numérica	Último contato no dia do mês
11	Month	Categórica	Mês: último contato no mês do ano: 'jan', 'fev', 'mar', ..., 'nov', 'dez'
12	Duration	Numérica	Duração do último contato: Tempo em segundos
13	Campaign	Numérica	Campanha: número de contatos realizados durante esta campanha e para este cliente (inclui o último contato)
14	Pday	Numérica	Número de dias que passaram depois de o cliente ter sido contactado pela última vez de uma campanha anterior (-1 significa que o cliente não foi contactado anteriormente)
15	Previous	Numérica	Anterior: número de contactos realizados antes desta campanha e para este cliente
16	Poutcome	Categórica	Resultado da campanha de marketing anterior "desconhecido", "outro", "fracasso", "sucesso"

Tabela 2. Descrição da variável de saída

Índice	Variável de Saída	Tipo de Variável	Descrição
17	y	Binária	O cliente efetuou um depósito a prazo? 'sim', 'não'

A Fig. 1 indica as estatísticas básicas do conjunto de dados como: número de registros (*count*), média (*mean*), desvio padrão (*std*), valor mínimo (*min*), o percentil 25% (primeiro quartil), percentil 50% ou mediana (segundo quartil), o percentil superior a 75% (terceiro quartil), e o valor máximo (*max*), estes dados são importantes para descrever e entender a forma da distribuição dos conjuntos de dados e possíveis *outliers*.

	age	balance	day	duration	campaign	pdays	previous
count	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000
mean	41.231948	1528.538524	15.658036	371.993818	2.508421	51.330407	0.832557
std	11.913369	3225.413326	8.420740	347.128386	2.722077	108.758282	2.292007
min	18.000000	-6847.000000	1.000000	2.000000	1.000000	-1.000000	0.000000
25%	32.000000	122.000000	8.000000	138.000000	1.000000	-1.000000	0.000000
50%	39.000000	550.000000	15.000000	255.000000	2.000000	-1.000000	0.000000
75%	49.000000	1708.000000	22.000000	496.000000	3.000000	20.750000	1.000000
max	95.000000	81204.000000	31.000000	3881.000000	63.000000	854.000000	58.000000

Figura 1. Estatísticas básica do conjunto de dados

2.1. Análise da matriz de correlações

Uma forma de analisar a relação entre as variáveis em um conjunto de dados é construir a chamada matriz de correlação, onde nos eixos verticais e horizontais estão dispostas as variáveis e em cada ponto de plano cartesiano é a relação entre as variáveis do eixo horizontal com a variável do eixo vertical, portanto, é uma matriz quadrada com diagonal principal toda igual a 1 e simétrica. Na Fig. 2 está indicado a matriz de correlação para o conjunto de dados em estudo (apenas com as variáveis numéricas), onde cada número indica coeficiente de correlação entre duas variáveis. Observa-se que o maior valor de correlação está entre as variáveis *pdays* e *previous* com o coeficiente igual a 0.51. Todos os outros valores estão distantes de -1 ou 1, o que indica que as variáveis não são fortemente correlacionadas.

A Fig. 3 indica a mesma matriz de correlação, porém, neste caso, foi adotado um mapa de cores para melhor visualizar os resultados. Observa-se que a maioria dos valores estão proximos do azul que indica um coeficiente de correlação igual a zero, ou seja, as variáveis nestes casos não estão relacionadas ou têm uma correlação fraca. O que quer dizer que as informações não são redundantes.

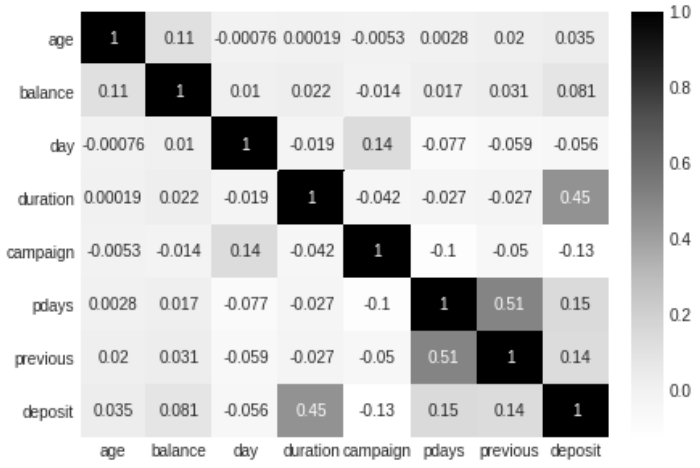


Figura 2. Matriz de correlação dos atributos numéricos em números

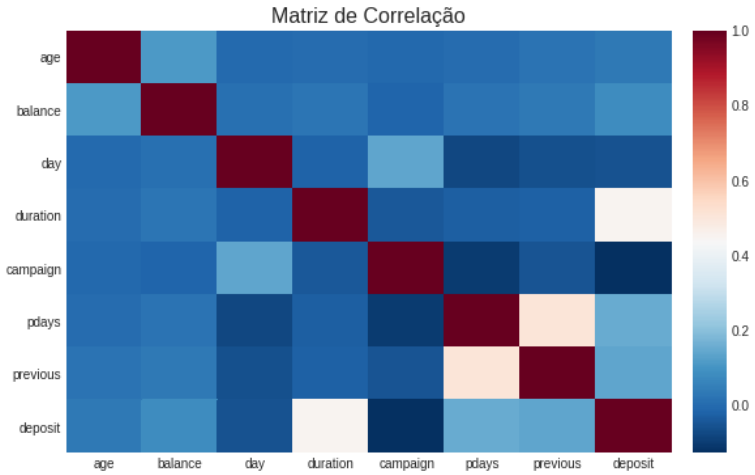


Figura 3. Matriz de correlação dos atributos numéricos com mapa de cores

Como apresentado na Tab. 1 o conjunto de dados possui outras variáveis que não são numéricas, neste caso foi aplicado o método *LabelEncoder* do *sklearn* para codificar os valores dos atributos categóricos em números inteiros. A matriz de correlação com todos os atributos do conjunto de dados está apresentada na Fig. 4. Onde observa-se que os atributos '*Poutcome*' e '*pdays'* estão com um coeficiente de correlação igual a -0,81 o que indica que as variáveis mudam em direções opostas, ou seja, quando uma aumenta a outra diminui e vice e versa. Para o problema destas variáveis indica que quanto maior o número de dias que se passaram desde o último contato com o cliente menor a chance de o resultado da campanha de marketing anterior ter sido um sucesso.

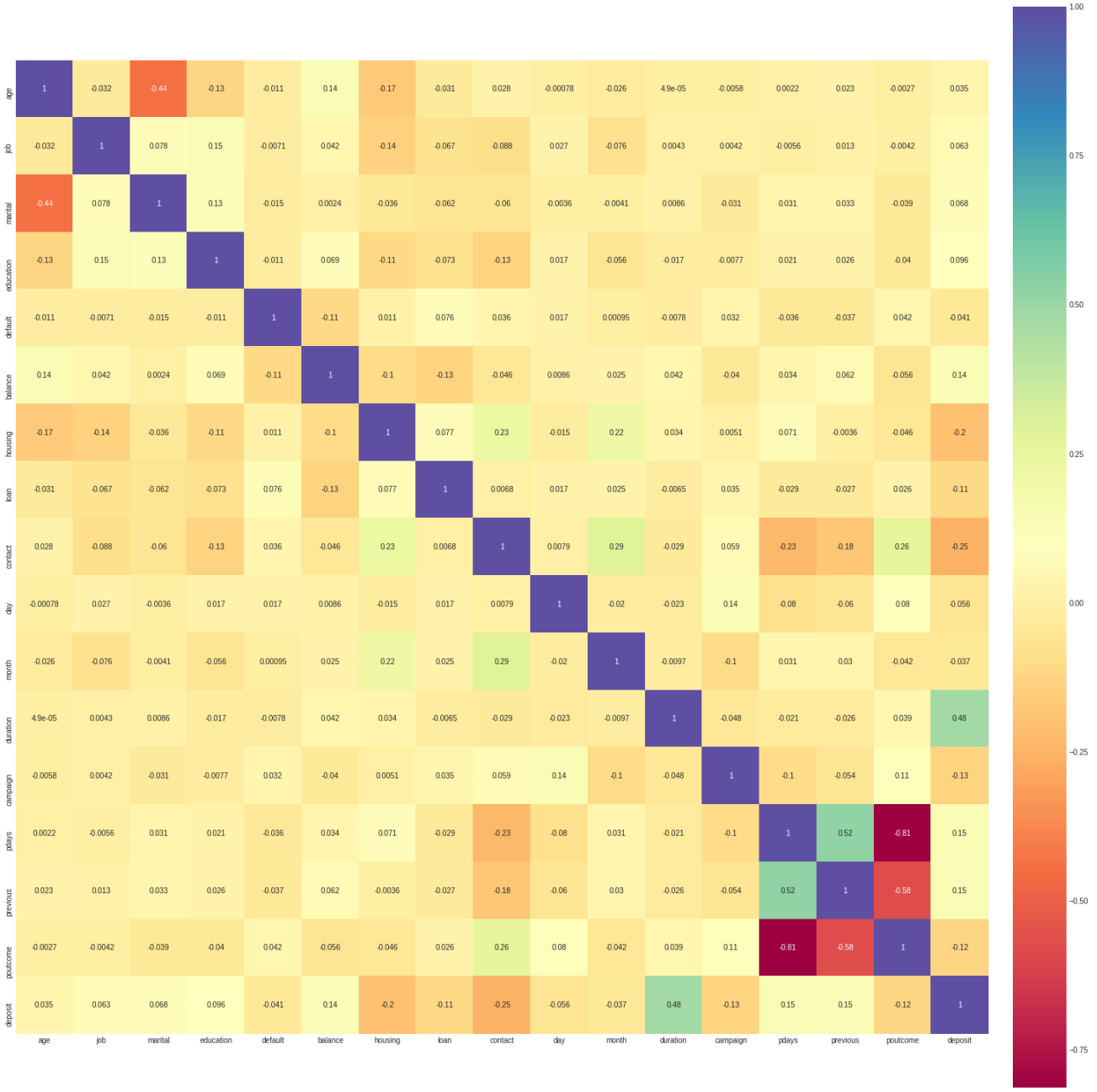


Figura 4. Matriz de correlação com todos os atributos

2.2. Análise de histogramas / box-plot

A Fig. 5 indica os histogramas de cada atributo numérico do conjunto de dados. Observa-se que os atributos *'age'* e *'day'* apresentam uma distribuição sem algum padrão enquanto os atributos possuem distribuições com padrões distorcidos à direita.

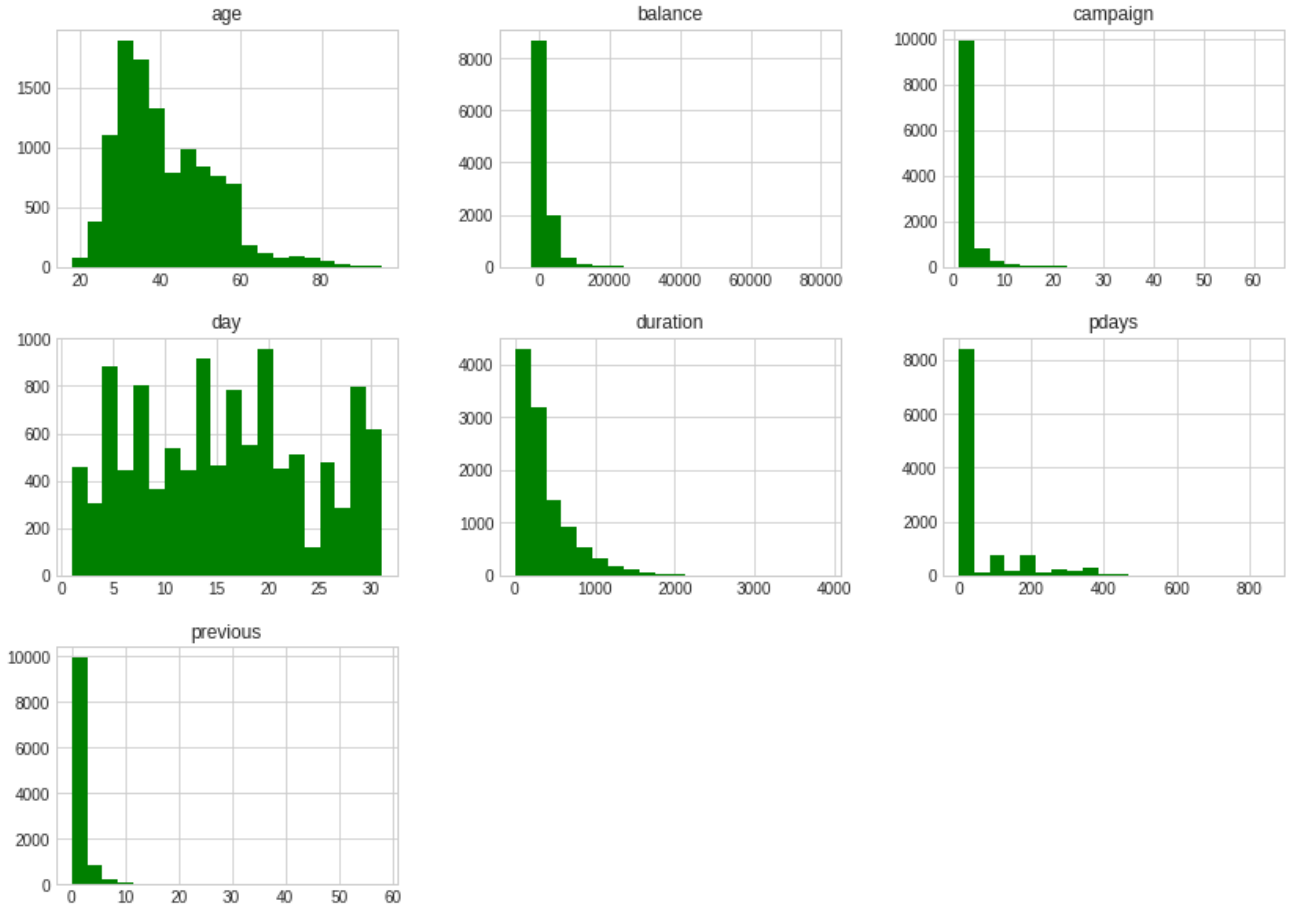


Figura 5. Histogramas dos atributos numéricos

Observa-se que os atributos '*pdays*', '*campaign*', '*previous*' possuem uma frequência alta para os primeiros valores do eixo *x*, como as primeiras barras dos histogramas para os três atributos. Esses pontos podem ser considerados *outliers*, após alguns cálculos é possível afirmar que apenas 0,5% dos valores de '*pdays*' são maiores que 500. Uma possível estratégia para reverter esse erro, seria substituir esses valores pela média dos dados que são menores que 500.

Para o atributo '*campaign*' que corresponde número de contatos realizados durante a campanha para determinado cliente calcula-se que apenas 0,07% são valores maiores que 40 ligações, números que provavelmente estão fora do esperado e classificam-se como outliers.

O atributo '*previous*' indica número de contatos realizados antes desta campanha e para este cliente e, portanto, valores maiores que 20 para esse problema podem ser classificados como valores

anormais, e foi observado que 0,13% dos registros do conjunto de dados em estudo apresentam valores do atributo '*previous*' maiores que 20.

Os dados identificados como fora dos valores esperados serão “melhorados” com a limpeza de dados como apresentado no item 2.4, usando a técnica das distancias que é mais confiável do que apenas observar distribuições de histogramas.

A Fig. 6 indica os histogramas dos atributos categóricos, pode-se dizer que todos os atributos apresentaram uma distribuição distorcida à direita, é importante salientar que nenhum registro apresenta valores ausentes ou incorretos neste caso, o que possibilitou a representação dos gráficos de histograma.

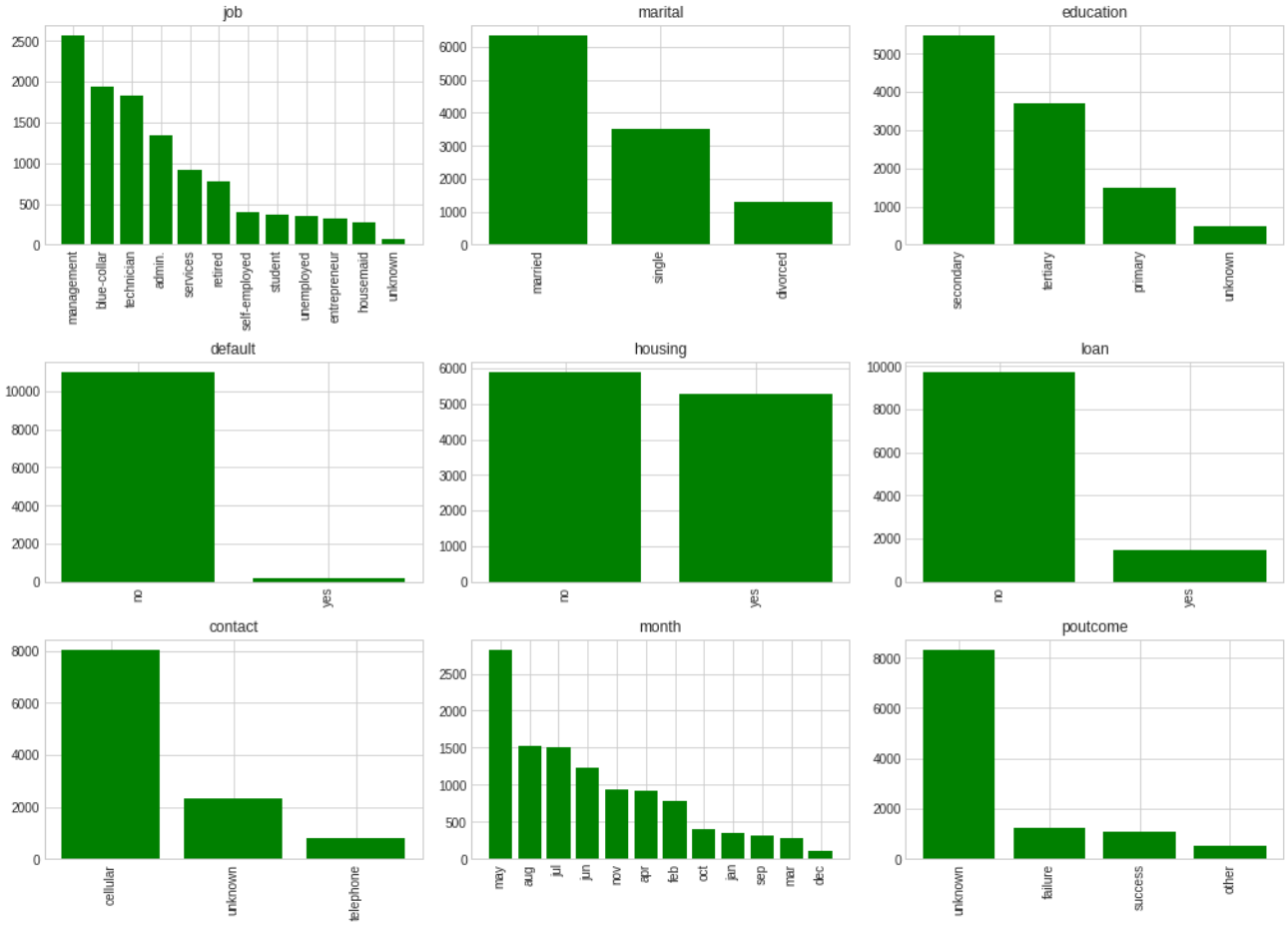


Figura 6. Histograma dos atributos categóricos

A Fig. 7 apresenta o histograma da variável de saída e observa-se que o problema é parcialmente balanceado.

Figura 7. Histograma da variável de saída

O chamado *box plot* é um tipo de gráfico usado para avaliar a distribuição dos dados, com base nas estatísticas básicas anteriormente calculadas na Fig. 1. Segundo Evsukoff [4] a construção do *box plot* segue o seguinte procedimento, de cima para baixo, o traço superior representa o limite superior, o primeiro traço da caixa é o valor do terceiro quartil (75%), seguido pelo segundo quartil ou mediana (50%) e o último traço da caixa é o valor do primeiro quartil (25%), o último traço é o valor do limite inferior. A Fig. 8 indica os *box plot* para os dados numéricos do *dataset* analisado neste trabalho.

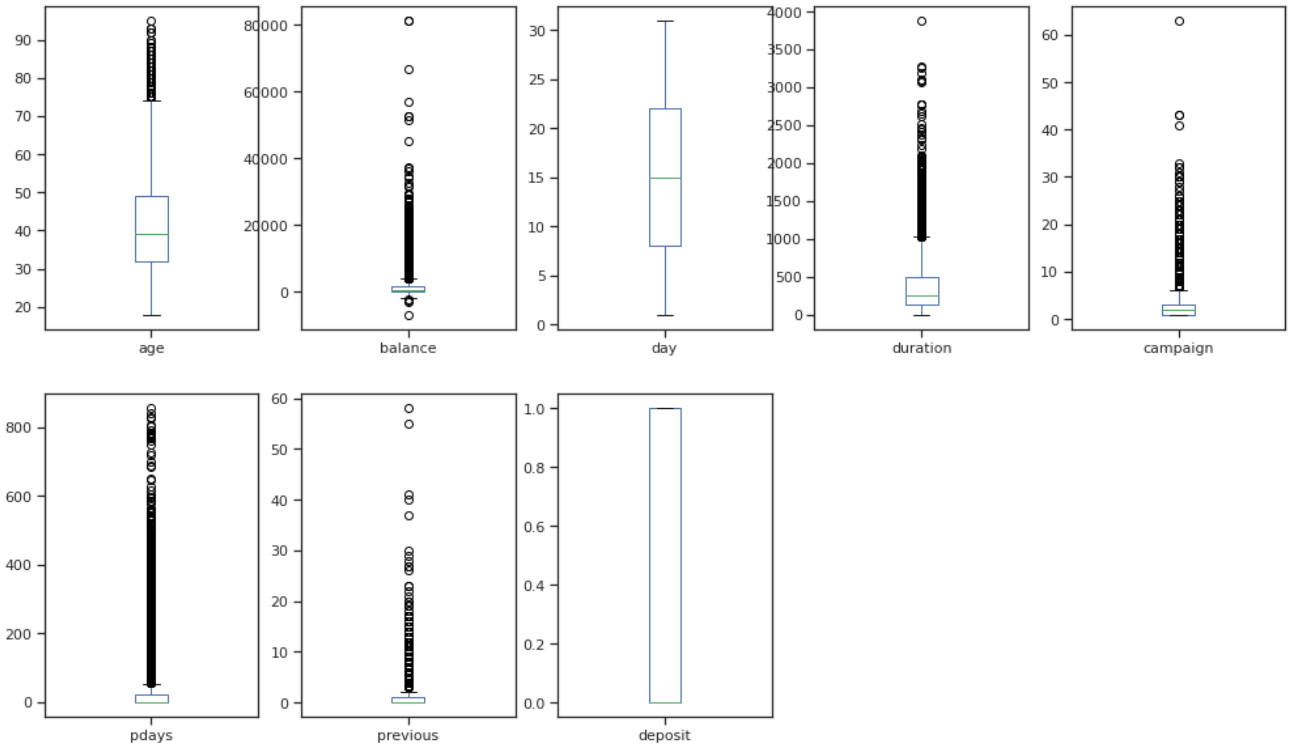


Figura 8. Box plot

A Fig. 10 apresenta um gráfico de projeção que também é um artificio para analisar a distribuição e correlação das variáveis de um conjunto de dados. Assim como a matriz de correlação, este tipo de gráfico também indica a combinação de todas as variáveis duas a duas, sendo, portanto, simétrico, porém neste caso, na diagonal principal estão os histogramas. Novamente observa-se que

não existem dados altamente correlacionados e novamente é detectado a existência de algumas variáveis com distribuições assimétricas.

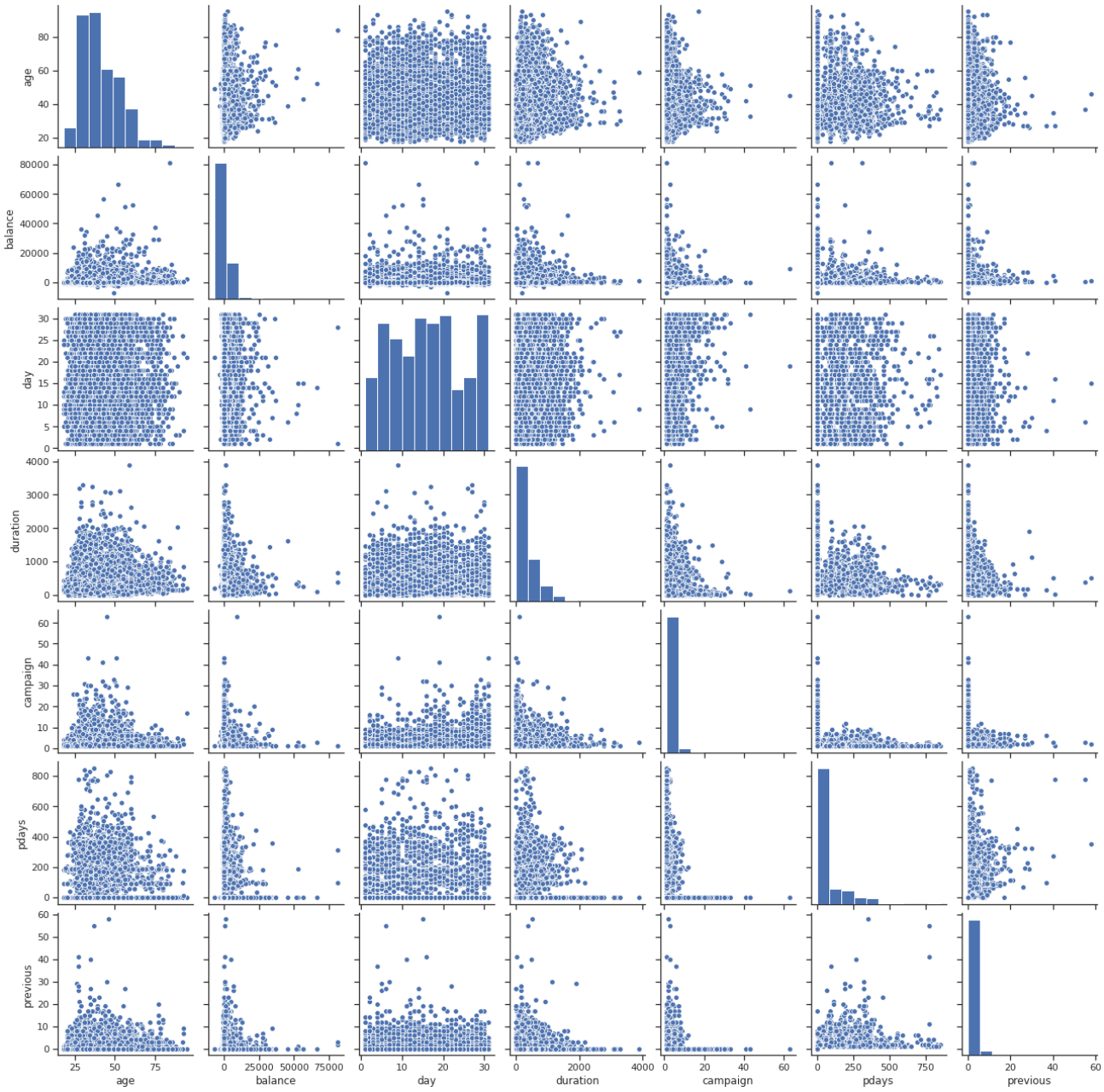


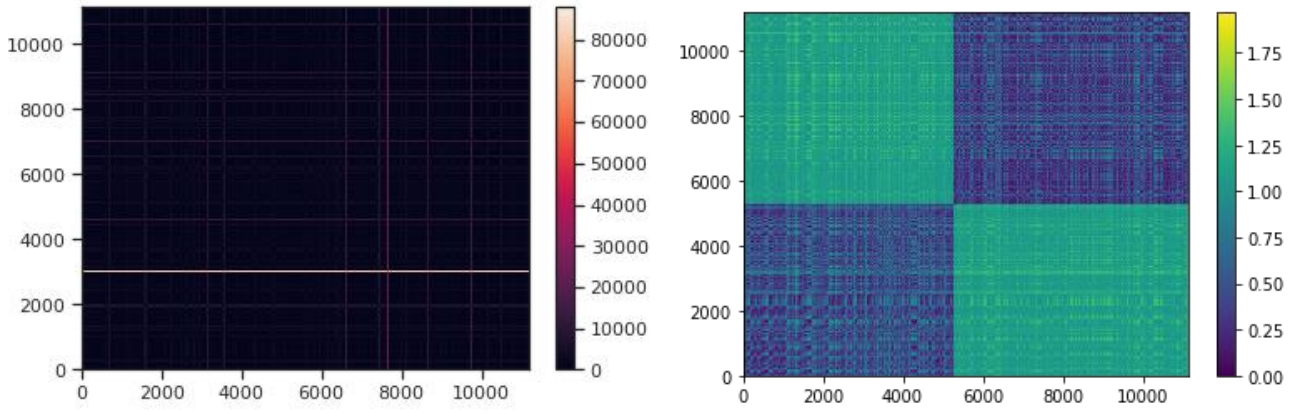
Figura 9. Gráfico de Projeção

2.3. Análise da matriz de distâncias

A matriz de distâncias para um conjunto de dados é uma matriz de dimensões: número de registros por número de registros, onde cada componente indica a distância ou similaridade entre estes registros, geralmente é apresentada como um mapa de cores e diferentes métricas de distâncias podem ser aplicadas, neste trabalho foi adotada a distância euclidiana, por ser a mais usual. Vale ressaltar que os dados são ordenados pela classe.

A Fig. 11 (a) indica a matriz de distâncias dos dados numéricos, observa-se que é um gráfico onde todas as cores pela escala são distâncias próximas de zero e em algumas linhas são valores de distâncias muito grandes, perto de 80000, isto indica que os dados não estão padronizados, ou seja, as variáveis não estão na mesma escala, o que também pode ser observado nos *box-plot*, onde as variáveis têm valores máximos e mínimos em escalas muito distantes, impedindo a correta interpretação da matriz de distâncias.

A solução para este problema será aplicar um método de limpeza dos dados com a padronização das variáveis, de forma que todos os registros estejam normalizados entre zero e um, esse procedimento de pré-processamento está disponível no *scikit learn* e é chamado de *MinMaxScaler*. A matriz de distancias para os dados padronizados está indicada na Fig. 11 (b) onde observa-se na escala de cores que as distâncias do gráfico são bem menores porque os dados foram padronizados e pelo gráfico é facilmente observado dois conjuntos de cores separados, indicando as duas classes do problema: o cliente efetuou (classe sim) ou não um depósito a prazo (classe não).



(a) (b)

Figura 10. Matriz de distâncias (a) com os dados não padronizados e (b) com os dados padronizados

2.4. Análise de outliers e valores ausentes

Os *outliers* podem ser definidos como valores que não fazem sentido e talvez foram obtidos erroneamente, sabe-se que estes prejudicam o desempenho dos modelos e podem indicar valores errados de predição. Uma forma de detectar *outliers* em um conjunto de dados é visualizar os dados após a padronização em um gráfico de distância média em função dos registros ordenados, como apresentado na Fig. 12, com este gráfico é possível observar o efeito da padronização (realizada no item anterior) na dispersão das variáveis. Analisando o gráfico é possível indicar um limite de corte para eliminar os valores com distâncias discrepantes, neste caso, considera-se que valores com distâncias médias maiores que 1.2 são *outliers*.

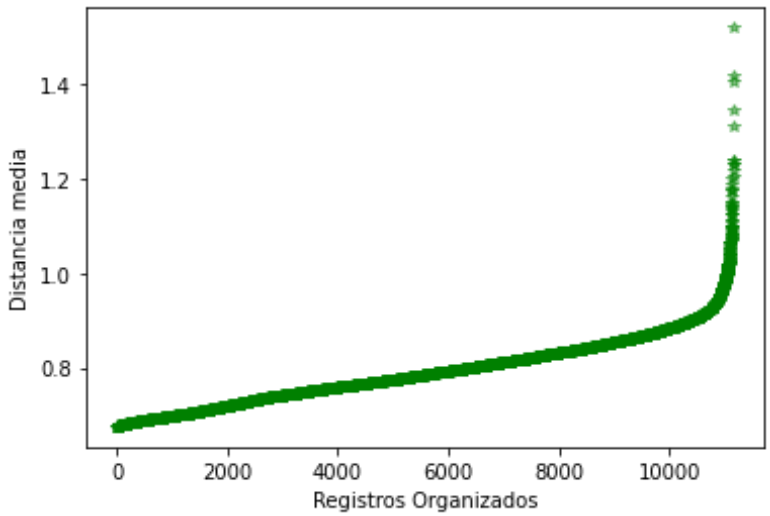


Figura 11. Detecção de outliers a partir da distância

A análise de valores ausentes também é uma etapa do pré-processamento muito importante, uma vez que valores faltantes atrapalham na execução dos modelos. Para o conjunto de dados em análise foi realizada a analise com o comando do Pandas [6] *df.isnull().sum()* que faz o somatório de valores ausente para cada variável e em todos os casos os valores foram zero, o que indica que todos os registros estão completamente preenchidos.

3. Metodologia:

A metodologia deste trabalho consiste em aplicar a técnica de validação cruzada [3] com 10 Folds (número de dobras) que é o mais usual, para selecionar os dados de teste e treinamento e aplicar modelos de classificação como (i) Naive Bayes, (ii) Regressão logística, (iii) SVM (máquina de vetores de suporte), (iv) KNN (*K Nearest Neighbor*), (v) Árvores de decisão, (vi) Florestas aleatórias e (vii) Gradiente boosting. A avaliação dos classificadores será feita a partir de análises das matrizes de confusão; da área sob a curva ROC (do inglês *receiver operating characteristic*), denominada AUC e outros índices de classificadores como Recall, Precisão e F1

3.1. Apresentação dos pré-processamentos realizados

Além da etapa de análise de outliers e valores ausentes como apresentado no item 2.4 os dados também precisaram de um tratamento nas variáveis categóricas. O conjunto de dados em estudo apresenta variáveis categóricas e os modelos de classificação só aceitam como entrada variáveis numéricas, é necessário transformar os atributos numéricos em categóricos (técnica conhecida como Categorical Encoder). Existem algumas diferentes maneiras de realizar essa tarefa, a mais comum e também adotada neste trabalho é a *OneHotEncoder* onde, cada variável categórica é mapeada para um

vetor que contém 1 e 0 denotando a presença ou ausência do recurso. O número de vetores depende do número de categorias para as características. Este método produz muitas colunas que retardam o aprendizado significativamente se o número da categoria for muito alto para a característica. No scikit-learn a função *OneHotEncoder* realiza este processo.

3.2. Descrição matemática dos modelos empregados

3.2.1. Naive Bayes

Um classificador Naive Bayes é um simples classificador probabilístico baseado na aplicação do teorema de Bayes (Eq. 1) que consegue prever a pertinência de um evento a uma dada classe a partir das probabilidades condicionais [12]. Onde $P(C_i|\mathbf{x})$ é a probabilidade a posteriori, ou seja a probabilidade de se observar a classe C_i dado as observações das variáveis *x*; $p(\boldsymbol{x}|C_i)$ é a probabilidade condicional, ou seja a probabilidade das variáveis $\boldsymbol{x}$ ocorrerem dado que a classe é C_i; $P(C_i)$ é a probabilidade a priori, ou seja, é a probabilidade da classe C_i sem nenhuma observação e $p(\boldsymbol{x}) = \sum_{i=1}^{m} p(\boldsymbol{x}|C_i)P(C_i)$. O classificador de Naive Bayes, portanto segue o teorema de Bayes para criar uma regra de decisão.

$$P(C_i|\mathbf{x}) = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{p(\boldsymbol{x})} \quad (1)$$

Apesar de suas suposições aparentemente simplificadas demais, os classificadores Bayes ingênuos têm funcionado bastante bem em muitas situações do mundo real, famoso pela classificação de documentos e filtragem de spam. A aplicabilidade deve-se ao fato de serem algoritmos que exigem uma pequena quantidade de dados de treinamento para estimar os parâmetros necessários [13].

3.2.2. Regressão Logística

O modelo de regressão logística é um modelo de classificação linear utilizado para estivar a probabilidade de uma instância pertencer a uma determinada classe [11]. Se esta probabilidade forma maior que 50% então o modelo prevê que a instancia pertence à classe positiva (rotulada como 1), caso contrário o modelo prevê que pertence à classe negativa (rotulada como 0). No modelo de regressão logística a probabilidade estimada é calculada com a Eq. 2. E a logística, que transforma um número entre 0 e 1, é uma função sigmoide apresentada na Eq. 3.

$$\hat{p} = h_\theta(\boldsymbol{x}) = \sigma(\theta^T.\boldsymbol{x}) \quad (2)$$

$$\sigma(t) = \frac{1}{1+\exp(-t)} \quad (3)$$

A previsão do modelo é realizada com a Eq. 4, uma vez que tem-se a probabilidade estimada $\hat{p} = h_\theta(\boldsymbol{x})$ que a instancia pertence ou não a classe positiva.

$$\hat{y} = \begin{cases} 0 \; se \; \hat{p} < 0.5 \\ 1 \; se \; \hat{p} \geq 0.5 \end{cases} \tag{4}$$

O modelo de regressão logística é treinado de forma a definir o vetor do parâmetro θ para o modelo estimar altas probabilidades para instâncias positivas e baixas probabilidades para instâncias negativas (Eq. 5). E por fim essa função é minimizada com o Método do gradiente descendente para encontrar o mínimo global.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - \hat{p}^{(i)}\right)\right] \tag{5}$$

3.2.3. SVM

Uma máquina de vetores de suporte (SVM) é um modelo de classificação poderoso [11] que consiste em um método de preenchimento da via mais larga possível entre as classes (linhas paralelas da Fig. 1) definindo assim um limite de classificação que determina se determinada instância pertence ou não a uma classe. Na Fig. 1 limites de decisão ruins estão esquematizados por linhas tracejadas e a linha continua representa limites de decisão adequados porque conseguem separar as duas classes e fica o mais distante possível das instancias de treinamento [14]. No primeiro gráfico da Fig. 1 as linhas tracejadas representas péssimas escolhas de classificadores porque estes não conseguem separar as classes corretamente e os limites de decisão chegam tão perto instancias que provavelmente não funcionarão bem para novas classificações.

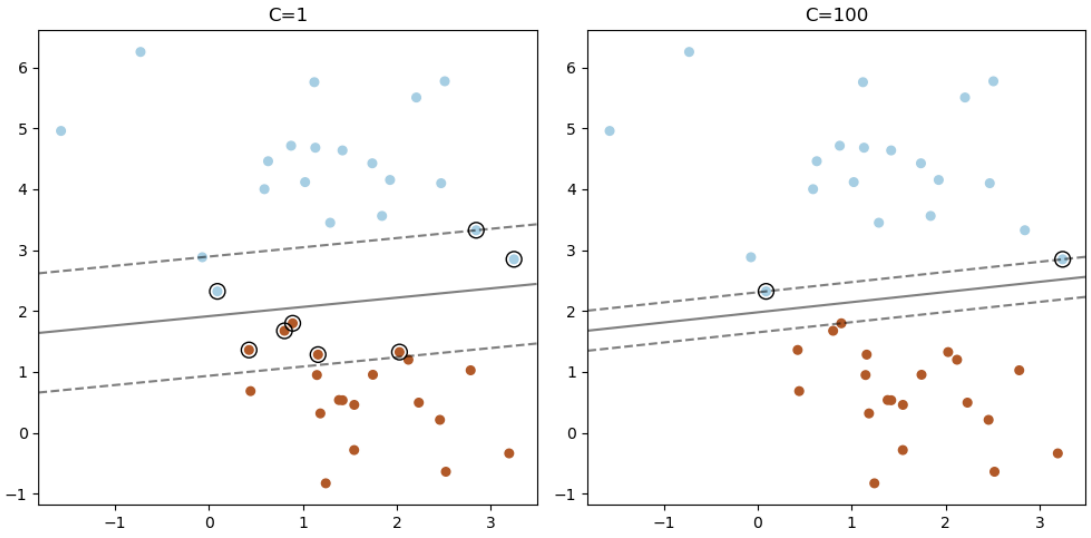


Figura 12. Exemplo de máquinas de vetores de suporte (SVM) [14]

3.2.4. KNN (K vizinhos mais próximos)

O método de classificação K vizinhos mais próximos (em inglês *K-Nearest Neighbour*) utiliza o conceito de semelhanças das características para determinação de uma classe. Nesse tipo de algoritmo a ideia geral é encontrar uma classe com base nos valores rotulados mais próximos [11].

A métrica de distância entre as partículas é dada por uma função de distância, que pode ser dada de várias formas, sendo a forma geral denominada como distância Minkowski de ordem *p*. Os vizinhos mais próximos geralmente são obtidos com a distância euclidiana, onde $p = 2$ [15]. A Fig. 13 apresenta um exemplo típico de uma classificação KNN para um problema de duas classes (ou seja, os círculos "rosa" e "azul") quando o parâmetro K (nº de vizinhos) é definido como "3" e "5". A estrela verde representa um ponto de amostra a ser classificado, que provavelmente será classificado como círculo "rosa".

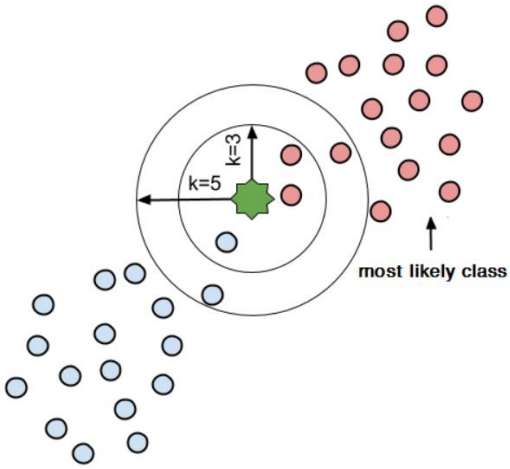


Figura 13. Um exemplo típico de uma classificação KNN para um problema de duas classes [15]

3.2.5. Gradient Boosting

O método de classificação Gradient Boosting é uma combinação de métodos de aprendizado "fracos" que resulta em um método forte, treinando vários modelos de árvores de decisão de forma gradual, aditiva e sequencial. A função de perda é uma medida que indica quão bons são os coeficientes do modelo na adequação dos dados subjacentes [16]. Dessa forma, depois de cada execução os algoritmos que obtiveram uma acurácia maior recebem um maior peso enquanto os que não conseguiram prever bem os valores recebem pesos menores, depois de várias iterações é obtido um algoritmo adaptado para prever a classe de determinado registro.

3.2.6. Árvore de decisão

Os modelos de árvore de decisão foram introduzidos por Breiman et al. [17]. Em linhas gerais o procedimento de árvores de decisão consiste na divisão do espaço i-dimensional, criado pelas *i* variáveis preditoras [18]. Segundo Garcia [19], as Árvores de Decisão são constituídas de nós, que representam os atributos, e de ramos, provenientes desses nós e que recebem as classes possíveis para esses atributos (cada ramo descendente corresponde a uma possível classe desse atributo). Nas árvores existem nós-folha (folha da árvore), que representam os diferentes valores de um conjunto de treinamento, ou seja, cada folha está associada a uma classe. Cada percurso na árvore (da raiz à folha) corresponde a uma regra de classificação. As Árvores de Decisão podem ser representadas como conjuntos de regras do tipo "se-então". As regras são escritas considerando o trajeto do nó raiz até uma folha da árvore.

3.2.7. Floresta aleatória

O modelo de Florestas Aleatórias consiste em combinar várias Árvores de Decisão. Em geral, as Florestas Aleatórias atingem boa acurácia preditiva quando comparadas a outros métodos de aprendizado de máquina supervisionados [20].

3.2.8. Rede Neural

O *Perceptron Multicamada* (MLP) é basicamente o empilhamento de vários Perceptrons (unidade linear com *threshold*). O MLP é comporto por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída (Fig 13). Os perceptrons de múltiplas camadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada com retropropagação de erro.

Segundo Falcão et al., 2013 para treinar as redes de Perceptron de Múltiplas Camadas é utilizado o algoritmo de *backpropagation* que consiste de duas fases: a propagação (*forward*) e a retropropagação (*backward*) de um conjunto de sinais através da rede. Na etapa de propagação existe a aquisição dos dados através da camada de entrada e sua propagação através de toda rede, gerando uma saída. Então a saída é comparada com a saída desejada e um valor de erro é calculado. Depois disso, inicia o processo de retropropagação, esse erro é propagado de volta à rede neural e utilizado para ajustar os pesos, buscando assim reduzir o máximo possível os erros a cada iteração para que o resultado se aproxime da saída desejada. Neste trabalho é adotado o modelo MLPClassifier e parâmetro alpha=1, com a biblioteca do *sklearn* [9].

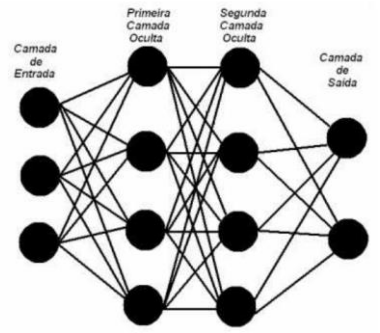


Figura 13. Arquitetura de RNA MLP com duas camadas ocultas. Fonte: Falcão et al., 2013

3.3. Descrição do procedimento experimental de avaliação dos modelos

Neste trabalho será aplicada a técnica de validação cruzada que consiste em dividir aleatoriamente o conjunto de treinamento em *K-folds* (neste caso cv = 10), ou seja, divide em 10 subconjuntos distintos e então treina e avalia o modelo 10 vezes escolhendo uma parte diferente de cada uma delas para avaliação e treinando nas outras 9 partes [11][21].

Esta técnica é amplamente empregada em problemas em que o objetivo da modelagem é a classificação. Busca-se então estimar o quão preciso é este modelo na prática, ou seja, o seu desempenho para um novo conjunto de dados.

A avaliação do desempenho de cada um dos modelos descritos no item 3.2 será realizada com estatísticas de avaliação como: *Recall, Precisão, F1, Acurácia* e *AUC.* Além da análise da matriz de confusão e curva ROC. A seguir será feita uma breve explanação de cada um destes métodos de avaliação de desempenho.

A matriz de confusão é o método mais indicado para avaliar o desempenho de um modelo de classificação [11]. Em problemas de duas classes a matriz é construída da seguinte forma: cada linha em uma matriz de confusão representa uma classe real e cada coluna representa uma classe prevista. Para um problema de duas classes, a matriz será 2x2 (Fig. 14) e a primeira posição representa o número de verdadeiro positivos, ou seja, valores que são verdadeiros e foram classificados como tal, porém a posição primeira linha e segunda coluna representa o número de falsos positivos, valores que correspondem a classe negativo e foram classificados como positivos, e assim respectivamente, a posição da segunda linha e primeira coluna corresponde aos falsos negativos, ou seja, pertencem a classe.

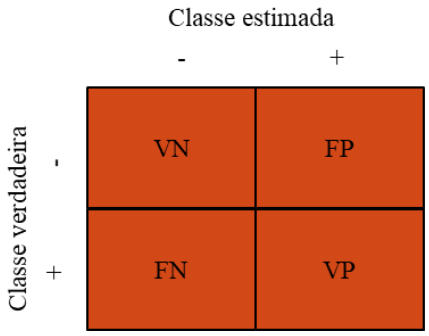


Figura 14. Matriz de confusão

Da matriz de confusão podem ser retiradas outras métricas como a acurácia das previsões positivas, também chamada de precisão do classificador, indicada na Eq. 6. Onde *VP* é o número de verdadeiros positivos e FP é o número de falsos positivos. A precisão é analisada em conjunto métrica revocação (Eq. 7), que pode ser entendida como a sensibilidade ou taxa de verdadeiros positivos, ou seja é a taxa de registros positivos que são corretamente classificadas. Onde *FN* é o número de falsos negativos. Na biblioteca *scikit learn* essas métricas são calculadas com a função precision_score (precisão) e recall_score (revocação).

$$precisão = \frac{VP}{VP+FP} \tag{6}$$

$$revocação = \frac{VP}{VP+FN} \tag{7}$$

Por fim, é comum combinar precisão e revocação em um único incide, chamado de pontuação F_1, que é a média harmônica entre as duas métricas apresentadas anteriormente (Eq. 8). No *scikit learn* essa métrica é calculada com o comando F1_score.

$$F_1 = \frac{2}{\frac{1}{precisão}+\frac{1}{revocação}} = \frac{VP}{VP+\frac{FN+FP}{2}} \tag{8}$$

Um modelo considerado ideal seria aquele que com alta precisão e baixa revocação, porém é impossível aumentar um e abaixar o outro, uma forma de encontrar o ponto ótimo é analisando o gráfico de compensação da precisão/revocação em função de valor do limiar. Dessa forma é possível encontrar um ponto de limiar onde a precisão e a revocação se cruzam e são as melhores possíveis para o problema.

A curva das características operacionais do receptor (ROC) é outra medida de desempenho de classificadores binários [11]. Consiste em um gráfico onde o eixo x é a taxa de falsos positivos (razão de instancias negativas incorretamente classificadas como positivas, igual a um menos a taxa de verdadeiros negativos) e o eixo y a taxa de verdadeiros positivos. Portanto a curva ROC é função da sensibilidade (revocação) versus a especificidade.

Um modelo de classificação adequado fica mais distante possível da linha tracejada e com a curva passando mais próximo possível do eixo vertical. A área abaixo da curva (AUC) também é analisada, de forma que quanto mais próximo de 1 melhor é o classificador. A acurácia neste trabalho é a média da acurácia obtida com a validação cruzada (em inglês *Crossval Mean Score*). Sabe-se que a acurácia para classificadores não é considerada a melhor métrica de desempenho, principalmente quando se analisa com conjuntos de dados desbalanceados, porem está também será calculada.

4. Resultados

A seguir estão ilustrados os resultados de matriz de confusão e curva ROC para todos os modelos de classificação em análise. E por fim uma comparação de todas as avaliações de desempenho calculadas para todos os modelos.

4.1. Resultados dos modelos lineares

4.1.1. Naive Bayes

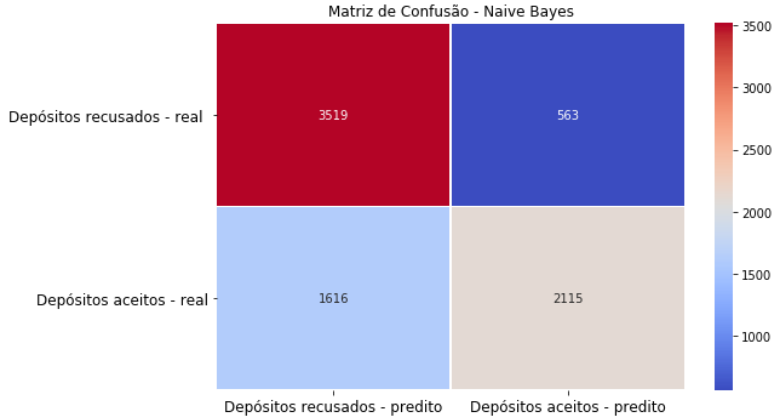


Figura 15. Matriz de confusão para o modelo de Naive Bayes

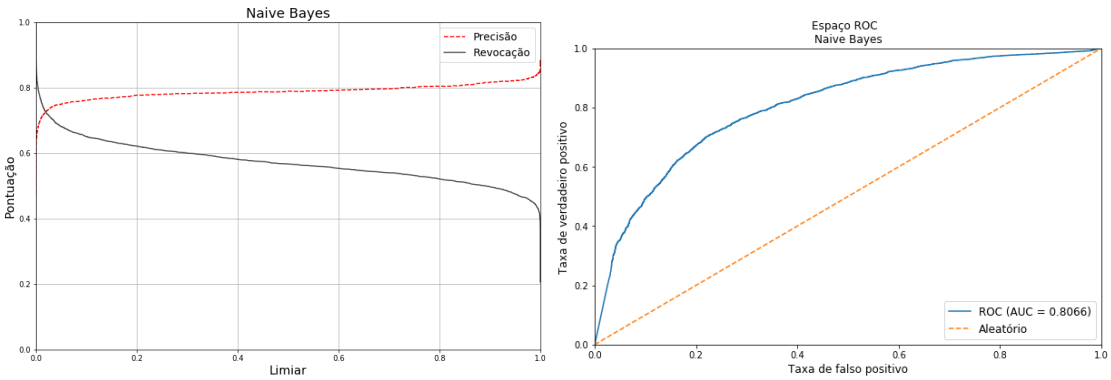


Figura 16. Compensação da precisão/revocação e curva ROC

4.1.2. Regressão Logística

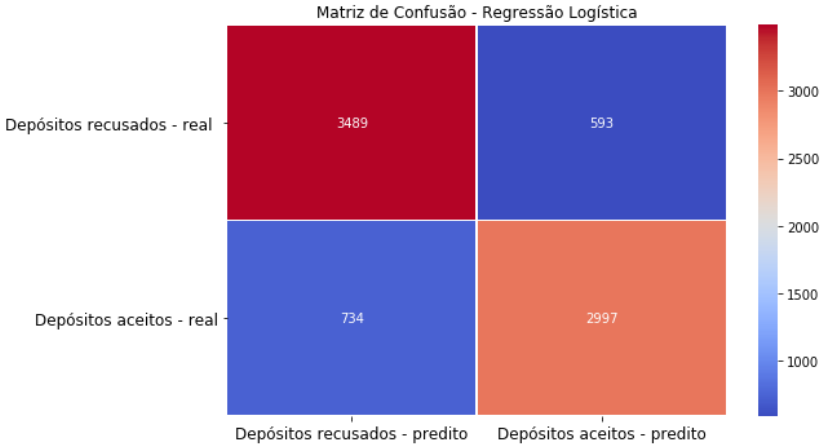


Figura 17. Matriz de confusão para o modelo de Regressão Logística

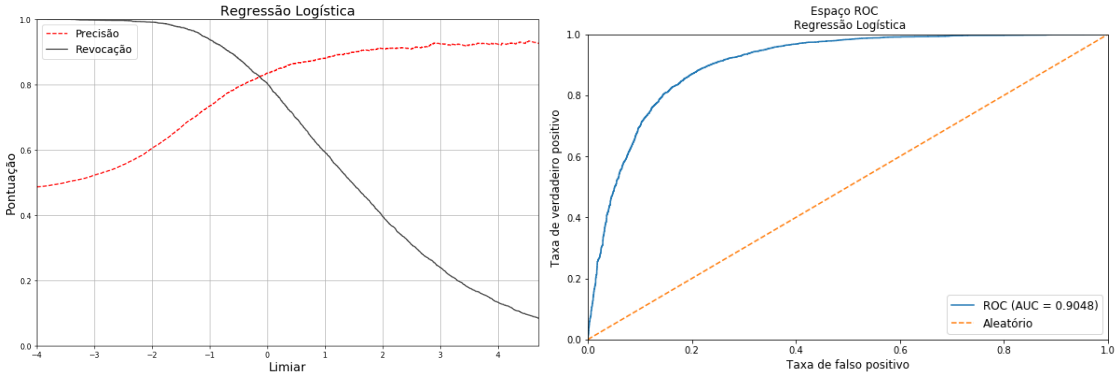


Figura 18. Compensação da precisão/revocação e curva ROC para Regressão Logística

4.1.3. SVM

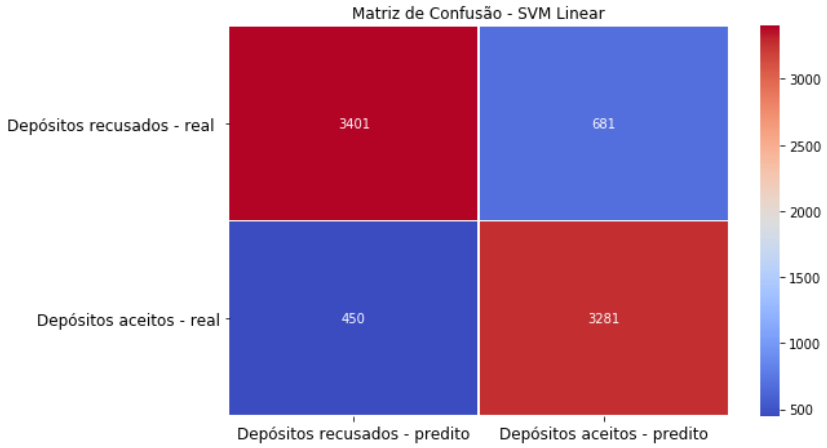


Figura 19. Matriz de confusão para o modelo de SVM Linear

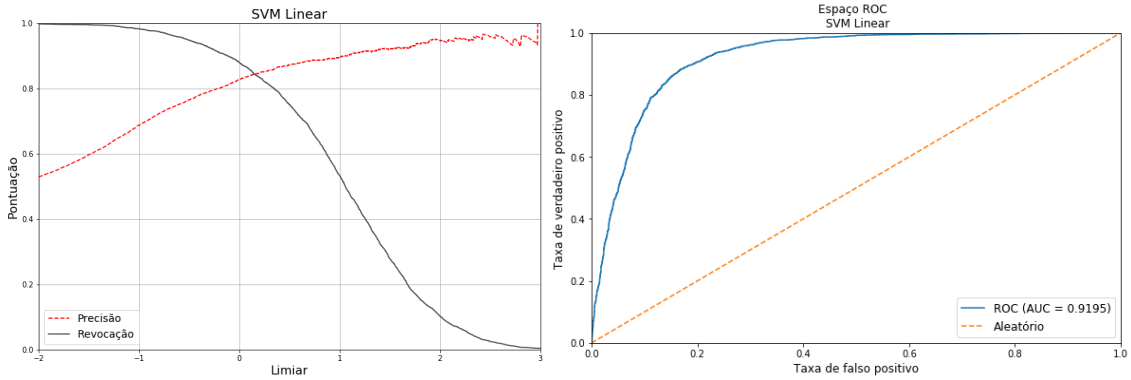


Figura 20. Compensação da precisão/revocação e curva ROC para o modelo de SVM Linear

4.2. Resultados de modelos não lineares

4.2.1. KNN

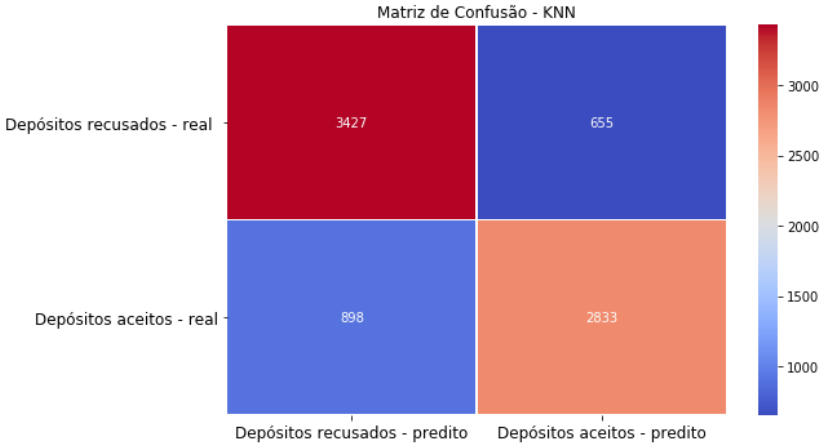


Figura 21. Matriz de confusão para o modelo de K vizinhos mais próximos KNN

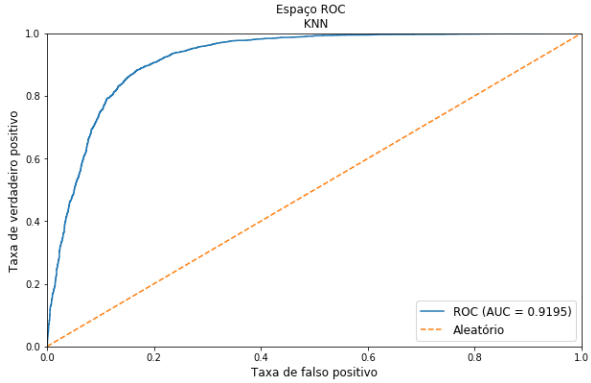


Figura 22. Curva ROC para o modelo KNN

4.2.2. Gradient Boosting

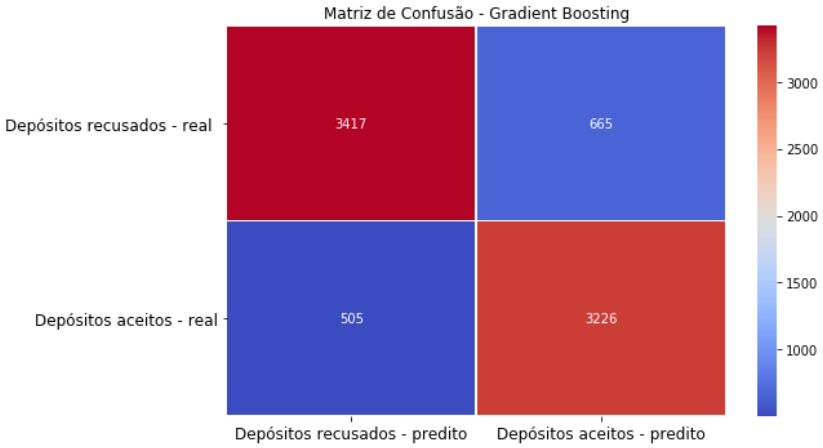


Figura 23. Matriz de confusão para o modelo de Gradient Boosting

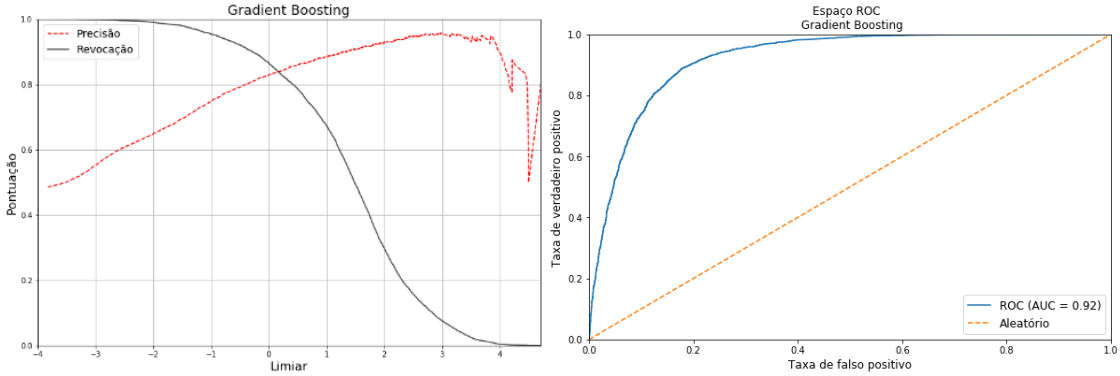


Figura 24. Compensação da precisão/revocação e curva ROC para o Gradient Boosting

4.2.3. Árvore de decisão

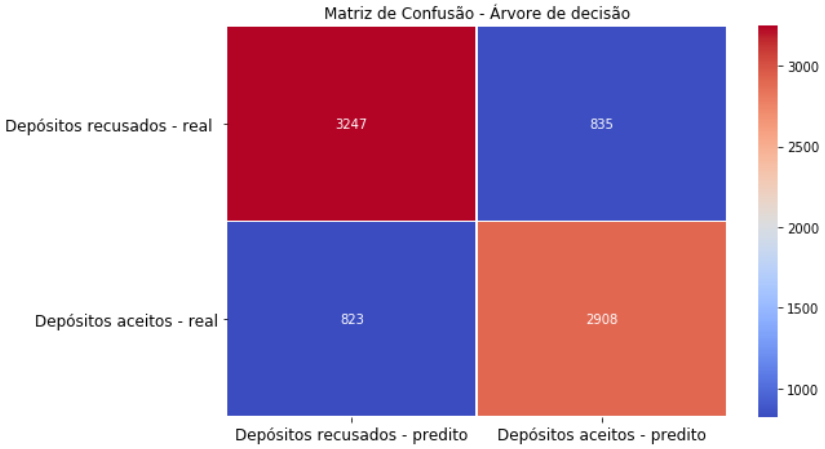


Figura 25. Matriz de confusão para o modelo de Árvore de decisão

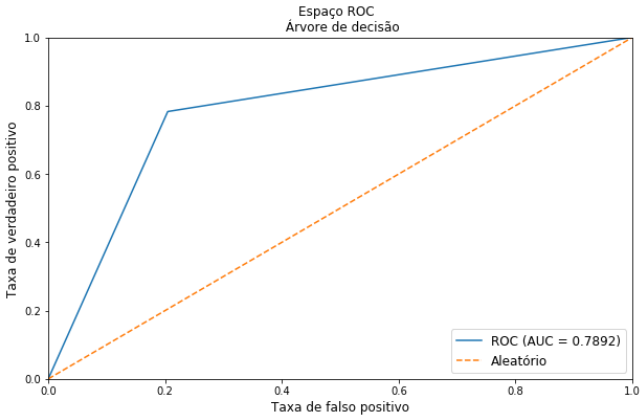


Figura 26. Curva ROC para o modelo de árvore de decisão

A Fig. 27 apresenta a importância de cada atributo no modelo para o modelo de árvore de decisão, isto porque sabe-se que quanto mais características, mais provavelmente o modelo irá sofrer superajuste (*overfitting*). Observa-se que o atributo "default" apresentou a menor influência no modelo e provavelmente poderia ser retirado das análises sem perdas significativas de resultados. O contrário ocorre com o atributo “duration” e “contact”, que tem elevada influência no modelo e não podem ser excluídas das análises.

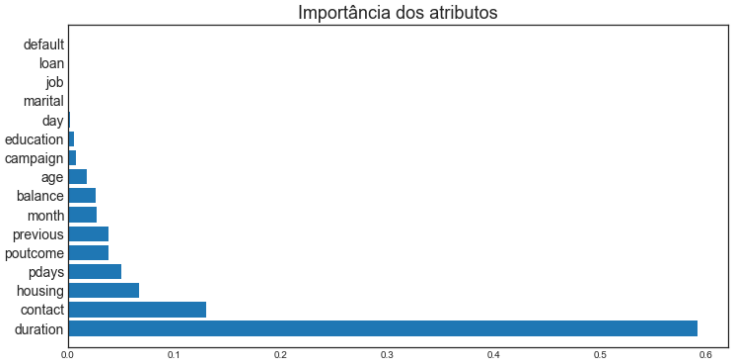


Figura 27. Importância dos atributos no modelo de árvore de decisão.

A Fig. 28 indica a árvore de decisão para o conjunto de dados, a previsão do valor final funciona da seguinte forma: suponha que deseja prever o se o cliente realizou ou não a compra de um determinado registro do conjunto de dados. Começando pelo nó da raiz (profundidade 0, na parte superior): este nó pergunta se a duração da ligação é menor do que 206.5. Se for, então o processo é deslocado para o nó filho esquerdo da raiz (profundidade 1, esquerda). Caso contrário, é deslocado para a direita, essa sequência é repetida até encontrar-se um nó da folha (que não tem nenhum nó filho), este irá indicar a classe (sim ou não) predita pelo modelo.

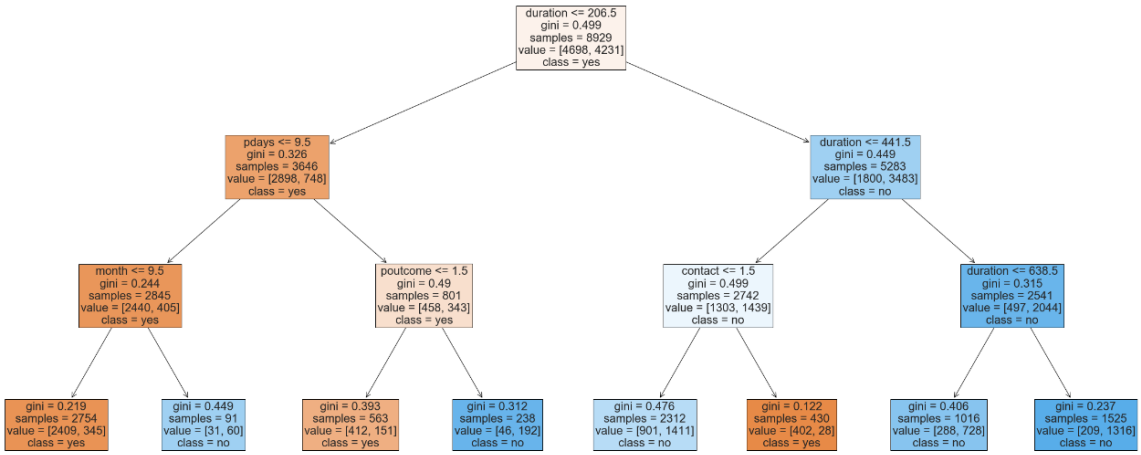


Figura 28. Árvore de decisão do conjunto de dados com profundidade igual a três

A árvore de decisão (Fig.29) indica uma árvore com profundidade igual a 4, onde observasse que mais atributos são avaliados o que dá maior precisão para o modelo, sabe-se que o quanto mais profunda é a arvore mais preciso serão seus resultados, porem plotar uma arvore com profundidade igual a 10, como foi usada no modelo fica inviável em termos de visualização.

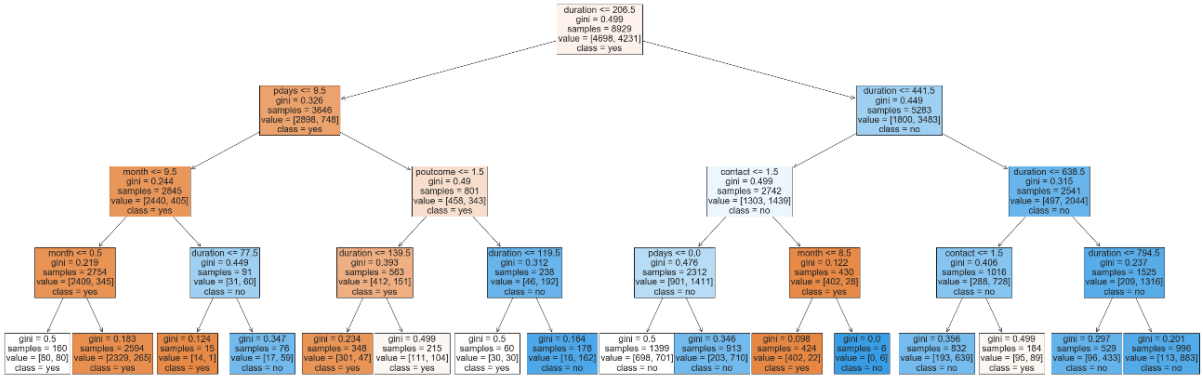


Figura 29. Árvore de decisão do conjunto de dados com profundidade igual a quatro

4.2.4. Floresta aleatória

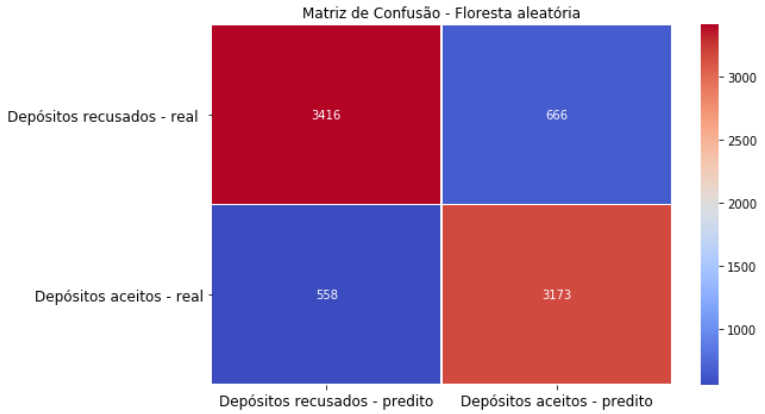


Figura 30. Matriz de confusão para o modelo de Floresta aleatória

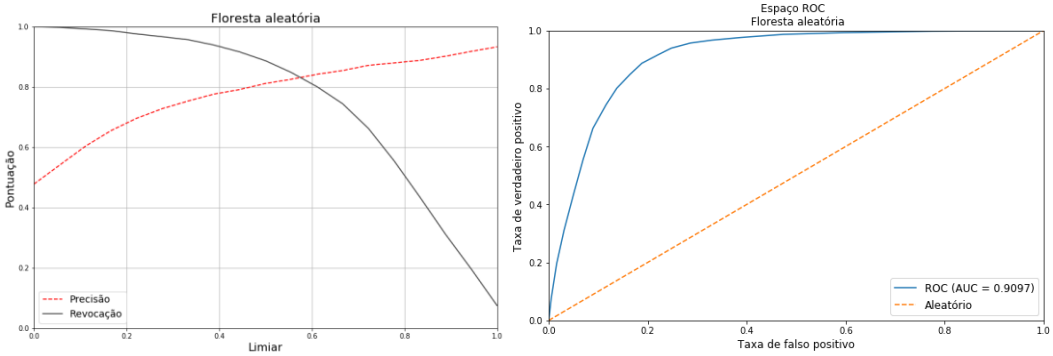


Figura 31. Compensação da precisão/revocação e curva ROC para a Floresta aleatória

4.2.5. Rede Neural

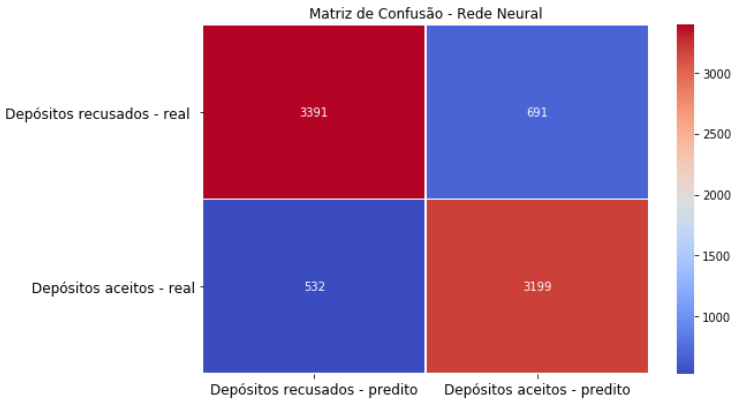


Figura 32. Matriz de confusão para o modelo de Redes Neurais

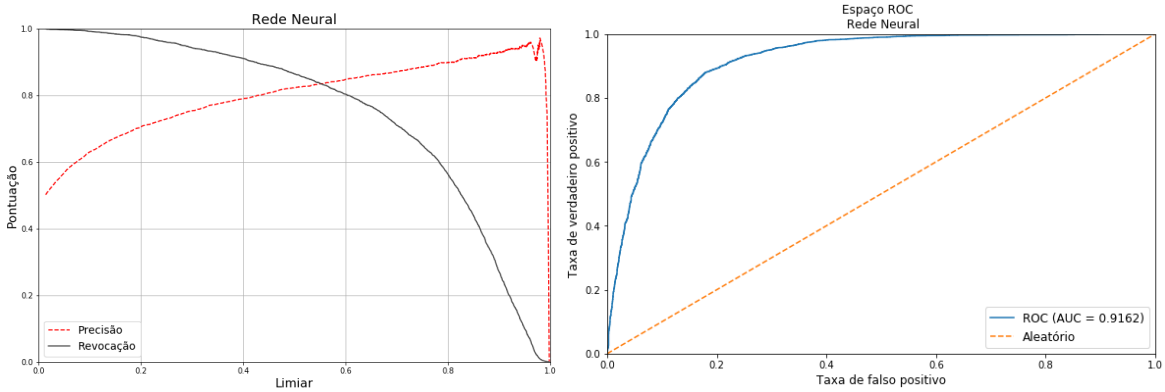


Figura 33. Compensação da precisão/revocação e curva ROC para a Rede Neural

4.3. Comparação / discussão dos resultados

Os modelos de classificação apresentados no item 3.2 foram aplicados no conjunto de dados em estudo, aplicando a técnica de validação cruzada com *cv = 10*, os resultados das estatísticas de validação *Recall, Precisão, F1, Acurácia* e *AUC* foram comparados, como indicado na Tab. 3.

Tabela 3. Comparação entre os modelos de classificação

Modelo	Recall	Precisão	F1	Acurácia	AUC
Regressão Logística	0.8033	0.8348	0.8187	0.8302	0.9048
SVM	0.8794	0.8281	0.8530	0.8552	0.9195
KNN	0.7593	0.8122	0.7849	0.8012	0.9195
Gradient Boosting	0.8646	0.8291	0.8465	0.8505	0.9200
Árvore de decisão	0.7794	0.7769	0.7782	0.7878	0.7892
Floresta aleatória	0.8504	0.8265	0.8383	0.8433	0.9097
Rede Neural	0.8574	0.8224	0.8395	0.8456	0.9162
Naive Bayes	0.5669	0.7898	0.6601	0.7211	0.8066

5. Conclusões

No presente trabalho dedicou-se uma certa cautela à etapa de pré-processamento com retiradas de *outliers* e tratamento dos dados, acredita-se que por isso os resultados dos modelos foram satisfatórios. Pode-se concluir que o pré-processamento tem papel fundamental na qualidade final dos resultados.

Com a execução de todos os modelos de classificação propostos neste trabalho pode-se afirmar que o modelo mais indicado para este conjunto de dados é o *Gradient Boosting*, uma vez que foi o modelo que apresentou maior área sob a curva ROC (AUC) que mede a capacidade de um modelo classificar corretamente um dado, este, que é o melhor índice de avaliação para modelos de classificação de duas classes. Cabe ressaltar que a complexidade do modelo não necessariamente está relacionada à um melhor resultado, isto porque o modelo de modelo de classificação linear SVM também se mostrou um ótimo classificador para esse conjunto de dados.

O trabalho de Moro et al. [2] analisou o conjunto de dados com modelos de regressão logística, árvores de decisão, redes neurais e SVM e obteve um valor para área sob a curva ROC: AUC = 0,8 com redes neurais, os resultados encontrados neste trabalho se mostram melhores, acredita-se que essa melhora foi devido a retirada dos *outliers* com distância discrepante. Os modelos de classificação ainda poderiam ser melhorados com estudos na busca dos parâmetros para cada modelo com a técnica de *Grid Search*, essa recomendação é dada para trabalhos futuros.

Referências

[1] UCI, Machine Learning Repository. Bank Marketing Data Set. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>. Acesso em: 10 de ago. de 2020.

[2] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014.

[3] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011.

[4] Evsukoff, Alexandre. Inteligência computacional: Fundamentos e aplicações [recurso eletrônico] / Alexandre Evsukoff. - 1. ed. – Rio de Janeiro: E-papers, 2020.

[5] NumPy v1.19 Manual Documentation. Disponível em: <https://numpy.org/doc/stable/reference/>. Acesso em: 17 de ago. de 2020.

[6] Pandas v1.11 Documentation. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 17 de ago. de 2020.

[7] Matplotlib: Visualization with Python v3.3.1. Disponível em: <https://matplotlib.org/contents.html>. Acesso em: 17 de ago. de 2020.

[8] Seaborn: Statistical data visualization. Disponível em: <https://seaborn.pydata.org/tutorial.html>. Acesso em: 17 de ago. de 2020.

[9] ScikitLearn: Machine Learning in Python v0.23.2. Disponível em: <https://scikit-learn.org/stable/user_guide.html>. Acesso em: 18 de ago. de 2020.

[10] MCKINNEY, Wes. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.", 2012.

[11] Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.

[12] Granik, Mykhailo, and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier." 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). IEEE, 2017.

[13] Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004.

[14] Linear Support Vector Classification. Scikit-learn, 2007. Disponível em: <https://scikit-learn.org/stable/auto_examples/svm/plot_linearsvc_support_vectors.html#sphx-glr-auto-examples-svm-plot-linearsvc-support-vectors-py >. Acesso em: 20 de set. de 2020.

[15] Haji Samadi, Mohammad Reza. (2015). Eye Tracking with EEG life-style.

[16] Friedman, Jerome H. "Stochastic gradient boosting." Computational statistics & data analysis 38.4 (2002): 367-378.

[17] L. Breiman, J. H. Friedman, R. A. Olshen, e C. J. Stone, Classification and Regression Trees, 1a Edição. Wadsworth: Routledge, 1984.

[18] R. L. de Morais, “Uso de Árvores Aleatórias para Classificação Sensorial de Arroz Cozido”, Bacharelado em Estatística, Universidade de Brasília (UnB), Brasília, 2017

[19] S. C. Garcia, “O Uso de Árvores de Decisão na Descoberta de Conhecimento na Área da Saúde”, Porto Alegre, Universidade Federal do Rio Grande do Sul (UFRGS), 2003

[20] M. Fernandez-Delgado, E. Cernadas, S. Barro, e D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?”, p. 49.

[21] J. P. Z. Cunha, “Um estudo comparativo das técnicas de validação cruzada aplicadas a modelos mistos”, Mestrado em Estatística, Universidade de São Paulo (USP), São Paulo, 2019.