UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Tratamento de dados desbalanceados em classificação binária com algoritmos em Python e aplicação em Marketing

Ramon Barbosa Rosa

Trabalho de Conclusão de Curso - MBA em Ciência de Dados (CEMEAI)

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Tratamento de dados desbalanceados em classificação binária com algoritmos em Python e aplicação em Marketing

Ramon Barbosa Rosa

USP - 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)

Barbosa Rosa, Ramon
B238t Tratamento de dados desbalanceados em classificação binária com algoritmos em Python e aplicação em Marketing / Ramon Barbosa Rosa; orientador Jorge L. Bazán. -- São Carlos, 2021.
76 p.

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

1. Ciência de Dados. 2. Dados desbalanceados. I. L. Bazán, Jorge, 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

RAMON BARBOSA ROSA

Tratamento de dados desbalanceados em classificação binária com algoritmos em Python e aplicação em Marketing

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 Ciência de Dados.

Área de concentração: Ciências de Dados

Orientador: Prof. Dr. Jorge L. Bazán

USP - São Carlos

2021

Esta página deve conter a ficha catalográfica e deve ser impressa no verso da folha de rosto. Para elaborar, acesse o endereço:
https://www.icmc.usp.br/institucional/estrutura-administrativa/biblioteca/servicos/ficha
ou procure um bibliotecário na Seção de Atendimento ao Usuário da Biblioteca do ICMC

FOLHA DE AVALIAÇÃO OU APROVAÇÃO

DEDICATÓRIA

Dedico este trabalho a todos os membros da comunidade de dados. São analistas, engenheiros e cientistas de dados, que após horas na frente do computador conseguem extrair um sinal de onde a maioria só ouve o ruído.

AGRADECIMENTOS

Agradeço em primeiro lugar à Deus, que me deu sabedoria e coragem para formar uma carreira a partir de meus talentos. Agradeço ao meu orientador, professor Jorge Bazán, que fez o que se espera de verdadeiro orientador, além de sempre se dispor a me auxiliar nas dificuldades durante a execução deste trabalho. Confesso que nem sempre consegui seguir à risca seus conselhos, de maneira que qualquer falha neste trabalho deve ser creditada a mim, somente. Por fim, agradeço à minha esposa e meu filho, que suportaram longas ausências minhas durante noites, madrugadas e fins de semana. A todos, meu muito obrigado!

EPÍGRAFE

“Nossa reconfortante convicção de que o mundo faz sentido repousa sobre uma base segura: nossa capacidade quase ilimitada de ignorar nossa ignorância.”

Daniel Kahneman (2011)

RESUMO

ROSA, R. B. **Tratamento de dados desbalanceados em classificação binária com algoritmos em Python e aplicação em Marketing**. 2020. 52 f. Trabalho de conclusão de curso (MBA em Ciência de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

Bases de dados desbalanceadas constituem-se em um desafio para modelagem em Machine Learning, em especial quando o desbalanceamento ocorre na variável resposta binária. Na ausência de qualquer tratamento prévio em bases nestas condições, os indicadores de performance e previsão tradicionais como a acurácia do modelo apresentam resultados que podem levar a conclusões enganosas acerca do melhor algoritmo para modelagem dos dados. Neste trabalho estudamos estas dificuldades propondo o uso de novos indicadores de performance de modelos e apresentamos soluções para tratamento do problema de desbalanceamento com algoritmos de reamostragem existentes tais como os de Sub-amostragem, sobre amostragem e ensemble classifiers com amostragem interna. Para exemplificar a aplicação destas técnicas, utilizamos um conjunto de dados originalmente desbalanceado, que tratamos inicialmente com uma série técnicas de pré-processamento e, em seguida, com cada um dos algoritmos de reamostragem disponíveis na linguagem Python. Como resultado obtivemos a confirmação da adequação destes procedimentos no tratamento de conjuntos de dados desbalanceados. Mais especificamente verificamos que o algoritmo Balanced Random Forest supera todos os demais em performance. Como corolário, verificamos que as métricas usuais de avaliação de performance de algoritmos nem sempre geram resultados satisfatórios quando se trata de processamento de dados desbalanceados.

Palavras-chave: Dados desbalanceados. Machine Learning. Python. Regressão Logística. Sub-amostragem. Sobre-amostragem. Ensemble Classifiers. Balanced Random Forest.

ABSTRACT

ROSA, R. B. **Treatment of unbalanced data in binary classification with algorithms in Python and application in Marketing**2020. 52 f. Trabalho de conclusão de curso (MBA em Ciência de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

Unbalanced databases are a challenge for modeling in Machine Learning, especially when the imbalance occurs in the variable binary response. In the absence of any previous treatment based on these conditions, traditional performance, and forecasting indicators such as the accuracy of the model present results that can lead to misleading conclusions about the best algorithm for data modeling. In this work, we study these difficulties by proposing the use of new model performance indicators and present solutions for addressing the imbalance problem with existing resampling algorithms such as the classes of Under-sampling, Over-sampling, and ensemble classifiers with internal sampling algorithms. To exemplify the application of these techniques, we used a data set originally unbalanced, which we initially treated with a series of pre-processing techniques and then with each of the resampling algorithms available in the Python language. As a result, we obtained confirmation of the adequacy of these procedures in the treatment of unbalanced data sets. More specifically, we found that the Balanced Random Forest algorithm outperforms al. others in performance. As a corollary, we found that the usual metrics of performance evaluation of algorithms do not always generate satisfactory results when it comes to processing unbalanced data.

Keywords: Imbalanced data. Machine Learning. Python. Logistic Regression. Under-sampling. Over-sampling. Ensemble Classifiers. Balanced Random Forest.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

No contexto da modelagem de dados utilizando técnicas de Machine Learning frequentemente nos deparamos com bases de dados em que uma das classes possui um número de amostras muito inferior em relação aos demais (HE, MA, 2013). O principal problema com o uso de bases desbalanceadas para geração de modelos é comprometimento da performance dos algoritmos padrão, já que, muitos deles, assumem uma distribuição equilibrada das classes na amostra. Em decorrência, os indicadores de performance dos modelos gerados através destes algoritmos apresentam resultados pouco representativos da realidade das bases estudas (HE, GARCIA, 2009). Além disso, a aplicação dos modelos resultantes nos conjuntos de teste irá sub representar a predição das classes (KUHN; JOHNSON, 2013). Diversas soluções têm sido desenvolvidas para lidar com problema do desbalanceamento de classes, sendo que as mais frequentes envolvem reamostrar os dados de maneira a tornar a proporção das classes mais equilibradas (BATISTA et al., 2004; PRATI et al., 2011).

Classes binárias representam situações em que a variável resposta de um fenômeno assume apenas dois possíveis valores, em geral, representados por 0, quando o fenômeno não ocorreu, e 1 para quando o fenômeno ocorrer (COX, 1972). Situações deste tipo ocorrem com frequência em diversos campos tais como Biomedicina, Finanças, Marketing, Economia etc.

1.2 Objetivos e contribuição deste trabalho

Os objetivos deste trabalho são: 1) Explicar os efeitos do desbalanceamento de classes binárias na modelagem de diversos fenômenos; 2) Estudar os procedimentos para tratamento de classes desbalanceadas utilizando diferentes algoritmos disponíveis em Python. Em particular, neste trabalho será dado foco às metodologias do paradigma frequentista que fazem ênfases de técnicas de amostragem tais como Over-Sampling e Under-Sampling; 3) Introduzir e comparar diferentes métricas de desempenho para avaliação dos algoritmos do item 2.

Especificamente, utilizaremos uma base de dados chamada Bank Marketing Data Set (MORO et al., 2014), disponível no UCI-Machine Learning Repository a qual apresenta uma proporção de 11 % de sucesso na variável de resposta binária.

Para este conjunto de dados utilizaremos as métricas tradicionalmente obtidos na Matriz de Confusão. Além disso, para melhor avaliar o desempenho dos algoritmos utilizaremos os

indicadores discutidos em HUAYANAY et al. (2019) como Índice de Jaccard, Gilbert Skill Score e Faith Index.

Esperamos, assim, ajudar a esclarecer o problema de desbalanceamento de dados, impactando positivamente o estudo de diversos fenômenos naturais, humanos e de negócios que apresentam a característica de produzirem classes desbalanceadas, tais como o estudo de doenças raras, previsão de falhas em engenharia e a prevenção de fraudes em seguros.

1.3. Estrutura do trabalho

Este trabalho está organizado como se segue: no capítulo 2 faremos a revisão da literatura relevante e mais recente sobre algoritmos para dados desbalanceados, em particular focando nos trabalhos que descrevem metodologias existentes na linguagem Python para classes binárias. Em seguida, descreveremos a metodologia utilizada, fazermos a análise descritiva das bases de dados, aplicaremos e avaliaremos os algoritmos para tratamento de dados.

2 REVISÃO BIBLIOGRÁFICA

2.1 Dados não balanceados

Em geral, algoritmos de Machine Learning utilizados para problemas de classificação são preparados para receber como entrada conjuntos de dados equilibrados. No entanto, em diversas situações, pode ocorrer desbalanceamento dos dados tanto no caso de variáveis dependentes categóricas, quanto no caso da variável resposta. Por sua vez, esse desequilíbrio pode ocorrer tanto quando a variável resposta possui duas classes (classificação binária) como na forma de múltiplas classes. Neste trabalho estamos interessados em avaliar o desbalanceamento de classes binárias na variável resposta. Para tanto, adotamos a definição proposta por DA SILVA et. Al (2020), que se segue: Seja Y uma variável de resposta binária com distribuição de Bernoulli, com o parâmetro p de probabilidade de sucesso. Seja k a diferença entre as probabilidades de sucesso e de fracasso dessa distribuição, de maneira que

$$k = p - (1 - p) = 2p - 1$$

A variável Y é dita desbalanceada se e somente se

$$k := |2p - 1| \geq 0.2$$

Dessa definição é fácil notar que se houver perfeito equilíbrio entre as classes, então as probabilidades são iguais e $k = 0$.

2.2 Tratamento de dados desbalanceados

A busca para soluções para o problema do desbalanceamento de classes situa-se em uma área ativa de pesquisa e várias soluções têm sido propostas (BATISTA et al., 2004; HAIXIANG et al., 2017; HAN et al., 2005; HE, MA, 2013; HUAYANAY et al., 2019; MAALOUF et al., 2018; VAN DER PAAL, 2014).

É possível encontrar soluções que se utilizam tanto do paradigma frequentista quanto bayesiano. Neste trabalho focamos nesse segundo paradigma. Segundo PRATI et al. (2003), em geral as técnicas propostas na literatura frequentista podem ser divididas em dois grandes grupos:

- Sobre amostragem (OS - Over-Sampling): utilização de técnicas para reamostrar os dados visando aumentar a proporção da classe sub-representada;
- Sub Amostragem (US - Under sampling) - consiste em reamostrar os dados visando reduzir a proporção da classe sub-representada.

Tanto no caso da Sub-amostragem quanto em sobre amostragem o objetivo é obter um conjunto de dados mais equilibrado em relação às classes da variável de interesse, no primeiro caso recriando novas ocorrências da variável em desequilíbrio e no segundo, eliminando o excesso de amostras da classe predominante. Possíveis problemas com esses procedimentos são, por um lado, a eliminação de amostras potencialmente úteis, e por outro o aumento da probabilidade de overfitting no modelo treinado (HE, GARCIA, 2009; PRATI et al., 2003). Visando minimizar a possibilidade de ocorrência destes problemas, BATISTA et al., 2003 e BATISTA et al., 2004 propuseram um terceiro grupo de técnicas composto Sobre Amostragem seguida de sub amostragem (OSUS). Todos os procedimentos anteriores (OS, US e OSUS), possuem uma forma de aplicação semelhante: utiliza-se o procedimento para reamostrar o conjunto de dados desbalanceado, ou seja, criar amostras ou reduzir as existentes e, em seguida, utiliza-se o novo conjunto reamostrado para treinar um algoritmo de Machine Learning (exemplos: SVM-Suport Vector Machines, KNN-K Nearest Neighbour etc.). Por fim, avalia-se o desempenho dos procedimentos utilizados por meio de métricas de performance, em geral computadas a partir da matriz de confusão. Recentemente têm sido propostas algumas técnicas que possuem os dois procedimentos (reamostragem e treino) em um único algoritmo. Tais técnicas são conhecidas

como Ensemble Models (GALAR et al., 2012). Em Python é possível encontrar técnicas desenvolvidas estes grupos, com diversas variações e neste trabalho utilizaremos todos quatro conjuntos de técnicas aqui citados.

2.3 Regressão Logística

Como baseline, utilizaremos um modelo que pressupõe uma variável resposta (ou target) dicotômica. Assim o target assume o valor 1 (um) se o fenômeno ocorre ou 0 (zero) se não ocorre. A Regressão Logística é um dos métodos estatísticos mais indicados para utilização com variáveis resposta dicotômicas (HOSMER; LEMESHOW, 2000). No caso do conjunto de dados que estamos utilizando, o objetivo é determinar a probabilidade *p* de que o prospect da ação de marketing venha a adquirir a oferta do produto de investimento, sendo que *p* expressa a probabilidade condicional de que *Y*, a variável resposta seja igual a 1, dadas as variáveis explicativas do modelo, ou seja,

$$p = E(Evento = 1 / x) = e^{Z} / (1 + e^{Z})$$

em que *Z* é o polinômio que representa a variável resposta do modelo geral em função das variáveis explicativas (indicadas pelo vetor *x*) e seus respectivos parâmetros. Vários autores têm observado que quando os dados são desbalanceados, a regressão logística não apresenta o melhor desempenho (HUAYANAY et al., 2019). Para contornar este problema diversos algoritmos foram propostos entre os quais podemos citar o uso de links assimétricos (HUAYANAY et al., 2019), reamostragem e modelos ensemble, citados anteriormente.

2.4 Python

Python é uma linguagem de programação em alto nível, criada em 1991 (ROSSUM, 1995). O intento original do autor era elaborar uma linguagem de fácil entendimento por programadores, com uma curva de aprendizagem mais atenuada quando comparada às linguagens tradicionalmente utilizadas à época, como C e Java (McKINNEY, 2018). Em pouco tempo, o uso de Python se disseminou entre a comunidade de programadores encontrando a cada dia novas aplicações. Uma característica importante do Python é sua natureza open-source, ou seja, é um software em que os usuários podem acessar o código fonte, distribuí-lo e modificá-lo de acordo com suas necessidades (HIPPEL, 2001). Graças a isso, a partir da década de 2010, o uso de Python para análise de dados e Machine Learning aumentou consideravelmente, em função

da criação de novos pacotes pela comunidade de usuários, em particular os pacotes Numpy, pandas e Scykit-Learn (GÉRON, 2019; McKINNEY, 2018).

Em razão de sua natureza open-source, os programas em Python podem ser acionados pelo usuário final através da Application Program Interface (API) de cada programa. Em 2017 foi criado um conjunto de programas chamado Imbalanced-Learn API, que inclui uma série de algoritmos para lidar com o problema de classes desbalanceadas, sejam elas binárias ou multi-classes (LEMAÎTRE et al. 2017). No próximo capítulo apresentamos as brevemente as referências dos métodos para tratamento de desbalanceamento de classes binárias.

2.5 Algoritmos em Python para tratamento de desbalanceamento

Neste estudo nos propomos avaliar o desempenho de diferentes algoritmos para dados de classificação binária quando existe desbalanceamento, assim, fizemos uma revisão dos diferentes algoritmos disponíveis que fazem ênfases em técnicas de amostragem, sejam estes de sobre amostragem ou de Sub-amostragem.

A seguir, na tabela 1 apresentamos os modelos disponíveis em Python para tratamento de classes binárias desbalanceadas binárias considerando técnicas de sobre amostragem

Tabela 1 – Técnicas de sobre Amostragem (Over-sampling)

Técnica	Descrição	Referências
SMOTE - Synthetic Minority Over-sampling Technique	Cria dados da classe minoritária por interpolação, através de um algoritmo kNN.	CHAWLA et al., 2002; HAN et al. 2005
SVM SMOTE - Support Vectors SMOTE	Trata-se de uma variante do SMOTE, que utiliza SVM para detectar a amostra a ser usada como modelo na geração de novas amostras.	NGUYEN et al., 2009
ADASYN - Adaptive synthetic sampling approach for imbalanced learning	Funciona de forma similar ao SMOTE, porém utiliza uma função de densidade para ponderar a geração de amostras.	HE et al., 2008

Fonte: Autor

Também, na tabela 2 apresentamos os modelos disponíveis em Python para tratamento de classes binárias desbalanceadas binárias considerando técnicas de sub amostragem

Tabela 2 – Técnicas de sub amostragem (Under-Sampling)

Técnica	Descrição	Referências
Extraction of majority-minority Tomek links	Detecta e remove os chamados Tomek Links, caracterizados pelo fato existirem duas amostras que são os vizinhos mais próximos entre si.	TOMEK, 1976
NearMiss	Seleciona aleatoriamente um subconjunto dos dados da classe majoritária conforme sua distância aos dados da classe minoritária seja a menor possível.	MANI, ZANG; 2003
Condensed Nearest Neighbour	Decide se uma amostra majoritária será excluída ou não com base na regra 1NN (um vizinho mais próximo). Um ponto de atenção é que esse método exige uma série de cuidados por ser sensível a amostras com ruído.	HART, 1968
One-Sided Selection	Similar ao método Condensed Nearest Neighbour, porém, utilizando, Tomek Links para remover amostras com ruído.	KUBAT, MATWIN, 1997
Neighboorhood Cleaning Rule	Remove amostras majoritárias que são classificadas erroneamente utilizado 3NN (três vizinhos mais próximos).	LAURIKKALA, 2001
Edited Nearest Neighbours	Utiliza um algoritmo kNN e "edita" o conjunto de dados removendo amostras que não concordam "o suficiente" com sua vizinhança	WILSON, 1972
Instance Hardness Threshold	Remove as amostras da classe majoritária como baixa probabilidade de ocorrência.	SMITH et al. 2014
Repeated Edited Nearest Neighbours	Variante do método Edited Nearest Neighbours, repetindo o algoritmo muitas vezes, o que geralmente acaba eliminando mais amostras da classe majoritária.	TOMEK, 1976

Al.KNN	Aplica um algoritmo KNN e remove amostras que não se parecem o suficiente com a vizinhança, considerando que os vizinhos mais próximos devem pertencer à mesma classe que a amostra inspecionada para mantê-lo no conjunto de dados	TOMEK, 1976

Fonte: Autor

Adicionalmente, identificamos técnicas que iniciam com o sobre amostragem e logo aplicam sub amostragem como as apresentadas na Tabela 3.

Tabela 3 – Técnicas de sobre amostragem seguida por sub amostragem

Técnica	Descrição	Referências
SMOTE + Tomek links	Remove possíveis ruídos das amostras geradas pelo método SMOTE, utilizando Tomek Links.	BATISTA et al., 2003
SMOTE + ENN	Remove possíveis ruídos das amostras geradas pelo método SMOTE, utilizando a técnica ENN (variante do kNN). Também chamado de SMOTEENN, remove mais amostras do que o método SMOTE + Tomek Links.	BATISTA et al., 2004

Fonte: Autor

Por fim, na Tabela 4, relacionamos as técnicas de Ensemble Learning que visam reduzir os problemas potenciais dos algoritmos anteriores e que são estão desenvolvidas em Python.

Tabela 4 - Ensemble Classifiers com amostragem interna

Técnica	Descrição	Referências
Easy Ensemble classifier	Faz a reamostragem de cada subconjunto dos dados ante de treinar cada estimador do Ensemble Classifier.	LIU et al., 2009
Balanced Random Forest	Cada árvore da floresta é avaliada através de amostragem bootstrap.	CHAO et al., 2004

RUSBoost	Aleatoriamente reamostra a classe majoritária, reduzindo-a antes de aplicar um modelo de Boosting.	SEIFFERT et al., 2010

Fonte: Autor

3 METODOLOGIA

3.1 Introdução

Neste capítulo descreveremos em detalhes a metodologia utilizada para análise, ressaltando os critérios de desempenho, os diferentes algoritmos de classificação utilizados e a base de dados utilizada para avaliar o desempenho dos diferentes algoritmos.

3.2 Abordagem do problema

Neste trabalho utilizaremos um procedimento comparativo entre os diversos métodos disponíveis em Python para desbalanceamento, de modo a identificar aqueles com melhor performance. Para tanto, em cada modelo aplicaremos uma técnica de desbalanceamento e, em seguida, utilizaremos um algoritmo de classificação binária. Esse tipo de algoritmo é adequado para casos em que a variável resposta possui natureza dicotômica, e um dos algoritmos mais utilizados é a Regressão Logística, que descreveremos a seguir. Além disso, para efeitos de reforço na comparação, adotaremos um modelo de base no qual não será aplicado nenhuma técnica de desbalanceamento. Assim, temos na Tabela 5 os modelos baseados nas famílias de métodos relacionados anteriormente.

Tabela 5 – Modelos a serem testados

Modelo	Descrição
Modelo 1	Somente Regressão Logística (Baseline)
Modelo 2	SMOTE + Regressão Logística
Modelo 3	SVM SMOTE + Regressão Logística
Modelo 4	ADASYN + Regressão Logística
Modelo 5	Tomek links + Regressão Logística
Modelo 6	NearMiss + Regressão Logística
Modelo 7	Condensed Nearest Neighbour + Regressão Logística

Modelo 8	One-Sided Selection + Regressão Logística
Modelo 9	Neighborhood Cleaning Rule + Regressão Logística
Modelo 10	Edited Nearest Neighbours + Regressão Logística
Modelo 11	Instance Hardness Threshold + Regressão Logística
Modelo 12	Repeated Edited Nearest Neighbours + Regressão Logística
Modelo 13	Al.KNN + Regressão Logística
Modelo 14	SMOTE + Tomek links + Regressão Logística
Modelo 15	SMOTE + ENN + Regressão Logística
Modelo 16	Easy Ensemble classifier
Modelo 17	Balanced Random Forest
Modelo 18	RUSBoost

3.3 Dados

Para testar empiricamente cada modelo descrito no item 2.2 utilizaremos uma base de dados chamada Bank Marketing Data Set (MORO et al., 2014), disponível no UCI-Machine Learning Repository. Os dados nesta base referem-se aos resultados de campanhas de marketing direto realizadas por uma instituição bancária portuguesa. As ações de marketing foram feitas através de contatos telefônicos com clientes potenciais da instituição. O resultado esperado da campanha era que o cliente realizasse aplicações financeiras na instituição (depósito a prazo). Com isso, o objetivo de cada método aqui aplicado será prever se o potencial cliente irá realizar ou não a aplicação financeira.

A base de dados original é composta de 41.188 registros (amostras) e de 20 variáveis, descritas a seguir com seu nome original em inglês no repositório de dados, o tipo de dado e os valores que assume:

- Variáveis de entrada:
  - Dados do cliente potencial:
    - 1 – AGE (idade): numérica
    - 2 – JOB (trabalho): tipo de trabalho (categórico: 'administrador', 'operário', 'empresário', 'empregada doméstica', 'administração', 'aposentado', 'autônomo', 'serviços', 'estudante', 'técnico', 'desempregado', 'desconhecido')

- 3 – MARITAL (estado civil) (categórico: 'divorciado', 'casado', 'solteiro', 'desconhecido'; observação: 'divorciado' significa divorciado ou viúvo)
- 4 – EDUCATION (nível educacional): categórica ('basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'analfabeto', 'professional.course', 'university.degree', 'unknown')
- 5 – DEFAULT (tem crédito inadimplente?): categórico ('não', 'sim', 'desconhecido')
- 6 – HOUSING (tem crédito habitacional?): categórico: 'não', 'sim', 'desconhecido')
- 7 – LOAN (tem empréstimo pessoal?): categórico ('não', 'sim', 'desconhecido')

o Dados relacionados com o último contato da campanha atual:

- 8 – CONTACT (tipo de comunicação do contato): categórico: ('celular', 'telefone');
- 9 – MONTH (mês do último mês de contato do ano): categórico ('jan', 'fev', 'mar', ..., 'nov', 'dez');
- 10 – DAY OF WEEK (último dia de contato da semana): categórico ('seg', 'ter', 'qua', 'qui', 'sex');
- 11 – DURATION (duração do último contato, em segundos): numérico. Observação importante: este atributo afeta fortemente o target (por exemplo, se duração = 0, então y = 'não'). No entanto, a duração não é conhecida antes que uma chamada seja realizada. Além disso, após o fim da chamada, y é obviamente conhecido. Assim, essa entrada deve ser incluída apenas para fins de benchmark e deve ser descartada se a intenção for ter um modelo preditivo realista. No entanto, optamos por manter essa variável para fins de teste mais acurado dos modelos.

o Outros atributos:

- 12 – CAMPAIGN: quantidade de contatos realizados durante esta campanha e para este cliente (numérico, inclui último contato)
- 13 – PDAYS: número de dias que passaram após o último contato do cliente de uma campanha anterior (numérico; 999 significa que o cliente não foi contatado anteriormente)

        - 14 – PREVIOUS: número de contatos realizados antes desta campanha e para este cliente (numérico)
        - 15 – POUTCOME: resultado da campanha de marketing anterior (categórica: 'fracasso', 'inexistente', 'sucesso')
    - Atributos de contexto social e econômico
        - 16 - EMP.VAR.RATE: taxa de variação do emprego - indicador trimestral (numérico)
        - 17 - CONS.PRICE.IDX: índice de preços ao consumidor - indicador mensal (numérico)
        - 18 - CONS.CONF.IDX: índice de confiança do consumidor - indicador mensal (numérico)
        - 19 - EURIBOR3M: taxa euribor 3 meses - indicador diário (numérico)
        - 20 - NR. EMPREGADOS: número de funcionários - indicador trimestral (numérico)
- Variável de saída (target):
    - 21 - Y - o cliente realizou a aplicação financeira? (binário: 'sim', 'não')

3.4 Análise exploratória dos Dados

A Variável target está distribuída da seguinte forma: 4.641 (11%) para ‘sim’, indicando que o cliente potencial realizações a aplicação financeira e 36.549 (89%) para ‘não’. Estes números indicam uma base bastante desbalanceada na variável resposta e, portanto, adequada para a análise que pretendemos fazer. Nos dedicamos a partir de agora a realizar a análise descritiva das variáveis. Num segundo momento, faremos as transformações necessárias para acomodação nos modelos, bem como a exclusão de variáveis com pouco ou nenhum impacto na variável de resposta. Em seguida faremos o processamento dos modelos e sua posterior comparação de desempenho. Para tanto, particionaremos a base de dados original em dois blocos, um de treino e outro de teste. Em todos as fases deste trabalho utilizamos a linguagem Python para gerar os gráficos e tabelas necessárias.

A análise exploratória descrita nesta seção foi desenvolvida com um Notebook em Python denominado Notebook 1 (Anexo 1).

3.4.1 – Análise exploratória das variáveis numéricas

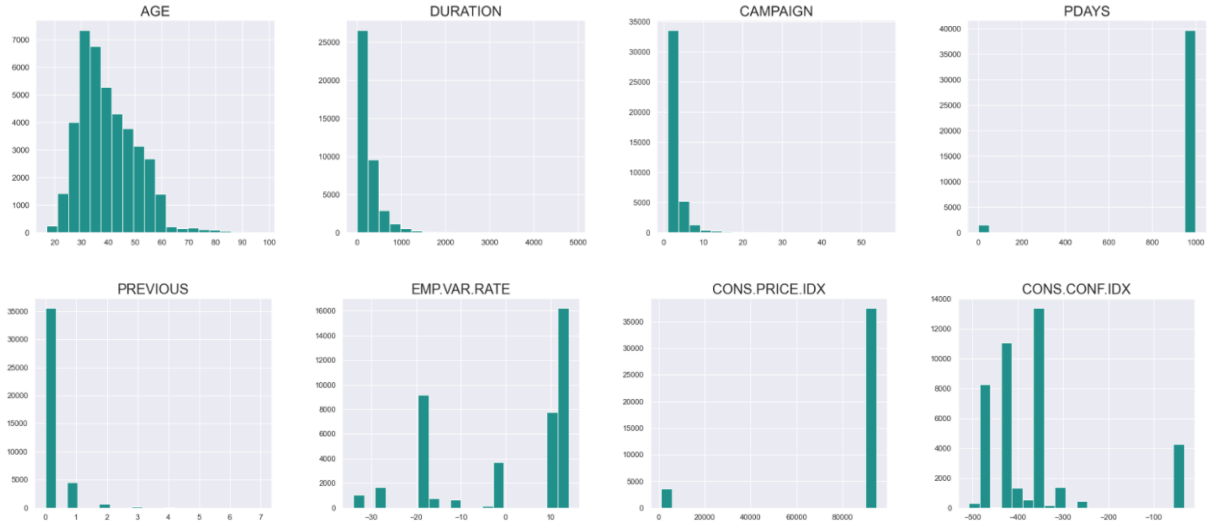
Na Tabela 6 apresentamos a análise descritiva das variáveis numéricas do conjunto de dados utilizado para aplicação dos modelos.

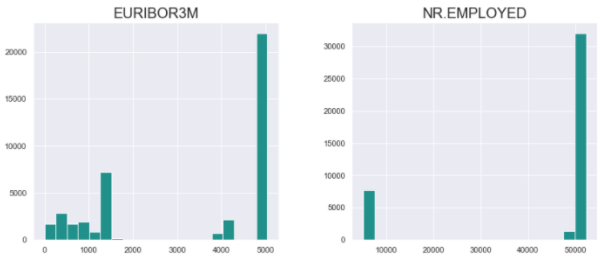
Tabela 6 – Estatísticas descritivas das variáveis numéricas – dados originais

Variável	Amostras	Média	Desvio Padrão	Mínimo	25%	50%	75%	Máximo
AGE	41.188	40,024	10,421	17	32	38	47	98
DURATION	41.188	258,285	259,279	0	102	180	319	4.918
CAMPAIGN	41.188	2,568	2,770	1	1	2	3	56
PDAYS	41.188	962,475	186,911	0	999	999	999	999
PREVIOUS	41.188	0,173	0,495	0	0	0	0	7
EMP.VAR.RATE	41.188	0,932	15,584	-34	-18	11	14	14
CONS.PRICE.ID	41.188	85.475,220	26.234,184	932	92.893	93.749	93.994	94.767
CONS.CONF.ID	41.188	-365,666	119,100	-508	-427	-403	-361	-33
EURIBOR3M	41.188	3.284,959	1.935,702	1	1.281	4.856	4.961	5.045
NR.EMPLOYED	41.188	42.864,892	18.170,198	5.191	50.175	50.991	52.281	52.281
TARGET	41.188	0,113	0,316	0	0	0	0	1

As variáveis numéricas apresentam alta discrepância no seu range de valores. Por exemplo, o valor médio de EMP.VAR.RATE é 0.93, enquanto a média para CONS.PRICE.IDX é 85475.2. Essa primeira visão dos dados aponta para a necessidade de realizar algum tipo de transformação para deixá-los em escala próxima. Faremos isso adiante.

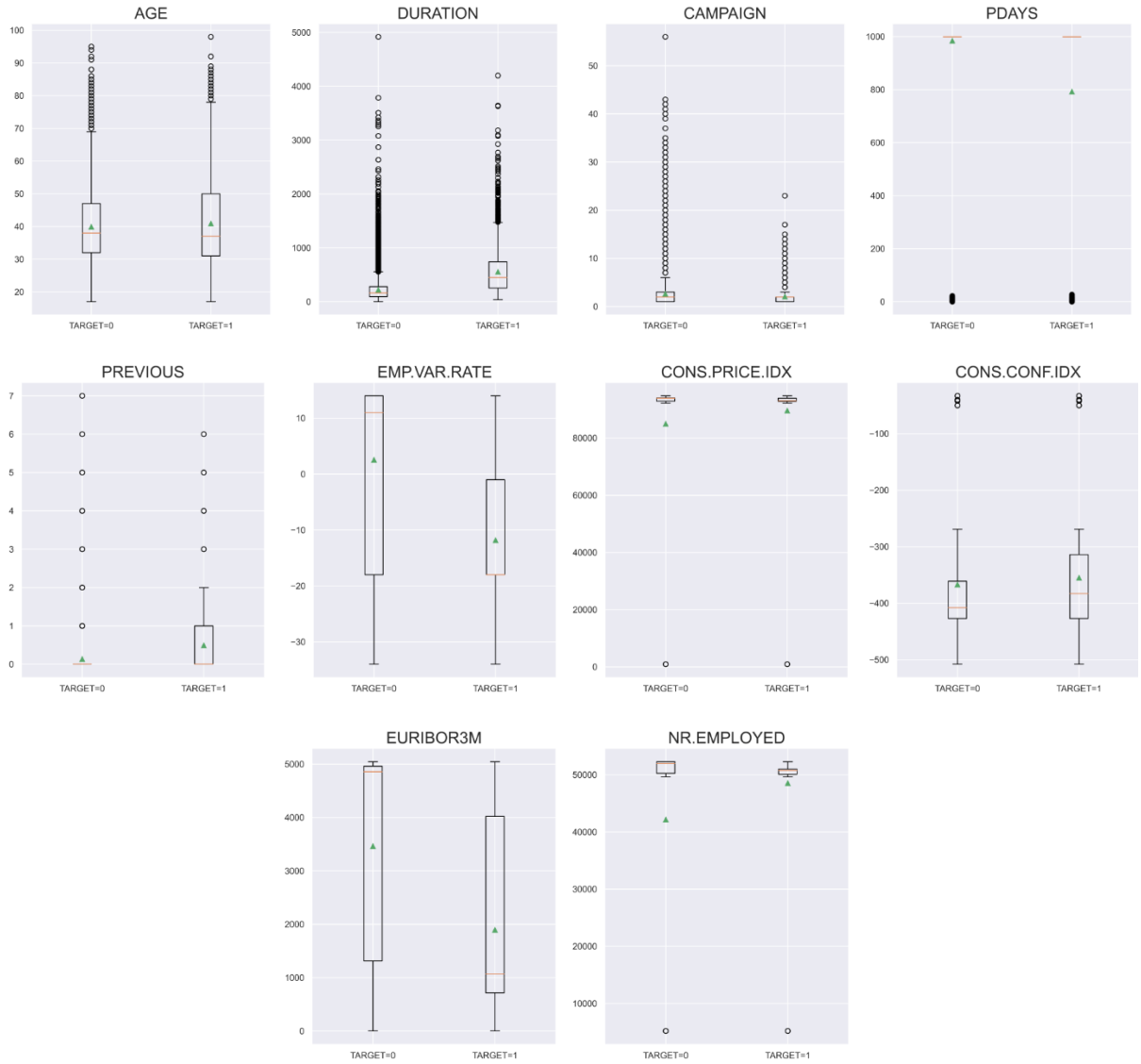
Gráfico 1 – Histograma das variáveis numéricas – dados originais





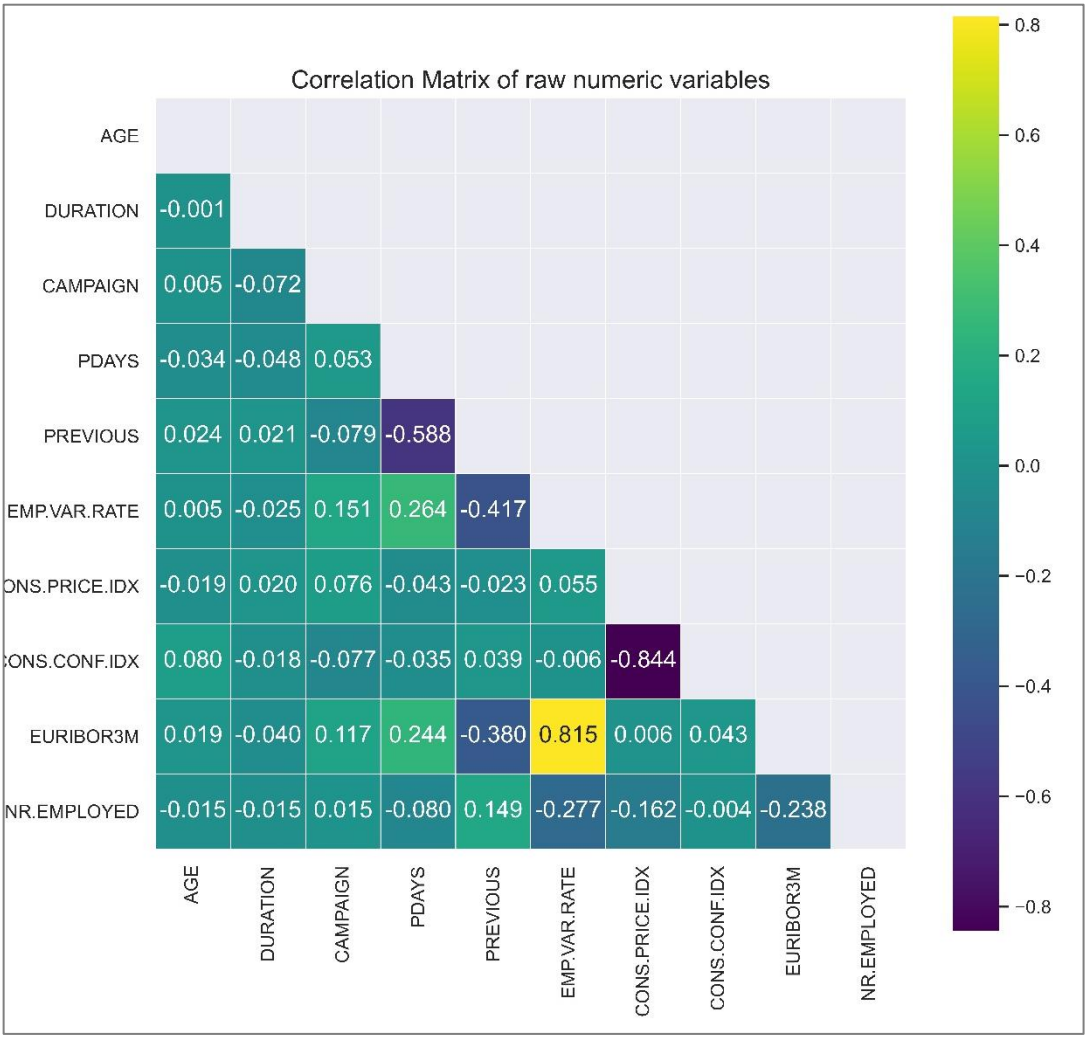
O histograma representa uma visão complementar à tabela de estatísticas descritivas. A partir da análise do histograma das variáveis numéricas, vemos que a maior parte delas tem distribuição assimétrica, confirmando a necessidade de algum tipo de transformação, conforme expresso no Gráfico 2

Gráfico 2 – Relação entre variáveis numéricas e variável resposta – Dados originais



A visualização utilizando gráficos de Box-Plot reforça o diagnóstico de assimetria dos dados e a possível presença de outliers. Mas, a informação mais importante é que os Box-Plot nos permitem avaliar a variação da variável resposta controlando por cada variável preditora. Nesse caso, observamos que as variáveis DURATION, PREVIOUS e EMP.VAR.RATE são as que possuem maior poder discriminante em relação ao TARGET, já que os gráficos de cada uma delas para as duas classes desse são as que apresentam perfil mais dessemelhante. Já CONS.CONF.IDX e EURIBONR3M são mediamente discriminantes, enquanto as demais variáveis AGE, CAMPAIGN, PDAYS, CONS.PRICE.IDX e NR. EMPLOYED têm baixo poder discriminatório. Conclusão: vamos retirar da base as variáveis numéricas AGE, CAMPAIGN, PDAYS, CONS.PRICE.IDX e NR. EMPLOYED por possuírem baixo poder discriminatório.

Gráfico 3 – Análise de Correlação entre as variáveis numéricas – Dados originais



Como complemente à nossa análise, avaliamos o grau de correlação entre as variáveis numéricas visando identificar pares altamente correlacionados e que poderiam prejudicar os algoritmos que vamos utilizar. A tabela acima exibe o coeficiente de correlação linear de Pearson entre as variáveis, duas a duas. Observamos três tuplas altamente correlacionados:

- CONS.CONF.IDX e CONS.PRICE.IDX (rô = -0.84); e
- EURIBOR3M com EMP.VAR RATE (rô = 0.81).
- PREVIOUS com EURIBOR3M (rô = -0.38) e com EMP.VAR.RATE (rô = -0.42)

No primeiro par, não teremos problemas pois já eliminamos a variável CONS.CONF.IDX na seção anterior. No segundo par, utilizamos como regra de decisão o poder discriminatório da variável em relação ao TARGET. Nesse caso, optamos por eliminar EMP.VAR RATE por possuir menor variabilidade em relação ao TARGET, quando comparada graficamente com EURIBOR3M. Optamos também por eliminar PREVIOUS.

Conclusão geral sobre as variáveis numéricas:

- Variáveis a serem mantidas: DURATION, PREVIOUS, CONS.CONF.IDX e EURIBONR3M;
- Variáveis a serem desconsideradas para a modelagem: AGE, CAMPAIGN, PDAYS, CONS.PRICE.IDX, NR. EMPLOYED e EMP.VAR RAT.

3.4.2 – Análise exploratória das variáveis categóricas

Tabela 7 – Distribuição das variáveis categóricas

Categoric variable: job

	Values	Qtde	Pct
0	admin.	10422	0.253035
1	blue-collar	9254	0.224677
2	technician	6743	0.163713
3	services	3969	0.096363
4	management	2924	0.070992
5	retired	1720	0.041760
6	entrepreneur	1456	0.035350
7	self-employed	1421	0.034500
8	housemaid	1060	0.025736
9	unemployed	1014	0.024619
10	student	875	0.021244
11	unknown	330	0.008012

Categoric variable: marital

	Values	Qtde	Pct
0	married	24928	0.605225
1	single	11568	0.280859
2	divorced	4612	0.111974
3	unknown	80	0.001942

Categoric variable: education

	Values	Qtde	Pct
0	university.degree	12168	0.295426
1	high.school	9515	0.231014
2	basic.9y	6045	0.146766
3	professional.course	5243	0.127294
4	basic.4y	4176	0.101389
5	basic.6y	2292	0.055647
6	unknown	1731	0.042027
7	illiterate	18	0.000437

Categoric variable: default

	Values	Qtde	Pct
0	no	32588	0.791201
1	unknown	8597	0.208726
2	yes	3	0.000073

Categoric variable: housing

	Values	Qtde	Pct
0	yes	21576	0.523842
1	no	18622	0.452122
2	unknown	990	0.024036

Categoric variable: loan

	Values	Qtde	Pct
0	no	33950	0.824269
1	yes	6248	0.151695
2	unknown	990	0.024036

Categoric variable: contact

	Values	Qtde	Pct
0	cellular	26144	0.634748
1	telephone	15044	0.365252

Categoric variable: month

	Values	Qtde	Pct
0	may	13769	0.334296
1	jul	7174	0.174177
2	aug	6178	0.149995
3	jun	5318	0.129115
4	nov	4101	0.099568
5	apr	2632	0.063902
6	oct	718	0.017432
7	sep	570	0.013839
8	mar	546	0.013256
9	dec	182	0.004419

Categoric variable: day_of_week

	Values	Qtde	Pct
0	thu	8623	0.209357
1	mon	8514	0.206711
2	wed	8134	0.197485
3	tue	8090	0.196416
4	fri	7827	0.190031

Categoric variable: poutcome

	Values	Qtde	Pct
0	nonexistent	35563	0.863431
1	failure	4252	0.103234
2	success	1373	0.033335

As variáveis JOB, EDUCATION, MONTH e DAY_OF_WEEK apresentam alto grau de variabilidade entre as classes que as compõem. Já as variáveis MARITAL, DEFAULT, HOUSING, LOAN, CONTACT e POUTCOME apresentam menos classes e maior concentração em pelo menos uma delas. Por exemplo, em POUTCOME, 86,3% das

observações estão na classe “nonexistent”, indicando que esse percentual de pessoas da amostra não participou de nenhuma campanha de marketing anterior. A existência de variáveis categóricas preditoras na base de dados com tamanha discrepância no percentual de classes exige uma análise mais detalhada de cada variável contra as classes da variável resposta, que é o que faremos a seguir.

3.4.2.1 – Relação entre variáveis categóricas e variável resposta

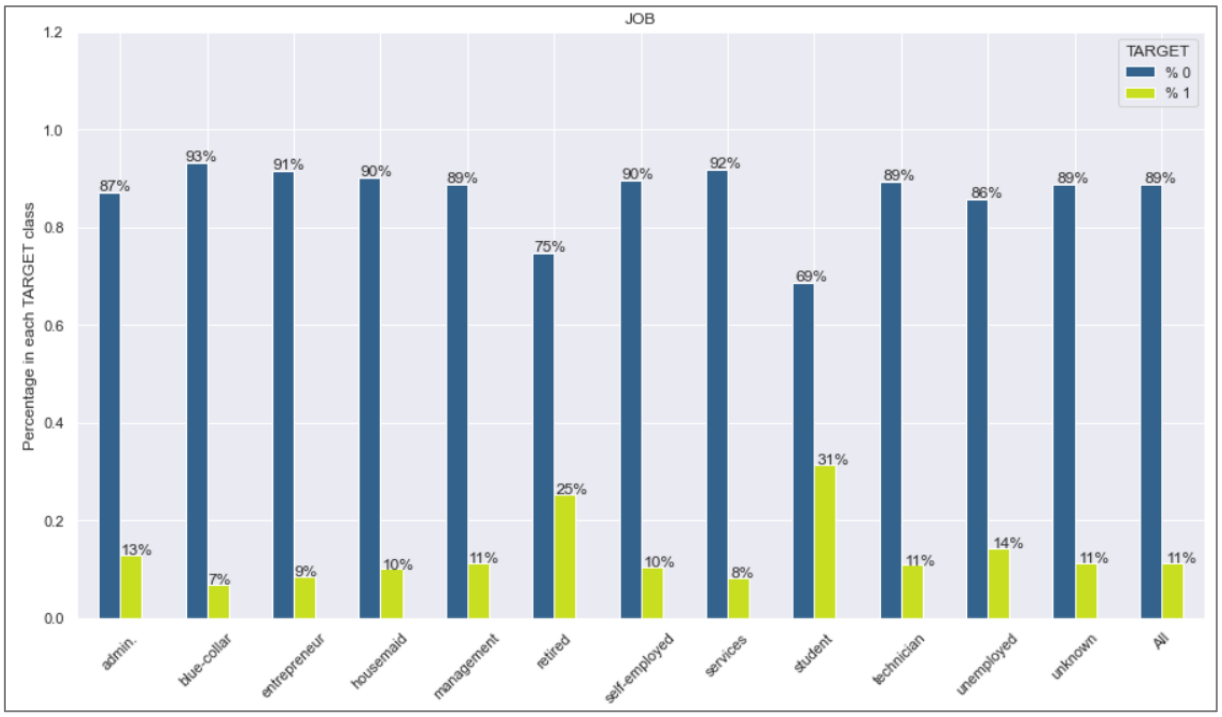
Para cada variável calculamos a estatística Qui-Quadrado para verificar associação entre a variável preditora e o Target. Sob a hipótese nula de não associação entre as variáveis o valor desta estatística nos fornece um parâmetro para decidirmos pela manutenção ou não da variável no modelo.

Tabela 8 – Estatística Qui-quadrado para variáveis categóricas e o Target

Variável	Estatística Qui-quadrado	p-value	Graus de liberdade	Resultado
JOB	0,513	1,000	11	Aceita H0
MARITAL	0,017	0,999	3	Aceita H0
EDUCATION	0,138	1,000	7	Aceita H0
DEFAULT	0,149	0,928	2	Aceita H0
HOUSING	0,000	1,000	2	Aceita H0
LOAN	0,000	1,000	2	Aceita H0
CONTACT	4,556	0,033	1	Rejeita HO
MONTH	1,699	0,995	9	Aceita H0
DAY_OF_WEEK	0,003	1,000	4	Aceita H0
POUTCOME	0,929	0,628	2	Aceita H0

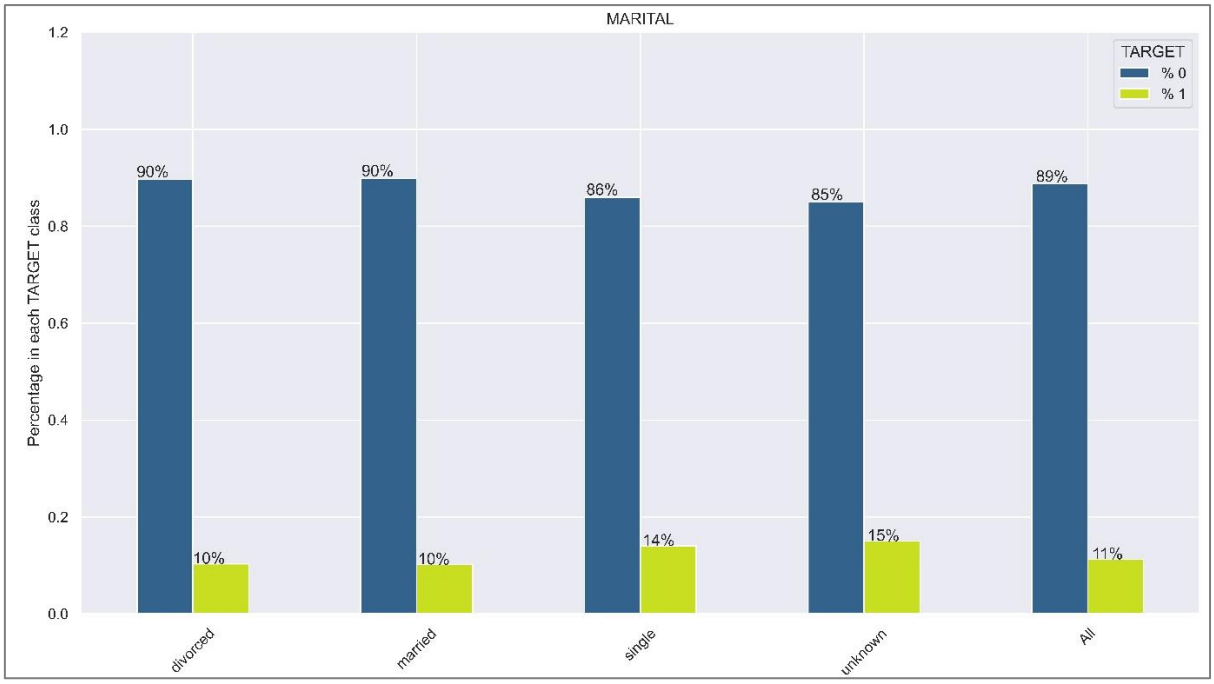
Complementarmente, nos gráficos a seguir apresentamos a distribuição das classes de cada variável preditora categórica para as classes da variável resposta (TARGET). A partir desses dados obtivemos algumas conclusões preliminares a respeito delas. Em geral, quando encontramos diferenças importantes na distribuição das classes dentro de cada variável, optamos por manter a variável no conjunto de dados, mesmo que o resultado do teste Qui-quadrado aponte para a não rejeição da hipótese nula.

Gráfico 4 – Distribuição da variável JOB em relação ao Target



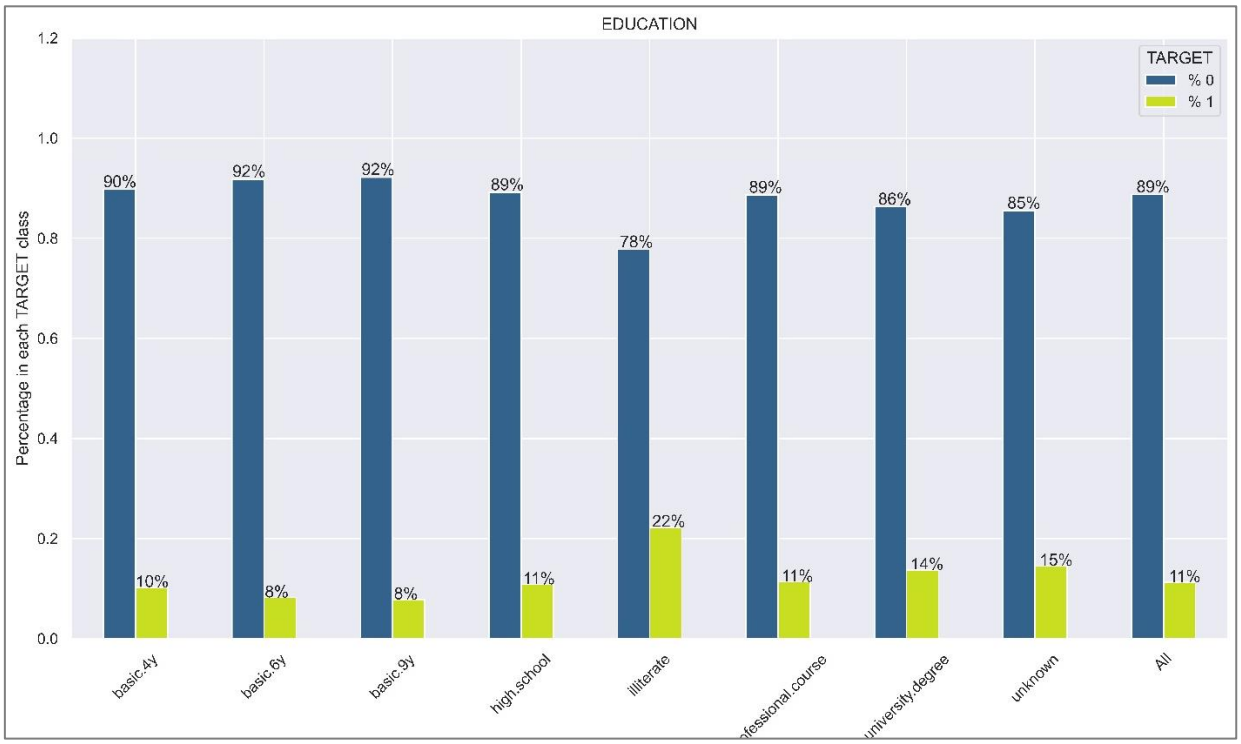
As classes “student” e “retired” destacam-se por apresentar maior proporção de ocorrências na classe minoritária (31% e 25% de “uns”, respectivamente). A interpretação desses dados à luz do seu campo de domínio indica que o fato de ser estudante ou aposentado indica capacidade discriminante para prever se o prospect da ação de marketing irá ou adquirir o produto. Já as demais variáveis apresentam, em seu conjunto, pouco poder discriminatório. Em conjunto, “student” e “retired” representam cerca de 6% das ocorrências da variável JOB, como pode ser visto nas tabelas da seção 4.1. Como conclusão, vamos transformar a variável categórica JOB em uma variável Dummy para apenas duas categorias: 1 se estudante ou aposentado e 0 para as demais classes.

Gráfico 5 – Distribuição da variável MARITAL em relação ao Target



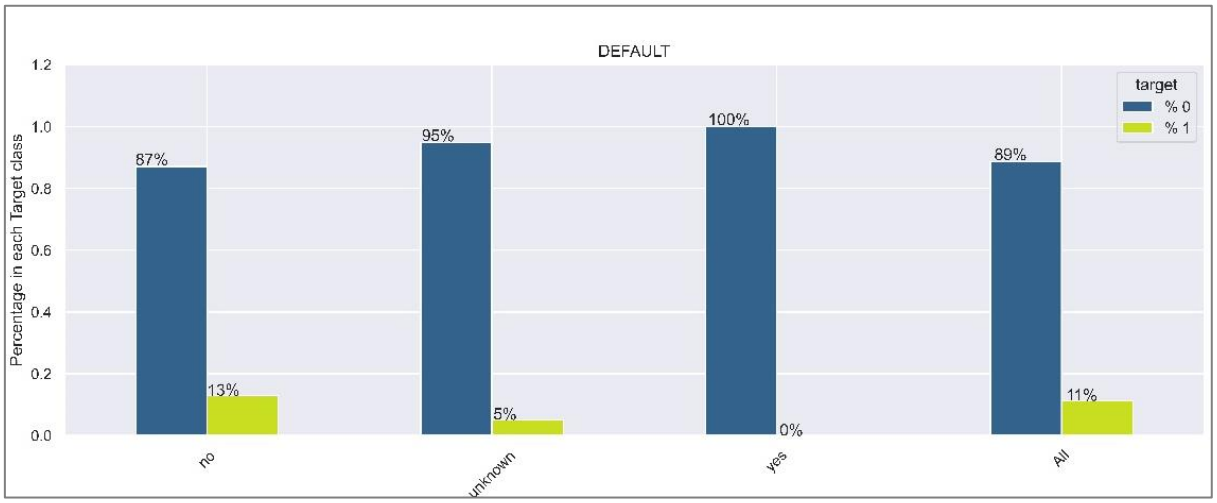
MARITAL – A distribuição das classes dessa variável não apresenta diferenças significativas entre si, indicando não haver poder discriminante para a variável TARGET. Conclusão: esta variável será excluída da base e não será considerada como preditora para a análise.

Gráfico 6 – Distribuição da variável EDUCATION em relação ao Target



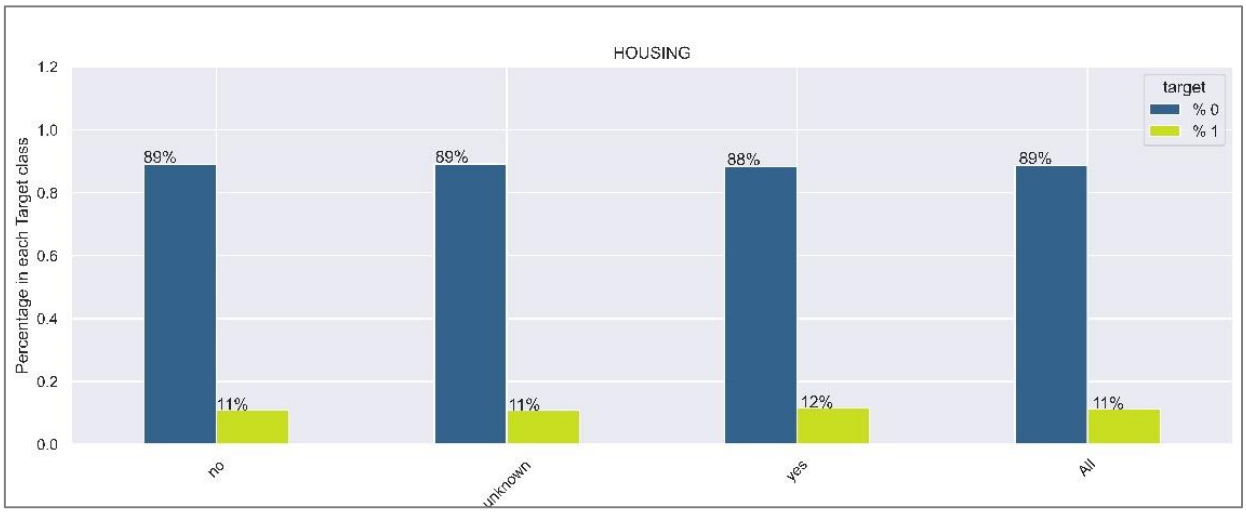
A classe ‘iliterate’ apresenta diferença significativa de distribuição para as classes do TARGET em relação às demais. No entanto, como pode ser observado nas tabelas da seção 4.1, a proporção dessa classe na variável EDUCATION é inferior à 0,01%, e, portanto, desprezível. Conclusão: esta variável será excluída da base e não será considerada como preditora para a análise.

Gráfico 7 – Distribuição da variável DEFAULT em relação ao Target



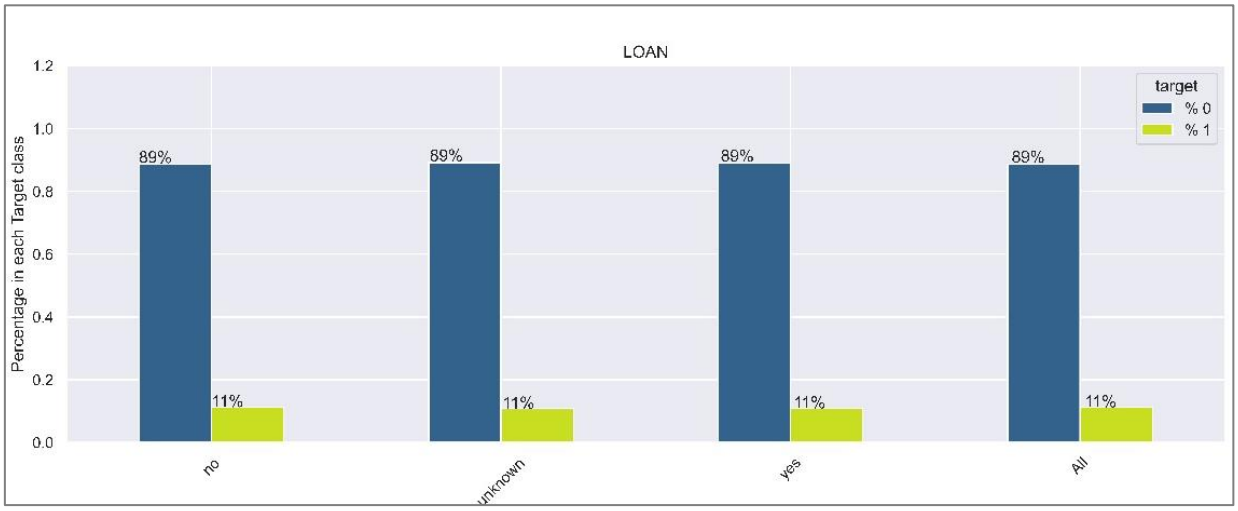
A classe “no” apresenta diferença significativa de distribuição para as classes do TARGET em relação às demais. Além disso, como pode ser observado nas tabelas da seção 4.1, a proporção dessa classe na variável DEFAULT é de 79,1%. Conclusão: vamos transformar a variável categórica DEFAULT em uma variável Dummy para apenas duas categorias: 1 se “no” e 0 para as classes “unknown” e “yes”.

Gráfico 8 – Distribuição da variável HOUSING em relação ao Target



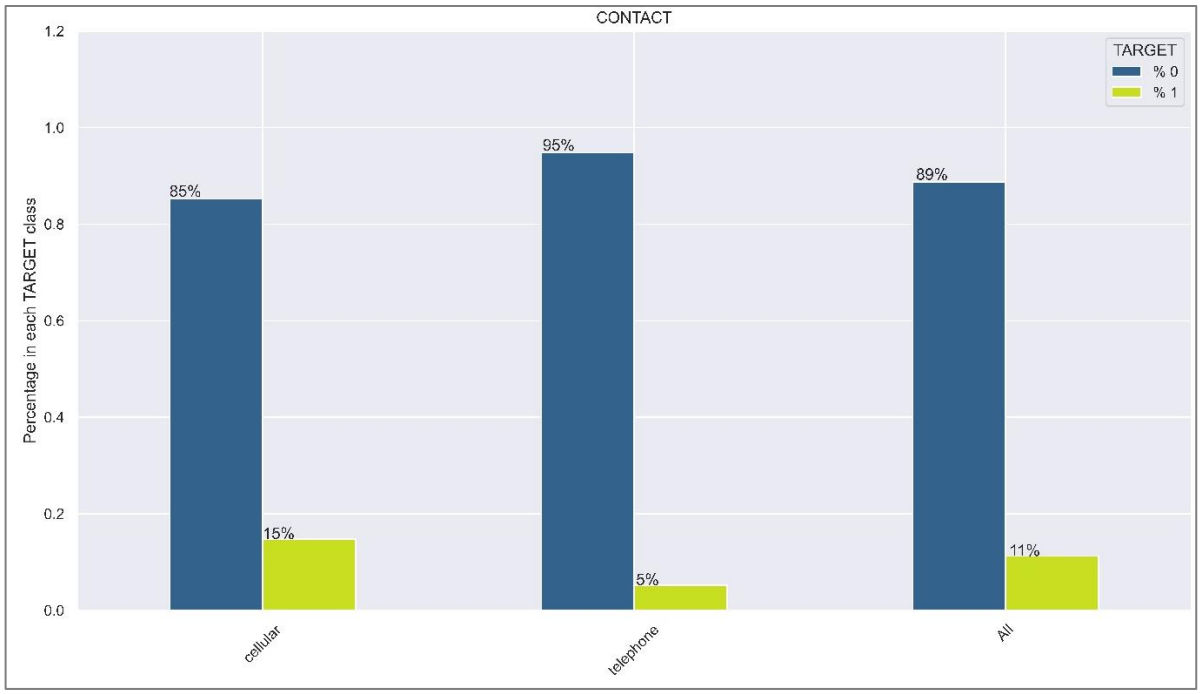
HOUSING – A distribuição das classes dessa variável não apresenta diferenças significativas entre si, indicando não haver poder discriminante para a variável TARGET. Conclusão: esta variável será excluída da base e não será considerada como preditora para a análise.

Gráfico 9 – Distribuição da variável LOAN em relação ao Target



A distribuição das classes dessa variável não apresenta diferenças significativas entre si, indicando não haver poder discriminante para a variável TARGET. Conclusão: esta variável será excluída da base e não será considerada como preditora para a análise.

Gráfico 10 – Distribuição da variável CONTACT em relação ao Target

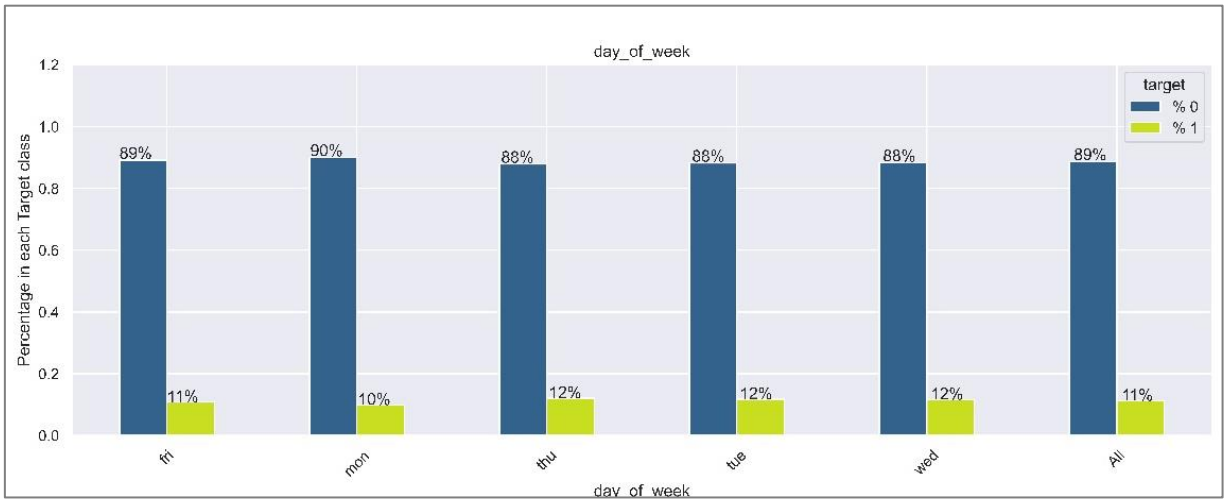


A classe “cellular” apresenta diferença significativa de distribuição para as classes do TARGET em relação às demais. Isso significa que quando o contato com o prospect para ofertar o produto é feito ligando-se para seu telefone celular, há maior chance de sucesso de que a venda do produto seja realizada. Além disso, como pode ser observado nas tabelas da seção 4.1, a proporção dessa classe na variável CONTACT é de 63%. Conclusão: vamos transformar a variável categórica CONTACT em uma variável Dummy para apenas duas categorias: 1 se “celullar” e 0 para a classe “telephone”.

Gráfico 11 – Distribuição da variável MONTH em relação ao Target

Vemos no gráfico acima que alguns meses do ano parecem favorecer o sucesso (percentual da classe 1 no TARGET) na venda do produto mais do que outros meses. Isso acontece quando o contato é feito nos meses de março (51% de sucesso), setembro (45%), outubro (44%) e dezembro (49%). Nos demais meses, a média de sucesso permanece em torno de 11%. Conclusão: vamos transformar a variável categórica MONTH em uma variável Dummy para apenas duas categorias: 1 para meses com percentual da classe 1 no TARGET superior à 40% e 0 para as classes com percentual inferior a esse valor.

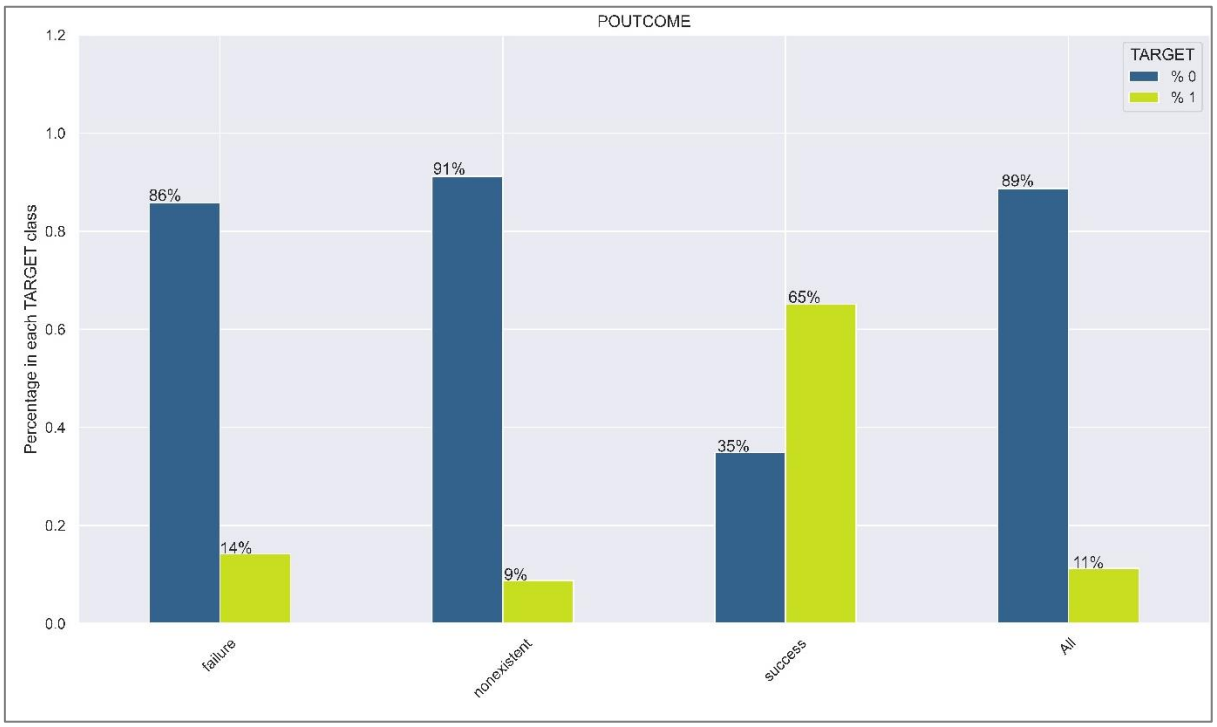
Gráfico 12 – Distribuição da variável DAY OF WEEK em relação ao Target



A distribuição das classes da variável DAY OF WEEK não apresenta diferenças significativas entre si, indicando não haver poder discriminante para a variável TARGET. Ou seja, ao contrário do mês em de contato, o dia da semana não faz diferença que a venda do produto seja

concluída ou não. Conclusão: esta variável será excluída da base e não será considerada como preditora para a análise.

Gráfico 13 – Distribuição da variável POUTCOME em relação ao Target



A classe “success” apresenta diferença significativa de distribuição para as classes do TARGET em relação às demais, indicando que 65% dos prospects que foram abordados na campanha de marketing anterior adquiriram o produto. Adicionalmente, como pode ser observado nas tabelas da seção 4.1, a proporção dessa classe na variável POUTCOME é de aproximadamente 3%, o que embora seja pequeno, não é desprezível. Além do mais, do ponto de vista da área de domínio, essa distribuição carrega uma informação importante para o gestor do produto, tal seja, a de que clientes que já adquiriram o produto anteriormente a partir de uma campanha de marketing, possuem mais probabilidade de o fazer novamente. Conclusão: vamos transformar a variável categórica POUTCOME em uma variável Dummy para apenas duas categorias: 1 para “success” e 0 para as demais classes.

3.5 – Conclusões após a análise das variáveis

- Variáveis categóricas:

  - Variáveis a serem categorizadas com zero e um: JOB, DEFAULT, CONTACT, MONTH e POUTCOME;
  - Variáveis a serem desconsideradas para a modelagem: MARITAL, EDUCATION, HOUSING, LOAN e DAY OF WEEK;
- Variáveis numéricas para transformação Raiz Quadrada e padronização (valor – média) / desvio-padrão: DURATION, PREVIOUS, EURIBOR3M, CONS.CONF.IDX.
- Variáveis finais para modelagem: DURATION, EURIBOR3M, CONS.CONF.IDX, JOB, DEFAULT, CONTACT, MONTH e POUTCOME.

3.5.1 - Resultados após todas as transformações

A Tabela 8 juntamente com os gráficos 15, 16 e 17 exibem os resultados após as fases de preparação dos dados. Os próximos passos são a especificação dos modelos e seu processamento.

Tabela 9 – Estatísticas descritivas das variáveis numéricas remanescentes

Variável	Amostras	Média	esvio Padr	Mínimo	25%	50%	75%	Máximo
DURATION	32.950	0	1	-2,202	-0,681	-0,187	0,484	8,362
EURIBOR3M	32.950	0	1	-2,618	-0,868	0,791	0,828	0,857
CONS.CONF.IDX	32.950	0	1	-2,616	-0,505	-0,212	0,228	2,496
JOB	32.950	0,063	0,243	0	0	0	0	1
DEFAULT	32.950	0,792	0,406	0	1	1	1	1
CONTACT	32.950	0,635	0,481	0	0	1	1	1
MONTH	32.950	0,049	0,216	0	0	0	0	1
POUTCOME	32.950	0,033	0,179	0	0	0	0	1

Gráfico 14 – Análise de Correlação entre as variáveis numéricas – Variáveis remanescentes

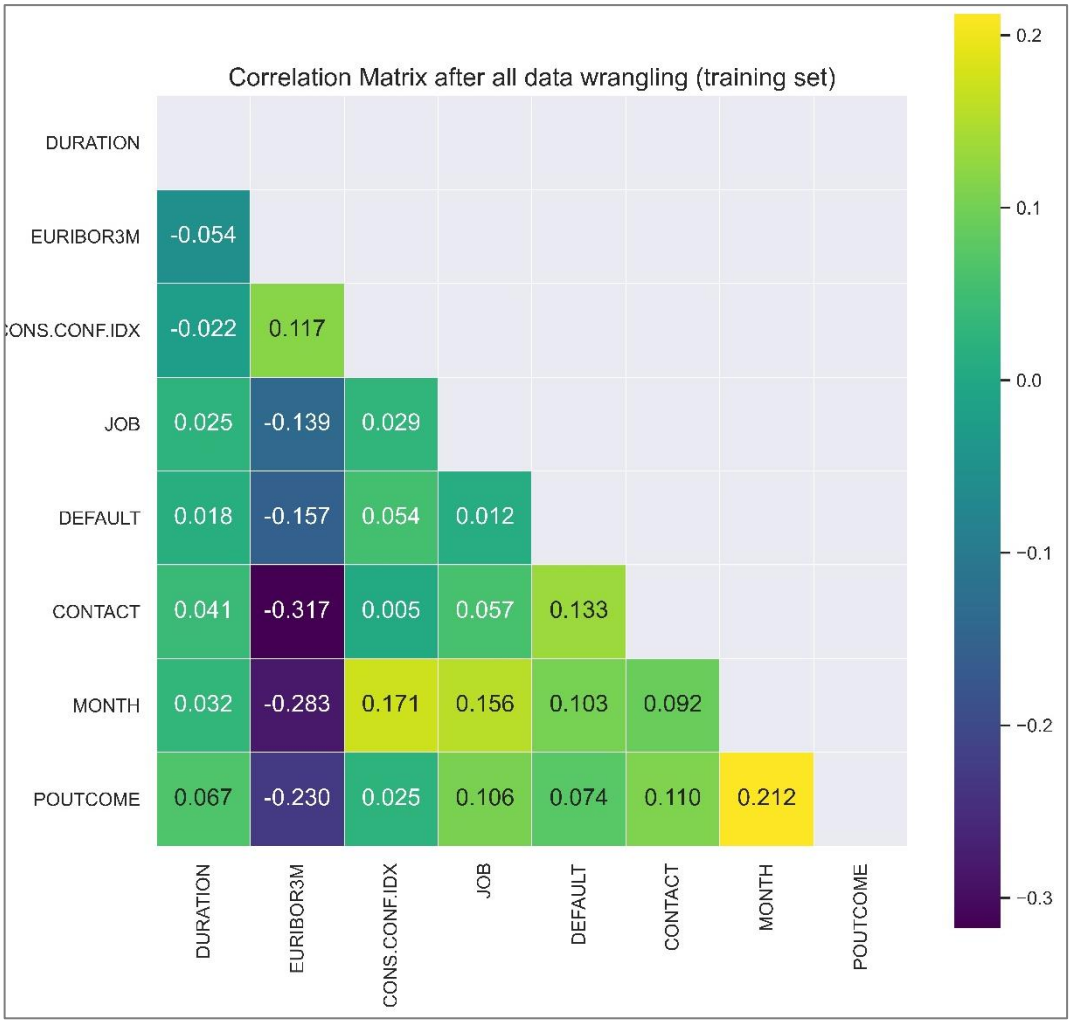


Gráfico 15 – Histograma das numéricas variáveis remanescentes

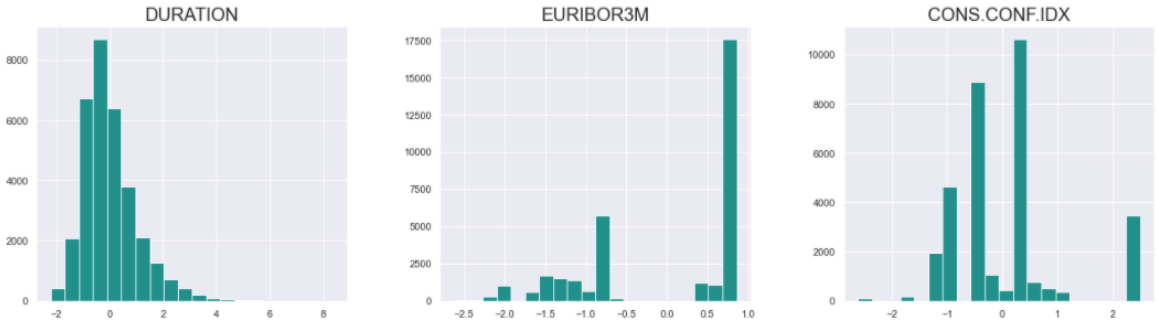
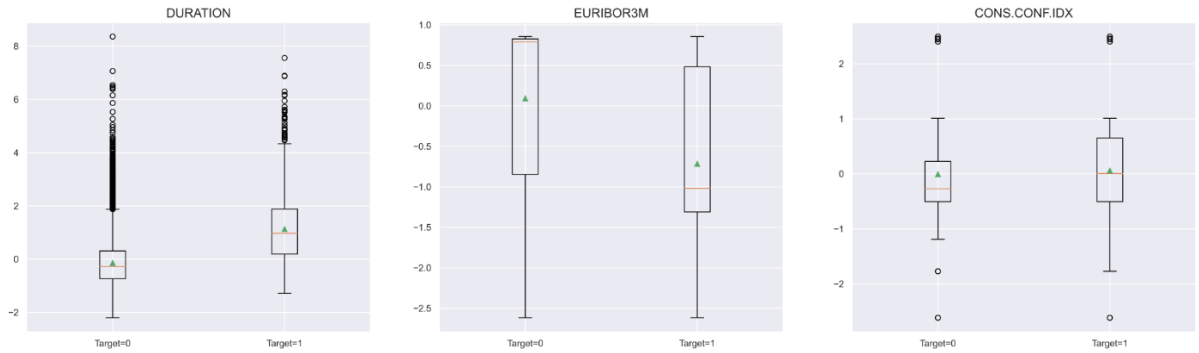


Gráfico 16 – Relação entre variáveis explicativas numéricas remanescentes e variável resposta



3.6 – Especificação dos modelos

3.6.1 – Descrição Geral

Em modelagem de dados utilizando Machine Learning é importante definir um fluxo de trabalho que delimite as fases desde o pré-processamento dos dados até a obtenção dos resultados. No presente estudo utilizamos o fluxo usualmente recomendado para modelagem em Machine Learning (FACELI et al. 2011; GÉRON, 2019). Seguindo esta metodologia, a base de dados original foi dividida em duas outros bases, uma para treino do modelo e outro para teste. Na tabela 9 vemos os como ficaram a distribuição das informações destas bases após este procedimento. Observamos que as bases remanescentes da divisão mantêm uma proporção de desbalanceamento das classes do Target muito próxima da original, o que é ideal para a modelagem.

Tabela 10 – Distribuição das classes do Target nos conjuntos de dados

Base de dados	Total de amostras	Target = 0		Target = 1	
Base Original	41.188	36.548	88,735%	4.640	11,265%
Conjunto de treino	32.950	29.245	88,756%	3.705	11,244%
Conjunto de teste	8.238	7.303	88,650%	935	11,350%

3.6.2 Algoritmos

Para os modelos baseline utilizamos a Regressão Logística disponível na biblioteca Scikit-Learn (PEDREGOSA et al., 2011), enquanto para os modelos de tratamento de dados desbalanceados utilizamos a biblioteca Imbalanced-Learn (LEMAÎTRE et al., 2019), ambos

disponíveis na linguagem Python. Os modelos foram processados utilizando com IDE (Integrated Development Environment) o ambiente Anaconda (ANACONDA, 2016). A Tabela 6 exibe o nome dos algoritmos e respectivas bibliotecas utilizadas. Os diferentes algoritmos foram implementados em Python no Notebook 1 (Anexo 1).

Tabela 11 – Algoritmos utilizados para processamento dos modelos

Biblioteca	Nome do Algoritmo	Descrição do modelo
Scikit-Learn	LogisticRegression	Regressão Logística
Statsmodels	Logit	Regressão Logística
Imbalanced-Learn	SMOTE	SMOTE
Imbalanced-Learn	SVMSMOTE	SVM SMOTE
Imbalanced-Learn	ADASYN	ADASYN
Imbalanced-Learn	TomekLinks	Tomek links
Imbalanced-Learn	NearMiss	NearMiss
Imbalanced-Learn	CondensedNearest Neighbour	Condensed Nearest Neighbour
Imbalanced-Learn	One-Sided Selection	One-Sided Selection
Imbalanced-Learn	NeighboorhoodCleaning Rule	Neighborhood Cleaning Rule
Imbalanced-Learn	EditedNearestNeighbours	Edited Nearest Neighbours
Imbalanced-Learn	InstanceHardnessThreshold	Instance Hardness Threshold
Imbalanced-Learn	RepeatedEditedNearest Neighbours	Repeated Edited Nearest Neighbours
Imbalanced-Learn	AllKNN	AllKNN
Imbalanced-Learn	SMOTETomek	SMOTE + Tomek links
Imbalanced-Learn	SMOTEENN	SMOTE + ENN
Imbalanced-Learn	EasyEnsembleClassifier	Easy Ensemble classifier
Imbalanced-Learn	BalancedRandomForest Classifier	Balanced Random Forest
Imbalanced-Learn	RUSBoostClassifier	RUSBoost

3.6.3 – Critérios de performance

Para avaliar os diversos modelos a serem computados é necessário estabelecermos critérios de performance. Em geral, a melhor forma de se testar um modelo de Machine Learning consiste em aplicá-lo em um conjunto de dados que não foi utilizado em seu treinamento. Nesse sentido, utilizaremos as seguintes métricas usuais de performance, derivadas da Matriz de Confusão (GÉRON, 2019):

- ACC - Acurácia;
- PRC - Precisão;
- REC - Recall., também chamada de Sensitividade ou Taxa de Verdadeiros Positivos;

- TNR - Especificidade, também chamada de Taxa de Verdadeiros Negativos (TVN);
- F1S - F1 Score;
- ROC - Área sobre a curva Receiver Operating Characteristic;

Seguimos HUAYANAY et al., 2019 e, adicionalmente aos métodos acima, aplicaremos os seguintes critérios adequados para análise comparativa de métodos para tratamento de dados desbalanceados:

- CSI - Jaccard Index ou Critical Success Index (JACCARD, 1901; JOUSSELME et al., 2001);
- GSS – Gilbert Skill Score (SCHAEFER, 1990);
- SSI - Sokal & Sneath Index (SOKAL & SNEAT, 1963);
- FAITH - Faith Index (FAITH 1992; 1994);
- PDIF – Pattern Difference.

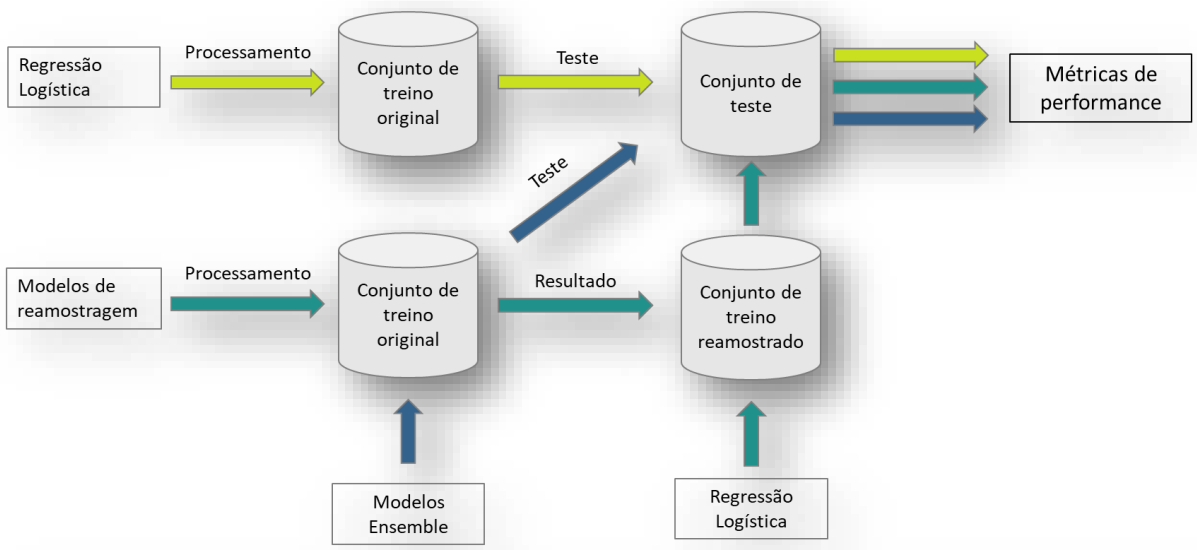
Os diferentes critérios estão incluídos no Notebook 1 (Anexo 1).

3.7 – Aplicação dos modelos

O procedimento utilizado para análise consistiu em dois passos. Primeiro aplicamos o modelo baseline no conjunto de teste e calculamos as métricas de performance. Em seguida utilizamos as técnicas de tratamento de dados desbalanceados e aplicamos novamente o modelo baseline, agora na base reamostrada. Para cada técnica do segundo passo calculamos as métricas de performance.

Os modelos da família Ensemble Classifier foram aplicados diretamente no conjunto de treino original por serem caracterizados por realizarem amostragem interna e por possuírem métodos de ajuste. A Figura 1 exibe detalhes do processo seguido para ambos os diferentes grupos de algoritmos.

Figura 1 – Diagrama de modelagem



Primeiramente aplicamos a Regressão Logística ao conjunto de treino original e depois aplicamos o modelo ajustado no conjunto de teste, registrando as métricas de performance (fluxo verde claro). No passo seguinte processamos a maior parte dos modelos de reamostragem no conjunto de treino original, obtendo um conjunto de treino reamostrado, sobre o qual aplicamos a Regressão Logística, obtendo novas métricas de performance. Este segundo passo foi repetido para cada um dos algoritmos (fluxo verde escuro). Por último, processamos os modelos Ensemble no conjunto de treino original e depois aplicamos os modelos ajustados diretamente no conjunto de teste, obtendo novas métricas de performance (fluxo azul)

Fonte: Autor

Adicionalmente, a fim de verificar se um modelo baseline otimizado com hiper parâmetros teria melhor performance do que um modelo não otimizado, utilizamos um segundo modelo baseline processado após o uso de técnicas de Grid-Search. A partir de um conjunto prévio de hiper-parâmetros, este procedimento escolhe o modelo com melhor performance (FACELI et al. 2011; GÉRON, 2019).

Durante os procedimentos de modelagem tivemos dificuldades com o modelo Condensed Nearest Neighbour (HART, 1968), que não processou em nenhuma interface de programação disponível. Após diversas tentativas sem sucesso, abandonamos a ideia de utilizar este modelo.

4 RESULTADOS

A Tabela 10 contém o sumário de resultados do modelo baseline de Regressão Logística sem otimização de hiper parâmetros, ou seja, com as condições padrão da biblioteca, aplicado a base de dados teste descrita na seção 3.3.

Esse sumário foi obtido através do pacote Statsmodels (SEABOLD & PERKTOLD, 2010) disponível para Python considerando o Notebook 1. Com exceção da variável JOB, os coeficientes das demais variáveis foram considerados significativos à 0,1%, indicando a rejeição da hipótese nula de que seriam estatisticamente iguais a zero. Com isso, as variáveis explicativas selecionados demonstram serem relevantes para o processo de modelagem, impactando a variável resposta.

Tabela 12 – Resultados do modelo de Regressão Logística não otimizado (baseline).

Variável	Coeficientes	Erro padrão	z	P>|z|	Intervalo de Confiança	
					[0.025	0.975]
DURATION	10.530	0,018	57.115	0,000 ***	1.017,0	1.089,0
EURIBOR3M	-0,855	0,020	-43.280	0,000 ***	-0,894	-0,817
CONS.CONF.IDX	0,123	0,019	6.595	0,000 ***	0,086	0,159
JOB	0,101	0,066	1.531	0,126	-0,028	0,231
DEFAULT	-21.079	0,033	-63.365	0,000 ***	-2.173,0	-2.043,0
CONTACT	-12.758	0,035	-36.260	0,000 ***	-1.345,0	-1.207,0
MONTH	12.873	0,070	18.460	0,000 ***	1.151,0	1.424,0
POUTCOME	22.345	0,078	28.726	0,000 ***	2.082,0	2.387,0

Significância para a estatística z: * p<0,05; **p<0,01; ***p<0,001 .
Variável dependente: 1 se o prospect da ação fez o investimento, 0 se o contrário.

Informações adicionais		**Avaliação**	
Amostras:	32.950	Pseudo R-quadrado.:	0,122
Graus de liberdade:	7	Log-Likelihood:	-10.175
Método:	MLE	LL-Null:	-11.585
		LLR p-value:	0,000

A Tabela 11 exibe os resultados após processamento das técnicas de reamostragem e treino dos modelos no conjunto de treino, e verificação no conjunto de teste. Na coluna “Modelo” temos o nome dos modelos, sendo que “Log Reg” e “Log Reg Tuned” são duas versões para o modelo baseline de Regressão Logística, sendo o primeiro com os hiper parâmetros default e o segundo otimizado após o uso da técnica de Grid-Search. As demais linhas da coluna “Modelo” contêm os modelos de reamostragem utilizados, conforme descritos nas Tabelas de 1 a 4.

Tabela 13 – Resultados de processamento dos modelos

Família de modelos	Modelo	Amostras	Tamanho das classes	ACC	PRC	REC	TNR	F1S	AUC	CSI	GSS	SSI	FAITH	PDIF
Baseline	Log Reg	8.238	0 = 7.303; 1 = 935	0,908	0,665	0,389	0,975	0,491	0,924	0,326	0,286	0,194	2,323	0,006
	Log Reg Tuned	8.238	0 = 7.303; 1 = 935	0,909	0,670	0,387	0,976	0,491	0,924	0,325	0,286	0,194	2,328	0,006
Sobre-amostragem (Over-sampling)	SMOTE	58.490	0 = 29.245; 1 = 29.245	0,842	0,407	0,859	0,840	0,552	0,924	0,381	0,308	0,236	1,729	0,009
	SVMSMOTE	58.490	0 = 29.245; 1 = 29.245	0,851	0,422	0,836	0,853	0,561	0,924	0,390	0,318	0,242	1,804	0,010
	ADASYN	58.878	0 = 29.245; 1 = 29.633	0,822	0,380	0,899	0,812	0,534	0,924	0,365	0,287	0,223	1,586	0,008
Sub-amostragem (Under-sampling)	TOMEK	31.903	0 = 28.198; 1 = 3.705	0,912	0,648	0,481	0,967	0,552	0,924	0,382	0,337	0,236	2,392	0,007
	NM	7.410	0 = 3.705; 1 = 3.705	0,832	0,375	0,730	0,844	0,496	0,885	0,330	0,256	0,197	1,621	0,017
	ENN	29.350	0 = 25.645; 1 = 3.705	0,899	0,545	0,658	0,930	0,596	0,924	0,425	0,369	0,270	2,268	0,010
	RENN	28.414	0 = 24.709; 1 = 3.705	0,892	0,517	0,716	0,914	0,600	0,924	0,429	0,369	0,273	2,195	0,010
	AllKNN	28.819	0 = 25.114; 1 = 3.705	0,894	0,525	0,692	0,920	0,597	0,924	0,426	0,367	0,270	2,216	0,010
	OSS	31.084	0 = 27.379; 1 = 3.705	0,911	0,647	0,481	0,966	0,552	0,924	0,381	0,337	0,236	2,390	0,007
	NCR	29.346	0 = 25.641; 1 = 3.705	0,899	0,547	0,655	0,931	0,596	0,924	0,424	0,369	0,269	2,272	0,010
Sobre-amostragem seguida por sub-amostragem	IHT	25.944	0 = 22.239; 1 = 3.705	0,867	0,451	0,800	0,875	0,577	0,923	0,405	0,338	0,254	1,940	0,010
	SMOTET	55.740	0 = 27.870; 1 = 27.870	0,838	0,401	0,865	0,835	0,548	0,924	0,378	0,303	0,233	1,701	0,009
	SMOTEENN	47.265	0 = 24.026; 1 = 23.239	0,826	0,385	0,883	0,819	0,536	0,924	0,366	0,289	0,224	1,613	0,008
Ensemble Classifiers com amostragem interna	EEC	41.188	-	0,867	0,456	0,884	0,865	0,601	0,924	0,430	0,362	0,274	1,962	0,006
	BRFC	41.188	-	0,847	0,419	0,917	0,838	0,576	0,924	0,404	0,331	0,253	1,781	0,005
	RUSBC	41.188	-	0,875	0,473	0,868	0,876	0,613	0,924	0,442	0,375	0,283	2,045	0,007

Em primeiro lugar, destacamos que, conforme esperado, o índice de Acurácia (ACC) apresentou ótimos resultados para todos os modelos em razão da natureza do conjunto de dados, em que a classe majoritária possui mais de 90% das amostras.

Para a área sobre a curva ROC (AUC) observamos que os valores são praticamente os mesmos, exceto na casela de decimais entre os diferentes modelos, demostrando que esses três indicadores tradicionalmente utilizados em Machine Learning não são adequados quando existe desbalanceamento de classes na variável resposta

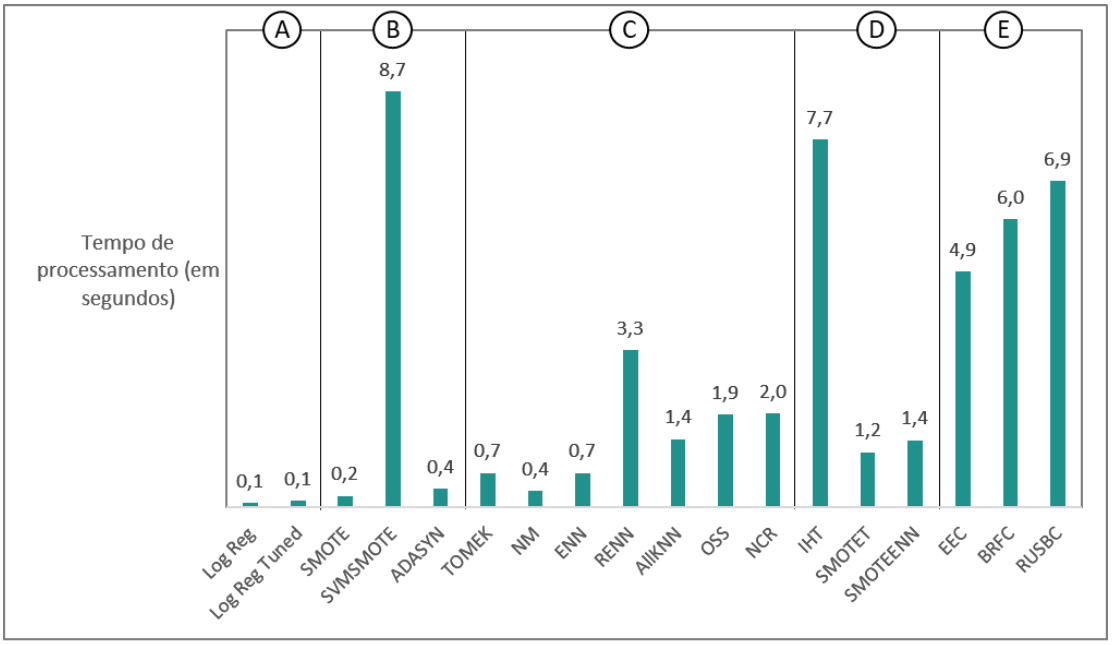
Em segundo lugar, vemos que os dois modelos baseline, como esperado, obtiveram a princípio enganoso, quando se olha o ACC. Comparando-se o nível de Recall. ou Taxa de Positivos Verdadeiros (REC) o modelo BRFC - Balanced Random Forest supera todos os demais em performance. Considerando-se o Índice de Jaccard (CSI) alguns modelos da classe de Sub-amostragem performaram melhor do que os Ensemble Classifiers, com exceção do modelo EasyEnsembleClassifier (EEC). O mesmo pode ser dito avaliando-se a performance com os indicadores CSS, SSI e PDFI.

Em geral, os resultados evidenciam os ganhos de performance obtido com técnicas de reamostragem para tratamento de classes binárias desbalanceadas.

Adicionalmente, avaliamos o custo computacional de cada modelo para uma análise da relação custo-benefício de cada. O Gráfico 17 exibe o tempo de processamento, em segundos, de cada modelo utilizado. Os modelos de Sub-amostragem apresentaram, em média, menor custo

computacional que os demais, enquanto os modelos da classe Ensemble Classifiers são os que mais demandam processamento que mais demandam processamento. Esse custo, porém, é compensado pela alta performance desta categoria.

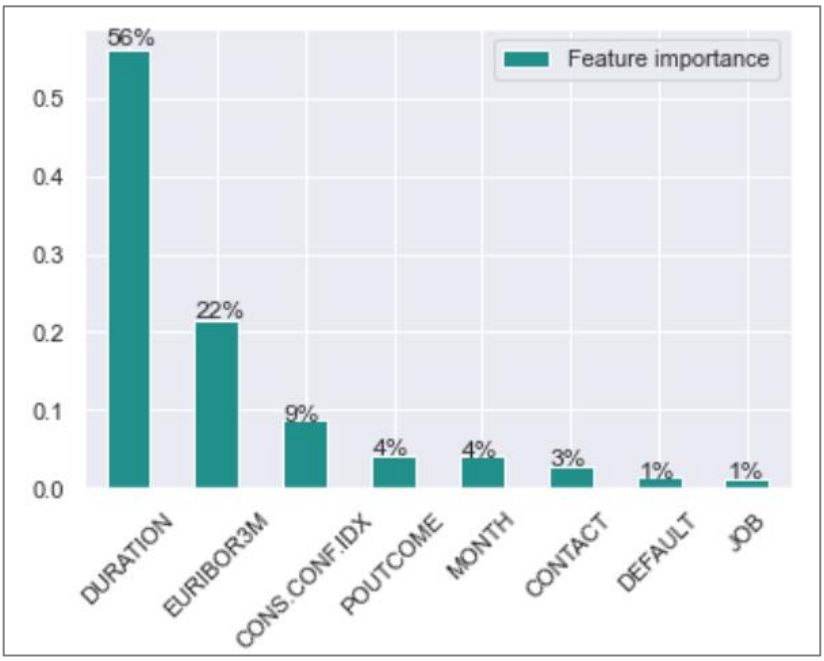
Gráfico 17 – Tempo de processamento dos modelos



A: Baseline; B: Sobre-amostragem (Over-sampling); C: Sub-amostragem (Under-sampling); D: Sobre-amostragem seguida por Sub-amostragem; E:Ensemble Classifiers com amostragem interna

A partir do modelo BRFC - Balanced Random Forest, estimamos o grau de importância de cada variável, expresso no Gráfico 18. O tempo de contato (DURATION) é a variável explicativa de maior importância no modelo, seguido da taxa Euribor (EURIBOR3M) e do índice de confiança do consumidor (COM.CONF.IDX). Isso significa que a probabilidade de um cliente realizar aplicações financeiras no banco em questão é diretamente proporcional ao tempo decorrido no contato realizado pela área de telemarketing do banco. Além disso, a decisão do cliente é afetada por fatores externos, de maneira que quanto piores os indicadores do ambiente econômico, menor chance do prospect fechar negócio com a instituição. Outros fatores apresentaram menor impacto na decisão do prospect, tais como o resultado da campanha de marketing anterior (POUTCOME), o mês da realização de contato (MONTH) e o meio de contato (CONTACT).

Gráfico 18 – Nível de importância de cada variável



5 DISCUSSÃO E CONCLUSÕES

Neste trabalho exemplificamos os problemas resultantes da utilização de modelos de Machine Learning diretamente sobre bases de dados desbalanceadas. Em especial, vimos que os indicadores tradicionais, principalmente a acurácia e a área sob a curva ROC são inadequados para mensuração de performance em razão da predominância da classe negativa na variável resposta, sendo necessário introduzir novos indicadores, como fizemos, e avaliá-los caso a caso, de acordo com o problema em questão. Em decorrência mostramos que a aplicação de técnicas de reamostragem para reequilibrar a proporção de classes nas amostras em conjunto com outros indicadores de performance, além dos tradicionalmente usados, são procedimentos adequados para tratamento dos dados, produzindo respostas satisfatórias quando aplicados ao conjunto de teste. Assim, entendemos que atingimos nossos objetivos iniciais de explicar os efeitos do desbalanceamento de classes binárias na modelagem de diversos fenômeno, de trazer um exemplo da aplicação com as bibliotecas disponíveis na linguagem Python e também de demonstrar que as métricas tradicionalmente utilizadas nem sempre apresentam bons resultados quando se trata de processar bases desbalanceadas na variável resposta.

Certamente este trabalho tem limitações e aponta para novas oportunidades de pesquisa. O fato de utilizarmos apenas um conjunto de dados é talvez a maior das limitações apresentadas. Em pesquisas futuras é importante utilizar, por exemplo, um conjunto de bases criadas por simulação, variando-se, por exemplo a proporção entre as classes da variável resposta. Esse procedimento poderá trazer mais elementos para se responder à questão sobre qual é a técnica de reamostragem que se aplica a um maior número de cenários. Certamente, também, não é possível afirmar categoricamente qual das métricas de avaliação de performance é a melhor, e nem esse era nosso objetivo, dado que este é um problema em aberto e que pode ser melhor explorado em estudos futuros.

REFERÊNCIAS

ANACONDA SOFTWARE DISTRIBUTION. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. <https://anaconda.com>.

BATISTA, G. E. A. P. A.; BAZZAN, A. L. C.; MONARD, M. C. **Balancing training data for automated annotation of keywords: A case study.** Proceedings of the 2nd Brazilian Workshop on Bioinformatics, pp. 10-18, 2003.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. **A study of the behavior of several methods for balancing machine learning training data.** ACM Sigkdd Explorations Newsletter, vol. 6(1), pp. 20-29, 2004.

CHAO, C.; LIAW, A.; BREIMAN, L. **Using random forest to learn imbalanced data**. University of California, Berkeley 110 (2004), pp.1-12, 2004.

CHAWLA, N. V.; BOWYER, K. W.; HAL., L. O.; KEGELMEYER, W. P. **SMOTE: synthetic minority over-sampling technique.** Journal of Artificial Intelligence Research, pp. 321-357, 2002.

COX, D.R. **The Analysis of Multivariate Binary Data**. Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 21, No. 2, pp. 113-120, 1972.

DA SILVA, A. N.; ANYOSA, S.; BAZÁN, J. L. **Modelagem Bayesiano de regressão binária para dados desbalanceados usando novas ligações**. Rev. Bras. Biom., Lavras, v.xx, n.x, p.1-10, 2021

FACELI, K.; LORENA, A.C.; GAMA, J.; CARVALHO, A.C.P.L.F. **Inteligência Artificial: Uma abordagem de aprendizado de máquin**a. LTC, Rio de Janeiro, 2011

FAITH, D.P. **Conservation evaluation and phylogenetic diversity**. Biol Cons 61: 1–10, 1992

FAITH, D.P. **Genetic diversity, and taxonomic priorities for conservation**. Biol Cons 68: 69–74, 1994

GALAR, M.; FERNANDEZ, A.; BARRENECHEA, E; BUSTINCE, H.; HERRERA, F.; **A review on ensembles for the class imbalance problem: bagging-, bosting-, and hybrid-based approaches**. IEEE Trans Syst Man Cybern Part C 42(4): 463-484, 2012

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlo**w. Second Edition, O'Reilly Media, Inc, Sebastopol, CA, 2019.

HAIXIANG, G; YIJINGA, L.; SHANG, J.; MINGYUNA, G; YUANYUEA, H.; BING, G. **Learning from class-imbalanced data: Review of methods and applications**. Expert Systems With Applications. Volume 73, Pp 220-2391, May 2017.

HAN, H.; WANG, W.-Y.; MAO, B.-H. **Borderline-smote: a new over-sampling method in imbalanced data sets learning**. Proceedings of the 1st International Conference on Intelligent Computing, pp. 878-887, 2005.

HART, P. E. **The condensed nearest neighbor rule**. IEEE Transactions on Information Theory. vol. 14(3), pp. 515-516, 1968.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. **ADASYN: Adaptive synthetic sampling approach for imbalanced learning**. Proceedings of the 5th IEEE International Joint Conference on Neural Networks, pp. 1322-1328, 2008.

HE, H; GARCIA, E. A. **Learning from Imbalanced Data**. IEEE Transactions on Knowledge and Data Engineering, Volume: 21, Issue: 9, Sept. 2009.

HE, H.; MA, Y., eds. **Imbalanced Learning: Foundations**, Algorithms, and Applications. New York: Wiley; 2013.

HIPPEL, E. V. **Learning from Open-Source Software**. MIT Sloan Management Review, Massachusetts Institute of Technology, Summer 2001.

HOSMER, D. W.; LEMESHOW, S. **Applied Logistic Regression.** Second Ed. New York: John Wiley & Sons, Inc., 2000.

HUAYANAY, A. C; BAZÁN, J. L.; CANCHO, V. G.; DEY, D. K. **Performance of asymmetric links and correction methods for imbalanced data in binary regression**. Journal of Statistical Computation and Simulation. Volume 89 - Issue 9, 2019.

JACCARD, P. **Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines** Bulletin de la Société Vaudoise des Sciences Naturelles, 37, 241–272, 1901

JOUSSELME, A.-L.; GRENIER, D.; BOSSÉ, E. **A new distance between two bodies of evidence**, Information Fusion 2, 91–101, 2001

KAHNEMAN, D. **Rápido e Devagar: duas formas de pensar**. Objetiva, Rio de Janeiro, 2011

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. Springer, New York, NY, 2013.

LAURIKKALA, J. **Improving identification of difficult smal. classes by balancing class distribution**. Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe, pp. 63-66; 2001.

LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. **Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning**. Journal of Machine Learning Research 18, 2017

LIU, X.-Y.; WU, J. ZHOU, Z.-H. **Exploratory under sampling for class-imbalance learning.** IEEE Transactions on Systems, Man, and Cybernetics, vol. 39(2), pp. 539-550, 2009.

KUBAT, M.; MATWIN, S. **Addressing the curse of imbalanced training sets: One-sided selection**. Proceedings of the 14th International Conference on Machine Learning, vol. 97, pp. 179-186, 1997.

MAALOUF, M; HOMOUZ, D.; TRAFALIS, T.B. **Logistic regression in large rare events and imbalanced data: A performance comparison of prior correction and weighting methods**. Computational Intelligence.; 34:161–174, 2018.

MANI, I.; ZHANG, J. **kNN approach to unbalanced data distributions: A case study involving information extraction**. Proceedings of the Workshop on Learning from Imbalanced Data Sets; pp. 1-7; 2003.

McKINNEY, W. **Python for Data Analysis: Data wrangling with pandas, NumPy, and Ipython**. O'Reilly Media, Inc, Sebastopol, CA, 2018.

MORO, S.; CORTEZ, P.; RITA, P. **A Data-Driven Approach to Predict the Success of Bank Telemarketing**. Decision Support Systems, Elsevier, 62:22-31, June 2014

NGUYEN, H. M.; COOPER, E. W.; KAMEI, K.; **Borderline over-sampling for imbalanced data classification**. Proceedings of the 5th International Workshop on computational Intelligence and Applications, pp. 24-29, 2009.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, G.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, É. **Scikit-learn: Machine Learning in Python**, JMLR 12, pp. 2825-2830, 2011

PRATI, R. C.; BATISTA, G. E. A. P. A.; SILVA, D. F. **Class imbalance revisited: a new experimental setup to assess the performance of treatment methods**. Knowl Inf Syst 45, pp. 247–270, 2015.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. **Uma experiência no Balanceamento Artificial de Conjunto de Dados para Aprendizado com Classes Desbalanceadas utilizando Análise ROC**. Proceedings of IV Workshop on Advances & Trends in AI for Problem Solving. 2003

ROSSUM, G. V. **Python Reference Manual**. Centre for Mathematics and Computer Science, Amsterdam, 1995.

SCHAEFER. JT. **The critical success index as an indicator of warning skill**. Weather Forecasting. 5(4):570–575, 1990

SEABOLD, S.; PERKTOLD, J. "**statsmodels: Econometric and statistical modeling with python**." Proceedings of the 9th Python in Science Conference. 2010.

SEIFFERT, C.; KHOSHGOFTAAR, T. M.; VAN HULSE, J.; NAPOLITANO; A. **RUSBoost: A hybrid approach to al.eviating class imbalance**. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40.1 (2010), pp 185-197, 2010.

SMITH, M. R.; MARTINEZ, T.; GIRAUD-CARRIER, C. **An instance level analysis of data complexity.** Machine learning, vol. 95(2), pp. 225-256, 2014.

SOKAL, R. R. and SNEATH, P. H. **Principles of Numerical Taxonomy**, San Francisco CA: Freeman, 1963

TOMEK, I. **An experiment with the edited nearest-neighbor rule.** IEEE Transactions on Systems, Man, and Cybernetics, vol. 6(6), pp. 448-452, 1976.

TOMEK, I. **Two modifications of CNN**. IEEE Transactions on Systems, Man, and Cybernetics. vol. 6; pp. 769-772, 1976.

VAN DER PAAL, B. **A comparison of different methods for modelling rare events data**. Thesis submitted in fulfillment of the requirements for the degree of Master of Statistical Data Analysis. Universiteit Gent. Academic year 2013-2014, 2014.

WILSON, D. **Asymptotic Properties of Nearest Neighbor Rules Using Edited Data**. IEEE Transactions on Systems, Man, and Cybernetics, vol. 2(3), pp. 408-421, 1972.