Avaliação da Campanha de telemarketing sobre vendas de plano de depósito bancário a termo

E. S. ITO *
*Ciência da Computação - Mestrado
E-mail: e159086@dac.unicamp.br
T. E. NAZATTO †
†Ciência da Computação - Mestrado
E-mail: t074388@dac.unicamp.br

***Resumo* – Avaliamos a eficiência do modelo de predição de promoção de vendas de depósitos bancários a termo (depósito à prazo em troca de pagamento de juros) para um banco português por meio de telemarketing e também tentamos traçar o perfil de clientes que aceitam esse tipo de plano, se há alguma influência devido à atributos (features) próprios, algum contexto sócio-econômico, ou a forma de abordagem do operador de telemarketing ao cliente. Diferente do estudo realizado por Moro et Al. [1] cujo estudo foi mais para satisfazer o banco com informações detalhadas da campanha para assim customizar futuros investimentos na área de marketing, enquanto que o nosso objetivo é puramente acadêmico, onde iremos aplicar conceitos que melhoram a qualidade da informação dos dados, por meio de preparação de dados, onde realizamos o balanceamento do dataset de treinamento e teste por meio da técnica SMOTE, criação de novas features através de variáveis dummy e a normalização dos dados para assim treinar o modelo, no nosso caso o Logistic Regression (LR) e o Decision Tree (DT). Essa metodologia foi fundamental para obter métricas superiores aos casos da referência. Obtivemos métricas (accuracy, recall, f1-score) com valores superiores a 94% enquanto que Moro et Al. [1] obtiveram 90%, Nelson Chris [2] com 91% e Susan Li [3] com 74%, utilizando o mesmo dataset. Diferente dos casos da referência, adicionamos uma atividade extra para realizar análise qualitativa, por meio de estatística básica das features (média, desvio padrão, t-test, quantificação básica) e também com o uso RFE (Recursive Feature Elimination) não para eliminar features, mas para priorizá-las, e assim finalmente poder traçar os perfis dos clientes que aceitam a subscrição do plano de depósito a termo. A eficiência dessa metodologia pode ser comprovada com a melhor qualidade nas métricas e um melhor conhecimento das features. Nosso projeto está armazenado no sítio [1].**

***Palavras-chave* – Machine Learning (ML), dataset (DS), Regressão Logística (RL).**

I. INTRODUÇÃO

A escolha do tema avaliação da campanha de vendas de depósitos bancários a termo foi baseado no artigo de Moro et al [1] e da disponibilidade dos dados no Repositório de Dados de Machine Learning da UCI [4]. O artigo refere-se à campanha de um banco português para obter mais clientes para um produto oferecido sobre depósitos bancários a termo, uma espécie de CDB do Brasil, cuja taxa de juro compete com o euribor3m, que é uma taxa interbancária entre bancos da União Européia, com duração de 3 meses. A campanha foi realizada por meio de chamadas telefônicas ao telefone fixo residencial ou ao celular do potencial cliente. Uma abordagem ao cliente é realizada com uma certa duração onde são explicado o produto de venda em questão e anotados uma série de dados do perfil do cliente e também do momento sócio econômico. Alguns aceitam a subscrição bancária em troca do pagamento do juros após um período de carência e outros simplesmente não aceitam.

Com os dados obtidos, o banco quis saber o que influencia o cliente na subscrição ao produto de venda em questão, se há possibilidade da predição de subscrição ao produto e quais dados dos perfis dos cliente ou quais dados do momento sócio econômico influenciam essa predição, porque se o custo-benefício de contratar uma empresa de telemarketing não compensar, poderia simplesmente diminuir o gasto com propaganda de acordo com Moro et al. [1]. Estes por meio de modelos obtidos por meio de Neural Network, decision tree e SVM obtiveram métricas com valores cerca de 90%. Neste artigo é feito algo similar mas utilizando o modelamento dos dados baseado em Logistic Regression, com a diferença que tratamos melhor os dados por meio de geração de mais features por meio de variáveis dummy a partir de dados categóricos, aumentando o número de 20 para 54. Balanceamentos a amostra com respostas positivas e negativas à campanha utilizando a técnica SMOTE (Synthetic Minority Over-sampling Technique), isso mitigaria a geração de falsos positivos (FP) during o cross-validation (CRV) e durante o teste, normalizamos os dados antes de aplicar o método Logistic Regression, o que tornaria as metricas mais precisas, como o accuracy, precision, recall e f1-score, após a geração da matriz de confusão. Enquanto Moro et al. [1] utilizaram sensibility analysis e decision tree para classificar as features mais importantes para a análise qualitativa, utilizamos estatísticas básicas como média, desvio padrão, quantificação dos dados e técnicas usadas por Susan Li [3] e Nelson Chris [2], como RFE e SMOTE que, unidas, atingiram um F1-score de mais de 94% com o LR e 91% com o DT.

Para reforçar o aprendizado de Inteligência Artificial, somando com a fácil disponibilização públlica do dataset na UCI [4], também encontrado no sítio kaggle.com, o qual se

[1] https://github.com/edbkei/MO416PROJ3/tree/master/Projeto3

observou que se tratava de uma caso típico de aprendizado supervisionado, onde a variável dependente é simplesmente aceitação ou não da subscrição bancária e os dados independentes eram valores categóricos (e.g. profissão, estado conjugal, educação, etc.) e dados numéricos (e.g. idade, número de contatos, valor do euribor3m, etc.), a utilização do método Logistic Regression foi entendido como uma opção viável.

Regressão Logística, ou Logistic Regression, em Machine Learning é uma técnica de aprendizado supervisionado que consiste na regressão de um modelo matemático que relaciona variáveis de entrada $X_i(i = 1, 2, ..., n)$ a diferentes grupos de classificação. Para isso, é usada a função *Sigmoid* para determinar a probabilidade de um determinado conjunto de variáveis a pertencerem a determinado grupo:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

Assim como em Regressões Lineares, na Regressão Logística o melhor modelo de classificação é encontrado através da utilização do algoritmo de Gradiente Descendente, atualizando os valores de θ_j até encontrar o minímo da função custo J:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)})) \tag{2}$$

Quando o melhor modelo de classificação é encontrado, tal classificação está relacionado apenas a uma classe, sendo considerado um modelo de classificação binária por apenas determinar se um dado pode ser considerado da classe em questão ou não.

Este artigo está dividido da seguinte forma: A Seção II descreve como será abordado os problemas que queremos tratar, sobre as bibliografias utilizadas como referência. A Seção III descreve a proposição do trabalho, de como será tratado a análise dos dados e como medir o desempenho da campanha de promoção de vendas de depósitos bancários. A Seção IV será descrito os materiais e métodos utilizados para aquisição, formatação dos dados, criação e teste do modelo. A Seção V mostrará os resultados dos experimentos e uma breve discussão dos resultados da análise. A Seção VI descreverá as principais conclusões do experimento.

II. Abordagem do Problema

A abordagem do problema será por meio das fases do Machine Learning (ML approach), como descritos nas seguintes subseções.

A. Extração dos Dados

Os dados serão extraídos da UCI [4], onde o ds bank-additional-full.csv contém 100% dos dados e será utilizado na fase de treinamento, validação e teste dos dados. O mesmo contém 41188 linhas e 20 colunas (features). E o ds bank-additional.csv contém 10% dos dados e 4199 linhas, mas como são dados extraídos do próprio ds bank-additional-full.csv, não há razão para utilizá-lo. Há outros dois ds, com menos dados, que não serão utilizados para este projeto. São o bank-full.csv e bank.csv.

As variáveis dos datasets (ds) extraídos da UCI [4] são as seguintes:

- Dados bancários do cliente:
  1) **age:** idade (numérico)
  2) **job:** tipo de trabalho (categórico: 'admin.', 'blue-collar', 'entrepreneur','housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
  3) **marital:** estado conjugal (categórico: 'divorced', 'married', 'single', 'unknown'. Nota: 'divorced' significa divorciado ou viuvez).
  4) **education:** educação (categórico: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
  5) **default:** está insolvente? (categórico: 'no','yes','unknown')
  6) **housing:** tem empréstimo de habitação? (categórico: 'no','yes','unknown')
  7) **loan:** tem empréstimo pessoal? (categórico: 'no', 'yes', 'unknown') relativo ao último contato da campanha corrente.
  8) **contact** tipo de contato realizado (categórico: 'cellular', 'telephone').
  9) **month:** mês do ano do último contato (categórico: 'jan', 'feb', 'mar', ..., 'nov', 'dec').
  10) **day_of_week:** dia da semana do último contato (categórico: 'mon', 'tue', 'wed', 'thu', 'fri').
  11) **duration:** duração do último contato em segundos (numeric). Nota importante. Este atributo afeta altamente a variável dependente (e.g. se duration=0, então y='no'). A variável duration não é conhecida antes que a chamada seja concluída. Também, após o fim da chamada, "y"é obviamente conhecido. Dessa forma, esta variável poderia ser somente incluída para propósito de benchmark e poderia ser descartado se a intenção fosse para aplicar num modelo de predição realístico.
- Atributos do contexto social e econômico:
  13) **emp.var.rate**: indicador trimestral da taxa de variação do emprego (numérico).
  14) **cons.price.idx:** índice de preço mensal de preço ao consumidor (numérico) - semelhante ao inpc/ipca do Brasil.
  15) **cons.conf.idx:** índice de confiança do consumidor - indicador mensal (numérico) - semelhante ao ICC da FGV.
  16) **euribor3m:** Taxa euribor 3 meses - indicador diário (numérico).
  17) **nr.employed:** número de pessoas empregadas - indicator trimestral (numérico).
- Outros atributos:
  18) **campaign:** número de contatos realizados durante a campanha e para este cliente (numérico, inclui o

último contato).

19) **pdays:** número de dias que se passaram após o último contato com o cliente desde a última campanha (numérico; 999 significa que o cliente não foi previamente contactado).
20) **previous:** número de contatos realizados antes desta campanha e para este cliente (numérico).
21) **poutcome:** resultado da campanha de marketing prévia (categórica: 'failure', 'nonexistent', 'success')

- Variável dependente (saída do modelo/objetivo desejado):
  21) **y:** o cliente se subscreveu ao plano de depósito a termo (binário: 'yes', 'no').

B. Preparação dos Dados

O notebook Project3.ipynb [2] dá mais detalhes de como foi realizado a preparação dos dados.

A variável dependente y foi transformado em dados binários, em vez dos dados categóricos yes e no. Bem como feito também na variável independente contact, onde o cellular ficou 0, e o telephone ficou 1.

Foi criado uma nova variável independente pdays_no_contact derivado do pdays, de forma que o valor 999 ficou com o valor 1 (não houve contato) e 0 (houve contato), seguindo orientação do Nelson Chris [2].

Foi verificado inicialmente que houve 11.26% de subscrição e 88.72% de não subscrição no ds bank-additional-full.csv. Se utilizado o ds de treinamento sem balanceamento, haveria o risco de o modelo fazer predição com maior número de FP (Falso Positivo). Seguindo a recomendação da Susan Li [3] , o dataset de treinamento foi balanceado utilizando o algoritmo SMOTE (Synthetic Minority Oversampling Tecnique).

As variáveis categóricas job, marital, education, default, housing, loan, month, day_of_week, poutcome foram transformadas em variáveis dummy, cujos valores viraram binários por meio da rotina get_dummies da módulo pandas. Assim como exemplo, a variável categórica marital, que tem valores married, single, unknown, viraram novas variáveis binárias marital_married, marital_single, marital_married. Esta técnica é denominada de *One-Hot Encoding* e tanto Nelson Chris como Susan Li utilizaram tal técnica para variáveis categóricas.

A separação do ds de treinamento em ds da variável independente ("y") e variáveis independentes ("X") por meio do atributo loc do módulo pandas.

A normalização do ds das variáveis independentes ("X") foi realizada após toda a preparação por meio do algoritmo StandardScaler do módulo sklearn.

C. Seleção das Features

Com a utilização do get_dummies, o número de variáveis aumentou de 20 para 54 variáveis independentes. Assim, Susan Li [3] utilizou a técnica RFE (Recursive Feature Elimination) para reduzir a quantidade de features, basicamente lista-se as variáveis independente com os seus pValue por meio da função summary2 do módulo Logit. As features que tiveram o pValue maiores que 5% seriam retirados manualmente da amostra. Mas o resultado não ficou bom, o precision, recall, f1-score ficaram em 74%.

Entretanto, o resultado ficou aquém do que se esperava: As métricas de precision, recall e f1-score ficaram em 74%. Após a constatação, adotamos os procedimentos feitos por Nelson Chris [2] que mantem todos os 54 features ou variáveis independentes. Durante o treinamento e no cross-validation (CRV) a melhora nos resultados foi evidente, com todas as métricas maiores que 90%.

A separação da variável dependente y e das variáveis dependentes X foram realizados pela função loc do módulo pandas. A função train_test_split realizou a separação da amostra de treinamento em 90% e amostra de teste em 10%. Obtendo assim variáveis y e X de treinamento e teste. E o balanceamento das linhas com respostas positivas ("yes"ou 1) e respostas negativas ("no"ou 0) das variáveis de y e X de treinamento e teste foram realizados pelo módulo SMOTE.

D. Treinamento dos Dados

Os dados de treinamento foram treinados com o algoritmo LogisticRegression do módulo sklearn.linear_model, e também com o DecisionTreeClassifer. O modelo foi obtido a partir da função fit (fitness) entre os dados de treinamento balanceados e normalizados de X e y.

E. Validação do Modelo (Cross Validation-CRV)

O modelo obtido a partir da amostra de treinamento foi inicialmente validado pela função score do modelo. Na próxima etapa foi obtido o vetor de predição do y por meio da função predict do modelo, com a qual foi gerado uma matriz de confusão entre o y real e o y predito, a curva ROC do modelo final e também o relatório de classificação por meio da função classification_report do módulo sklearn.metrics, para obter os valores de accuracy, precision, recall e f1-score.

A matriz de confusão mostra uma matriz de 2 x 2, onde estão registrados os números de Falsos Positivos (FP), Falsos Negativos (FN), Verdadeiros Positivos (TP) e Verdadeiros Negativos (TN). Respostas verdadeiras são TP+TN, respostas falsas são FP+FN. A curva ROC é gerado com o módulo roc_auc_score to sklearn, bem como os valores de AUC (Area Under the Curve).

F. Teste do Modelo

Após realizar a predição do modelo obtido na etapa de treinamento dos dados utilizando a amostra de validação (CRV) como base para comparação, a mesma metodologia utilizada na etapa anterior é realizada na amostra de testes. Como tal amostra é separada da etapa de treinamento, a amostra de testes é importante para descobrirmos se o modelo está enviesado de alguma forma, ocasionando casos de *Overfitting* caso tal afirmação seja verdadeira.

Todo modelo tem um certo viés (*bias*) e deve sempre ser revisado de tempos em tempos pois a obtenção de novos

[2]https://github.com/edbkei/MO416PROJ3/tree/master/Projeto3

dados em um certo período pode ocasionar uma alteração do comportamento do modelo, de modo a ser o mais generalista possível.

G. Análise Qualitativa

Foram utilizados os dados de treinamento para realização da análise qualitativa, levando-se em conta que a distribuição dos dados é a mesma que a da amostra de teste. Como o desempenho das métricas no CRV é similar ao de teste, se entende que a distribuição dos dados seja a mesma.

Como não foram descartados nenhuma feature, todas as features serão analisados com mais profundidade, de forma que serão detectados as features com menor relevância e aquelas com maiores chances de obter respostas positivas da campanha. Assim, será inicialmente verificado o pvalue das features com a função summary2 do módulo stasmodels.api, aquelas features com valores maiores que 1% seriam as features com menos relevantes. Por exemplo, education_illiterate tem pvalue 8.72% poderia ser considerado com menor relevância. A amostra de dados tem poucos casos de illiterate (analfabetos) comparados ao university.degree (nível universitário). Para verificar a importância das features, foi utilizado o RFE (Recursive Feature Elimination), assim foram listados as 50, 40, 30, 20, 10 features mais importantes.

As estatísticas básicas (média, desvio padrão)das variáveis numéricas foram obtidas com a função describe() do pandas. E quantificação dos dados das variáveis categóricas foram realizadas pela função value_counts() do pandas.

A comparação de grupos de variáveis numéricas de amostras com respostas positivas e negativas foi realizado com a função ttest_ind do módulo scipy.stats. O pvalue $<1\%$ indica rejeição da hipótese nula de igualdade entre as médias.

Para efeito de comparação, será também utilizado o classificador Decision Tree (DT) do sklearn. Será utilizados todas as 54 features, pois há uma redução da qualidade nas métricas em 1% se reduzisse as features para 50 ou 40.

III. Trabalho Proposto

O trabalho proposto será o desenvolvimento do modelo de predição de subscrição ao depósito bancário por meio do processo explicado no modelo da Figura 1. Basicamente, essa busca ocorrerá de forma interativa tratando o dataset da UCI [4]. A aceitação do modelo no CRV e no teste de modelo será por meio dos valores das métricas. Espera-se um desempenho superior a 90% nas métricas de precision, recall, AUC, F1-score.

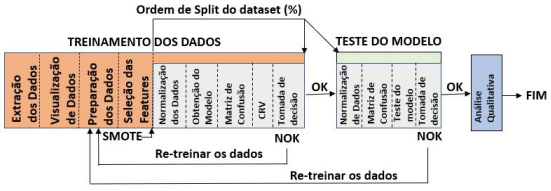

Figura 1. Estratégia de Treinamento, Cross-validation e Teste.

A. Tabelas

Aqui a amostra de teste é submetida ao mesmo modelo obtido na fase de treinamento, validado com o teste CRV (Cross-Validation). Será o obtido a matriz de confusão, medido o accuracy, precision, recall, F1-score, AUC e também será gerado o gráfico ROC.

A tabela I contém os valores das métricas obtidas durante o CRV e teste do modelo.

Tabela I
TABELA DE DESEMPENHO.

Modelo	Amostra	AUC	Accuracy	Precision	Recall	F1
LR(CRV)	90%	0.89	0.94	0.94	0.94	0.94
LR(Teste)	10%	0.95	0.95	0.95	0.95	0.95

A tabela II é um sumário qualitativo das features obtida do Jupyter notebook Python3.ipynb.

O RFE 50+ significa que a feature seria eliminada se houvessem 50 features mais importante entre 54 possíveis. E da mesma forma, há 40+, 30+, 20+, 10+. A feature com 50+ significa que é não significante (N.S.), possivelmente devido a poucos casos na amostra e também foi detectado um bug na feature default_yes, o índice de captura não existia, portanto considerado também não significante (N.S.). Uma feature com 20+ indica que pode estar no grupo dos 30, 40 e 50 mais importantes. O RFE (10) indica que a feature está entre as 10 mais importantes e, assim, pode estar em quaisquer grupos dos 20, 30, 40, 50 features mais importantes. Uma feature pode não ser significativa para a determinação do y (aceitação ou não do plano), mas o seu valor estatístico pode ajudar na intepretação do perfil do cliente.

O dash (-) indica que não existe o valor ou pode ser insignificante, enquanto a notação média (medium) $\pm$ desvio padrão (std) são utilizadas para as features numéricas. A coluna Valor Estatístico mostra valores para os casos positivos (aceitou o plano de depósito bancário) e eventualmente os casos negativos (i.e. não aceitou o plano) são postas na coluna Comentário. Assim, a notação no(....) indica valores em quantidade obtidos para os casos negativos.

Era de se esperar que se houvesse uma interseção de uma feature com resposta positiva [mínimo, máximo] contra resposta negativa [mínimo, máximo], o pvalue seria maior que 1% para aceitar a hipótese nula de igualdade das médias entre duas amostras, porém o único caso que isso ocorreu foi na feature age (idade). Porém isso não ocorreu para outros casos, o pvalue foi menor que 1% indicando que haveria que rejeitar a hipótese nula. Mesmo assim, a faixa de valores foram listadas na Tabela II

IV. Materiais e Métodos

Este trabalho pode ser avaliado por meio de comparação das métricas e pelas estratégias utilizadas. As métricas utilizadas referem-se aos valores obtidos durante a fase de teste do modelo, que são o AUC, Accuracy, Precision, Recall e F1. As

Tabela II
TABELA QUALITATIVA DAS FEATURES

feature	Importância (RFE)	Valor Estatístico	Comentário
age	40+	40 ± 13 anos	pvalue >1%
contact	10+	c30257/t2626	no(c20020/t12863)
duration	40+	9.1 ± 6.6 min	no (3.6 ± 3.4)
campaign	40+	1.7 ± 1.2	no(2.6 ± 2.8)
pdays	40+	789 ± 405 dias	no(984 ± 120)
previous	30+	0.3 ± 0.7	no(0.1 ± 0.4)
emp.var.rate	30+	-1.2 ± 1.6	no(0.2 ± 0.4)
cons.price.idx	40+	93.3 ± 0.6	no(93.6 ± 0.5)
cons.conf.idx	40+	-39.8 ± 5.9	no(-40.5 ± 4.3)
euribor3m	40+	2.1 ± 1.7	no(3.8 ± 1.6)
nr.employed	40+	5094 ± 87	no(5176 ± 64)
pdays_no_contact	50+	-	N.S.
job_blue_collar	(10)	1452	no(7735)
job_entrepreneur	10+	162	no(1197)
job_housemaid	(10)	125	no(863)
job_management	20+	448	no(2304)
job_retired	20+	1612	no(1145)
job_self-employed	10+	159	no(1152)
job_services	10+	441	no(3280)
job_student	30+	564	no(536)
job_technician	20+	1492	no(5437)
job_unemployed	40+	166	no(789)
job_unknown	10+	33	no(264)
marital_married	20+	11748	no(12729)
marital_single	20+	6544	no(8943)
marital_unknown	50+	12	no(62), N.S.
education_basic.6y	(10)	260	no(1862)
education_basic.9y	(10)	796	no(5036)
education_high.school	(10)	2395	no(7580)
education_illiterate	50+	3	no(14), N.S.
education_professional.course	(10)	1027	no(4186)
education_university.degree	(10)	5834	no(9463)
education_unknown	(10)	348	no(1342)
default_unknown	20+	983	no(7305)
default_yes	50+	-	N.S., bug
housing_unknown	30+	116	no(790)
housing_yes	30+	10503	no(15745)
loan_unknown	30+	116	no(1224)
loan_yes	30+	1224	no(4991)
month_aug	20+	2277	no(4969)
month_dec	40+	197	no(85)
month_jul	20+	2387	no(5886)
month_jun	20+	2028	no(4289)
month_mar	30+	757	no(236)
month_may	20+	3799	no(236)
month_nov	10+	1172	no(3327)
month_oct	30+	940	no(364)
month_sep	30+	861	no(286)
day_of_week_mon	(10)	1647	no(6876)
day_of_week_thu	10+	2469	no(6797)
day_of_week_tue	10+	2155	no (6410)
day_of_week_wed	10+	2205	no (6497)
poutcome_nonexistent	30+	13188	no(3681)
poutcome_success	(10)	5942	no(428)

estratégias são o SMOTE para balanceamento das amostras, Normalization dos dados antes de ajustar o modelo, RFE para eliminação de features no caso de [3] e para determinação de importâncias das features no nosso caso, análise qualitativa (A.Q.) e utilização de estatística básica (B.S.), como por exemplo média, desvio padrão, totalização dos dados, aplicação do t-test para comparação de amostra de dados.

A tabela III mostra as métricas utilizadas e as estratégias.

Tabela III
COMPARAÇÃO DE DESEMPENHO ENTRE REFERÊNCIAS SOBRE O USO DE LR E DT E SUAS MÉTRICAS E ESTRATÉGIAS UTILIZADAS.

Técnicas	Este (LR)	Este (DT)	[3]	[2]	[1]
AUC	0.95	0.91	0.74	-	0.71
Accuracy	0.95	0.91	0.74	0.91	-
Precision	0.95	0.91	0.74	-	-
Recall	0.95	0.91	0.74	-	-
F1	0.95	0.91	0.74	-	-
SMOTE	yes	yes	yes	no	no
Normalization	yes	yes	no	yes	no
RFE	yes	no	yes	no	no
A.Q.	yes	yes	no	no	yes
B.S.	yes	no	no	no	yes

V. RESULTADOS E DISCUSSÃO

A campanha de promoção de vendas de depósito bancário a termo, por meio de telemarketing, obteve subscrição de apenas 11%, obtido na etapa de preparação dos dados. Com os dados preparados, balanceados e normalizados, foi possível treinar finalmente os dados. O modelo ajustado com LogisticRegression do módulo sklearn, obteve um escore de 93% durante a fase de treinamento. Depois na fase de CRV (Cross validation), obteve-se 31008 TN, 30577 TP, 1875 FP, 2308 FN por meio da matriz de confusão. Mostrando-se um modelo bastante equilibrado, confirmado com o desempenho das métricas accuracy (acurácia) de 94%, weighted avg (média ponderada) com 94% de precision (precisão), 94% de recall, 94% de f1-score e 89.4% de AUC. Na fase de teste com 10% do dataset, foi realizado o ajuste com o modelo e foi obtido um escore de 94.8%, obteve-se uma matriz de confusão com 3507 de TN, 3443 de TP, 158 de FP e 222 de FN. Utilizando-se da predição da variável y (subscrição ou não ao depósito a termo) com o y real, obteve-se a classificação com o accuracy de 95%, weighted avg de precision de 95%, recall de 95%, f1-score de 95% e AUC de 95%. Mostrando que o modelo obtido na fase de treinamento é também ajustado para a fase de teste. As métricas tiveram um bom desempenho porque não foram descartados nenhumas das 54 features, inclusive aquelas geradas pelas variáveis dummy.

Na análise qualitativa foi possível entender melhor os perfis dos clientes que aceitaram a subscrição da campanha do depósito a termo. Estes tem em média 40 anos, tem formação no mínimo com ensino primário (de 9 anos) até o nível universitário, trabalham na maioria como operário (blue-collar), aposentados, técnicos, em menor grau funcionário administrativo, serviços, até estudante. Uma proporção de 64% de casados e 35% de solteiros. Menos de 5% tem empréstimo pessoal. Apesar da maioria desses clientes não terem sido abordados anteriormente em nenhuma campanha anterior, aqueles que foram abordados com sucesso na campanha passada se subscreveram com maior chance de sucesso na atual campanha. Estes foram abordados 2.78 vezes mais anteriormente que aqueles que não aceitaram nesta campanha. Na maior parte das vezes, os clientes foram abordados por meio de celular. Daqueles que não aceitaram, 40% foram atendidos por meio de telefone fixo. A duração média da chamada foi de 9.1 minutos contra 3.7 minutos para quem não

se subscreveu. Aparentemente, o cliente que vai se subscrever tende a extender mais a duração da chamada. Os meses de maior abordagem ao cliente foram maio, junho, julho e agosto -correspondente ao verão europeu - porém novembro foi uma das 20 mais importante variável, segundo o RFE. Segunda-feira, foi considerado uma das 10 mais importante feature, por alguma razão. O cliente teve em média 1.72 contatos contra 2.62 daqueles clientes que não aceitaram a campanha. As variáveis do momento sócio-econônomico, apesar de ter sido testada com o t-test do sklearn com pvalue $<1\%$, não dá para garantir que os índices sejam diferentes entre os casos positivos e os negativos, pois as médias estão entre 1-2% entre um e outro, estes são o caso do número de empregos, taxa de variação do emprego, índice do preço ao consumidor, e índice de confiança do consumidor, exceto a euribor3m que estava 55% do valor dos casos negativos, pois nem configuram entre os 30 features mais importantes, segundo o RFE.

Em relação ao procedimento de preparação dos dados, é possível verificar que [2] melhorou os resultados de accuracy em relação a [3] apenas com a realização do processo de normalização dos dados, entretanto [3] melhorou o *bias* utilizando o RFE e, principalmente, a técnica SMOTE. Com isso, é possível verificar que mais importante do que a análise em si é verificar se a amostra de dados está harmônica o suficiente e suas features estão praticamente em uma mesma escala, e a combinação das técnicas utilizadas em ambos reflete nos resultados obtidos por este projeto. Também foi observado que com o RFE, que a amostra com as 40 mais importantes features, penaliza mais as features numéricas como age, duration, campaign, pdays, cons.price.idx, cons.conf.idx, euribor3m, nr.employed do que as features categóricas. E exige um adicional esforço com estatística básica para realizar análise qualitativa, a abstração é os valores das features categóricas se dá por meio de quantificação dos valores. Dessa forma, utilizamos também o classificação do Logistic Regression para as features mais importantes (Feature Importances). Uma vantagem dessa classificação é que os valores positivos dessas features são predições para aquelas features que influenciam as respostas positivas (classe 1) e as negativas (classe 0). Observamos que a taxa euribor3m no topo dos mais importantes, seguida de duration (duração em segundos da abordagem ao cliente), poutcome_success (sucesso na campanha passada), education_illiterate (analfabeto), cons.price.idx (índice do preço ao consumidor), month_mar (mês de março) e age (idade do cliente) que são todas de classe 1, as outras features são de classe 0. Pode-se observar que a classificação do Logistic Regression, não penaliza as features de valores numéricos. A única feature listada entre as 10 mais importante no RFE é o poutcome_success, que também é importante no Logistic Regression. Porém o Logistic Regression penaliza as features de cunho sócio econômicos, como o nr.employed (numéro de trabalhadores empregados), emp.var.rate (taxa de variação do emprego), education-university.degree eeducation.university (universitário), job_blue_collar (operário), diferente do RFE, que seriam importante para a classe 1. Não haveria razão lógica para listar o education_illiterate entre as principais, pois casos de analfabetos não significativos.

O Decision Tree (DT) parece realizar uma classificação mais balanceada entre as features categóricas e numéricas, tanto com os atributos do clientes e com as métricas sócio econômicos. No topo das features mais importante, estão o duration, nr.employed, cons.conf.idx, contact, euribor3m, age, cons.price.idx, campaign, education_high.school, day_of_week_mon. E as features realmente menos significativas, inclusive para o RFE, default_yes, marital_unknown, education_illiterate, month_mar, housing_unknown, job_unknown, month_dec, pdays_no_contact. E a vantagem do DT, é que se pode gerar um grafo com a árvore de decisão, onde há nós de decisão com subvalores das features numéricas, assim há vários nós de decisão com subvalores de duration, introduzindo um abstração maior para a análise qualitativa.

VI. Conclusões

Este projeto consolidou o conhecimento de todas as etapas para realização de um projeto de Machine Learning, desde a extração, tratamento, seleção, normalização e balanceamento dos dados, antes do treinamento da amostra ou dataset com algum modelo, a obtenção das métricas com valores com qualidade depende muito de como são tratados os dados nas etapas iniciais.

O ponto forte do projeto deste artigo foi o procedimento adotado na etapa inicial de preparação dos dados. Os pontos fracos seriam a análise qualitativa, a classificação das features mais importantes (feature importance) varia conforme o classificador utilizado, no entanto, o decision tree parece ser mais equilibrado entre outros como o Logistic Regression e o RFE. DT tem a vantagem de poder realizar uma categorização com uma abstração muito maior, por meio de grafos da árvore de decisão.

Utilizar tal procedimento é recomendável para treinar datasets com tipos numéricos e categóricos, pois existe uma boa chance de se obter um bom modelo de predição devido, principalmente, ao uso de normalização e síntese artificial de dados. O classificador decision tree pode ser uma boa alternativa também e tem um bom desempenho. Se a árvore ficar muito grande, o grafo gerado em .png, .jpg., .pdf perde muito a qualidade, portanto é melhor gerar grafo em formato .svg e depois convertê-lo para .png.

Referências

[1] P. Moro, Sergio; Cortez and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014. 1, 5

[2] N. Chris, *Bank Marketing campaign prediction using logistic regression*, 2019 (acessado Julho 24, 2020). [Online]. Available: https://medium.com/@ogbeide331/bank-marketing-campaign-prediction-using-logistic-regression-d3a1072ac155 1, 3, 5, 6

[3] S. Li, *Building A Logistic Regression in Python, Step by Step*, 2019 (acessado Julho 24, 2020). [Online]. Available: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8 1, 3, 5, 6

[4] M. L. Repository, *Bank Marketing Data Set*, 2014 (acessado Julho 18, 2020). [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing 1, 2, 4