Um estudo de caso do uso de mineração de dados e aprendizado de máquina no aprimoramento de inspeções de estações rádio base

Marcelo Veloso Maciel[1] orcid.org/0000-0001-7666-8494

Carmelo Bastos-Filho [1] orcid.org/0000-0002-0924-5341

Victor Mendonça de Azevedo [2] orcid.org/0000-0003-2943-4622

[1] Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil

[2] FITec – Inovações Tecnológicas, Recife, Brasil

E-mail do autor principal: Marcelo Veloso Maciel marcelovmaciel@gmail.com

Resumo

Atualmente, empresas de telefonia despendiam força de trabalho num lento processo de vistoria de Estações Rádio Base. Tendo como cenário a emergência da indústria 4.0, incorporar algoritmos de inteligência computacional no aceleramento desse processo figura como uma vantagem competitiva. É nesse contexto que este trabalho apresenta uma solução algorítmica com o objetivo de auxiliar técnicos e engenheiros de telecomunicação na tarefa de determinar quais itens da vistoria são passíveis de abono. Por meio da utilização de ferramentas de mineração de dados e processamento de linguagem natural extraiu-se a informação necessária para treinar algoritmos de aprendizado de máquina que sugerem aos usuários quais itens tem maior probabilidade de abono. O trabalho, portanto, representa um esforço preliminar na aceleração do processo dessas vistorias.

Palavras-Chave: Estação Rádio Base; mineração de dados; aprendizado de máquina; processamento de linguagem natural.

Abstract

Currently, phone companies spend workforce in a sluggish process of radio base stations inspections. Having as background the emergence of the Industry 4.0, to incorporate computational intelligence algorithms in the speeding up of this process figures as a competitive advantage. It is in this context that this work presents an algorithmic solution with the objective of helping telecommunications technicians and engineers in the task of determining which inspection items are dispensable. The necessary information to train machine learning algorithms to suggest to users which items have the highest probability of dispense was extracted through the application of data mining and natural language processing tools. This work, therefore, represents a preliminary effort in the acceleration of those inspections.

Key-words: *Radio Base Station; data mining; machine learning, natural language processing.*

DOI: 10.25286/repa.v5i2.1322

1 INTRODUÇÃO

Nas últimas décadas a temática do impacto social da inteligência artificial vem tomando centralidade no imaginário prospectivo do cidadão médio, da comunidade científica e dos agentes estatais [1, 2, 3]. A ascensão do assunto na opinião pública não é desconexa de mudanças no contexto econômico e político [4]. A difusão da internet na sociedade, culminando nas tecnologias IoT [5], faz com que dados passem a ser consideradas pela The Economist[1] o novo petróleo.

Esse papel dos dados pressupõe a capacidade dos agentes econômicos de extrair valor deles. É essa a seara de inserção dos algoritmos de inteligência computacional, particularmente os de aprendizado de máquina. Algoritmos de aprendizado de máquina são aqueles que aprendem com uma experiência com relação a alguma tarefa e uma medida de performance se a performance na tarefa melhora com a experiência [6]. Se os dados são o novo petróleo então os algoritmos utilizados para extrair informação e aprender com esses dados podem ser considerados os novos motores da economia.

Embora grandes empresas de tecnologia como Google, Facebook e Amazon façam uso de grandes arquiteturas de redes neurais artificiais as quais necessitam de dezenas de horas de treinamento em unidades de processamento gráfico, a realidade da maior parte das empresas que buscam se inserir nessa nova era algorítmica difere em escopo [7]. Se por um lado a inteligência artificial traz a possibilidade de uma riqueza de aplicações e otimizações no processo produtivo das empresas, por outro lado se faz necessária uma infraestrutura de dados que permita a aplicação dessas técnicas e uma "pipeline" de mineração e recuperação de informação [8]. Ademais a restrição orçamentária e computacional e o imperativo da interpretabilidade[2] do funcionamento dos algoritmos direciona os agentes, nesses casos medianos, à algoritmos mais bem estabelecidos e simples em comparação aos de alta publicização [11].

O presente estudo apresenta um caso de sucesso da aplicação de sistemas inteligentes de recuperação e análise de informação de relativa simplicidade no aprimoramento de um processo rotineiro na indústria de telecomunicações: a inspeção da instalação de estações rádio base. O trabalho propõe uma ferramenta de auxílio à checagem dos itens da estação por meio de ferramentas de mineração de dados e processamento de linguagem natural, aplicadas à extração de informações das estações contidas em documentos armazenados em servidores das empresas de telefonia, e aprendizado de máquina aplicadas, por sua vez, à predição do "Status" dos itens.

O trabalho este organizado da seguinte forma: na Seção 2 é apresentada uma descrição mais precisa do problema. Na seção 3 apresenta-se a metodologia e solução proposta. Por fim na Conclusão discute-se as limitações e prospectos suscitados pelo empreendimento.

2 DESCRIÇÃO DO CASO

Como referenciado anteriormente o sistema alvo de interesse está inserido no âmbito da indústria de telecomunicações. Na rede de celulares a mediação entre o celular dos usuários e as companhias telefônicas é feita pelas Estações Rádio Base (doravante ERB ou sítio celular). São nesses sítios que estão instalados os equipamentos necessários para a comunicação entre aparelhos celulares e as centrais de comunicação das agências telefônicas. Nesses ambientes são realizadas vistorias frequentes tendo em vista sua relevância para a qualidade do serviço de telefonia. Nessas vistorias são checados itens referentes ao tipo de site, equipamentos de radiofrequência, dentre outros. Essa vistoria é um trabalho conjunto entre técnicos que visitam os sítios e engenheiros de telecomunicação que analisam as informações. Atualmente essa troca de informação é feita da seguinte maneira: o técnico visita a ERB e para cada item de um checklist tiram fotos que são enviadas a um sistema onde são aceitas ou rejeitadas pelos engenheiros na central. Contudo, nem todo item precisa ser checado a depender de condições particulares da ERB. Estes itens são, portanto, suprimidos.

[1] Fonte: <https://tinyurl.com/y39u52kk>. Acessado em 1 de Novembro de 2019.

[2] No contexto de aprendizado de máquina a interpretabilidade é definida por [9], p.2 "como a habilidade de explicar ou apresentar em termos compreensíveis para humanos". Uma definição equivalente de interpretabilidade é: o grau no qual um humano pode compreender a causa de uma decisão [10].

Em conversas com técnicos e engenheiros responsáveis pelas inspeções foram identificadas ao menos duas possibilidades de aplicação de inteligência computacional no aperfeiçoamento do processo: a definição de quais itens são suprimidos e quais são avaliados. O problema da dispensa do item, enfoque do presente trabalho, é que os técnicos não sabem de antemão quais itens devem ser suprimidos em um determinado sítio. Ao chegarem a ERB, desta forma, primeiro devem checar dentre centenas de itens em uma lista quais são dispensáveis e só então iniciam o trabalho da vistoria propriamente dita. Isso contribui drasticamente para a lentidão da atividade. Técnicos estimam que a supressão de itens constitui 2/3 do tempo da vistoria. A contribuição do presente estudo para a redução do tempo despendido nessa checagem é descrita em seguida.

3 MÉTODO E SOLUÇÃO PROPOSTA

Têm-se por problema a determinação de quais itens de um checklist são passíveis de abono. Isso pode ser modelado como um problema de classificação binária: dado um conjunto de características de um sítio e qual o item desejamos prever se ele é da classe "abonado" ou não [12]. Especialistas apontaram a seguinte lista de características de um sítio que os próprios técnicos usam para abonar manualmente os itens:

- Tipo de site';
- Tipo de tecnologia;
- Frequência;
- Equipamentos de radiofrequência (RF).

Essas informações, contudo, não estão prontamente disponíveis. Uma fonte possível de informação são documentos disponíveis em um sistema interno das empresas de telefonia cujo acesso foi dado, em formato pdf. Também foi dado acesso à base de checklists dos sítios. Identificou-se 602 ERBs cadastrada nesse sistema das quais foram baixados cerca de 150 documentos em conjunção com os checklists de fevereiro a setembro. Dentre os documentos foram identificados 3 padrões. Como um esforço inicial trabalhou-se na extração de informação de um único padrão. Dado esse recorte de um único tipo de documento, a base criada a partir da intersecção das bases de checklists e de documentos tem uma cardinalidade de 44.

As características das ERBs estavam de presentes de forma não estruturada em tabelas e textos nos documentos. A informação não contida nas tabelas, extraídas por meio de pacotes especializados, foi obtida por meio da tokenização dos textos. Desta forma gerou-se automaticamente uma base de características dos sítios. A partir da intersecção entre a base de características e a base de itens gerou-se um banco de dados de 19000 observações. Na base existem 322 itens únicos, com uma mediana de 243 itens por ERB, e 19 atributos onde todos menos "Item" e "Tipo de Site" são variáveis binárias.

Uma inspeção inicial na base permite identificar um desbalanceamento no número de itens avaliados x os suprimidos, no "Status" do item. O desbalanceamento das classes impacta na performance preditiva de modelos, na medida em que o modelo ganha um viés para a classe majoritária simplesmente pelo maior número de observações dessa classe, aumentando, portanto, o número de falso negativos [12]. Como demonstrado na Figura 1 o número de itens avaliados era mais do dobro dos itens suprimidos, de forma que se optou pela sobreamostragem da classe minoritária por meio de um método de interpolação padrão: o SMOTE (Synthetic Minority Over-sampling Technique) [13].

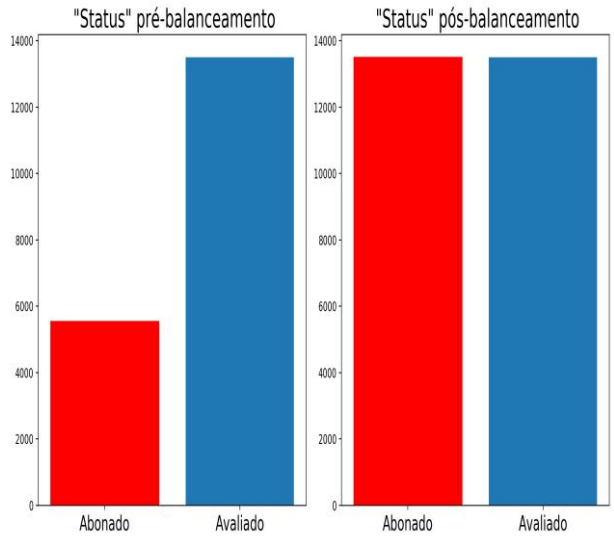

Figura 1 – Número de Abonados vis-à-vis Avaliados pré e pós balanceamento.

Após o rebalanceamento os atributos "Item" e "Tipo de Site" foram codificados por meio de one-hot-encoding. Se um atributo categórico tem n valores essa codificação transforma esse atributo em

DOI: 10.25286/repa.v5i2.1322

n atributos com um deles igual a 1 e os outros iguais a 0. A Tabela 1 demonstra essa codificação. Se um atributo, digamos Fruta, tem como possíveis valores (banana, maçã) e a base tem 3 observações [maçã; banana; maçã], a codificação gera duas novas colunas Maçã e Banana (que substituem a coluna Fruta), preenchidas por 1 quando observação teria o valor equivalente na coluna Fruta e com 0 quando não.

Fruta	Maçã	Banana
maçã	1	0
banana	0	1
maçã	1	0

Tabela 1 – Concatenação entre coluna original e colunas derivadas do one-hot encoding.

Uma vez concluído o pré-processamento partiu-se para o uso de modelos preditivos de aprendizado de máquina. Foi feita a validação cruzada (k-fold com 10 folds), dos seguintes modelos[3]: Decision Tree, Multilayer Perceptron, Logistic Regression, Random Forest, e Xgboost.

O perceptron multicamadas (multilayer perceptron) é uma rede neural de múltiplas camadas (no mínimo 3: uma camada de input, uma escondida e uma de output) com alimentação direta (feedforward) [12]. Foi testada uma MLP com a seguinte configuração:

- (hidden_layer_sizes=(100,),
- activation='relu',
- solver='adam',
- alpha=0.0001,
- batch_size='auto',
- learning_rate='constant',
- learning_rate_init=0.001,
- power_t=0.5,
- max_iter=200,
- shuffle=True.
- random_state=None,
- tol=0.0001,
- verbose=False,
- warm_start=False,
- momentum=0.9,
- nesterovs_momentum=True,
- validation_fraction=0.1,
- beta_1=0.9,
- beta_2=0.999,
- epsilon=1e-08, n_iter_no_change=10),
- early_stopping=False.

A máquina de vetores de suporte (Support Vector Machine) representa, ou mapeia, as instâncias como pontos num espaço vetorial, e busca separa-las por meio de um hiperplano de forma que as categorias ou classes sejam separadas por uma região cuja margem seja a maior possível [12]. A parametrização utilizada na validação cruzada foi:

- C=1.0,
- kernel='rbf',
- degree=3,
- gamma='auto_deprecated',
- coef0=0.0,
- shrinking=True,
- probability=False,
- tol=0.001,
- cache_size=200,
- class_weight=None,
- verbose=False,
- max_iter=-1,
- decision_function_shape='ovr',
- random_state=None.

A regressão logística (Logistic Regression) é um modelo estatístico que representa a relação entre variáveis independentes e uma variável dependente binária por meio de uma função logística. Os parâmetros da função costumam ser estimados por

[3] Os modelos foram implementados usando as bibliotecas de python scikit-learn (https://scikit-learn.org/stable/index.html) e XGboost (https://xgboost.readthedocs.io/en/latest/index.html). As parametrizações usadas foram as padrões das respectivas bibliotecas.

meio de máxima verossimilhança [12]. Os hiperparâmetros do modelo foram os seguintes:

- (penalty='l2',
- dual=False,
- tol=0.0001,
- C=1.0,
- fit_intercept=True,
- intercept_scaling=1,
- class_weight=None, random_state=None,
- solver='warn',
- max_iter=100,
- multi_class='warn',
- verbose=0,
- warm_start=False,
- n_jobs=None,
- l1_ratio=None.

O XGBoost é a implementação da técnica Gradient Boosting, que é um ensemble de estimadores mais simples como árvores de decisões, construído iterativamente. Aplica-se modelos simples na base e avalia-se a função de perda deles, que é utilizada para otimizar a estrutura de novos modelos[4]. A parametrização testada foi:

- max_depth=3,
- learning_rate=0.1,
- n_estimators=100,
- verbosity=1,
- objective='binary:logistic',
- booster='gbtree',
- tree_method='auto',
- n_jobs=1,
- gpu_id=-1,
- gamma=0,
- min_child_weight=1,
- max_delta_step=0,
- subsample=1,
- colsample_bytree=1,
- colsample_bylevel=1,
- colsample_bynode=1,
- reg_alpha=0,
- reg_lambda=1,
- scale_pos_weight=1,
- base_score=0.5,
- random_state=0,
- missing=None.

A árvore de decisão (Decision Tree) consiste em aplicar recursivamente uma métrica, usualmente a impureza de gini, para definir qual atributo separa melhor as instâncias de treinamento. Uma vez definido o atributo separa-se o banco em "galhos" aos quais se aplica novamente essa métrica, até esgotarmos o conjunto de atributos (ou chegar-se a um limite predefinido de profundidade) [14]. A configuração da árvore foi a seguinte:

- criterion='gini',
- splitter='best',
- max_depth=None,
- min_samples_split=2,
- min_samples_leaf=1,
- min_weight_fraction_leaf=0.0,
- max_features=None,
- random_state=None,
- max_leaf_nodes=None,
- min_impurity_decrease=0.0,
- min_impurity_split=None,
- class_weight=None,
- presort=False.

A floresta aleatória (Random Forest), por sua vez, é simplesmente um conjunto de árvores de decisão, onde cada árvore de decisão é treinada em uma amostra sorteada da base (sorteia-se tanto linhas quanto colunas da base, com reposição) [15]. A parametrização testada foi:

- (n_estimators='warn',

[4] Para uma introdução a Gradient Boosting ver https://xgboost.readthedocs.io/en/latest/tutorials/model.html.

DOI: 10.25286/repa.v5i2.1322

- criterion='gini',
- max_depth=None,
- min_samples_split=2,
- min_samples_leaf=1,
- min_weight_fraction_leaf=0.0,
- max_features='auto',
- max_leaf_nodes=None,
- min_impurity_decrease=0.0,
- min_impurity_split=None,
- bootstrap=True,
- oob_score=False,
- n_jobs=None,
- random_state=None,
- verbose=0,
- warm_start=False,
- class_weight=None.

A Figura 2 demonstra a distribuição de acurácias, (número de predições corretas) / (número total de predições), dos classificadores de cada tipo. O perceptron multicamadas foi o classificador com acurácia mais "instável". A despeito da acurácia mediana ser aproximadamente 0,77 houve um caso em que ela foi de 0,50, devido a estocasticidade do algoritmo. A máquina de vetores de suporte foi o algoritmo com pior acurácia, algo a ser investigado em trabalhos futuros. Tendo em vista a interpretabilidade da regressão logística foi feito um Grid Search[5] com objetivo de melhorar a acurácia do classificador, mas sem sucesso.

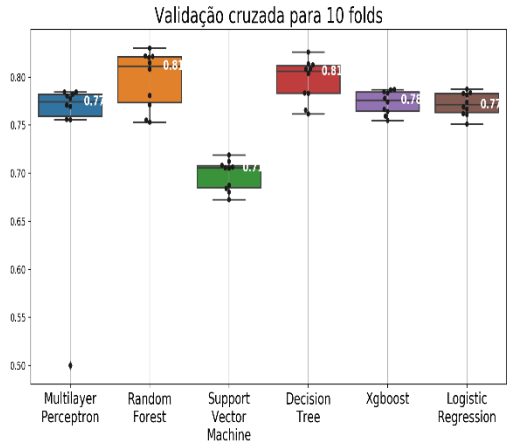

Figura 2 – Distribuição de acurácias. Acurácia mediana anotada em cada caixa.

A árvore de decisão e a floresta aleatória foram os classificadores com melhor performance. O problema de interesse e banco de dados contêm características que [14] indica serem particularmente "tratáveis" por meio de árvores de decisão: os atributos têm poucos valores, o output assume valores discretos, descrições disjuntas são requeridas e os dados de treino podem conter erros, o que explica sua performance.

O conjunto de floresta aleatórias da validação cruzada têm uma acurácia mediana um pouco maior do que as árvores de decisão (0,81 x 0,806), mas com um custo computacional muito maior. A árvore de decisão figura então como o algoritmo utilizado nesse estágio do projeto. A matriz de confusão, que apresenta o melhor classificador identificado por meio de Grid Search, na Figura 3 evidencia uma noção mais completa da performance do classificador: em torno de 16% das classificações como "avaliável" são falsos negativos.

[5] Modelos de aprendizado de máquina apresentam parâmetros, conhecidos como hiperparâmetros, que não são aprendidos internamente, pois são configurações do modelo. É considerada uma boa prática testar diferentes combinações de hiperparâmetros para identificar qual parametrização tem a melhor métrica de avaliação. O Grid Search é quando testa-se exaustivamente as parametrizações estabelecidas [15].

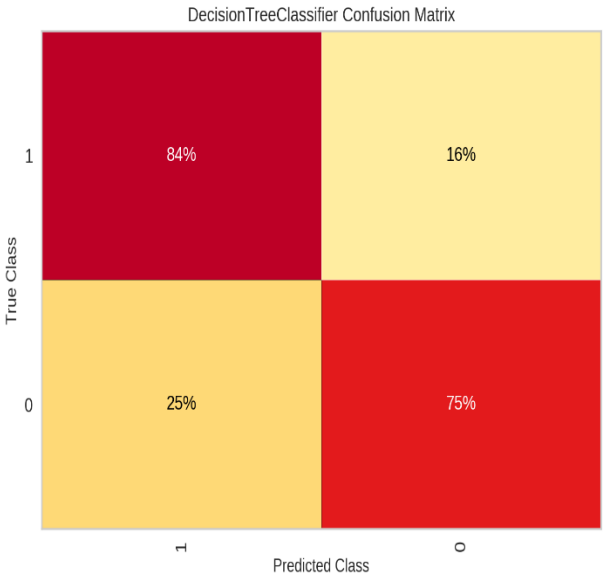

Figura 3 – Matriz de confusão num banco de teste de 60%. 1 é "Abonado".

Em conversas com especialistas, definiu-se que o "output" de interesse dos usuários seria quais itens teriam a maior probabilidade de serem suprimidos em determinado sítio. A árvore de decisão permite determinar a probabilidade de uma instância ser de uma classe, no nosso caso a classe "Abonado". Sendo assim a solução é a seguinte:

1 o usuário indica qual a ERB de inspeção;

2 extrai-se da base construída qual as características do sítio;

3 as características pré-processadas são enviadas ao classificador treinado, a árvore de decisão, que retorna as probabilidades de pertencimento à classe "Abonado" de cada item do site;

4 retorna-se ao usuário a lista ordenada, pela probabilidade decrescente de pertencimento à classe, dos itens do site.

4 CONCLUSÃO

No presente trabalho apresentou-se um caso de aperfeiçoamento do processo de inspeção de estações rádio base por meio de mineração de dados e inteligência artificial. A mineração de dados contidos em documentos contidos nos servidores internos de empresas de tecnologias em conjunção com um modelo simples e interpretável de aprendizado de máquina nos permitiu contribuir no processo produtivo.

Há, contudo, muito a ser feito. Primeiramente, estender a mineração para todos os tipos de documentos contidos nos servidores é o próximo passo. Segundo, investigar como melhorar a acurácia dos classificadores, dado que temos um limiar máximo de aproximadamente 84%. A partir da investigação dos servidores e conversas com usuários, pode-se conjecturar que o não cumprimento dos procedimentos de inspeção nas respostas aos itens gera ruído que confunde os classificadores. A despeito disso, ainda há a necessidade de investigar como aperfeiçoar os algoritmos independentemente da qualidade dos dados. Por fim, a determinação de quais itens são abonáveis é somente a primeira tarefa, pois o problema de determinar, algoritmicamente, quais itens são aceitos ou rejeitados há de requerer uma pletora de estudos adicionais.

Agradecimentos

Agradeço a FITec/SECTI/CMA-Parqtel/UPE/FACEPE..

Referências

[1] CAMERON, J.; WISHER, W. ***Terminator 2: Judgment Day*.** [S.l.]: USA, 1991

[2] COCKBURN, I. M.; HENDERSON, R.; STERN, S. ***The impact of artificial intelligence on innovation*.** [S.l.], 2018

[3] MAKRIDAKIS, S**. The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms.** *Futures*, Elsevier, v. 90, p. 46–60, 2017.

[4] KOGUT, B. M. ***The global internet economy***. [S.l.]: MIT Press, 2003..

[5] GUBBI, J. et al. **Internet of things (iot): A vision, architectural elements, and future directions.** ***Future generation computer systems***, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.

[6] CARBONELL, J. G.; MITCHELL, T. M.; MICHALSKI, R. S. ***Machine learning: An artificial***

DOI: 10.25286/repa.v5i2.1322

***intelligence approach*.** [S.l.]: Springer-Verlag, 1984.

[7] CANZIANI, A.; PASZKE, A.; CULURCIELLO, E. **An analysis of deep neural network models for practical applications.** *arXiv preprint arXiv:1605.07678*, 2016.

[8] MITCHELL, T. M. et al. **Machine learning**. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997.

[9] DOSHI-VELEZ, F.; KIM, B. **Towards a rigorous science of interpretable machine learning.** *arXiv preprint arXiv:1702.08608*, 2017.

[10] MILLER, T. **Explanation in artificial intelligence: Insights from the social sciences**. *Artificial Intelligence*, Elsevier, 2018.

[11] DREISEITL, S.; OHNO-MACHADO, L. **Logistic regression and artificial neural network classification models: a methodology review.** *Journal of biomedical informatics*, Elsevier, v. 35, n. 5-6, p. 352–359, 2002.

[12] JAMES, G. et al. ***An introduction to statistical learning***. [S.l.]: Springer, 2013.

[13] CHAWLA, N. V. et al. **Smote: synthetic minority over-sampling technique**. *Journal of artificial intelligence research*, v. 16, p. 321–357, 2002.

[14] MITCHELL, T. M. et al. **Machine learning**. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997.

[15] GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. [S.l.]: O'Reilly Media, 2019.