Nanodegree Engenheiro de Machine Learning

Proposta de projeto final

Pedro Loureiro

30 de Maio de 2018

I. Definição

Visão Geral do Projeto

Empresas públicas, em geral, que prestam serviço à população recebem milhares de solicitações e reclamações mensalmente. A Compesa, empresa de distribuição de água e saneamento de Pernambuco, e na qual eu trabalho, não foge a esta regra. Apenas no ano de 2017 foram 11941 aberturas de Registro de Atendimento (RA), as quais geram Ordens de Serviço (OS), do tipo Vazamento, destas, 5285 eram válidas para execução, ou seja, não foram abertas em duplicidade ou não era um vazamento real.
Infelizmente boa parte destes problemas tem origem na infraestrutura antiga da cidade, onde são utilizados canos de ferro que enferruja com o tempo e causam os vazamentos e faltas de água em bairros ou cidades inteiras. Como não há orçamento suficiente para atender todas as demandas o mais rápido possível é necessário priorizá-las baseado em fatores como o local em que ocorreu o problema e o número de pessoas afetadas.

Descrição do Problema

Apesar da grande capacidade humana em resolver problemas de priorização, este processo pode tornar-se lento e até ineficaz dependendo da quantidade de informações necessárias para resolver o problema. No caso da priorização das Ordens de Serviço (OSs) há características não quantitativas, como o local da ocorrência, fonte da reclamação e tempo que está aberto que não podem ser calculadas com softwares tradicionais, necessitando intervenção humana. Utilizando

técnicas de classificação é possível calcular um fator de priorização para cada uma destas reclamações, assim, no futuro, quando uma nova OS for criada, será possível calcular este fator e ordená-las, priorizando sempre aquelas com o maior fator calculado. Existe a possibilidade de OSs sempre ficarem para trás, entretanto, estas OSs ficarão mais antigas, aumentando sua chance de serem classificadas como prioritárias devido a data de abertura mais antiga.

Métrica

Algoritmos de classificação possuem diversas técnicas para medir sua efetividade como a acurácia, precisão e sensibilidade.

A acurácia verifica a porcentagem de acertos sobre o total da amostra, entretanto, ela pode não ser suficiente para determinar se uma solução está satisfatória. Em casos em que Falso Positivos são mais importantes que Falso Negativos, por exemplo, esta métrica é falha, pois considera que todos os casos tem a mesma importância, o que no problema apresentado está errado, afinal, é menos grave enviar uma equipe para solucionar um vazamento pequeno, não prioritário (Falso Positivo) do que não enviar uma equipe para um vazamento grande, prioritário (Falso Negativo)

A precisão considera o total de amostras corretamente preditas (Verdadeiro Positivo) sobre o total de amostras consideradas verdadeiras, ou seja, preditas como verdadeiras (Verdadeiro Positivo + Verdadeiro Negativo) respondendo a pergunta: Do total de OSs classificadas como prioritárias, quantas realmente eram prioritárias ?.

A sensibilidade divide o total de amostras corretamente preditas (Verdadeiro Positivo) sobre o total de amostras que são realmente verdadeiras (Verdadeiro Positivo + Falso Negativo). Do total de OSs realmente prioritárias, quantas marcamos como prioritárias ?.

Como tanto a precisão como a sensibilidade são métricas importantes para o problema abordado utilizaremos o F1 Score como métrica principal de avaliação. O F1 score é um valor entre a precisão e a sensibilidade e pode ser calculado como 2*((precisão*sensibilidade)/(precisão+sensibilidade)).

II. Análise

Exploração de Dados

A Compesa possui um software comercial, o GSAN, no qual são armazenados todos as Ordens de Serviço. O software permite filtrar estas informações por tipo, incluindo vazamentos, que será trabalhado. Os dados serão filtrados para limitá-los aos que foram registrados no ano de 2017 e válidos para execução, ou seja, não foram abertos em duplicidade ou não era um vazamento real, fornecendo as seguintes características:

- Data Abertura da OS (**DATA_ABERTURA_OS**): Data em que a Ordem de Serviço foi aberta. Quanto maior o tempo de aberto mais chances há de ser priorizada.
- Meio de Solicitação (**MEIO_SOLICITACAO**): Forma como o RA foi aberto gerando as OSs.
    - - 1: BALCAO
    - - 4: INTERNO
    - - 6: TELEFONE
    - - 8: INTERNET
    - - 9: OUTROS
    - - 10: AGENCIA MOVEL
    - - 11: MOBILE

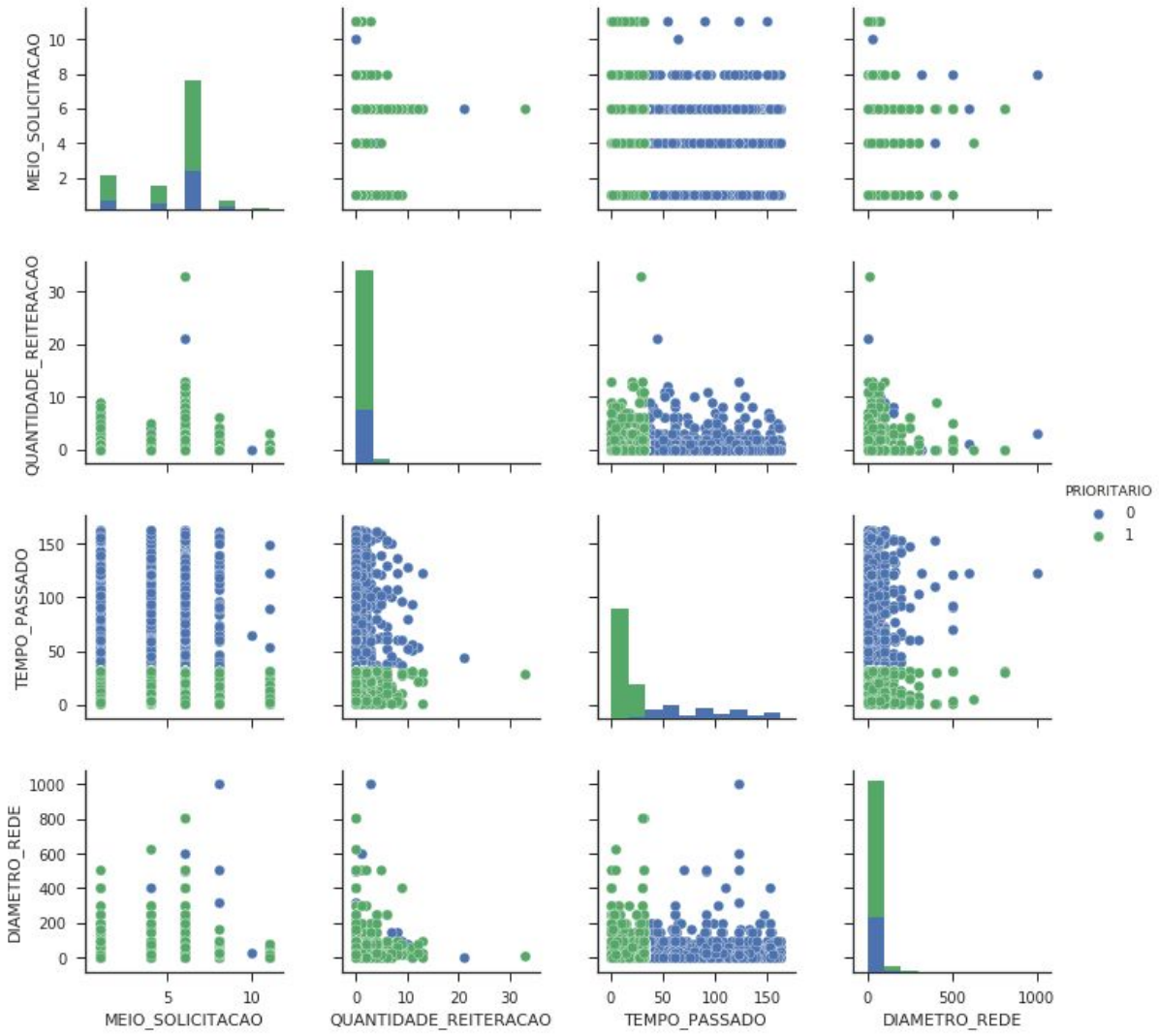
  A empresa mostra interesse em priorizar RAs abertas pelo mobile, pois é mais barato e fideliza o cliente.

- Cidade da Ocorrência (**CIDADE**): Cidade onde a ocorrência acontece.
- Bairro da Ocorrência (**BAIRRO**): Bairro onde a ocorrência acontece.
- Quantidade de Reiterações (**QUANTIDADE_REITERACAO**) : Reiterações são aberturas de novas RAs relativas a RA original informando que o cliente reportou novamente sobre o mesmo problema em datas diferentes.
- Diâmetro da rede (**DIAMETRO_REDE**): O diâmetro da rede é um bom indicador do tamanho do vazamento. Logo, quanto maior o diâmetro maiores as chances de priorização.
- Data de Encerramento da OS (**ORSE_TMENCERRAMENTO**): Data em que a Ordem de Serviço foi concluído. Este atributo não será considerado no treinamento, existindo aqui apenas para referência.

- Tempo Passado (**TEMPO_PASSADO**): É o tempo passado entre a data de abertura e a data de encerramento. Caso este tempo seja 0, indica que não é uma OS válida, sendo removida do conjunto de dados.
- Prioritário (**PRIORITARIO**): Esta coluna não existe no banco de dados e será calculada.Serão consideradas prioritárias aquelas OSs cujo tempo entre a data de abertura e a data de encerramento seja menor que a mediana entre estas datas. É uma coluna binária, 0 ou 1,onde 1 são as OS priorizadas.

Visualização Exploratória

O gráfico abaixo representa a relação dos atributos entre si e sua priorização:



Nota-se que alguns atributos se relacionam diretamente, como a quantidade de reiterações e o diâmetro da rede, onde o aumento de reiterações tende a ter um

menor diâmetro, o que pode ser explicado pelo fato da maioria das redes com menor diâmetro serem residenciais.

Nota-se também a relação entre a quantidade de reiterações e a priorização, onde a maioria das OSs priorizadas são aquelas com maior quantidade de reiterações.

Algoritmos e Técnicas

O problema a ser resolvido aqui é de classificação, pois devemos determinar se uma OS será priorizada ou não e a probabilidade disto ocorrer. Foram escolhidos quatro algoritmos de classificação para realizar os testes:

1. **Naive Bayes**: Como a quantidade de dados para análise é grande, o naive bayes foi escolhido para teste devido a sua velocidade na obtenção do resultado, entretanto, seu score f1 pode ser muito baixo pois ele não entende relações entre features.
2. **Árvore de Decisão:** A grande vantagem das árvores de decisão é seu fácil entendimento, o que facilitaria a explicação da solução para os gestores. A maior desvantagem é sua pouca flexibilidade caso o cenário seja muito diferente do treinado e a tendência a ser ineficiente com o aumento de atributos. É pouco provável que os cenários mudem frequentemente para vazamentos, por isso as árvores de decisão parecem ser uma boa candidata.
3. **Support Vector Machine:** Tem uma série de vantagens como a facilidade em lidar com grandes quantidades de dados e a velocidade de classificação. A maior desvantagem é a dificuldade para realizar o treinamento, pois é necessário ajustar bem o parâmetro *soft margin (c)* para obter um bom resultado. Além disso, para objetivo deste trabalho é necessário obter a probabilidade de acerto no algoritmo, o que o torna o tempo de treinamento ainda mais longo.
4. **Logistic Regression :** Parece ser a técnica mais interessante para o problema que estamos tentando resolver. Logistic regression foi criado exatamente para calcular a probabilidade de algo ocorrer. Este algoritmo funciona apenas para classificações binárias, o que se encaixa perfeitamente no nosso problema.

Estes quatro algoritmos serão comparados quanto ao tempo de execução e o seu f1 score. Aquele que obtiver o melhor resultado será o escolhido.

Benchmark

O assunto em questão está mais restrito a empresas de saneamento ou prefeituras que são responsáveis pela distribuição de água, por isso não foi encontrado nenhum artigo relativo a este assunto para comparar diretamente. Baseado em

outros problemas de priorização, em áreas diferentes, como a de manutenção de máquinas industriais, serão utilizados algoritmos que permitam visualizar a probabilidade de determinada Ordem de Serviço ser executada ou não.

Atualmente a classificação se uma ordem de serviço é prioritária ou não é realizada manualmente e dura algumas horas. O algoritmo será considerado válido se sua pontuação F1 for de ao menos 80%, este valor foi escolhido porque acredito que valores como 90% poderiam causar overfitting deixando OSs que deveriam ser priorizadas fora do ranking de priorização. É possível que este valor possa mudar ao longo do tempo caso seja necessário adicionar mais ou menos OSs para serem executadas.

III. Metodologia

Pré Processamento de Dados

Os dados aqui tratados foram retirados da base do sistema comercial da Compesa. Algumas condições foram criadas na extração:

1. OSs abertas entre 01/01/2017 e 31/12/2017
2. OSs que já foram encerradas.
3. OSs do tipo vazamento.

Ao obter estes dados foi ainda necessário algumas extrações e tratamento de dados. Inicialmente a coluna **TEMPO_PASSADO** foi criada sendo ela a diferença ,em dias, das colunas **ORSE_TMENCERRAMENTO** e **DATA_ABERTURA_OS.**

Com a coluna **TEMPO_PASSADO** criada foi necessário remover os registros cujo **TEMPO_PASSADO** estava com valor igual a 0. Estes registros são inválidos porque representam OSs que foram encerradas mas não foram realmente executadas, a exemplo das duplicadas.

A próxima etapa foi remover os outliers. Neste caso foram removidas as OSs cujo tempo de execução foi pequeno ou grande demais. Com o **TEMPO_PASSADO** tratado foi criada a coluna **PRIORITARIO.**

A coluna **PRIORITARIO** foi criada após ser obtido o tempo médio de execução das OSs. Aquelas OSs que tinham um **TEMPO_PASSADO** menor que o tempo médio foram consideradas prioritárias.

Os dados coletados possuem diversas colunas de dados categóricos, como **CIDADE**, **BAIRRO** e **DIAMETRO DE REDE** e **MEIO DE SOLICITAÇÃO**. Para executar os algoritmos de classificação propostos foi necessário criar Dummy Variables.

Implementação

A implementação exigida não foi de grande complexidade. Houve necessidade apenas de alguns algoritmos para o pré processamento dos dados e outros para medir a qualidade das técnicas escolhidas assim como o tempo de processamento.

As bibliotecas utilizadas foram: Numpy, Pandas, Sklearn e Seaborn. Todas com a versão 3 do Python.

Foi também realizada uma tentativa de uso do PCA para identificar atributos relacionados. Entretanto esta técnica não se mostrou eficiente para dados categóricos, mesmo transformando-os em dummy variables. Durante algumas pesquisas identifiquei que o algoritmo ideal seria o MCA, que foi utilizado também não obtendo bons resultados e por isso foi retirado do código.

Os algoritmos para medição do tempo e qualidade das técnicas foram testados com uma base de 2000 registros. Estes algoritmos mostram o tempo de treinamento e tempo da predição além do valor F1 para o conjunto de treinamento e teste para cada um dos classificadores informados (Nayve Bayes, Árvore Decisão, SVM e Logistic Regression).

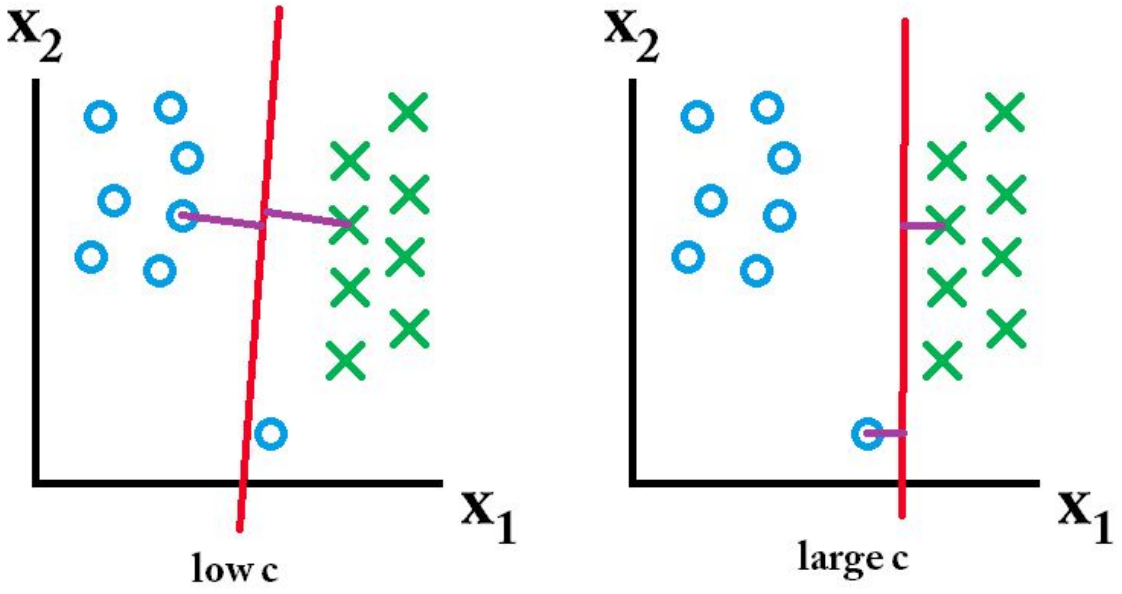
Posteriormente foi criado uma função chamada *predict_otimized* que recebe como parâmetro um classificador e os parâmetros utilizados para sua otimização. Esta função utiliza o gridSearch do Sklearn para calcular algumas combinações de parâmetros e também é responsável por descobrir qual o melhor classificador para o conjunto de dados informado e utiliza-lo para obter as OSs priorizadas.

Refinamento

A utilização do gridSearch permite testar diversos parâmetros possíveis de determinado classificador. Baseado nos resultados obtidos anteriormente o Nayve Bayes foi descartado.

Para a árvore de decisão foi utilizado o parâmetro max_depth (Profundidade Máxima da árvore). Os valores escolhidos foram obtidos por observação, uma vez que notou-se que a medida que o max_depth cresce a pontuação f1 diminuía até obter os mesmos resultados do algoritmo não otimizado.

Em Logistic Regression e no SVM foi utilizado o parâmetro c. Como estes algoritmos utilizam a distância entre um hiperplano e os valores, o parâmetro c indica a margem permitida para encontrar estes valores, quanto mais baixo o valor de c, maior a margem. A imagem abaixo representa como o parâmetro c funciona:



Demonstração do parâmetro c. Fonte: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Logo os seguintes resultados foram obtidos com os 3 classificadores restantes para os dados de teste:

Classificador	Parâmetros	F1 Antes Otimização	F1 Após Otimização
Arvore de Decisão	max_depth':[1,3,5]	0.7400	0.8356
Logistic Regression	'C':[0.3,0.5,1]	0.8271	0.8352
Support Vector Machine	'C':[0.1,0.5,1]	0.8354	0.8356

Como pode ser observado os resultados após otimização foram muito similares. O algoritmo com melhor ganho foi o de Árvore de Decisão com um ganho de 16% no F1 Score.

IV. Resultados

Validação e Avaliação do Modelo

O conjunto de dados foi separado entre treinamento e teste. A validação ocorreu utilizados os dados de teste obtendo melhores resultados após a otimização.

Esta validação ocorreu com 25% do total de dados, ou seja, cerca de 1300 registros foram testados. Como estes dados são reais, acredita-se que a validação esteja correta.

Justificativa

No benchmark foi descrito que um f1 Score de 80% seria o suficiente para validar o modelo. Como mostrado na seção refinamento 3 dos 4 algoritmos testados conseguiram resultados um pouco melhores com os dados de teste, com uma média de 83%.

Acredito que a solução encontrada pode ser utilizada sem problemas em situações reais. Entretanto, algumas outras variáveis, precisam ser introduzidas, como a acessibilidade do local do vazamento e se o local está sem receber água (rodízio).

V. Conclusão

Reflexão

O processo utilizado por este projeto pode ser resumido em:

1. Obtenção dos requisitos com o gerente da área responsável pela execução das OSs.
2. Extração dos dados no GSAN.
3. Tratamento dos dados no projeto.
4. Análise dos dados para verificar relações entre eles.
5. Treinamento da base
6. Teste dos algoritmos sem otimização
7. Otimização e teste dos algoritmos.
8. Extração das OSs priorizadas

O mais desafiador foi encontrar relações entre os dados (Passo 4) uma vez que o planejamento era utilizar PCA, o que não foi possível. Foi interessante perceber entretanto que alguns relacionamentos encontrados foram surpreendentes. Acreditava-se por exemplo que o diâmetro de rede teria maior influência na decisão de OSs prioritárias, o que não aconteceu.

O fato de haver pouca relação entre os dados também preocupou um pouco, mas os algoritmos mostraram que esta relação não precisa existir para se obter um bom desempenho.

Melhorias

Um aspecto que poderia ser melhorado é a otimização dos dados. Seria necessário testar uma maior variedade de parâmetros para verificar se f1 score melhora um pouco mais. Além disso outros atributos, os quais não foi possível extrair informações no momento, poderão ser adicionados, o algoritmo treinado novamente e verificado se os resultados se aproximam ainda mais do mundo real.

Poderia ser considerado também o aspecto do tempo de predição para que não fique inviável obter as OSs priorizadas no tempo mais curto possível.