Universidade Federal de Goiás
Instituto de informática
Profa Nádia Félix Felipe da Silva

Relatório Competição Inteligência Computacional

Alunos:

Ian Marcos Chaves:
Matrícula: 201802684

Thallis André Faria
Matrícula: 201802707

Diciplina: Inteligência Computacional

Dezembro
2022

Universidade Federal de Goiás
Instituto de Informática
Disciplina: Inteligência Computacional

Relatório

Primeiro Relatório da participação dos Alunos Ian Chaves e Thallis André do Curso de Engenharia de Computação da Universidade Federal de Goiás, como requisito parcial para Aprovação da Disciplina Inteligência Computacional.

Dezembro
2022

Conteúdo

1 Resumo

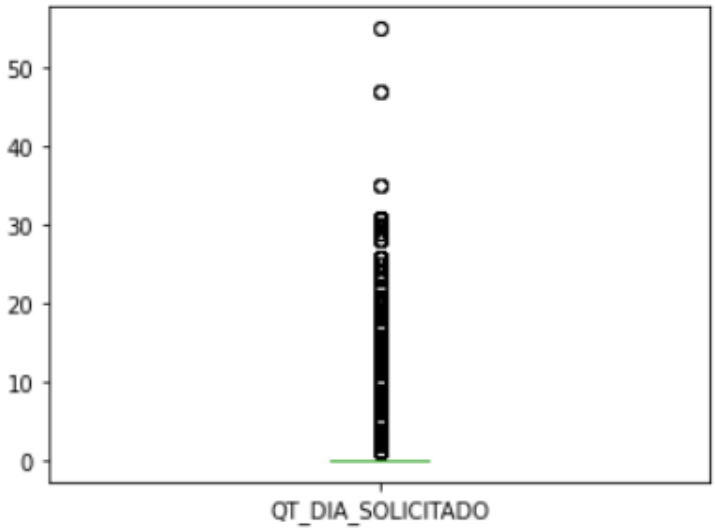
O trabalho consiste em analisar dados médicos de operadoras de planos de saúde, com o intuito de ajudar a possivelmente automatizar processos, no quesito de análise de solicitações. Em um primeiro momento, foi realizada uma análise dos dados, principalmente observando quais campos estavam nulos, qual era a volumetria dos dados, quantas e quais classes possuíam cada campo.

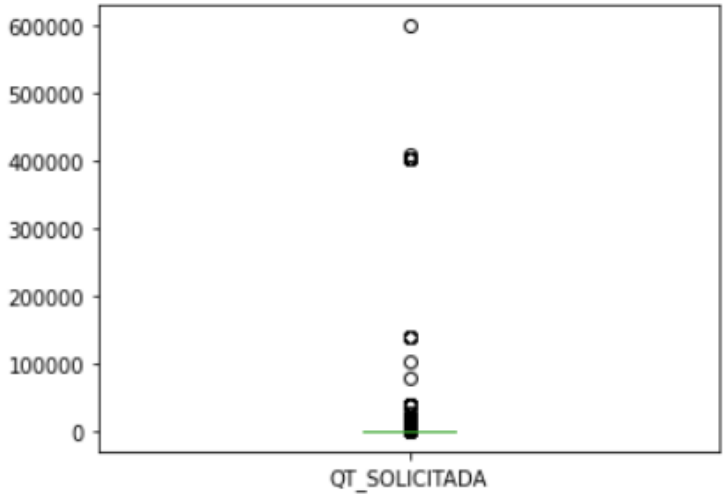
Logo em seguida foram utilizados diversos métodos de pré processamento de dados e feature Engineering, para preparar melhor nossa massa de dados para ser fornecida a um modelo de aprendizado de máquina. Feito isso, a próxima etapa é treinar, e testar modelos, avaliando sua performance a partir de diversas métricas.

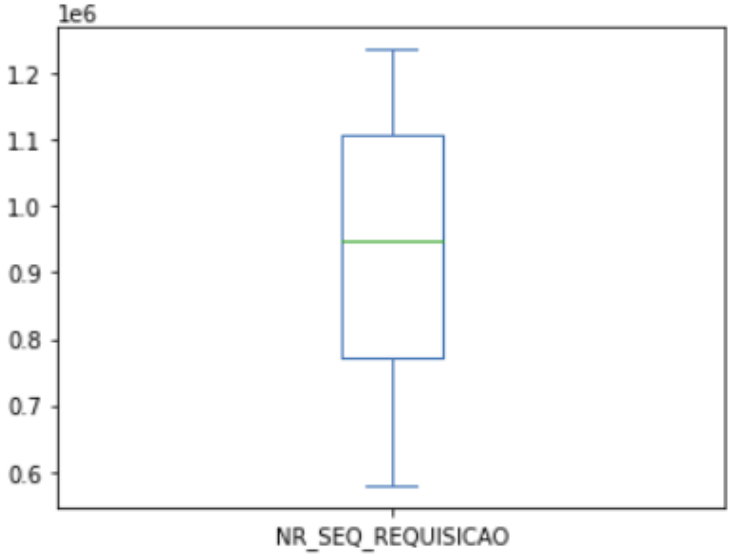
2 Descrição do Conjunto de dados

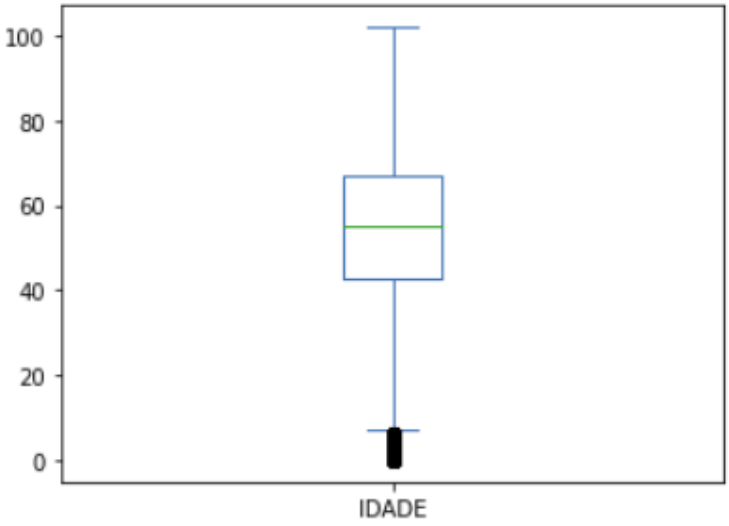
Na área de ciência de dados, fazemos uso de enormes bases, para extrair insights, treinar modelos preditivos, analisar bases históricas e encontrar padrões, prever vendas, classificar clientes, evitar churn, dentre diversas atividades. Para realizar tais tarefas, o uso de dados é essencial, e entre acessar uma base de dados, e gerar algum insight, há um vasto caminho a ser percorrido.

Para isto começamos a analisar a base de dados fornecida, a partir do uso da biblioteca pandas, muito utilizada para análise de dados. Começamos observando os dados numéricos, e para isso utilizamos uma primeira vizualização de dados chamada de BoxPlot:




600000
500000
400000
300000
200000
100000
0
QT_SOLICITADA


1e6
1.2
1.1
1.0
0.9
0.8
0.7
0.6
NR_SEQ_REQUISICAO



A coluna "Idade", foi gerada a partir de outra coluna, chamada "DT-NASCIMENTO", que possui a data de nascimento do solicitante, e assim, a partir disso, calculamos a idade em anos do indivíduo em questão.

```
0         1966-08-10                    0         55
1         1978-01-27                    1         42
2         1967-11-20                    2         54
3         1966-01-13                    3         55
4         1956-05-01                    4         65
             ...                                  ..
227117    1964-07-15                    227117    57
227118    1962-05-19                    227118    59
227119    1968-07-03                    227119    52
227120    1970-04-27                    227120    50
227121    1973-08-26                    227121    48
Name: DT_NASCIMENTO, Length: 227122, dtype: datetime64[ns]   Name: IDADE, Length: 227122, dtype: int64
```

A figura Boxplot, nos mostra uma interessante descrição de dados numéricos, começando por linhas que representam os quartis 0, 25, 50 (mediana) 75, e 100. As linhas do 25 e 75 são representadas em formato de caixa, e dentro dessa caixa temos a mediana dos dados sendo o quartil 50, assim temos uma noção gráfica da distribuição numérica dos nossos dados. Temos no BoxPlot, representados em círculos os outliers, ou seja, os valores que se destoam dos dados pertencentes ao seu dataset.

Nas colunas com dados categóricos, analisa-se o número de classes distintas em uma colunas, assim como a quantidade de valores nulos presentes. Em relação a valores nulos, foi realizado um pré processamento tornando estes, iguais a zero.

3 Descrição de atividades

Tendo em vista a primeira análise dos dados citadas na seção anterior, seguimos para as próximas etapas envolvendo o pré processamento dos dados, para inseri-los em modelos. Um método de pré processamento muito importante para modelos lineares, é a padronização de variáveis numéricas.

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

Assim como observado na fórmula acima, a padronização consiste em utilizar a média dos valores numéricos, assim como o desvio padrão, para atribuir a mesma escala presente originalmente nos dados, porém alterando seus valores para um intervalo entre 0 e 1. Esta etapa é notávelmente importante para a utilização de modelos lineares, como regressão linear e regressão logistica.

Foi utilizado também, como pré processamento de dados categóricos, métodos de encoding, que transformam atributos categóricos de um dataset, em valores numéricos correspondentes. A partir da biblioteca do SkLearn, foi feito o uso do "Label encoder", que analisa uma coluna especificada do dataset, destaca os distintos atributos categóricos presentes nesta coluna, e os transforma em valores numéricos correspondentes. Digamos por exemplo, que um dataset analisado tenha uma coluna chamada "Regiões", e os tipos categóricos distintos dentro desta coluna sejam [Norte, Nordeste, Centro-Oeste, Sul, Sudeste], se fizermos o uso do LabelEncoder nesta coluna, seriam transformados respectivamente "Norte"em 0, "Nordeste"em 1, e assim por diante, formando uma lista [0, 1, 2, 3, 4]. Já o one hot encoding tem como premissa, transformar essa lista citada anteriormente, em uma matriz N por N (no nosso exemplo 5x5), com zeros em todas as linhas e colunas, e na diagonal principal, onde representar aquele atributo categórico, receberá 1, na posição da linha referente. Seja 0 nossa representação da classe norte, a [posição 0,0] dessa matriz, receberá o valor de 1, para representar a mesma classe.

Foram utilizados diversos modelos para análise, dentre eles: Árvores de decisão, florestas aleatórias, Máquinas de vetor de Suporte, K-Nearest Neigburs e LightGBM. Para avaliação optou-se pela utilização da métrica f1-score. O modelo utilizado com mais ênfase, e melhor explorado, foi o LightGBM, que é um modelo de Boosting, baseado em árvores, que faz o uso de técincas

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

como bagging. O modelo LightGBM, cria em um primeiro momento, uma árvore que generaliza a maior parte dos dados, e para o restante dos dados que não foram inicialmente bem divididos, ele gera outra estrutura de árvore e repete esse processo, seguindo as iterações. Esse modelo de boosting, possui diversos hiperparâmetros a serem explorados. Para tal exploração foi feito o uso de uma biblioteca chamada Optuna, no qual varre de forma estocástica o espaço de busca, afim de encontrar os melhores hiperparâmetros para nosso modelo de LightGBM. Após uma extensiva busca, foram encontrados os parâmetros abaixo com melhor resultado:

params = 'n_estimators': 10000, 'learning_rate': 0.29001290111164385, 'num_leaves': 1560, 'max_depth': 20, 'min_data_in_leaf': 4100, 'lambda_l1': 0, 'lambda_l2': 70, 'min_gain_to_split': 0.02564077641748419, 'bagging_fraction': 0.7, 'bagging_freq': 1, 'feature_fraction': 0.9

4 Análise dos Resultados

Para avaliarmos nossos modelos de machine learning utilizamos divérsas métricas, sendo as mais notáveis:

Precisão, que é calculada a partir da fórmula:

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

Sendo assim, a precisão, é uma métrica que pune quando resultados considerados falsos positivos aparecem nas predições do nosso modelo. Sendo falsos positivos, aqueles que no momento de realizar os testes, sabe-se que era um label negativo (por exemplo: status "Negado") mas o modelo previu como positivo (status "Aprovado").

A métrica recall é dada pela a fórmula abaixo, e pune de forma análoga a precisão, mas nesse caso, quando há maior ocorrência de falsos negativos.

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Temos a métrica acurácia, na qual analisa todos os dados que o modelo previu corretamente, em relação a todos os dados do nosso dataset. E por

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

último, temos a métrica utilizada para avaliar nosso modelo de LightGBM, a métrica F1-Score, que é uma média harmônica entre a precisão e o Recall, de modo a evitar que nosso modelo tenha grandes vieses.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Após extensivo treino de modelos, e busca de hiperparâmetros, obtemos um f1-score próximo a 0.7, o que foi rasoável a nos ranquearmos em 9° lugar na competição.

Bibliografia

AGUIRRE, L. A. Introdução à Identificação de Sistemas, Técnicas Lineares e Não lineares Aplicadas a Sistemas Reais. Belo Horizonte, Brasil, EDUFMG. 2004.