Previsão de IDH em Cidades Brasileiras com Modelos de Machine Learning

Antônio Eduardo de Oliveira Carmo
UFERSA
Mossoró - RN, Brasil
antonio.carmo@alunos.ufersa.edu.br

Eduardo Paz Vieira
UFERSA
Mossoró - RN, Brasil
eduardo.vieira@alunos.ufersa.edu.br

Paulo Henrique Almeida de Andrade
UFERSA
Mossoró - RN, Brasil
paulo.andrade@alunos.ufersa.edu.br

I. Introdução

O Índice de Desenvolvimento Humano (IDH) é uma métrica composta que baseia-se em três dimensões fundamentais: saúde, educação e renda [1]. Tendo em vista sua importância para o entendimento de fatores sociais de uma cidade, este artigo busca a criação de um modelo com algoritmos de *Machine Learning* para a previsão do IDH a partir de diferentes dados de vários municípios brasileiros.

II. Pré-Processamento

Com o objetivo de preparar a base de dados *Brazilian Cities* [2] para o treinamento dos modelos, conduziu-se um pré-processamento juntamente com a análise dos dados para identificação das possíveis abordagem de resolução.

A. Tratamento de Valores Nulls

O *dataset*, como mostrado na Figura 1, veio poluído com vários valores inválidos do tipo Null. Devido às diferentes naturezas de cada coluna, optou-se por abordar o tratamento de maneiras diferentes para cada *feature*.

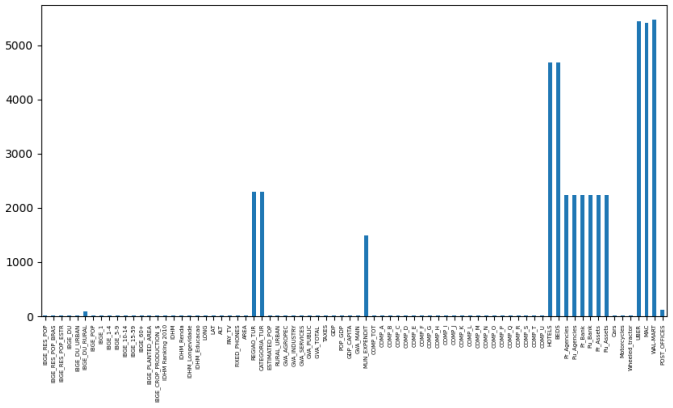

Fig. 1. Valores nulos de cada coluna.

Primeiramente, colunas consideradas essenciais foram escolhidas para que qualquer entrada que não informe estas características fosse removida do dataset. Após aplicar o tratamento nas colunas de coordenadas e área, e em todas as envolvidas com o IBGE, 84 linhas no total foram removidas.

Ao analisar a coluna de despesas dos municípios percebeu-se também uma quantidade alta de valores nulos. Evitando mais remoções, um modelo regressor K-Nearest Neighbors (KNN) de 8 vizinhos foi treinado com um *cross validation* da parte não-nula do dataset, para que este gerasse valores estimados para os municípios com despesas faltantes. O modelo usou dados de geolocalização e tamanho da população dos municípios, alcançando um R2 de 0,76.

Para as colunas da categoria de turismo e região turística, interpretou-se que os valores faltantes significam a ausência de pontos turísticos. Dessa forma, atribuiu-se a categoria "Nenhum" para os nulos e removeu-se a coluna de região turística, visto que esta é apenas uma descrição.

Por fim, para as demais colunas numéricas, percebeu-se que os valores nulos provavelmente significariam que não há quantidades ou valores relevantes da coluna faltante naquele município. Dessa forma, os nulos restantes receberam o valor 0 como substituto.

B. Análise de Variáveis Categóricas

Dando início à análise das variáveis categóricas, foram identificadas colunas que não acrescentam informações relevantes para o modelo, como o nome e estado da cidade. Estas, juntamente com as colunas de Índice de Desenvolvimento Humano (IDH), foram removidas do *dataset*.

Já a variável identificada como *GVA_MAIN* mostra quais atividades econômicas mais contribuem com o *Gross Values Added* (valor acrescentado bruto) do município. Como esta coluna não trás valores, apenas lista as atividades, considerou-se que não agregaria muita relevância para o modelo, sendo também descartada.

Por fim, as colunas de tipologia e categoria de turismo foram submetidas ao algoritmo *get_dummies* da biblioteca Pandas para transformá-las em variáveis numéricas.

C. Redução de Dimensionalidade

Visto que, neste ponto, o *dataset* contém 82 colunas, conduziu-se uma redução de dimensionalidade para retirar variáveis redundantes ou com baixa relevância para a previsão do IDH.

A partir deste ponto, duas variações do dataset surgiram. A primeira, segue sem um filtro de *Principal Component Analysis* (PCA), já a outra, realizou o PCA para três subconjuntos de variáveis relacionadas do dataset.

Na variação com PCA, a contagem individual de cada tipo de indústria foi unificada em um eixo. O mesmo aconteceu com as duas variáveis submetidas ao *get_dummies*: tipologia e categoria de turismo. Esta análise condensou 33 colunas em apenas 3.

Visto que a quantidade de colunas continuou alta, realizou-se uma análise das correlações entre cada variável. Aquelas com correlação entre si superior a 0,95 foram consideradas redundantes e apenas uma representante foi mantida para cada caso. De forma semelhante, as com correlação inferior a 0,05 com a coluna alvo IDH foram consideradas pouco relevantes para o modelo, sendo também descartadas.

Dessa forma, a variação sem PCA finalizou com 35 colunas, enquanto o *dataset* com PCA, foi reduzido para 24 variáveis.

D. Normalização e Outliers

Sabendo que alguns modelos, como KNN, são altamente sensíveis às variações de escala das variáveis, ralizou-se uma normalização dos dados com o método *Standart Scaler* da biblioteca *sklearn*.

Para a análise de *outliers* observou-se que o dataset naturalmente trará um grande número de *outliers* que podem ser importantes para a criação do modelo, visto que estes não são erros, apenas grandes centros urbanos ou municípios muito pequenos.

Dessa forma, outras duas variações do dataset foram geradas. A primeira submeteu as colunas *IBGE_RES_POP*, *GVA_INDUSTRY*, *Pr_Bank*, *Pu_Bank*, *Wheeled_tractor* e *HOTELS* à remoção dos *outliers*. Já a outra variação, manteve apenas a normalização sem remoção de qualquer linha.

Na Tabela I pode-se perceber as características finais de cada *dataset* em termo de linhas e colunas.

TABLE I
DATASETS FINAIS

Dataset	Descrição	N.° Linhas	N.° Colunas
brazil_cities	original	5573	81
br_cities_00	com *outliers*, sem PCA	5489	35
br_cities_01	com *outliers*, com PCA	5489	24
br_cities_10	sem *outliers*, sem PCA	4305	35
br_cities_11	sem *outliers*, com PCA	4305	24

III. Treinamento dos Modelos

Para a construção do modelo, os algoritmos comparados foram: árvore de decisão, floresta aleatória, floresta extremamente aleatória e KNN. Todos as implementações foram importadas da biblioteca *sklearn*.

Cada algoritmo passou pela construção usando ajustes de hiper-parâmetros com a função *grid_search* da *sklearn*. Já para a separação dos dados de treinamento e validação, optou-se pela validação cruzada, visto que esta trás maior representatividade de cada parte do *dataset* e reduz o risco de *overfitting*.

Para os algoritmos com árvores de decisão, os hiper-parâmetros variados foram: profundidade máxima, *split* mínimo das amostras e o mínimo de amostras nas folhas. Além disso, nas florestas variou-se também o número de árvores.

Já durante o treinamento dos modelos KNN, variou-se o número de vizinhos, distribuição dos pesos e o tipo de algoritmo para realizar o cálculo da distância entre os pontos.

IV. Resultados e Conclusão

Após todas as iterações e combinações dos modelos com cada *dataset*, os resultados dos melhores modelos, mostrados na Figura 2, indicam uma tendência de maior R^2 em modelos com algoritmos de florestas e para aqueles treinados com outliers e PCA (*dataset* br_cities_01).

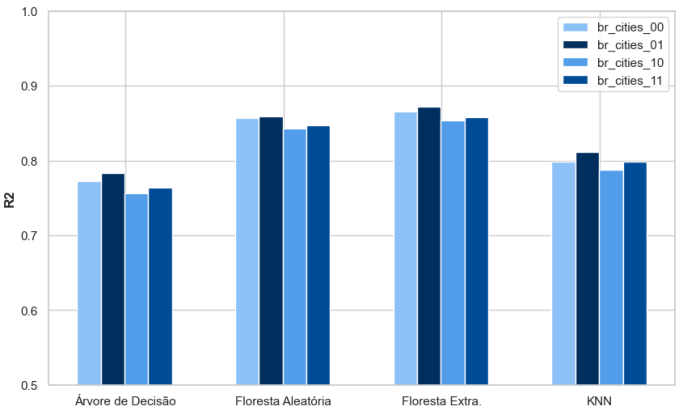

Fig. 2. Valores nulos de cada coluna.

Para todos os modelos de floresta houve vantagem daqueles que não limitaram a profundidade das árvores, o que poderia provocar um *overfitting* se não houvesse a aleatoriedade das amostras e dos atributos. Além disso, as melhores florestas apresentavam o número de árvores igual a 50, visto que acima disso, o ganho era insignificante e não compensava o aumento no tempo de resposta.

Já para o KNN, o valor de vizinhos igual a 9, juntamente com o algoritmo de *ball-tree* e o peso distribuído de acordo com a distância dos pontos trouxeram os melhores resultados entre as variações deste algoritmo.

Com isso, conclui-se o trabalho com a escolha do modelo de floresta extremamente aleatória, treinado usando o br_cities_01, resultando em um R^2 igual a $0,872$.

References

[1] United Nations Development Programme. “O que é o IDH” undp.org. https://www.undp.org/pt/brazil/o-que-e-o-idh (acessado em Set. 21, 2024).

[2] Cristiana Parada. “Brazilian Cities on kaggle.com.” kaggle.com. https://www.kaggle.com/datasets/crisparada/brazilian-cities (acessado em Set. 19, 2024)