Análise da Qualidade de Fontes Hídricas

Luis G. S. Gonzaga - 233612; Gustavo L. Silva - 183444;
Gabriela A. Resende - 172580; Marina L. Santos - 183955

14 de dezembro de 2021

Resumo

Neste trabalho nos estudamos uma base de dados com informações sobre a qualidade da água de diferentes regiões da Europa. Estes dados estão relacionados a informações socioeconômicas regionais as amostras registradas. Com isso foram desenvolvidos modelos preditivos utilizando diferentes abordagens, como modelos utilizando *ensemble* e técnicas de *transfer learning*. Assim, obtemos uma precisão de 80% das amostras no conjunto de teste.

1 Introdução

A água é imprescindível a vida humana. Entretanto, o acesso a água própria ao consumo é um privilégio para muitas pessoas. Sabemos que, na última década, cerca de 3.5 milhões de pessoas morreram devido à falta de acesso a água com sanitização adequada [1]. Essas pessoas provavelmente fazem parte da estatística de que ao menos 785 milhões de pessoas não possuem serviços básicos de fornecimento de água potável [2].

O monitoramento de fontes hídricas é fundamental para o combate as fatalidades causadas pelo consumo de água não-potável. Esta análise deve ser feita constantemente e requer alguns recursos constantes, como acessibilidade dos locais e transporte ágil para os laboratórios [3].

Nosso objetivo foi criar modelos capazes de detectar recursos hídricos impróprios para uso humano baseado em fatores regionais das amostras. Para isso, utilizamos a base de dados disponível no *Kaggle*, a *Water Quality Dataset* [4]. Esta base possui dados socioeconômicos das amostras analisadas de diferentes corpos d'água. Para analisar os dados e montar os modelos preditivos, foram utilizadas diferentes técnicas em aprendizagem de máquina, como redes convolucionais profundas, *transfer learning*, algoritmos de ensemble e detecção de atributos com *random forests*.

2 Trabalhos Relacionados

Métodos com auxílio de computadores são benéficos ao monitoramento de recursos hídricos poluídos. Na literatura algumas técnicas são empregadas com frequência na predição de qualidade destas fontes. Abaurrea et al. [5] usaram regressão para desenvolver métodos de simulação para qualidade da água. Abobakr et al. [6] usaram máquina de vetores de suporte. Existem trabalhos [7] em que pesquisadores encontraram sucesso com o uso de redes neurais no problema de classificar a qualidade de amostras.

Na base de dados relacionada *Water Quality: Analysis (Plotly) and Modelling*, presente no *Kaggle*, diferentes considerações foram obtidas pelo autor [8]. Modelos que se baseavam em métodos de ensemble tiveram os melhores resultados em suas análises. Adicionalmente, modelos com uso de *Random Forests* e *XGBoost* foram os que obtiveram melhor otimização no treinamento.

O autor da base *Water Quality Dataset* trabalhou os dados nos problemas de regressão e classificação [9]. Para o problema de classificação foram obtidos bons resultados usando métodos similares aos usados pelo autor da base *Water Quality: Analysis (Plotly) and Modelling*.

3 Métodos

Nesta seção será apresentada a metodologia de desenvolvimento do projeto. Primeiro, foi feito um tratamento da base de dados para torna-la padronizada para o uso dos algoritmos selecionados. Na criação dos modelos foi tomada uma abordagem em largura, utilizando muito das técnicas estudadas em sala, como redução de dimensionalidade, redes convolucionais, *transfer learning* e etc. Adicionalmente, um breve estudo foi feito em relação a abordagem do autor original da base em relação a divisão das classes nos dados.

3.1 Dados

Primeiro, a base de dados não possuí separação em base de treine e base de testes. Para definir os elementos que fariam parte do conjunto de teste durante o decorrer do projeto, a base foi dividida em treino e teste antes de iniciar a análise dos dados. Isso foi feito com o objetivo de que o conjunto de testes permanecesse resguardado contra vieses que pudessem surgir.

A coluna objetivo para a predição da base, *resultMeanValue*, é composta de valores contínuos de parâmetros de qualidade. Como o objetivo deste trabalho foi em predizer quais fontes eram próprias a consumo humano estes valores foram divididos em duas classes, amostras sujas e limpas. O critério para separação foi

estabelecido com base no critério original do autor, amostras com valores superiores a 500 foram classificadas como sujas, as demais como limpas. Adicionalmente, testes foram feitos para o caso de uma separação em multiclasses, neste caso foi criado um novo intervalo para uma nova classe entre 500 e 10 que possuem impurezas, mas podem ser consumidas. Entretanto, não foram observadas vantagens em se realizar esta abordagem.

Como a base foi obtida pela junção de diferentes bases de dados relacionadas amostras da mesma localização, alguns exemplos tinham dados faltantes. Como o número de valores faltantes são menores que 200 instâncias (1% do total), foi optado por se fazer um preenchimento dessas colunas utilizando a médias dos valores do treino.

Alguns dos atributos presentes na base possuíam maior parte de seus valores como nulos, então, testes foram feitos a fim de determinar se estas colunas poderiam não influenciar o resultado final. Com isso, foi possível perceber que a maior parte destas colunas influenciavam o resultado, como pode ser visto na Seção 4.

Finalmente, os dados foram normalizados utilizando o *standard scaler*. E uma análise das colinearidades, correlação e *outliers* foi realizada a fim de remover dos dados na base. Sem que demais alterações precisassem ser realizadas.

3.2 Primeiros resultados

Para definir um *baseline* como referência para o problema foi utilizado um código criado para a base *Water Quality: Analysis (Plotly) and Modelling* [8]. Inicialmente, nenhuma coluna foi excluida para os resultados iniciais. Foram criados diversos modelos baseados em diferentes algoritmos usando estratégias variadas. Dentre eles, os com melhores resultados foram *Gradient Boosting* "XGB", *Adaboost* "Ada" e o *Multilayer Perceptron* "MLP". Nesses primeiros testes os algoritmos foram usados apenas com seus parâmetros padrão. A Tabela 1 apresenta os principais resultados encontrados inicialmente.

Modelo	**Precisão**	**Acurácia**
MLP	0.8191	0.9893
ADA	0.8142	0.9896
XGB	0.8113	0.99

Tabela 1: Resultados obtidos nos modelos mais significativos na classificação e regressão

Em média os melhores modelos apresentaram uma precisão de 80% e acurácia de 98%. Este valor de acurácia já era esperado, pois 97% dos elementos representam amostras limpas. A melhor métrica para avaliar a eficácia do modelo foi

definida como a precisão dos modelos. Isso foi adotado por se tratar de um problema sensível a falsos positivos, onde amostras sujas podem ser consumidas como se contivessem água potável.

3.3 Seleção de atributos

Uma das tarefas realizadas no projeto foi a de seleção de atributos. Inicialmente, testes foram feitos para determinar a importância de quatro colunas no processo de classificação. Estas colunas possuíam em sua maior parte valores nulos, sendo elas *composition_rubber_leather_percent composition_wood_percent composition_yard_gar-den_green_waste_percent* e *literacyRate_2010_2018*.

Em seguida uma seleção de atributos relevantes foi realizada com o uso de modelos de florestas aleatórias. Algoritmos de classificação baseados em arvores de decisão criam uma seleção de atributos relevantes na separação dos elementos da base em subconjuntos com menor impureza. Assim, quanto maior for a redução de impureza nos subconjuntos pela seleção de um atributo, mais relevante é este atributo no processo de classificação [10].

Outro processo aplicado foi o da extração de atributos da base. Neste processo, foi aplicado um algoritmo de redução de dimensionalidade PCA (*Principal Component Analysis*). Este algoritmo utiliza resultados presentes na área de geometria analítica para calcular um hiperplano, com dimensionalidade inferior ao da base de dados, mais próximo possível dos dados presentes na base. Os elementos da base são projetados neste plano, reduzindo assim o número de atributos presentes na base de dados. Este processo é realizado a fim de se obter pouca perda de informações em relação a base original [10].

3.4 Redes neurais profundas

Aprendizagem profunda é um processo que permite o aprendizado de representação de informação com múltiplos níveis de abstração pelo uso de modelos com múltiplas camadas de processamento [11]. Assim, redes neurais profundas são redes neurais artificiais que fazem uso das ideias presentes em aprendizagem profunda. Algumas das arquiteturas mais famosas em redes neurais profundas são as redes neurais convolucionais, que fazem uso de camadas de filtro de convolução para realizar uma melhor extração de informação dos dados.

Neste projeto foi feito uma implementação adaptada de uma arquitetura conhecida como *LeNet* [12]. Esta é uma rede relativamente simples com cinco camadas, sendo estas duas camadas de convolução com *pooling* e três camadas *fully conected* de neurônios. Originalmente, a rede recebia como entrada uma matriz 2D, neste projeto as alterações foram feitas para que a rede interpretasse valores 1D como entrada para a base de dados estudada.

Outro processo que pode ser realizado em redes profundas é o de *transfer learning* (do inglês, transferência de aprendizagem). Esse processo surge do processo natural humano de utilizar conhecimento adquirido anteriormente em outras tarefas. Para aplica-lo é necessário utilizar modelos previamente treinados e os transferir, com alguns ajustes, ao problema que se deseja resolver.

Em redes neurais convolucionais isto é feito substituindo a camada de entrada para o padrão do novo problema e as últimas camadas *fully conected* para aprender a saída do novo problema. Normalmente este processo é feito com o uso de imagens e com todas as camadas centrais completamente 'congeladas'. Entretanto, é possível realizar o processo sem que as camadas centrais sejam congeladas a fim de adaptar ainda mais o modelo ao novo problema.

Neste projeto, *transfer learning* foi aplicado utilizando um modelo da arquitetura *ResNet50* pré-treinado com a base de dados *ImageNet* [13]. Para se aplicar este método na base deste projeto algumas alterações foram requeridas. Por limitações do *software* utilizado, os valores de entrada necessariamente precisam satisfazer no mínimo uma imagem com dimensões $32x32$ e três canais. Assim, além de substituir as camadas necessárias em um processo normal de transfer learning, foram adicionadas uma camada *fully conected* antes da entrada da rede com o objetivo de ajustar a entrada dos elementos da base.

4 Experimentos

Para conduzir os experimentos foram criados diversos modelos, todos treinados usando a mesma divisão do conjunto de treino em treino e validação. Cada um dos modelos segue um algoritmo ou arquitetura de rede discutidos na Seção 3 e procura melhorar o *baseline* obtido. Modelos diferentes criados com o mesmo método são diferenciados por diferentes escolhas de otimizadores da descida do gradiente, como SGD com momentum, Adam, Adamax e etc.

Inicialmente, foram testados modelos de redes convolucionais com e sem *transfer learning*. A rede treinada sem o uso de *transfer learning* foi a *LeNet* e com transfer learning foi utilizado um modelo treinado pela *ImageNet* na arquitetura *ResNet50*. Os modelos obtidos pela arquitetura *LeNet* utilizavam taxa de treinamento entre 0.1 e 0.01 e para os modelos *ResNet50* valores iguais a 0.001. Os modelos com *transfer learning* foram treinados com taxas de aprendizagem menores para que o conhecimento prévio não fosse perdido. Os resultados deste experimento podem ser verificados nos resultados da Tabela 2.

Modelo	Otimizador	Precisão	Modelo	Otimizador	Precisão
LeNet	SGD	0.82	ResNet (Frozen)	SGD	0.49
LeNet	Adam	0.87	ResNet (Frozen)	Adam	0.49
LeNet	Adamax	0.88	ResNet (Frozen)	Adamax	0.73
LeNet	Adagrad	0.74	ResNet (Unfrozen)	SGD	0.49
LeNet	Adadelta	0.80	ResNet (Unfrozen)	Adam	0.83
			ResNet (Unfrozen)	Adamax	0.81

Tabela 2: Resultados com redes convolucionais e *transfer learning*

Com isso é possível perceber uma melhora significativa dos resultados pela rede LeNet. As redes que utilizaram *transfer learning* com camadas intermediarias congeladas não apresentaram qualquer melhora. Já para as redes com *transfer learning* e camadas intermediarias descongeladas foi possível perceber um leve ganho de performance na classificação. Adicionalmente, os melhores otimizadores para este problema foram o Adam e Adamax.

Após este processo inicial, foram testados os mesmos algoritmos usados no *baseline* e as arquiteturas convolucionais. Entretanto, para o próximo passo foi desenvolvida uma seleção de atributos utilizando *randon forest* (RF) e o método de redução de dimensionalidade PCA.

Em ambos os processos uma variância de 99% da base foi mantida. No caso da *randon forest* isso diminui a dimensão do base de 28 atributos para 21. Os atributos removidos incluiram os previamente citados na Seção 3 (quatro atributos que possuiam em sua grande maioria valores nulos) e os atributos *gdp procedureAnalysedMedia parameterWaterBodyCategory* e *procedureAnalysedFraction*. Para o algoritmo PCA a base foi reduzida para 18 dimensões, entretanto pela natureza do processo não é possível extrair informação de importância dos atributos. Os resultados deste experimento podem ser verificados nos resultados da Tabela 3.

Modelo	Otimizador	Precisão (RF)	Precisão (PCA)
Adaboost	-	0.89	0.88
XGboost	-	0.82	0.81
MLP	-	0.73	0.77
LeNet	Adamax	0.49	0.78
ResNet (Unf)	Adamax	0.49	0.88

Tabela 3: Resultado com redutores e seletores de dimensionalidade.

Foi possível perceber uma melhora considerada nos modelos com *transfer learning* e PCA e em ambos os casos para os algoritmos *Adaboost* e *XGboost*. Entretanto, para os modelos de redes perceptron de múltiplas camadas e para a

arquitetura LeNet, a redução de dimensionalidade foi prejudicial a performance dos modelos.

um último modelo treinado foi um classificador por votos entre os melhores modelos *Adaboost* e *XGboost* usando como critério de desempate a probabilidade das classes preditas entre ambos. Para isso foi realizado uma sintonização dos hiper parâmetros de cada modelo e um treinamento usando *cross-validation* com 5 divisão no conjunto de treino.

Finalmente, foi testada a performance dos melhores modelos no conjunto de teste separada ao início deste projeto. Os modelos escolhidos para este teste foram o LeNet, utilizando a base de dados completa; ResNet50 com o uso de *transfer learning* e com a base reduzida pelo PCA; o modelo com classificador por votação entre os modelos Adaboost e XGboost utilizando a base com seleção de atributos pelo *randon forest* e redução pelo PCA. Os resultados do experimento final no conjunto de teste podem ser verificados nos resultados da Tabela 4.

Modelo	**Otimizador**	**Precisão**
LeNet	Adamax	0.79
ResNet50 (PCA e Unf)	Adam	0.80
Vot. Classifier (RF)	-	0.49
Vot. Classifier (PCA)	-	0.73

Tabela 4: Performance dos modelos no conjunto de testes.

Podemos perceber por estes resultados que os melhores modelos foram os trainados pelas redes convolucionais. O modelo usando a arquitetura *ResNet50* se provou robusto a *overfiting* por mantar uma performance considerável no conjunto de testes. Adicionalmente, apesar do classificador por votação (*Vot. Classifier*) ter obtido os melhores resultados no conjunto de validação, foi possível perceber um *overfitting* destes modelos com perda considerável de performance no conjunto de testes.

5 Conclusão

Predizer a qualidade da água no conjunto de dados provou ser uma tarefa desafiadora. Os melhores métodos obtidos na obtenção de modelos eficazes foram com o uso de *ensemble*, redes profundas e *transfer learning*. Dentre estes, o modelo que obteve o melhor desempenho no conjunto de testes foi o que utilizava a técnica de *transfer learning* na rede *ResNet50*. Para os demais modelos foi possível observar um grau de *overfitting* no conjunto de treino. Isso mostra como a técnica de *transfer learning* pode ser uma poderosa ferramenta na adaptação de modelos mesmo

em problemas muito distintos.

Futuramente pretende-se utilizar diferentes modelos que podem ser mais otimizadas para o problema estudado. Adicionalmente, é possível estudar técnicas de *data augmentation* para amostras de água suja, a fim de evitar o *overfitting* observado. É interessante adaptar as conclusões encontradas para expandir o modelo para uma escala globalizada. Por fim, os resultados em seleção de atributos importantes poderiam ser utilizados para pesquisar relações entre políticas ambientais e a qualidade da água.

Referências

[1] International decade for action 'Water for Life' 2005-2015. `https://www.who.int/news-room/fact-sheets/detail/drinking-water`. Accessed: 2021-12-06.

[2] Drinking-water. `https://www.who.int/news-room/fact-sheets/detail/drinking-water`. Accessed: 2021-12-06.

[3] Water supply, sanitation and hygiene monitoring. `https://www.who.int/teams/environment-climate-change-and-health/water-sanitation-and-health/monitoring-and-evidence/water-supply-sanitation-and-hygiene-monitoring`. Accessed: 2021-12-06.

[4] OZGURDOGAN. Water quality dataset. `https://www.kaggle.com/ozgurdogan646/water-quality-dataset`, 2021.

[5] Jesús Abaurrea, Jesús Asín, Ana C Cebrián, and Miguel A García-Vera. Trend analysis of water quality series based on regression models with correlated errors. *Journal of Hydrology*, 400(3-4):341–352, 2011.

[6] Abobakr Saeed Abobakr Yahya, Ali Najah Ahmed, Faridah Binti Othman, Rusul Khaleel Ibrahim, Haitham Abdulmohsin Afan, Amr El-Shafie, Chow Ming Fai, Md Shabbir Hossain, Mohammad Ehteram, and Ahmed Elshafie. Water quality prediction model based support vector machine model for ungauged river catchment under dual scenarios. *Water*, 11(6):1231, 2019.

[7] Longqin Xu and Shuangyin Liu. Study of short-term water quality prediction model based on wavelet neural network. *Mathematical and Computer Modelling*, 58(3-4):807–813, 2013.

[8] JAY. Water quality: Analysis (plotly) and modelling. https://www.kaggle.com/jaykumar1607/water-quality-analysis-plotly-and-modelling/notebook, 2021.

[9] OZGURDOGAN. Water quality prediction-classification/regression. https://www.kaggle.com/ozgurdogan646/water-quality-prediction-classification-regression, 2021.

[10] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2017.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[12] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[13] Imagenet. https://image-net.org/. Accessed: 2021-12-06.