Trabalho Prático - Predição de Spam

1st Pedro Henrique Melo Araujo
Centro Tecnológico de Joinville (CTJ.)
Universidade Federal de Santa Catarina (UFSC.)
Joinville, Brasil
pedromeloaraujo1999@gmail.com

I. Introdução

Esse trabalhou visou a implementação dos conceitos aprendidos na disciplina de aprendizado de maquina. Dessa forma, propôs-se a implementação de algoritmos para processamento e classificação do banco de dados *spambase* [1].

II. Análise exploratória da base de dados

Inicialmente, foi analisado a distribuição de classes no banco de dados. Como pode ser visto em 1 as classes mostram-se desbalanceadas.

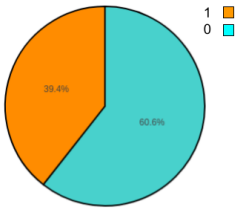

Fig. 1. Distribuição de classes no banco de dados.

A análise dos atributos mostrou uma diferença de escala entre eles. Além disso, também pode ser observado a presença de linhas com valores repetidos que por inspeção visual do banco de dados constatou-se que eram formadas só por campos com zero. Logo, como pré-processamento dos dados foi proposto a normalização dos atributos e eliminação das linhas com campos totalmente nulos, uma vez que elas não representam informações pertinentes para o treino dos modelos.

III. Modelos de classificação

Os modelos escolhidos para classificação do banco de dados foram: SVM, KNN e MLP. A qualidade dos modelos foi baseada na técnica de validação cruzada aninhada com otimização de hiperparâmetros. Ou seja, os dados foram divididos em 5 partes nas quais uma seria para teste e as outras para treinamento. Em seguida, esse processo é realizado repetidamente até cobrir todo o conjunto de dados. Esse mesma estratégia é aplicada de forma aninhada em cada conjunto de treinamento só que visando a otimização do modelo. Logo, os dados são separados também em 5 partes sendo uma de validação, onde os hiperparâmetros serão avaliados, e as outras de treinamento.

A métrica *F-beta Score* foi utilizada para avaliar os modelos e decidir os melhores valores de hiperparâmetros na etapa de validação. Essa métrica se baseia na média harmônica entre a precisão e o *Recall* que estão relacionados as predições corretas da classe positiva o que permite a construção de modelos que tem como objetivo principal não deixar passar *spams*. Além disso, essa métrica é mais adequada para bancos de dados desbalanceados como comentado na seção anterior.

O código realizado nos experimentos pode ser encontrado em https://github.com/PHM-araujo/Machine-Learning/blob/master/Practical_work/Spam_prediction.ipynb

A. SVM

Para o modelo *Support Vector Machine* os hiperparâmetros otimizados foram a penalidade para erros de classificação (C) e a distância de influência usada no *kernel* RBF (Gamma). Os valores testados na otimizados são apresentados em I.

TABLE I
Valores dos hiperparâmetros usados na otimização

C	**1**	**10**	**100**
Gamma	**0.1**	**0.01**	***0.001***

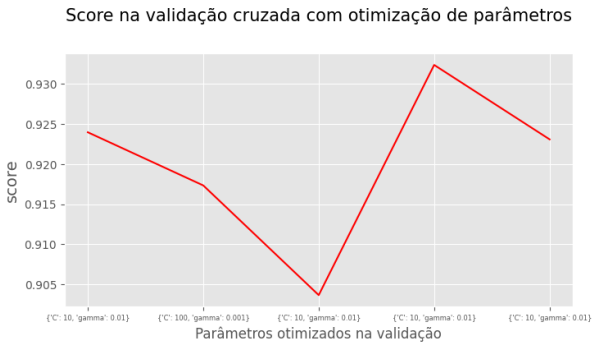

Fig. 2. Valores do F-score com os parâmetros otimizados.

TABLE II
F-score com os valores ótimos

Fold	***C***	***Gamma***	***F-score***
1	**10**	**0.01**	**0.924**
2	**100**	**0.001**	**0.917**
3	**10**	**0.01**	**0.903**
4	**10**	**0.01**	**0.932**
5	**10**	**0.01**	**0.923**

Dessa forma, pode-se observar que os melhores valores de hiperparâmetros para C e Gamma são 10 e 0.01 respectivamente.

B. KNN

Para o algoritmo *K nearest neighbors* os hiperparâmetros a serem otimizados foram o número de vizinhos mais próximos e a função de pesos usada relacionada a como os vizinhos influenciam na predição.

TABLE III
VALORES DOS HIPERPARÂMETROS USADOS NA OTIMIZAÇÃO

Núm. vizinhos	**5**	**10**	**50**
Função de peso	**Uniforme**	**distância**	

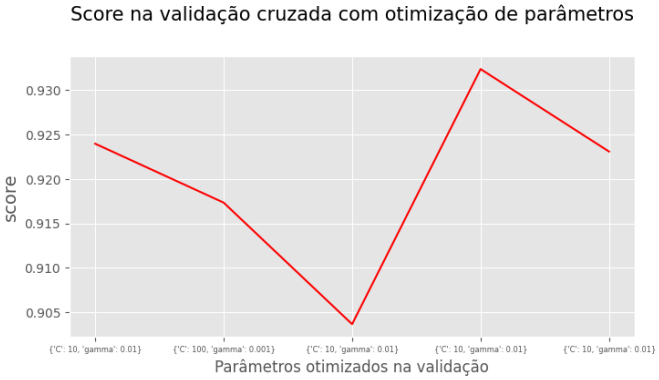

Fig. 3. Valores do F-score com os parâmetros otimizados.

TABLE IV
F-SCORE COM OS VALORES ÓTIMOS

Fold	***Núm. vizinhos***	***Função de peso***	***F-score***
1	**10**	**distância**	**0.905**
2	**10**	**distância**	**0.904**
3	**10**	**distância**	**0.887**
4	**10**	**distância**	**0.914**
5	**10**	**distância**	**0.897**

A partir de IV fica claro que os melhores valores de hiperparâmetros para o número de vizinhos e função de peso foram 10 e distância respectivamente.

C. MLP

No *Multi layer perceptron* os hiperparâmetros otimizados foram o números de neurônios em cada uma das três camadas, a função de ativação e a taxa de aprendizado.

TABLE V
VALORES DOS HIPERPARÂMETROS USADOS NA OTIMIZAÇÃO

Núm. de neurônios	**(150, 100, 50)**	**(100, 100, 100)**
Função de ativação	**relu**	**tangente hiperbólico**
Taxa de aprendizado	**0.001**	**0.01**

Fig. 4. Valores do F-score com os parâmetros otimizados.

TABLE VI
F-SCORE COM OS VALORES ÓTIMOS

Fold	***Função ativação***	***Núm. neurônios***	***Taxa de aprendizado***	***F-score***
1	**relu**	**(150, 100, 50)**	**0.01**	**0.915**
2	**relu**	**(150, 100, 50)**	**0.001**	**0.929**
3	**relu**	**(100, 100, 100)**	**0.001**	**0.901**
4	**relu**	**(100, 100, 100)**	**0.01**	**0.942**
5	**relu**	**(100, 100, 100)**	**0.001**	**0.918**

A partir de VI fica evidente que os melhores valores de hiperparâmetros para o número de neurônios em cada camada escondida, função de ativação e taxa de aprendizado são 100, relu e 0.001 respectivamente.

REFERENCES

[1] Hopkins,Mark, Reeber,Erik, Forman,George & Suermondt,Jaap. (1999). Spambase. UCI Machine Learning Repository. https://doi.org/10.24432/C53G6X.

[2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[3] ONOPRISHVILI, Tornike. SpamBase — Data Exploration & Analysis. 2021. Disponível em: https://medium.com/@tonop15/spambase-data-exploration-analysis-9a3d6d83ee78. Acesso em: 22 maio 2023.

[4] JASKOWIAK, Pablo Andretta. Avaliação de desempenho de modelos de classificação e regressão. Joinville: Ppgese, 2023. 48 slides, color.