

OTIMIZAÇÃO DE HIPERPARÂMETROS E COMPARAÇÃO DE MODELOS

MOISÉS SALES

1. ALGORITMO K-NN

1.1. Selecionando Hiperparâmetros. Foi realizada uma busca sistemática, através do RandomizedSearch, em busca dos melhores hiperparâmetros para o modelo. O seguinte espaço de busca foi criado:

- Número de vizinhos (n-neighbors): conjunto indo $\{1, 100\}$ vizinhos;
- Peso (weights): 'uniform' ou 'distance';
- Métrica para calcular a distância: 'euclidean' e 'manhattan'.

Os melhores hiperparâmetros obtidos, com 10 interações na função RandomizedSearch, foram:

Número de vizinhos: **99**, Métrica para distância: **Manhattan**, Peso: **Distância**

1.2. Avaliando o modelo. Treinando o modelo com os hiperparâmetros que foram selecionados pelo RandomizedSearch, temos os seguintes resultados para as métricas avaliativas do modelo:

Acurácia Valid. Cruzada	Acurácia conj. teste	F1-score	Precisão	Recall
18%	18%	17%	20%	19%

TABELA 1. Avaliação do modelo K-NN.

2. ALGORITMO LVQ

2.1. Selecionando Hiperparâmetros. Foi realizada uma busca sistemática, através do RandomizedSearch, em busca dos melhores hiperparâmetros para o modelo. O seguinte espaço de busca foi criado:

- Número de protótipos por classe: $[1, 2, 3, 4, 5]$;

O melhor hiperparâmetro obtido, com 3 interações na função RandomizedSearch, foram:

Número de protótipos por classe: **2**

2.2. Avaliando o modelo. Treinando o modelo com o hiperparâmetro que foi selecionado pelo RandomizedSearch, temos os seguintes resultados para as métricas avaliativas do modelo:

Acurácia Valid. Cruzada	Acurácia conj. teste	F1-score	Precisão	Recall
17%	18%	16%	16%	19%

TABELA 2. Avaliação do modelo LVQ

3. ALGORITMO SVM

3.1. Selecionando Hiperparâmetros. Foi realizada uma busca sistemática, através do RandomizedSearch, em busca dos melhores hiperparâmetros para o modelo. O seguinte espaço de busca foi criado:

- C: [0.01, 0.1, 1];
- Kernel: 'poly', 'rbf', 'sigmoid';
- Gamma: [0.1, 1];
- Grau: [3, 5]

Os melhores hiperparâmetros obtidos, com 3 interações na função RandomizedSearch, foi:

C: **1**, kernel: **poly**, Gamma: **0.1** , Grau: 5

3.2. Avaliando o modelo. Treinando o modelo com os hiperparâmetros que foram selecionados pelo RandomizedSearch, temos os seguintes resultados para as métricas avaliativas do modelo:

Acurácia Valid. Cruzada	Acurácia conj. teste	F1-score	Precisão	Recall
20%	22%	20%	24%	23%

TABELA 3. Avaliação do modelo SVM.

4. CONCLUSÕES

Podemos observar que todos os 3 algoritmos resultaram em resultados semelhantes e insatisfatórios.

	Acurácia Valid. Cruzada	Acurácia conj. teste	F1-score	Precisão	Recall
K-NN	18%	18%	17%	20%	19%
LVQ	17%	18%	16%	16%	19%
SVM	20%	22%	20%	24%	23%

TABELA 4. Comparativo entre todos os modelos.

O modelo que melhor desempenhou entre os candidatos foi o SVM, que obteve um desempenho melhor em todas as métricas.

Se tratando dessa base de dados, uma melhor escolha das features se mostra necessária. A inclusão, ou exclusão, de certas variáveis pode incrementar a capacidade preditiva do modelo. Visto que, com as variáveis utilizadas, o desempenho não foi agradável. Também pode-se analisar a possibilidade de utilizar diferentes modelos de aprendizado de máquina para lidar com esse banco de dados.