Alunos: Bruno Lopes **5º DSM**
Marcelo Augusto

IMPLEMENTAÇÃO DO ALGORITMO KNN PARA CLASSIFICAÇÃO DE SINTOMAS DE GRIPE

RESUMO

Este relatório apresenta o processo de implementação e avaliação do algoritmo de aprendizado de máquina K-Nearest Neighbors (KNN) para a classificação da doença gripe com base em um conjunto de sintomas. Utilizando uma base de dados numérica obtida na plataforma Kaggle, o modelo foi treinado e avaliado variando-se o hiperparâmetro *K* nos valores 3, 5 e 7. Para cada configuração, foram calculadas as principais métricas de desempenho, incluindo acurácia, precisão, recall e F1-Score, além da geração de matrizes de confusão para análise dos resultados. A comparação entre os valores de *K* permitiu identificar a configuração com melhor desempenho para a tarefa proposta.

1 INTRODUÇÃO

O avanço do aprendizado de máquina tem permitido o desenvolvimento de aplicações em diversas áreas, incluindo a saúde. A capacidade de classificar e auxiliar na previsão de diagnósticos com base em dados sintomáticos pode contribuir para o suporte a profissionais e para a otimização de processos de triagem. Doenças como COVID-19, gripe e resfriado apresentam sintomas semelhantes, o que pode dificultar a identificação precisa no momento inicial.

Nesse contexto, este trabalho tem como objetivo implementar e avaliar o desempenho do algoritmo K-Nearest Neighbors (KNN) na classificação dessas três doenças respiratórias. O estudo analisa a variação do hiperparâmetro *K* (número de vizinhos mais próximos), considerando os valores 3, 5 e 7, com o intuito de identificar a configuração que apresenta melhor desempenho preditivo para a base de dados utilizada.

2 METODOLOGIA

Esta seção descreve os materiais e métodos utilizados para a construção e avaliação do modelo de classificação.

2.1 Base de Dados

A base de dados utilizada neste projeto foi obtida na plataforma Kaggle, sob o título *“COVID, Flu, Cold Symptoms Dataset”* (CONWAY, 2021). O conjunto é composto por registros de pacientes contendo sintomas relacionados a diferentes doenças respiratórias. Após a conversão para formato numérico, os dados passaram a apresentar 20 variáveis correspondentes aos sintomas e uma coluna adicional (“TYPE”), utilizada como variável alvo no processo de classificação.

Cada sintoma é representado por um valor binário, indicando presença (1) ou ausência (0). Para fins deste estudo, adotou-se a seguinte regra de classificação: TYPE = 1 representa casos de gripe, enquanto TYPE = 0 corresponde a outras doenças respiratórias.

2.2 Pré-processamento dos Dados

O tratamento inicial dos dados consistiu na remoção de registros contendo valores ausentes (NaN), com o objetivo de preservar a consistência do conjunto utilizado para treinamento e teste. Em seguida, os dados foram separados em dois subconjuntos: um contendo as variáveis de entrada (X), representadas pelos sintomas, e outro contendo os rótulos (Y), correspondentes ao tipo de doença. Posteriormente, o conjunto foi dividido em 70% para treinamento e 30% para teste do modelo, por meio da função *train_test_split* da biblioteca Scikit-learn.

2.3 Algoritmo K-Nearest Neighbors (KNN)

O algoritmo K-Nearest Neighbors (KNN) é um método de aprendizado supervisionado, não paramétrico e baseado em instância. Sua classificação ocorre por meio da análise dos *K* vizinhos mais próximos de um novo dado no espaço de características, sendo atribuída a classe predominante entre esses vizinhos. A proximidade entre os pontos é determinada por uma medida de distância, comumente a distância euclidiana. A definição do valor de *K* é um fator essencial, uma vez que impacta diretamente o desempenho do modelo e sua capacidade de generalização

2.4 Implementação e Avaliação do Modelo

O algoritmo foi implementado em linguagem Python, com o auxílio da biblioteca Scikit-learn. Um laço de repetição foi utilizado para treinar e testar o modelo com os valores de K definidos (3, 5 e 7). Para cada iteração, as seguintes métricas de avaliação foram calculadas:

- **Acurácia:** Percentual de classificações corretas.
- **Precisão:** Dentre todas as classificações de uma classe, quantas estavam corretas.
- **Recall (Sensibilidade):** Dentre todos os exemplos de uma classe, quantos foram corretamente classificados.

- **F1-Score:** Média harmônica entre precisão e recall.
- **Matriz de Confusão:** Tabela que visualiza o desempenho do algoritmo, mostrando os valores verdadeiros versus os preditos.

3 RESULTADOS E DISCUSSÃO

O modelo foi executado para cada valor de K, e os resultados consolidados são apresentados abaixo.

Tabela 1 – Resultados das métricas para K=3, K=5 e K=7

Métrica	K=3	K=5	K=7
Acurácia	92,22%	92,83%	93,12%
Precisão	0,92	0,93	0,93
Recall	0,92	0,93	0,93
F1 - Score	0,92	0,93	0,93

Tabela 1 - Avaliação do desempenho do modelo KNN utilizando acurácia, precisão, recall e F1-Score com variações de K. Fonte: Autoria própria (2025).

Observa-se um leve aumento na acurácia e nas demais métricas conforme o valor de K aumenta de 3 para 7. Isso sugere que considerar um número maior de vizinhos tornou o modelo ligeiramente mais robusto a ruídos e outliers, resultando em uma classificação mais generalista e precisa para este conjunto de dados. O valor de K=7 apresentou o melhor desempenho geral.

As matrizes de confusão geradas para cada valor de K permitem uma análise visual dos erros e acertos.

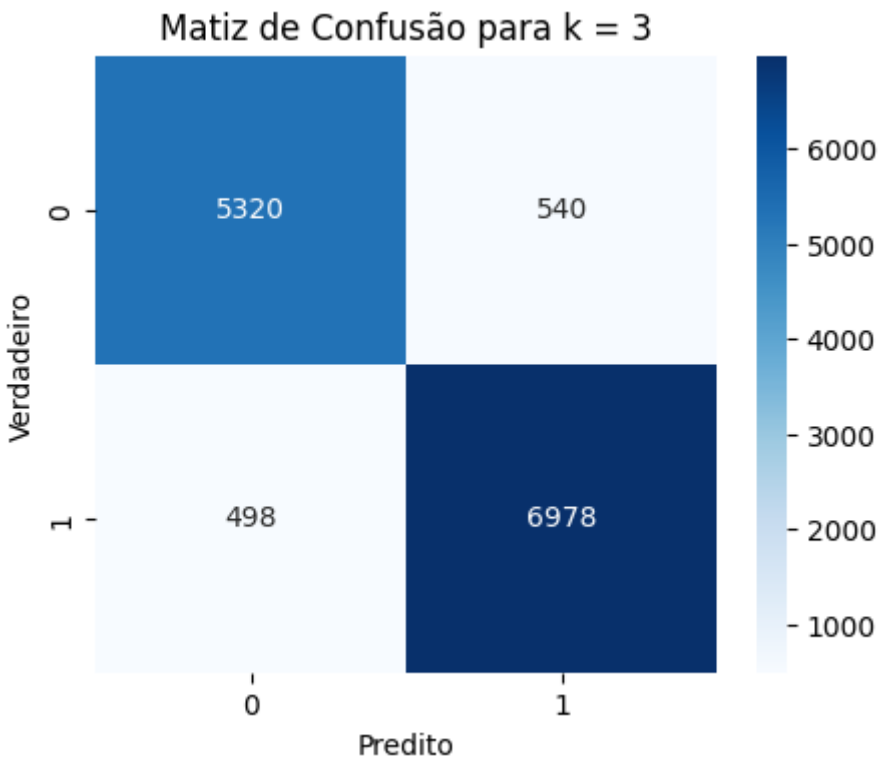

Figura 1 - Matriz de confusão do modelo KNN com K = 3. Fonte: Autoria própria (2025).

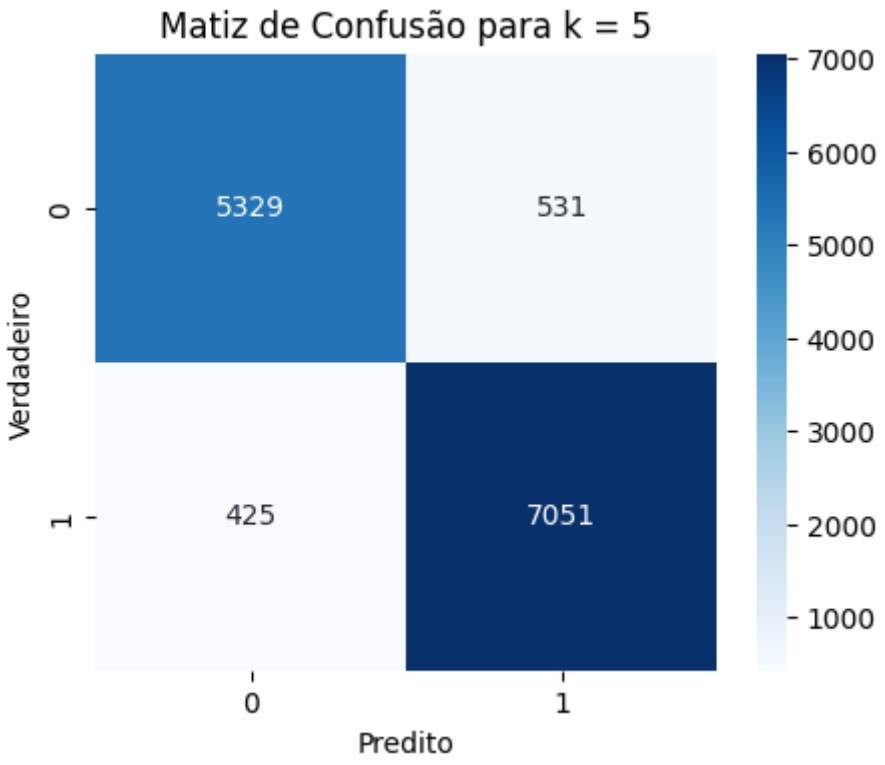

Figura 2 - Matriz de confusão do modelo KNN com K = 5. Fonte: Autoria própria (2025).

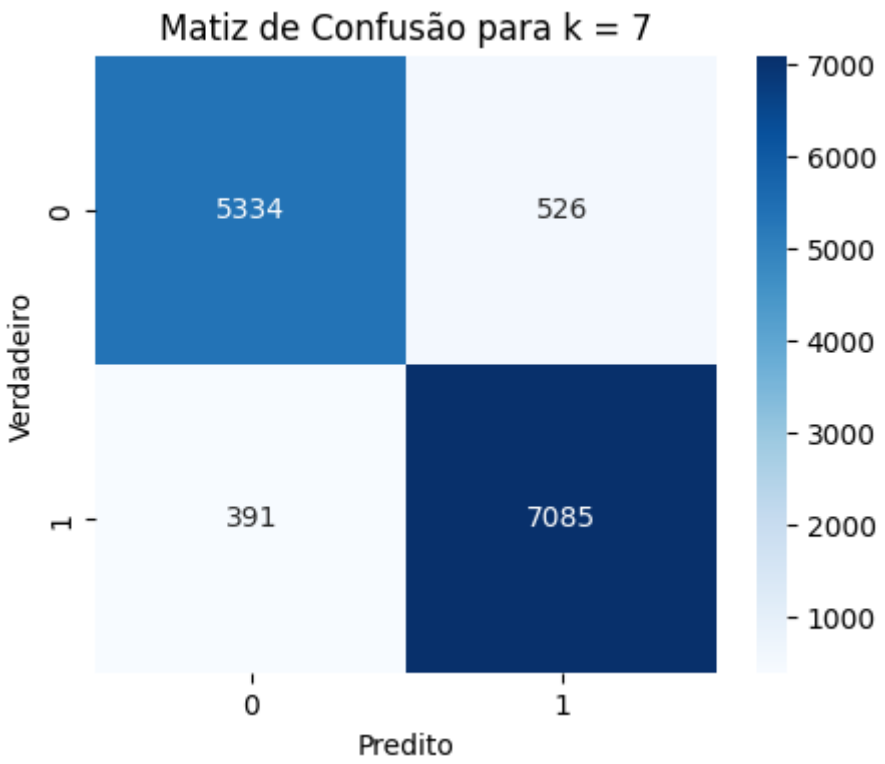

Figura 3 - Matriz de confusão do modelo KNN com K = 7. Fonte: Autoria própria (2025).

4 CONCLUSÃO

A implementação do algoritmo KNN para a classificação de sintomas relacionados à gripe demonstrou-se eficaz, alcançando acurácia superior a 92% em todas as configurações avaliadas. A análise da variação do hiperparâmetro *K* indicou que o valor *K* = *7* apresentou o melhor desempenho, com acurácia de 93,12%.

Os resultados evidenciam que o KNN é uma ferramenta promissora para auxiliar na triagem e identificação de casos de gripe com base nos sintomas apresentados. Como trabalhos futuros, recomenda-se a aplicação de outras técnicas de pré-processamento, o teste com diferentes algoritmos de classificação e a utilização de uma base de dados mais ampla, visando validar e aprimorar o modelo..

5 APÊNDICE A - CÓDIGO FONTE

O código-fonte completo *main.py* utilizado para a implementação, treinamento e avaliação do algoritmo KNN é apresentado a seguir:

Trecho 1: Importação de bibliotecas e leitura dos dados

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score, confusion_matrix, f1_score, precision_score, recall_score
import matplotlib.pyplot as plt
import seaborn as sns

dados = pd.read_csv('large_data_convertido.csv')
dados.dropna(inplace=True)
```

Figura 4 - Código em Python utilizando a biblioteca Scikit-Learn para implementação do algoritmo KNN aplicado à previsão de gripe.Fonte: Autoria própria (2025).

Trecho 2: Preparação e divisão dos dados

```python
x = np.array(dados.iloc[:, :-1])
y = np.array(dados['TYPE'])

#divisão base de treino e teste - 30% teste e 70% treino
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, train_size=0.7)
```

Figura 5 - Código em Python utilizando a biblioteca Scikit-Learn para implementação do algoritmo KNN aplicado à previsão de gripe.Fonte: Autoria própria (2025).

Trecho 3: Treinamento, predição e avaliação do modelo

```python
neighboor = [3, 5, 7]

previsoes = []
for n in neighboor:
    knn = KNeighborsClassifier(n)
    knn.fit(x_train, y_train)
    previsto = knn.predict(x_test)
    previsoes.append(previsto)
    acuracia = accuracy_score(y_test, previsto) * 100
    matriz_confusao = confusion_matrix(y_test, previsto)
    f1 = f1_score(y_test, previsto, average='weighted')
    precisao = precision_score(y_test, previsto, average='weighted')
    recall = recall_score(y_test, previsto, average='weighted')

    print(f"\n--- Resultados para k={n} ---")
    print(f"Acurácia: {acuracia:.2f}%")
    print(f"Precisão: {precisao:.2f}")
    print(f"Recall (Sensibilidade): {recall:.2f}")
    print(f"F1-Score: {f1:.2f}")

    # imagem da matriz de confusão
    plt.figure(figsize=(5, 4))
    sns.heatmap(matriz_confusao, annot=True, fmt='d', cmap='Blues')
    plt.title(f'Matiz de Confusão para k = {n}')
    plt.xlabel('Predito'); plt.ylabel('Verdadeiro')
    plt.savefig(f'matriz_confusao_k_{n}.png', bbox_inches='tight')
    plt.close()
```

Figura 6 - Código em Python utilizando a biblioteca Scikit-Learn para implementação do algoritmo KNN aplicado à previsão de gripe.Fonte: Autoria própria (2025).

6 REFERÊNCIAS

CONWAY, Walter. **COVID, Flu, Cold Symptoms Dataset**. Versão 1. Kaggle, 2021. Disponível em: https://www.kaggle.com/datasets/walterconway/covid-flu-cold-symptoms. Acesso em: 29 nov. 2025.