

# Attention Model-Enhanced k-Nearest Neighbor Algorithms for Wine Quality Classification

Zhilong Wu

*Dept. of Electrical and Computer Engineering  
The Ohio State University  
Columbus, Ohio  
thomaswu@onmail.com*

**Abstract**—This study employs multiple ML models to predict wine taste preferences using physicochemical data from 1,599 wine samples. To optimize predictions and address the limitations of high-dimensional data, we propose enhancing the k-Nearest Neighbor (k-NN) algorithm with an attention-based mechanism. While methods such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) face challenges in dimensionality reduction for k-NN tasks, an attention model helps mitigate these by focusing on the most relevant features. By incorporating the "Attention Is All You Need" mechanism, the k-NN algorithm can improve accuracy and computational efficiency in high-dimensional environments. A sensitivity analysis is also conducted to guide the selection of variables and models, offering deeper explanatory insights. The results showed the applicability of Attention mechanism into k-NN and its practicality in reality.

**Index Terms**—Machine Learning, Feature selection, Classifiers, Wine quality

## I. INTRODUCTION

The wine industry is increasingly investing in technology to enhance both sales and production, with quality evaluation and certification playing a crucial role in this process. Certification safeguards human health by preventing illegal wine adulteration and ensuring market standards are met. Typically, the certification process includes assessing wine quality through both physicochemical and sensory tests. Physicochemical tests rely on laboratory measurements, while sensory tests depend on human expertise. However, the link between these two types of analyses is intricate and not fully understood, making accurate taste-based wine classification difficult.

Advancements in information technology now enable the collection, storage, and analysis of large, complex datasets to support better decision-making and improve outcomes. Machine learning algorithms help extract meaningful insights from unstructured data, with commonly used algorithms including linear and multivariate regression, neural networks, and support vector machines. Each of these algorithms has specific advantages depending on the data type. The key, however, is selecting the right variables and models, as overly simple models may fail to capture important insights, while overly complex models risk overfitting the data.

Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) present limitations when applied in high dimension k-NN tasks. LDA relies on assumptions of linear separability and normally distributed classes with

equal covariances, which may not hold in real-world datasets, resulting in suboptimal class separation. Furthermore, LDA constrains dimensionality reduction to one less than the number of classes. PCA, while maximizing variance, does not prioritize features that enhance classification, and its linear nature may fail to capture complex non-linear relationships, potentially diminishing k-NN's performance.

In this study, we present a practical application where multiple machine learning models are employed to predict wine taste preferences using readily available analytical data from the certification stage. Unlike previous studies, we utilize a large dataset containing 1,599 samples. A sensitivity analysis is conducted to provide explanatory insights, assessing how variations in the input domain impact the results. This analysis helps guide the simultaneous selection of variables and models. To determine the optimal parameters with minimal computational effort, we use conventional k-NN algorithm. Knowing that k-NN algorithms under-performs under high dimensional environments, the main purpose of this article is to assess attention model based k-NN by comparing accuracy with other existing models. Incorporating the "Attention Is All You Need" mechanism into k-NN can enhance its dimension reduction capabilities by focusing on the most important features in the data. By applying self-attention, the model can learn to weigh the relevance of different features and identify which are most crucial for making predictions. This approach reduces the influence of less important dimensions and helps the k-NN algorithm focus on the most relevant aspects of the dataset. Similar to how attention highlights key relationships in natural language processing, in k-NN, it enables the model to perform more efficiently, improving accuracy and reducing computational complexity in high-dimensional spaces.

Our contribution is as follows:

- \*Summarize Model
- \*Result

## II. RELATED WORKS

1) :

## III. METHODOLOGY

This study was completed on Visual Studio Code Jupyter Notebook, Python Version 3.12.4. The operation system is Windows 11 64-bit. The system has an NVIDIA RTX 4080

GPU, running CUDA version 12.4, PyTorch 2.4.1, and an AMD CPU with 32 GB RAM. The work is implemented by programming with python on Jupyter Notebook, proposing the framework, and using the packages provided by Sci-kit Learning.

#### *A. Data Preprocessing*

The dataset, extracted from file 'WineQT.csv', undergoes initial preprocessing. The 'quality' feature is binarized, with wines rated 6 and above flagged '1' as high quality and the rest as '0' as low quality. The dataset is then split into 80 percent training set and 20 percent testing set using stratified sampling. Feature standardization is applied using StandardScaler to normalize the input features.

#### *B. Model Architecture*

The core of this study's approach is an Improved Attention Model, which consists of two main components:

1) *Attention Feature Extractor*: This module learns to weigh the importance of different input features dynamically. It uses a two-layer feedforward network to compute attention weights, which are then applied to the input features.

2) *Classification Network*: A three-layer feedforward neural network processes the attention-weighted features. It incorporates batch normalization and dropout for regularization.

#### *C. Training Procedure*