Assignment1

Field	Detail
Name	조수민 (cho sumin)
Student ID	201540321
Course	Intelligent Information Applications
Date	September 29, 2025

1. Introduction

This report aims to build and evaluate machine learning models for predicting wine quality based on its chemical properties, using the 'Wine Quality Data' dataset.

For this purpose, three distinct classification models were implemented and compared:

1. Logistic Regression
2. k-Nearest Neighbors (k-NN)
3. Naive Bayes

The primary objective is to compare the predictive accuracy of these models and identify the most suitable one for the given dataset.

💡

Code Summary

This project was conducted to build and compare wine quality prediction models using Python libraries such as `pandas`, `scikit-learn`, and `matplotlib`.

1. **Data Preprocessing:**
    - The `Wine_Quality_Data.csv` dataset was loaded using `pandas`, and the target variable, `quality`, was separated from the features.
    - The non-numeric `color` feature was excluded from the model training.
    - The dataset was split into a training set (80%) and a testing set (20%) using the `train_test_split` function.
    - All features were standardized using `StandardScaler`, which was fitted only on the training data to prevent data leakage.
2. **Model Training & Evaluation:**
    - **Logistic Regression:** A `LogisticRegression` model from `scikit-learn` was trained, and its accuracy was measured on the test set. Furthermore, the model's coefficients (`coef_`) were visualized to analyze feature importance.
    - **k-Nearest Neighbors (k-NN):** To find the optimal hyperparameter `k`, accuracies were calculated for `k` values ranging from 1 to 30 and visualized in a plot. The value that yielded the highest performance, `k=9`, was selected for the final model.
    - **Naive Bayes:** A `GaussianNB` model, suitable for continuous numerical data, was trained and its prediction accuracy was evaluated.
    - The performance of each model was primarily assessed using **accuracy**, and **confusion matrices** were visualized for an intuitive understanding of the prediction results.

2. Data Preprocessing

> 💡 데이터 유출(Data Leakage)은 모델이 정답(테스트 데이터)을 미리 훔쳐보는 행위.

To ensure an objective evaluation and enhance model performance, the following data preprocessing steps were performed.

- **Data Split:** The dataset was divided into a training set (80%) and a testing set (20%) to fairly evaluate the generalization performance of the models. This was accomplished using `sklearn.model_selection.train_test_split` with `random_state=42` for reproducibility.
- **Feature Scaling:** Given that the features in the dataset have different units and value ranges, `sklearn.preprocessing.StandardScaler` was used to standardize all features. This step minimizes the influence of scale differences on model training, which is particularly crucial for distance-based algorithms like k-NN. To prevent data leakage, the scaler was fitted only on the training data and then used to transform both the training and testing sets.

3. Modeling

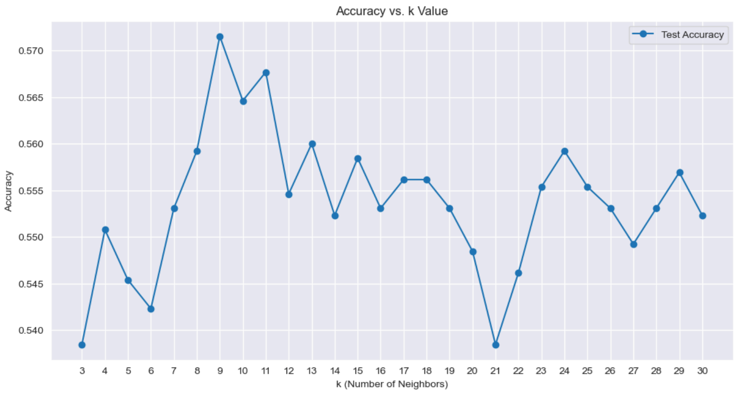
1) Logistic Regression

A standard Logistic Regression model was applied to predict the wine quality class.

```
def multi_class_logistic_regression(input_features):

    # 1단계: 각 클래스(품질 등급)에 대한 점수를 각각 계산한다.
    # (선형 계산은 동일하지만, 이제는 등급별로 점수가 하나씩 나옵니다.)
    score_for_quality_5 = calculate_linear_score_for_5(input_features)
    score_for_quality_6 = calculate_linear_score_for_6(input_features)
    score_for_quality_7 = calculate_linear_score_for_7(input_features)
    # ... 모든 등급에 대해 반복 ...

    all_scores = [score_for_quality_5, score_for_quality_6, score_for_qualit
y_7, ...]


    # 2단계: 이 모든 점수를 '소프트맥스 함수'에 통과시켜 각 등급에 속할 확률들로 바꾼다.
    # (결과: 모든 확률의 합은 1이 됩니다. 예: [0.25, 0.65, 0.10, ...])
    probabilities = softmax(all_scores)


    # 3단계: 계산된 확률들 중 가장 높은 값을 가진 등급을 최종 결과로 선택한다.
    # (예: 6등급일 확률이 65%로 가장 높으므로 '6등급'으로 예측)
    prediction = find_class_with_highest_probability(probabilities)

    return prediction
```

2) k-nn

To find the optimal hyperparameter `k`, an experiment was conducted by varying `k` from 1 to 30 and measuring the corresponding accuracy. The results showed that the model achieved its highest performance when `k=9`. This value was subsequently adopted for the final k-NN model.



3) Naive Bayes

As all features in the dataset are continuous numerical values, the `GaussianNB` (Gaussian Naive Bayes) model, which is well-suited for such data distributions, was used for the classification task.

4. Results & Analysis

1) Model Performance Comparison:

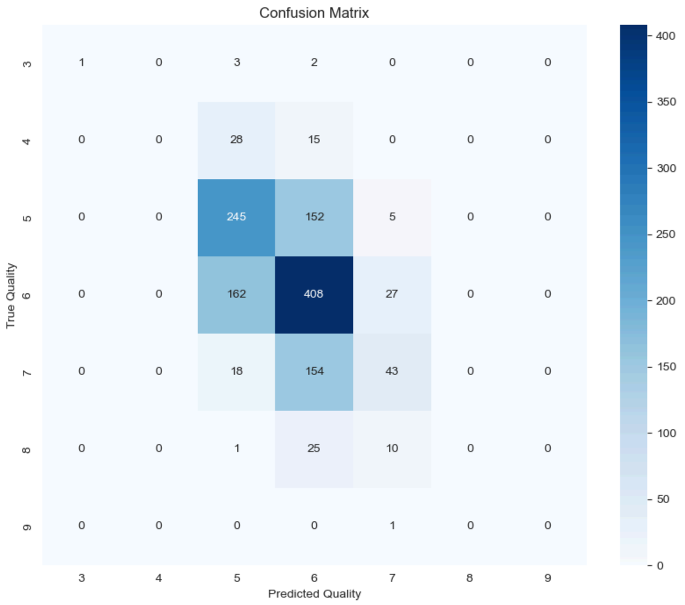
The prediction accuracy of each model on the test data is summarized in the table below. The k-NN model, after hyperparameter tuning, achieved the highest accuracy of approximately 57%.

Model	Accuracy
Logistic Regression	0.5361
K-NN (K=9)	0.5715
Naive Bayes	0.4653

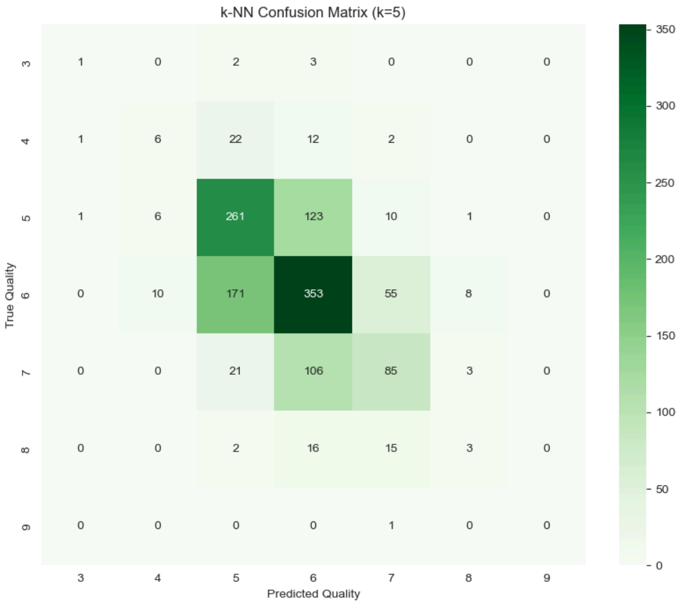
2) Confusion Matrix Analysis:

The confusion matrices for all three models revealed a common trend: they performed relatively well in classifying wines of average quality (grades 5 and 6), which constitute the majority of the data. However, they struggled to accurately predict wines at the extreme ends of the quality spectrum, where data is sparse.

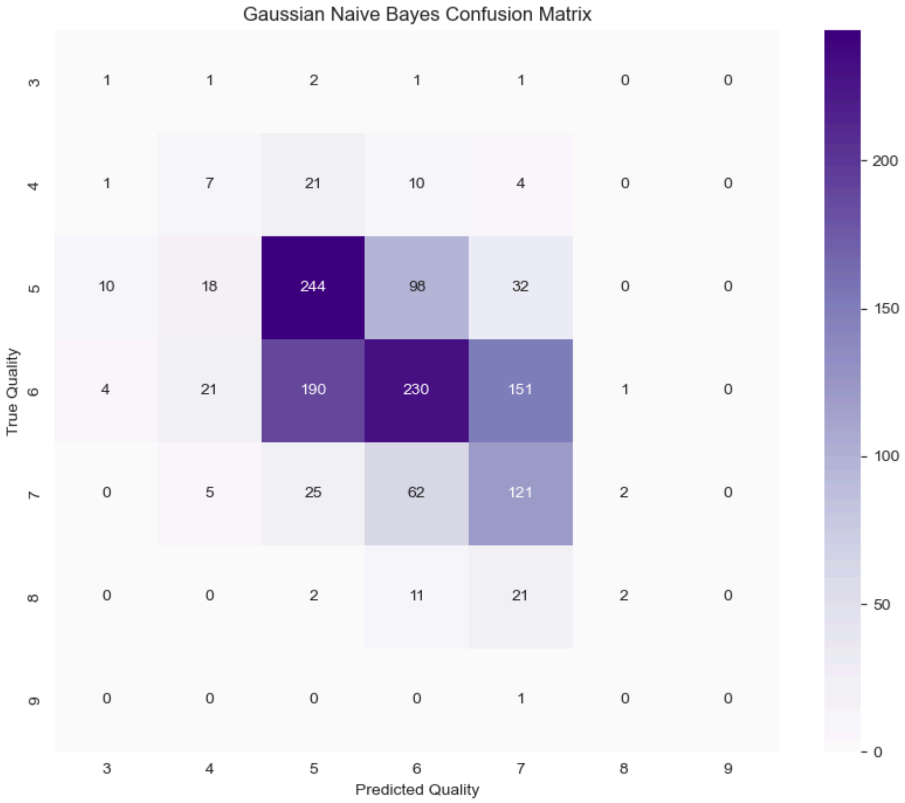
- **Logistic Regression**



- **K-NN**

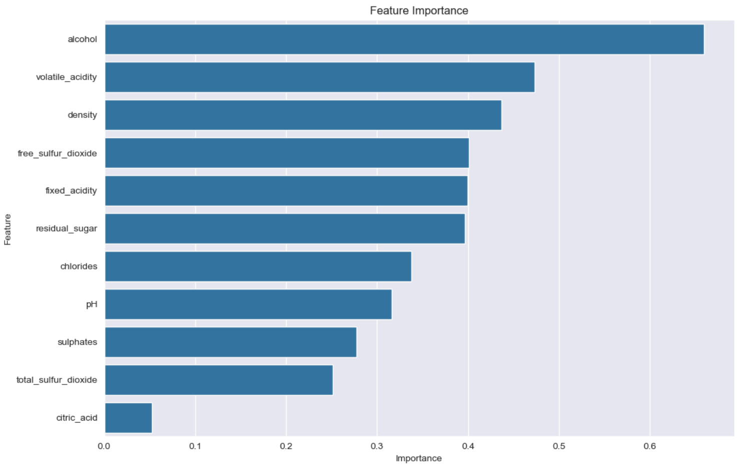


- **Naive-Bayes**



3) Feature Importance Analysis:

By analyzing the learned coefficients (coef_) of the Logistic Regression model, key features influencing wine quality prediction were identified. The analysis indicated that features such as 'alcohol' and 'volatile acidity' were significant predictors. This analysis was limited to Logistic Regression, as k-NN and Naive Bayes models do not provide coefficients in a directly interpretable manner.



5. Conclusion

💡 Best Model : k-NN model(k = 9)

This study compared the performance of Logistic Regression, k-NN, and Naive Bayes models for predicting wine quality. The **k-NN model, with an optimized `k` value of 9, demonstrated the highest accuracy at 57.15%**. This result highlights the importance of proper hyperparameter tuning in improving model performance. Furthermore, the analysis of the Logistic Regression model provided valuable insights into which chemical properties are most influential in determining wine quality.

However, the overall accuracy of the models did not exceed 60%, which may be attributed to the **class imbalance** issue present in the original dataset.

💡 데이터들 대부분이 quality = 5,6,7이다.

Future work could focus on addressing this imbalance through techniques like oversampling. Additionally, incorporating the categorical feature 'color' via onehot encoding could potentially lead to further improvements in prediction accuracy.