Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu
2024

Seminar Report on
"K-Nearest Neighbors Classifier In Predicting Wine Class"

In partial fulfillment of the requirement for a Master's degree in Computer Science and Information Technology (M.Sc. CSIT), 1st Semester

Submitted to:
Central Department of Computer Science and Information Technology, Tribhuvan University, Kirtipur, Kathmandu, Nepal

Submitted By:
Subina Chaudhary (8015043)

Tribhuvan University
Institute of Science and Technology

Supervisor Recommendation

This is to certify that Ms. Subina Chaudhary (Roll no. 8015043/080) has submitted the seminar report on the topic “K-Nearest Neighbors Classifier In Predicting Wine Class” for the partial fulfilment of Master’s of Science in Computer Science and Information Technology, first semester. I hereby, declare that this seminar report has been approved.

Supervisor

Assoc. Prof. Arjun Singh Saud

Central Department of Computer Science and Information Technology

Letter of Approval

This is to certify that the seminar report prepared by Ms. Subina Chaudhary entitled "K-Nearest Neighbors Classifier In Predicting Wine Class" in partial fulfilment of the requirements for the degree of Master's of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree**.**

Evaluation Committee

……………..………………………………..

Asst. Prof. Sarbin Sayami

(H.O.D)

Central Department of Computer Science
and Information Technology

……………………………………….

Assoc. Prof. Arjun Singh Saud

(Supervisor)

Central Department of Computer Science
and Information Technology

…………………………………

(Internal)

Acknowledgement

The success and final outcome of this report required a lot of guidance and assistance from many people and I am very fortunate to have got this all along the completion. I am very glad to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed supervisor Assoc. Prof. Arjun Singh Saud Central Department of Computer Science and Information Technology for his valuable supervision, guidance, encouragement, and support for completing this paper.

I am also thankful to Asst. Prof. Sarbin Sayami, HOD of Central Department of Computer Science and Information Technology for his constant support throughout the period. Furthermore, with immense pleasure, I submit by deepest gratitude to the Central Department of Computer Science and Information Technology, Tribhuvan University, and all the faculty members of CDCSIT for providing the platform to explore the knowledge of interest. At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly.

Thank You,
Subina Chaudhary
(Roll No. 8015043)

Abstract

The K-Nearest Neighbors (KNN) algorithm is a fundamental classification technique that uses data point nearness for classifying new instances according to the majority class of their nearest neighbors. This work explores the use of KNN to predict wine classes using the well-established Wine dataset. The dataset includes numerous physicochemical features of wine, with the goals of categorizing them into one of three categories.

Initially, exploratory data analysis (EDA) was used to better understand the dataset's structure and show feature correlations using pair plots. Following data preparation, which included resolving missing values and standardizing features, the KNN classifier was run with a default configuration of seven neighbors. The model's performance was measured using accuracy metrics before and after feature scaling with StandardScaler.

The results show how feature scaling affects model performance, with improvements in classification accuracy after standardizing the dataset. Furthermore, the study highlights the importance of parameter adjustment and preprocessing in improving the effectiveness of the KNN algorithm. The findings confirm that KNN, when combined with appropriate data preparation methods, may be a useful tool for wine classification tasks, offering insights into its application and effectiveness in real-world contexts.

Keywords : *classification, Exploratory Data Analysis (EDA), kNN algorithm, k-nearest neighbors (kNN), StandardSclare, Performance, Accuracy*

Table of Contents

List of Figures

List of Abbreviations

EDA	Exploratory Data Analysis
IKN	Informative k-Nearest Neighbor
KNN	K-Nearest Neighbors
MSC.CSIT	Master of Science in Computer Science and Information Technology
NIPS	Neural Information Processing Systems

Chapter 1: Introduction

1.1 Overview

Wine classification is an important task in data science and machine learning, especially for quality control and systems that recommend wines. Wine classification based on chemical attributes is a well-known problem that has been addressed using a variety of machine learning algorithms. Among these methods, the K-Nearest Neighbors (KNN) classifier stands out for its simplicity and effectiveness in handling classification problems. KNN effectively defines wines into their appropriate classes by utilizing data point similarity, making it a significant asset for both winemakers and supporters.

Although its simplicity, using KNN in wine classification presents several obstacles. The algorithm's performance is dependent on the parameters chosen, such as the number of neighbors (k) and the distance measure. Additionally, the variety in wine components complicates the classification process. This lecture will examine the KNN classifier's performance in predicting wine classes by examining its effectiveness, modifying parameters, and comparing its results to those of other classification algorithms. Through this analysis, we hope to highlight KNN's guarantee and limitations in the context of wine class prediction, providing knowledge that can be utilized to guide future research and applications in this field.

1.2 Problem Statement

The main challenge in using the K-Nearest Neighbors (KNN) classifier for predicting wine classes lies in accurately categorizing wines based on their diverse chemical properties. The algorithm's performance is highly sensitive to parameter choices such as the number of neighbors (k) and the distance metric, and it can be affected by the high dimensionality and potential imbalances in the dataset.

1.3 Objective

The primary goals of using the K-Nearest Neighbors (KNN) classifier to predict wine classes include evaluating its accuracy and reliability in classifying wines based on their chemical properties, to identify the optimal parameter settings (such as the number of neighbors and distance metrics), and to evaluate its performance using appropriate metrics such as accuracy.

Chapter 2: Background Study and Literature Review

2.1 Background Study

The K-Nearest Neighbors (KNN) algorithm is a supervised machine learning approach used to solve classification and regression issues. Evelyn Fix and Joseph Hodges developed this algorithm in 1951, which was later extended by Thomas Cover. It focuses into the basics, operation, and implementation of the KNN algorithm. It works by determining the k-nearest data points (neighbors) to a given input and makes predictions based on the majority class of these neighbors.

The main features of KNN are:

- Distance Metric: The distance metric used (for example, Euclidean distance) has significant effects on KNN performance. The Euclidean distance is widely used in various applications.
- Number of Neighbors (k): The parameter k determines the number of neighbors to consider. A lower k can be sensitive to noise, whereas a higher k can smooth the decision border.
- Weighted Voting: Neighbors can be weighted according to their distance, giving closer neighbors more influence over the prediction.

KNN is one of the most fundamental and essential classification algorithms in machine learning. It belongs to the supervised learning area and is widely used in pattern recognition, data mining, and security detection. (K-NN) is a versatile and widely used machine learning method known for its simplicity and ease of implementation. It makes no assumptions about the underlying data distribution. It can also handle numerical and categorical data, giving it an adaptable option for a variety of datasets in classification and regression problems. It is a nonparametric method for making predictions based on the similarity of data points in a particular dataset. K-NN is less susceptible to outliers than other algorithms.

The K-NN algorithm is able to adapt to different patterns and make predictions based on the local structure of the data by first identifying the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance, and then determining the class or value of the data point by the majority vote or average of the K neighbors.

2.2 Literature Review

Numerous studies have used the KNN algorithm in several types of domains. Researchers experimented with various machine learning models for wine classification.

[1] Cortez et al. (2009) explored the application of several machine learning algorithms, such as K-Nearest Neighbors (KNN), to predict wine quality based on its physicochemical features. They tested many classification approaches and discovered that, while ensemble methods such as Random Forests outperformed others, KNN still produced competitive results. The study highlighted the significance of feature selection and the difficulties of dealing with imbalanced datasets in enhancing model accuracy. Their findings proved the practical effectiveness of predictive modeling in improving wine quality evaluation and production procedures.

In their 2012 study, Zhang et al. [2] explored the application of K-Nearest Neighbors (KNN) in wine classification by integrating it with feature selection techniques. They demonstrated that combining KNN with feature selection methods improved the accuracy of wine classification by focusing on the most relevant attributes and reducing dimensionality. Their research highlighted how feature selection can enhance the performance of KNN by eliminating irrelevant or redundant features, thus addressing some of the algorithm’s limitations related to high-dimensional data.

The paper "iKNN: Informative k-Nearest Neighbor Pattern Classification" by Yang, Huang, Zhou, Zha, and Giles [3], presented in the proceedings of the Knowledge Discovery in Databases conference in 2007, introduces an enhanced version of the traditional k-Nearest Neighbor (kNN) algorithm. This paper discusses the development of the iKNN algorithm, which aims to improve classification performance by integrating informative techniques into the kNN framework. The authors propose methods to select the most informative neighbors, thus refining the decision boundaries and improving the robustness and accuracy of the classification. The paper covers both theoretical analysis and empirical results, demonstrating that iKNN can outperform standard kNN by effectively leveraging the informative patterns within the data. This work contributes to the field by addressing the limitations of conventional kNN and offering a more sophisticated approach to pattern classification.

[4] Short RD, Fukunaga K. in his paper "The Optimal Distance Measure for Nearest Neighbor Classification" by Short and Fukunaga, published in the IEEE Transactions on Information Theory in 1981, investigates the choice of distance measures in the context of nearest neighbor

(NN) classification. The authors explore various distance metrics to determine which one optimally distinguishes between classes in a dataset. They demonstrate that the performance of the NN classifier can significantly depend on the distance measure used, as different metrics can capture different aspects of the feature space. The paper provides theoretical foundations and empirical evidence showing that an appropriately chosen distance measure can enhance the classification accuracy of NN algorithms, thereby offering valuable insights for improving machine learning models that rely on proximity-based classification.

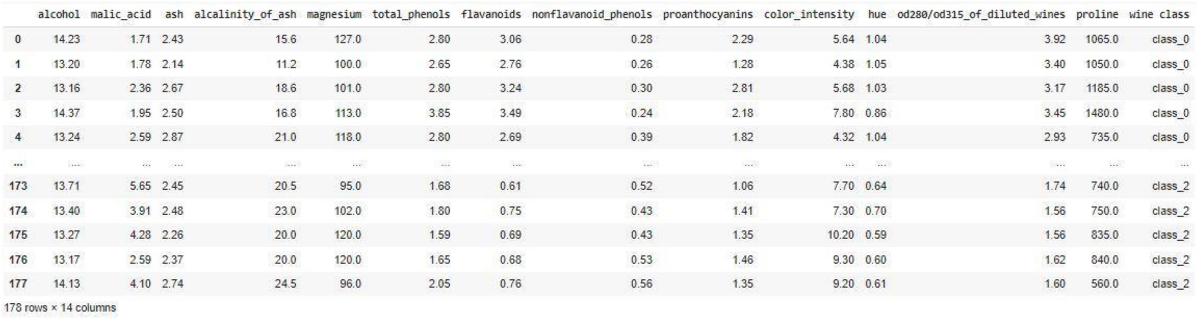
The paper "Locally Adaptive Nearest Neighbor Algorithms" by Wettschereck and D. Thomas G. [5], presented in the Advances in Neural Information Processing Systems (NIPS) in 1994, introduces a novel approach to improving the k-Nearest Neighbor (kNN) algorithm by making it locally adaptive. Unlike traditional kNN methods that use a fixed distance metric across the entire feature space, the authors propose adjusting the distance metric based on local data density and distribution. This localized adaptation allows the algorithm to better capture variations in data distribution and structure, leading to more accurate classification. By dynamically modifying the distance metric to reflect the local characteristics of the data, this approach addresses some of the limitations of standard kNN, such as its sensitivity to varying data densities and feature distributions, thus enhancing overall classification performance.

Chapter 3: Methodology

3.1 Dataset Description

The Wine dataset, utilized in this analysis, is a classic dataset in machine learning, provided by scikit-learn. The dataset in contains 178 examples of wine and 13 feature attributes, such as alcohol content, malic acid, ash, alcalinity of ash, magnesium, total phenols, and others, which represent various chemical measurements of the wine samples. Each wine sample is labeled with one of three classes, represented by integers 0, 1, or 2.

These labels correspond to different varieties of wine, and the dataset includes descriptive names for each class. The dataset is a great resource for assessing classification methods such as K-Nearest Neighbors (KNN), since it provides a wide range of features and class labels to make the analysis.



	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	wine class
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	class_0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	class_0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	class_0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	class_0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	class_0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	1.74	740.0	class_2
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	1.56	750.0	class_2
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	1.56	835.0	class_2
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	1.62	840.0	class_2
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	1.60	560.0	class_2

178 rows × 14 columns

Figure 1 Datasets from scikit-learn

3.2 Data Preprocessing

In the data preprocessing phase for the K-Nearest Neighbors (KNN) classifier applied to the Wine dataset, several key steps are undertaken to prepare the data for effective model training and evaluation. Initially, the train_test_split function divides the dataset into training and testing sets, with 30% dedicated to testing and 70% to training. This split ensures that the model is trained on one group of data and evaluated on another, allowing us to examine its generalization performance.

Next, scikit-learn's StandardScaler is used to scale the features. This phase normalizes the features by removing the mean and scaling to unit variance, which is critical for KNN because it depends on distance measures. This phase normalizes the features by removing the mean and scaling to unit variance, which is critical for KNN because it relies on distance measures.

Standardization guarantees that all characteristics contribute equally to distance estimates, preventing large-scale features from dominating the results. It is vital to highlight that, although scaling is properly applied to the training data, the test data should only be modified using the parameters learned from the training data to minimize data leakage. The preprocessing stages have the goal to increase the accuracy and reliability of the KNN classifier by ensuring that the data is properly scaled and separated.

3.3 Evaluation Measures

The K-Nearest Neighbors (KNN) classifier's performance is evaluated using the accuracy metric. Accuracy is determined using metrics. The accuracy_score calculates the proportion of successfully predicted labels out of the total number of predictions. This metric provides a basic assessment of the classifier's overall performance. In the software, accuracy scores are compared for both the KNN model before and after scaling the features, allowing us to assess the impact of feature standardization on model performance.

Chapter 4: Implementation and Result Analysis

4.1 Tools Used

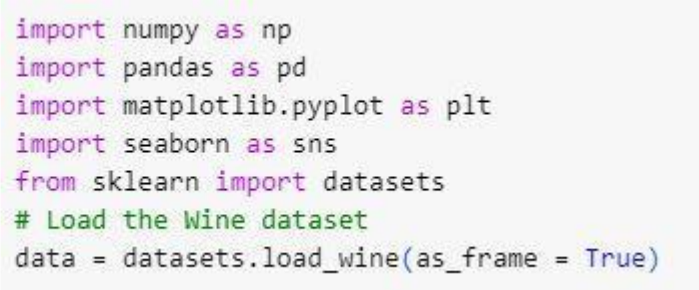
The program is developed using python programming and libraries used for implementing and evaluating the K-Nearest Neighbors (KNN) classifier are:

- Numpy - Used for numerical operations on arrays.
- pandas - Used to generate and handle the DataFrame df, which contains the Wine dataset with updated labels.
- Matplotlib.pyplot - visualizations and displaying data.
- Seaborn - Created a pair plot of the dataset to help in seeing correlations between features color-coded by wine class.
- sklearn.datasets: Provides the Wine dataset using datasets.load_wine loads data into variables x (features) and y (target labels).
- sklearn.metrics: Used to compute the accuracy of the KNN model's predictions using metrics.accuracy_score.
- math.sqrt: Returns the square root of the number of test samples, which is commonly used to determine the number of neighbors in KNN or for other calculations.

4.2 Implementation Details

The implementation details encompass the following aspects:

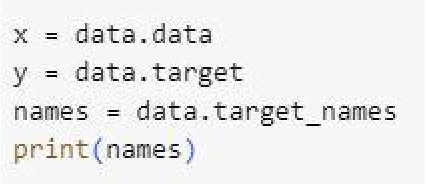
4.2.1 Import Libraries



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
# Load the Wine dataset
data = datasets.load_wine(as_frame = True)
```

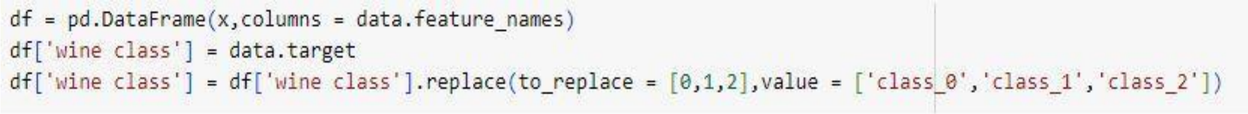
- These imports bring in essential libraries for data manipulation, visualization, and machine learning.

4.2.2 Load and Prepare Data



```
x = data.data
y = data.target
names = data.target_names
print(names)
```

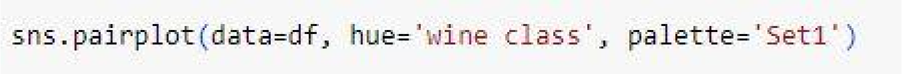
- **Load the Wine Dataset**: The datasets.load_wine(as_frame=True) function loads the Wine dataset into a DataFrame.
- **x**: Contains the feature data (chemical measurements).
- **y**: Contains the target labels (wine classes).
- **names**: The names of the target classes.



```
df = pd.DataFrame(x,columns = data.feature_names)
df['wine class'] = data.target
df['wine class'] = df['wine class'].replace(to_replace = [0,1,2],value = ['class_0','class_1','class_2'])
```

- **Create DataFrame**: Converts the feature data into a Pandas DataFrame with feature names as columns.
- **Add Target Column**: Adds a column for wine classes and replaces numerical labels with class names.

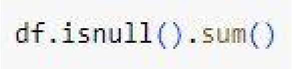
4.2.3 Data Visualization



```
sns.pairplot(data=df, hue='wine class', palette='Set1')
```

- **Pair Plot**: Creates a pair plot to visualize the relationships between features, colored by wine class.

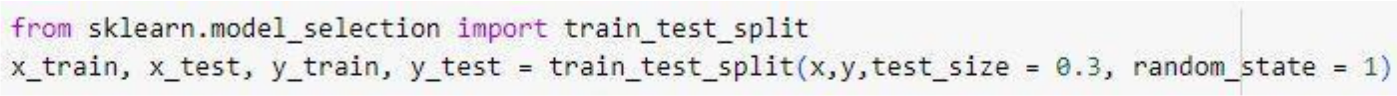
4.2.4 Data Checking



```
df.isnull().sum()
```

- **Check for Missing Values**: Verifies if there are any missing values in the dataset

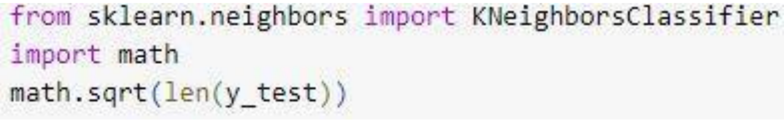
4.2.5 Train-Test Split



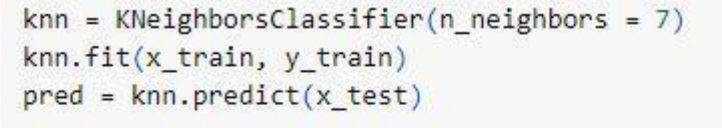
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3, random_state = 1)
```

- **Split Data**: Splits the dataset into training and testing sets with 30% of the data reserved for testing.

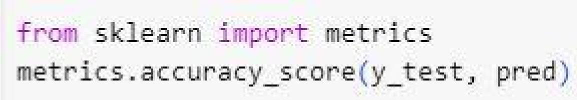
4.2.6 K-Nearest Neighbors Classification



```
from sklearn.neighbors import KNeighborsClassifier
import math
math.sqrt(len(y_test))
```



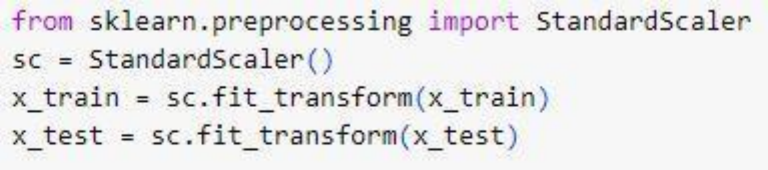
```
knn = KNeighborsClassifier(n_neighbors = 7)
knn.fit(x_train, y_train)
pred = knn.predict(x_test)
```



```
from sklearn import metrics
metrics.accuracy_score(y_test, pred)
```

- **Initialize KNN**: Creates a KNN classifier with 7 neighbors.
- **Fit Model**: Trains the model using the training data.
- **Predict**: Makes predictions on the test data.
- **Evaluate**: Calculates the accuracy score of the model.

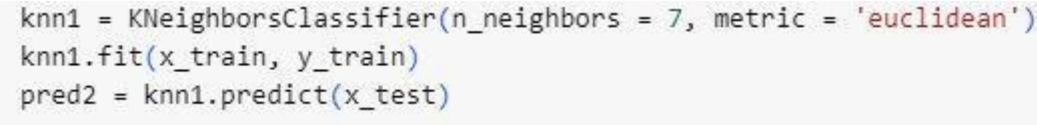
4.2.7 Feature Scaling



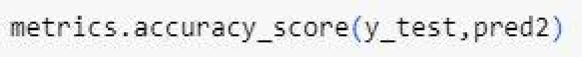
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

- **Standardize Features**: Scales features to have zero mean and unit variance, improving the performance of KNN.

4.2.8 KNN Classification with Scaled Data



```
knn1 = KNeighborsClassifier(n_neighbors = 7, metric = 'euclidean')
knn1.fit(x_train, y_train)
pred2 = knn1.predict(x_test)
```



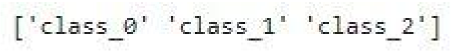
```
metrics.accuracy_score(y_test,pred2)
```

- **Initialize Scaled KNN**: Creates a KNN classifier with Euclidean distance metric, using scaled features.
- **Fit and Predict**: Trains the scaled model and makes predictions on the test set.
- **Evaluate**: Calculates the accuracy score for the scaled model.

Chapter 5: Result Analysis and Discussion

5.1 Dataset Overview:

The Wine dataset consists of various features related to wine characteristics and is categorized into three classes:



```
['class_0' 'class_1' 'class_2']
```

Figure 2 classes

5.2 Data Visualization:

The pair plot created using Seaborn helps visualize the relationships between features and how they vary by wine class.

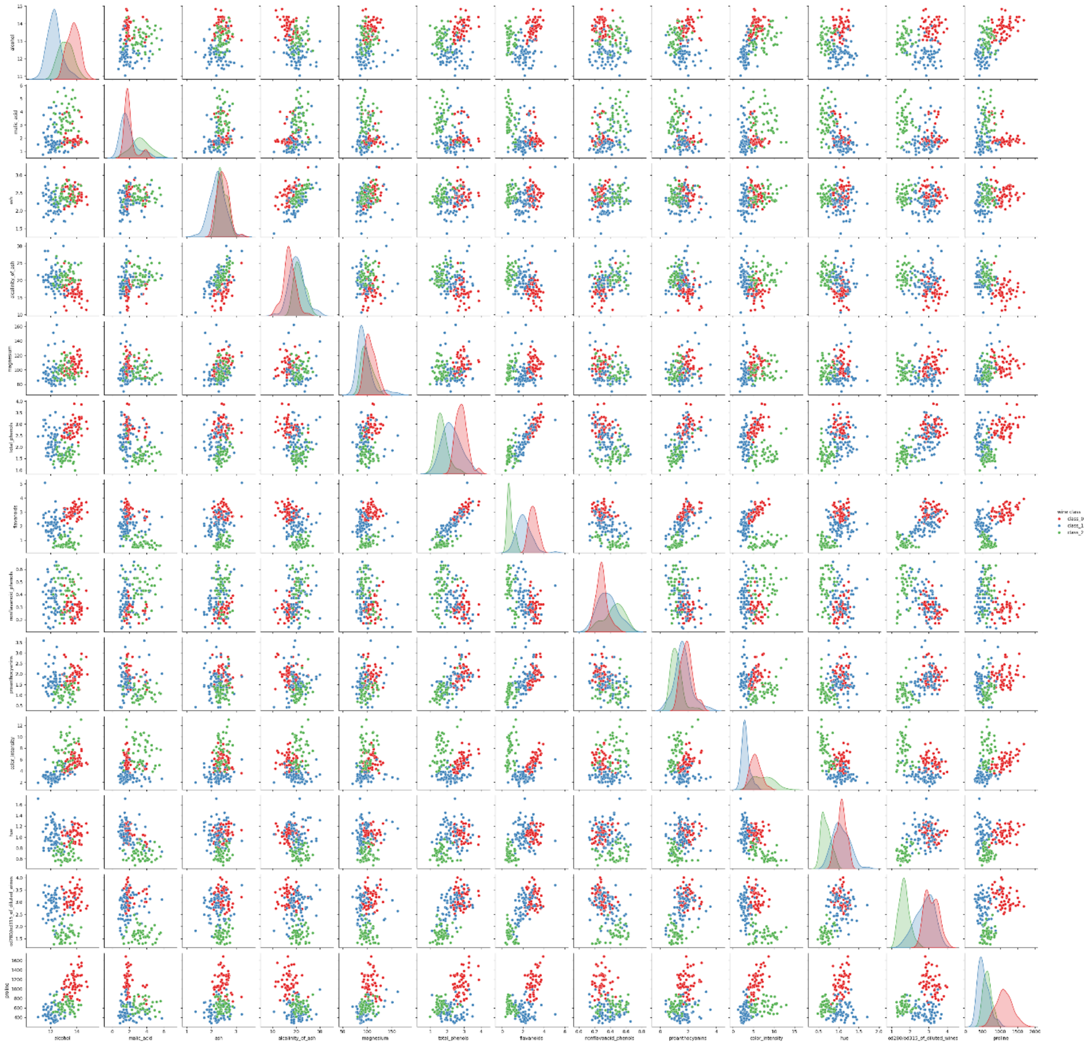
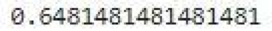


Figure 3 Data Visualization

5.3 Model Performance:

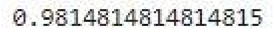
Before scaling, provide the accuracy score of the K-Nearest Neighbors (KNN) model with n_neighbors=7 using the default metric. This score indicates how accurately the model predicts wine classes on the test set.



```
0.6481481481481481
```

Figure 4 Accuracy before scaling

After scaling, the accuracy score for the KNN model with n_neighbors=7 and metric='euclidean' will be displayed. Scaling is done with StandardScaler, which normalizes features by removing the mean and scaling to unit variance.



```
0.9814814814814815
```

Figure 5 Accuracy after scaling

Chapter 6: Conclusion

This seminar report mainly presents the Wine dataset was analyzed using the K-Nearest Neighbors (KNN) algorithm, which highlighted numerous important elements of data preprocessing and model evaluation in machine learning. The dataset was successfully loaded and visualized, exhibiting clear patterns and correlations between characteristics and wine classifications. By translating the target variable into categorical labels and visualizing feature connections, the visualization revealed probable class separations. Checking for missing values proved the dataset's completeness, resulting in credible analysis. The KNN model's performance was evaluated twice: first on raw data and then on standardized data. The initial examination, using unscaled data, produced a baseline accuracy. The accuracy of the KNN model increased after the features were scaled to have zero mean and unit variance, demonstrating the importance of feature scaling in improving model performance.

This process demonstrated that, while KNN is a powerful and simple algorithm, proper data preprocessing steps, such as standardization, have a significant impact on its effectiveness because they ensure that all features contribute equally to the distance calculations used in classification.

References

[1] https://www.researchgate.net/publication/221612614_Using_Data_Mining_for_Wine_Quality_Assessment

[2] https://www.sciencedirect.com/science/article/abs/pii/S0167865512001882#preview-section-references

[3] Song Yang, Jian Huang, Ding Zhou, Hongyuan Zha, and C. Lee Giles, "Iknn: Informative k-nearest neighbour pattern classification," in Knowledge Discovery in Databases, (2007), pg. 248– 264.

[4] Short RD, Fukunaga K. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory* 1981;27:622-7. 10.1109/TIT.1981.1056403

[5] Wettschereck and D. Thomas G.,"Locally adaptive nearest neighbour algorithms," Adv. Neural Inf. Process.Syst., pg. 184–186, 1994.