Exploratory Data Analysis Report

Soudabeh Bolouri

Introduction:

In this exercise we will have a look at the raw data before doing any machine learning. The objective is to familiarize yourself with the data, see if there are any potential problems or obstacles to applying machine learning, and if and what preprocessing may be helpful. We want to explore the distribution of selected features within the "winequality-white.csv" dataset before and after applying a normalization technique using the *Logarithmic Transformation*. We finally showed the distribution of feature values before and after preprocessing.

Dataset Description:

The dataset that we utilized in this exercise is the "Wine Quality" dataset, precisely the "white" wine version. It contains data on different features of white wines, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, PH, sulphates, and alcohol. The dataset includes a totality of 4,898 rows and 12 features. The "quality" of the wine is the target variable, which we desire to predict. Also, we did not find any missing values in the dataset.

Experimental Setup:

We set up the experiment as follows using the Python programming language:

First, we visualized the distributions of the features before preprocessing using histograms. The histograms illustrate the spread and frequency distribution of each feature's values across the dataset.

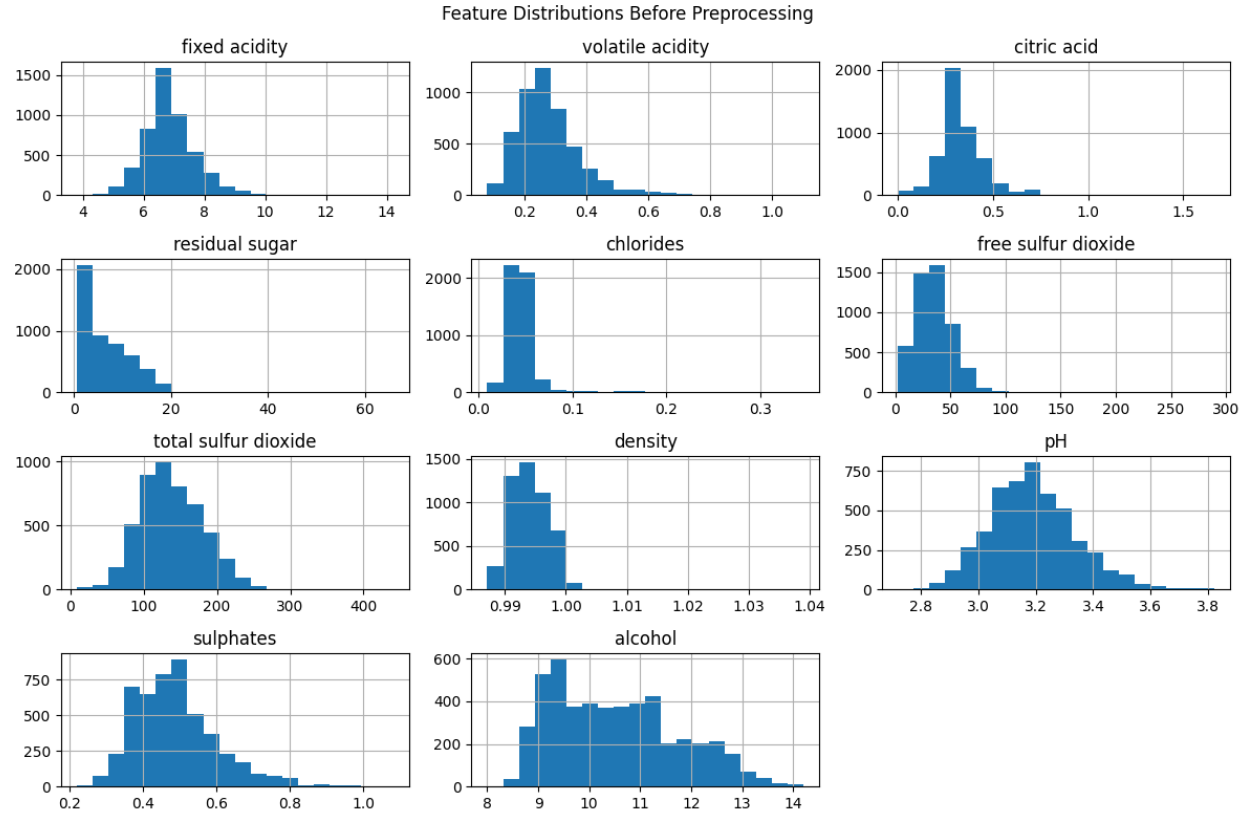
Then we performed a logarithmic transformation on all features except the target variable 'quality'. Combining Pandas' apply function with NumPy's log function produces this transformation. To avoid taking the logarithm of zero or negative values, a lambda function, **lambda x: np.log(x + 1)**, is used; this function computes the natural logarithm of each value in the selected features after adding 1.

After applying the logarithmic transformation to the features, histograms are created again for these transformed features.

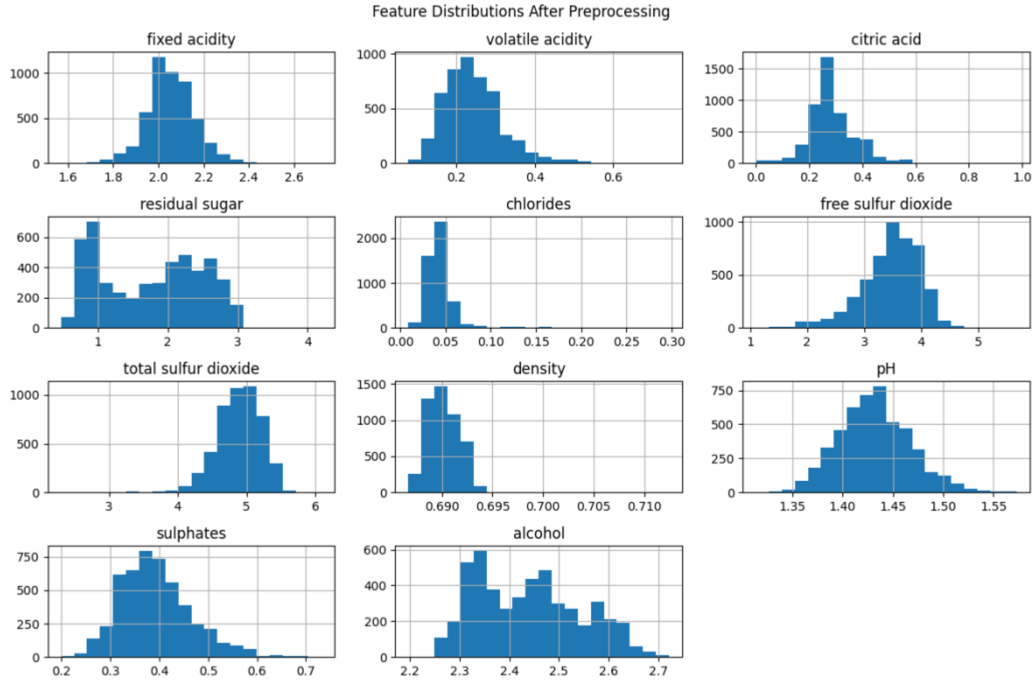
Results:

To demonstrate how logarithmic transformations make modeling easier, we compared skewness in the data. The skewness of a distribution describes the degree of asymmetry of its distribution around the mean. This helps to understand the shape and nature of data distributions. If a distribution is symmetric, values are evenly spread around the mean, and it is balanced; the Skewness measure measures how far it deviates from this symmetry.

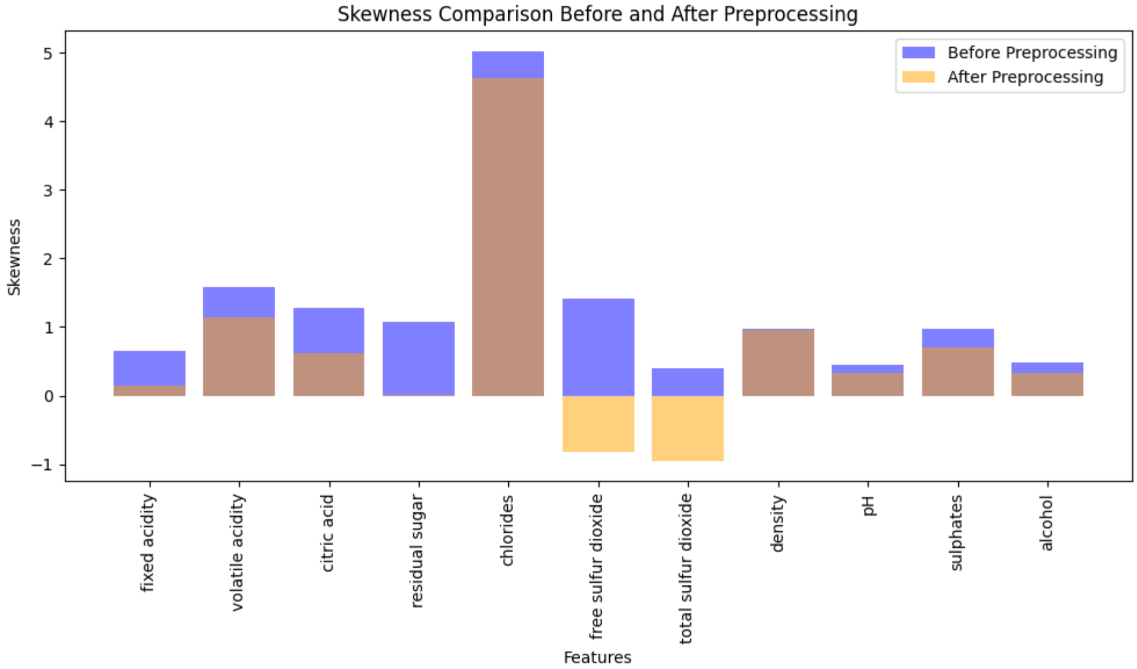
In the first histogram, the original distributions of the features were displayed.



And the distribution of features after applying the logarithmic transformation are as bellow:



The visualization below presents a comprehensive comparison of skewness values for each feature in the dataset before and after applying logarithmic transformations.



Logarithmic transformation leads to a reduction in skewness across all features, suggesting a more symmetric distribution. As a result of this reduction in skewness, the data is aligned closer to a symmetric pattern, which is a favorable characteristic for several machine learning algorithms. When the distribution is more balanced, the algorithm can make more accurate predictions and gain meaningful insight from the data.

References:

[1] https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/#

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[3] https://www.w3schools.com/python/matplotlib_histograms.asp

[4] https://medium.com/@kyawsawhtoon/log-transformation-purpose-and-interpretation-9444b4b049c9