Data Preprocessing Routine for Classification and Clustering

Prathamesh Kulkarni (Student ID: 36004026)
School of Computing and Communications
Lancaster University
Lancaster, England
p.p.kulkarni@lancaster.ac.uk

***Abstract*—The pre-processing of data is an important stage before you apply models to the data during a machine learning project. This paper examines the pre-processing of data, examining the individual steps involved in the pre-processing and observing how each step affects data. It is implemented on two separate data sets, ”ClimateDataBasel.csv” and ”WLA.csv”.**

***Index Terms*—Data Pre-Processing, Missing Values, Outlier Detection, Numerical and Categorical Data, Feature Selection, Normalization/Standardization, Balancing Data, Train-Test Split.**

I. Introduction

Data pre-processing is one of the steps involved in most data analysis, machine learning, or AI projects. We need to process data as there might be several issues with the raw data due to human errors, or some fault in data filling, lack of data, or the data was just not filled in for many privacy concerns and these errors can lead to many issues if not dealt with before applying models or using it for analysis as this may lead to some serious issues, the analysis can be wrong or the machine learning model can behave in an unforeseen way or give wrong results or just not perform well. We can see that by not processing the data properly, it might affect businesses in their analysis and decision making, and hence to create more accurate models and perform a better analysis we need to process the data before creating the model. The data pre-processing can be carried out in multiple steps and each step leads to cleaner and quality data to train models. The aim of data pre-processing is accuracy, completeness, consistency, timeliness, believability, and interpretability of the data. Data pre-processing routines work on finding missing values, detecting outliers, selecting features, normalization and standardization, balancing, and splitting the data. [1] [3]

II. Pre-processing

We will now discuss the individual data pre processing steps that can be used.

A. Null/Missing Values

Missing Values are the observations that are not present in the dataset. Missing values often occur in datasets which can be due to many reasons, the event did not happen, or the data was not available. It could be that the person who entered the data did not know the right value, or did not care if a field was not filled in, or calculation issues can also result in missing values[citation]. Missing data can severely affect the model performances hence should be handled by using various techniques like Mean and Median Imputation where we replace the mean or median where the data is missing, End of Distribution Imputation where a value is chosen from the tail end of the data. This value signifies that the actual data for the record was missing. In Arbitrary Value Imputation, arbitrary values are selected in a way that they do not belong to the dataset. For categorical data techniques like Frequent Category Imputation in which the missing values are replaced with the most frequently occurring values, in Missing Category Imputation the missing value is replaced with arbitrary category. Other techniques are Predictive Value Imputation, Imputation Using K-NN, Reduced-Feature Models, Retrieval of Missing Data Using Rough Sets, Retrieval of Missing DataUsing Soft Sets [4]. In both the datasets we are working on, there are no missing values which is a good thing. We have implemented an if-else condition to deal with removing the missing values in case there are any in the future.

B. Outlier detection

It is important to detect and deal with outliers, which are observations that greatly differ from the rest of the observations in a data set. Outliers can skew or bias data analysis. Therefore, it is critical to detect and deal with outliers appropriately. An important part of data analysis is detecting outliers. It is used in many domains to identify interesting and emerging patterns, trends, and anomalies. Most common causes of outliers on a data set: Data entry errors, Measurement errors, Experimental errors, Intentional, Data processing errors, Sampling errors, Natural.

We have used three methods for our implementation which are the z-score approach, quartile range approach and Distance-Based Approach using Mahalanobis Distance [6] [7] [8]

1) Z-score: A data point’s z-score represents how far it deviates from the mean. 68 percent of data points lie within +/- 1 standard deviation. 95 percent lie within +/- 2 standard deviations. 99.7 percent lie within +/- 3 standard deviations.

Z score and Outliers:

In cases where the z score of a data point exceeds 3, it can be an outlier.

The formula for z-score is:

Z score = (x -mean) / standard deviation

2) Inter Quartile Range Approach: Inter Quartile Range Approach A quartile represents every 25th percent of the total data points. The first quartile represents the 25th percentile of values, the second is the median (50th percentile), and the third and fourth represent 75th and 100th(maximum) percentiles, respectively. The distance between the first and third quartile represents the range of values in the middle 50 percent. We find the interquartile range and multiply it by k, which is typically 1.5. After that, we have the range of values beyond Q3 + K*I. To be considered an outlier, a point must satisfy any one of the following conditions: It must exceed the 75th percentile + 1.5 IQR, or it must be less than the 25th percentile - 1.5 IQR [5]

3) Mahalanobis Distance: Mahalanobis Distance (MD) measures how far a point is from its distribution. Because it uses a matrix of covariances to determine the distance between data points and the centre, it works quite effectively for multivariate data. MD identifies outliers based on the distribution pattern of the data points. The use of covariance in calculating distance in n-dimensional space allows for determining the true threshold border based on the variation.

Mathematically, the Mahalanobis Distance (MD) is calculated as:

MD2 = (x − m)V -1(x − m)

where:

x represents the values of an observation, m represents the mean of each variable, and V represents the variance-covariance matrix

C. Numerical and Categorical data

Since machine learning and deep learning work on numerical data, we must identify categorical variables to convert them into numerical data using various available techniques. The technique involves One Hot Encoding, Label Encoding, Frequency Encoding, Ordinal Encoding, Mean Encoding, among others. We don't need to deal with this issue since our data is numerical. However, it should always be considered as part of the pre-processing.

D. Feature selection

Data with high dimensions make data analysis difficult and machine learning models more complex. To solve this problem, feature selection eliminates the redundant and irrelevant variables. Removing this data improves accuracy and decreases computation time. The selected features are therefore only those that are relevant to the task at hand. Keep in mind that feature selection is not the same as dimensionality reduction. In feature selection, the objective is to identify the best features for constructing useful models. In common usage, correlation refers to the likelihood that two variables have a linear relationship with each other. As a result, features with high correlations are linearly dependent, meaning they have almost the same effect on the dependent variable. Therefore, we can drop one of the two features when two features have a high correlation. [12] There are many features selection techniques, but since we do not know the target variable yet, we cannot use advanced features selection techniques.

E. Feature Scaling

Different datasets have different attributes, such as magnitude, variance, standard deviation, mean value, etc. Statistics can be affected by differences in the scale or magnitude of attributes. Linear models prefer variables whose ranges are larger over those whose ranges are smaller. Euclidean distances can also be affected by feature magnitudes. Feature scaling is one of the most important transformations you can apply to your data. With few exceptions, Machine Learning algorithms don't perform well when input numerical attributes have different scales. Scaling your data is one of the most important transformations you can perform. Normalization and standardization are the two most common methods of ensuring that all attributes share a common scale. [2] [9]

When the data distribution is unknown or when it is not Gaussian, normalization is used. Data normalization involves transforming the numeric values of columns to a common scale, without distorting the differences in the range of values. In our implementation, we have used min-max scaling. Where normalized data = data - minimum element / max element - min element

In standardization, the data is moved so that the mean is 0 and the standard deviation is 1. It is useful when we compare measurements with different units. Variables that are measured at different scales may contribute to the analysis in different ways, leading to bias. In standardization, the data is assumed to have a Gaussian distribution.

The formula used is: Standard data = data - mean / standard deviation.

F. Balancing Data

We observe that the data is unbalanced in terms of number of observations across the classes in a dataset. Imbalanced data refers to a situation where the number of observations for each class is different. As a result, machine learning models typically favour the class with the highest proportion of observations, also known as the majority class, which may lead to misleading results. In particular, this may be problematic if we are interested in the correct classification of a ”rare” class (also called minority class) but find high accuracy that is actually the result of the correct classification of the majority class. Consequently, if the model doesn't have the necessary amount of information about the class, it won't be able to make an accurate prediction. In our project, we have used under-sampling to balance the dataset, in which we delete the rows of the majority class and match the number of both majority and minority classes.

G. Splitting Data

We need to split the data in training and testing the data sets so we can train the machine learning model on the training data and see how it performs on the testing data which it has not seen during training. This is to make sure our model is not under or overfitting the data and is not just remembering the training set. We usually split the data in an 80:20 ratio where 80 percent is the training data, so even if we are splitting the data we are making sure we are providing the model with lots of examples to learn, and 20 percent data is used for testing the model. Another common practice is to divide the data into a 75:5:20 ratio where 75 percent is training data, 5 percent is validation data, and 20 percent is the testing data. But this is ratio can be changed as per individual preferences. In our data sets we have divided the data in an 80:20 ratio.

III. Conclusion

We have checked for missing values that can affect the models and the libraries we are working with and can lead to incorrect results, we have checked for outliers and removed them so that the model won't be affected by the outliers, but removing outliers may not always be a good idea but for data is necessary, we then found out the important features using covariance matrix also called feature selection, Afterwards, we normalized the data so that all the features will be in the same range so that the machine learning model won't prioritize a specific feature, we then balanced the data so that it would not be biased toward one particular label. We were unable to use more advanced techniques to detect outliers or select features since there is no target variable available, which prevented us from using more accurate and useful methods to perform these tasks.

IV. Acknowledgment

I should acknowledge the material provided by professor Angelov, Plamen P. during the lab sessions, and would also acknowledge the creator of the library as libraries like numpy, pandas, seaborn, sklearn, scipy are used in the pre processing.

References

[1] Jiawei Han, Jiawei, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques : Concepts and Techniques”, Morgan Kaufmann, June 2011

[2] AI Sciences OU, “Data Preprocessing with Python for Absolute Beginners”, Packt Publishing, March 2021

[3] Angelov, Plamen P., “Autonomous learning systems: from data streams to knowledge in real-time”, John Wiley and Sons Ltd, 2012

[4] S. Pratro and B. S. Panda, ”A Novel Concept and Review on Retrieval of Missing Data,” 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020, pp. 1-4, doi: 10.1109/ICCSEA49143.2020.9132962.

[5] Vinutha, H.P., Poornima, B., Sagar, B.M., “Detection of outliers using interquartile range technique from intrusion dataset”, Advances in Intelligent Systems and Computing, 701, pp. 511-518, 2018

[6] J. Xi, ”Outlier Detection Algorithms in Data Mining,” 2008 Second International Symposium on Intelligent Information Technology Application, 2008, pp. 94-97, doi: 10.1109/IITA.2008.26.

[7] Al-Sakib Khan Pathan, “The State of the Art in Intrusion Prevention and Detection”, Auerbach Publications, January 2014

[8] Joanne Rodrigues-Craig, “Product Analytics: Applied Data Science Techniques for Actionable Consumer Insights”, Addison-Wesley Professional, September 2020

[9] Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition”, O'Reilly Media, Inc., September 2019

[10] Suresh Kumar Mukhiya, Usman Ahmed, “Hands-On Exploratory Data Analysis with Python”, Packt Publishing, March 2020

[11] John W Graham, “Missing Data Analysis: Making It Work in the Real World”, Annual Review of Psychology 60(1):549-76, January 2009

[12] https://www.towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf

V. Appendix

A. Figures in the code

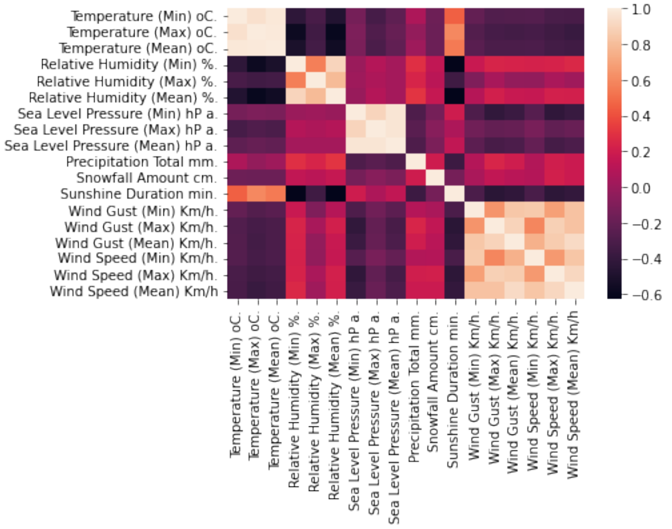

Fig. 1. Heatmap showing the correlation between different variables

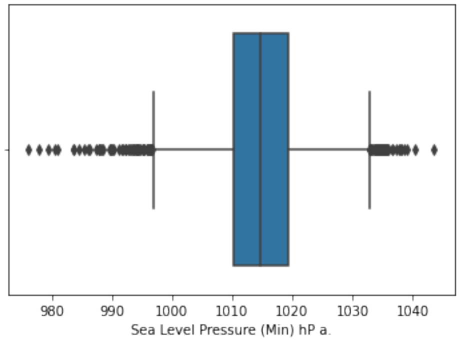

Fig. 2. Boxplot showing the outliers before any outliers were detected

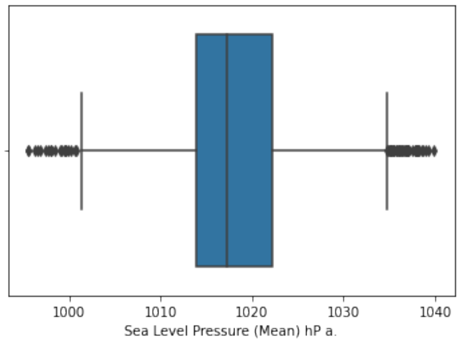

Fig. 3. Boxplot showing how the zscore approach for outlier detection has removed some of the outliers

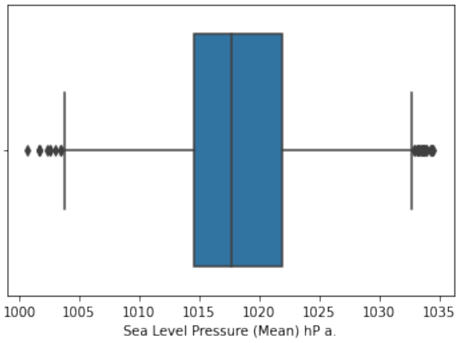

Fig. 4. Boxplot showing how the IQR approach removes outliers

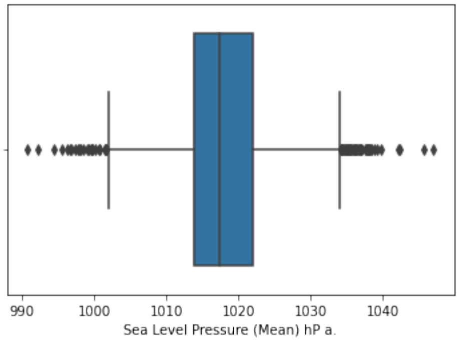

Fig. 5. Boxplot showing how the distance approach removes outliers