Computer Science and Networking, School of Computing and Data Science, Wentworth Institute of Technology

Data Mining

Dr. Salem Othman

Summer 2023

Assignment 2: Data

Operations:

	Attribute Type	Description	Examples	Operations
Categorical Qualitative	Nominal	Nominal attribute values only distinguish. (=, ≠)	zip codes, employee ID numbers, eye color, sex: {*male, female*}	mode, entropy, contingency correlation, $\chi 2$ test
	Ordinal	Ordinal attribute values also order objects. (<, >)	hardness of minerals, {*good, better, best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric Quantitative	Interval	For interval attributes, differences between values are meaningful. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
	Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, current	geometric mean, harmonic mean, percent variation

Task 1: Operations

You are assigned to conduct an in-depth investigation of all the operations associated with each attribute type mentioned above. This investigation should encompass the following elements:

1. Comprehensive explanation of each operation.

2. Elucidation of why a specific operation is suitable for its corresponding attribute type, and why it isn't as relevant or beneficial for the other three attribute types.

3. Illustration of how these operations are applied in real-world contexts using the provided examples.

Keep in mind that all operations for each attribute must be addressed thoroughly to receive full credit for this assignment. You will be expected to present your findings and the process of your research to the class.

Task 2: Reduce Noise

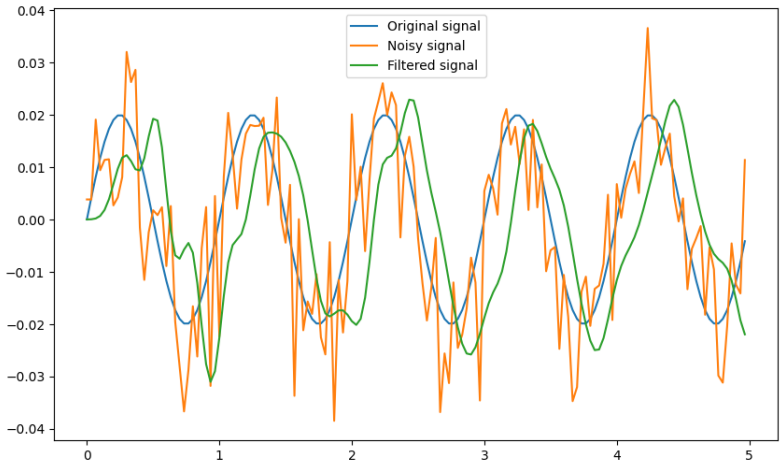
Question: Imagine that you are analyzing audio data from a voice recording and you notice that it contains a significant amount of noise, probably due to poor recording quality or background noises.

Your task is to use the Python scipy library's built-in functions to apply a digital filter to this audio data to reduce the noise. Specifically, use a **Butterworth filter**, which is a type of signal processing filter designed to have as flat as frequency response as possible in the passband.

To simulate this scenario, generate a noisy sine wave and then apply a Butterworth filter to it. Plot both the noisy signal and the filtered signal for comparison.

Hint: Use the scipy.signal.butter and scipy.signal.lfilter functions to create and apply the Butterworth filter. You will need to decide on appropriate parameters for the filter based on the characteristics of your signal and noise.

This exercise will help you understand the practical application of digital filters in reducing noise in real-world data, a common task in various fields including signal processing and data analysis.



Some useful links about Butterworth filter:

https://www.youtube.com/watch?v=dmzikG1jZpU

https://www.youtube.com/watch?v=-EHRFrDujhc

Task3: Detecting Outliers in Health Insurance Claims

Your task is to detect outliers in a health insurance claims dataset. The dataset "Medical Cost Personal Dataset" is available on Kaggle and it captures information about patients and their corresponding charges for various treatments.

https://www.kaggle.com/datasets/mirichoi0218/insurance

The 'charges' column in this dataset can be considered for outlier analysis. Outliers in this context could represent potential cases of insurance fraud or data entry errors that could have substantial implications for healthcare providers.

Use the **Isolation Forest algorithm** to detect outliers in the 'charges' column and analyze the results. Here's how you can approach the task:

1. Load the dataset into a pandas DataFrame.

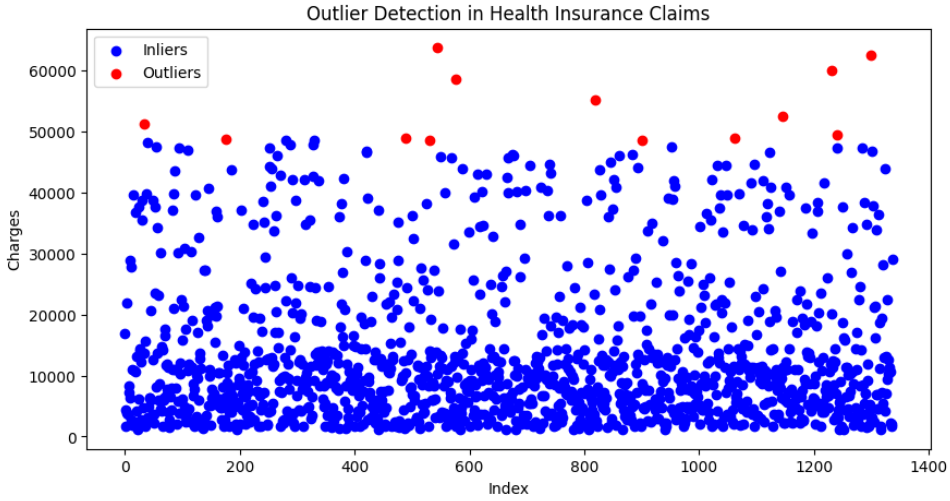
2. Use the IsolationForest algorithm from the sklearn.ensemble module. Choose an appropriate value for the 'contamination' parameter based on your initial assessment of the data.

3. Fit the model to the 'charges' column and predict the outliers.

4. Visualize the results using a scatter plot where outliers are marked with a different color.

5. Analyze your results. Do the identified outliers make sense? Could there be potential reasons for these outliers?

Remember, it is just as important to be able to interpret the results as it is to run the code. Good luck!



Deliverables:

You should upload your file to Brightspace with all the required information.