**Experiment 1**

**Name: SAP ID:**

|  |  |
| --- | --- |
| **Date:** |  |
| Aim | **Data Pre-processing: Clean and pre-process a given time series dataset, handling missing values.** |
| Software | Colab |
| Pre-requisite | Internet and required dataset |
| Theory | https://miro.medium.com/v2/resize:fit:700/1*qPy-8oH-yXi5M3tc7aJdSA.jpeg  Time series data is found everywhere, and to perform the time series analysis, we must preprocess the data first. Time Series preprocessing techniques have a significant influence on data modeling accuracy.   1. Definition of time-series data and its importance. 2. Preprocessing steps for time series data. 3. Structuring time-series data, finding the missing values, denoising the features, and finding the outliers present in the dataset.   To begin with, let’s understand the definition of time series first:  An example of time series would be the gold prices. In this case, our observation is the gold price collected over a while after fixed time intervals. The time unit could be in minutes, hours, days, years, etc. But the time difference between any two consecutive samples will be the same.  In this article, we will see the common time-series preprocessing steps that should be carried out before diving into the data modeling part. Let’s look at the common problems associated with the time-series data.  Preprocessing Time Series Data  Time Series data holds a lot of information, but generally, it is not visible. The common problems associated with time series are un-ordered timestamps, missing values (or timestamps), outliers, and noise in the data. Of all the mentioned problems, handling the missing values is the most difficult one. Since the conventional imputation (A technique used to take care of the missing data by replacing the missing values to retain most of the information) methods are not applicable while working with the time series data. To analyze the real-time analysis of this preprocessing, we will use Kaggle’s Air Passenger dataset,  Link for Dataset: <https://www.kaggle.com/datasets/rakannimer/air-passengers>  **Structuring Time Series**  Time Series data is generally found in unstructured formats, i.e., Timestamps could be mixed and not properly ordered. Also, most of the time, the date-time column has default string data type, and it is essential to covert the data-time column to date-time datatype first before applying any operation to it. Let’s implement this into our dataset:    **Missing Value Imputation in Time Series**  Handling the missing values in time series data is a challenging task. Conventional imputation techniques are not applicable for the time-series data since the sequence in which values are received matters. To address this problem, we have the following Interpolation methods:  Interpolation  Interpolation is a commonly used technique for time series missing value imputation. It helps in estimating the missing data-point using the two surrounding known data points. This method is simple and most intuitive. However, interpolation further has the following sub-methods:   * Time-Based Interpolation * Spline Interpolation * Linear Interpolation           All methods have given a reliable set of imputations. Imputations from these methods make more sense when the missing value window ( width of missing data) is small. For instance, if several consecutive values are missing, it becomes harder for these methods to estimate them.  **Denoising a Time Series**  Noise elements in a time series can cause significant problems, and noise removal is highly recommended before building any model. The process of carefully minimizing the noise is called denoising. Following are some methods commonly used for removing the noise from a time series:   * Rolling means   The Rolling mean is simply the mean for a window of previous observations, where the window is a sequence of values from the time series data. Mean is calculated for each ordered window. This can greatly help in minimizing the noise in time series data.    **Fourier Transform**  Fourier Transform can help remove the noise by converting the time series data into the frequency domain, and from there, we can filter out the noisy frequencies. Then, we can apply the inverse Fourier transform to obtain the filtered time series. Let’s use Fourier transform on the Google Stock Price.    **Outlier Detection in Time Series**  An outlier in time series refers to a sudden peak or drop in the trend line. We are not concerned with the factors causing the outliers, but certainly, there can be multiple factors. We will keep ourselves confined with the detection of outliers. Let’s take a look at the available methods for detecting the outliers:  Rolling Statistical Bound based approach  This method is most intuitive and works for almost all kinds of time series. In this method, upper and lower bounds are created based on specific statistical measures like mean and standard deviation, Z and T scores, and percentile of the distributions. For instance, we can define our upper and lower bound as:  https://miro.medium.com/v2/resize:fit:453/1*tiNRiw0Oipb4h6BKVflDZA.png  Taking the mean and standard deviation of the whole series is not advisable for outlier detection since the bound would be static in that case. The bounds should be created on a rolling basis, like considering a continuous set of observations to create bounds and then shifting to another window. This method is highly effective and simple for outlier detection.  **Isolation Forest**  As the name suggests, Isolation forest is a decision tree-based machine learning algorithm for anomaly detection. It works by isolating the data points on a given set of features using the decision tree’s partitions. In other words, It takes a sample out of the dataset and builds trees over that sample until each point is isolated. To isolate a data point, partitions are made randomly by selecting a split between the max and min values of that feature until each point is isolated. Random partition of features will create shorter paths in trees for the anomalous data points and thus distinguishing them from the rest of the data.  https://miro.medium.com/v2/resize:fit:700/1*P8YwDuGw48GTWV1sI7DBog.png  **K-means Clustering**  K-means clustering is again an unsupervised machine learning algorithm frequently used to detect outliers in time series data. This algorithm looks at the data point in the dataset and groups the similar data points into K number of clusters. Anomalies are distinguished by measuring the distance of a data point to its nearest centroid. If the distance is greater than a certain threshold value, the data point is marked as an anomaly. K-Means algorithm uses the Euclidean Distances for comparison.  https://miro.medium.com/v2/resize:fit:496/1*mHy3NgfkqGeasIPH9nLBlA.jpeg |
| Result |  |
| Conclusion | Write your detail understanding about the experiment. |