Pregrad

Data Preprocessing in Machine Learning

Introduction

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Such is the hype of machine learning and data science now a days that beginners or wannabe beginners think that they only need to apply machine learning algorithms on data set using Python & R packages and this will create the magic of AI. And then reality bites them when they are told that the very first thing they have to do is data preprocessing in machine learning, which will not only consume majority of their time but might also be equally boring.

A good data preprocessing in machine learning is the most important factor that can make a difference between a good model and a poor machine learning model. In this post we will first understand the need of data preprocessing and then present a nutshell view of various steps that are involved in this process.

Pregrad

Need of Data Preprocessing in Machine Learning

Garbage In Garbage Out

In computer science, there is a concept of Garbage In Garbage Out which means that faulty & poor quality of input, even to best of computing system will produce only a bad output.

This principle also applies to data that we feed to a machine learning system. If data, full of inconsistencies is given as input to machine learning system, it will in turn only create a poorly trained model which produces meaningless results. On the other hand if before building model, the garbage data is properly preprocessed and converted to quality, clean data even the resulting machine learning model will be of great quality.

Pregrad

Need of Data Preprocessing

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithm are executed in one data set, and best out of them is chosen.

Pregrad

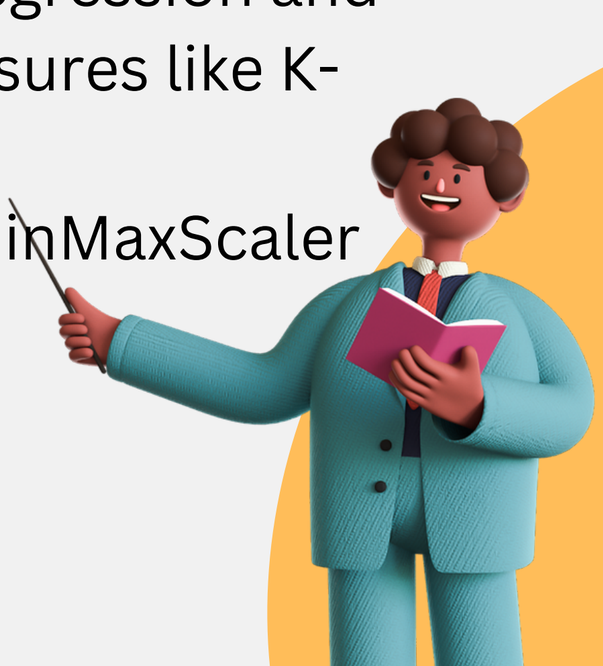
1. Rescale Data

When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.

This is useful for optimization algorithms used in the core of machine learning algorithms like gradient descent.

It is also useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like K-Nearest Neighbors.

We can rescale your data using scikit-learn using the MinMaxScaler class.



Pregrad

2. Binarize Data (Make Binary)

We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.
This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.
We can create new binary attributes in Python using scikit-learn with the Binarizer class.

3. Standardize Data

Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
We can standardize data using scikit-learn with the StandardScaler class.

Pregrad

Steps for Data Preprocessing in Machine Learning

Data Preprocessing in Machine Learning can be broadly divide into 3 main parts –
Data Integration
Data Cleaning
Data Transformation

There are various steps in each of these 3 broad categories. Depending on the data and machine learning algorithm involved, not all steps might be required though.
Let us now cover these one by one.

Data Integration and formatting

During hackathon and competitions, you are usually provided with a single csv or excel containing all training data. But in real world, your source of data might not be this simple. In real life, you might have to extract data from various sources and have to integrate it.

Again not all sources might be giving you data in same format. For e.g. one source might be giving you data in csv and other in XML. So during data integration from various sources, you will have to bring entire data in a common format.

Data Cleaning

Dealing with Missing data

It is not uncommon to have some missing data in the real world data set. Most of the machine learning algorithms will not work with missing or null data. So it becomes important to deal with missing data. Some of common measures taken are

Remove the column if there are plenty of rows with null values.

Remove the row if there are plenty of columns with null values.

Replace the missing value by mean or median or mode of that column depending on data distribution in that column.

In case of categorical feature column, we can consider missing data as a new category in itself by replacing the missing values by ‘NA’ or ‘Unknown’ or some other relevant term.

In this method we try to apply regression or classification techniques to come up with educated guesses of possible candidate to replace missing value.

Remove Noise from Data

Noise are slightly erroneous data observations which does not comply with trend or distribution of rest of data. Though each error can be small but collectively noisy data results in poor machine learning model. Noise in data can be minimized or smooth out by using below popular techniques –

Binning

Regression

Remove Outliers from Data

Outliers are those observation that has extreme values, much beyond the normal range of values for that feature. For example, a very high salary of CEO of a company can be an outlier if we consider salary of other regular employees of the company. Even few outliers in data set can contribute to poor accuracy of machine learning model. The common methods to detect outliers and remove them are –

Clustering

Box Plot

Dealing with Duplicate Data

The approach to deal with duplicate data depends on the fact whether duplicate data represents the real world scenario or is more of a inconsistency. If it is former than duplicate data should be preserved, else it should be removed.

Dealing with Inconsistent Data

Some data might not be consistent with the business rule. This requires domain knowledge to identify such inconsistencies and deal with it.

Data Transformation

Feature Scaling

Feature scaling is one of the most important prerequisite for most of the machine learning algorithm. Various features of data set can have their own range of values and far different from each other. For e.g. age of human mostly lies within range of 0-100 years but population of countries will be in ranges of millions.

This huge differences of ranges between features in a data set can distort the training of machine learning model. So we need to bring the ranges of all the features at common scale. The common approaches of feature scaling are –

Mean Normalization
Min-Max Normalization
Z-Score Normalization or Standardization

Data Transformation

Categorical Data

Categorical data, also known as qualitative data are text or string based data. Example of categorical data are *gender of persons (Male or Female), names of places (India, America, England), color of car (Red, White)*.

Most of the machine learning algorithms works on numerical data only and will not be able to process Categorical data. So we need to transform categorical data into numerical form without loosing the sense of information. Below are the popular approaches to convert categorical data into numerical form –

Label Encoding
One Hot Encoding
Binary Encoding

Data Transformation

Dealing with Imbalanced Data Set

Imbalanced data sets are type of data set in which most of the data belongs to only one class and very few data belongs to other class. This is common in case of medical diagnosis, anomaly detection where the data belonging to positive class is a very small percentage.

For e.g. only 5-10% of data might belong to a disease positive class which can be an expected distribution in medical diagnosis. But this skewed data distribution can trick the machine learning model in training phase to only identify the majority classes and it fails to learn the minority classes. For example, the model might fail to identify the medical condition even though it might be showing a very high accuracy by identifying negative scenarios.

We need to do something about imbalanced data set to avoid a bad machine learning model.

Below are some approaches to deal with such situation –

Under Sampling Majority Class

Over Sampling Minority Class

SMOTE (Synthetic Minority Oversampling Technique)

Data Transformation

Dimension reduction

Sometimes too much of information is not good. This is also applicable to data set which has way too many features (dimensions). A model trained on hundred of features might turn out to be a utter garbage. This is also known as curse of dimensionality.

To avoid this, we need to reduce dimension of the data set. The common methods of dimension reductions are –

PCA (Principal Component Analysis)

Factor Analysis

LDA (Linear Discriminate Analysis)

Data Transformation

Training-Test Data Split

This is usually the last step of data preprocessing just before training supervised machine learning model. In this step, given data set, after undergoing all cleaning and transformation is divided into two parts – one for training of machine learning model and second for testing the trained model.

Though there is no rule of thumb, but usually training-test split is done randomly at 80%-20% ratio. While splitting data set, care has to be taken that there is no loss of information in training data set.

Pregrad

Do You Have Any Question?