Common issues with the data and Data cleaning

Common issues with data

1. Ethical Issues
2. Bias in Data Sources
3. Noise in Data
4. Sources of Randomness

▼ Related Articles/Blogs/ Websites

- https://www.privacypolicies.com/blog/global-privacy-laws-explained/
- https://www.dlapiperdataprotection.com/

Cleaning the data

Machine learning is a data driven technology and thus, it's very important to know how to obtain these datasets, cleaning & shaping and curating (How to Curate A Ground Truth For Your Business Dataset) these datasets to meet your needs. It's also important to know the pitfalls of real-world datasets and how to avoid them. Further, to obtain a satisfactory solution one should learn how to cope with the shortcomings of data collection process and the Data Processing framework (A Beginners Guide for Data Preprocessing) as a whole.

Introduction to Data Preprocessing in Machine Learning

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data

tds https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d

A Beginners Guide for Data Preprocessing

▼ Related Courses/Tutorials

- Machine Learning: Zero to Mastery by Andrei Neagoi & Daniel Bourke

▼ Useful Articles, Blogs & Papers

- Data preprocessing for machine learning: options and recommendations

- Data Preprocessing vs. Data Wrangling
- This paper (Learning From Multiple Annotators: A Survey) by Sharan Vaswani and Mohamed Osama Ahmed provides a survey of existing techniques for creating a labelled dataset.
- Crowdsourcing is one way of getting labelled data, by outsourcing to vast, not-necessarily-expert groups of people. But the "wisdom of the crowd" isn't always accurate. This paper (Inferring the Ground Truth Through Crowdsourcing) describes some ways to use crowdsourced data without necessarily requiring everyone in your crowd to be accurate.

Data Preprocessing may include the following steps.

Handling Missing Data

Dealing with Missing Data

Nearly all of the real-world datasets have missing values, and it's not just a minor nuisance, it is a serious problem that we need to account for. Missing data - is a tough problem, and, unfortunately, there is no best way to deal with it.

https://medium.com/@danberdov/dealing-with-missing-data-8b71cd819501

Handling Outliers

The outliers in your data set are often a mixed bag. On one hand because they're so different from the main group, they may contain key information and represent meaningful anomalies that are important for understanding what's going on. On the other hand, they might be meaningless noise or mistakes that throw off your model. In terms of machine learning, training your QuAM with a data set that includes outliers may allow your QuAM to generalize well, but it could also confuse it as it tries to compensate for those far-off data points. In general, it's good to look at both possibilities.

Explore your data with and without outliers. If you decide that you need them for building your QuAM, select a learning algorithm that will be robust enough to handle them. If you find they're not useful, then it's better to just drop them.

Grouping Sparse Classes

Sparse classes are categories of features that have very few total observations. They can cause problems with some machine learning algorithms because they either try to compensate more than they should for these uncommon categories or ignore them entirely, and this results in inappropriate QuAMs. Histograms, are a good way to identify sparse classes. If you decide you don't want these sparse classes, you can combine them with the closest relevant class or combine them with other sparse classes into some other class.

Feature Scaling

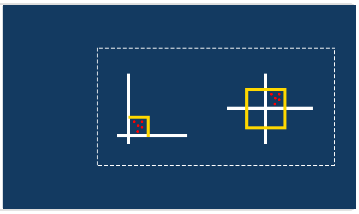
Also known as Data Transformation. The most frequently used techniques are Data Normalization & Standardization of Data.

You can always start by fitting your model to raw, normalized and standardized data and compare the performance for best results. It is a good practice to fit the scaler on the training data and then use it to transform the testing data. This would avoid any data leakage during the model testing process. Also, the scaling of target values is generally not required.

Feature Engineering: Scaling, Normalization, and Standardization (Updated 2023)

Learn how feature scaling, normalization, & standardization work in machine learning. Understand the uses & differences between these methods.

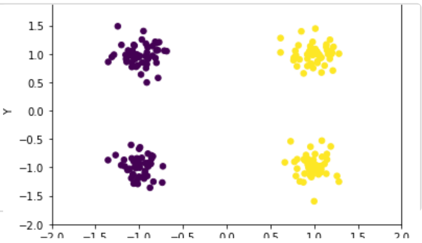
https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/



Understand Data Normalization in Machine Learning

If you're new to data science/machine learning, you probably wondered a lot about the nature and effect of the buzzword 'feature normalization'. If you've read any Kaggle kernels, it is very likely that you found feature normalization in the data

tds https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0



— **When to use Feature Scaling?**

Here’s the curious thing about feature scaling – it improves (significantly) the performance of some machine learning algorithms and does not work at all for others. Feature scaling should be used with these following algorithms -

- Gradient Descent Based Algorithms like linear regression, logistic regression, neural network, etc that use gradient descent as an optimization technique require data to be scaled.
- Distance Based Algorithms like KNN, K-means, and SVM are most affected by the range of features. This is because behind the scenes they are using distances between data points to determine their similarity.

On the other hand, Tree-based algorithms are fairly insensitive to the scale of the features.

When and Why to Standardize Your Data?

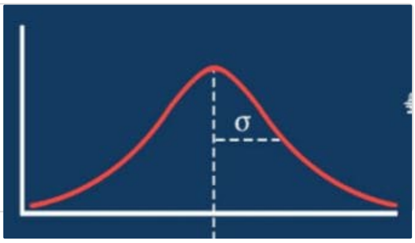
Some ML developers tend to standardize their data blindly before "every" Machine Learning model without taking the effort to understand why it must be used, or even if it's needed or not. So the goal of this post is to explain how, why

https://builtin.com/data-science/when-and-why-standardize-your-data

Data Transformation: Standardization vs Normalization - KDnuggets

Increasing accuracy in your models is often obtained through the first steps of data transformations. This guide explains the difference between the key feature scaling methods of standardization and normalization, and demonstrates when

https://www.kdnuggets.com/2020/04/data-transformation-standardization-normalization.html



Encoding

Machine learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model.

Encoding techniques vary based on the data itself. But most commonly used encoding techniques are Ordinal Encoding, One Hot Encoding & Dummy Variable Encoding.

— Some categories may have a natural relationship to each other, such as a natural ordering. For example the variable, *place: first, second, third* does have a **natural ordering of values**. This type of categorical variable is called an ordinal variable because the values can be ordered or ranked. For this type of categorical variables **ordinal encoding** is used.

— For categorical variables such as, *color: red, green, blue*, **where no ordinal relationship exists**, the integer encoding may not be enough at best, or misleading to the model at worst.

In this case, a **one-hot encoding** can be applied which transforms each categorical feature with “n” possible values into n_categories binary features, with one of them 1, and all others 0.

— If we use a **linear regression model** then the data must be encoded using **Dummy Variable Encoding**. The one-hot encoding creates one binary variable for each category. The problem is that this representation includes redundancy. For example, if we know that [1, 0, 0] represents “*blue*” and [0, 1, 0] represents “*green*” we don’t need another binary variable to represent “*red*“, instead we could use 0 values for both “*blue*” and “*green*” alone, e.g. [0, 0]. This is called a dummy variable encoding, and always represents C categories with C-1 binary variables.

Ordinal and One-Hot Encodings for Categorical Data - MachineLearningMastery.com

Machine learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model. The two most popular techniques are an Ordinal Encoding and a One-Hot

https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

To know about other encoding techniques read -

What are Categorical Data Encoding Methods | Binary Encoding

Learn what is categorical data and various categorical data encoding methods such as binary encoding, dummy, target encoding etc.

https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/