52100674 – Trần Thị Vẹn

Study issues in data processing for machine learning:

1. **Data cleaning**

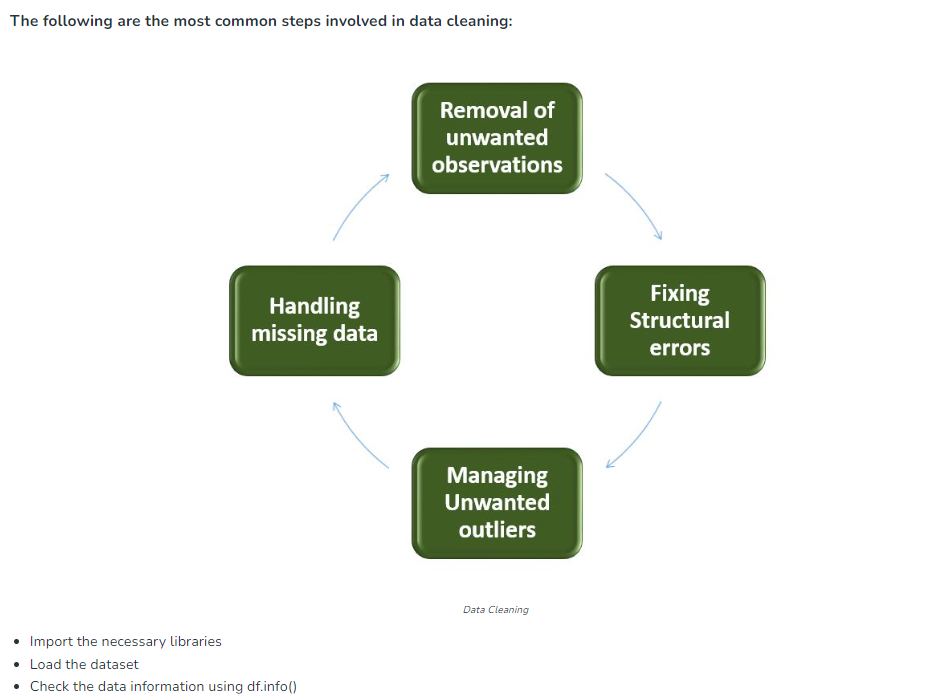
Data cleaning is one of the important parts of machine learning. It plays a significant part in building a mode.

Data cleaning is the process of preparing data for analysis by weeding out information that is irrelevant or incorrect.

This is generally data that can have a negative impact on the model or algorithm it is fed into by reinforcing a wrong notion.

Data cleaning not only refers to removing chunks of unnecessary data, but it’s also often associated with fixing incorrect information within the train-validation-test dataset and reducing duplicates.

**How to clean data for Machine Learning?**

****

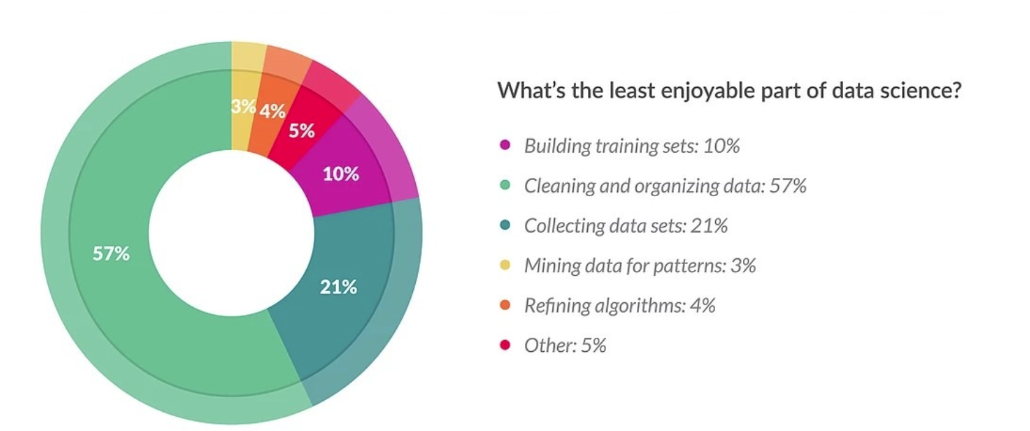
**Data cleaning is often the least enjoyable part of data science—and also the longest.**

indeed, cleaning data is an arduous task that requires manually combing a large amount of data in order to:

a) reject irrelevant information.

b) analyze whether a column needs to be dropped or not.

Automation of the cleaning process usually requires an extensive experience in dealing with dirty data. It’s kinda tricky to implement in a manner that doesn’t bring about data loss.



**The importance of data cleaning**

Data cleaning is a key step before any form of analysis can be made on it.

Datasets in pipelines are often collected in small groups and merged before being fed into a model. Merging multiple datasets means that redundancies and duplicates are formed in the data, which then need to be removed.

Also, incorrect and poorly collected datasets can often lead to models learning incorrect representations of the data, thereby reducing their decision-making powers. It's far from ideal.

The reduction in model accuracy, however, is actually the least of the problems that can occur when unclean data is used directly.

Models trained on raw datasets are forced to take in noise as information and this can lead to accurate predictions when the noise is uniform within the training and testing set—only to fail when new, cleaner data is shown to it.

Data cleaning is therefore an important part of any machine learning pipeline, and you should not ignore it.

**Data cleaning vs. data transformation**

As we’ve seen, data cleaning refers to the removal of unwanted data in the dataset before it’s fed into the model.

Data transformation, on the other hand, refers to the conversion or transformation of data into a format that makes processing easier.

In data processing pipelines, the incoming data goes through a data cleansing phase before any form of transformation can occur. The data is then transformed, often going through stages like normalization and standardization before further processing takes place.

1. **Data inspection and exploration:**

This step involves understanding the data by inspecting its structure and identifying missing values, outliers, and inconsistencies.

**Check the duplicate rows: df.duplicated()**

**Check the data information using df.info()**

**The descriptive structure of the data using df.describe()**

**Check the categorical and numerical columns df[cat\_col].nunique()**

1. **Removal of unwanted observations**

This includes deleting duplicate/ redundant or irrelevant values from your dataset. Duplicate observations most frequently arise during data collection and Irrelevant observations are those that don’t actually fit the specific problem that you’re trying to solve.

Redundant observations alter the efficiency to a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results.

Irrelevant observations are any type of data that is of no use to us and can be removed directly.

Now we have to make a decision according to the subject of analysis, which factor is important for our discussion. As we know our machines don’t understand the text data. So, we have to either drop or convert the categorical column values into numerical types. Here we are dropping the Name columns because the Name will be always unique and it hasn’t a great influence on target variables. For the ticket, Let’s first print the 50 unique tickets.

**df['Ticket'].unique()[:50]**

**Drop Name and Ticket columns.**

**df1 = df.drop(columns=['Name','Ticket'])**

**3. Handling missing data:**

Missing data is a common issue in real-world datasets, and it can occur due to various reasons such as human errors, system failures, or data collection issues. Various techniques can be used to handle missing data, such as imputation, deletion, or substitution.

**4. Handling outliers:**

Outliers are extreme values that deviate significantly from the majority of the data. They can negatively impact the analysis and model performance. Techniques such as clustering, interpolation, or transformation can be used to handle outliers.

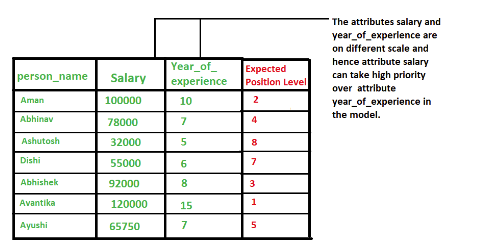
1. **Data normalization**

Data normalization is a technique used in data mining to transform the values of a dataset into a common scale. This is important because many machine learning algorithms are sensitive to the scale of the input features and can produce better results when the data is normalized.

There are several different normalization techniques that can be used in data mining, including:

* Min-Max normalization: This technique scales the values of a feature to a range between 0 and 1. This is done by subtracting the minimum value of the feature from each value, and then dividing by the range of the feature.
* Z-score normalization: This technique scales the values of a feature to have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean of the feature from each value, and then dividing by the standard deviation.
* Decimal Scaling: This technique scales the values of a feature by dividing the values of a feature by a power of 10.
* Logarithmic transformation: This technique applies a logarithmic transformation to the values of a feature. This can be useful for data with a wide range of values, as it can help to reduce the impact of outliers.
* Root transformation: This technique applies a square root transformation to the values of a feature. This can be useful for data with a wide range of values, as it can help to reduce the impact of outliers.

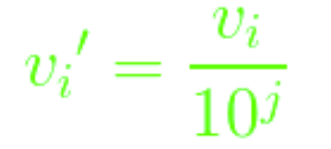
Normalization is generally required when we are dealing with attributes on a different scale, otherwise, it may lead to a dilution in effectiveness of an important equally important attribute(on lower scale) because of other attribute having values on larger scale. In simple words, when multiple attributes are there but attributes have values on different scales, this may lead to poor data models while performing data mining operations. So they are normalized to bring all the attributes on the same scale.



**Methods of Data Normalization**

1. **Decimal Scaling**

It normalizes by moving the decimal point of values of the data. To normalize the data by this technique, we divide each value of the data by the maximum absolute value of data. The data value, vi, of data is normalized to vi‘ by using the formula below – where j is the smallest integer such that max(|vi‘|)<1

****

Let the input data is: -10, 201, 301, -401, 501, 601, 701

To normalize the above data

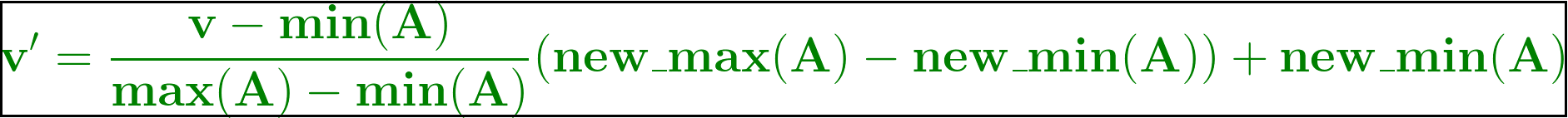
**Step 1: Maximum absolute value in given data(m): 701**

**Step 2: Divide the given data by 1000 (i.e j=3)**

**Result: The normalized data is: -0.01, 0.201, 0.301, -0.401, 0.501, 0.601, 0.701**

1. **Min-Max Normalization**

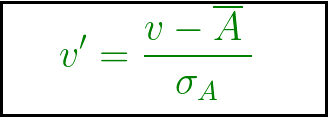
In this technique of data normalization, linear transformation is performed on the original data. Minimum and maximum value from data is fetched and each value is replaced according to the following formula.



Where A is the attribute data, Min(A), Max(A) are the minimum and maximum absolute value of A respectively. v’ is the new value of each entry in data. v is the old value of each entry in data. new\_max(A), new\_min(A) is the max and min value of the range(i.e boundary value of range required) respectively.

1. **z-Score Normalization(zero-mean Normalization)**

In this technique, values are normalized based on mean and standard deviation of the data A. The formula used is: v’, v is the new and old of each entry in data respectively. σA, A is the standard deviation and mean of A respectively.



Data normalization in data mining can have a number of advantages and disadvantages.

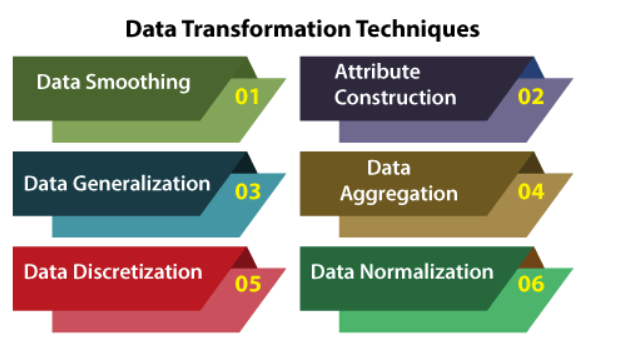
Advantages:

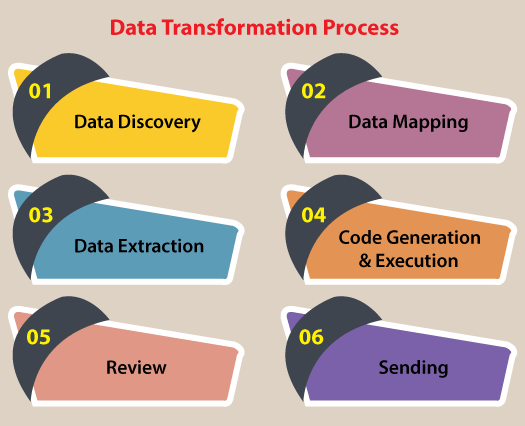
* Improved performance of machine learning algorithms: Normalization can help to improve the performance of machine learning algorithms by scaling the input features to a common scale. This can help to reduce the impact of outliers and improve the accuracy of the model.
* Better handling of outliers: Normalization can help to reduce the impact of outliers by scaling the data to a common scale, which can make the outliers less influential.
* Improved interpretability of results: Normalization can make it easier to interpret the results of a machine learning model, as the inputs will be on a common scale.
* Better generalization: Normalization can help to improve the generalization of a model, by reducing the impact of outliers and by making the model less sensitive to the scale of the inputs.

Disadvantages:

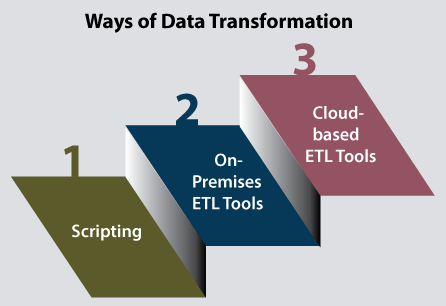
* Loss of information: Normalization can result in a loss of information if the original scale of the input features is important.
* Impact on outliers: Normalization can make it harder to detect outliers as they will be scaled along with the rest of the data.
* Impact on interpretability: Normalization can make it harder to interpret the results of a machine learning model, as the inputs will be on a common scale, which may not align with the original scale of the data.
* Additional computational costs: Normalization can add additional computational costs to the data mining process, as it requires additional processing time to scale the data.

1. **Data transformation**

****



* Data Discovery: During the first stage, analysts work to understand and identify data in its source format. To do this, they will use data profiling tools. This step helps analysts decide what they need to do to get data into its desired format.
* Data Mapping: During this phase, analysts perform data mapping to determine how individual fields are modified, mapped, filtered, joined, and aggregated. Data mapping is essential to many data processes, and one misstep can lead to incorrect analysis and ripple through your entire organization.
* Data Extraction: During this phase, analysts extract the data from its original source. These may include structured sources such as databases or streaming sources such as customer log files from web applications.
* Code Generation and Execution: Once the data has been extracted, analysts need to create a code to complete the transformation. Often, analysts generate codes with the help of data transformation platforms or tools.
* Review: After transforming the data, analysts need to check it to ensure everything has been formatted correctly.
* Sending: The final step involves sending the data to its target destination. The target might be a data warehouse or a database that handles both structured and unstructured data.



* Scripting: Data transformation through scripting involves Python or SQL to write the code to extract and transform data. Python and SQL are scripting languages that allow you to automate certain tasks in a program. They also allow you to extract information from data sets. Scripting languages require less code than traditional programming languages. Therefore, it is less intensive.
* On-Premises ETL Tools: ETL tools take the required work to script the data transformation by automating the process. On-premises ETL tools are hosted on company servers. While these tools can help save you time, using them often requires extensive expertise and significant infrastructure costs.
* Cloud-Based ETL Tools: As the name suggests, cloud-based ETL tools are hosted in the cloud. These tools are often the easiest for non-technical users to utilize. They allow you to collect data from any cloud source and load it into your data warehouse. With cloud-based ETL tools, you can decide how often you want to pull data from your source, and you can monitor your usage.

Present the problem and solution through practical examples. You should use UCI for data and Pandas for your solution.

Submit pdf file and code file in jupyter notebook format.