2.1 - Data Pre-Processing

Giovanni Della Lunga
giovanni.dellalunga@unibo.it

Introduction to Machine Learning for Finance

Bologna - February, 2022

Data Pre-Processing

- Raw data rarely comes in the form and shape that is necessary for the optimal performance of a learning algorithm.
- On the other hand, the success of a machine learning algorithm highly depends on the quality of the data fed into the model.
- Real-world data is often dirty containing outliers, missing values, wrong data types, irrelevant features, or non-standardized data.
- The presence of any of these will prevent the machine learning model to properly learn.
- For this reason, transforming raw data into a useful format is an essential stage in the machine learning process.

Data Pre-Processing

The topics that we will cover in this lesson are as follows:

- Removing and imputing missing values from the dataset
- Getting categorical data into shape for machine learning algorithms
- Feature Normalization
- Selecting relevant features for the model construction

Subsection 1

Missing Data

Dealing with Missing Data

- The real-world data often has a lot of missing values.
- The cause of missing values can be data corruption or failure to record data.
- The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.

Dealing with Missing Data

Delete Rows/Columns with Missing Values

- One of the easiest ways to deal with missing data is simply to remove the corresponding features (columns) or training examples (rows) from the dataset entirely;
- Missing values can be handled by deleting the rows or columns having null values. If columns have more than half of the rows as null then the entire column can be dropped. The rows which are having one or more columns values as null can also be dropped.
- Remember that, in pandas, rows with missing values can easily be dropped via the **dropna** method.

Dealing with Missing Data

Delete Rows/Columns with Missing Values

Pros:

- A model trained with the removal of all missing values creates a robust model.

Cons:

- Loss of a lot of information.
- Works poorly if the percentage of missing values is excessive in comparison to the complete dataset.

Dealing with Missing Data

Imputing missing values with Mean/Median:

- Columns in the dataset which are having numeric continuous values can be replaced with the mean, median, or mode of remaining values in the column.
- This method can prevent the loss of data compared to the earlier method.

Dealing with Missing Data

Imputing missing values with Mean/Median

Pros:

- Prevent data loss which results in deletion of rows or columns
- Works well with a small dataset and is easy to implement.

Cons:

- Works only with numerical continuous variables.
- Can cause data leakage
- Do not factor the covariance between features.

Subsection 2

Categorical Data

What is Categorical Data?

- Categorical data is a form of data that takes on values within a finite set of discrete classes.
- It is difficult to count or measure categorical data using numbers and therefore they are divided into categories.
- There are two types of categorical variables: **I. Ordinal Variables** and **II. Nominal Variables**

Categorical Data: Ordinal Variables

- These variables maintain a natural order in their class of values.
- If we consider the level of education then we can easily sort them according to their education tag in the order of *High School* ¡ *Under-Graduate* ¡ *post-Graduate* ¡ *PhD*.
- The review rating system can also be considered as an ordinal data type where 5 stars is definitely better than 1 star.

Categorical Data: Nominal Variables

- These variables do not maintain any natural/logical order.
- The color of a car can be considered as Nominal Variable as we cannot compare the color with each other.
- It is impossible to state that **Red** is better than **Blu** (subjective!).
- Similarly, Gender is a type of Nominal Variable as again we cannot differentiate between Male, Female, and Others.

Categorical Data: Encoding

- Most of the Machine learning algorithms are designed to work with numeric data.
- Hence, we need to convert Categorical(text) data into numerical data for model building.
- There are multiple encoding techniques to convert categorical data into numerical data. Let us look at some of them.

Categorical Data: Ordinal Encoding

- **Ordinal Encoding** primarily encodes ordinal categories into ordered numerical values.
- Ordinal encoding maps each unique category value to a specific numerical value based on its order or rank.
- Consider for example an education degree feature in a given data frame. Here, we can define the ordering of the categories when creating an ordinal encoder using sklearn. so, for example, we can arrange the order inside the categories as a list in ascending order. First, we have the High School followed by Associate,Master, and then Ph.D. at the end.
- *OrdinalEncoder(categories=[[HS, AS, M.S, Ph.D.]])*

Categorical Data: One Hot Encoding

- if there is no ordinal relationship between the categorical variables then ordinal encoding might mislead the model.
- This is because the ordinal encoder will try to force an ordinal relationship on the variables to assume a natural ordering, thus resulting in poor performance.
- In this case, One Hot encoder should be used to treat our categorical variables.
- It will create dummy variables by converting N categories into N features/columns.

Categorical Data: One Hot Encoding

- Considering the gender example. If we have a male in the first row, then its value is 1. Also if we have a female in the second row then its value is 0. Whenever the category exists its value is 1 and 0 where it does not.
- We can one-hot encode categorical variables in two ways: by using **get_dummies** in pandas and by using **OneHotEncoder** from sklearn.

	female	male
0	0	1
1	1	0
2	0	1
3	0	1
4	1	0

Categorical Data: Label Encoding

- The label encoder will convert each category into a unique numerical value.
- If implemented with Sklearn, then this encoder should be used to encode output values, i.e. y, and not the input X.
- It is similar to the ordinal encoder except, here the numeric values are assigned automatically without following any sort of natural order.
- Generally, the alphabetical order of the categorical values is used to determine which numerical value comes first.

Categorical Data: Label Encoding

- Consider for example a target variable Job Status that has four different categories.
- After applying label encoding to this column the four different categories are mapped into integers 0: Full Time, 1: Intern, 2: Part-Time, and 3:Unemployed.
- With this, it can be interpreted that Unemployed have a higher priority than Part-Time, Full Time, and Intern while training the model, whereas, **there is no such priority or relation between these statuses**.
- We cant define the order of labels with the label encoding technique.

Categorical Data: Label Encoding

```
from sklearn.preprocessing import LabelEncoder
lbe = LabelEncoder()
df['Employment Status']= lbe.fit_transform(df['Employment Status'])
df.head()
```

	Name	Gender	Education	State	Marital Status	Employment Status
0	Adam	male	Ph.D	NY	M	2
1	Dina	female	M.S	NY	D	2
2	John	male	AS	MI	S	0
3	Elton	male	HS	AL	M	3
4	Gina	female	HS	AK	D	1

Subsection 3

Feature Scaling (Normalization)

Feature Scaling

- Before using many ML algorithms (including those for unsupervised learning), it is important to scale feature values so that they are comparable.
- Z-score scaling involves calculating the mean (μ) and SD (σ) from the values of each feature from the training set. Scaled feature values for all data sets are then created by subtracting the mean and dividing by the SD.
- The scaled feature values have a mean of zero and SD of one.

$$\text{scaled feature value} = \frac{V - \mu}{\sigma} \tag{1}$$

Feature Scaling

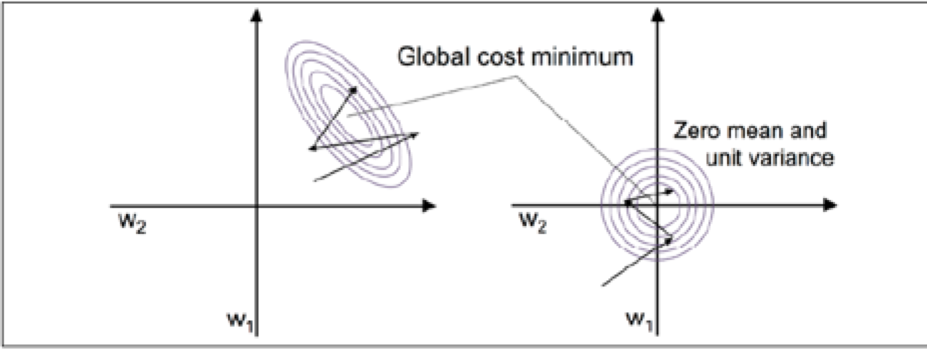
- Min-max scaling involves calculating the maximum and minimum value of each feature from the training set.
- Scaled feature values for all data sets are then created by subtracting the minimum and dividing by the difference between the maximum and minimum.
- The scaled feature values lie between zero and one:

$$\text{scaled feature value} = \frac{V - V_{min}}{V_{max} - V_{min}} \tag{2}$$

Data Cleaning

- Dealing with inconsistent recording
- Removing unwanted observations
- Removing duplicates
- Investigating outliers
- Dealing with missing items

Standardization and Gradient Descent



Principal Component Analysis