

FEATURE ENGINEERING

Krishna Kant Parida, Arpan Maity

CS460 Machine Learning, School of Computer Sciences,
National Institute of Science Education and Research (NISER), Bhubaneswar,
Homi Bhabha National Institute (HBNI)

February 26, 2023

PART I: FEATURE ENGINEERING

1	What is Feature Engineering ?	4
1.1	Example	5
1.2	Significance	6
2	Processes in Feature Engineering	7
3	Steps in Feature Engineering	8
3.1	Preparation of Data	8
3.2	Exploratory Data Analysis (EDA)	9
3.3	Benchmark	10

PART II: FEATURE ENGINEERING TECHNIQUES

1	Techniques involved in Feature Engineering	12
2	One-hot encoding	12
3	Bucketing/Binning	17
3.1	quantile bucketing	21
4	Normalization	22
4.1	Introduction	22
4.2	Scaling to a range	23
4.3	Clipping	24
4.4	Log Scaling	25
4.5	Standardization or Z-score	26
5	Handling Missing Features	27
5.1	Solving Techniques	28
6	Handling imbalanced dataset	29
6.1	Solving techniques	30

Part I

FEATURE ENGINEERING

WHAT IS FEATURE ENGINEERING ?

- ▶ It is used to select, manipulate and transform raw data into specific features which are used for preparing and improvising the predictive model. [Harshil Patel and Writer Aug 2021]
- ▶ Often called the **pre-processing** step of machine learning.
- ▶ Such features are added as new variables to the predictive model which weren't there in the training data set.
- ▶ Feature engineering involves much of the **Domain Knowledge**, i.e., the purpose for which the predictive model is being built and hence leads to identification and application of dependencies and improvisations among the features.

WHAT IS FEATURE ENGINEERING ?

EXAMPLE

Unless the user enters their own custom frame titles and subtitles, Elegant Slides automatically inserts the section title and, if specified, the subsection title as frame titles and frame subtitles.

WHAT IS FEATURE ENGINEERING ?

SIGNIFICANCE

Feature engineering "improves" the features in such a way that they describe the structures inherent in the provided dataset. Involving better features for the training leads to following advantages:

- ▶ **More flexibility:** Good features provide more flexibility, which helps to analyze datasets with much rigid features which keep on adding to various constraints to the model, making it more complex in terms of understanding as well as execution.
- ▶ **Simpler and fast Models:** As features are more flexible, less complex models can be built which are faster to run, easier to understand and easier to maintain. It alleviates the need to choose overfitting models with maximally screened and optimized parameters.
- ▶ **Better results:** Models made from better features are well balanced, i.e., neither underfitted nor overfitted with the training dataset. This leads to better results during testing.

PROCESSES IN FEATURE ENGINEERING

Feature engineering consists of various processes, as listed below:

- ▶ **Feature Creation:** Creating features can involve creating new parameters by adding or removing some features by using relation among them.
- ▶ **Transformations:** It is a function that transforms features from one representation to another. Such transformations are chosen which make the model more flexible and be able to take variety of data as input. It help to speed up training and increase the accuracy of the model.
- ▶ **Feature Extraction:** Involves extracting features from the raw data to identify useful information and generate new variables to be used in the model. It basically compresses the data and reduces them into manageable quantities for modelling, without distorting the original dependencies and significant information.
- ▶ **Feature Selection:** All above processes help us to either create or identify the appropriate features useful for building the model. As features are selected, any redundant feature which may negatively impact the model by reducing the overall performance and accuracy are removed.

STEPS IN FEATURE ENGINEERING

PREPARATION OF DATA

- ▶ The very first step of feature engineering.
- ▶ Raw data is acquired from various resources and is prepared into a suitable format in order to be used in the model.
- ▶ It involves various sub-procedures as listed:
 - **Cleaning:** Removal of incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.
 - **Delivery:** Process of sending the dataset from the data source to the system in which the model will be trained.
 - **Augmentation:** Set of techniques for artificially increase the amount of data by generating data points from existing data.
 - **Fusion:** Involves integration of the data obtained and delivered from several data sources for producing consistent, accurate and useful dataset.
 - **Ingestion/Loading:** Final process of importing and loading data for the model training.

STEPS IN FEATURE ENGINEERING

EXPLORATORY DATA ANALYSIS (EDA)

- ▶ One of the most important step, powerful yet simple step in feature engineering.
- ▶ It refers to the critical process of performing initial investigations on data so as to discover patterns, spot anomalies, test hypothesis and check initial assumptions.
- ▶ EDA involves analysis, investing and exploration of data and summarization of its main properties and characteristics into statistics and graphical representations.
- ▶ It can also be used to see what the given dataset can reveal beyond the formal training of the model, which also involves testing initial hypothesis, providing a better understanding of the data set variables and underlying relationships between them.
- ▶ It was originally developed by American mathematician John Tukey in the 1970s, and it continues to be a widely used method in the data discovery process.

STEPS IN FEATURE ENGINEERING

BENCHMARK

- ▶ It is a process of setting a standard **baseline** for accuracy and precision, in order to compare all the variables from it.
- ▶ Benchmark also keeps some minimal expectations from a model, and model satisfying such requirements is called benchmark/baseline model.
- ▶ Basically, benchmark are used as measure for comparing the performance among various machine learning model, and are independent of the amount of complexity in each model, as well as to that of the benchmark model.
- ▶ One can note that baseline model can also be replaced by any other model which are at par with the initial expectations but are more efficient and less complex. Hence, baseline is infact updated as per the model requirements.

Part II

FEATURE ENGINEERING TECHNIQUES

ONE-HOT ENCODING

- ▶ Simplest and basic categorical-column encoding method.
- ▶ Encoding procedure involves the representation of the element of any finite set by the index of that element in the set, i.e., the element under consideration is assigned to index "1", where all other elements are assigned values within the range of 0 to $n - 1$.
- ▶ Now, each of the digits out of $n - 1$ digits are treated as a new column, which allows removal of the original categorical column to which the element belong to.
- ▶ As it assigns a unique binary number of multiple digits for each possible case or category, making it different from other binary encoding schemes.

EXAMPLE

Considering we have a feature as color, we need 3 many bits and they are Red, Blue, and Green. Now to encode them we will follow the below procedure.

- ▶ For every feature value a fixed position in the array is assigned which is 1 and the rest is 0. It is like a vector where the basis is formed by the feature value and the coefficients form the encoded combination.

Feature	Encoding
Red	100
Blue	010
Green	001

Table. One-hot encoded table for the eg.

ONE COLD ENCODING

In the previous combination if the 0 are replaced with 1 and the 1 is replaced with 0. This type of encoding is known as one-cold encoding. The encoding is shown in the table below.

Feature	Encoding
Red	011
Blue	101
Green	110

Table. One-cold encoded table for the eg.

ADVANTAGES

- ▶ Categorical column is mapped into multiple binary columns, which are easy-to-use and faster to parse through.
- ▶ Features encoded by one-hot encoding can be easily applied into the models since each of the new binary column corresponds to a category in the original column.
- ▶ It is most useful for such models (like linear SVM) where all features are equally important and hence are assigned with same weight in the model.

DISADVANTAGES

- ▶ Any weight for each of the category, or any ordinal relation between the categories are removed. Making them not much useful for models involving Hierarchical classification.
- ▶ Features with very high cardinality in terms of categories are encoded with very large amount of dimensions corresponding to each of the binary column.
- ▶ This leads to various undesired issues like high training variance, decrease in accuracy and significant consumption of memory and computation.

BUCKETING / BINNING

- ▶ A form of transformation method from feature involving numerical value or attribute to a classification or categorical attribute.
- ▶ Usually for transformation, **bins** or **buckets** of specific numerical ranges are made such that any numerical attribute falling under the given range for the bin are classified accordingly.
- ▶ This method, other than numerical attribute, can also work on categorical attribute belonging to larger groups.
- ▶ Therefore, they are also useful in terms of classification based on hierarchy of categories, or preserving the ordinal relation between the categories.

EXAMPLE

Consider a distribution of age of a group with 10 people. We want to categorize it into 3 categories. Let the data be the following, Age=[10,20,29,30,60,65,40,27,35,28]. consider 3 categories, **Young**(<25), **Middle**(25<x<50), **Old**(>50). Then we have the given table for the bucketing.

Feature	Frequency
Young	2
Middle	6
Old	2

Table. Bucketing eg.

ADVANTAGES

- ▶ It is useful and sensible to use when the numerical values within a range of value have implications of a shared trait, which reduces overfitting.
- ▶ As values are classified into discrete categories, it reduces the impact of small errors by rounding off a given numerical value to its nearest representative.
- ▶ Data transformed in such a way can be easily processed or visualized using graphical representation like Histograms, bar graphs etc.

DISADVANTAGES

- ▶ Not useful when a required amount of precision for the model is crucial.
- ▶ It will look for common properties even if there are no such traits are present, due to which they are susceptible to outliers.

BUCKETING / BINNING

QUANTILE BUCKETING

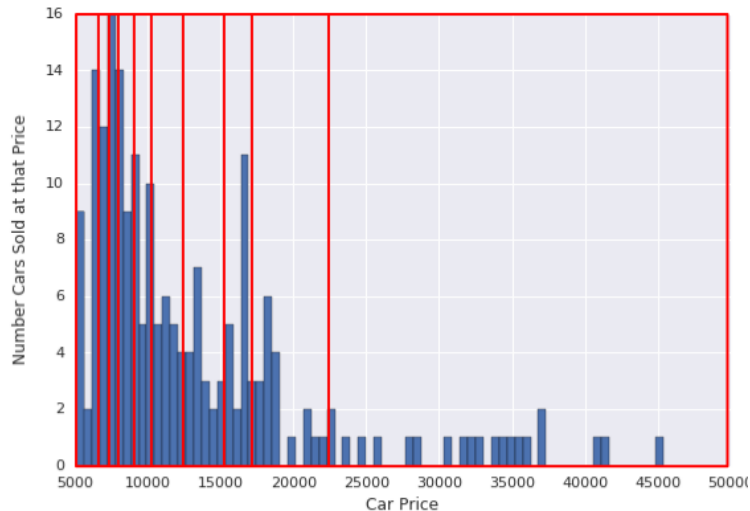


Figure. Quantile bucketing

In some problems, we divide the whole distribution with equal intervals. Some problem requires that the group size (frequency) has to be equal. In the later case we have to divide the area of the distribution into equal parts, such a bucketing technique is called [quantile bucketing](#). Such a bucketing will have a different range for the different categories. One such categorization is shown here.

NORMALIZATION

INTRODUCTION

- ▶ Normalization is used to transform features to be on a similar scale.
- ▶ Normalization is done to make the model perform better and make the training stable.
- ▶ The common techniques of normalization are:
 1. scaling to a range
 2. clipping
 3. log scaling
 4. z-score Normalization(**Standardization**)

NORMALIZATION

SCALING TO A RANGE

Scaling to a range is a common technique used for normalization in which the feature values within their normal range are transformed to a particular range which is usually 0 and 1. Consider the values to be from within the range of x_{min} and x_{max} then a value x is transformed as,

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

This is a good method for normalization if the following are true:

1. There are very less or no outliers in the data within our concerned range. We are sure about the fact that the distribution is approximately uniform within the range $(x_{max}-x_{min})$.

NORMALIZATION

CLIPPING

Clipping is used to eliminate the outliers in the data in case the outlier value is extreme. The clipping can be done for the minimum value as well as the maximum value. Consider the example if we are interested in the value of temperature in the order of mK for a certain problem and we have a very less number of values of the order of K then we can assign a particular value for those feature values which are greater than the threshold. The values above the threshold are fixed exactly to be the fixed value(say, $1K$). The same can be done on the minimum as well.

NORMALIZATION

LOG SCALING

Log scaling is as simple as the name suggests, the value of the feature is transformed into the log scale. This is useful when the distribution of the data is exponential approximately. When a small number of feature values is high in frequency and other values are less in frequency(This will give an exponentially decaying histogram). The relation for scaling is:

$$x' = A \log(x) \quad (2)$$

Where A is just a constant.

NORMALIZATION

STANDARDIZATION OR Z-SCORE

standardization or Z-score is the transformation of the feature value which is the number of standard deviations away from the mean, mathematically,

$$x' = \frac{x - \mu}{\sigma} \quad (3)$$

Where μ is the mean and σ is the standard deviation.

This scaling is useful when there are outliers and they are not extreme so that we don't have to clip them. Thus, it reduces the effect of the outliers on the features.

HANDLING MISSING FEATURES

The raw data needs pre-processing in terms of deciding the features and their values. The data might have a very less number of values for a particular feature or a very less number of features might have values. (refer to the table)

Here the f are the features and

(-) are the places that do not have a value. Here the feature f_1 have most of the values missing.

For serial No.2 only the feature f_2 has a value.

The f_2 has a value

missing for Sl No. 2.[[Tlameo Emmanuel 2022](#)]

Sl No.	f_1	f_2	f_3	f_4
1				
2	-		-	-
3	-			
4	-			-

Table. Table for data

HANDLING MISSING FEATURES

SOLVING TECHNIQUES

- ▶ The complete row can be removed from the data. In the above table Sl. No. 2. This can be done if we have enough data which allows such an operation or if the rows of such kind are less in number.
- ▶ The feature can be removed as well for e.g., f_1 if there are very very few values for a particular feature.
- ▶ The values can be predicted which are known as imputation methods. There are many imputation methods.
- ▶ value at the missing position with the average of the values present for that feature. For example in the case of f_3 at the position Sl. No. 2.
- ▶ The missing values can be interpolated from the known values.
- ▶ There are other data generation techniques which include algorithms like **K-NN**

HANDLING IMBALANCED DATASET

- ▶ The data set with skewed proportions for different classes results in an imbalanced dataset. If for a particular class, there is a large number of values and for another class, the values are very less in number then we have an imbalance in the data set.
[“<https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data>” visited-26-Feb-2023]
- ▶ The class that has a large proportion of values is called the Majority class and the class with a smaller proportion is called the minority class.
- ▶ In such a case it is difficult to find the relationship between the minority class and the labels, the model might ignore the minority class data. The model will be biased with the majority class.




HANDLING IMBALANCED DATASET

SOLVING TECHNIQUES

- ▶ **Downsampling and Upweighting:** The model is trained with a subset of the majority class and then a weight is added to the downsampled class by the factor it was downsampled.
- ▶ **Sampling techniques:**
 - **Data removal:** The values from the majority class can be removed by **Randomization** as well as **Prototype**.
 - **Addition of Data:** The values can be added to the minority class by **interpolation** and different **data generation** techniques.

The advantages of downsampling and upweighting technique is that it helps for the faster convergence of the model with lesser disk space while upweighting ensures the calibration or the results are in the probabilities.

REFERENCES I

-  Harshil Patel, Software Developer and Technical Writer (Aug 2021). “What is Feature Engineering — Importance, Tools and Techniques for Machine Learning”. In: *Towards Data Science*.
-  “<https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data>” (visited-26-Feb-2023). In.
-  Tlameo Emmanuel, Thabiso Maupong et. al. (2022). “A survey on missing data in machine learning”. In: *Journal of Big Data*.