# Feature Construction and Feature Selection

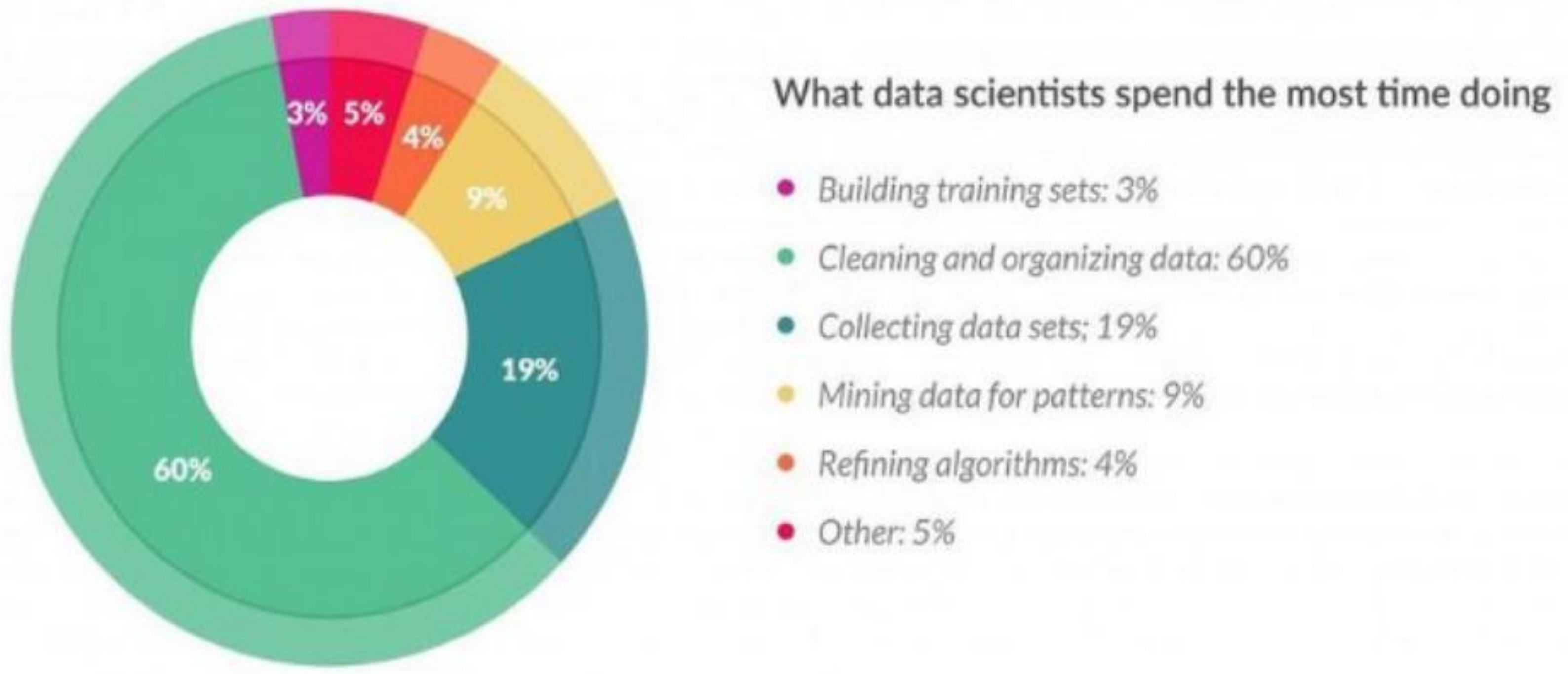
# Feature Engineering

## What is a feature and why we need the engineering of it?

- Basically, all machine learning algorithms use some input data to create outputs.
- This input data comprise features, which are usually in the form of structured columns.
- Algorithms require features with some specific characteristic to work properly.
- Need for **feature engineering** arises.
- Feature engineering efforts mainly have two goals:
    1. Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
    2. Improving the performance of machine learning models

# Feature Engineering

According to a survey in Forbes, data scientists spend **80%** of their time on **data preparation:**



Source: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/

## Feature Construction

- ❑ Imputing categorical features
- ❑ Encoding categorical variables
- ❑ Extending numerical features
- ❑ Text-specific feature construction

## Feature Construction

- ❑ Feature construction is the application of a set of constructive operators to a set of existing features resulting in construction of new features.
- ❑ Feature construction (also known as *constructive induction* or *attribute discovery*) is a form of data enrichment that adds derived features to data.
- ❑ Research demonstrated that feature construction can allow machine learning systems to construct more accurate models across a wide range of learning tasks.

## Imputing categorical features

### What is data imputation?

- ❑ **Imputation** is the process of replacing the missing data with approximate values.
- ❑ Instead of deleting any columns or rows that has any missing value, this approach preserves all cases by replacing the missing data with the value estimated by other available information.

## Imputing categorical features

- ❑ Missing values are one of the most common problems
- ❑ The reason for the missing values might be
  - ➢ human errors,
  - ➢ interruptions in the data flow,
  - ➢ privacy concerns.
- ❑ Missing values affect performance of the machine learning models.
- ❑ Some machine learning platforms automatically drop the rows which include missing values
- ❑ Decreases model performance because of the reduced training size
- ❑ Most of algorithms do not accept datasets with missing values.

# Imputing categorical features

- ❑ Simplest solution to missing values is to drop the rows or the entire column.
- ❑ There is not an optimum threshold for dropping but you can use **70%**.
- ❑ Try to drop rows and columns which have missing values with higher than this threshold.
- ❑ Replacing the missing values with the **maximum occurred value** in a column is a good option for handling categorical columns.
- ❑ If values in the column are **distributed uniformly** and there is not a dominant value, imputing a category like “**Other**” might be more sensible.

# Imputing categorical features

Based on problem at hand, we can try to do one of following:

1. Mode is one of the option which can be used
2. Missing values can be treated as a separate category by itself. We can create another category for the missing values and use them as a different level
3. If the number of missing values are lesser compared to the number of samples and also total number of samples is high, we can also choose to remove those rows in our analysis
4. We can also try to do an imputation based on the values of other variables in the given dataset. We can identify related rows to the given row and then use them for imputation
5. We can also run a model to predict missing values using all other variables as inputs.

# Imputing categorical features

## Other Imputation Methods

- ❑ Regression Imputation
- ❑ Maximum Likelihood
- ❑ Stochastic Regression Imputation
- ❑ Hot-Deck Imputation
- ❑ Cold-Deck Imputation

# Encoding categorical variables

- ❑ Machine learning and deep learning models, like those in Keras, require all input and output variables to be numeric.
- ❑ Two kinds of categorical data-
  1. **Ordinal Data:** The categories have an inherent order
  2. **Nominal Data:** The categories do not have an inherent order
- ❑ If your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model.

# Encoding categorical variables

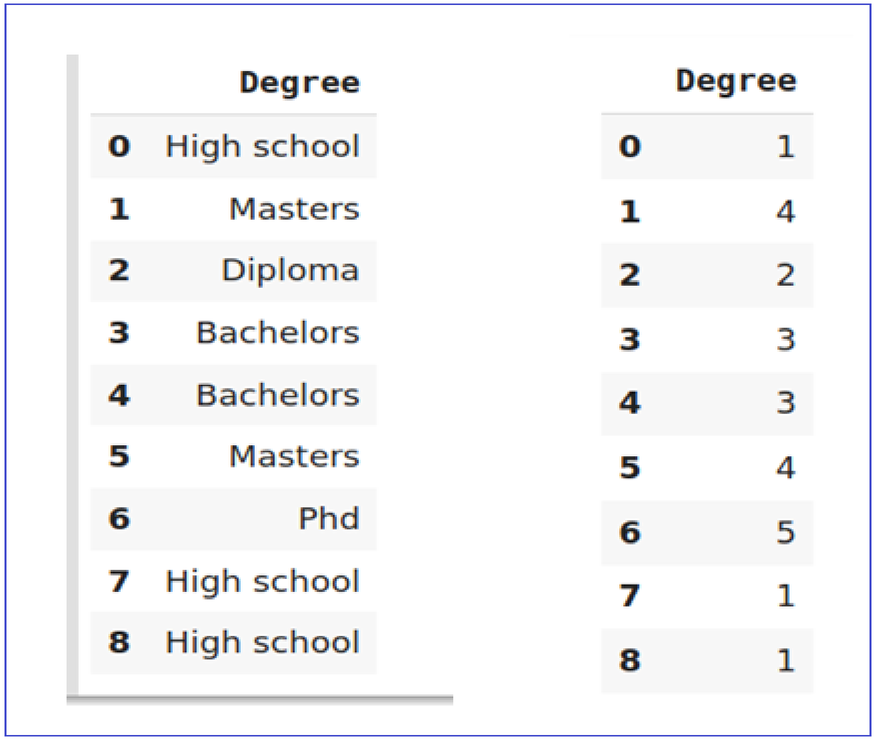
**The most popular techniques are**

- ❑ Label Encoding or Ordinal Encoding
- ❑ One hot Encoding
- ❑ Dummy Encoding
- ❑ Effect Encoding
- ❑ Binary Encoding
- ❑ BaseN Encoding
- ❑ Hash Encoding
- ❑ Target Encoding

# Encoding categorical variables

## Label Encoding or Ordinal Encoding

- ❑ Used when the categorical feature is ordinal.
- ❑ Retaining the order is important.
- ❑ Hence encoding should reflect the sequence.
- ❑ In Label encoding, each label is converted into an integer value.



|   | Degree |
|---|---|
| 0 | High school |
| 1 | Masters |
| 2 | Diploma |
| 3 | Bachelors |
| 4 | Bachelors |
| 5 | Masters |
| 6 | Phd |
| 7 | High school |
| 8 | High school |

|   | Degree |
|---|---|
| 0 | 1 |
| 1 | 4 |
| 2 | 2 |
| 3 | 3 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |
| 7 | 1 |
| 8 | 1 |

# Encoding categorical variables

## One Hot Encoding

- ❑ Use when the features are nominal (do not have any order).
- ❑ For each level of a categorical feature, create a new variable.
- ❑ Each category is mapped with a binary variable containing either 0 or 1.
- ❑ Here, 0 represents absence, and 1 represents presence of that category.
- ❑ Newly created binary features are known as **Dummy variables.**
- ❑ Number of dummy variables depends on levels in categorical variable.

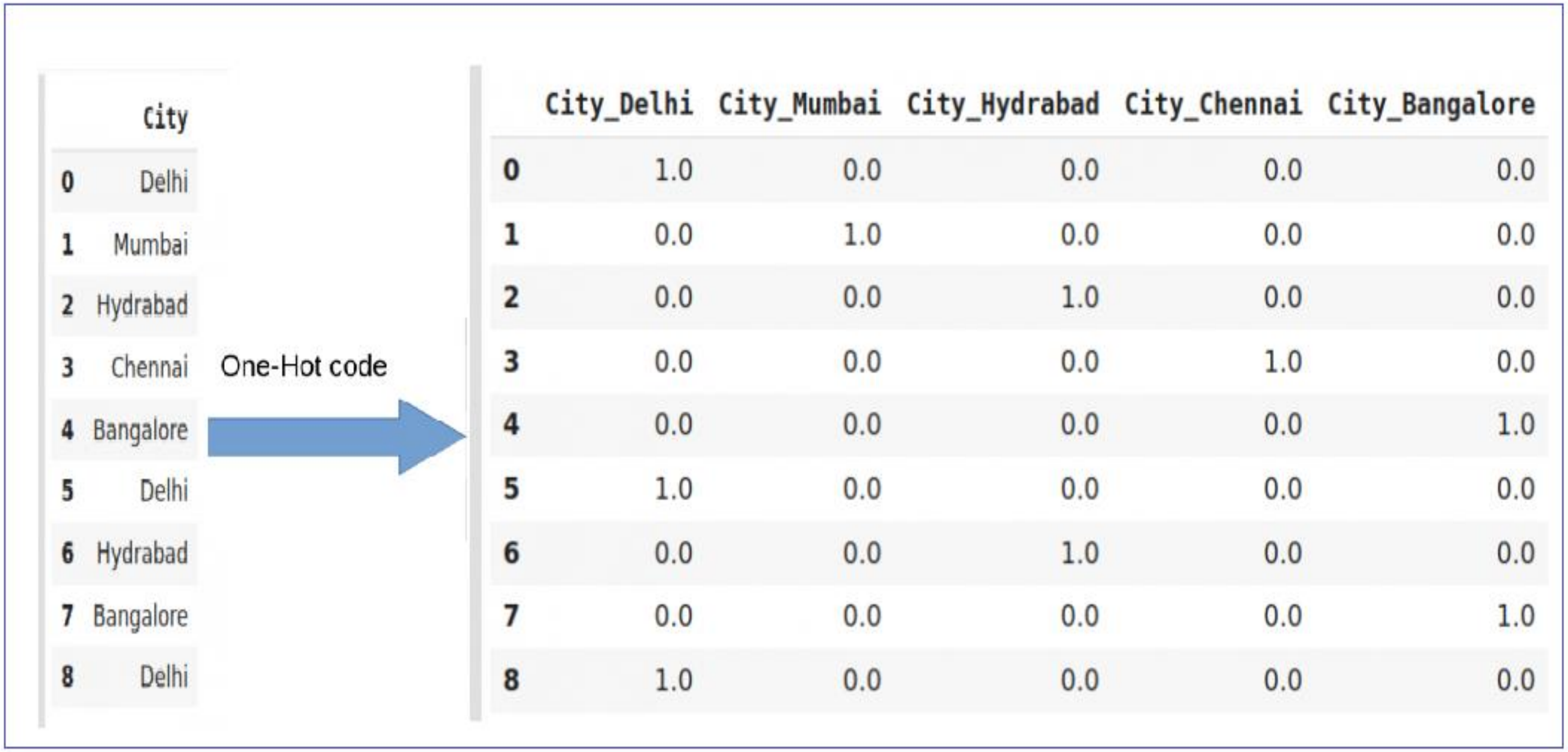
| Index | Animal |
|---|---|
| 0 | Dog |
| 1 | Cat |
| 2 | Sheep |
| 3 | Horse |
| 4 | Lion |

One-Hot code →

| Index | Dog | Cat | Sheep | Lion | Horse |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 1 | 0 |

## Encoding categorical variables

### One Hot Encoding Example



| | City |
|---|---|
| 0 | Delhi |
| 1 | Mumbai |
| 2 | Hydrabad |
| 3 | Chennai |
| 4 | Bangalore |
| 5 | Delhi |
| 6 | Hydrabad |
| 7 | Bangalore |
| 8 | Delhi |

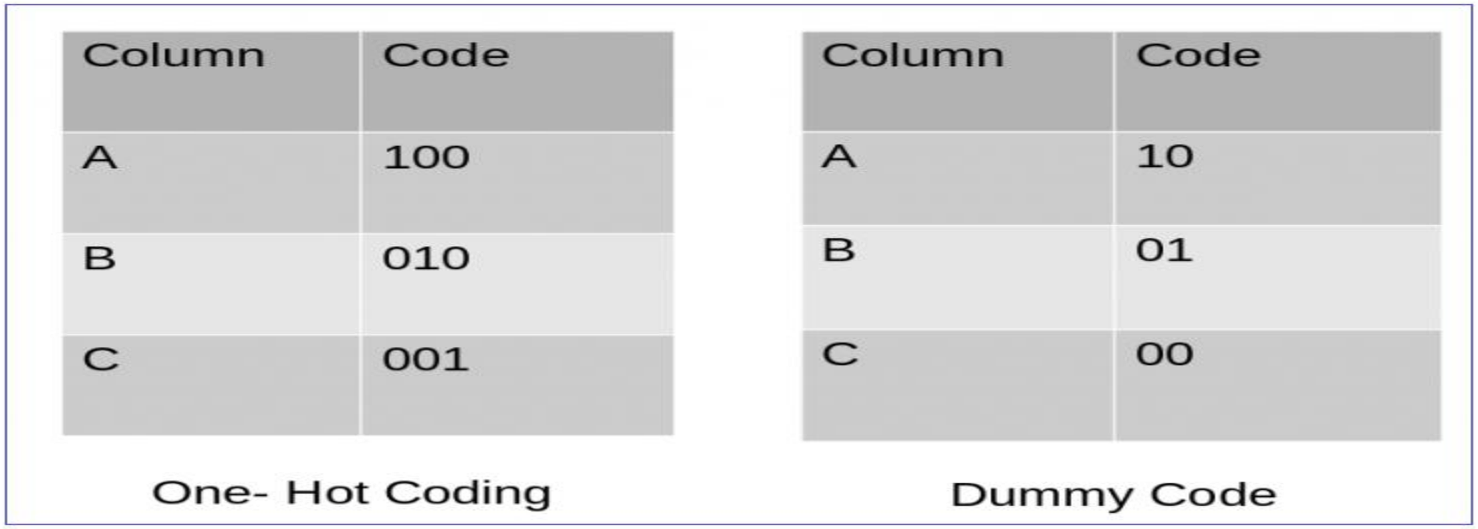
One-Hot code

| | City_Delhi | City_Mumbai | City_Hydrabad | City_Chennai | City_Bangalore |
|---|---|---|---|---|---|
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 5 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| 8 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |

# Encoding categorical variables

## Dummy Encoding

- Dummy coding scheme is similar to one-hot encoding.
- Transforms the categorical variable into a set of binary variables (also known as dummy variables).
- One-hot encoding, for N categories in a variable, it uses N binary variables.
- The dummy encoding is a small improvement over one-hot-encoding.
- Dummy encoding uses N-1 features to represent N labels/categories.



| Column | Code |
|---|---|
| A | 100 |
| B | 010 |
| C | 001 |

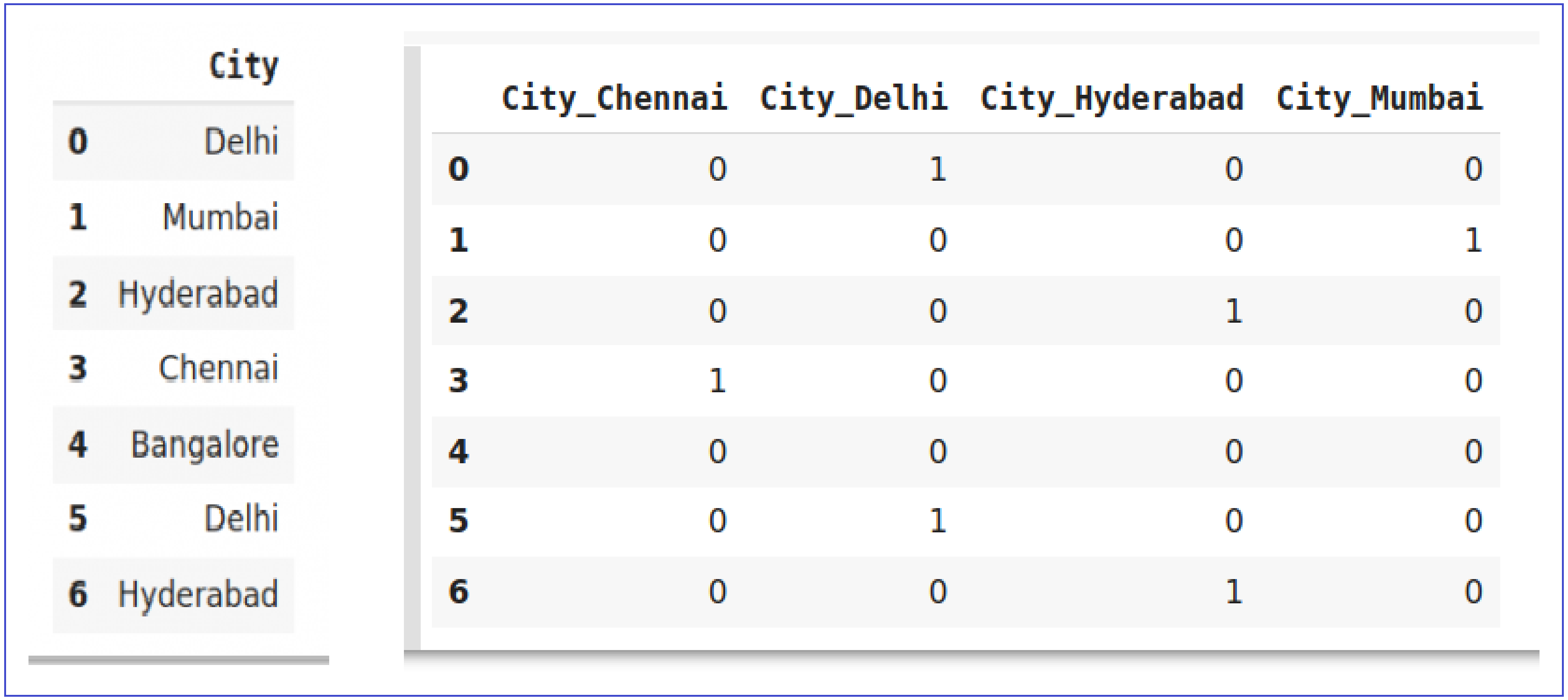
One- Hot Coding

| Column | Code |
|---|---|
| A | 10 |
| B | 01 |
| C | 00 |

Dummy Code

# Encoding categorical variables

## Dummy Encoding Example



| | City |
|---|---|
| **0** | Delhi |
| **1** | Mumbai |
| **2** | Hyderabad |
| **3** | Chennai |
| **4** | Bangalore |
| **5** | Delhi |
| **6** | Hyderabad |

| | City_Chennai | City_Delhi | City_Hyderabad | City_Mumbai |
|---|---|---|---|---|
| **0** | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 1 |
| **2** | 0 | 0 | 1 | 0 |
| **3** | 1 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 |
| **5** | 0 | 1 | 0 | 0 |
| **6** | 0 | 0 | 1 | 0 |

# Encoding categorical variables

## Drawbacks of One-Hot and Dummy Encoding

- One hot encoder and dummy encoder are two powerful and effective encoding schemes.
- Very popular among the data scientists,
- **But may not be as effective when-**
  1. A large number of levels are present in data.

     For example, a column with 30 different values will require 30 new variables for coding.
  2. If we have multiple categorical features in the dataset.

     e.g. a dataset having 10 or more categorical columns.
- **Drawbacks**
  - These two encoding schemes introduce sparsity in the dataset i.e. several columns having 0s and a few of them having 1s
  - They might lead to a **Dummy variable trap**.

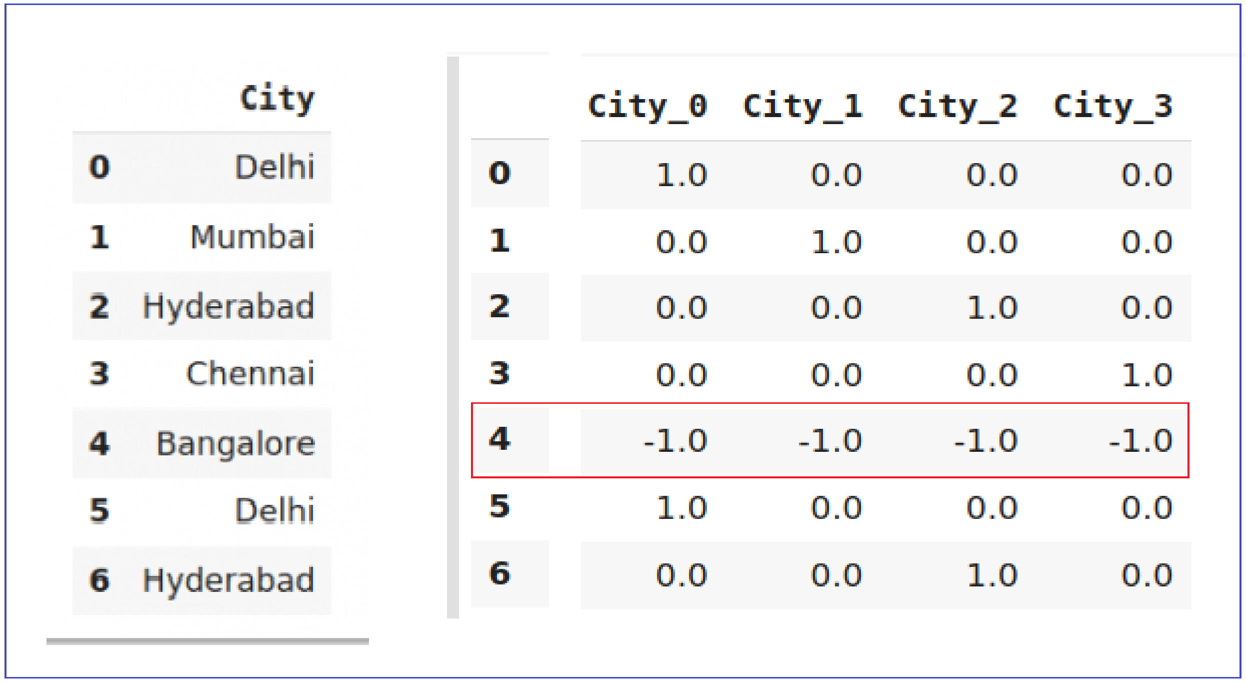
    It is a phenomenon where features are highly correlated.

    That means using the other variables, we can easily predict the value of a variable.

# Encoding categorical variables

**Effect Encoding**:

- ❑ Also known as **Deviation Encoding** or **Sum Encoding.**
- ❑ Effect encoding is almost similar to dummy encoding, with a little difference.
- ❑ In dummy coding, used 0 and 1 to represent the data but in effect encoding, we use three values i.e. 1,0, and -1.
- ❑ Row containing only 0s in dummy encoding is encoded **as -1**.



| | City |
|---|---|
| **0** | Delhi |
| **1** | Mumbai |
| **2** | Hyderabad |
| **3** | Chennai |
| **4** | Bangalore |
| **5** | Delhi |
| **6** | Hyderabad |

| | City_0 | City_1 | City_2 | City_3 |
|---|---|---|---|---|
| **0** | 1.0 | 0.0 | 0.0 | 0.0 |
| **1** | 0.0 | 1.0 | 0.0 | 0.0 |
| **2** | 0.0 | 0.0 | 1.0 | 0.0 |
| **3** | 0.0 | 0.0 | 0.0 | 1.0 |
| **4** | -1.0 | -1.0 | -1.0 | -1.0 |
| **5** | 1.0 | 0.0 | 0.0 | 0.0 |
| **6** | 0.0 | 0.0 | 1.0 | 0.0 |

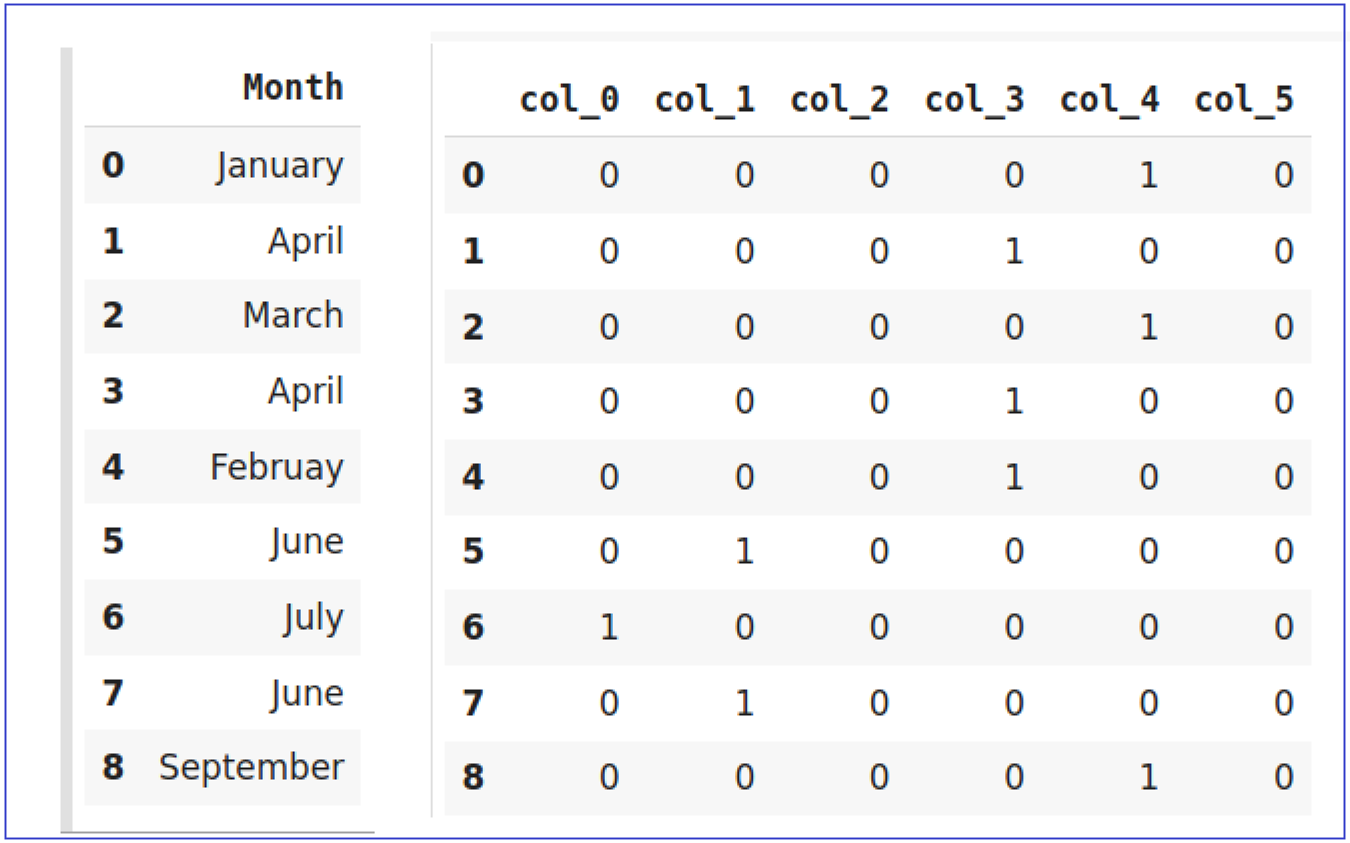
# Encoding categorical variables

## Hash Encoder

- ❑ Hashing is transformation of arbitrary size input in the form of a fixed-size value.
- ❑ Can use hashing algorithms to perform hashing operations i.e. to generate hash value of an input.
- ❑ Further, hashing is a one-way process (can not generate original input from the hash representation)
- ❑ Like one-hot encoding, Hash encoder represents categorical features using new dimensions
- ❑ Can fix the number of dimensions after transformation
- ❑ Can use hash functions like Message Digest (MD, MD2, MD5)

# Encoding categorical variables

## Hash Encoder



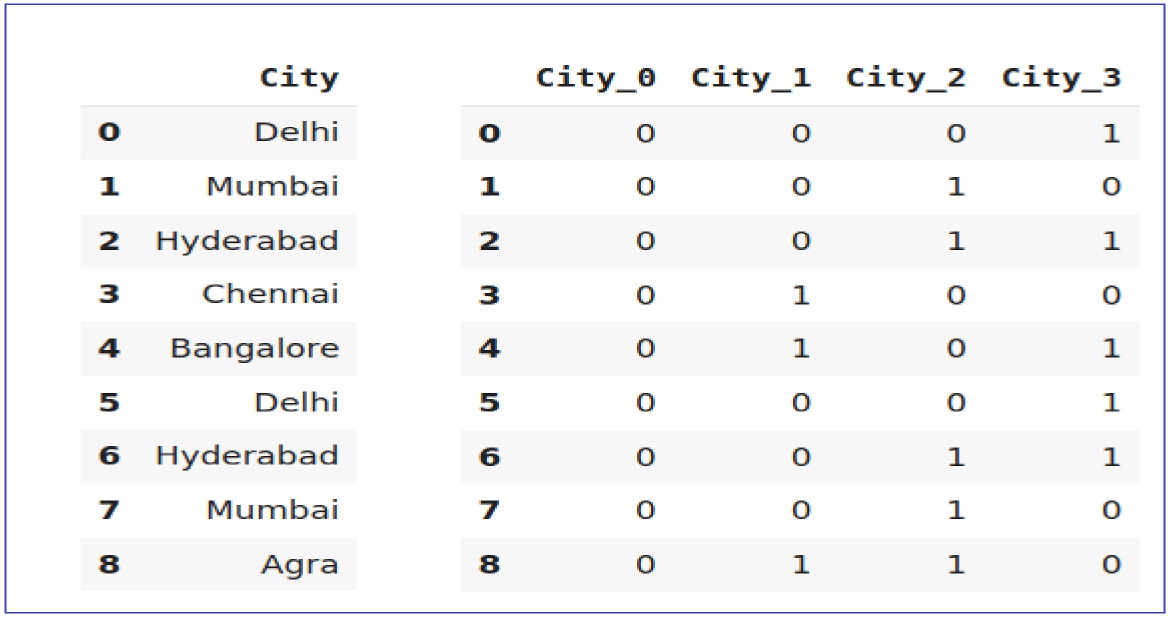
|   | Month |
|---|---|
| **0** | January |
| **1** | April |
| **2** | March |
| **3** | April |
| **4** | Februay |
| **5** | June |
| **6** | July |
| **7** | June |
| **8** | September |

|   | col_0 | col_1 | col_2 | col_3 | col_4 | col_5 |
|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 1 | 0 | 0 |
| **5** | 0 | 1 | 0 | 0 | 0 | 0 |
| **6** | 1 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 1 | 0 | 0 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 1 | 0 |

# Encoding categorical variables

## Binary Encoding

- Binary encoding is a **combination of Hash encoding and one-hot encoding**.
- In this encoding scheme,
  - the categorical feature is first converted into numerical using an ordinal encoder.
  - Then the numbers are transformed in the binary number.
  - After that binary value is split into different columns.
- Binary encoding works well when there are a high number of categories.



| | City |
|---|---|
| **0** | Delhi |
| **1** | Mumbai |
| **2** | Hyderabad |
| **3** | Chennai |
| **4** | Bangalore |
| **5** | Delhi |
| **6** | Hyderabad |
| **7** | Mumbai |
| **8** | Agra |

| | City_0 | City_1 | City_2 | City_3 |
|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 1 |
| **1** | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 1 | 1 |
| **3** | 0 | 1 | 0 | 0 |
| **4** | 0 | 1 | 0 | 1 |
| **5** | 0 | 0 | 0 | 1 |
| **6** | 0 | 0 | 1 | 1 |
| **7** | 0 | 0 | 1 | 0 |
| **8** | 0 | 1 | 1 | 0 |

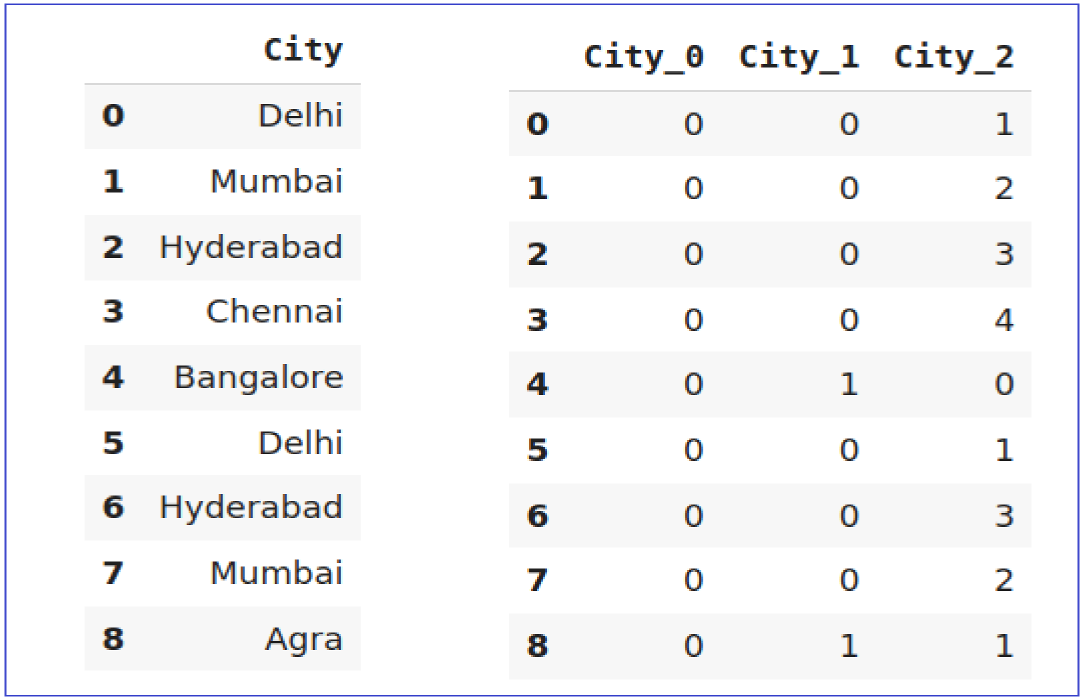
# Encoding categorical variables

## Base N Encoding

- ❑ What is Base here?
  - ➢ In the numeral system, the Base or the radix is the number of digits or a combination of digits and letters used to represent the numbers.
  - ➢ The most common base we use in our life is 10 or decimal system as here we use 10 unique digits i.e. 0 to 9 to represent all the numbers.
  - ➢ Another widely used system is binary i.e. the base is 2.
  - ➢ It uses 0 and 1 i.e 2 digits to express all the numbers.
- ❑ For Binary encoding, the Base is 2 which means it converts the numerical values of a category into its respective Binary form.
- ❑ If you want to change the Base of encoding scheme you may use Base N encoder.
- ❑ In the case when categories are more and binary encoding is not able to handle the dimensionality then we can use a larger base such as 4 or 8.

# Encoding categorical variables

## Base N Encoding



| | City |
|---|---|
| 0 | Delhi |
| 1 | Mumbai |
| 2 | Hyderabad |
| 3 | Chennai |
| 4 | Bangalore |
| 5 | Delhi |
| 6 | Hyderabad |
| 7 | Mumbai |
| 8 | Agra |

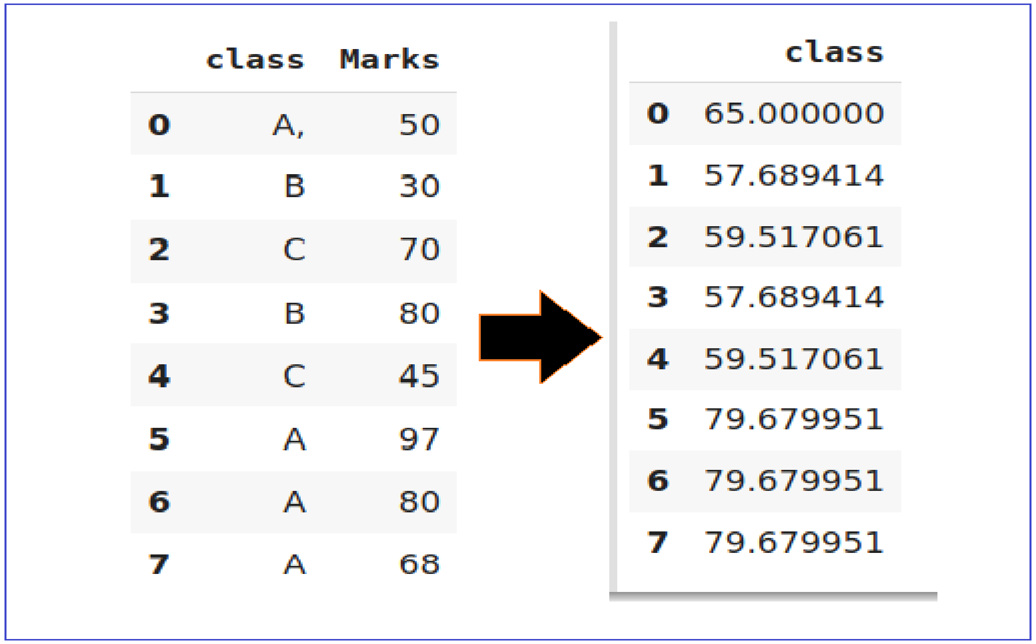
| | City_0 | City_1 | City_2 |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 2 |
| 2 | 0 | 0 | 3 |
| 3 | 0 | 0 | 4 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 |
| 6 | 0 | 0 | 3 |
| 7 | 0 | 0 | 2 |
| 8 | 0 | 1 | 1 |

- ❑ Used base 5 also known as the Quinary system.
- ❑ It is similar to Binary encoding.
- ❑ While Binary encoding represents the same data by 4 new features the BaseN encoding uses only 3 new variables.
- ❑ reduces the number of features required - improving memory usage

# Encoding categorical variables

## Target Encoding

- ❑ Target encoding is a Baysian encoding technique.
- ❑ Bayesian encoders use information from dependent/target variables to encode the categorical data.
- ❑ In target encoding, mean of the target variable is calculated for each category and replace the category variable with the mean value.
- ❑ In the case of the categorical target variables, the posterior probability of the target replaces each category.



| | class | Marks |
|---|---|---|
| 0 | A, | 50 |
| 1 | B | 30 |
| 2 | C | 70 |
| 3 | B | 80 |
| 4 | C | 45 |
| 5 | A | 97 |
| 6 | A | 80 |
| 7 | A | 68 |

| | class |
|---|---|
| 0 | 65.000000 |
| 1 | 57.689414 |
| 2 | 59.517061 |
| 3 | 57.689414 |
| 4 | 59.517061 |
| 5 | 79.679951 |
| 6 | 79.679951 |
| 7 | 79.679951 |

# Encoding categorical variables

## Target Encoding

- ❑ Target encoding performed for train data only and code the test data using results obtained from the training dataset.
- ❑ Although, a very efficient coding system, it has the following **issues** responsible for deteriorating the model performance-
  1. It can lead to target leakage or overfitting. To address overfitting we can use different techniques.
     - I. In the leave one out encoding, the current target value is reduced from the overall mean of the target to avoid leakage.
     - II. In another method, we may introduce some Gaussian noise in the target statistics. The value of this noise is hyperparameter to the model.
  2. The second issue, we may face is the improper distribution of categories in train and test data.

## Encoding categorical variables

**To summarize,**

- Encoding categorical data is an unavoidable part of the feature engineering.
- It is more important to know what coding scheme should we use.
- Having into consideration the dataset we are working with and the model we are going to use.

## Bucketing continuous features into categories

- ❑ The problem of working with raw, continuous numeric features is that
  - ➢ Distribution of values in these features may be skewed.
  - ➢ This signifies that some values will occur quite frequently while some will be quite rare.
  - ➢ Besides this, there is also another problem of the varying range of values in any of these features.
- ❑ Directly using these features can cause a lot of issues and adversely affect the model.
- ❑ Hence there are strategies to deal with this, which include binning and transformations.

## Bucketing continuous features into categories

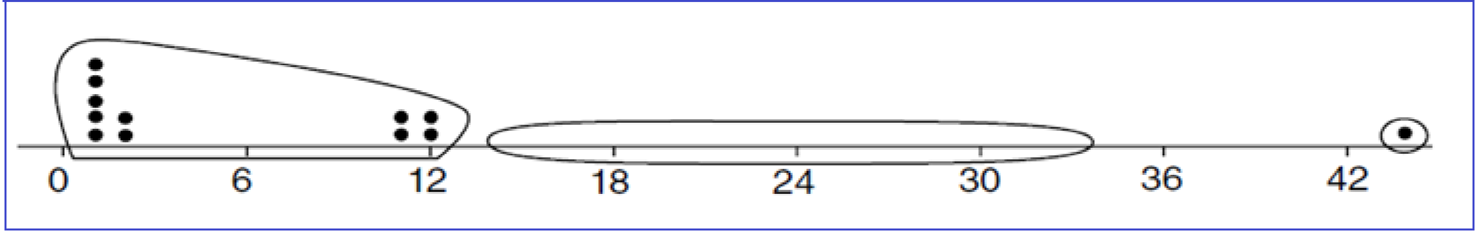
- ❑ Binning, also known as quantization is used for transforming continuous numeric features into discrete ones (categories).
- ❑ These discrete values or numbers can be thought of as categories or bins into which the raw, continuous numeric values are binned or grouped into.
- ❑ Each bin represents a specific degree of intensity and hence a specific range of continuous numeric values fall into it.
- ❑ Specific strategies of binning data include fixed-width and adaptive binning.

# Bucketing continuous features into categories

## Fixed-Width Binning

- ❑ Have specific fixed widths for each of the bins
- ❑ pre-defined by the user analyzing the data.
- ❑ Each bin has a pre-fixed range of values assigned to that bin on the basis of
  - ➢ some domain knowledge,
  - ➢ rules or
  - ➢ constraints.

Data set - $X = \{1, 1, 1, 1, 1, 2, 2, 11, 11, 12, 12, 44\}$ and k=3



## Bucketing continuous features into categories

### Adaptive Binning

- ❑ Drawback of fixed-width binning
  - ➢ irregular bins - not uniform
  - ➢ Some of the bins might be densely populated and some of them might be sparsely populated or even empty!
- ❑ Adaptive binning is a safer strategy in these scenarios where let the data speak for itself!
- ❑ Use the data distribution itself to decide bin ranges.

# Bucketing continuous features into categories

## Adaptive Binning

- ❑ Quantile based binning is a good strategy to use for adaptive binning.
- ❑ Quantiles are specific values or cut-points which help in partitioning the continuous valued distribution of a specific numeric field into discrete contiguous bins or intervals.
- ❑ Thus, q-Quantiles help in partitioning a numeric attribute into *q* equal partitions.
- ❑ Popular examples of quantiles include the
    - ❑ 2-Quantile known as the median which divides data distribution into **two equal** bins,
    - ❑ 4-Quantiles known as the *quartiles* which divide the data into **4 equal** bins and
    - ❑ *10-Quantiles* also known as the *deciles* which create **10 equal** width bins.

## Bucketing continuous features into categories

**Other common methods**

1. *Equal frequency binning*
2. *Binning by clustering*
3. *Binning based on predictive value*

# Extending numerical features

- ❑ Transforms like raising input variables to a power can help to better expose the important relationships between input variables and the target variable.
- ❑ These features are called interaction and polynomial features
- ❑ Sometimes these features can result in improved modeling performance
- ❑ Some machine learning algorithms prefer or perform better with polynomial input features.
- ❑ Can use polynomial features transform to create new versions of input variables for predictive modeling
- ❑ Degree of the polynomial impacts the number of input features created by the transform.

# Extending numerical features

## Polynomial Features

- ❑ Created by raising existing features to an exponent.
- ❑ For example, if a dataset had one input feature X, then a polynomial feature would be the addition of a new feature (column) where values were calculated by squaring the values in X, e.g. X^2.
- ❑ This process can be repeated for each input variable in the dataset, creating a transformed version of each.
- ❑ Polynomial features are a type of feature engineering, e.g. the creation of new input features based on the existing features.
- ❑ The "*degree*" of the polynomial is used to control the number of features added, e.g. a degree of 3 will add two new variables for each input variable.
- ❑ Typically a small degree is used such as 2 or 3.

## Extending numerical features

### Polynomial Features

- ❑ Common to add new variables that represent the interaction between features
- ❑ e.g. a new column that represents one variable multiplied by another.
- ❑ Can be repeated for each input variable creating a new "*interaction*" variable for each pair of input variables.
- ❑ A squared or cubed version of an input variable will change the probability distribution, separating the small and large values.
- ❑ Separation that is increased with the size of the exponent.
- ❑ This separation can help some machine learning algorithms make better predictions.
- ❑ Useful for regression predictive modeling tasks such as linear regression and logistic regression.

# Text-specific feature construction

- ❑ Have been working with categorical and numerical data
- ❑ Text data is much more complex than single—category text
- ❑ Text data example - service like Yelp – users write reviews in text format
- ❑ Reviews useful for machine learning purposes, for example, in predicting best restaurant to visit
- ❑ Large part of communicate is through written text
- ❑ Much can be garnered from this information through modeling
- ❑ For example, we can conduct a sentiment analysis from Twitter data.
- ❑ This type of work can be referred to as **natural language processing** (**NLP**).
- ❑ Computers can be programmed to process natural language text.

## Text-specific feature construction

### Bag of words representation

- All machine learning models require numerical inputs
- Need to convert text data into numerical features
- Common method to transform a corpus into a numerical representation, a process known as vectorization, is through a method called **bag-of-words.**
- Documents are described by word occurrences while completely ignoring positioning of words in the document
- Text is represented as a **bag**, without regard for grammar or word order, and is maintained as a set
- A bag of words representation is achieved in the following three steps:
    1. Tokenizing
    2. Counting
    3. Normalizing

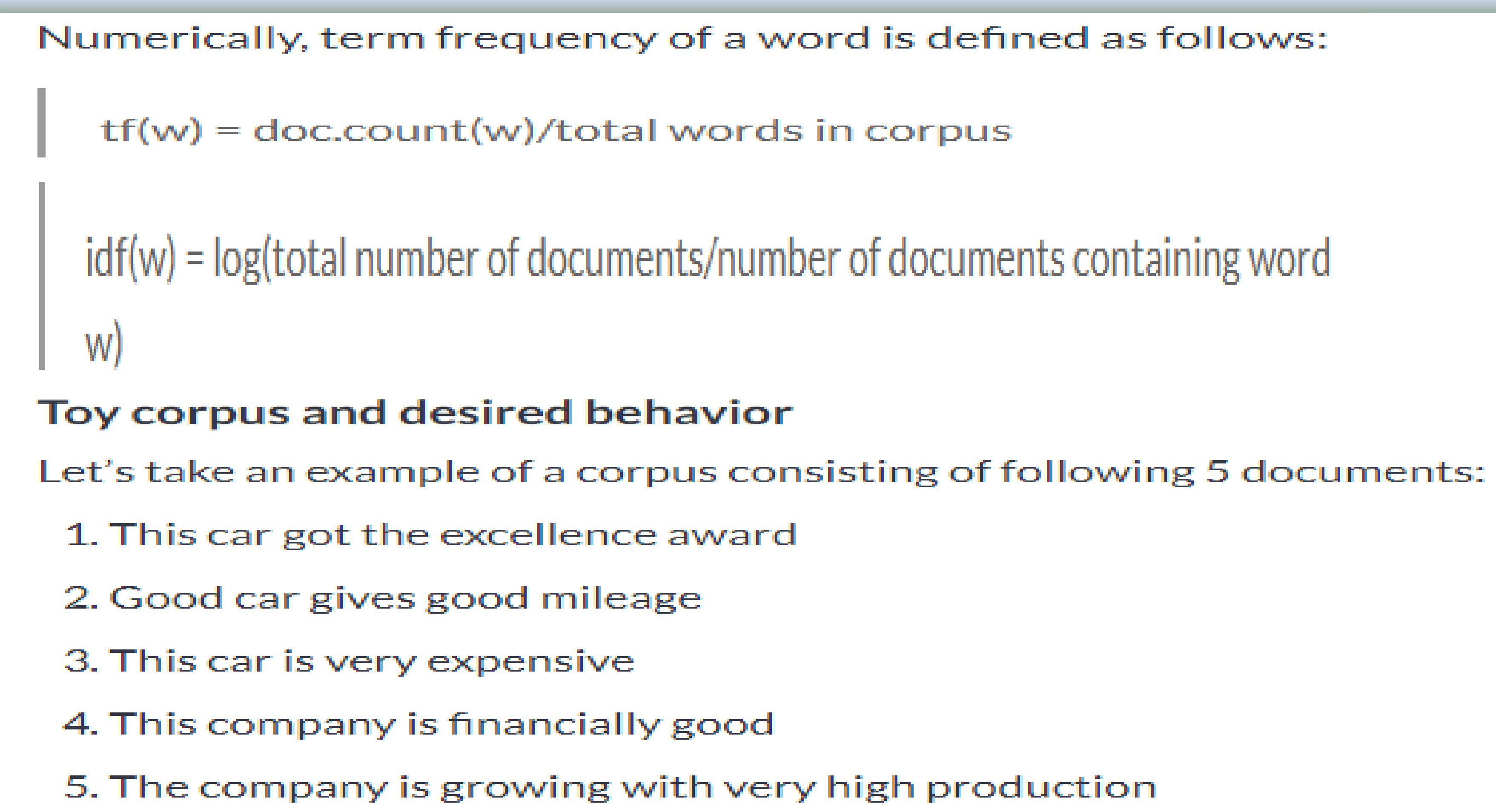
# DOCUMENT REPRESENTATION

- The methods of giving weights to the features may vary.
- The simplest is the binary in which the feature weight is either zero OR one.
- More complex weighting schemes are possible that take into account the frequencies of the word in the document, in the category, and in the whole collection.
- The most common **TF-IDF scheme** gives the word w in the document d the weight

$$\mathit{TF\text{-}IDF\_Weight}(w, d) = \mathit{TermFreq}(w, d) \cdot \log(N / \mathit{DocFreq}(w)),$$

where TermFreq(w, d) is the frequency of the word in the document,
N is the number of all documents, and
DocFreq(w) is number of documents containing word w.

# DOCUMENT REPRESENTATION



Numerically, term frequency of a word is defined as follows:

> tf(w) = doc.count(w)/total words in corpus

> idf(w) = log(total number of documents/number of documents containing word w)

**Toy corpus and desired behavior**

Let's take an example of a corpus consisting of following 5 documents:

1. This car got the excellence award
2. Good car gives good mileage
3. This car is very expensive
4. This company is financially good
5. The company is growing with very high production

**Tf-idf(car) for D1 = 1/16 * log(5/3) = 0.0625 * 0.2218 = 0.01386 - Normalized TF**
**Tf-idf(car) for D1 = 1 * log(5/3) = 0.4771 - Non-Normalized TF**

**Assume total Number of Words in Corpus = 16**

# DOCUMENT REPRESENTATION

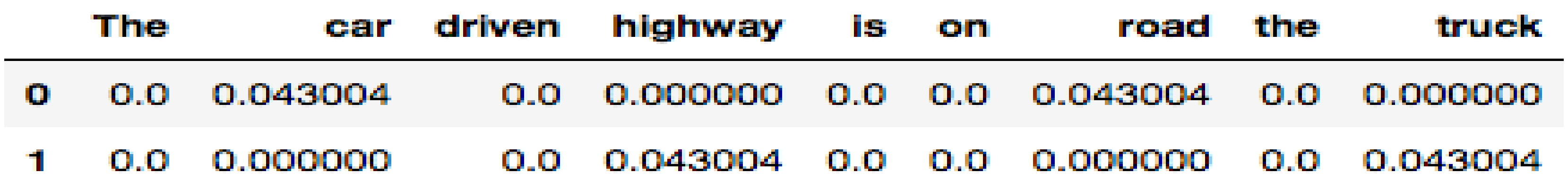
- **Let’s another take an example to get a clearer understanding.**

  **Sentence 1 :** The car is driven on the road.

  **Sentence 2 :** The truck is driven on the highway.

- In this example, each sentence is a separate document.

| Word | TF | | IDF | TF*IDF | |
|---|---|---|---|---|---|
| | A | B | | A | B |
| The | 1/7 | 1/7 | log(2/2) = 0 | 0 | 0 |
| Car | 1/7 | 0 | log(2/1) = 0.3 | 0.043 | 0 |
| Truck | 0 | 1/7 | log(2/1) = 0.3 | 0 | 0.043 |
| Is | 1/7 | 1/7 | log(2/2) = 0 | 0 | 0 |
| Driven | 1/7 | 1/7 | log(2/2) = 0 | 0 | 0 |
| On | 1/7 | 1/7 | log(2/2) = 0 | 0 | 0 |
| The | 1/7 | 1/7 | log(2/2) = 0 | 0 | 0 |
| Road | 1/7 | 0 | log(2/1) = 0.3 | 0.043 | 0 |
| Highway | 0 | 1/7 | log(2/1) = 0.3 | 0 | 0.043 |



| | The | car | driven | highway | is | on | road | the | truck |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 0.043004 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.043004 | 0.0 | 0.000000 |
| 1 | 0.0 | 0.000000 | 0.0 | 0.043004 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.043004 |

***Thank You !!!***