BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Data Science

Data wrangling and Feature Engineering

Categorical Encoding

Dr. Ramakrishna Dantu

Associate Professor, BITS Pilani

Disclaimer and Acknowledgement

- The content for these slides has been obtained from books and various other source on the Internet
- I here by acknowledge all the contributors for their material and inputs.
- I have provided source information wherever necessary
- I have added and modified the content to suit the requirements of the course

Data wrangling and Feature Engineering – Part-1

- Data cleaning
- Data Aggregation, Sampling,
- Handling Numeric Data
  - Discretization, Binarization
  - Normalization
  - Data Smoothening
- Dealing with textual Data
- Managing Categorical Attributes
  - Transforming Categorical to Numerical Values
  - Encoding techniques
- Feature Engineering
  - Feature Extraction (Dimensionality Reduction)
  - Feature Construction
  - Feature Subset selection
    - Filter methods
    - Wrapper methods
    - Embedded methods
  - Feature Learning
- Case Study involving FE tasks

Categorical Encoding

Categorical Encoding

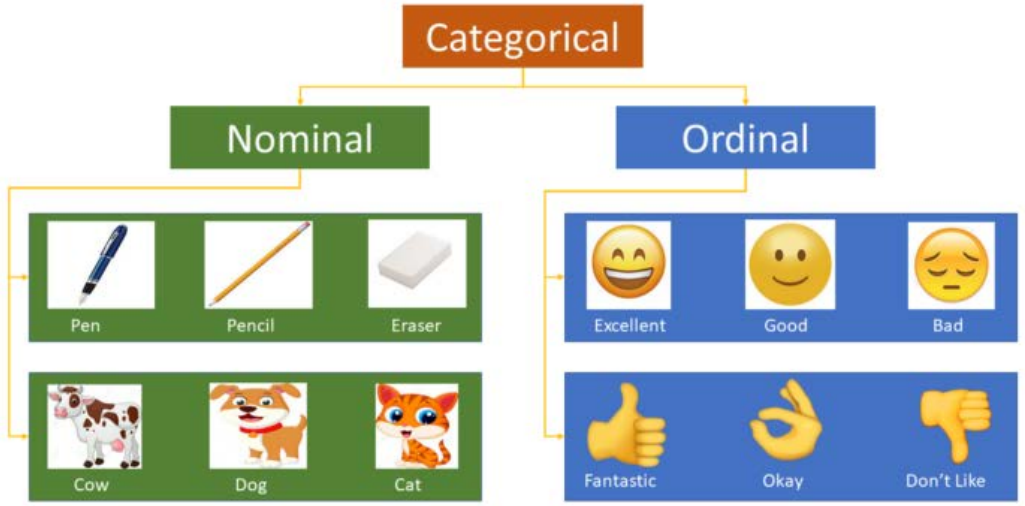
Learning objectives

- Articulate the need for categorical encoding
- List and define various data encoding methods
- Make the calculations that are necessary to get meaningful encodings

Categorical Encoding

Categorical Data

- Categorical variables can be divided into two categories:
  - Nominal (No particular order) and
  - Ordinal (some ordered)
- Nominal Variables
  - Red, Yellow, Pink, Blue
  - Singapore, Japan, USA, India, Korea
  - Cow, Dog, Cat, Snake
- Ordinal Variables
  - High, Medium, Low
  - "Strongly agree", Agree, Neutral, Disagree, and "Strongly Disagree"
  - Excellent, Okay, Bad



Source: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

Categorical Encoding

Overview

- Many machine learning algorithms cannot use non-numeric data
- These non-numeric attributes (features) are represented by strings
- We need to transforming them into numbers before using machine learning algorithms
- Categorical encoding refers to replacing the category strings by a numerical value
- The performance of many algorithm varies based on how the categorical variable is encoded
- The goal of categorical encoding is:
  - To build predictive features from categories
  - To produce variables that can be used to train machine learning models

Categorical Encoding

Different Types of Categorical Encoding

- One Hot Encoding
- Label Encoding
- Ordinal Encoding
- Helmert Encoding
- Binary Encoding
- Frequency Encoding
- Mean Encoding
- Weight of Evidence Encoding
- Probability Ratio Encoding
- Hashing Encoding
- Backward Difference Encoding
- Leave One Out Encoding
- James-Stein Encoding
- M-estimator Encoding

Source: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

Categorical Encoding

One Hot Encoding

- The most common way to represent categorical variables is using the one-hot encoding or one-out-of-N encoding, also known as dummy variables
- One hot encoding consists of encoding each categorical variable with a set of boolean variables, which take values 0 or 1
- One hot encoding indicates if a category is present for each observation

Status		Student	Unemployed	Employee	Retired
Student		1	0	0	0
Unemployed		0	1	0	0
Employee		0	0	1	0
Retired		0	0	0	1

Categorical Encoding

One Hot Encoding

- Using "k-1" fields
  - More generally, a categorical variable should be encoded by creating k-1 binary variables, where k is the number of distinct categories.
    - In the case of binary variables, like gender where k=2 (male / female) we need to create only 1 (k - 1 = 1) binary variable
  - One hot encoding into k-1 binary variables takes into account that we can use 1 less dimension and still represent the whole information:
    - If the observation is 0 in all the binary variables, then it must be 1 in the final (not present) binary variable.
- Encoding categorical variables into k - 1 binary variables, is better, as it avoids introducing redundant information

Status		Student	Unemployed	Employee
Student		1	0	0
Unemployed		0	1	0
Employee		0	0	1
Retired		0	0	0

Categorical Encoding

One Hot Encoding – Avoid the dummy trap!!

- Using "k-1" fields
  - When defining dummy variables, a common mistake is to define too many variables
  - If a categorical variable can take on k values, it is tempting to define k dummy variables
  - A kth dummy variable is redundant; it carries no new information
  - And it creates a severe multicollinearity problem for the analysis
    - Multicollinearity occurs when independent variables in a regression model are correlated
- If the degree of correlation between variables is high, it can cause problems when you fit the model and interpret the results.
- Using k dummy variables when only k - 1 dummy variables are required is known as the dummy variable trap. Avoid this trap!

Status		Student	Unemployed	Employee
Student		1	0	0
Unemployed		0	1	0
Employee		0	0	1
Retired		0	0	0

Source: https://stattrek.com/multiple-regression/dummy-variables.aspx

Categorical Encoding

One Hot Encoding

- There are a few occasions when the variable is encoded using k dummy variables:
  - when building tree based algorithms
  - when doing feature selection by recursive algorithms
  - when interested in determine the importance of each single category

workclass	Government Employee	Private Employee	Self Employed	Self Employed Incorporated
Government Employee	1	0	0	0
Private Employee	0	1	0	0
Self Employed	0	0	1	0
Self Employed Incorporated	0	0	0	1

Categorical Encoding

One Hot Encoding

- Advantages
  - The result is binary (rather than ordinal) and everything sits in an orthogonal vector space
  - Makes no assumption about the distribution or categories of the variable
  - Keeps all the information of the categorical variable
  - Suitable for linear models
- Disadvantages
  - High cardinality (many categories), expands the feature space and we will start fighting with the curse of dimensionality
  - Does not add extra information while encoding
  - Many dummy variables may be identical, introducing redundant information

Categorical Encoding

Label / Integer Encoding

- Label encoding consists in replacing the categories by digits from 1 to n (or 0 to n-1, depending the implementation)
  - Where n is the number of distinct categories of the variable.
- The numbers are assigned arbitrarily
- This encoding method allows for quick benchmarking of machine learning models

Status		Status
Student		1
Unemployed		2
Employee		3
Retired		4

Categorical Encoding

Label / Integers Encoding

- One issue with this approach is there is no relation or order between the classes
  - but the algorithm might consider them as having some order or relationship
  - For example, it may look like (Cold < Hot < Very Hot < Warm….0 < 1 < 2 < 3)
- Advantages
  - Straightforward to implement
  - Does not expand the feature space
  - Can work well enough with tree based algorithms
- Limitations
  - Does not add extra information while encoding
  - Not suitable for linear models
  - Does not handle new categories in test set automatically

Categorical Encoding

Frequency/Count Encoding

- Categories are replaced by the count or percentage of observations that show that category in the dataset
- Captures the representation of each label in a dataset
- Very popular encoding method in Kaggle competitions
- Assumption:
  - the number observations shown by each category is predictive of the outcome

	Count Encoding	Freq Encoding
Status	**Status**	**Status**
Student	3	3/8 = 0.375
Unemployed	1	1/8 = 0.125
Student	3	3/8 = 0.375
Retired	2	2/8 = 0.25
Retired	2	2/8 = 0.25
Employee	2	2/8 = 0.25
Student	3	3/8 = 0.375
Employee	2	2/8 = 0.25

Categorical Encoding

Frequency/Count Encoding

- Advantages
  - Straightforward to implement
  - Does not expand the feature space
  - Can work well enough with tree based algorithms
- Limitations
  - Not suitable for linear models
  - Does not handle new categories in test set automatically
  - If 2 different categories appear the same number of times in the dataset, they will be replaced by the same number:
    - may lose valuable information.

	Count Encoding	Freq Encoding
Status	Status	Status
Student	3	3/8 = 0.375
Unemployed	1	1/8 = 0.125
Student	3	3/8 = 0.375
Retired	2	2/8 = 0.25
Retired	2	2/8 = 0.25
Employee	2	2/8 = 0.25
Student	3	3/8 = 0.375
Employee	2	2/8 = 0.25

Categorical Encoding

Ordinal Encoding

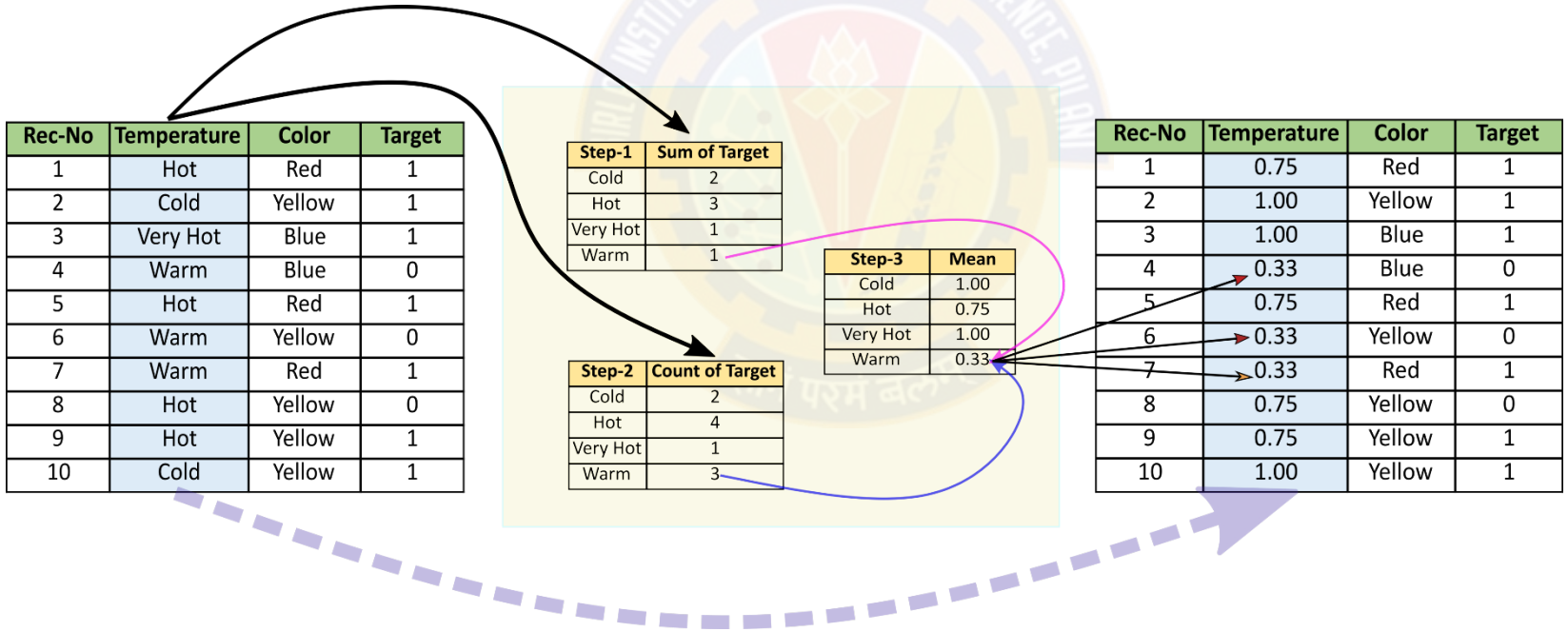
- To encode a categorical feature that has a natural order, such as a movie rating (e.g., "bad," "average," "good"), the simplest option is to use ordinal encoding:
  - sort the categories in their natural order and map each category to its rank
  - e.g., "bad" maps to 0, "average" maps to 1, and "good" maps to 2

Temperature		Status
Very hot		0
Hot		1
Warm		2
Cold		3
Freezing		4

Categorical Encoding

Mean/Target Encoding

- Mean Encoding takes the number of labels into account along with the target variable to encode the labels



Rec-No	Temperature	Color	Target
1	Hot	Red	1
2	Cold	Yellow	1
3	Very Hot	Blue	1
4	Warm	Blue	0
5	Hot	Red	1
6	Warm	Yellow	0
7	Warm	Red	1
8	Hot	Yellow	0
9	Hot	Yellow	1
10	Cold	Yellow	1

Step-1	Sum of Target
Cold	2
Hot	3
Very Hot	1
Warm	1

Step-2	Count of Target
Cold	2
Hot	4
Very Hot	1
Warm	3

Step-3	Mean
Cold	1.00
Hot	0.75
Very Hot	1.00
Warm	0.33

Rec-No	Temperature	Color	Target
1	0.75	Red	1
2	1.00	Yellow	1
3	1.00	Blue	1
4	0.33	Blue	0
5	0.75	Red	1
6	0.33	Yellow	0
7	0.33	Red	1
8	0.75	Yellow	0
9	0.75	Yellow	1
10	1.00	Yellow	1

Source: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

Categorical Encoding

Mean/Target Encoding

- Mean Encoding takes the number of labels into account along with the target variable to encode the labels

Job	Age	Target
Data Scientist	45	1
Data Scientist	40	0
Data Analyst	32	1
Data Engineer	35	1
Data Scientist	42	1
Data Analyst	33	1
Data Architect	44	1
Data Architect	50	0
Data Scientist	50	1
Data Engineer	36	1
Data Analyst	36	0

Job	Target Count
Data Scientist	4
Data Analyst	3
Data Engineer	2
Data Architect	2

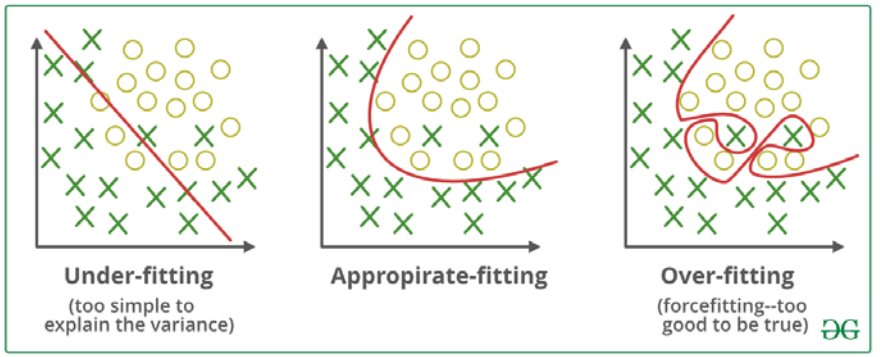
Job	Target Sum
Data Scientist	3
Data Analyst	2
Data Engineer	2
Data Architect	1

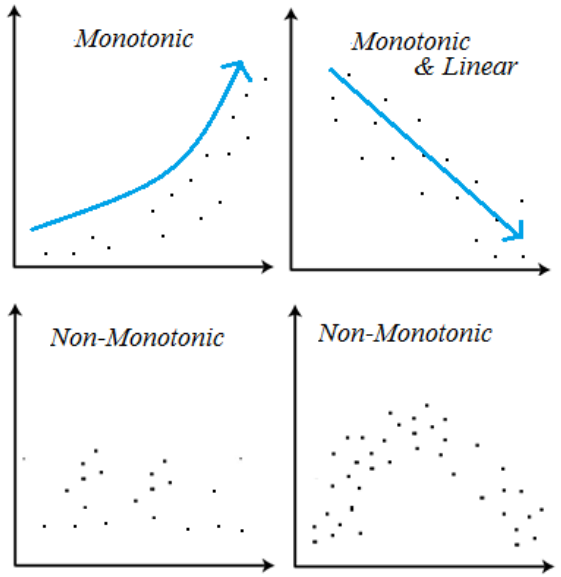
Job	Mean
Data Scientist	3/4 = 0.75
Data Analyst	2/3 = 0.67
Data Engineer	2/2 = 1.00
Data Architect	1/2 = 0.50

Categorical Encoding

Mean/Target Encoding

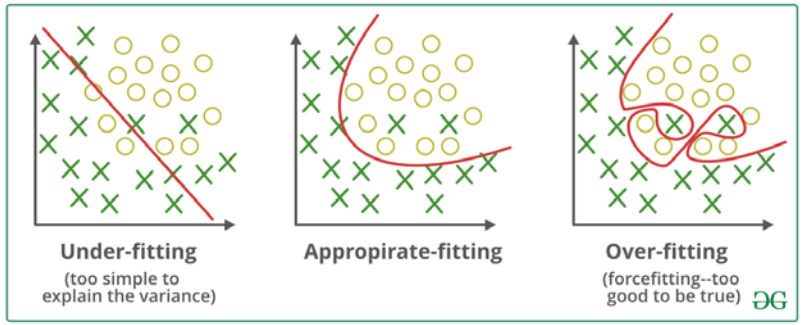
- Advantages
  - Increases the quality of a classification
  - Straightforward to implement
  - Does not expand the feature space
  - Creates monotonic relationship between categories and target
- Disadvantages
  - May lead to over-fitting
  - Difficult to implement together with cross-validation with current libraries
  - If 2 categories show the same mean of target, they will be replaced by the same number => potential loss of value
  - The fact that we are encoding the feature based on target classes may lead to data leakage, rendering the feature biased



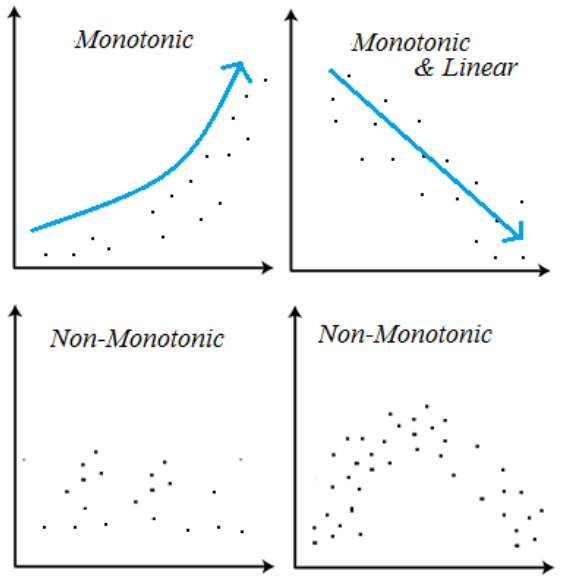


Categorical Encoding

Mean/Target Encoding



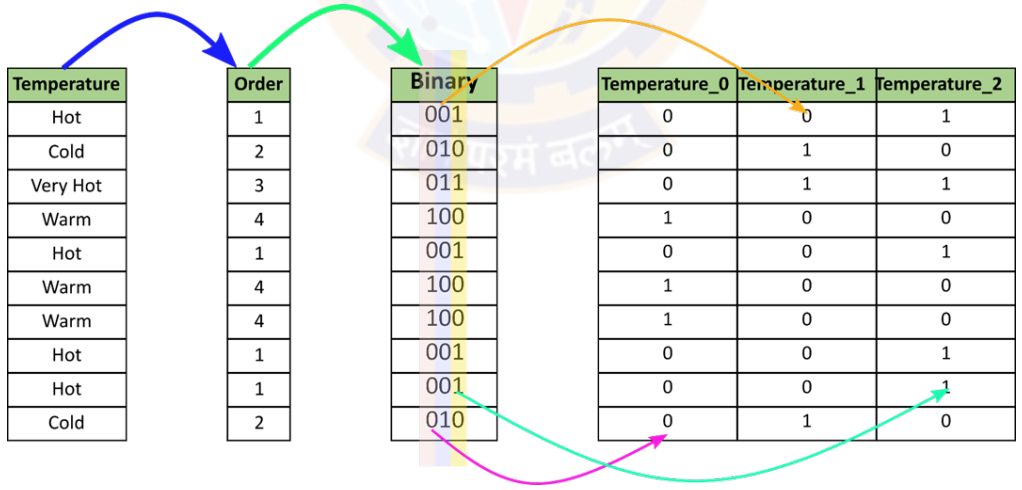
- Note:
  - Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample
  - Data leakage occurs when information from outside the training dataset is used to create the model
  - Data leakage can cause us to create overly optimistic if not completely invalid predictive models
  - Monotonic variables increase (or decrease) in the same direction, but not always at the same rate
  - Linear variables increase (or decrease) in the same direction at the same rate



Categorical Encoding

Binary Encoding

- Binary encoding can be thought of as a hybrid of one-hot and hashing encoders
- Here's how it works:
  - The categories are encoded as per Ordinal Encoding
  - Then those integers are converted into binary code, so for example 5 becomes 101 and 10 becomes 1010
  - Then the digits from that binary string are split into separate columns
  - So if there are 4–7 values in an ordinal column then 3 new columns are created:
    - one for the first bit, one for the second, and one for the third.



Temperature
Hot
Cold
Very Hot
Warm
Hot
Warm
Warm
Hot
Hot
Cold

Order
1
2
3
4
1
4
4
1
1
2

Binary
001
010
011
100
001
100
100
001
001
010

Temperature_0	Temperature_1	Temperature_2
0	0	1
0	1	0
0	1	1
1	0	0
0	0	1
1	0	0
1	0	0
0	0	1
0	0	1
0	1	0

Categorical Encoding

Binary Encoding

- Advantages
  - Straightforward to implement.
  - Does not expand the feature space too much.
  - Binary creates fewer features than one-hot
    - It is more memory efficient
  - Preserves some uniqueness of values in the column
  - Binary really shines when the cardinality of the column is higher
    - E.g., 50 US states
- Disadvantages
  - It exposes the loss of information during encoding.
  - It lacks the human-readable sense
  - With only three levels, the information embedded becomes muddled
  - There are many collisions and the model can’t glean much information from the features

Categorical Encoding

Feature Hashing

- The central part of the hashing encoder is the hash function
- Feature hashing, also known as the hashing trick, is a fast and space-efficient way of vectorizing features
  - i.e. turning arbitrary features into indices in a vector or matrix
- It works by applying a hash function to the features and using their hash values as indices directly
- Feature hashing maps each category in a categorical feature to an integer within a pre-determined range

Source: Dealing with categorical features with high cardinality: Feature Hashing

Categorical Encoding

Feature Hashing

- This technique is applied typically in text mining
- The technique involves converting data into a vector of features
- When this is done using hashing, we call the method "feature hashing" or "the hashing trick"
- Let's say our text is: "Taking my dog for a walk is fun."
- We would like to represent this as a vector
- The first thing we need is to fix the length of the vector (the number of dimensions) we are going to use, let's say we would like to use 5 dimensions.
- Once we fix the number of dimensions we need a hash function that will take a string as input and returns a number between 0 and n-1, in our case between 0 and 4
- Any good hash function can be used and you just use h(string) mod n to make it return a number between 0 and n-1

Categorical Encoding

Feature Hashing

- Once we apply the hash function, we can simply construct our vector as: (2,0,3,3,0,2,2,1)

Feature	Murmurhash3	Divisor	Remainder
Taking	642267302	5	2
my	1357816490	5	0
dog	2982218203	5	3
for	2982218203	5	3
a	1009084850	5	0
walk	242017537	5	2
is	2021799277	5	2
fun	2537323311	5	1

	0	1	2	3	4
Taking	0	0	1	0	0
my	1	0	0	0	0
dog	0	0	0	1	0
for	0	0	0	1	0
a	1	0	0	0	0
walk	0	0	1	0	0
is	0	0	1	0	0
fun	0	1	0	0	0

Categorical Encoding

Feature Hashing

- In document classification task, the input to the machine learning algorithm (both during learning and classification) is free text
- From this text, a bag of words (BOW) representation is constructed
  - The individual tokens are extracted and counted, and each distinct token in the training set defines a feature (independent variable) of each of the documents in both the training and test sets.
- Therefore, the bags of words for a set of documents is regarded as a term-document matrix where each row is a single document, and each column is a single feature/word
  - The entry i, j in such a matrix captures the frequency (or weight) of the j'th term of the vocabulary in document i
- Typically, these vectors are extremely sparse
- The common approach is to construct a dictionary representation of the vocabulary of the training set, and use that to map words to indices

Source: Wikipedia – Feature Hashing

Categorical Encoding

Feature Hashing

- Hash tables are common candidates for dictionary implementation
- E.g., the three documents
  - John likes to watch movies.
  - Mary likes movies too.
  - John also likes football.
- can be converted, using the dictionary

Feature	Murmurhash3	Divisor	Rem
John	4006378949	8	5
likes	1103617568	8	0
to	152217691	8	3
watch	926942048	8	0
movies	3188341541	8	5
Mary	435663154	8	2
too	4227687170	8	2
also	1706587157	8	5
football	1708191722	8	2

Index	Value
0	likes watch
1	
2	Mary too football
3	to
4	
5	John movies
6	
7	

Feature	1	2	3	4	5	6	7	8
Index	0	1	2	3	4	5	6	7
Values	likes		Mary			John		

Source: Wikipedia – Feature Hashing

Categorical Encoding

Feature Hashing

- The three documents
  - John likes to watch movies.
  - Mary likes movies too.
  - John also likes football.

Feature	Murmurhash3	Divisor	Rem
John	4006378949	8	5
likes	1103617568	8	0
to	152217691	8	3
watch	926942048	8	0
movies	3188341541	8	5
Mary	435663154	8	2
too	4227687170	8	2
also	1706587157	8	5
football	1708191722	8	2

Index	Value
0	likes watch
1	
2	Mary too football
3	to
4	
5	John movies also
6	
7	

Feature	1	2	3	4	5	6	7	8
Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Doc-1	2	0	0	1	0	2	0	0
Doc-2	1	0	2	0	0	1	0	0
Doc-3	1	0	1	0	0	2	0	0

Source: Wikipedia – Feature Hashing

Categorical Encoding

Feature Hashing

- Advantages
  - The number of dimensions will be far less than the number of dimensions with encoding like One Hot Encoding.
    - Even if we have over 1000 distinct categories in a feature and we set b=10 as the final feature vector size (a pre-determined range), the output feature set will still have only 10 features as compared to 1000 binary features if we used a one-hot encoding scheme
- Disadvantages
  - Hash functions can hash different keys to the same integer value (this is known as 'collision')
    - It will certainly happen in this case as we had represented 1000 distinct categories as 10 columns
- Note
  - You have to do a trade-off between the No. of categories getting mapped to the same integer value (% of collision) and the final feature vector size (b i.e. a pre-determined range)
  - Here, b is nothing but n_components

Source: Dealing with categorical features with high cardinality: Feature Hashing

Thank You!