BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Data Science

Data wrangling and Feature Engineering

Feature Engineering

Sankara Nayaki K
sankaranayaki@wilp.bits-pilani.ac.in

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

Course Handout - Modules

- Introduction to Data Science
- Data Analytics
- Data Science Process
- Data Science Teams
- Data and Data Models
- Data wrangling and Feature Engineering
- Data visualization
- Storytelling with Data
- Ethics for Data Science

Table of Contents

Feature Engineering - Case Study 1

Real Estate price prediction

- Problem Definition: Given the characteristics in terms of landmark, facilities, complaints etc. of property predict the sale price.
- Data Set Description: The dataset has 108 Features and 6664 instances. The dependent variable is the sale price.

Name of the person	Address
Pincode	Area code
Building area	Common Area
Type of flooring	Distance to school
Distance to bus depot	Crime rate

Feature Engineering - Case Study 2

Customer Retention

- Problem Definition: Given a data set of customer purchasing data gathered over a year, predict which customers will return to make a purchase.
- Data Set Description: The set contains around 350 million rows and 11 features.

Customer ID	Store ID
Department ID	Product Category
Manufacturer	Band of Product
Date of Purchase	Product Measure
Purchase Quantity	Product Price

Feature Engineering - Case Study 3

Medical diagnosis

- Problem Definition: Given a data set of cancer patients and the tests they have undergone to predict whether a patient will have cancer or not.
- Data Set Description: The set contains around 100 million rows and 35 features.

Patient ID	Name
Age	Gender
Profession	Marital Status
Endoscopy	Diagnostic Imaging
Blood Tests	Biopsy

Feature

- Feature is a property of an object under study.
- Features are the basic building blocks of datasets.

Building Area	Common Area	Type of flooring	Distance from bus deport	Sales price per sq.ft
1134	350	Marble	16503.22	6715
2000	325	Vitrified Tiles	16321.19	3230
2544	950	Vitrified Tiles	15619.92	6588

Feature Engineering

- Feature Engineering is the process of selecting and extracting useful, predictive signals from data.
- The goal is to create a set of features that best represent the information contained in the data, producing a simpler model that generalizes well to future observations.

Table of Contents

Why Feature Scaling?

- Features with bigger magnitude dominate over the features with smaller magnitudes.
- Good practice to have all variables within a similar scale.
- Euclidean distances are sensitive to feature magnitude.
- Gradient descent converges faster when all the variables are in the similar scale.
- Feature scaling helps decrease the time of finding support vectors.

Why Feature Scaling?

- For distance-based methods, normalization helps prevent attributes with initially large ranges (e.g., income) from out-weighing attributes with initially smaller ranges (e.g., binary attributes).

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesF	Age	Outcome
6	148	72	35	0	34	0.627	50	1
1	85	66	29	0	27	0.351	31	0
8	183	64	0	0	23	0.672	32	1
1	89	66	23	94	28	0.167	21	0
0	137	40	35	168	43	2.288	33	1
5	116	74	0	0	26	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35	0.134	29	0
2	197	70	45	543	31	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	38	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27	1.441	57	0
1	189	60	23	846	30	0.398	59	1

Algorithms Sensitive to Feature Magnitude

- Linear and Logistic Regression
- Neural Networks
- Support Vector Machines
- KNN
- K-Means Clustering
- Linear Discriminant Analysis (LDA)
- Principal Component Analysis (PCA)

Normalization

- Scale the feature magnitude to a standard range like $[0, 1]$ or $[-1, +1]$.
- Techniques
  - Min-Max normalization
  - z-score normalization
  - Normalization by decimal scaling
- Impact of outliers in the data ???

Min-Max Scaling

- Min-max scaling squeezes (or stretches) all feature values to be within the range of $[0, 1]$.

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad \text{for range}[0, 1]$$

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)}(new_{max} - new_{min}) + new_{min} \qquad \text{for range}[new_{min}, new_{max}]$$

Min-Max Normalization

Suppose that the minimum and maximum values for the attribute income are \$12,000 and \$98,000, respectively. The new range is [0.0,1.0]. Apply min-max normalization to value of \$73,600.

$$\begin{aligned}\hat{x} &= \frac{x - \min(x)}{\max(x) - \min(x)}(new_{max} - new_{min}) + new_{min} \\ &= \frac{73600 - 12000}{98000 - 12000}(1.0 - 0.0) + 0.0 \\ &= 0.716\end{aligned}$$

z-score Normalization

- In z-score normalization (or zero-mean normalization), the values for an attribute, x, are normalized based on the mean $\mu(x)$ and standard deviation $\sigma(x)$ of x.
- The resulting scaled feature has a mean of 0 and a variance of 1.
- New range is $[-3\sigma, +3\sigma]$.

$$\hat{x} = \frac{x - \mu(x)}{\sigma(x)}$$

z-score Normalization

Suppose that the mean and standard deviation of the values for the attribute income are \$54,000 and \$16,000, respectively. Apply z-score normalization to value of \$73,600.

$$\begin{aligned}\hat{x} &= \frac{x - \mu(x)}{\sigma(x)} \\ &= \frac{73600 - 54000}{16000} \\ &= 1.225\end{aligned}$$

Decimal Normalization

- Normalizes by moving the decimal point of values of attribute x.
- The number of decimal points moved depends on the maximum absolute value of x.
- New range is $[-1, +1]$.

$$j = \text{smallest integer such that } \max(|\hat{x}|) < 1$$
$$\hat{x} = \frac{x}{10^j}$$

Decimal Normalization

Example 1		
CGPA	Formula	Normalized CGPA
2	2/10	0.2
3	3/10	0.3
Example 2		
Bonus	Formula	Normalized Bonus
450	450/1000	0.45
310	310/100	0.31
Example 3		
Salary	Formula	Normalized Salary
48000	48000/100000	0.48
67000	67000/100000	0.67

Table of Contents

Discretization

- Convert continuous attribute into a discrete attribute.
- Why?
  - Some statistical methods or machine learning algorithms like decision trees can handle discrete attributes only.
- Issues
  - How to choose the number of intervals K ?
  - How to define the cut points which are relevant according to the studied problem?

Age	Alzheimer
60	Yes
65	Yes
45	No
55	Yes
50	No

Discretization Techniques

- Unsupervised discretization
  - Equal-interval binning
  - Equal-frequency binning
- Supervised discretization
  - Entropy-based discretization

Unsupervised discretization

- Class labels are ignored.
- The best number of bins k is determined experimentally.
- User specifies the number of intervals and/or how many data points to be included in any given interval.
- Use Binning methods.

Unsupervised discretization

- Heuristics used to choose intervals
  1. The number of intervals for each attribute should not be smaller than the number of classes (if known).

     $$K \geq C \text{ classes}$$

  2. Choose the number of intervals, n_{F_i}, for each attribute, $F_i (i = 1, \ldots, n)$ where n is the number of attributes)

     $$n_{F_i} = \frac{M}{3} \times C$$

     where M is the number of training examples and C is the number of known classes.

Binning Methods

1. Equal width binning
   - Each bin has equal width.

$$width = interval = \frac{\max - \min}{\#bins}$$

   - Highly sensitive to outliers.
   - If outliers are present, the width of each bin is large, resulting in skewed data.
2. Equal frequency binning
   - Specify the number of values that have to be stored in each bin.
   - Number of entires in each bin are equal.
   - Some values can be stored in different bins.
3. Binning by clustering
   - Find natural gaps in the data.

Binning Example

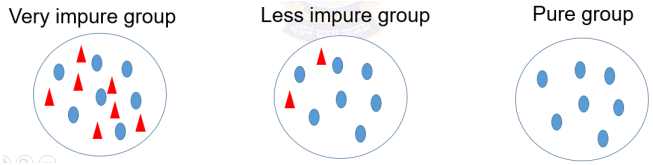
Original Data	53, 56, 57, 63, 66, 67, 67, 67, 68, 69, 70, 70, 70, 70, 72, 73, 75, 75, 76, 76, 78, 79, 80, 81			
Method		Bin1	Bin 2	Bin 3
Equal Width	81-53 = 28 28/3 = 9.33	[53, 62) = 53, 56, 57	[62, 72) = 63, 66, 67, 67, 67, 68, 69, 70, 70, 70, 70	[72, 81] = 72, 73, 75, 75, 76, 76, 78, 79, 80, 81
Equi Frequency	24 /3 = 8 24 /3 = 8	53, 56, 57, 63, 66, 67, 67, 67	68, 69, 70, 70, 70, 70, 72, 73	75, 75, 76, 76, 78, 79, 80, 81
Clustering	some variation	53, 56, 57, 63 66, 67, 67, 67, 68, 69	70, 70, 70, 70 72, 73, 75, 75	76, 76, 78, 79, 80, 81

Supervised Discretization

- Class labels are used.
- Entropy-based discretization
  - Entropy is the measure of the impurity /uncertainity in a group.

$$E(S) = \sum_{i=1}^{C} -p_i \log_2 p_i$$

  - Homogenous group has less entropy.
  - Heterogenous group has more entropy.



Supervised Discretization

- Information Gain
  - Measures how much “information” a feature gives us about the class.
  - Features that perfectly partition should give maximal information.
  - Unrelated features should give no information.
  - It measures the reduction in entropy.

Supervised Discretization

- Entropy based Discretization
  1. Sort examples in increasing order.
  2. Choose a value that forms an interval. (There can be m intervals.)
  3. Calculate the entropy measure of this discretization. $E(S) = \sum_{i=1}^{C} -p_i \log_2 p_i$
  4. Calculate entropy for the target given a bin.

$$E(S, F) = \sum_{\nu \in F} \frac{|S_\nu|}{|S|} E(S_\nu)$$

  5. Calculate Information Gain given a bin.

$$I(F) = E(S) - E(S, F)$$

  6. Apply the process recursively until some stopping criterion is met.

Entropy based Discretization Example

For the given data, find out how to discretize Runs.

Runs	53	56	57	63	66	67	67	67	68	69	70	70
Won	Y	Y	Y	N	N	N	N	N	N	N	N	Y
Runs	70	70	72	73	75	75	76	76	78	79	80	81
Won	Y	Y	N	N	N	Y	N	N	N	N	N	N

Entropy based Discretization Example

- Discretize the feature Runs using 2 bins using the vallue 60. So two Bins ≤ 60 and > 60.
- Compute the entropy for the target.

Runs	
Y	N
7	17

$$
\begin{aligned}
E(Runs) &= E(7, 17) \\
&= -\frac{7}{24}\log_2\frac{7}{24} - \frac{17}{24}\log_2\frac{17}{24} \\
&= 0.871
\end{aligned}
$$

Entropy based Discretization Example

- Compute the entropy for the target for the given bin.

Runs	Won	
	Y	N
≤ 60	3	0
> 60	4	17

$$E(Won, Runs) = P(\leq 60) * E(3,0) + P(> 60) * E(4,17)$$
$$= \frac{3}{24} * \left(-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right) + \frac{17}{24} * \left(-\frac{4}{21}\log_2\frac{4}{21} - \frac{17}{21}\log_2\frac{17}{21}\right)$$
$$= 0.615$$

- Calculate the information gain.

$$IG(Won, Runs) = 0.87 - 0.615 = 0.256$$

Entropy based Discretization Example

Runs	53	56	57	63	66	67	67	67	68	69	70	70	70	70	72	73	75	75	76	76	78	79	80	81
Matches Won	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N

		Matches Won	
		Y	N
Runs	<= 60	3	0
	> 60	4	17

Information Gain = 0.256

		Matches Won	
		Y	N
Runs	<= 70	6	8
	> 70	1	9

Information Gain = 0.101

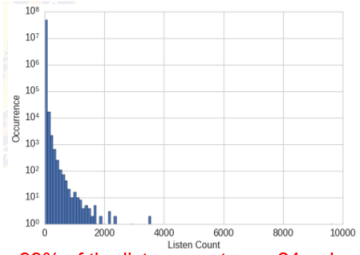
		Matches Won	
		Y	N
Runs	<= 75	7	11
	> 75	0	6

Information Gain = 0.148

Table of Contents

Why Binarization?

- Echo Nest Taste Profile Dataset
  - There are more than 48 million triplets of user ID, song ID, and listen count.
  - The full dataset contains 1,019,318 unique users and 384,546 unique songs.
  - Build a recommender system to recommend songs to users.



99% of the listen counts are 24 or lower

- Does a large listen count means the user really likes the song?

Binarization

- Binarization maps a continuous or categorical attribute into one or more binary attributes.
- Must maintain ordinal relationship.
- Binarization Techniques
  - One-hot encoding
  - Ordinal or Label Encoding
  - Count or Frequency Encoding

Binarization

- Example
  Assume an ordinal attribute for representing service of a restaurant: (Awful, Poor, OK, Good, Great)
  Use 3 bits.

Service Quality	X1	X2	X3
Awful	0	0	0
Poor	0	0	1
OK	0	1	0
Good	0	1	1
Great	1	0	0

  Unintended relationships: $X2$ and $X3$ are now correlated because “good” is encoded using both attributes,

One-Hot Encoding

- Encode each categorical variable with a set of boolean variables which take values 0 or 1, indicating if a category is present for each observation.
- One binary attribute for each categorical value.
- Advantages
  - Makes no assumption about the distribution or categories of the categorical variable .
  - Keeps all the information of the categorical variable .
  - Suitable for linear models.
- Disadvantages
  - Expands the feature space.
  - Does not add extra information while encoding.
  - Many dummy variables may be identical, introducing redundant information .
  - Number of resulting attributes may become too large.

One-Hot Encoding

- Example
  Assume an ordinal attribute for representing service of a restaurant: (Awful,Poor,OK,Good,Great)
  Use 5 bits.

Service Quality	X1	X2	X3	X4	X5
Awful	0	0	0	0	1
Poor	0	0	0	1	0
OK	0	0	1	0	0
Good	0	1	0	0	0
Great	1	0	0	0	0

Label Encoding

- Replace 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.
- Allows for quick benchmarking of machine learning models.
- Advantages
  - Straightforward to implement.
  - Does not expand the feature space.
  - Work well enough with tree based algorithms.
- Disadvantages
  - Does not add extra information while encoding.
  - Not suitable for linear models.
  - Does not handle new categories in test set automatically.

One-Hot Encoding

- Example
  Assume an ordinal attribute for representing service of a restaurant:
  (Awful,Poor,OK,Good,Great)

Service Quality	Integer Value
Awful	0
Poor	1
OK	2
Good	3
Great	4

Frequency Encoding

- Categories are replaced by the count or percentage of observations of each category.
- Assumption: the number observations shown by each category is predictive of the target.
- Advantages
  - Straightforward to implement.
  - Does not expand the feature space.
  - Work well enough with tree based algorithms.
- Disadvantages
  - Not suitable for linear models.
  - Does not handle new categories in test set automatically.
  - If two different categories appear the same amount of times in the dataset, that is, they appear in the same number of observations, they will be replaced by the same number: may lose valuable information.

Table of Contents

Feature Creation

- Create new attributes that can capture important information in a data set much more efficiently than the original attributes.
- Two general methodologies:
  - Feature Extraction
  - Feature Construction

Feature Extraction

- Mapping Features to a New Space
  - Fourier transform
  - Wavelet transform
  - Scale-Invariant Feature Transform (SIFT)

Feature Construction

- Create dummy features
  - Often used to convert categorical variable into numerical variables.

Customer ID	Gender	Payment Method		Online Banking	Credit Card	Debit Card
C001	Female	Online Banking		1	0	0
C002	Male	Online Banking		1	0	0
C003	Female	Credit Card		0	1	0
C004	Male	Debit Card		0	0	1

Feature Construction

- Create derived features

Customer ID	Gender	Session Begin	Session End		Session Duration
C001	Female	15-06-2019 10:30	15-06-2019 11:15		45
C002	Male	13-06-2019 08:00	13-06-2019 08:03		3
C003	Female	02-06-2019 16:25	02-06-2019 18:35		125
C004	Male	01-06-2019 11:20	01-06-2019 13:00		100

THANK YOU