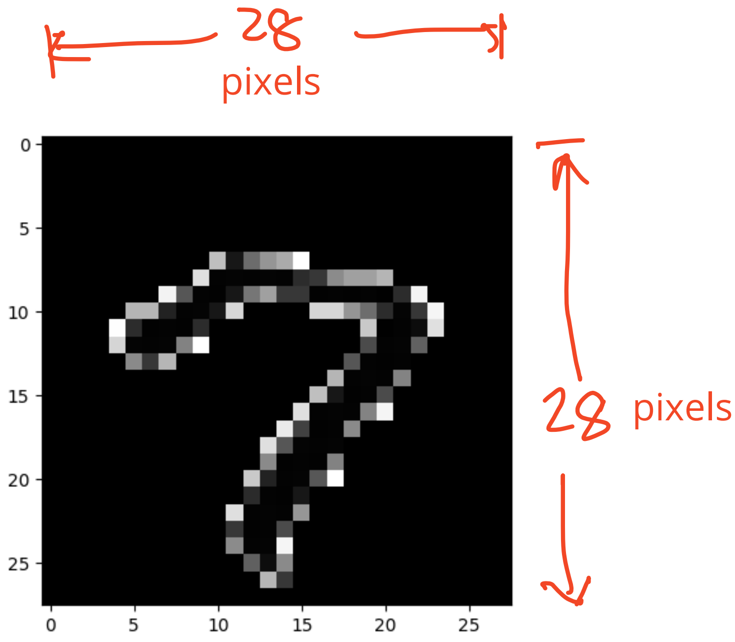
Continued:

- Understanding the overall "structure" of the data
  - PCA
  - t-SNE ✓

- Feature Transformation
  - Encoding & Binning.

The MNIST Data Set

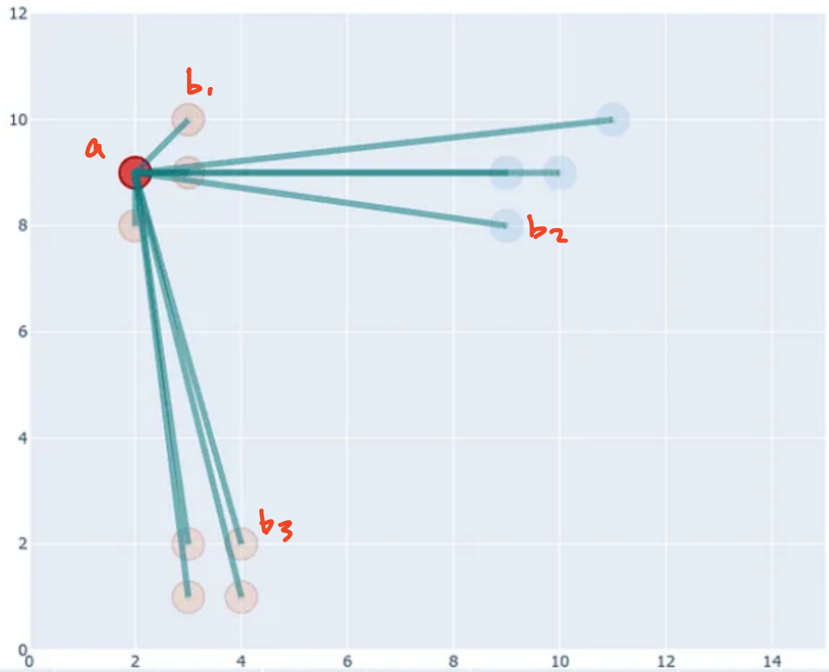
	A	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK
1	label	6x13	6x14	6x15	6x16	6x17	6x18	6x19	6x20	6x21	6x22	6x23	6x24	6x25	6x26
2	5	3	18	18	18	126	136	175	26	166	255	247	127	0	0
3	0	0	0	48	238	252	252	252	237	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	67	232	39	0	0	0
5	1	0	0	0	0	0	0	124	253	255	63	0	0	0	0
6	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	2	0	0	0	13	25	100	122	7	0	0	0	0	0	0
8	1	237	253	252	71	0	0	0	0	0	0	0	0	0	0
9	3	43	105	255	253	253	253	253	253	174	6	0	0	0	0
10	1	5	63	197	0	0	0	0	0	0	0	0	0	0	0
11	4	0	0	0	0	0	0	0	0	0	143	247	153	0	0
12	3	254	254	254	254	254	66	0	0	0	0	0	0	0	0
13	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	3	155	155	131	52	0	0	0	0	0	0	0	0	0	0
15	6	38	178	252	253	117	65	0	0	0	0	0	0	0	0
16	1	168	242	28	0	0	0	0	0	0	0	0	0	0	0
17	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	2	164	211	250	250	194	15	0	0	0	0	0	0	0	0
19	8	0	0	0	0	0	0	0	11	203	229	32	0	0	0
20	6	0	0	75	247	143	10	0	0	0	0	0	0	0	0
21	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	4	0	0	0	0	0	112	252	125	4	0	0	0	0	0
23	0	0	0	0	0	96	205	251	253	205	111	4	0	0	0
24	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	121	254	136	0	0	0	0	0
26	1	0	0	29	249	254	254	9	0	0	0	0	0	0	0
27	2	246	253	253	253	253	253	220	154	17	3	0	0	0	0
28	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	3	207	255	254	254	254	97	80	80	44	0	0	0	0	0



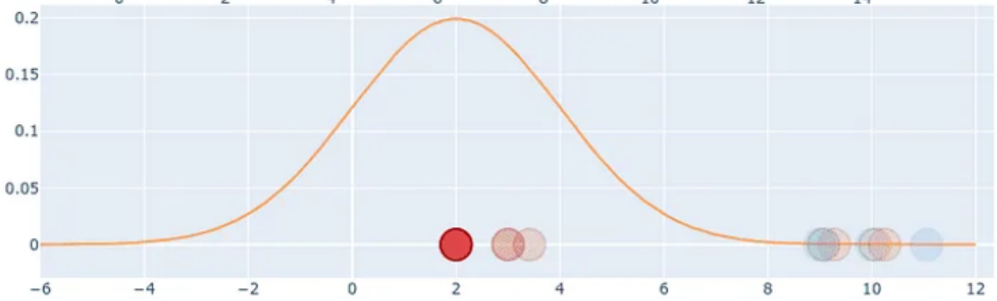
MNIST stands for “Modified National Institute of Standards and Technology” database. It is a large database of small, square 28x28 pixel grayscale images of handwritten single digits between 0 and 9. The MNIST database contains 60,000 training images and 10,000 testing images, with each image labeled with the respective digit that it represents.

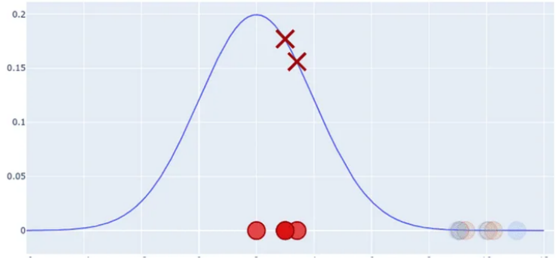
How to UNDERSTAND the structure of such a large data set?

t-SNE
(t-distributed Stochastic Neighbour Encoding)



- t-SNE is a machine learning algorithm that is **used for dimensionality reduction and data visualization**.
- It works by finding the similarity measure between pairs of instances in higher and lower dimensional spaces, and tries to maintain the probability distribution for data samples in lower dimensions the same as the probability distribution of data samples in higher dimensions.
- The main advantage of t-SNE is the **ability to preserve local structure**, meaning that points which are close to one another in the high-dimensional data set will tend to be close to one another in the chart - which aspect is advantageously used for visualization.





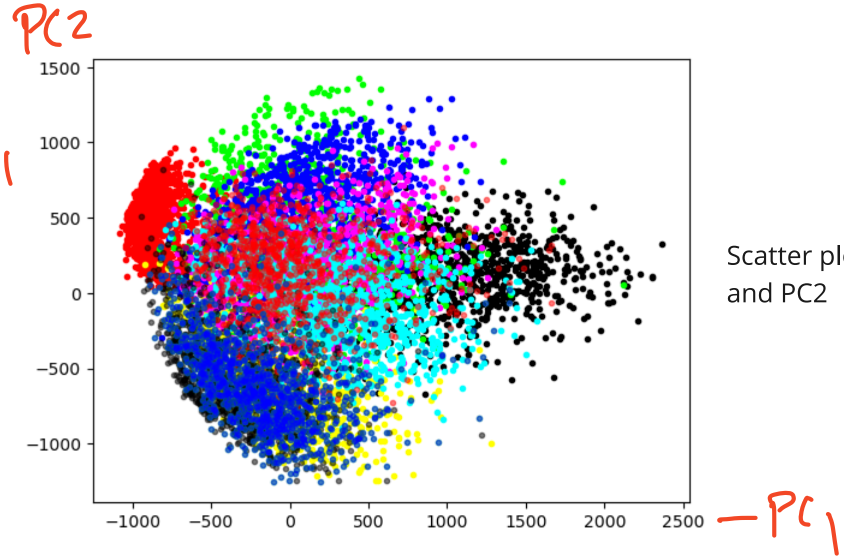
$p(a|b)$

Visualization of MNIST data using PCA and t-SNE

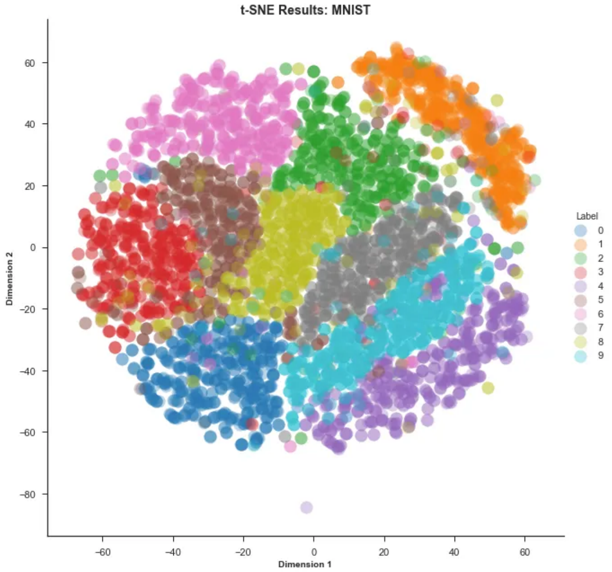
MNIST Data Set

	A	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK
1	label	6x13	6x14	6x15	6x16	6x17	6x18	6x19	6x20	6x21	6x22	6x23	6x24	6x25	6x26
2	5	3	18	18	18	126	136	175	26	166	255	247	127	0	0
3	0	0	0	48	238	252	252	252	237	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	67	232	39	0	0	0
5	1	0	0	0	0	0	0	124	253	255	63	0	0	0	0
6	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	2	0	0	0	13	25	100	122	7	0	0	0	0	0	0
8	1	237	253	252	71	0	0	0	0	0	0	0	0	0	0
9	3	43	105	255	253	253	253	253	253	174	6	0	0	0	0
10	1	5	63	197	0	0	0	0	0	0	0	0	0	0	0
11	4	0	0	0	0	0	0	0	0	0	143	247	153	0	0
12	3	254	254	254	254	254	66	0	0	0	0	0	0	0	0
13	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	3	155	155	131	52	0	0	0	0	0	0	0	0	0	0
15	6	38	178	252	253	117	65	0	0	0	0	0	0	0	0
16	1	168	242	28	0	0	0	0	0	0	0	0	0	0	0
17	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	2	164	211	250	250	194	15	0	0	0	0	0	0	0	0
19	8	0	0	0	0	0	0	0	11	203	229	32	0	0	0
20	6	0	0	75	247	143	10	0	0	0	0	0	0	0	0
21	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	4	0	0	0	0	0	112	252	125	4	0	0	0	0	0
23	0	0	0	0	0	96	205	251	253	205	111	4	0	0	0
24	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	121	254	136	0	0	0	0	0
26	1	0	0	29	249	254	254	9	0	0	0	0	0	0	0
27	2	246	253	253	253	253	253	220	154	17	3	0	0	0	0
28	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	3	207	255	254	254	254	97	80	80	44	0	0	0	0	0

Note: Usually, PCA precedes t-SNE; output of PCA becomes the input to t-SNE



Scatter plot of PC1 and PC2



Scatter plot of 2D t-SNE

Feature Encoding

Label

$$\underbrace{y}_{\text{Categorical}} = f(\underbrace{x_1, x_2 \ldots x_k}_{\text{some of these may be categorical}})$$

Categorical

M | F

R | O | G | B | Y | ....

Need to encode these values into 'numbers' – 0, 1, 2, 3, 4, 5 (Label Encoding).

Some of these may be categorical

... they need to be encoded prior to training ML models

When either the dependent variable or some of the independent variables are 'categorical' then, they have to be appropriately encoded prior to being used for training ML models.

- Label encoding
- One-hot encoding
- Binary encoding
- Integer encoding
- Frequency encoding
- Target encoding

Label Encoding	• Typically used to encode the 'label' or the 'target variable' (y) if it is a nominal variable (ie. there is no inherent order) • Each possible category is encoded into an integer
one-hot encoding	• In this scheme, each possible category is given a column of it's own. For example if the possible categories in the Gender column are M and F, then two columns are created - Gender_M and Gender_F. The category 'M' is encoded as [1, 0] while category 'F' is encoded as [0, 1] • See example in the net slide • This scheme results in an explosion of columns if there are too many categorical variables, with too many category values in each.
Binary Encoding	• This method is useful to overcome the drawbacks associated with one-hot encoding • In this scheme, the number of added columns are significantly lesser than in the case of one-hot encoding. • See the next slide for explanation
Integer Encoding	• This method is used to encode categorical variables that are ordinal - ie. their values have an inherent order • Each category values is replaced with an integer that reflects it's position in the order • Eg. A -> 5, B -> 4, C -> 3, etc. if the alphabets represent grades (the numbers represent their 'value')
Frequency Encoding	• Frequency encoding is a technique used to encode categorical variables into numeric values based on their frequency of occurrence in the dataset column. • Frequency encoding can be effective for nominal features, especially when the number of unique values is high. It is a preferable method since it gives good labels, unlike one-hot encoding, which eliminates the order but causes the number of columns to expand vastly
Target Encoding	• Target encoding is a technique used to encode categorical variables into numeric values based on the target variable. (See subsequent slide for example) • It can be effective for categorical features, especially when the number of unique category values is high - and likely to lead to column explosion if one-hot-encoding is used. • One of the challenges with target encoding is overfitting

Note: The frequently used schemes are : Label encoding, Integer Encoding and One-hot Encoding

ONE-HOT-ENCODING

x_1	x_2		x_3	Y
Gender	AgeYears	AgeMonths	HeightInCm	WeightInKg
M	18	11	186	69.5
M	20	1	162	65
M	19	9	170	77
M	18	7	183	72
M	19	8	176	64

$$Y = f(x_{1\_M}, x_{1\_F}, x_2, x_3)$$

Gender_M	Gender_F
1	0
0	1

Encoding of Categorical Independent Variable
"ONE-HOT-ENCODING"

Problem? with one-hot-encoding

when there are a large number of categorical Indep-Variables.

ONE-HOT-ENCODING

Data: original

	Gender	AgeYears	AgeMonths	HeightInCm	WeightInKg	WeightInKgNew
0	M	18	11.0	186.0	69.5	70.308
1	M	20	1.0	162.0	65.0	61.769
2	M	19	9.0	170.0	77.0	66.754
3	M	18	7.0	183.0	72.0	69.516
4	M	19	8.0	176.0	64.0	66.211

Data: one-hot-encoded

	AgeYears	AgeMonths	HeightInCm	WeightInKg	WeightInKgNew	Gender_F	Gender_M
0	18	11.0	186.0	69.5	70.308	0	1
1	20	1.0	162.0	65.0	61.769	0	1
2	19	9.0	170.0	77.0	66.754	0	1
3	18	7.0	183.0	72.0	69.516	0	1
4	19	8.0	176.0	64.0	66.211	0	1

Binary Encoding

This scheme Involves the following steps:

1. Assign a numerical value to the category
2. Express the numerical value in Binary notation
3. Introduce a column for each bit in the binary notation

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

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

Frequency Encoding :

M → 75

F → 8

$c_1 \rightarrow n_1$

$c_2 \rightarrow n_2$

$c_3 \rightarrow n_3$

$c_4 \rightarrow n_4$

$c_5 \rightarrow \vdots$

c_6 -

In this case, the category values (eg. M) are replaced with it's frequency in the column (eg. 75)

Target Encoding :

Y	Colour (x_1)	x_2	x_3
2.4	R	...	...
3.1	G	...	...
3.6	G		
2.8	R		
1.9	B		
2.55	O		
2.3	R		
3.1	B		

$$R \rightarrow \frac{2.4 + 2.8 + 2.3}{3}$$

→ Average of the target for that category value.

All 'R' values in the column are replaced by the average calculated above = 2.5

Feature Binning

- Feature binning is used in machine learning when continuous numerical features need to be converted into categorical features.
- Binning is a technique that involves dividing continuous numerical features into distinct groups or "bins" based on ranges that are determined.
- It can be used for several reasons, including problem simplification - where permissible, reducing the impact of outliers and noise in the data, handling non-linear relationships, and reducing the number of unique values in a feature.
- There are several types of binning methods, including equal width binning, equal frequency binning, and quantile binning

Example: Height values 'binned' into category T/M/S

Gend	FinalAge	HeightInC	WeightInK	HtCategory
M	18.91666667	186	69.5	T
M	20.08333333	162	65	M
M	19.75	170	77	M
M	18.58333333	183	72	T
M	19.66666667	176	64	T
M	19.33333333	173	63	T
M	19.58333333	174	57	T
M	23.08333333	170	63	M
M	18.91666667	171	61	T
M	18.75	175	56	T
F	18.75	165	63	M
F	35.5	172	76	T
M	19.5	167	58	M
M	18.91666667	180	60	T
M	19.16666667	180	60	T
M	18.58333333	170	79	M
M	19.16666667	183	62	T