Data Visualization

Lecture 8

Data Preprocessing

Feature Scaling & Data Resampling

Machine Learning Basics

AI vs Machine Learning Vs Deep Learning

- AI is the broad concept of making machines intelligent, mimic human intelligence through a set of algorithms. The field focuses on three skills: learning, reasoning, and self-correction to obtain maximum efficiency.
- Machine Learning is a subset of AI, uses algorithms that learn from data to make predictions.
- Deep Learning is a subset of machine learning that uses artificial neural networks to process and analyze information.

Machine Learning Types

- Supervised Learning: Trains models on labeled data to predict or classify new, unseen data. is generally categorized into two main types:
    - Classification where the goal is to predict discrete labels or categories
    - Regression where the aim is to predict continuous numerical values.
- Unsupervised Learning: Finds patterns or groups in unlabeled data, like clustering.
- Semi-supervised Learning: This approach combines a small amount of labeled data with a large amount of unlabeled data. It’s useful when labeling data is expensive or time-consuming.
- Reinforcement Learning: Learns through trial and error to maximize rewards, ideal for decision-making tasks.

Linear Regression

- It assumes that there is a linear relationship between the input and output, meaning the output changes at a constant rate as the input changes. This relationship is represented by a straight line.
- best-fit line is the straight line that most accurately represents the relationship between the independent variable (input) and the dependent variable (output).
- The goal of the best line is to minimize the difference between the actual data points and the predicted values from the model.
- Multiple linear regression generalizes the case of one predictor to several predictors (more than on independent variable)

Linear Regression

- Predicting house price based on number of rooms. The formula for best fit line is:

$$Y = b_0 + b_1X$$

- Predicting house price based on number of rooms, proximity to the ocean, median income of the neighborhood. The formula for the best fit line is:

$$Y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3$$

Estimation of Mean Response

- Fitted regression line can be used to estimate the mean value of y for a given value of x.
- Example
  - The weekly advertising expenditure (x) and weekly sales (y) are presented in the following table.

y	x
1250	41
1380	54
1425	63
1425	54
1450	48
1300	46
1400	62
1510	61
1575	64
1650	71

Point Estimation of Mean Response

- From previous table we have:

$$n = 10 \qquad \sum x = 564 \qquad \sum x^2 = 32604$$
$$\sum y = 14365 \qquad \sum xy = 818755$$

- The least squares estimates of the regression coefficients are:

$$b_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} = \frac{10(818755) - (564)(14365)}{10(32604) - (564)^2} = 10.8$$

$$b_0 = 1436.5 - 10.8(56.4) = 828$$

K-Nearset Neighbor

- It works by finding the "k" closest data points (neighbors) to a given input and makes a predictions based on the majority class
- K-Nearest Neighbors is also called as a lazy learner algorithm because it does not learn from the training set immediately instead it stores the entire dataset and performs computations only at the time of classification.
- KNN uses distance metrics to identify nearest neighbor; To identify nearest neighbor we can use below distance metrics:
  - Euclidean Distance: $d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$
  - Manhattan Distance: $d = |(x2 - x1) + (y2 - y1)|$

K-Nearset Neighbor

Person	Weight (kg)	Exercise (min/day)	Class
A	120	20	Unfit
B	110	15	Unfit
C	75	90	Fit
D	70	100	Fit
E	68	95	Fit
New Person	**82**	**40**	?

A fitness center wants to classify new clients as **“Fit”** or **“Unfit”** using KNN based on two features:

- **Weight (kg)** – large scale (40–120)
- **Daily Exercise Time (minutes)** – small scale (0–120)

A new person arrives: **Weight:** 82 kg, **Exercise time:** 40 minutes

K-Nearset Neighbor

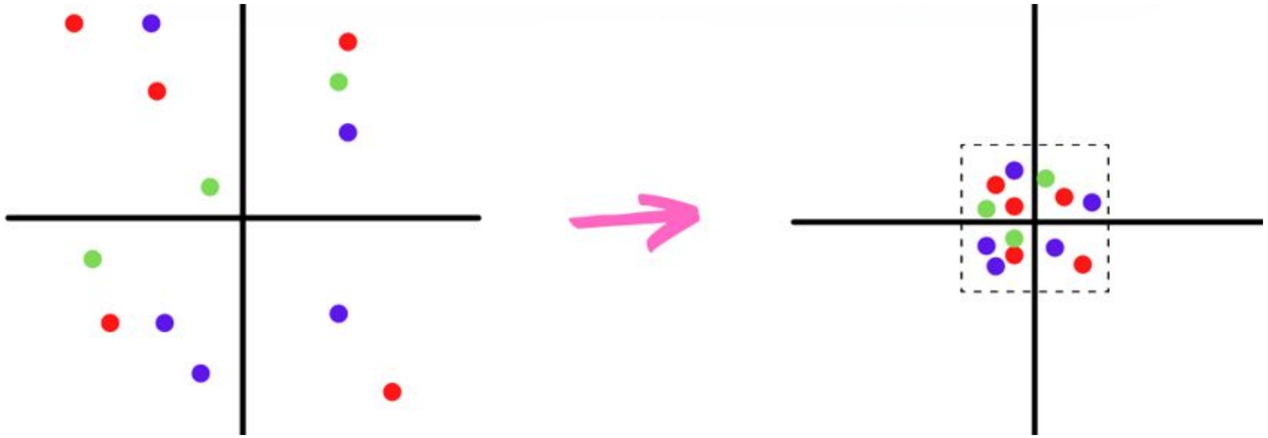
Person	Weight (kg)	Exercise (min/day)	Class	Distance
A	120	20	Unfit	
B	110	15	Unfit	
C	75	90	Fit	
D	70	100	Fit	
E	68	95	Fit	
New Person	**82**	**40**	?	

- The closest three people are B, A, C.
- Prediction: New person is under “unfit” Class

Feature Scaling

What is Feature Scaling?

- In Data Processing, we try to change the data in such a way that the model can process it without any problems.
- Feature Scaling is one such process in which we transform the data into a better version.
- Feature Scaling is done to normalize the features in the dataset into a finite range.



Why Feature Scaling?

- Real Life Datasets have many features with a wide range of values like for example let’s consider the house price prediction dataset. It will have many features like no. of. bedrooms, square feet area of the house, etc.

- As you can guess, the no. of bedrooms will vary between 1 and 5, but the square feet area will range from 500-2000. This is a huge difference in the range of both features.

- Many machine learning algorithms that are using Euclidean distance as a metric to calculate the similarities will fail to give a reasonable recognition to the smaller feature, in this case, the number of bedrooms, which in the real case can turn out to be an actually important metric.

K-Nearset Neighbor

Person	Z-Weight	Z-Exercise	Class	Distance
A	1.434	-1.13	Unfit	
B	0.979	-1.261	Unfit	
C	-0.619	0.668	Fit	
D	-0.852	0.925	Fit	
E	-0.945	0.796	Fit	
New Person	**-0.302**	**-0.617**	?	

KNN doesn’t know which feature is important. It only knows which numbers are bigger. Bigger numbers dominate the distance unless you scale.

Feature Scaling Techniques

- Types of Feature Scaling:
  - Standardization:
    - Standard Scaler
  - Normalization:
    - Min Max Scaling
    - Mean Normalization
    - Max Absolute Scaling
    - Robust Scaling.

What Is Normalization?

- Normalization is a data preprocessing technique used to adjust the values of features in a dataset to a common scale.

- Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

- Here's the formula for normalization:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

What Is Standardization?

- Standardization is another scaling method where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero, and the resultant distribution has a unit standard deviation.

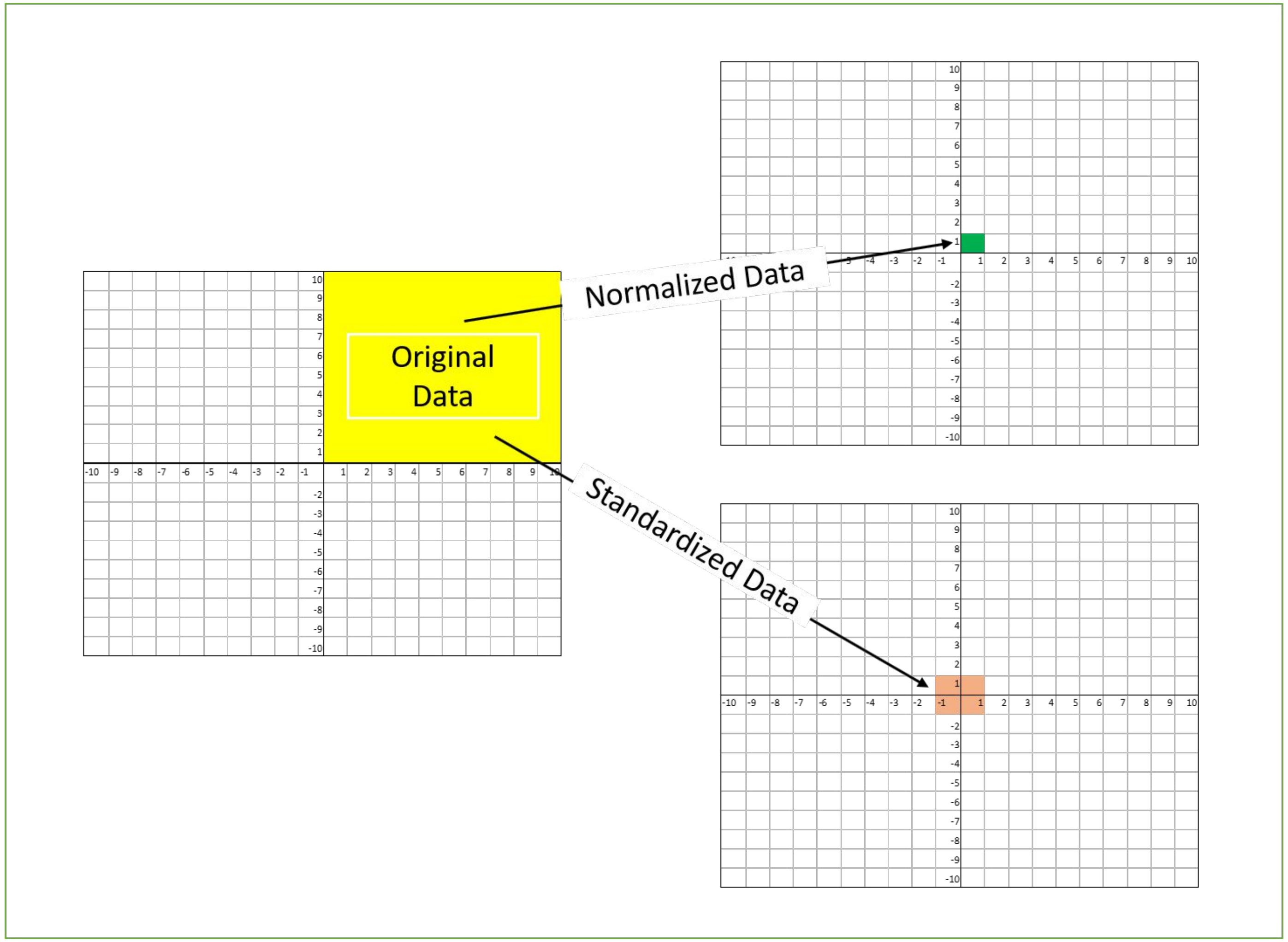
- Here's the formula for standardization:

$$X' = \frac{X - \mu}{\sigma}$$

Normalization vs. Standardization

Aspect	**Normalization**	**Standardization**
Core Idea	Scaling is done by the highest and the lowest values.	Scaling is done by mean and standard deviation.
Use Case	Distance-based or gradient-based models that benefit from bounded inputs, such as k-NN and many neural networks	Models that assume or exploit normality and variance structure, like linear/logistic regression, SVMs, and PCA.
Scale	Most commonly to a fixed range such as [0,1]	Not bounded
Sensitivity to Outliers	Affected by outliers	Less affected by outliers
Distribution assumptions	Often used when the data distribution is unknown or not Gaussian.	It is used when the data is Gaussian or normally distributed
Common alternative names	It is also known as Scaling Normalization	It is also known as Z-Score

Normalization vs. Standardization



Standardization (Standard Scaler)

- Calculate the z-value for each of the data points and replaces those with these values.

$$X_{new} = \frac{X - X_{mean}}{\sigma}$$

```
y1_new = (y1-np.mean(y1))/np.std(y1)
y2_new = (y2-np.mean(y2))/np.std(y2)
```

Normalization (Min Max Scaler)

- In min-max you will subtract the minimum value in the dataset with all the values and then divide this by the range of the dataset(maximum-minimum). In this case, your dataset will lie between 0 and 1 in all cases

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

```
y1_new = (y1-min(y1))/(max(y1)-min(y1))
y2_new = (y2-min(y2))/(max(y2)-min(y2))
```

Normalization (Mean Normalization)

Instead of using the min() value in the previous case, in this case, we will be using the average() value

$$X_{new} = \frac{X - X_{mean}}{X_{max} - X_{min}}$$

```
y1_new = (y1-np.mean(y1))/(max(y1)-min(y1))
y2_new = (y2-np.mean(y2))/(max(y2)-min(y2))
```

Normalization (Absolute Maximum Scaler)

- Find the absolute maximum value of the feature in the dataset
- Divide all the values in the column by that maximum value
- If we do this for all the numerical columns, then all their values will lie between -n and m.

```
y1_new = y1/max(y1)
y2_new = y2/max(y2)
```

Normalization (Robust Scaler)

- In this method, you need to subtract all the data points with the median value and then divide it by the Inter Quartile Range(IQR) value.

$$X_{new} = \frac{X - X_{median}}{IQR}$$

```
from scipy import stats
IQR1 = stats.iqr(y1, interpolation = 'midpoint')
y1_new = (y1-np.median(y1))/IQR1
IQR2 = stats.iqr(y2, interpolation = 'midpoint')
y2_new = (y2-np.median(y2))/IQR2
```

Techniques Comparison

Scaler	Use When	Avoid When
StandardScaler	- Data roughly normal - No heavy outliers	- Data has strong outliers - Distribution is extremely skewed
MinMaxScaler	- You need 0–1 range - Features originally have very different scales and want to keep relative distances	- Outliers present - You require robustness to future unseen values that may fall outside the min/max
Mean Normalization	- You want features centered at 0 but still scaled by their range instead of variance	- Outliers present
MaxAbsScaler	- Data is already centered at or near 0 and may be sparse	- Features are not centered and contain large outliers
RobustScaler	- You want features on a comparable scale for distance/gradient-based models but do not want to trim outliers.	- Very small sample sizes, where robust statistics like IQR may be unstable. - Data is already well-behaved and close to Gaussia

Data Resampling

What is Imbalanced datasets?

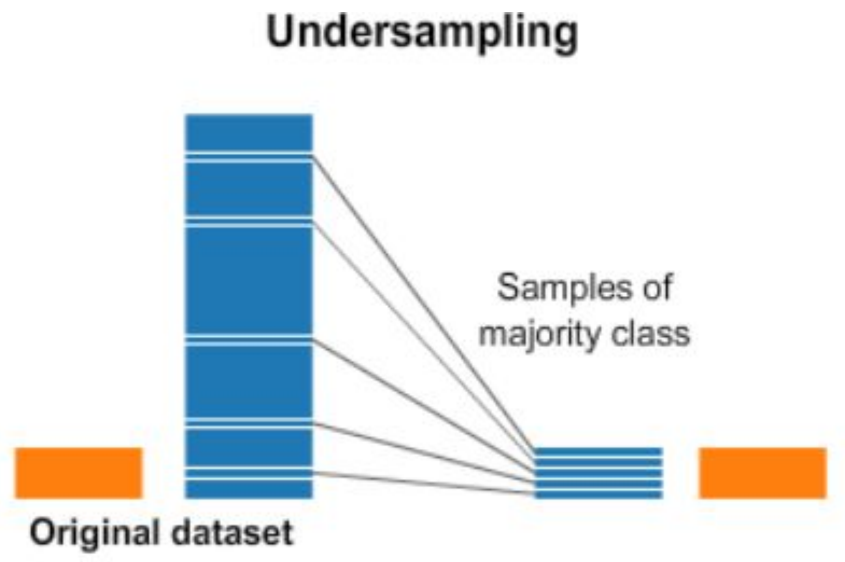
- Imbalanced datasets are those where there is a severe skew in the class distribution, such as 1:100 or 1:1000 examples in the minority class to the majority class.

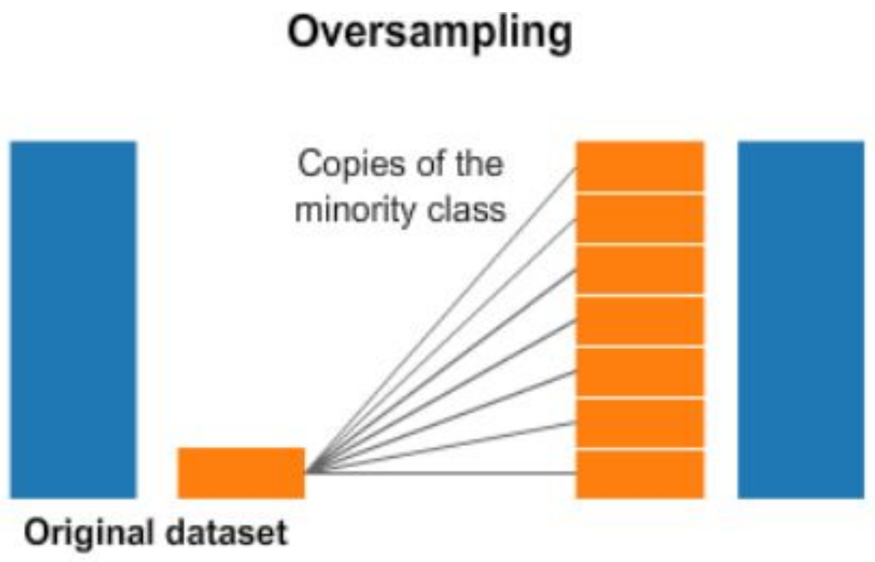
- This bias in the training dataset can influence many machine learning algorithms, leading some to ignore the minority class entirely. This is a problem as it is typically the minority class on which predictions are most important.

- One approach to addressing the problem of class imbalance is to resample the training dataset.

What is Resampling?

- A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and / or adding more examples from the minority class (over-sampling).



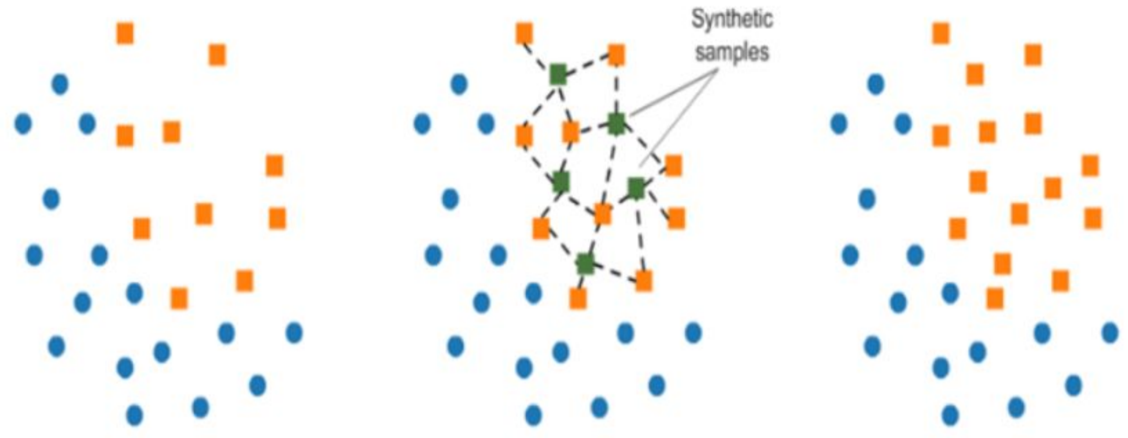


Resampling Techniques

- Resampling techniques: -
  - Random Under-Sampling: -
    - Random Under-sampling aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.
  - Random Over-Sampling: -
    - Over-Sampling increases the number of instances in the minority class by randomly replicating them to present a higher representation of the minority class in the sample.

Resampling Techniques

- Synthetic Minority Over-sampling Technique (SMOTE): -
  - This technique is followed to avoid over-fitting which occurs when exact replicas of minority instances are added to the main data set. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original datasets.



Resampling Techniques (Random Under-Sampling)

Advantages	Disadvantages
• It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge.	• It can discard potentially useful information which could be important for building rule classifiers. The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set.

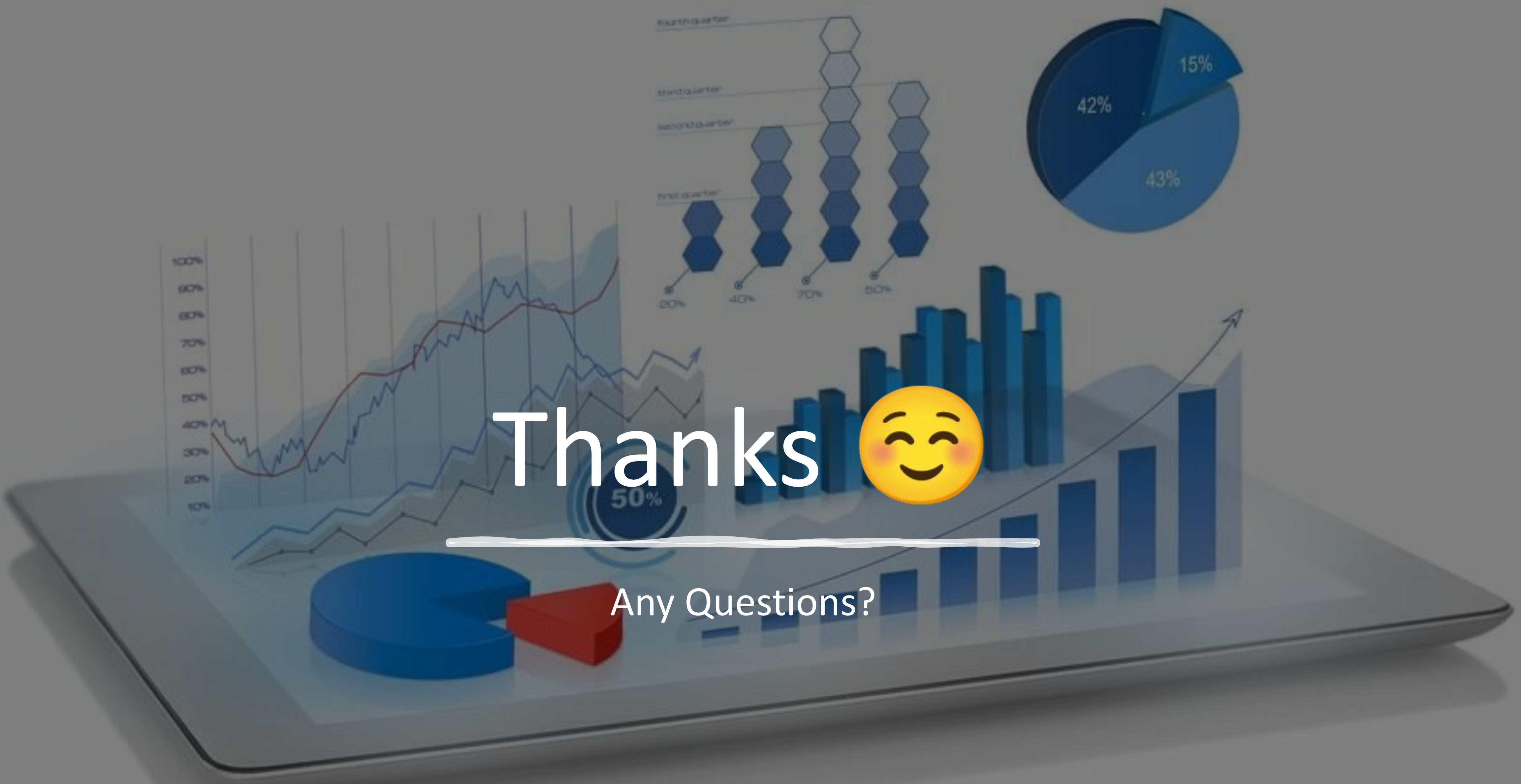
Resampling Techniques (Random Over-Sampling)

Advantages	Disadvantages
• Unlike under sampling this method leads to no information loss. Outperforms under sampling	• It increases the likelihood of over-fitting since it replicates the minority class events.

Resampling Techniques (SMOTE)

Advantages	Disadvantages
• Mitigates the problem of over-fitting caused by random oversampling as synthetic examples are generated rather than replication of instances. Also there is no loss of useful information	• While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise.

Thanks ☺️



Any Questions?