Data Mining for Business

Feature Engineering and Dimension Reduction

Dr. Shipra Maurya

Department of Management Studies

IIT (ISM) Dhanbad

Email: shipra@iitism.ac.in

Feature Engineering

- It is the process of creating features from raw data to make ML algorithms work more efficiently.
- It improves machine learning model performance and prediction accuracy as with feature engineering, predictive models can deeply understand the dataset and perform well on unseen data.
- Not a generic method. Different datasets require different approaches

Feature Engineering Processes

- Feature Selection – an iterative process that starts before model development and continues till the final model is chosen
- Feature Construction :
    - Binning continuous variables to discrete
    - Encoding Categorical variable into Numeric variable
    - Creating new features from existing features
- Variable transformation (only for continuous variables)
- Feature Scaling
- Feature Extraction or Dimensionality reduction – PCA and t-SNE

Feature Engineering Tools

Tools/Measures	Support for type of databases	Feature engineering	Feature selection	Open source implementation	Support for time series
Featuretools	Relational Tables	Yes	Yes	Yes	Yes
AutoFeat	Single Table	Yes	Yes	Yes	No
TSFresh	Single Table	Yes	Yes	Yes	Yes
FeatureSelector	Single Table	No	Yes	Yes	No
OneBM	Relational Tables	Yes	Yes	No	Yes
Cognito	Single Table	Yes	Yes	No	No

- Although there are above tools available for automated feature engineering, still the best approach is to perform it manually

Image Source : Internet

Feature Scaling

- Feature Scaling is a technique to standardize the independent features present in the data in a fixed range (makes data unitless)

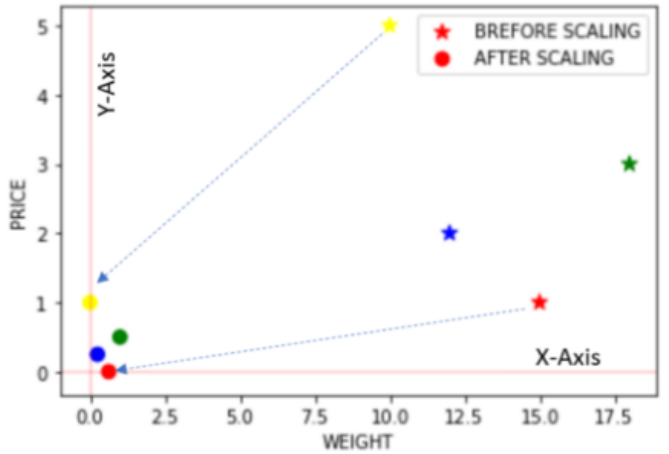
- Why scaling?
- When to scale?
  - Algorithms in which feature scaling matters
  - Algorithms in which features scaling does not matter

- **Techniques of Scaling:**
  - Normalization
  - Standardization

Feature Scaling: Normalization

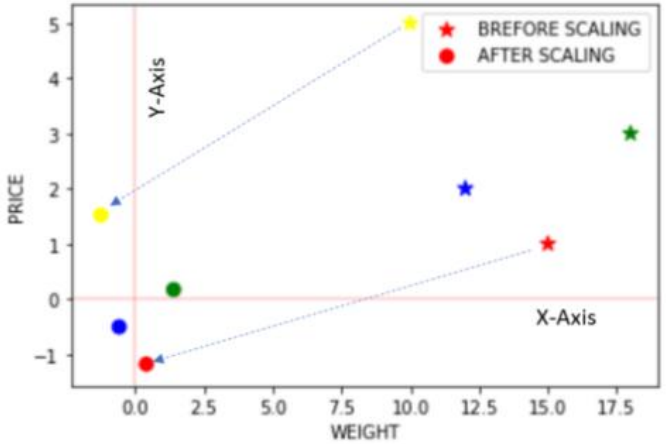
- Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$



Feature Scaling: Standardization

- It assumes that data is normally distributed within each feature and scales them such that the distribution centered around 0, with a standard deviation of 1.



$$x_{new} = \frac{x - \mu}{\sigma}$$

Image Source : Internet

Feature Selection

- Feature selection means selection of required features which have more association with the target variable. It will help in building a good model by eliminating redundant features.
- **Feature selection importance:**
  1. Simple models are easy to interpret
  2. Reduction in training time
  3. Enhanced generalization of the model by reducing overfitting
  4. Drop redundant variables – highly correlated variables

Feature Selection Methods

1. Domain knowledge
2. Filter Methods
3. Wrapper Methods
4. Embedded Methods
5. Hybrid Methods

Feature Selection- Filter Methods

Univariate –

- Basic filter methods
  - Constant features,
  - Quasi-constant features,
  - Duplicated features
- Mutual Information Score
- Chi-squared score
- Linear Regression F-test
- ANOVA F-test
- Univariate ROC-AUC/RMSE

Multivariate - Correlation

Features/ Target	Continuous (Regression problem)	Categorical (Classification Problem)
Continuous	Linear Regression F-test (f-regression)	ANOVA F-test (f-classification)
Categorical		Chi-square test

Feature Selection : Wrapper Methods

- **Forward Feature Selection**: starts with no feature and adds one at a time
- **Backward Feature Elimination**: starts with all features present and removes one feature at the time
- **Exhaustive Feature Selection**: tries all possible feature combinations
- **Bidirectional Search**: performs both forward and backward feature selection simultaneously in order to get one unique solution

Note – mlxtend library in Python

Feature Selection : Embedded Methods

- **Using Regularization**
  - Lasso regression or L1 Regularization
  - Ridge regression or L2 Regularization
  - Elastic Nets or L1/L2 Regularization
- **Tree-based feature importance**
  - Feature Importance using feature score

What is Feature Importance?

- It refers to techniques that calculate a score for all the features for a given model — the scores simply represent the “importance” of each feature

- A higher score indicates the larger effect of the feature on the target variable

- **Need for Feature Importance –**
  - Understanding the relationship between features and target variable
  - Model improvement – drop the features with low importance to reduce dimensionality
  - Communication and Interpretation of model findings to the stakeholders

Methods to calculate Feature Importance

- Feature Importance for Linear Models
- Feature Importance for Tree-based Models
    - Gini Importance or Mean decrease in Impurity
    - Permutation-based Feature Importance
    - SHAP Value – Shapley Additive Explanations

Gini Importance or Mean Decrease in Impurity

- Used to calculate the node impurity and feature importance
- It is basically a reduction in the impurity of a node weighted by the number of samples that are reaching that node from the total number of samples. This is known as node probability.

$$p = \frac{\sum_{j:\text{node j splits on feature i}} ni_j}{\sum_{j \in \text{all nodes}} ni_j}$$

$$GI = 1 - \sum_{i=1}^{n} (p)^2$$

$$GI = 1 - \left[\left(P_{(+)}\right)^2 + \left(P_{(-)}\right)^2 \right]$$

Image Source : Internet

Permutation-based Feature Importance

- It calculates relative importance score independent of model used
- **Steps:**
  1. Pick any random feature, shuffle the values in that feature, measure the model performance (predicted and actual output)
  2. Returns the feature in its original form (undo reshuffle)
  3. Repeat above steps for all the features in the model
  4. Final important features will be calculated by comparing individual score with mean importance score

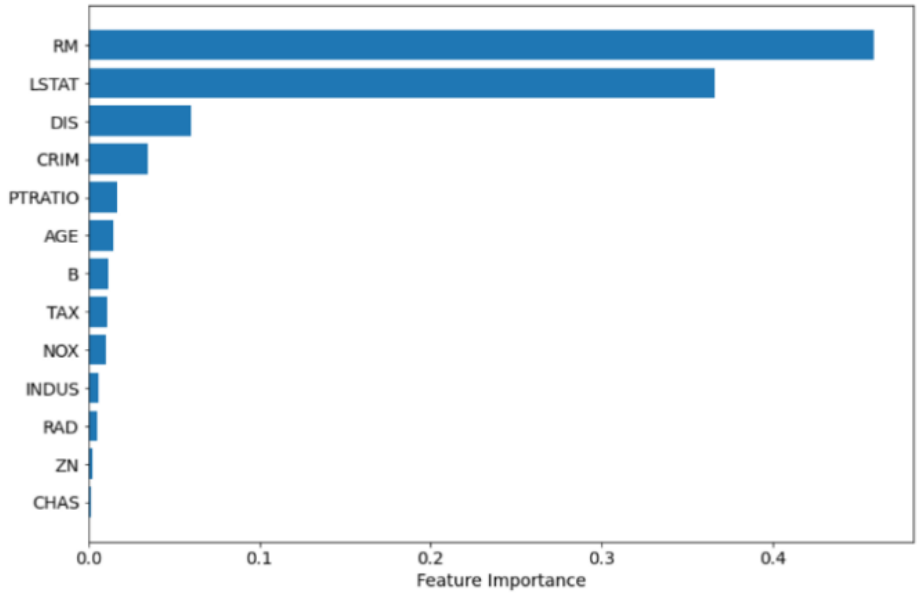


Image Source : Internet

SHAP values

- SHAP value is the average marginal contribution of a feature value across all the possible combinations of features
- Step 1 – calculate the predicted values without including the feature for which SHAP value is being calculated
- Step 2 – calculate the predicted values including the feature for which SHAP value is being calculated
- Step 3 - Difference between Step 2 – Step 1 is the SHAP value of the specific feature

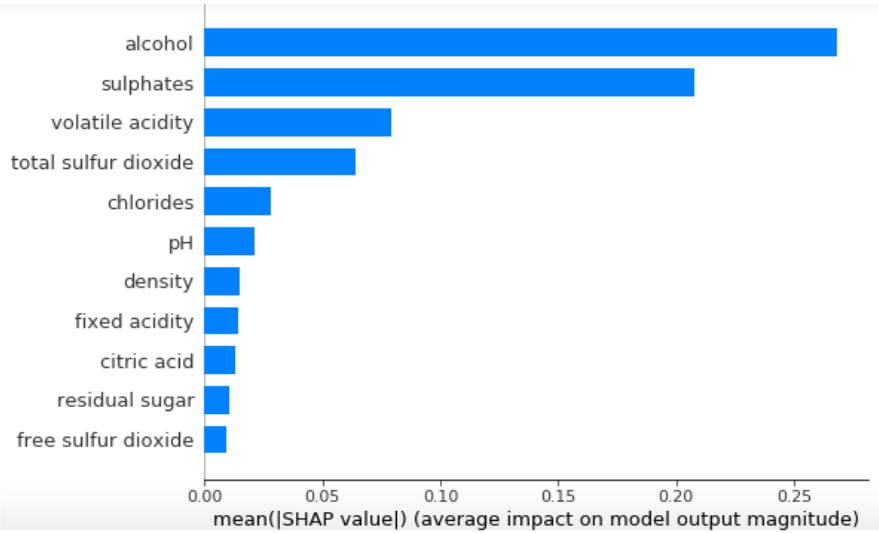


Image Source : Internet

Feature Selection : Hybrid Methods

- Using Filter and Wrapper methods
- Using Embedded and Wrapper methods
    - Recursive feature elimination
    - Recursive feature addition

Variable Transformation (1/2)

- It is the process of converting raw data into a format or structure that would be more suitable for model building

- **Why Variable transformation?**
  - the algorithm is more likely to be biased when the data distribution is skewed
  - transforming data into the same scale allows the algorithm to compare the relative relationship between data points better

- Variables are transformed when we apply supervised learning algorithms

Variable Transformation (2/2)

- Most commonly used methods are:
  - Logarithmic Transformation → f(x)=ln(x)
  - Square root transformation → f(x)=$\sqrt{x}$
  - Reciprocal Transformation → f(x) = 1/x
  - Exponential or Power transformation → f(x)=exp(x) or f(x)=x^n most used is x^2 to reduce left Skewness
  - Box-cox transformation – evolution of exponential transformation. Value of λ ranges between -5 to +5

$$x_i^{(\lambda)} = \begin{cases} \dfrac{x_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(x_i) & \text{if } \lambda = 0, \end{cases}$$

Image Source : Internet

Feature Construction

- Feature construction is the process of creating new features and modifying existing features in the dataset
- Processes included in FC:
  - Continuous features can be binned in order to improve the model performance
  - Categorical features will have to be encoded to convert them in numerical features
  - Create new features from existing features – using domain knowledge on the given dataset

Binning continuous features

- Create a flag using threshold. For eg. If age is more than 60 then 1 else 0 etc.
- Create categories from the continuous features
- Example, binning age variable into different generations/decades/threshold depending upon the nature of business problem

Encoding Categorical Features

- Label encoding
- One-hot encoding
- Frequency encoding
- Mean (Target) encoding
- Rare label encoding
- Weight of Evidence (WoE) – categorical target

Column	Target	Target Mean	Target Mean (numerical value)
red	1	3/5	0.6
green	1	2/3	0.67
red	0	3/5	0.6
green	0	2/3	0.67
blue	1	2/4	0.5
red	0	3/5	0.6
red	1	3/5	0.6
blue	0	2/4	0.5
red	1	3/5	0.6
blue	0	2/4	0.5
blue	1	2/4	0.5
green	1	2/3	0.67

Note – category_encoders library in python

Feature – X (values)	Number of events	Number of non-events	Percentage events	Percentage non-events	WOE	IV
A	90	2400	90/490 = 0.184	2400/9510 = 0.25	Ln(0.184/0.25)=-0.3065	0.02
B	130	1300	130/490 = 0.265	1300/9510 = 0.14	Ln(0.265/0.137)=0.659	0.175
C	80	3500	80/490 = 0.16	3500/9510 =0.37	Ln(0.16/0.37)=0.838	0.176
D	100	1210	100/490 = 0.2	1210/9510 =0.18	Ln(0.2/0.18) = 0.105	0.002
E	90	1100	90/490 = 0.184	1100/9510 = 0.12	Ln(0.184/0.12)=0.427	0.026
Sum	490	9510				0.399

Image Source : Internet

Information Value (IV)

- Used for feature selection in classification problem. Represents a features' predictive power

$$IV = \sum_{i=1}^{h} \left(WoE_i * \left(percentage\ of\ events - percentage\ of\ non - events \right) \right)$$

$$where\ h\ represents\ the\ number\ of\ bins\ of\ categories\ in\ a\ feature$$

Information Value	Predictive power
<0.02	Useless
0.02 to 0.1	Weak predictors
0.1 to 0.3	Medium Predictors
0.3 to 0.5	Strong predictors
>0.5	Suspicious

Image Source : Internet

Dimension Reduction

- Dimension reduction is the process of reducing number of features in the dataset so that data mining algorithms can operate efficiently
- This step has some overlaps with feature selection process
- Dimension reduction can be done using following methods:
  - Domain knowledge
  - Automatic dimensionality reduction techniques such as PCA

Curse of Dimensionality

- Big data applications generally have too many features

- More number of features lead to too much noise in the dataset (useful models are not possible)→ patterns and structures are discernible → hence need reduction in number of features with minimal sacrifice of accuracy

- Dimension reduction is also known as Feature extraction or Factor selection

Principal Component Analysis (PCA)

- It is an unsupervised learning method
- PCA should be used when data are not independent
- It allows the data scientist to reorient the data so that the first few dimensions account for as much of the available information as possible
- It seeks to maximize the variance
- Each principal component is uncorrelated with all others
- Each principal component is an exact linear combination (i.e. weighted sum) of the original variables
- **Applications:**
  - Dimensionality reduction hence reducing the processing time of ML models
  - Identifying patterns of association among variables

PCA Intuition (1/4)

FIGURE 4.3
Stylized three-dimensional view of shape of distribution of X_1, X_2, and X_3. Shadows represent shapes in two dimensions

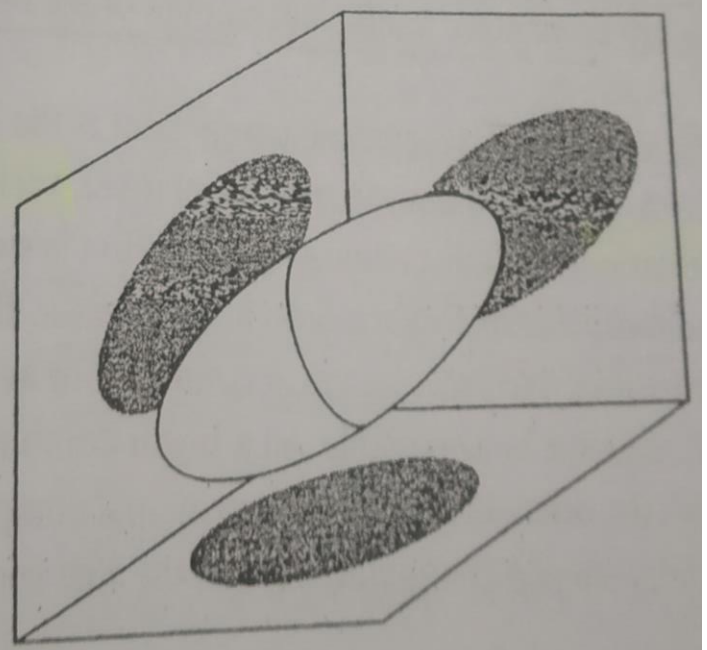


FIGURE 4.4
Three-dimensional scatter plot of actual values of X_1, X_2, and X_3

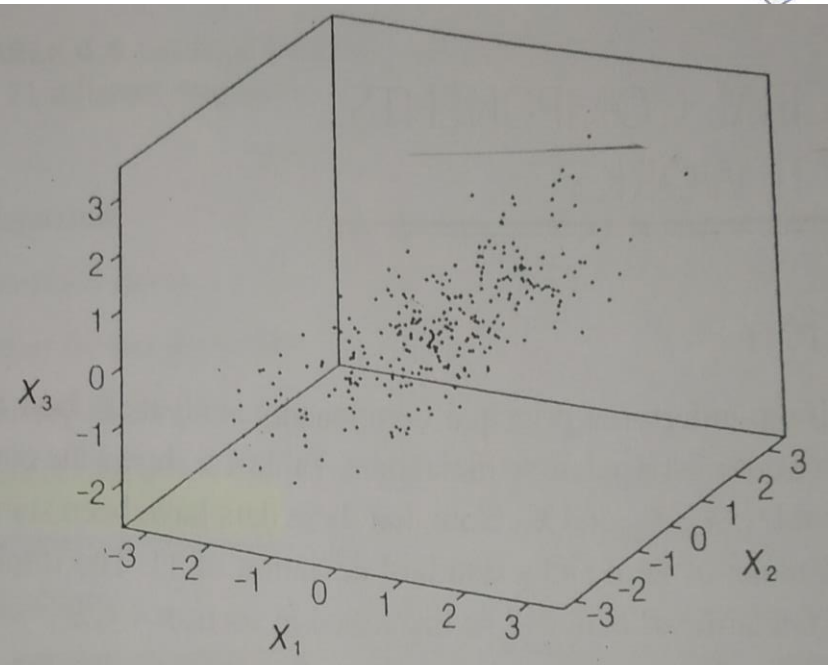


Image Source : Analyzing Multivariate Data Book by Lattin, Carroll & Green

PCA Intuition (2/4)

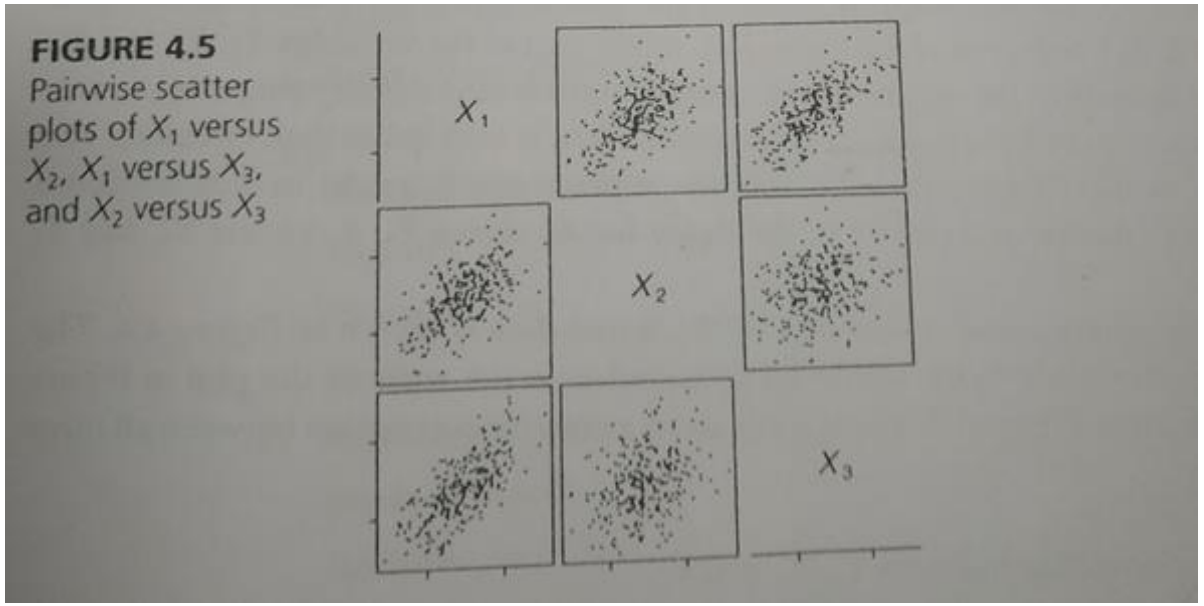


FIGURE 4.5
Pairwise scatter plots of X_1 versus X_2, X_1 versus X_3, and X_2 versus X_3

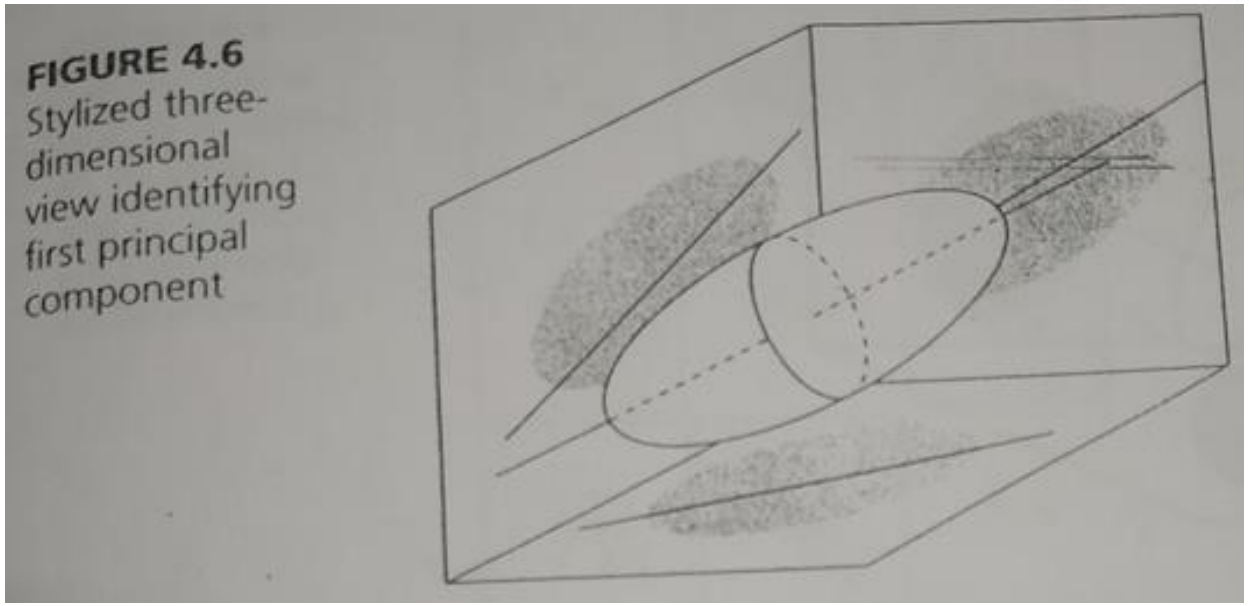


FIGURE 4.6
Stylized three-dimensional view identifying first principal component

Image Source : Analyzing Multivariate Data Book by Lattin, Carroll & Green

PCA Intuition (3/4)

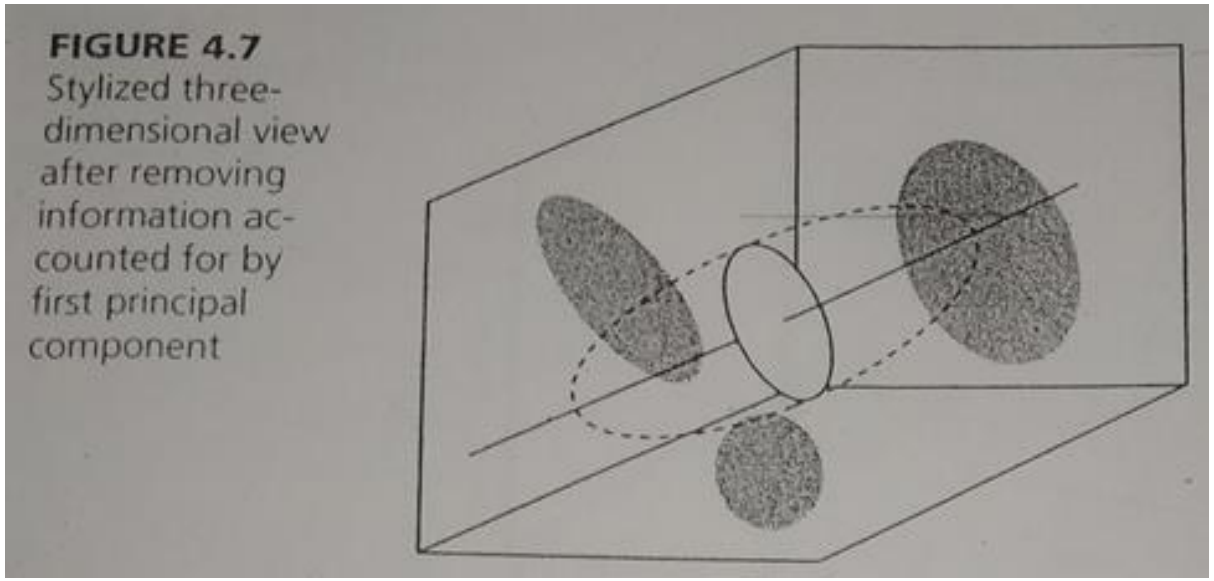


FIGURE 4.7
Stylized three-dimensional view after removing information accounted for by first principal component

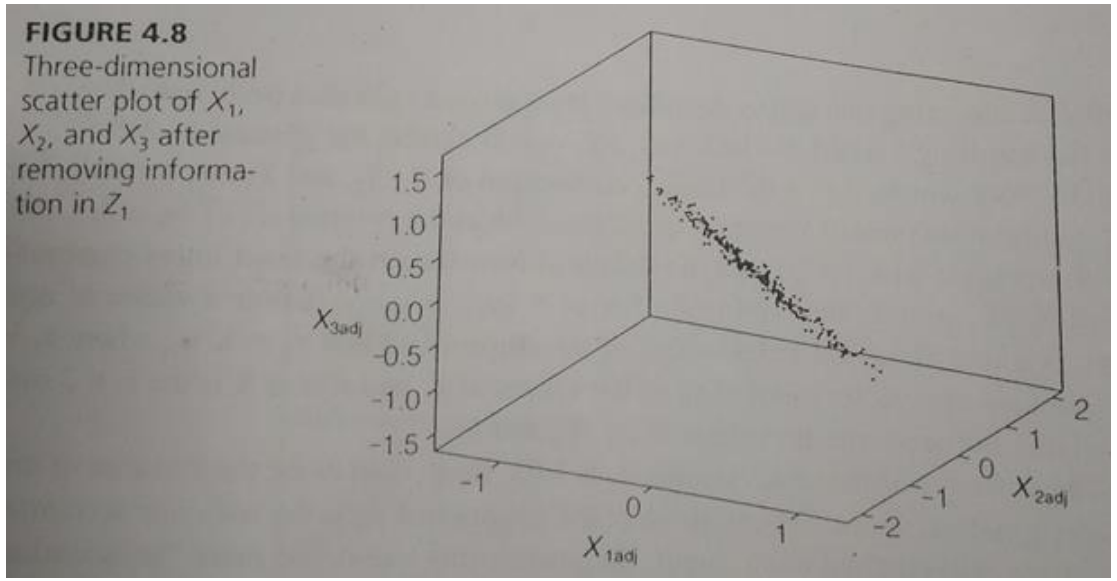


FIGURE 4.8
Three-dimensional scatter plot of X_1, X_2, and X_3 after removing information in Z_1

Image Source : Analyzing Multivariate Data Book by Lattin, Carroll & Green

PCA Intuition (4/4)

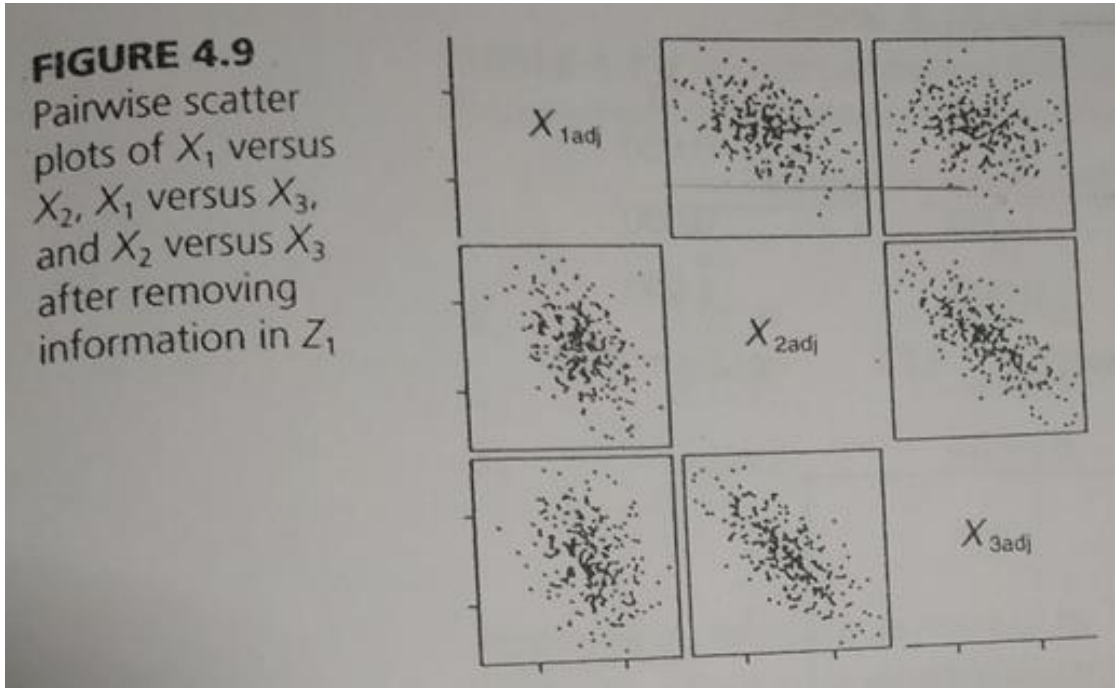


FIGURE 4.9
Pairwise scatter plots of X_1 versus X_2, X_1 versus X_3, and X_2 versus X_3 after removing information in Z_1

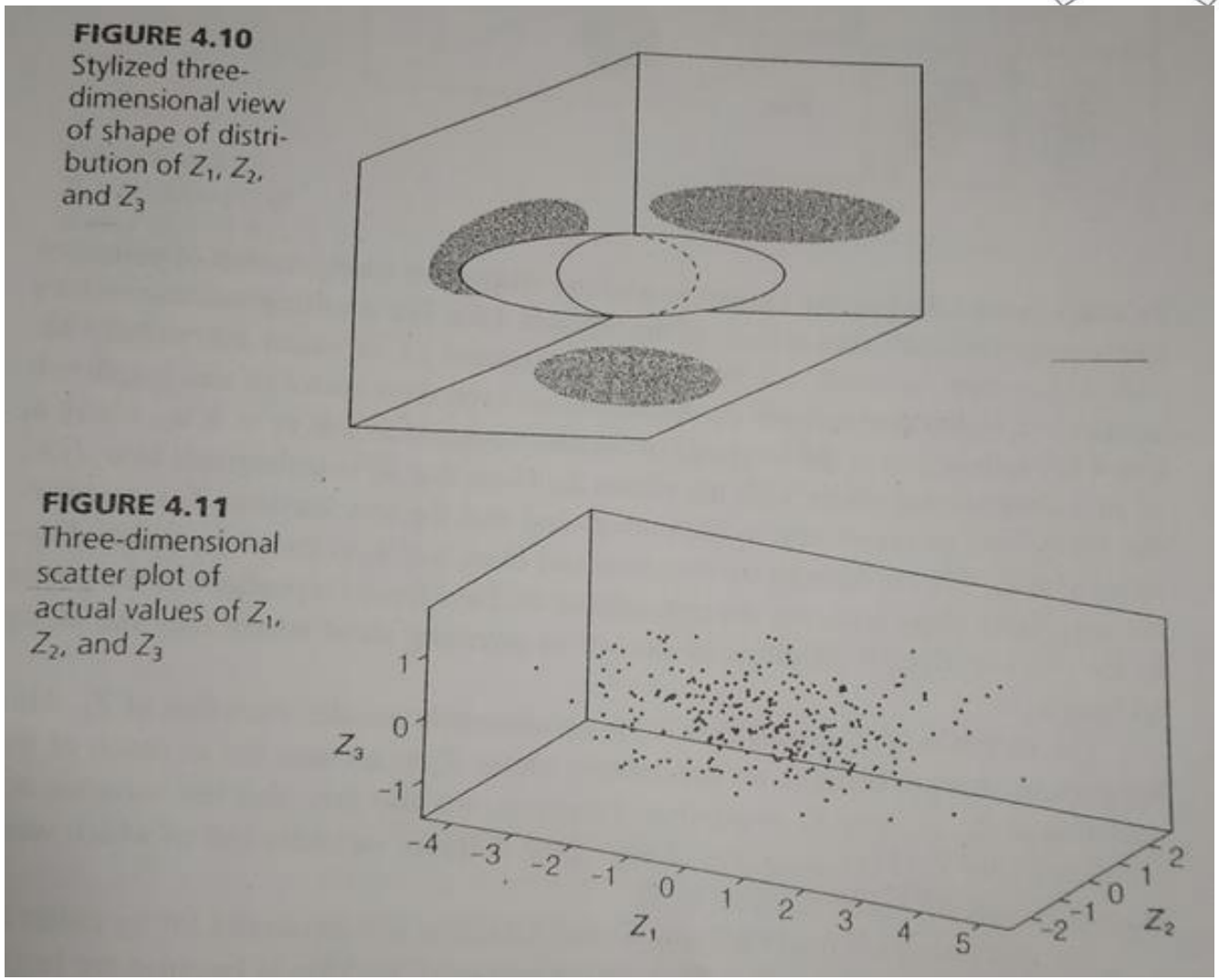


FIGURE 4.10
Stylized three-dimensional view of shape of distribution of Z_1, Z_2, and Z_3

FIGURE 4.11
Three-dimensional scatter plot of actual values of Z_1, Z_2, and Z_3

Image Source : Analyzing Multivariate Data Book by Lattin, Carroll & Green

PCA – Key Terminologies

- **Principal Component (z)** - is the linear combination of the original variables
- **Eigenvalue (λ)** –measures the variance of the principal component. By design, the solution is chosen so that the λ1>λ2>…..λp.
- **Proportion of Variation accounted by Principal components:** the sum of variances of all the principal components is equal to p, the number of variables. Hence the proportion of variation accounted by first c principal components is given by:

$$\sum_{i=1}^{c} \frac{\lambda_i}{p}$$

- **Principal Component Loadings** – correlation between the Original variables and the principal components (z)

Steps in PCA

- Standardize the features
- Develop Covariance matrix (input for PCA)
- Compute eigenvectors of covariance matrix
- Plot scree plot and decide the number of PC to be chosen
- Compute the explained variance
- Model validation

How to decide how many Principal Components?

- **Scree Plot** – retain all the principal components above the elbow in the plot given below
- **Kaiser's Rule** – retain all principal components which have eigenvalue > 1

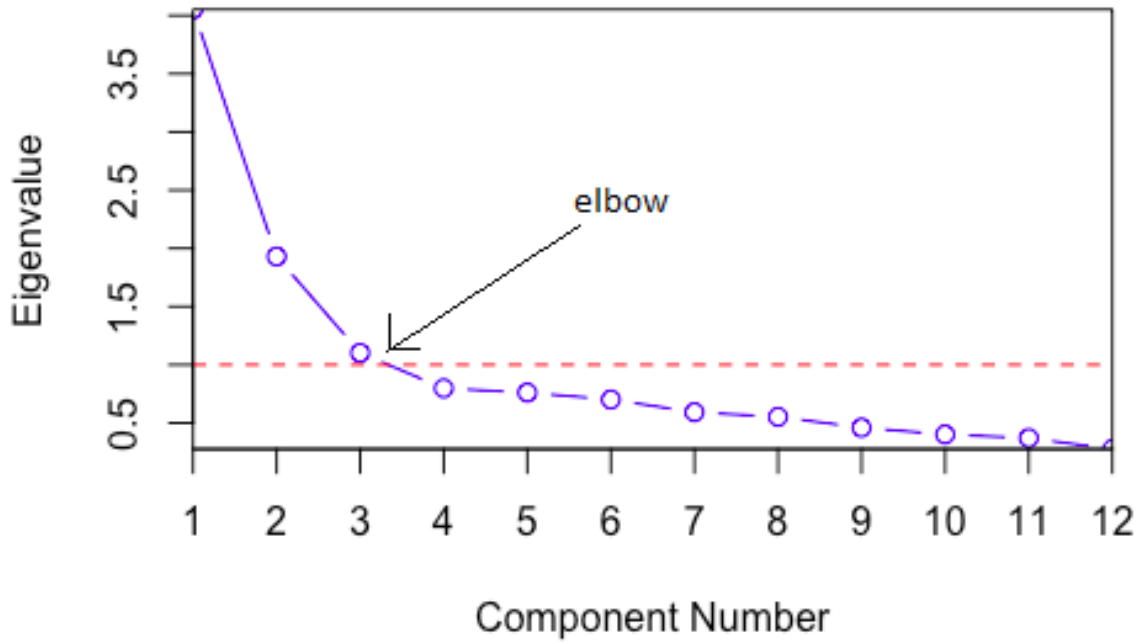


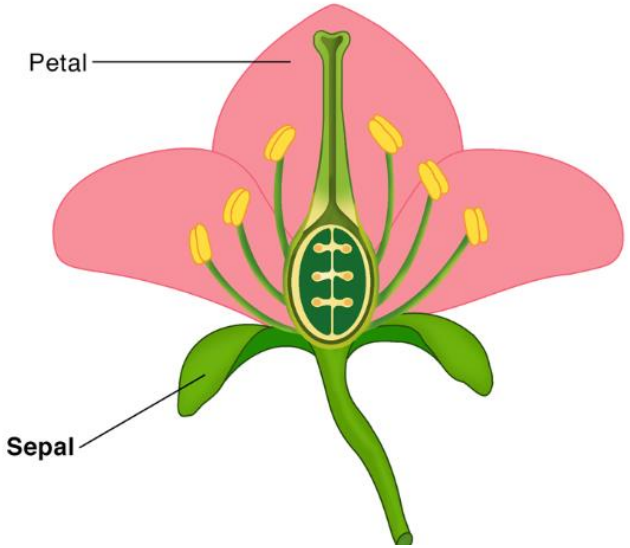
Image Source : Internet

PCA Model Validation

- Large dataset – Holdout sample
- Small dataset – 1. Jackknife Validation; 2. Bootstrapping

PCA Example

- IRIS Dataset



Iris setosa

Iris virginica

Iris versicolor

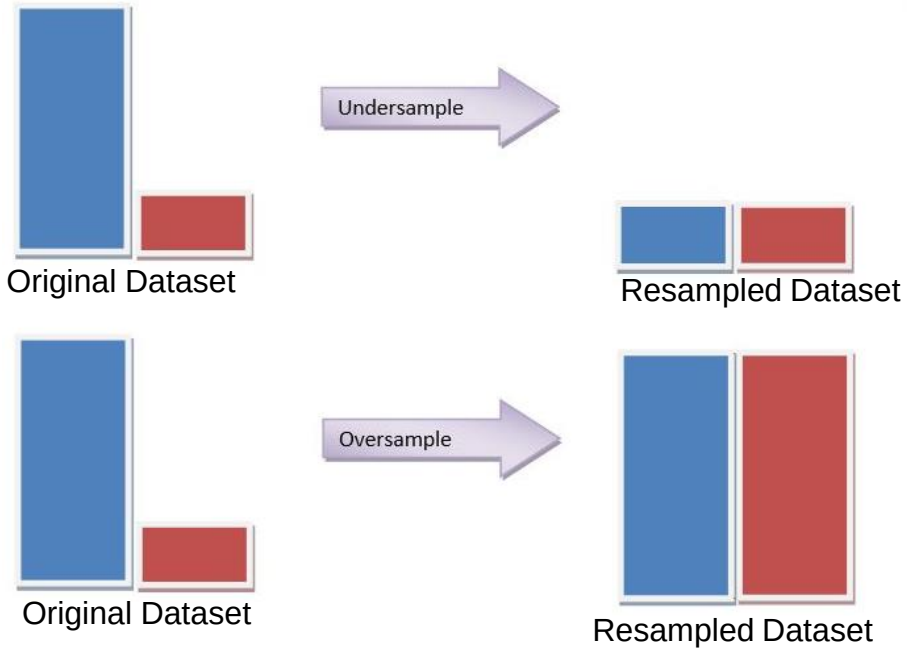
Image Source : Internet

Imbalanced Data Distribution

- The dataset is called as imbalanced when the observations in one of the classes are much higher or lower than the other classes
- It happens in classification problems
- Is an imbalanced data distribution cause for concern?

Imbalanced Data handling techniques

- **Data-level approach**
  - Under-sampling methods
    - Random undersampling
    - Tomek links (select eg. to delete)
    - Near-miss algorithm (select eg. to keep)
  - Over-sampling methods
    - Random oversampling
    - Synthetic Minority Oversampling Technique (SMOTE)
  - Oversampling followed by Undersampling
- **Algorithmic Ensemble approach**
- **Balanced Weight method**



Note – Imbalanced-learn library in python

Image Source : Internet

Undersampling methods: Tomek Links

- Tomek links are pairs of very close instances, but of opposite classes
- In this method, we remove the majority class from the Tomek link, which provides a better decision boundary for a classifier
- It will not produce a balanced dataset rather it will clean the dataset by removing Tomek links. It may result in easier classification problem

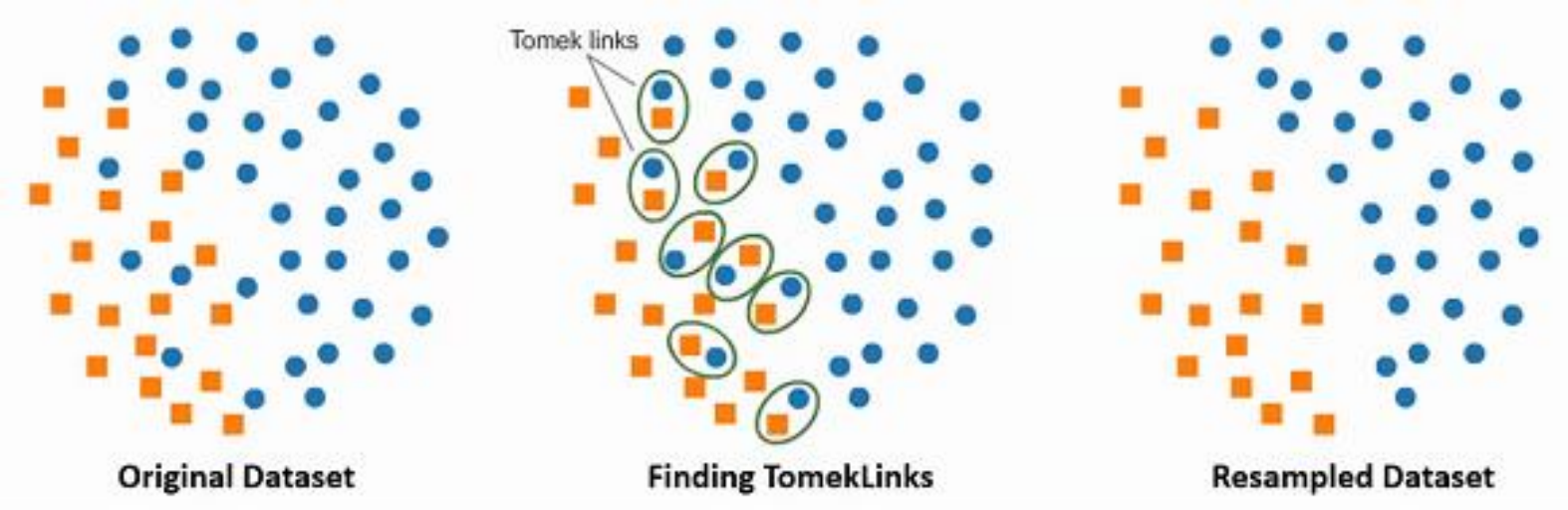


Image Source : Internet

Undersampling methods: Near-miss Algorithm

- Based on k-nearest neighbor algorithm
- The method starts by calculating the distances between all instances of the majority class and the instances of the minority class.
- Then k instances of the majority class that have the smallest distances to those in the minority class are selected to be retained.
- If there are n instances in the minority class, NearMiss will result in k × n instances of the majority class.

Image Source : Internet

Oversampling methods: SMOTE

- SMOTE – Synthetic Minority Oversampling TEchnique
- Synthesizes elements for the minority class, based on those that already exist
- It randomly choses a point from the minority class and computes **k-nearest neighbors** for this point
- Further, synthetic points are added between the chosen points and its neighbors
- Steps -
    - Randomly pick a point from the minority class.
    - Compute the k-nearest neighbors (for some pre-specified k) for this point.
    - Add k new points somewhere between the chosen point and each of its neighbors.

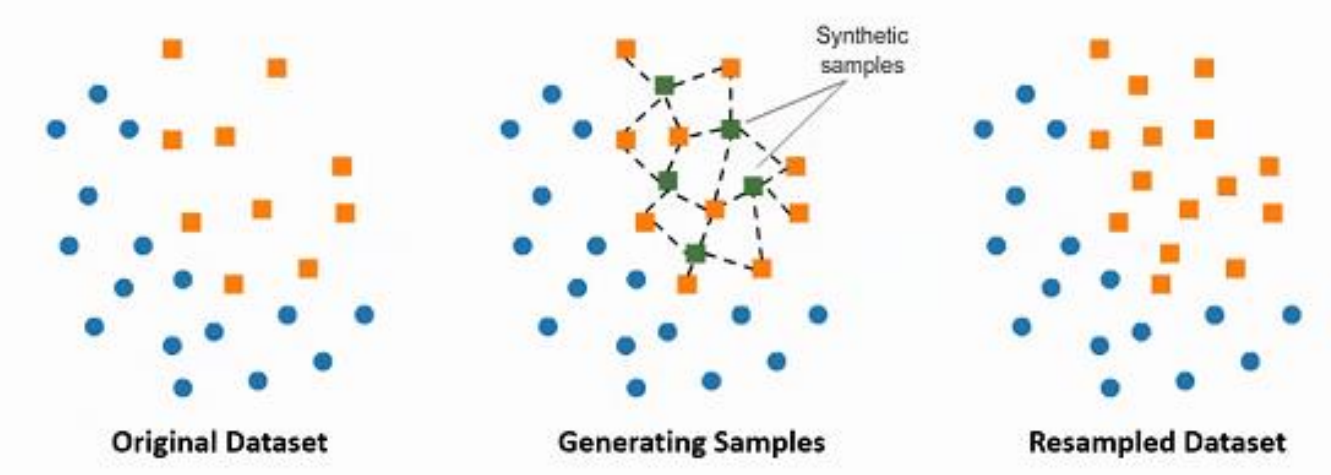


Image Source : Internet

Oversampling followed by Undersampling

- It uses an undersampling method (Tomek) with an oversampling method (SMOTE), called as SMOTE-Tomek links
- It combines the SMOTE ability to generate synthetic data for minority class and Tomek Links ability to remove the data that are identified as Tomek links from the majority class

Algorithmic Ensemble Approach

- Develop several classifier models from the original model dataset and then combine their predictions

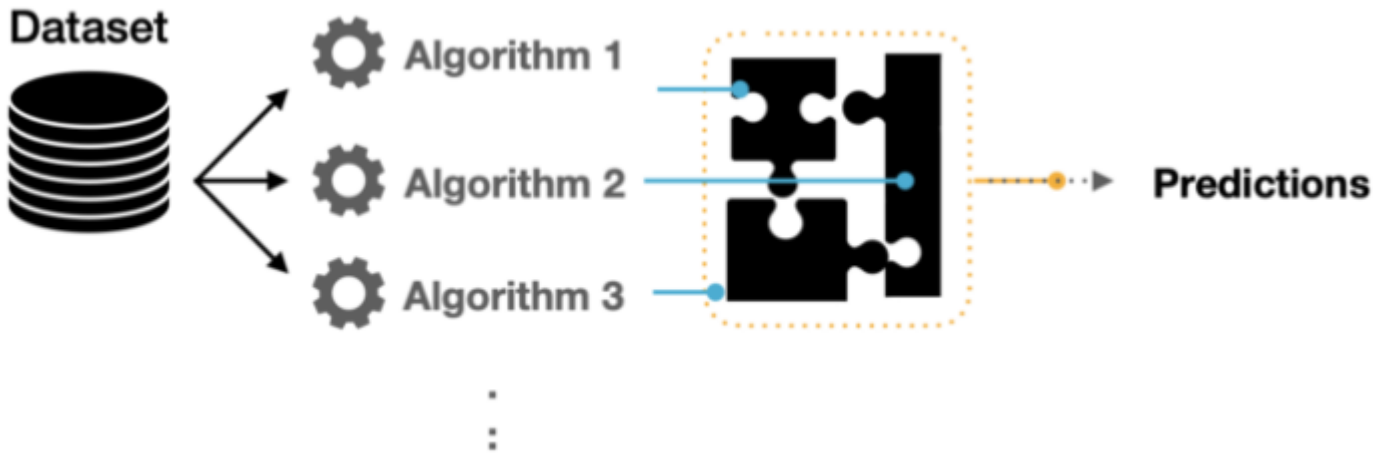


Image Source : Internet

Balanced Weight

- It is one of the most widely used methods for imbalanced classification models
- It modifies the class weights of the majority and minority classes during the model training process to achieve better model results
- This method does not modify the majority and minority class ratio
- Instead, it penalizes the wrong predictions on the minority class by giving more weight to the loss function

Thank you!