3 - Offline Testing of AI Systems

Offline testing method

- The cycle of trying out different combination of algorithms with different hyperparameters is crucial to the train-test method.
- Different algorithms with different hyperparameters (parameters influencing an algorithm) are applied on training data and benchmarked against test data.
- Metrics are computed to judge the best algorithm & hyperparameter combination.
- Once baselined, this model is then moved to deployment, either as a standalone AI system or as part of the bigger system.

3.1 Data Preparation and Preprocessing

3.1.1 Steps of Data Preparation and Preprocessing

Input data could be in the form of database tables, CSV (comma separated values) files, or it could be unstructured data such as images, audios, videos or running texts. The required data is acquired from various sources, internal and external.
After acquisition, the data needs thorough cleanup and some processing before it can be fed to the algorithms for training and testing.
The following steps are performed for data preparation and preprocessing

- Data manipulation
- Data filtering
- Data preprocessing activities include
  - Data Imputation, i.e. Dealing with Missing Values
  - Data visualization to get a big picture and Treating Anomalies and Outliers
  - Correlation Analysis and Reducing Dimensions

Several data format (e.g., image, text) specific preprocessing steps need to be performed to make data format suitable for training. When the data volume is too large, perform data-reduction without losing the information. For structured data missing, data values may need to be filled. All such data preprocessing steps are required to obtain desired accuracy and better predictability in the models.

3.1.2 Data Preparation

Data preparation includes:

1. **Data manipulation**: Changing the structure of the given data for example, adding a new column, dropping some rows, etc.

2. **Data filtering**: Reducing the size of both structured (table/matrix) and unstructured (image/text) data for improving data quality.

Data Filtering for Structured Data

- Data filtering allows to reduce the amount of data needed for processing
  - Columns that have unique values for each record (like, row-ID) don't contribute either
  - Columns that have the same values for each record don't contribute either
- Data filtering can be done by
  - Rows
  - Columns
  - Combination of Rows and Columns
- Done by specifying conditions for filtering

3.1.3 Processing of Unstructured Data (Images)

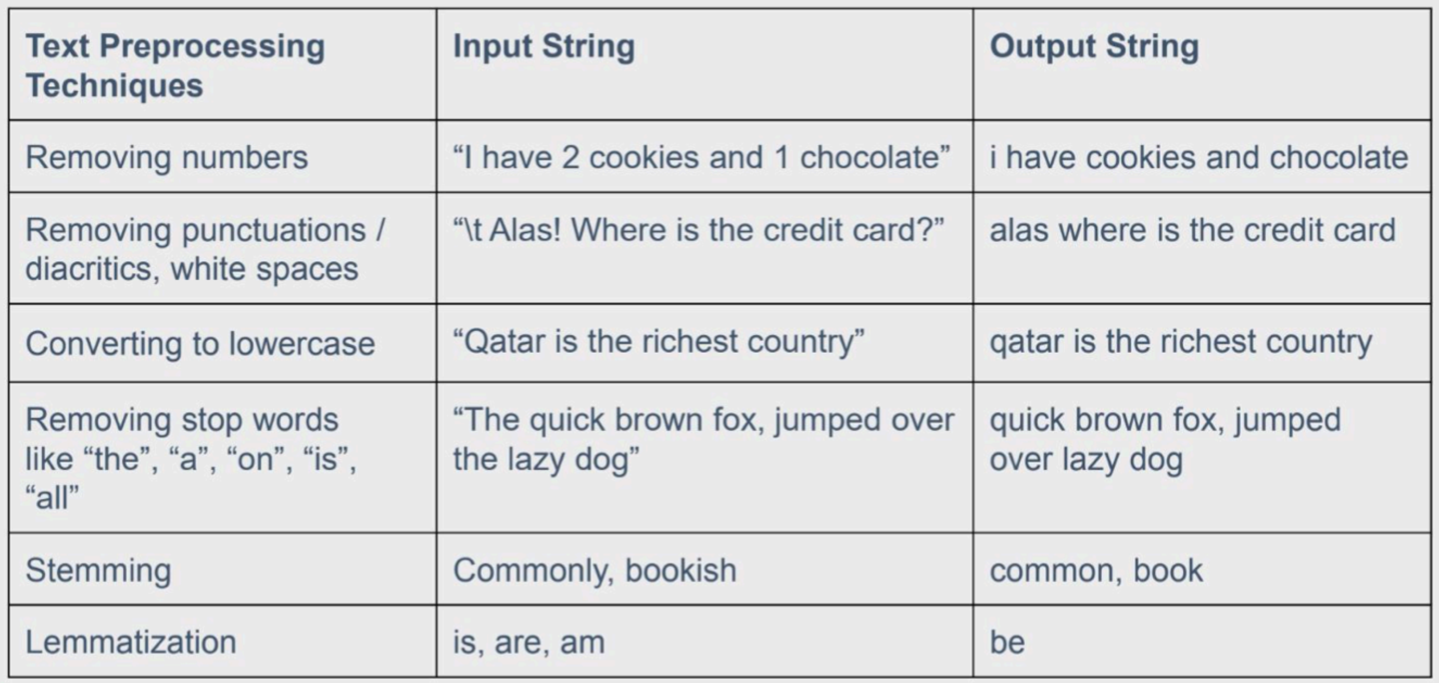
Removing noise from the image and resizing it are the common operations carried out on images for designing computer vision algorithms. [UDI]

Data Preprocessing for Unstructured Data (Images)

- Image preprocessing steps:
  - Resize Image
    - All images input to AI model should be of same size.
    - Common requirements in image processing.
  - Remove Noise and add blur
    - Noise removal is used as pre-processing stage in computer vision algorithms to enhance image structures at different scales.
    - Blurring reduces the focus on unimportant features and thus, alleviates the importance/weightage of the key aspects of an image.
    - It allows for the image to look smother.

3.1.4 Processing of Unstructured Data (Text)

Text data preprocessing can be done in multiple steps of syntactic changes depending on the need of the ML model. For example, removing numerals, conversion of uppercase to lowercase, removing punctuations, white spaces, removing stop words, perform stemming/lemmatization, etc. [UDT]



Text Preprocessing Techniques	Input String	Output String
Removing numbers	“I have 2 cookies and 1 chocolate”	i have cookies and chocolate
Removing punctuations / diacritics, white spaces	“\t Alas! Where is the credit card?”	alas where is the credit card
Converting to lowercase	“Qatar is the richest country”	qatar is the richest country
Removing stop words like “the”, “a”, “on”, “is”, “all”	“The quick brown fox, jumped over the lazy dog”	quick brown fox, jumped over lazy dog
Stemming	Commonly, bookish	common, book
Lemmatization	is, are, am	be

Stemming and lemmatization

- Stemming is the process of reducing words to their word stem, base or root form, by removing prefixes and suffixes, for example:
  - Books => book, booking => book
  - Running => runn, dining => din
- Lemmatization, unlike stemming, uses lexical knowledge bases to get the correct base forms of words i.e., dictionary-compilation form, canonical form
  - e.g., is, am, are, been, being => be
  - Running => Run, dining => din
- Stemming and lemmatization are alternative to each other, serving to the same purpose – “finding root word”

3.1.5 Data Imputation

Data collected from the field may have null or missing values requiring replacing null values with some appropriate values. Null or missing values can be imputed with measures of central tendency (mean, median or mode), K- Nearest-Neighbor [DI1] method, or a regression-based approach.

Different approaches

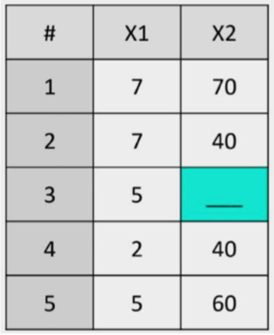
- Imputation by Measures of Central Tendency
- Imputation by Mean of Column
- Imputation by Median of Column
- Imputation by Mode of Columns (Categorical Variables)
- Nearest Neighbor Methods
- KNN Based Imputation
- Regression based approach
- Hot Deck approach, etc.

Imputation Approach Selection

- Mode – for discrete/ordinal/categorical values based column
  - E.g. for ‘Outlet_Size’ column as it has ordinal/categorical values like Small, Medium, High
- Mean – for continuous values based column
  - E.g. for ‘Item_Weight’ column as it has continuous data values
- KNN – for data that are continuous, discrete, ordinal, and categorical (i.e., all kinds of missing data).
  - E.g. KNN fixed both discrete columns (‘rad’ and ‘tax’) as well as continuous one (‘rm’ column).
  - It is preferred over mean / mode when the data value range is high, i.e. (max – min) is high

KNN explained

- KNN – how k-nearest neighbors are found by the algorithm (internally)
- Let us use KNN to fill missing a cell (row-3, column X2)
- For instance, find three nearest neighbors between rows
  - Use the X1 column to find the distance between rows
    - Dist(row3,row1) = | 5 – 7 | = 2
    - Dist(row3,row2) = | 5 – 7 | = 2
    - Dist(row3,row3) = | 5 – 2 | = 3
    - Dist(row3,row4) = | 5 – 5 | = 0
  - So, row1, row2 and row5 are 3-NN to row3
  - Fetch X2 column values for row1, row2 and row5 and take mean = (70 + 40 + 60) / 3 =56.66



#	X1	X2
1	7	70
2	7	40
3	5	___
4	2	40
5	5	60

3.1.6 Data Visualization

Visualizing the data helps understanding its structure and the relationship among its attributes, which is not possible by merely looking at the numbers or text provided. There are various types of visualizations. The most commonly used visualization methods are: line plots for continuous values, histograms for discrete ones, box plots, bar charts, pie charts, etc. They give meaningful insight of the available data at the first go.

Types of plots:

Univariate: The simplest form of analysis, in which the data being analyzed is a single variable. For example, the age of a population or the weight of a population, etc. They are analyzed individually and their relationship is never considered. Line plots, histograms, frequency distribution, bar charts and box plots are utilized for the analysis of univariate data.

Bivariate: This kind of analysis is carried out to find the relationship between two variables in the given data set. Plotting one variable against another on a XY plane helps to find the first hand relationship between two variables. For example, the relationship between the age and weight of the population under consideration. For this kind of analysis, you can use scatter plots or correlograms.

Multivariate: The analysis of three or more variables. Mesh plots and 3D plots are some of the ways one can visualize multivariate data and discover relationships among them.

3.1.7 Anomaly/Outliers Detection

Observations which do not follow the expected pattern for a given data set fall into the category of outliers, for example, fraud detection or hack attacks.
If outliers are not frequent and do not contribute in the critical events, then they can be removed. But in practice, they should be thoroughly investigated before pruning them from the dataset.

Causes of Outliers

- Error: Outliers in this case are outcomes of an error in the measurement, data entry and sampling, e.g. the temperature data recorded in Celsius for most records but for a few others, in Fahrenheit, by mistake.
- Natural: Some outliers can occur in a natural situation, e.g., if a flood incident occurs once in 100-years, it is a natural outlier.
- Intentional: Dummy outliers made to validate detection methods, e.g., lab- grown artificial records used for testing corner scenarios.

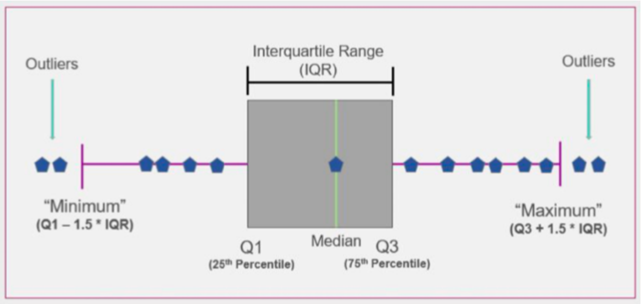
3.1.8 Outliers Detection Techniques

While computing various statistics with the given data set, often it is helpful to make use of a visual box-plot to see the data distribution. Box-plots help determine the outlier position of the given data set.

The ends of the box are the upper and lower quartiles. The whiskers are the two lines outside the box that extend to the highest and lowest data thresholds beyond which datapoints are considered to be outliers.

Box Plot

- Box and Whisker plot for a single variable
- Box between Q1 and Q3 (quartiles*)
- Whiskers denote thresholds:
  - Left whisker: Q1 – 1.5 * IQF
  - Right whisker: Q3 + 1.5 * IQR
  - Inter-quartile range: (*IQR = Q3 – q1)



*Note:

- Median is the value of the middle element of a sorted list
- Median divides the list into two halves:
  - Q1 (first quartile) – middle element of the 1st half
  - Q3 (third quartile) – middle element of the 2nd half

3.1.9 Dimensionality Reduction

ML problems often have large numbers of input features but not all of them contribute to the classification or regression output. The higher the number of features, the harder it is to visualize the training set. The technique of reducing the number of variables under consideration is called dimensionality reduction.

The need to work with fewer dimensions than the original dimensions arises from:

- Cost and speed factors
- Memory requirement
- Avoiding redundancy
- Identifying the most relevant part of the data for further processing

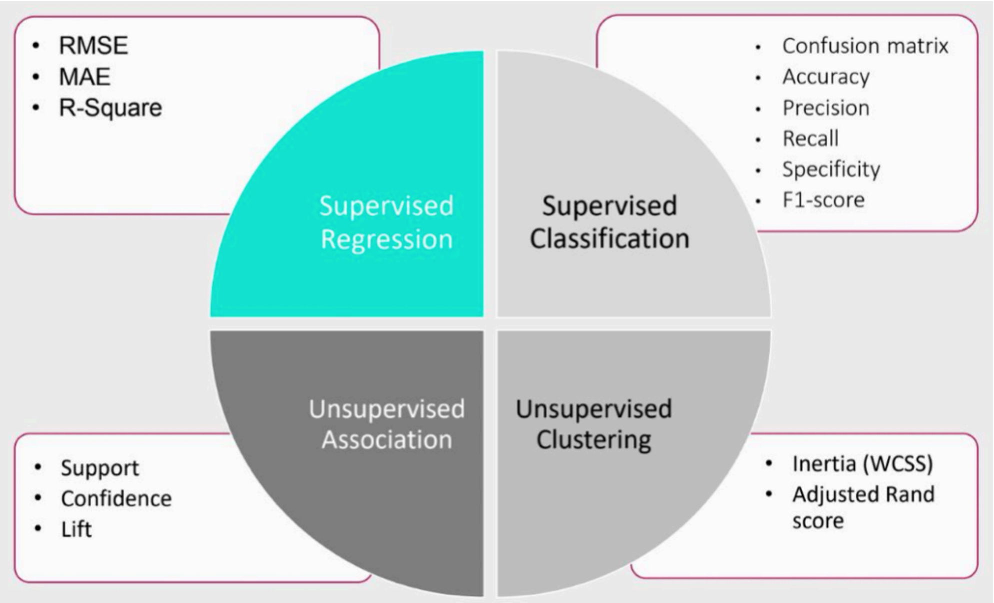
Methods of Dimensionality Reduction:

- Irrelevant feature elimination is removing columns that are not contributing to the output variable:
  - Univariate analysis helps to remove the columns whose value does not change across rows. For instance, consider the sales transaction data from a single retail store, then columns such as store name, store location would not change across rows and should be removed.
  - Too low-density data can easily be dropped after investigation to save space. For example, database row ID based columns have a unique value for every row and should be removed.
- Bivariate analysis removes one among the pair of highly correlated input attributes (thus, it is also known as correlation analysis), for instance, in a store inventory data, 'item price' and 'item quantity' are highly correlated attributes.
- Principal component analysis: PCA reduces the dimensions of larger datasets extensively, yet preserves the information to the maximum. It deduces a new set of

independent variables (called principal components) and puts them in order of reducing significance. The required number of top principal components (thus, reduced number of variables or dimensions) can then be selected still preserving the maximum possible information of the original dataset.

3.2 Metrics

Common metrics



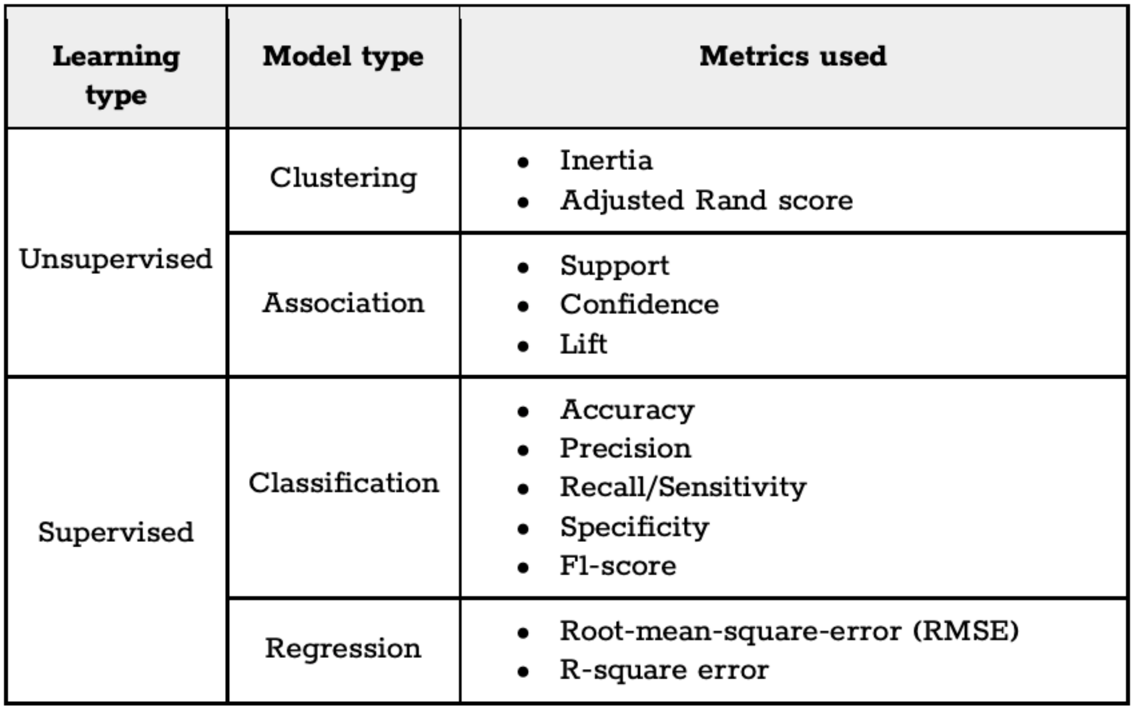
3.2.1 Role of Metrics

Metrics are evaluation parameters for a trained model and can be seen as the measurement of how far the trained model delivers accurate and reliable results. For a given type of algorithm, metrics can be used to compare trained models with each other.

- Metrics offer an objective means to verify model effectiveness using test datasets.
- Choosing appropriate metrics is a key to verification. The wrong metrics selection can result in the deployment of an unsuitable model.
- Each Machine Learning model is trying to solve a problem with a different objective, using a different dataset and hence, it is important to understand the context before choosing a metric.
- The responsibility of selecting appropriate metrics today lies with the developers, however,
  - The roles of dev and test are evolving in AI.
  - Testers with appropriate knowledge are likely to be involved at this stage.

3.2.2 Metrics for Supervised and Unsupervised Learning

The problem objectives for supervised and unsupervised learning models are different. Thus, the metrics to evaluate the models are different.



Learning type	Model type	Metrics used
Unsupervised	Clustering	• Inertia • Adjusted Rand score
	Association	• Support • Confidence • Lift
Supervised	Classification	• Accuracy • Precision • Recall/Sensitivity • Specificity • F1-score
	Regression	• Root-mean-square-error (RMSE) • R-square error

3.2.3 Inertia and Adjusted Rand Score

For an unsupervised clustering-based model, inertia or WCSS (within-cluster- sum-of-squares) is the average spread of a cluster across all the discovered clusters. The smaller Inertia value means better clustering, as it means that the data points within a cluster are closer to each other. The cluster size (and thus, the value of Inertia) will go down naturally as the number of clusters goes up. However, Inertia stops decreasing significantly beyond a certain number of clusters; this point shows the optimum value for Inertia and the number of clusters for a given dataset – this method is known as the elbow method.

When the actual values of the labels are available for every data point, the **adjusted Rand score** is preferred over Inertia. It is a measure of similarity between the cluster assignments (by the model) and the actual separate classes.

3.2.4 Support, Confidence and Lift metrics

Association Rule Mining

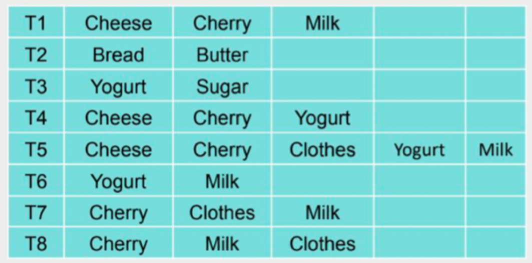
- It helps answer questions such as
  - Which features occur together
  - Which features are correlation
  - An example – People who buy battery operated devices are likely to buy batteries.
- Good metrics help evaluate the correction, relevance, importance of the association.
- Some of these metrics are:
  - Support
  - Confidence
  - Lift

Support for an itemset measures how frequently it is appearing in transactions. For example, if the item 'bread' is present in 7 out of 10 total transactions at a retail store, its support is 70%.

- Support – the likelihood of an itemset X to appear in a set of transactions
- Support S(X) = (No. of transactions with X) / Total no. of transactions

Example: X = {Cherry, Clothes, Milk}

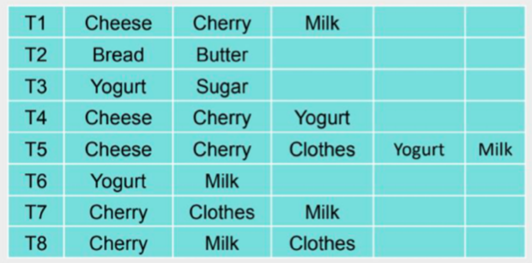
- Txn's having {Cherry, Clothes, Milk}:
  - T5, T7, T8
- S(X) = 3/ 8



T1	Cheese	Cherry	Milk		
T2	Bread	Butter			
T3	Yogurt	Sugar			
T4	Cheese	Cherry	Yogurt		
T5	Cheese	Cherry	Clothes	Yogurt	Milk
T6	Yogurt	Milk			
T7	Cherry	Clothes	Milk		
T8	Cherry	Milk	Clothes		

Confidence measures the likelihood of itemset Y appearing, given that X has appeared.

- Confidence measures the likelihood of Y happening, given X has already happened.
- Expressed as (X -> Y) i.e., X causes Y
- Confidence (X -> Y) $= \frac{\text{\#Txn (X and Y)}}{\text{\#Txn (X)}} = \frac{\text{Support (X and Y)}}{\text{Support (X)}}$
- Confidence is directional i.e.,
  Confidence (Yogurt -> Milk) ≠
  Confidence (Milk -> Yogurt)



T1	Cheese	Cherry	Milk		
T2	Bread	Butter			
T3	Yogurt	Sugar			
T4	Cheese	Cherry	Yogurt		
T5	Cheese	Cherry	Clothes	Yogurt	Milk
T6	Yogurt	Milk			
T7	Cherry	Clothes	Milk		
T8	Cherry	Milk	Clothes		

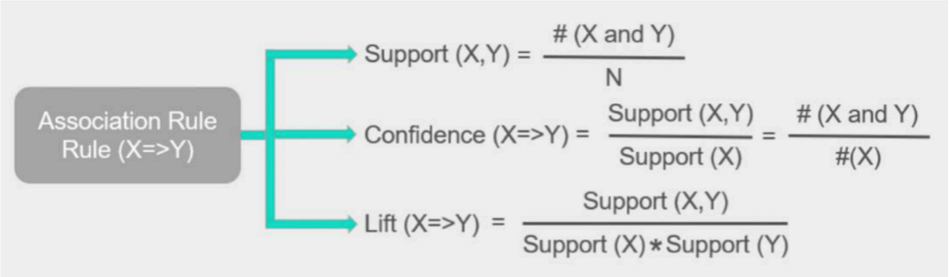
Example: Buying milk when yogurt is bought
#Txn's (Yogurt and Milk together) = 2
#Txn's (Yogurt) = 4
Confidence C(Yogurt -> Milk) = 2/4 = 50%

Lift is used to eliminate scenarios where two itemset are occurring together very frequently (thus, the confidence metric value will be high). However, the two itemset may not have any interdependency. Besides, this metric can reveal if more occurrence of itemset X means more or less occurrence of itemset Y (i.e., positive or negative association between X and Y).

- Examples of various kinds of associations:
  - The sale of "shoes" is almost the same as that of "shoes and bread" together, which means that the sales of these items ar not affected by each other, i.e., no association.
  - The sale of "milk powder alone" is higher than that of "milk-powder and milk" together, i.e. negative association (harming each other's sale).
  - The sale of "butter and bread" is higher than "butter alone", i.e. there is a positive impact by the sale of bread on butter's sale, i.e. bread => butter.
- The above associations can be detected and measured by metric called lift; e.g., how the sale of butter is lifted by the sale of bread.
- Sometimes, two items can occur together even if they have no association (e.g., sale of shoes & bread).
- Lift is used to eliminate such cases:
  - L = 1 means no association; X and Y are independent
  - L > 1 means positive association; More occurrence for X means more occurrence for Y and vice-versa
  - L < 1 means negative association; More occurrence for X means less occurrence for Y and vice-versa
- Lift is computed by using formula: $L = \frac{S(X, Y)}{S(X) * S(Y)}$

Summary of Association Rule Mining Metrics

- Using only one or two of the metrics is likely to yield incorrect results.
- The Inferences need to be evaluated on the basis of all the metrics mentioned.
- Note: There are many more metrics available



Association Rule Rule (X=>Y)

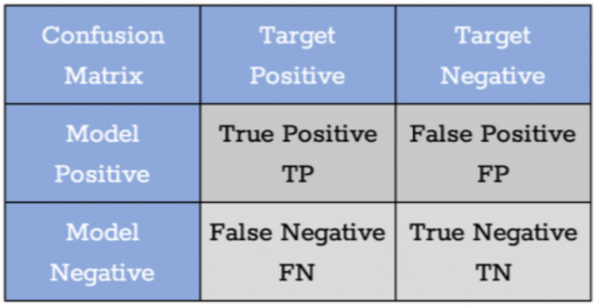
$$\text{Support }(X,Y) = \frac{\#\,(X \text{ and } Y)}{N}$$

$$\text{Confidence }(X\Rightarrow Y) = \frac{\text{Support }(X,Y)}{\text{Support }(X)} = \frac{\#\,(X \text{ and } Y)}{\#(X)}$$

$$\text{Lift }(X\Rightarrow Y) = \frac{\text{Support }(X,Y)}{\text{Support }(X) * \text{Support }(Y)}$$

3.2.5 Confusion Matrix

Supervised classification metrics are computed using a confusion matrix made- up of the counts of true positives, false positives, true negatives, false negatives.



Confusion Matrix	Target Positive	Target Negative
Model Positive	True Positive TP	False Positive FP
Model Negative	False Negative FN	True Negative TN

3.2.6 Accuracy, Precision, Recall, Specificity and F1-Score

The **Accuracy** of a model shows what percentage or fraction of total classifications were done accurately by the trained model on a test dataset. This metric becomes a poor choice of metric if one class of data dominates over the others.
Accuracy = (TP + TN) / (TP +TN + FP +FN).

Precision measures how accurately the model classifies true positives. Precision = TP/(TP+FP).

Recall measures how far the model failed or missed to detect the positives. Recall = TP/(TP+FN).

In order to minimize false positives, high precision is required, whereas to minimize false negatives, recall should be high.

Like precision but opposite to recall, **specificity** measures how accurately the model classifies true negatives.

Specificity = TN / (TN + FP)

F1-score is computed as the harmonic mean of precision and recall. A low F1- score represents the poor quality of the model at detecting positives. F1-score will have a value between 0 and 1. Close to 1 means that there is good quality and no false data disturbing the result.

3.2.7 RMSE and R-Square

The supervised regression model metrics represent how well the regression line fits the actual data points.

RMSE (root-mean-square-error) is a measure of how far the data points are from the regression line. It is measured as the standard deviation of prediction errors. The value of RMSE changes if the same dataset is measured in a different unit.

R-square is a measure of how better the predictions by the regression line are compared to using mean as a predictor. Its value ranges from 0 to 1 and is independent of the unit used to measure data points.

3.3 Model Evaluation

The metrics values depend on how the datapoints for training and validation are chosen. Thus, it becomes the key aspect for model evaluation that datapoints for training and validation are selected in a completely unbiased way.

3.3.1 Training, Validation and Testing Datasets

- Post training, an ML model is validated by computing model metrics against a validation dataset (different from its training dataset).
- The dataset for training, validation and testing phases must come from the same or similar source(s) to ensure the effectiveness of metrics.
- A model trained using one kind of dataset may perform very poorly on a dataset originated from very different source(s).
- In order to rule out anomalies due to different data sources, one can split the available dataset on three brackets:
    - Training set
    - Validation set
    - Test set

The training dataset contains the data that the model is trained on. The validation dataset is used by the machine learning algorithm to evaluate if the training was effective. In every run of ML (which is a process of many iterations), the training dataset and validation dataset are combined again and split in a different way so that the algorithm uses different combinations of data to learn from.

The testing dataset is a separate dataset that is used after the ML process is finished to validate whether the algorithm has been adequately trained. The testing dataset should not be used during the training process. [Wiki1]

In the post-training phase, an ML model is evaluated and tested with a dataset different from its training dataset. However, the dataset for training, validation and testing phases must come from the same or similar source(s) to ensure the effectiveness of metrics. A model trained using one kind of dataset may perform very poorly on a dataset originated from very different source(s).

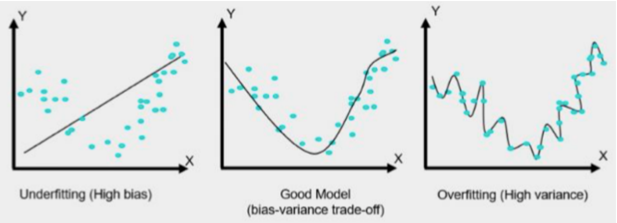
3.3.2 Underfitting and Overfitting

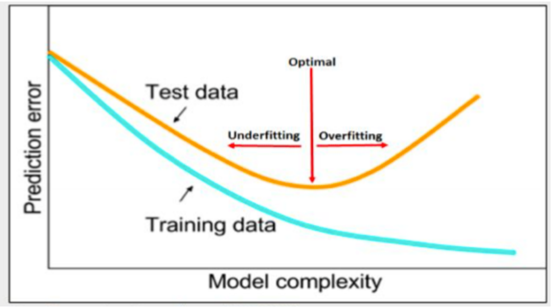
If a supervised ML model is too simplistic to fit the training data points (i.e., it fails to represent the data trend) it is an example of underfitting. Conversely, an overfitting model tries to fit the training data points too much and this often results in poor prediction accuracy during the subsequent validation or testing phases.

The underfitting and overfitting nature of a model can also be explained in terms of Bias and variance errors. If a model is oversimplified and does not learn from all the provided features to represent, it is said to have high-bias and suffers from poor prediction accuracy.

If the model prediction performance varies highly by changing the training dataset slightly, it is said to be a model of high variance (too dependent on the training dataset). A good model has to achieve low bias and low variance. This is known as the bias-variance tradeoff.

- Underfitting – The model misses the underlying data pattern (high bias).
  - Scarcity of data
  - Too few parameters
  - Using a simple model for complex data
- Overfitting – When a model fits the training data too well (high variance).
  - Because of the noise the data or
  - The model is complex, with many parameters
  - Leads to poor generalization
- A trade-off is required





Bias or Variance trade-off

1. Although train-test combination offers a reasonable model with predictive capabilities, issues often arise in terms of bias or variance.
2. There is a risk of getting either high bias or high variance if we rely on only one training test combination.
3. Hence, some minor variations are suggested to the train-test model in order to guarantee a better handling of these risks.
4. These variations are revealed in multiple passes of train and test.
5. Commonly used techniques are:
   - Bootstrap
   - K-fold Cross Validation
   - Leave-one-out Cross Validation

3.3.3 Cross-validation methods

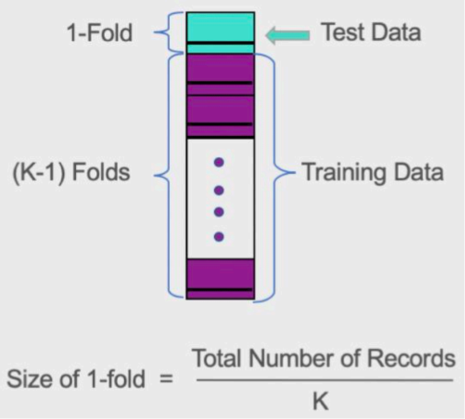
The way the available dataset is split into training and validation datasets may lead to high bias or high variance. In order to overcome this, multiple split combinations must be tried out before concluding model metrics. Some useful methods are split-test, bootstrap, K-fold cross-validation and leave-one-out cross-validation. Each of these methods repeats the training and validation process several times and the model performance is averaged out across all runs.

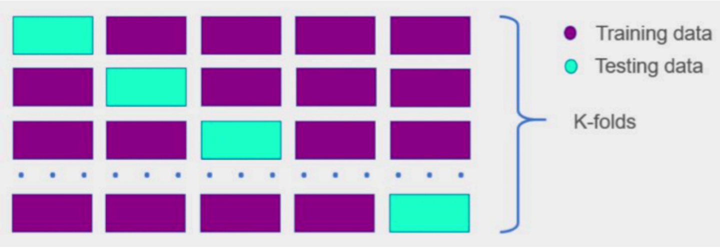
Split-test divides the data into parts for training and testing datasets but in each iteration in different ratios. This helps revealing how different splits can produce different results.

Bootstrap works by picking random data points from the entire dataset for training and uses the remaining dataset for validation.

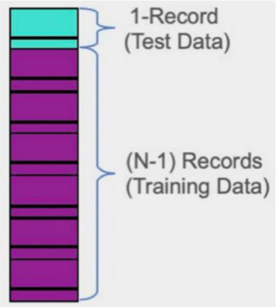
K-fold cross-validation divides the whole dataset into k-folds. Keep one subset for validation and use rest (k-1) subsets for training the model. Repeat this process k-times, by taking a different subset for validation in each iteration.

- Thus, k-different model instances are validated and the performance of the model is averaged out over k-runs.





Leave-one-out cross-validation is similar to K-fold cross-validation, only one data point from the dataset is left out for validation and the rest is used for training, there is lees bias as the whole dataset is used for training, so it requires more execution time.



3.4 Analytics

3.4.1 Types of Analytics

Analytics is one of the primary tasks when it comes to understanding the available dataset. Data analytics can be done at four levels and each subsequent level is a natural extension to the previous level.

Descriptive analytics is about deducing the statistical summary of data in terms of measures for central tendency (mean, median, mode) as well as measures for dispersion (variance, standard deviation). This helps getting insight about the past and answer: “What has happened?” [DA1]

Exploratory analytics is about visualizing the dataset at a high level to see its patterns and variations.

Predictive analytics is modeling the input variables and predicting the probability of outcomes.

Prescriptive analytics is comparing all viable predictions resulting from predictive analytics and choosing/prescribing the best among them.