**Cluster Analysis and Segmentation**

Cluster analysis is a popular unsupervised learning algorithm. Unsupervised means the data is unlabeled.

With labeled data, each data point receives a label that defines the group or category to which the data point belongs. For example:

* Tweets can be labeled in terms of their content category such as weather, sports, cooking, etc.
* Customers can be labeled as belonging to high-risk or low-risk groups.

In contrast, unlabeled data has no assigned labels so data analysis algorithms must rely on the identification of natural similarities, associations, and groupings (e.g. groups of customers). Additionally, the unlabeled nature of the data means that there is no target variable being predicted.

Cluster analysis is frequently used for various kinds of segmentation (identifying natural groupings of customers) including market segmentation, customer segmentation, etc. Most cluster analysis methods involve the use of a distance measure to calculate the closeness among the data items as illustrated in the diagram below:

Chart, bubble chart

Description automatically generated

Data points in one cluster tend to be closer and more similar to each other than to data points in a different cluster.

Cluster analysis can be used as a standalone tool for gaining insight into data distribution or as a preprocessing step for use by other algorithms.

Most cluster analysis methods involve the use of a some type of distance measure to calculate the closeness (similarity) among the data items so that the most similar items can be grouped (clustered) together. In the diagram below, note that:

* Data points within one cluster are more similar and closer to one another (intra-cluster distances are minimized)
* Data points in separate clusters are less similar and more distant from one another (inter-cluster distances are maximized)

Diagram

Description automatically generated

There are two major types of cluster analyses: *partitional clustering* and *hierarchical clustering*.

*Partitional clustering* divides data objects into non-overlapping subsets (clusters) such that each data object is included in exactly one subset (see below). The most common example of partitional clustering is *k-means clustering*.

Chart

Description automatically generated

*Hierarchical clustering* represents a set of nested clusters organized as a hierarchical tree (see below).

Diagram, venn diagram

Description automatically generatedChart, box and whisker chart

Description automatically generatedDiagram

Description automatically generatedChart, box and whisker chart

Description automatically generated

The most popular clustering algorithm is *k*-means clustering, a partitional clustering approach in which each cluster is associated with a cluster center (a centroid or center point) and each point in the data set is assigned to the cluster with the closest distance to its centroid. With *k*-means clustering, the number of clusters has to be specified a priori.

Here are the steps to a *k*-means clustering approach:

1. Specify the number of clusters *k* into which the data will be partitioned.
2. Randomly select initial cluster centers for the *k* clusters.
3. Assign each record to the closest cluster center. (Each cluster center “owns” a subset of records. This results in *k* clusters, C1, C2, ...., Ck, each owning a subset of the data points.)
4. For each of the *k* clusters, calculate the cluster centroid. Notice here that a centroid is an arbitrary central point of all items that belong to the same cluster.
5. Check to see if the conversion happened or termination point is reached.
6. Repeats Steps 3 – 5 until convergence or termination. (The *k*-means algorithm terminates when centroids no longer change. Often, the stopping condition is changed to a condition where *relatively few points change clusters*. The algorithm can also terminate when a specified maximum number of iterations is reached.)

The diagrams below shows the six iterations of *k-*means clustering:

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

The most common measure of evaluating *k*-means clusters is a sum of squares error (SSE), where for each point, the error is the distance to the nearest cluster. To get the SSE, we square these errors and then sum them:

Text

Description automatically generated with medium confidence

In this equation, x is a data point in cluster Ci and mi is the representative point for cluster Ci. We can show that mi corresponds to the center (mean) of the cluster. Given two clusters, we can choose the one with the smallest error. One easy way to reduce SSE is to increase the number of clusters, *k.*

Note: A good clustering with a smaller value of *k* can have a lower SSE than a poor clustering with higher value of *k.*

As with most algorithms, *k*-means clustering has strengths and weaknesses.

**Strengths:**

* It is intuitive and works quite well with numerically dominated variables and data sets.
* It is relatively efficient: O( n \* *k* \* I \* d ), where

n = number of points,

*k* = number of clusters,

I = number of iterations, and

d = number of attributes

**Weaknesses:**

* It is applicable only when the mean is defined and it has a poor fit for categorical data.
* It is necessary to specify *k*, the number of clusters, a priori.
* Selecting initial seed points may affect the clusters. Re-sorting the data can result in different cluster assignments.
* It is unable to handle noisy data and outliers. To overcome this, data scientists must normalize the data and suppress the outliers.
* It is not suitable for discovering clusters with non-globular (circular or elliptical) shapes.
* For some data sets, it might yield empty clusters.

The diagram below illustrates the sensitivity of *k*-means clustering to outliers. Note that suppressing the two outliers immediately results in a much tighter cluster.

Chart, pie chart

Description automatically generatedDiagram

Description automatically generated with medium confidence

Since k-means clustering only works with globular shapes, an attempt to cluster data shaped like two croissants will produce undesired results (see the diagram below).

Chart, scatter chart

Description automatically generated

To overcome this limitation, the data scientist needs to cluster the data into very small globular clusters and then treat them separately (see the diagram below):

Chart, scatter chart

Description automatically generated

There are several applications of cluster analysis:

* **Marketing:** Understanding your customers, discovering customer segments, and developing targeted marketing programs
* **Anomaly Detection:** Identifying unusual patterns (e.g. traffic flow variations) and outliers
* **City Planning:** Identifying groups of houses according to their house type, value, and geographical location
* **Earthquake Studies:** Identifying the locations of faults based on earthquake epicenter clustering
* **Intrusion Detection:** Identifying instances of anomalous (intrusive) user behavior in large system log files
* **Insurance:** Identifying groups of motor insurance policyholders having a high average claim cost

Collaborative Filtering, Association Rules Mining (Market Basket Analysis)

**Collaborative Filtering**

*Collaborative filtering* is a method for making automatic predictions (filtering) about user opinions, tastes, and preferences by collecting preferences from other users (collaborating). It is based on the assumption that personal tastes are correlated.

For example, if Nick and Mike both like pears and apples, and Nick likes peaches, then Mike is more likely to like peaches.

Diagram, schematic

Description automatically generated

Collaborative filtering is a technique used by recommender systems to recommend the “best” items based on a user’s previous purchases and on the opinions of like-minded users whose purchase patterns are similar (e.g. recommendations for movies or online purchases).

One of the most successful implementations of collaborative filtering is Amazon’s item-to-item collaborative filtering algorithm, which is used by millions of users on a daily basis (Hardesty, 2019).

**Association Rules Mining and Market Basket Analysis**

*Association rules mining*, frequently called *market basket analysis*, is an unsupervised learning algorithm that detects associations (affinities) between variables (items or events). Association rules mining helps users understand buying patterns, which can help to increase revenue and/or profit margins. If a retailer determines with 80 percent probability that customers purchase bread, milk, and bananas together during the same trip, they could:

* Place these items on opposite sides of the store thereby increasing the probability of customers traveling a longer distance, seeing more items, and buying items they didn’t plan to buy along the way.
* Avoid promoting (offering discounts on) more than one of these three items at a time. This effectively increases the profit margin since shoppers are already likely to buy the non-discounted items.
* Placing these items close to one another so customers will see them and be reminded that they need to buy all of them together, which increases both customer convenience and revenue.

There are several applications of market basket analysis:

* Cross-selling and up-selling
* Targeted promotions
* Product bundling
* Smart promotional pricing
* Store planograms
* Assortment optimization

The most frequently used algorithm for association rules mining is called *Apriori.* It is designed to operate on data sets containing transactions such as point-of-sale transactions representing items bought together in a single basket by a customer. Apriori uses a "bottom-up" approach, in which frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. Apriori terminates when no more candidate sets can be generated (as illustrated in the diagram below). Once the frequent itemsets are determined, the most prevailing association rules can be created and used to benefit the business.

Diagram

Description automatically generated

When assessing the power and usefulness of association rules, we use two key metrics: *support* (how often the items go together into the same basket) and *confidence* (how often one or more items go into a basket when other item(s) are already there). A powerful rule is expected to meet a minimum threshold of supper and have high enough confidence for practical purposes.

# Classification-Type Prediction Models

As we pointed out in M3, predictive models can be of two types: classification and regression. They mainly differ in the type of target variable they predict. Classification models predict a class label and regression models predict a numerical variable. The same set of input variables, which can be all numeric, categorical, or a combination of both, can be used for both regression and classification predictions. The following section explains the evaluation of performance metrics for classification-type prediction models.

## Evaluating Model Performance

At the model evaluation stage of the project, the data scientist must evaluate the model (or models) from both data analysis and business perspectives. Before proceeding to final deployment, the model must be evaluated to ensure it achieves the business objectives as outlined in CRISP-DM Phase 5  (see the figure below). At the end of this phase, the data scientist must decide whether or not to deploy the analytical results.

Diagram

Description automatically generated

In general, the data scientist will need to answer the following three questions:

1. What is desired from the data mining results?
2. Is the model demonstrating a strong enough goodness of fit?
3. Is the model providing business value in a meaningful way?

The simplest way to measure model accuracy is based on the classification error rate:

accuracy =number of correct decisions / total number of decisions

While classification accuracy is a popular measure, it is usually too simplistic for the application of data mining to real business problems. In order to achieve more meaningful results, data scientists need to decompose and count the different types of correct and incorrect decisions made by a classifier.

### Confusion Matrix

A confusion matrix is an n x n matrix that helps visualize the performance of a classification model having n classes. The rows correspond to predicted classes and the columns correspond to actual classes. (Although this is usually the arrangement of a confusion matrix, some software products have the labeling of the rows and columns switched.) The values contained in the confusion matrix can be used to measure classifier accuracy for each of n classes. Each example in a test set has an actual class label and a predicted class label (i.e., the class predicted by the classifier). A binary (2 x 2) confusion matrix is shown below.

|  |  | **Actual Class:** | **Actual Class:** |
| --- | --- | --- | --- |
|  |  | **P** | **N** |
| **Predicted Class:** | **P** | **TP** | FN |
| **Predicted Class:** | **N** | FP | **TN** |

In the matrix, P = positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative.

The main diagonal (italicized cells) contains the count of correct decisions:

|  |  | **Actual Class:** | **Actual Class:** |
| --- | --- | --- | --- |
|  |  | **P** | **N** |
| **Predicted Class:** | **P** | **TP** | FN |
| **Predicted Class:** | **N** | FP | **TN** |

### Unbalanced Classes and Rare Event Detection

In practice, many classification problems involve the detection of rare events. If a particular class appears very infrequently, classification can be used to identify those rare occurrences (e.g., defaulted loans, consumers responding to marketing campaigns, defective equipment, etc.).

For problems with unbalanced (“skewed”) class distributions, an evaluation based on accuracy does not provide an objective outcome.

For example, if only one out of 1000 transactions is fraudulent, a model that predicts that none of the transactions are fraudulent would be considered 99.9% accurate (a 999:1 ratio). In such cases, the model will always choose the most prevalent class. While this model does indeed have 99.9% accuracy, it doesn’t turn out to be particularly useful for credit card transactions since it would not be able to identify fraudulent transactions.

In order for the algorithm and corresponding measure of accuracy to work for unbalanced classes, data scientists have three major options for bringing the proportion between minority and majority classes closer to 50/50.

1. **Oversample the minority class** by taking multiple instances. In our 999 to one ratio example, we would need to select the same fraudulent case 999 times to bring the ratio to 999:999.
2. **Undersample the majority class** by taking a random sample. In our 999:1 ratio example, we would need to draw only one out of 999 non-fraudulent transactions to bring the ratio to 1:1.
3. **Generate synthetic samples** using techniques such as the synthetic minority oversampling technique (SMOTE), which uses a “nearest neighbors” algorithm to generate new synthetic data for the minority class to properly train and evaluate the model.
4. **Hybrid option** (performing any combination of the three solutions above)

The choice between the options listed above depends largely on the volume of available data. In case of big data (on the order of millions of transactions), it should be safe to just undersample. In situations where we have very little data, we might have to oversample. Otherwise, we can use a combination of these options to obtain the best results.

### Costs and Benefits

Classification algorithms do not make a distinction between false positive and false negative errors. However, the real-world implications of the differences between the two types of error are quite important.

Consider travel security. If a passenger is flagged for additional security screening but isn’t carrying any explosives, this would constitute a false positive error. It is inconvenient but not dangerous. However, if a passenger is carrying explosives but is not chosen for additional screening (and then goes on to board the airplane and detonates the explosives), this would constitute a false negative error with obvious serious consequences.

In this example, the costs and benefits of each decision should be weighted differently. Obviously, you will want to err on the side of over-screening rather than under-screening suspicious passengers.

In scenarios that are simpler to calculate—such as estimating the costs and benefits of retaining wireless communications customers—we can compute cost-benefit values for each decision pair.

Correct classifications (true positives and negatives) correspond to b\*TP and b\*TN, respectively (where b denotes ‘benefit’). The benefit is continued revenue from the retained customers over a period of time.

Incorrect classifications (false positives and negatives) correspond to c\*FN and c\*FP, respectively (cost, ‘c’, is usually a negative of benefits). The cost consists of discounts given to customers to retain them (e.g., a discounted smartphone).

|  |  | **Actual Class:** | **Actual Class:** |
| --- | --- | --- | --- |
|  |  | **P** | **N** |
| **Predicted Class:** | **P** | **b\*TP** | c\*FN |
| **Predicted Class:** | **N** | c\*FP | **b\*TN** |

### Model Performance Evaluation

Based on the entries in the confusion matrix, we can describe various evaluation metrics:

|  |  | **Actual Class:** | **Actual Class:** |
| --- | --- | --- | --- |
|  |  | **P** | **N** |
| **Predicted Class:** | **P** | **TP** | FN |
| **Predicted Class:** | **N** | FP | **TN** |

accuracy = (TP + TN) / (TP + FP + TN + FN)

precision = TP / (TP + FP)

recall = TP / (TP + FN)

specificity = TN / (TN + FP)

sensitivity = TP / (TP + FN)

True Positive Rate (TPR) = TP / (TP + FN)

False Positive Rate (FPR) = TP / (TP + FP)

F1 score= (precision \* recall) / (precision + recall)

## Visualizing Model Performance

The most frequently used methods for model performance visualization are receiver operating characteristic (ROC) curves and lift and gain charts.

### Receiver Operating Characteristics (ROC)

Receiver operating characteristic (ROC) curves are widely used along with an associated metric, the area under the curve (AUC). An ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

Chart, line chart

Description automatically generated

**AUC: Area Under the ROC Curve**

The area under the ROC curve (AUC) consists of the entire two-dimensional area underneath the ROC curve from (0,0) to (1,1). A near-perfect model—one that correctly identifies nearly all the “ones” correctly and then nearly all the” zeroes”—would have an ROC curve that hugs the upper left corner and an AUC of nearly 1.0 . An AUC value of 0.5 is deemed to be the baseline, which represents predictions no better than random chance.

Chart, pie chart

Description automatically generated

Often the business goal is not to classify every case, but rather to do a good job in identifying the “ones” without capturing too many “zeroes,” so we focus on the cases most likely to be “ones.” For example, a direct marketer wants to focus their effort on reaching only the most probable purchasers and an insurance investigator wants to spend time only on the likeliest frauds.

### Gain and Lift Charts

Gain and lift charts are used to visualize the performance of classification models. They are calculated as the ratio between the results obtained with and without the model.

**The cumulative gains** chart shows the percentage of the overall number of cases in a given category "gained" by targeting a percentage of the total number of cases.

The straight diagonal line is the "baseline" curve (lower line). If you select 10% of the cases from the scored dataset at random, you would expect to "gain" approximately 10% of all of the cases (upper line). The farther above the baseline a curve lies, the greater the gain.

Chart, line chart

Description automatically generated

**The lift chart** is derived from the cumulative gains chart and shows actual lift from the mode. The values on the y axis correspond to the ratio of the cumulative gain for each curve to the baseline. The greater the area between the lift curve and the baseline, the better the model

Chart, line chart

Description automatically generated

# Regression-Type Prediction Models

Regression-type prediction models differ from classification type prediction models in one key aspect. In a classification model, the predicted variable is a nominal value (i.e., a class label) while in a regression model, the predicted variable is a numeric value. In both kinds of models, the independent variables can consist of a combination of both numeric and nominal values. Regression-type models make use of machine learning techniques (e.g regression tree, neural network, support vector machine, and k-nearest neighbor techniques) as well as the most common statistical technique, namely regression.

Regression analyses are used for two primary purposes: hypothesis testing (investigating potential relationships between different variables) and prediction (estimating the values of response variables based on one or more explanatory variables). These two uses are not mutually exclusive. Regression’s explanatory power is also the foundation of its predictive ability. In hypothesis testing (theory building), a regression analysis can reveal the relationships between a number of explanatory variables (often represented mathematically as xi ) and a response variable (often represented as y). Predictions are expressed mathematically as a linear sum of explanatory variables, each multiplied by a coefficient. These equations can be used to predict or forecast the values of the response variables for a given set of explanatory variables.

## Correlation Versus Regression

Because regression analyses originated in correlation studies and because both methods attempt to describe the association between two (or more) variables, these two terms are often confused by professionals. A correlation between two (or more) variables makes no a priori assumption as to whether one variable is dependent on the other(s) and is not concerned with any causal relationship between the variables. Instead, it gives an estimate about the degree of association between the variables. On the other hand, a regression analysis attempts to describe the dependence of a response variable on one or more explanatory variables, where it implicitly assumes that there is a one-way causal effect from the explanatory variables to the response variable, regardless of whether the effect is direct or indirect. Also, while a correlation analysis is interested in low-level relationships between two variables, regression is concerned with the relationships between all explanatory variables and the response variable.

## Simple Versus Multiple Regression

If a regression equation is built between one response variable and one explanatory variable, it is called simple regression. For instance, a regression equation built to predict or explain the relationship between the height of a person (explanatory variable) and the weight of a person (response variable) is a good example of simple regression. Multiple regression is an extension of simple regression and features more than one explanatory variable. For instance, if we build upon the previous example and include not only the height of a person but also other personal characteristics (e.g., BMI, gender, and ethnicity) to predict the weight of the person, we would be performing a multiple regression analysis. In both cases, the relationships between the response variables and the explanatory variables are linear and additive in nature. If the relationships are not linear, then we might want to use one of many other nonlinear regression methods to better capture the relationships between the input and output variables.

## Visualization

To understand the relationship between two variables, the simplest thing to do is to create a scatter plot, where the y-axis represents the values of the response variable and the x-axis represents the values of the explanatory variable. Such a scatter plot shows the changes in the response variable as a function of the changes in the explanatory variable. In the illustration shown below, there is a positive relationship between the two—as the values of the explanatory variables increase, so do the values of the response variables.

Chart, scatter chart

Description automatically generated

Each point represents a particular pairing of an explanatory variable (xi) and a response variable (yi). A simple regression analysis aims to find a mathematical representation of this relationship. It tries to find the equation of a straight line passing through the plotted dots such that the distance between each data point (xi,yi) and this line is minimized. Thus, the y values on this line are the predicted values for corresponding values of x.

Several methods/algorithms have been proposed to identify this regression line, of which the most commonly used approach is the ordinary least squares (OLS) method. The OLS method aims to minimize the sum of squared residuals (i.e., squared vertical distances between the observed points and the corresponding point on the regression line), which leads to a mathematical expression for the estimated slope of the regression line (known as the β parameter). For simple linear regression, this relationship between the response variable (y) and the explanatory variable(s) (x) can be shown as a simple equation, as follows:

y = β0 + β1x

In this equation, β 0 is called the intercept, and β 1 is called the slope. Once OLS determines the values of these two coefficients, the simple equation can be used to forecast values of y for given values of x. The sign and the value of β1 also reveal the direction and the strength of the relationship between the two variables.

If the model involves more than one explanatory variable, a multiple linear regression analysis is necessary, which results in additional β coefficients—one for each additional explanatory variable. As the following formula shows, the additional explanatory variables would be multiplied with new βi coefficients and summed together to establish a linear additive representation of the response variable.

y = β0 + β1x1 + β2x2 + ... + βnxn