# **MODELLING**

# **Supervised Learning methods:**

**Naïve Bayes**

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The reason that Naive Bayes algorithm is called Naive is not because it is simple. It is because the algorithm makes a very strong assumption about the data having features independent of each other while in reality, they may be dependent in some way.

If this assumption of independence holds, Naive Bayes performs extremely well and often better than other models. Naive Bayes can also be used with continuous features but is more suited to categorical variables. If all the input features are categorical, Naive Bayes is recommended. However, in case of numeric features, it makes another strong assumption which is that the numerical variable is normally distributed.

Here are some tehniques for improving power of Naive Bayes Model:

* If continuous features do not have normal distribution, we should use transformation or different methods to convert it in normal distribution.
* If test data set has zero frequency issue, apply smoothing techniques “Laplace Correction” to predict the class of test data set.
* Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance.

**Pros:**

* It is easy and fast to predict class of test data set. It also performs well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**Cons**:

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

**Decision Trees**

The major advantage of using decision trees is that they are intuitively very easy to explain. They closely mirror human decision-making compared to other regression and classification approaches. They can be displayed graphically, and they can easily handle qualitative predictors without the need to create dummy variables.

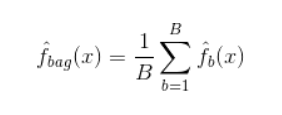
However, decision trees generally do not have the same level of predictive accuracy as other approaches, since they aren't quite robust. A small change in the data can cause a large change in the final estimated tree.

By aggregating many decision trees, using methods like bagging, random forests, and boosting, the predictive performance of decision trees can be substantially improved.

**Bagging**

The decision trees suffer from high variance, meaning if you split the training data into 2 parts at random, and fit a decision tree to both halves, the results that you get could be quite different. In contrast, a procedure with low variance will yield similar results if applied repeatedly to distinct dataset.

Bagging, or bootstrap aggregation, is a technique used to reduce the variance of your predictions by combining the result of multiple classifiers modeled on different sub-samples of the same dataset.



in which you generate B different bootstrapped training datasets. You then train your method on the bth bootstrapped training set in order to get f^b(x), and finally average the predictions.

**Pros**

* Reduces variance in comparison to regular decision trees
* Can provide variable importance measures
* Can easily handle qualitative (categorical) features
* Out of bag (OOB) estimates can be used for model validation

**Cons**

* Not as easy to visually interpret
* Does not reduce variance if the features are correlated

**Boosting**

Boosting is another approach to improve the predictions resulting from a decision tree. Like bagging and random forests, it is a general approach that can be applied to many statistical learning methods for regression or classification

In Boosting, trees are grown sequentially: each tree is grown using information from previously grown trees. Boosting does not involve bootstrap sampling; instead, each tree is fitted on a modified version of the original dataset.

Boosting is very useful when you have a lot of data and you expect the decision trees to be very complex. Boosting has been used to solve many challenging classification and regression problems, including risk analysis, sentiment analysis, predictive advertising, price modeling, sales estimation and patient diagnosis, among others.

**Pros**

* Somewhat more interpretable than bagged trees/random forest as the user can define the size of each tree resulting in a collection of stumps (1 level) which can be viewed as an additive model
* Can easily handle qualitative (categorical) features

**Cons**

* Unlike bagging and random forests, it can overfit if number of trees is too large

**Random Forests**

Random Forests is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job.

Random Forests provides an improvement over bagged trees by a small tweak that decorrelates the trees. As in bagging, you build a number of decision trees on bootstrapped training samples. But when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors. This is the main difference between random forests and bagging; because as in bagging, the choice of predictor m=p.

Random Forests is very effective at estimating missing data and maintaining accuracy when a large proportion of the data is missing. It can also balance errors in datasets where the classes are imbalanced. Most importantly, it can handle massive datasets with large dimensionality. However, one disadvantage of using Random Forests is that you might easily overfit noisy datasets, especially in the case of doing regression

**SVM:**

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well

**Advantages**

* High Dimensionality: SVM is an effective tool in high-dimensional spaces, which is particularly applicable to document classification and sentiment analysis where the dimensionality can be extremely large.
* Memory Efficiency: Since only a subset of the training points are used in the actual decision process of assigning new members, just these points need to be stored in memory (and calculated upon) when making decisions.
* Versatility: Class separation is often highly non-linear. The ability to apply new kernels allows substantial flexibility for the decision boundaries, leading to greater classification performance.

**Disadvantages**

* Kernel Parameters Selection: SVMs are very sensitive to the choice of the kernel parameters. In situations where the number of features for each object exceeds the number of training data samples, SVMs can perform poorly. This can be seen intuitively as if the high-dimensional feature space is much larger than the samples. Then there are less effective support vectors on which to support the optimal linear hyperplanes, leading to poorer classification performance as new unseen samples are added.
* Non-Probabilistic: Since the classifier works by placing objects above and below a classifying hyperplane, there is no direct probabilistic interpretation for group membership. However, one potential metric to determine the "effectiveness" of the classification is how far from the decision boundary the new point is.
* Time Consumed: When compared with other machine learning algorithms for the same problem, SVMs consume a lot of time.

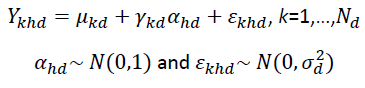
For hospital rating problem, we use Random Forests to classify the hospitals into 5 categories each representing one of the ratings (1-5).

# **Unsupervised Learning Methods:**

We employ latent variable modeling (LVM) to estimate a group score for the dimension of quality represented by the measures in each group. CMS constructed a separate LVM for each group so that a total of seven latent variable models are used.

LVM is a statistical modeling approach which assumes each measure reflects information about an underlying, unobserved dimension of quality. LVM accounts for the relationship, or correlation, between measures for a single hospital. Measures that are more consistent with each other, as well as measures with larger denominators, have a greater influence on the derived latent variable. The model estimates for each hospital the value of a single latent variable representing an underlying dimension of quality; this estimate is the hospital’s group score.

Equation for Latent Variable Model within each group d is as follows:



Let 𝑌𝑘ℎ𝑑 denote the standardized score for hospital h and measure k in group d. 𝛼ℎ𝑑 is the hospital-specific group-level latent trait (random effect) for hospital h and group d. 𝛾𝑘𝑑 is the loading (regression coefficient of the latent variable) for measure k, which shows the relationship with the group score of group d. 𝑁𝑑 is the total number of measures in group d. 𝛼ℎ𝑑 follows a Normal distribution with mean 0 and variance 1. The assumption of unit variance here is an innocuous choice of units required to identify the parameter 𝜇𝑘𝑑 and 𝛾𝑘𝑑.

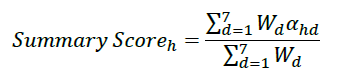
The influence of an individual measure on the group score is represented by the measure’s “loading.” A loading is produced for each measure in a group when applying the LVM; these statistically estimated measure loadings are regression coefficients based on maximum likelihood methods using observed data and are not subjectively assigned. A loading reflects the degree of the measure’s influence on the group score relative to the other measures included in the same group.

We give more weight to measure scores that are more precise by using a weighted likelihood method as follows:



L is the likelihood function. 𝑁𝑘𝑑 is the total number of hospitals for measure k in group d and 𝑛𝑘ℎ𝑑 is the denominator for hospital h and measure k in group d. A hospital with larger denominator will be weighted more in the LVM.

After estimating the group score for each hospital and each group, we calculate a weighted average to combine the seven group scores into a single hospital summary score.



To get the final rating, we use a k-Means clustering technique. The k-means clustering analysis is a standard method for creating categories (or clusters) so that the observations (or scores) in each category are closer to their category mean than to any other category mean. The number of categories in this case is 5. Hospitals were organized into one of five categories such that a hospital’s summary score is “more like” that of the other hospitals in the same category and “less like” the summary scores of hospitals in the other categories. The final Star Rating categories were structured such that the lowest group is one star and the highest group is five stars