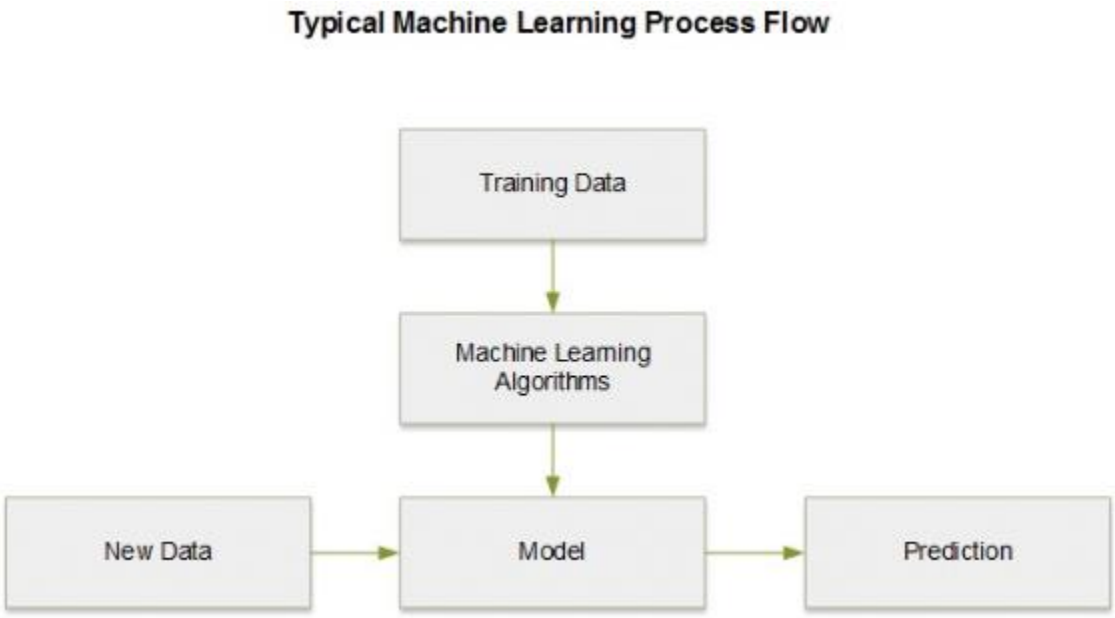
SUPERVISED LEARNING AND COMPARISON OF SL TECHNIQUES

Supervised learning is a type of machine learning algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values. From it, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. A test dataset is often used to validate the model. Using larger training datasets often yield models with higher predictive power that can generalize well for new datasets.

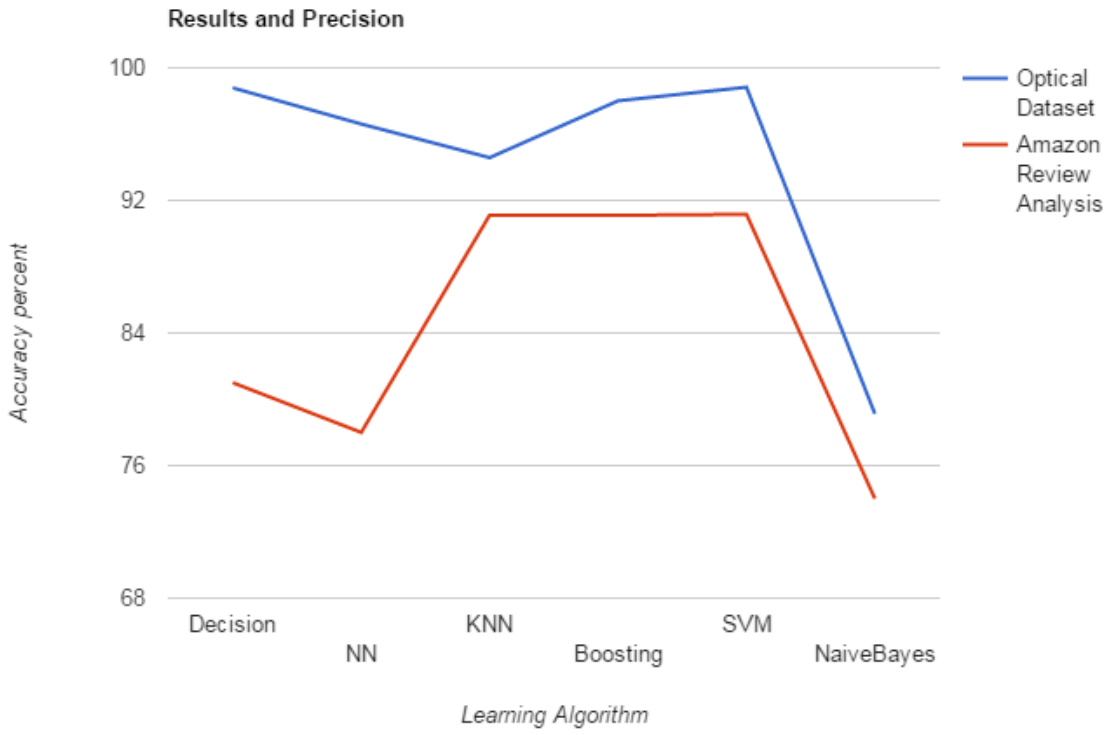


Machine learning can be generalized in steps of gathering data, extracting features and then selecting an appropriate model that represents the given model and then fitting the model to the test data set. While this method captures all major stages of a learning process in general, some new stages may be added and some may be circumvented. For example, in case of Amazon data review, once data was gathered, further analysis required a data pre-processing step, which quantified textual data. Different data set pose different challenges. This stage gets so momentous that the accuracy and the process of modeling and fitting the problem space is drastically influenced. Feature extraction includes identifying degrees of freedom in a data set that contribute to the classification of a given test data. Interesting and efficient ideas like dimensionality reduction tremendously supplement the robustness of the model. This stage may implement ideas like principal component analysis for dimensionality reduction and compression.

RESULTS AND PRECISION

The following are the results for a small space of 1000 samples in each category.

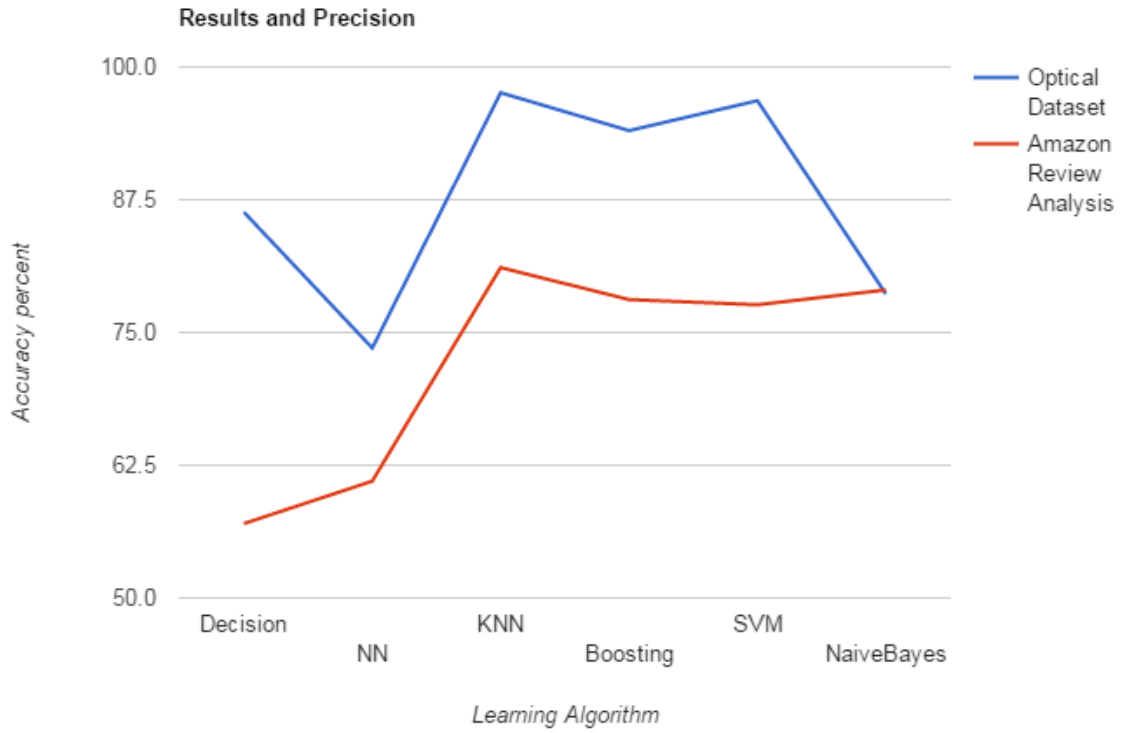
METHOD OF LEARNING	OPTICAL CHARACTER RECOGNITION(%)	AMAZON REVIEW ANALYSIS(%)
DECISION TREE	98.79	81
NEURAL NETWORK	96.61	78
K NEAREST NEIGHBOUR	94.58	91.11
BOOSTING	98.01	91.1
SUPPORT VECTOR MACHINE	98.83	91.16
NAÏVE BAYES	79.13	74.01



Every method applied on the given data set does less accurately on textual data of Amazon product reviews and better on the quantized set of optical character recognition. As remarked earlier, the efficiency and accuracy of learning methods depend on the pre-processing done on data.

The following results are observed for the entire data set.

METHOD OF LEARNING	OPTICAL CHARACTER RECOGNITION(%)	AMAZON REVIEW ANALYSIS(%)
DECISION TREE	86.34	57
NEURAL NETWORK	73.54	61
K NEAREST NEIGHBOUR	97.58	81.11
BOOSTING	94.01	78.1
SUPPORT VECTOR MACHINE	96.83	77.61
NAÏVE BAYES	78.63	79.01



An acute observation is that textual data, regardless of technique, does rather poorly, especially because of a large sparse matrix and partly because of the absence of a robust technique for corpus textual pre-processing. In particular, nearest neighbor has proved to be more appropriate for the given data sets weighed on Euclidian distance.

Behavior of learning techniques

Decision Tree*:*

Information Gain

Information gain is usually a good measure for deciding the relevance of an attribute. It is often used to decide which of the attributes are the most relevant, so they can be tested near the root of the tree. The more a training sample is to contain information, the closer it is to the

root. Information gain measures how well a given attribute separates the training example according to target classification. This is so because an early branching of the tree based on the most information filters out the irrelevant paths.

Handling Over fitting:

To handle over fitting, we use reduced error pruning in which we consider each decision node in the tree to be considered for pruning and nodes are pruned iteratively, always choosing the node whose removal mostly increased the decision tree accuracy over the validation set.

Missing values and continuous value attribute

To handle the missing value attribute, we assign it the most common value among training examples at node n. For continuous value attribute, we define a new discrete valued attribute that partition the continuous attribute values into discrete set of intervals.

Complexity of decision trees.

The complexity of decision trees is upper-bounded by the training sample space. Every offshoot of the tree corresponds to a training datum. If there are n samples for training, then the tree can have, at most, n levels of node. In other words, the depth of a decision tree can never be more than the test space. When the decision tree is fully modeled, it is more certain to fit noise as well as actual data

NEURAL NETWORK

Backpropagation:

The algorithm repeats a two phase cycle, propagation and weight update. When an input vector is presented to the network, it is propagated forward through the network, layer by layer, until it reaches the output layer. The output of the network is then compared to the desired output, using a loss function, and an error value is calculated for each of the neurons in the output layer. The error values are then propagated backwards, starting from the output, until each neuron has an associated error value which roughly represents its contribution to the original output. Back propagation uses these error values to calculate the gradient of the loss function with respect to the weights in the network. In the second phase, this gradient is fed to the optimization method, which in turn uses it to update the weights, in an attempt to minimize the loss function.

Nature of model.

Artificial neural networks form a network of hypotheses to arrive at a conclusion. In course of doing so, it entertains back propagation and feedback among hypotheses. Arrival at a conclusion is a result of multiple mixes and matches of hypotheses. Therefore, artificial neural networks can describe the result in addition to just predicting the result.

Error metric.

The error metric of neural networks is defined by an activation function instead of the output Itself. This avoids the problem of having to confront the inability to differentiate output, aiding in achieving back propagation

K-Nearest Neighbor:

Weight and vicinity

It is an instance-based learning technique which assumes that all instances correspond to n points in an n-dimensional space. The vicinity of points is determined by the Euclidian distance between them. Instances with the least distance from the given point—or those with the largest inverse of distance—are said to be closest to a given point. Therefore, given an instance, its class is determined by the class of the closest instance/s. The choice of neighbors is highly data-dependent. The effect of noise is suppressed by large training data set. Another possible value for this parameter is distance, which weighs points based on in the inverse of their distance from the point in question. While uniform weight considers every instance within a radius alike, weighing by distance places more weight near the centre or the query point. Preference of one two another depends on a sound knowledge of the nature of input data

Influence of K

The influence of density of clustering influences the prediction. However, the way in which it does solely relies upon the sample space size. The value of k is directly constrained by the sample size: k should be lesser than or equal to the size of sample space. For a small data set, deciding how the number of neighbors affects the precision in prediction gets difficult. In case of optical character recognition, a hundred samples were considered to observe how k affects accuracy.

Instance-based approach.

Each target function can be associated with a different approximation for each unique query. The drawback of this is that computation takes place at the time of classification instead of during training. Hence, the cost of classification is high

Influence of irrelevant neighbors/Proper Dataset

Nearest neighbor algorithm is accurate to core if the data set has two features with numerous Training observations evenly scattered throughout the input space. The framework of nearest neighbors is reliant on closest values. For a given sample, it need necessarily not be true that all its nearest neighbors are of the same class. Hence, irrelevant features can affect subsequent construal. Therefore, each neighbor is weighed differently to overcome the negative impact of irrelevant attributes.

BOOSTING

Number of Estimators:

Boosting is an ensemble technique, which coalesces several base hypotheses (or base estimators) in order to improve generalisability over one strong hypothesis. Boosting formulates base hypotheses sequentially and attempts to reduce the bias of their combined strong hypothesis. One factor that is blatant to affect error rate the is number of base estimators. As we consider more base estimators, we are specializing each degree of freedom

to some extent, the combined estimator of which, therefore, reaches more accuracy with less error rate.

Support Vector Machines:

C and accuracy:

The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyper plane if that hyper plane does a better job of getting all the training points classified correctly C, in fact, has no direct effect on the accuracy of the model in linearly-separable data sets. Nevertheless, it can influence the accuracy of linearly inseparable models: for small values of C, we can expect high rates of error. As C increases, error rate gradually decreases and the change in error rate becomes almost insignificant after a particular value of C. The graph plots a monotonically and asymptotically decreasing curve. Although it is a general tendency to evaluate a model based on the accuracy of prediction, it is not an appropriate method to assess the performance of a model. Other parameters like RMS, ROC Curves, PR Curves, confusion matrices, etc. help us to determine the performance of a model

Kernel points and hyperplane:

Kernel points act as reference to classifying the training data on a hyper plane. This classification is coupled with the type of kernel chosen. For linearly separable data set, the linear model/kernel is most apt. In fact, the optical Character recognition assumes this kernel for modeling and does poorly with any other, particularly with rbf. For Amazon data set, the kernel does not seem to have much of an influence on the accuracy. Any kernel like rbf or Laplacian is sufficiently efficient in modeling the given data set.

Degree:

The degree of the polynomial kernel function('poly'). This parameter is ignored by all other kernels. The default value is 3. The degree parameter is only used if kernel is set to poly..

ROC Curves:

ROC curves are graphs of false positive rates plotted against true positive rates. The curve defines the behaviour of the modeled system. A curve above the ideal uphill diagonal (left-to-right) denotes an efficient model, whereas a curve drawn below the ideal ROC curve represents a poor model.

Naïve Bayes:

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular

feature is independent of the value of any other feature, given the class variable. The observation with the highest likelihood directly leads us to the class to which the sample belongs.

Parameter Evaluation and Event Models:

The assumptions on distributions of features are called the event model of the Naive Bayes classifier. For discrete features multinomial and Bernoulli distributions are used and these led to two distinct models – Gaussian Naïve Bayes and Multinomial naïve Bayes. When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial where pi is the probability that event i occurs. In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks where binary term occurrence features are used rather than term frequencies. Here is the major difference between multinomial and Bernoulli model

	multinomial model	Bernoulli model
event model	generation of token	generation of document
random variable(s)	$X = t$ iff t occurs at given pos	$U_t = 1$ iff t occurs in doc
document representation	$d = \langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle, t_k \in V$	$d = \langle e_1, \ldots, e_i, \ldots, e_M \rangle,$ $e_i \in \{0, 1\}$
parameter estimation	$\hat{P}(X = t\|c)$	$\hat{P}(U_i = e\|c)$
decision rule: maximize	$\hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(X = t_k\|c)$	$\hat{P}(c) \prod_{t_i \in V} \hat{P}(U_i = e_i\|c)$
multiple occurrences	taken into account	ignored
length of docs	can handle longer docs	works best for short docs
# features	can handle more	works best with fewer
estimate for term the	$\hat{P}(X = \text{the}\|c) \approx 0.05$	$\hat{P}(U_{\text{the}} = 1\|c) \approx 1.0$