ASSIGNMENT-1

Ankur Garg
01013202717

Q1. What are the differences between Linear discriminant analysis and nonlinear discriminant analysis? (5)
A1.
LDA: Assumes: data is Normally distributed. All groups are identically distributed, in case the groups have different covariance matrices, LDA becomes Quadratic Discriminant Analysis. LDA is the best discriminator available in case all assumptions are actually met. QDA, by the way, is a non-linear classifier.

SVM: Generalizes the Optimally Separating Hyperplane(OSH). OSH assumes that all groups are totally separable, SVM makes use of a 'slack variable' that allows a certain amount of overlap between the groups. SVM makes no assumptions about the data at all, meaning it is a very flexible method. The flexibility on the other hand often makes it more difficult to interpret the results from a SVM classifier, compared to LDA.

SVM classification is an optimization problem, LDA has an analytical solution. The optimization problem for the SVM has a dual and a primal formulation that allows the user to optimize over either the number of data points or the number of variables, depending on which method is the most computationally feasible.
SVM can also make use of kernels to transform the SVM classifier from a linear classifier into a non-linear classifier. Use your favorite search engine to search for 'SVM kernel trick' to see how SVM makes use of kernels to transform the parameter space.

LDA makes use of the *entire* data set to estimate covariance matrices and thus is somewhat prone to outliers. SVM is optimized over a subset of the data, which is those data points that lie on the separating margin. The data points used for optimization are called support vectors, because they determine how the SVM discriminate between groups, and thus support the classification.

SVM doesn't really discriminate well between more than two classes. An outlier robust alternative is to use logistic classification. LDA handles several classes well, as long as the assumptions are met. I believe, though (warning: terribly unsubstantiated claim) that several old benchmarks found that LDA usually perform quite well under a lot of circumstances and LDA/QDA are often go to methods in the initial analysis.

Q2. What is bagging and boosting? (5)

A2. Bagging is used when the goal is to reduce the variance of a decision tree classifier. **Here the objective is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees.** As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

Bagging Steps:

- Suppose there are N observations and M features in training data set. A sample from training data set is taken randomly with replacement.
- A subset of M features are selected randomly and whichever feature gives the best split is used to split the node iteratively.
- The tree is grown to the largest.
- Above steps are repeated n times and prediction is given based on the aggregation of predictions from n number of trees.

Advantages:

- Reduces over-fitting of the model.
- Handles higher dimensionality data very well.
- Maintains accuracy for missing data.

Disadvantages:

- Since final prediction is based on the mean predictions from subset trees, it won't give precise values for the classification and regression model.

Python Syntax:

- rfm = RandomForestClassifier(n_estimators=80, oob_score=True, n_jobs=-1, random_state=101, max_features = 0.50, min_samples_leaf = 5)
- fit(x_train, y_train)
- predicted = rfm.predict_proba(x_test)

Boosting

Boosting is used to create a collection of predictors. **In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree.** When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. This process converts weak learners into better performing model.

Boosting Steps:

- Draw a random subset of training samples d1 without replacement from the training set D to train a weak learner C1
- Draw second random training subset d2 without replacement from the training set and add 50 percent of the samples that were previously falsely classified/misclassified to train a weak learner C2
- Find the training samples d3 in the training set D on which C1 and C2 disagree to train a third weak learner C3
- Combine all the weak learners via majority voting.

Advantages:

- Supports different loss function (we have used 'binary:logistic' for this example).
- Works well with interactions.

Disadvantages:

- Prone to over-fitting.
- Requires careful tuning of different hyper-parameters.

Python Syntax:

- from xgboost import XGBClassifier
- xgb = XGBClassifier(objective='binary:logistic', n_estimators=70, seed=101)
- fit(x_train, y_train)
- predicted = xgb.predict_proba(x_test)

Q3. Use KNN to solve this problem below (10)

A3. Here is for training sample.

x1=Acidddurability	X2=Strength	Y=classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Let this be the test sample

x1=Acidddurability	X2=Strength	Y=classification
3	7	?

1. Determine the parameter k=the no.of nearest neighbours.
   Say k=3
2. Calculate the distance between queryinstance and all the training samples.

Coordinate of query instance is (3,7) ,instead of calculating the distance we compute square distance which is faster to calculate(without squareroot)

x1=Acidddurability	X2=Strength	Distance b/w queryinstance and all the training samples
7	7	(7-3)^2+(7-7)^2=16
7	4	(7-3)^2+(4-7)^2=25
3	4	(3-3)^2+(4-7)^2=9
1	4	(1-3)^2+(4-7)^2=13

3. Sort the distance and determine Nearest neighbors based on the kth minimum distance.

x1=Aciddurability	X2=Strength	Distance b/w queryinstance and all the training samples	Rank minimum distance	Is it include 3-nearestneighbour
7	7	(7-3)^2+(7-7)^2=16	3	Yes
7	4	(7-3)^2+(4-7)^2=25	4	No
3	4	(3-3)^2+(4-7)^2=9	1	Yes
1	4	(1-3)^2+(4-7)^2=13	2	Yes

4. Gather the category Y of the nearest neighbours .

x1=Aciddurability	X2=Strength	Distance b/w queryinstance and all the training samples	Rank minimum distance	Is it include 3-nearestneighbour	Category of nearestneighbor
7	7	(7-3)^2+(7-7)^2=16	3	Yes	Bad
7	4	(7-3)^2+(4-7)^2=25	4	No	-
3	4	(3-3)^2+(4-7)^2=9	1	Yes	Good
1	4	(1-3)^2+(4-7)^2=13	2	Yes	Good

-> the second row inthe last column that the category of nearest neighbours (Y) is not included becoz the rank of this data is more than 3(=k).

5. Use simple majority of the category of nearest neighbors as the prediction value of query instance.

We have 2 good and 1 bad ,since,2>1 So we conclude that a new paper tissue that pass laboratory test with x1=3 and x2=7 is included in Good category.

x1=Aciddurability	X2=Strength	Y=classification
3	7	Good

Q4. Explain the generative probabilistic classification? (5)

A4. A generative model could generate new photos of animals that look like real animals, while a discriminative model could tell a dog from a cat. GANs are just one kind of generative model.

More formally, given a set of data instances X and a set of labels Y:

- **Generative** models capture the joint probability p(X, Y), or just p(X) if there are no labels.
- **Discriminative** models capture the conditional probability p(Y | X).

A generative model includes the distribution of the data itself, and tells you how likely a given example is. For example, models that predict the next word in a sequence are typically generative models (usually much simpler than GANs) because they can assign a probability to a sequence of words.

A Generative Model learns the joint probability distribution p(x,y). It predicts the conditional probability with the help of Bayes Theorem. A Discriminative model learns the conditional probability distribution p(y|x). Both of these models were generally used in supervised learning problems.

Generative classifiers

- Assume some functional form for **P(Y), P(X|Y)**
- Estimate parameters of **P(X|Y), P(Y)** directly from training data
- Use Bayes rule to calculate **P(Y |X)**

Q5. What is machine learning? Discuss the issues in machine learning and the steps required for selecting right machine learning algorithm. (5)

A5. Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions.

the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data.

Here are some important considerations while choosing an algorithm.

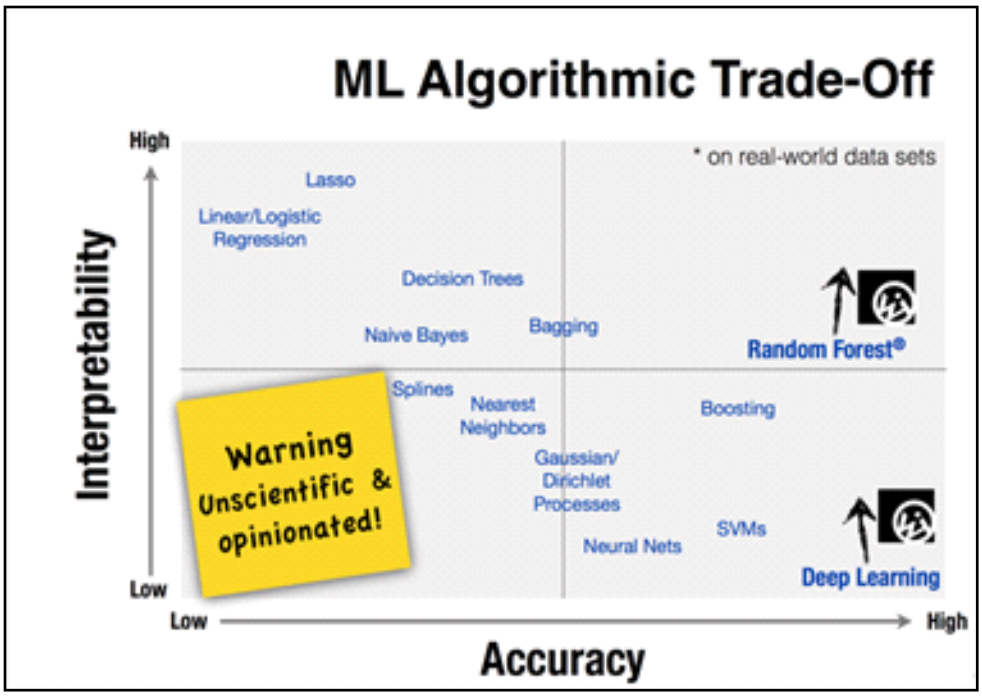
1. Size of the training data

It is usually recommended to gather a good amount of data to get reliable predictions. However, many a time, the availability of data is a constraint. So, if the training data is smaller or if the dataset has a fewer number of observations and a higher number of features like genetics or textual data, choose algorithms with high bias/low variance like Linear regression, Naïve Bayes, or Linear SVM.

If the training data is sufficiently large and the number of observations is higher as compared to the number of features, one can go for low bias/high variance algorithms like KNN, Decision trees, or kernel SVM.

2. Accuracy and/or Interpretability of the output

Accuracy of a model means that the function predicts a response value for a given observation, which is close to the true response value for that observation. A highly interpretable algorithm (restrictive models like Linear Regression) means that one can easily understand how any individual predictor is associated with the response while the flexible models give higher accuracy at the cost of low interpretability.



Some algorithms are called Restrictive because they produce a small range of shapes of the mapping function. For example, linear regression is a restrictive approach because it can only generate linear functions such as the lines.

Some algorithms are called flexible because they can generate a wider range of possible shapes of the mapping function. For example, KNN with k=1 is highly flexible as it will consider every input data point to generate the mapping output function. The below picture displays the trade-off between flexible and restrictive algorithms.

Now, to use which algorithm depends on the objective of the business problem. If inference is the goal, then restrictive models are better as they are much more interpretable. Flexible models are better if higher accuracy is the goal. In general, as the flexibility of a method increases, its interpretability decreases.

3. Speed or Training time

Higher accuracy typically means higher training time. Also, algorithms require more time to train on large training data. In real-world applications, the choice of algorithm is driven by these two factors predominantly.

Algorithms like Naïve Bayes and Linear and Logistic regression are easy to implement and quick to run. Algorithms like SVM, which involve tuning of parameters, Neural networks with high convergence time, and random forests, need a lot of time to train the data.

4. Linearity

Many algorithms work on the assumption that classes can be separated by a straight line (or its higher-dimensional analog). Examples include logistic regression and support vector machines. Linear regression algorithms assume that data trends follow a straight line. If the data is linear, then these algorithms perform quite good.

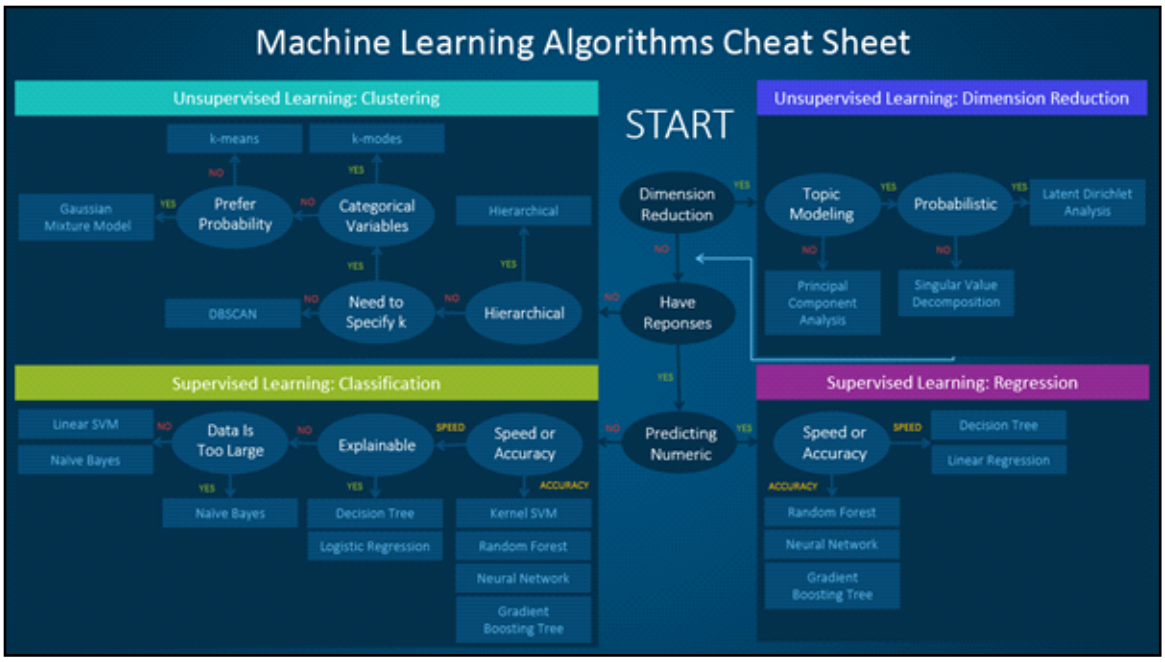
However, not always is the data is linear, so we require other algorithms which can handle high dimensional and complex data structures. Examples include kernel SVM, random forest, neural nets.

The best way to find out the linearity is to either fit a linear line or run a logistic regression or SVM and check for residual errors. A higher error means the data is not linear and would need complex algorithms to fit.

5. Number of features

The dataset may have a large number of features that may not all be relevant and significant. For a certain type of data, such as genetics or textual, the number of features can be very large compared to the number of data points.

A large number of features can bog down some learning algorithms, making training time unfeasibly long. SVM is better suited in case of data with large feature space and lesser observations. PCA and feature selection techniques should be used to reduce dimensionality and select important features.

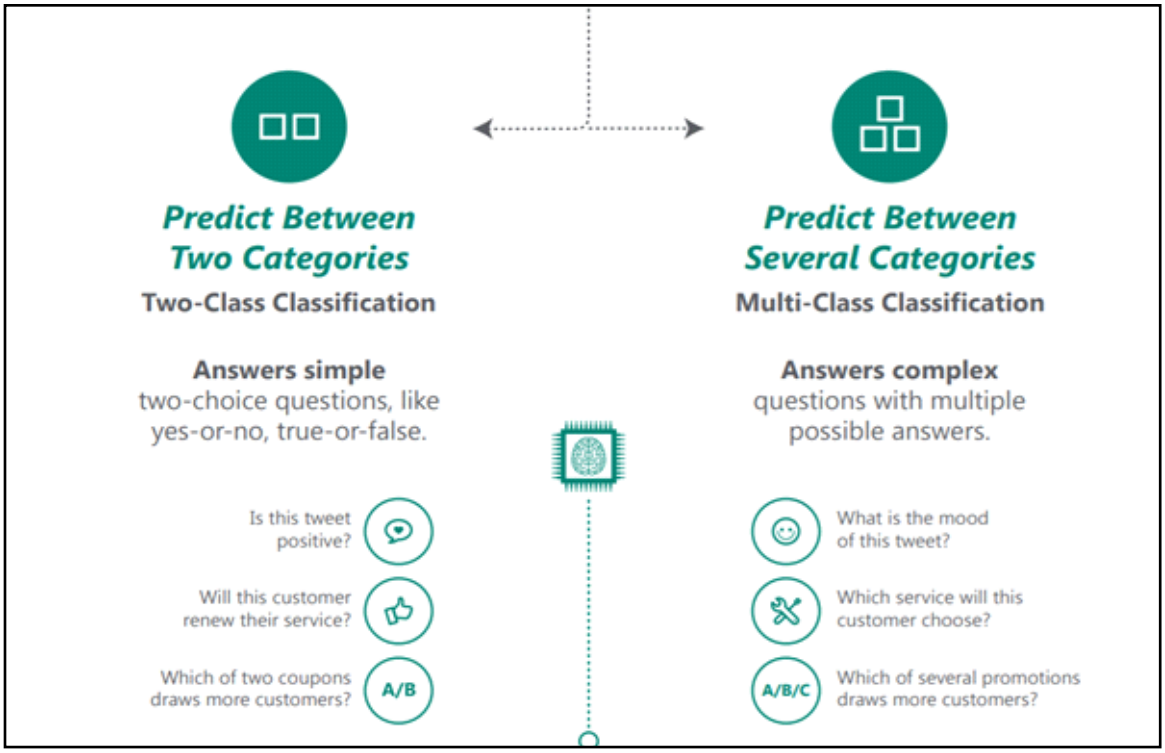


Machine learning algorithms can be divided into supervised, unsupervised, and reinforcement learning.

Supervised learning algorithms are employed where the training data has output variables corresponding to the input variables. The algorithm analyses the input data and learns a function to map the relationship between the input and output variables. Supervised learning can further be classified into Regression, Classification, Forecasting, and Anomaly Detection.

Unsupervised Learning algorithms are used when the training data does not have a response variable. Such algorithms try to find the intrinsic pattern and hidden structures in the data. Clustering and Dimension Reduction algorithms are types of unsupervised learning algorithms.
The below infographic simply explains Regression, classification, anomaly detection, and clustering along with examples where each of these could be applied.


Regression
Forecast the future by estimating the relationship between variables.
Predict Values
Estimate product demand
Predict sales figures
Analyze marketing returns
Predict credit risk
Detect fraud
Catch abnormal equipment readings
Find Unusual Occurrences
Anomaly Detection
Identify and predict rare or unusual data points.
Clustering
Separate similar data points into intuitive groups.
Discover Structure
Perform customer segmentation
Predict customer tastes
Determine market price
Predict Categories
Classification
Identify what category new information belongs in.


Predict Between Two Categories
Two-Class Classification
Answers simple two-choice questions, like yes-or-no, true-or-false.
Is this tweet positive?
Will this customer renew their service?
Which of two coupons draws more customers?
A/B
Predict Between Several Categories
Multi-Class Classification
Answers complex questions with multiple possible answers.
What is the mood of this tweet?
Which service will this customer choose?
A/B/C
Which of several promotions draws more customers?

The main points to consider when trying to solve a new problem are:

- Define the problem. What is the objective of the problem?
- Explore the data and familiarise yourself with the data.
- Start with basic models to build a baseline model and then try more complicated methods.