# Lecture\_2c

Hello and welcome to the third lecture during this week. We will be discussing classification algorithms in this lecture. We'll start by having a recap of classification and we are going to look at common classification algorithms.

Classification problems in machine learning could take on several forms. One of the most popular forms are unary classification, binary classification, multiclass classification, multilabel classification and imbalanced classification. There are many classification algorithms which can be used or tuned to address classification problems, be it unary classification, binary classification, multiclass, classification, multilabel classification and imbalanced classification. And the common ones include: logistic regression, support vector machine, discriminant analysis, naïve bayes and k-nearest neighbour. And these are the algorithms that we'll be discussing in this lecture. Now, it's very important to note that logistic regression is not regression. Instead, it's a classification algorithm that shares a similar mathematical formulation to date of linear regression. But it's not the same. Logistic regression, as the name suggests, uses a logistic function to model a binary dependent variable.

In other words, a binary classification.

We have more complex extensions that can be used to achieve multiclass classification. And we have an illustration here that depicts the logistic function that has modelled a binary dependent variable, which has been used for binary classification between false samples and true samples Logistic regression is often employed as the starting point for binary classification. That is when we have a data centre can be separated by a single linear boundary. It can also be used as a baseline for the evolution of more complex classification methods. Support Vector Machine, or SVM. Typically, support vector machine classifiers carry out classification by finding a linear decision boundary, otherwise known as a hyperplane that demarcates all data points of one class from those of the other class. Take for instance, if we have a 10,000 dimensional dataset, support vector machine is going to work by putting all on feature vectors on an imaginary 10,000 dimensional plot, then it draws an imaginary 10,000 dimension line which is the hyperplane to separate the two classes into 10,000 dimensional dataset. The optimum hyperplane or linear decision boundary for an SVM classification model is the one having the largest margin between the two classes of the dataset, when the data is linearly separable, and you have an illustration there of the margin that has been referred to. For nonlinearly separable data, that is when the hyper plane cannot assume the form of a straight line, a hinge loss function has to be introduced to penalise feature vectors on the wrong side of the plane to make support vector machine applicable for nonlinearly separable data.

Generally, SVM classifiers work well for high dimensional data and for nonlinearly separable data.

It's good to note that even though SVM classifiers are typically for binary classification, they can also be used for multiclass classification with a technique called error-correcting output codes (ECOC). ECOC mainly walks by allowing a multiclass classification problem to be reframed or remodelled as multiple binary classification problems that binary classification models such as SVM models can address. There are two ways to go about this. The first approach is one versus the rest where we split a multiclass problem into one binary problem per class. And the second approach is one versus one where a multiclass problem is split into one binary problem per each pair of classes. A discriminant analysis classifier classifies data by finding the combinations of features in the dataset. And to do this, it assumes that the data in different classes are generated based on Gaussian distributions, in other words, normal distributions.

To train a discriminant analysis model or classifier, the parameters for Gaussian distribution for each class in the data set must be found. And these parameters are used to determine the class of new data. Typically, the parameters we are interested in are the standard deviation and the mean.

And once we have the Gaussian and distribution parameters, we can calculate the discriminant boundaries for the classes. The boundaries could be linear or quadratic functions. If we have linear boundaries, this is referred to as linear discriminant analysis and if we have a quadratic function, this is referred to as quadratic discriminant analysis as illustrated in the diagram. Discriminant analysis classifiers are often preferred to be used under the following conditions: when simple and interpretable classification models are required. When memory usage during training is a major concern and when fast prediction is required. Also, take note – when a mixture of Gaussian models are used per class, it is called Mixture Discriminant Analysis, and you have the illustration there showing the decision boundaries for linear discriminant analysis, quadratic discriminant analysis and mixture discriminant analysis.

Naïve bayes. This is a classifier that is based on Bayes' theorem, probability theorem, and it classifies data by assuming that the presence of a particular feature in a class is unrelated to the presence of any other feature in the dataset. In other words, each feature in the dataset makes an independent equal contribution to the outcome. Intuitively, we can tell that the above assumption is very naive because in practise we expect to have some level of interdependence or relation between the features, hence the name Naïve bayes. Take for instance, if we had to build a classifier for assessing the risk in loan applications, It seems counterintuitive to ignore the correlations between features such as age, education level, an income of the applicants. The naïve bayes classifiers are best used for: small datasets contending many parameters, classifiers requiring ease of interpretation and classifiers that are suitable for scenarios that are not necessarily captured in the training data. Vivid examples of such classification problems can be found in financial and medical applications. Very similar to the discriminant analysis classifier, a typical Naïve bayes classifier applies probability density information to the given data and classifies new data based on the highest probability of the new data belonging to a particular class. If a Gaussian distribution is assumed, as it is in most cases, then the naïve bayes classifier can be extended to what we call the Gaussian naïve bayes classifier And you have the Gaussian distribution there, the bell shipped curve. Typically, for Gaussian naïve bayes classifiers, the main estimates required from the training data, as before, are the mean and the standard deviation. k-nearest neighbour, also called kNN. This is a non parametric classification algorithm that works by categorising objects based on the classes of their nearest neighbours and the given dataset.

Given a new observation, or a new example, kNN works by finding the k nearest neighbours to the new observation in the dataset and it returns a label or class A category that has the largest posterior probability amongst the response values for the k points to establish the class of the new point, or to predict the class of the point. The predictions from kNN assume that he objects in proximity or near each other are similar. And several distance metrics could be used to find the nearest neighbour. The common ones include the Euclidean distance, cosine similarity, city block and Chebyshev. The kNN nearest neighbour algorithm is best used when memory usage and the prediction speed of the trained model are a lesser concern. In this lecture, we've had a recap of classification and we've also looked at common classification algorithms.