# Lecture\_7a

Hello and welcome to the first lecture in this week. We are going to be discussing the machine learning implementation. In this lecture video, a step-by-step guide for machine learning implementation is going to be presented and model performance evaluation is also going to be discussed.

The majority of machine learning problems tend to relate to handling data and building or finding them the most appropriate model, typically classification or regression model. Real-world data can assume a plethora of forms: it could be messy and noisy. It could be incomplete. It could be numeric.

It could be images. It could be signals and several other forms. Prior to adopting data for machine learning, it's very important for the data to be pre-processed by the data analyst and the process of pre-processing the data may involve the use of specialised knowledge or even specific tools.

Generally, signal processing techniques and descriptive statistics may be employed for feature extraction or feature engineering as part of the data pre-processing. Feature engineering will be discussed later. It's also very important to select the most suitable model in machine learning to avoid overfitting and underfitting, and the process of selecting the most suitable model is always a balancing act. This is because there's always going to be trade-offs between model speed, accuracy and complexity. Overfitting occurs when the model models the training data too well. In other words, when the model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on the new data made available to the model. Underfitting on the other hand occurs when a model can neither model the training data nor generalise to new data.

Some initial queries for machine learning implementation. What type or kind of data is to be used? Is it static data such as continuous data? Is a static data such as discrete or categorical data? Is it

dynamic data coming from a dynamic environment? And what inferences are to be drawn, or what insights are to be gained from the data? How and where will the inferences and insights be applied?

The answers to these queries will help to decide whether supervised learning or unsupervised learning or even reinforcement learning is to be employed. So, it's very important that data analysts answer these queries prior to machine learning implementation. The core stages required for implementation of machine learning are shown at a glance in the workflow below and remember, the ultimate goal is to keep trying until a near-optimum model which is suitable for the intended application is found. At the first stage, the data is accessed and loaded. At the second stage the data is pre-processed. At the third stage, features are extracted from the pre-processed data. At the fourth stage, models are trained using the extracted features in the second stage. And at the fifth stage, an iteration is carried out continuously until an optimum or near-optimum model is found.

Once a near-optimum model is found, this model is integrated into the desired system or application as the best trained model. Stage one – the data to be used can be from a number of sources. They could be real world experimental data performed by the data analyst or another party. They could also come from databases such as the Google dataset search and it's very important to always acknowledge or cite the data source when the dataset is being used for machine learning implementation are from databases. Typically, the raw data used for machine learning is often available as text or CSV or any other suitable format. And before using the dataset for machine learning, it's also important to ensure that the dataset is complete, that is there's no missing data and it's also free of noise. This is because machine learning algorithms cannot tell the difference between noise and useful information in datasets. a priori. Second stage is pre-process the data. It's always a good practise to inspect the imported or loaded data to check for outliers, that is data points that may lie outside the rest of the data points in the dataset. To check for missing values: blanks are nan numbers in the dataset. If there are outliers, the data analyst must decide whether the outliers can be excluded or whether they need to be included because they indicate certain attributes or characteristics the model, ought to account for. Typically, for very large dataset

with very few outliers. The outliers can be ignored. On the other hand, for many applications such as fraud or anomaly detection, outliers are very important because they provide crucial information

and as a result, it must be included. For machine learning implementation, missing values can be handled by either data interpolation or data imputation techniques. Data interpolations simply means replacing missing data with estimated values and data imputation simply means replacing missing data with substituted values.

Once the dataset has been established and ascertained to be clean and complete, then the dataset can be partitioned by the data analyst into the test set, which is to be used for testing the model and the training set, which is to be used for building and training the model. By testing the model against data that wasn't used in the modelling process, that is the test set, the performance of the model with unknown data becomes very visible. This approach is called hold-out. In machine learning implementation, hold-out is when the dataset is split into a training set and a test set. The training set is what the model is trained on and the test set is used to see how well the model performs on unseen data. In machine learning implementation, hold-out is mainly used for larger datasets, typically datasets have been over 10,000 observations. A very common split when using the hold-out method is using 80% of the data for training and the remaining 20% of the data for testing. Cross-validation. In machine learning implementation, cross-validation could also be used to see how the well the model performs on unseen data, and cross-validation is also called k-Fold cross-validation. To implement cross-validation, the dataset is randomly split up into k groups.

One of the groups is used as the test set and the rest are used as the training set. The model is then trained on the training set and scored on test set. Then the process is repeated until each unique group has been used. as the test set. 5-fold cross-validation. In machine learning, 5-fold cross validation is commonly used when the dataset is not large, typically, from a few hundreds up to 10,000 observations. For the popular 5-fold cross-validation, the dataset will be split into 5 groups and the model would be trained and tested 5 separate times so each group would get a chance to be the test set. Feature extraction. Feature extraction is also called feature engineering, and it's a vital part of machine learning, primarily because it turns raw data into information that can be used by machine learning algorithms. It's good to note that there is no limit to the number of features that can be derived from raw data. In machine learning, depending on the source and type of data,

descriptive statistical techniques and signal processing techniques are often used for feature engineering or feature extraction.

As illustrated in this table, we can see that for sensor data where we are primarily interested in extracting signal properties from raw sensor data to have a higher-level information.

We can use things like peak analysis, pulse and transition metrics and spectrum measurements. For image and video data where we're very much interested in extracting features such as edge locations, resolution and colour. We can use techniques which will involve bag of visual words, histograms, minimum eigenvalue algorithms to detect corner locations in image, and we could also use edge detection techniques. For transactional data where we're very much interested in calculating derived values that enhanced the information in the data. We could use techniques such as timestamp decomposition and aggregate value calculation. Stage 4 – building and training the model. For practical machine learning implementation, it is always best to start by building and training simple models before gradually progressing into building and training more complex models for the same machine learning problem. Take for instance, to address a classification and/or regression problem, a basic decision tree can be built and trained as an initial step. The performance of the initial model or the initial decision tree can then be evaluated before building more efficient and robust models. In machine learning, there are a number of ways to evaluate the model's performance. And some of the popular approaches include, but are not limited to the goodness of fit, mean squared error, prediction or validation accuracy, prediction speed, training time and several others. Training time. This is the time taken by machine learning algorithm to construct and train a model.

Prediction speed. This is the total number of predictions made by a model divided by the time taking to make the predictions. And the predictions speed is usually measured in observations per second. Goodness of fit. For regression models, a statistical measure of the closeness of data to the fitted regression line always helps to deduce the goodness of fit of the regression model. A very popular statistical measure used for this deduction is the R-squared statistic. R-squared statistic is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The R-squared statistic is conventionally the percentage of the response variable variation that is explained by the regression model. Mathematically, it can be estimated as follows. The sum of squares regression expressed as a ratio of the sum of squares total, which is also equal to one take away the ratio of the sum of squares received error to the sum squares total.

The R-square statistic is always the value between 0 and 1, or 0% and for the R-square statistic indicates that the regression model explains none of the variability of the response data around its mean. And a value of 1 for the R-square statistic indicates that the regression model explains all the variability of the response data around its mean. In other words, the higher the R-square value, the better the regression model fits the data. Another statistical measure which helps in evaluating a regression model is this means squared error, popularly known as MSE. The MSE is the average squared difference between the estimated values and the actual values. Mathematically, it can be estimated by squaring the sum of the residual errors and dividing this sum by the total number of observations. The square root of the MSE is the root mean squared error. The value of MSE is always positive or greater than zero. A value close to zero will represent better quality of the regression model. In other words, an MSE of zero represents the fact that the regression model is a perfect predictor. Prediction or validation accuracy. Conventionality, the prediction accuracy of a classification model is a ratio of the correct predictions made by the classification model to the total number of predictions made by the classification model. Mathematically, it's the sum of the true positives and true negatives expressed as a ratio of the sum of the true positives, true negatives, false positives and false negatives. Assuming a dummy class x in a given classification problem, a true positive is an observation predicted to be in class x and was actually in class X.

A true negative is an observation predicted not to be in class x and was not in class X. A false positive is an observation predicted to be in class x, but was not in class x. And the false negative is an observation predicted not to be in class x, but was actually in class x. False positive and false negative. For many practical classification problems, false positives are less tolerated compared to false negatives. Given this spam detection classification problem, a false positive in spam detection is a situation in which a contact sends you an email, but the classification model labels it as spam

and as a result, the email is not seen. On the other hand, the false negative is less of a problem because if the classification model doesn't detect a small percentage of spam messages, it's typically not a big issue. Confusion matrix. A confusion matrix is also called an error matrix and it's a specific table layout that allows visualisation of the performance of the classification model as illustrated here. It's primarily has two dimensions (the true and predicted), and identical sets of classes can be found in both dimensions. So as you can see here, for the true class, we have setosa, versicolour and virginica, and the same for the predicted. We have setosa, versicolour and virginica. The confusion matrix can also appear as a table with two rows and two columns showing the number of false positives, false negatives, true positives and true negatives, which are FP, FN,TP and TN as described before. The confusion matrix again, for the high risk flower classification problem, we can tell by inspection that all 50 observations for Iris Setosa were correctly predicted, that is all flowers that were Iris Setosa were predicted correctly to be Iris Setosa. However, for Iris Versicolour 47 were correctly predicted and 3 were predicted to be Iris Virginica.

For Iris Virginica, 46 observations which were Iris Virginica where predicted to be Iris Virginica. However, for observations which were Iris Virginica were predicted to be Iris Versicolour. ROC and and AUC. Many classification models return confidence scores of predictions, that is posterior probabilities. As a result, their prediction accuracies' summaries can also be viewed using their receiver operating characteristic curves, that is ROC curves and the areas on the ROC curves, that is AUCs. ROC is a probability curve and AUC represents the degree or measure of separability. Both ROC and AUC tell how much the model is capable of distinguishing between classes. For a given ROC curve, they prediction accuracy summary is provided using a combination of the true positive rate and the false positive rate, and by discretising the confidence scores returned by the classification model.

The true positive rate is also called recall or sensitivity and it is found by the ratio of the true positives to the sum of the true positives and the false negatives. Specificity on the other hand is found by the ratio of the true negatives to the sum of the true negatives and false positives. And the false positive rate is 1 minus specificity, which can also be found by the ratio of false positives to the sum of false positives and true negatives. Note that the discrete scores are the prediction thresholds used to predict the classes of the observations in the dataset. The higher the area under the ROC curve, that is AUC now, the better the classifier. A classifier with a AUC higher than 0.5 is better than a random classifier. If AUC is lower than 0.5, then something must be wrong with the model. A perfect classifier would have an AUC of 1, and you can tell intuitively that we want the AUC for the classifier to be as close to 1 as possible.

In this video, we've discussed a step-by-step guide for machine implementation and we've also looked at model performance evaluation to focus on regression models and classification models.