**DSA1101**

**STATISTICS**

By observation, data shows a tendency to agglomerate around a central value. This is called **central** **tendency**. This central value can be used to summarize data.

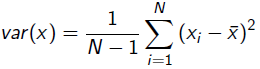
The mean, The median

mode is the most frequent value in a data set

The tendency for data points to spread around a central value

is called **dispersion**.

The sample variance



The sample standard deviation is the square root of the variance

The range is the difference between the largest and smallest values.

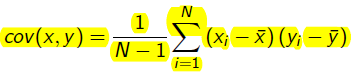
x=c(1 ,3 ,5 ,7 ,10 ,10 ,11)

> max (x)-min (x)

10

**Measures of association**

For two data vectors, their sample covariance is given by



The sample correlation coefficient  -1 ≤ rxy≤ 1, to compare the degree of association

Effects of scale and location changes

**Location** changes: for two data vectors x and y, suppose we create two new data vectors u = x + a and v = y + b, where a and b are constants

* Then the mean, median and mode for u will be different from those for x (similarly for v and y)
* **Sample variance, standard deviation, covariance, and rxy are invariant to changes in location**

Scale changes: for two data vectors x and y, suppose we create two new data vectors u = a - x and v = b - y, where a and b are constants.

* Then the mean, median and mode for u will be different from those for x (similarly for v and y)
* Sample variance, standard deviation and covariance between the two data vectors will also be different
* **Sample variance** changes **multiplicatively by the square of the scale**, var(u) = (-5)^2 \*var(x)
* **Sample standard deviation** changes according to the **absolute value of the scale**, sd(u) = |-5|sd(x)
* rxy is invariant to scale changes (provided the constants a and b are of the same sign)

**LINEAR REGRESSION**

In simple linear regression we are given a response or a dependent variable y and a single explanatory variable x, and we fit the model, **y = β0 + β1x + ε** where β0 (the intercept) and β1 (the slope) are two unknown parameters and ε is an "error term". We will ignore ε for now,

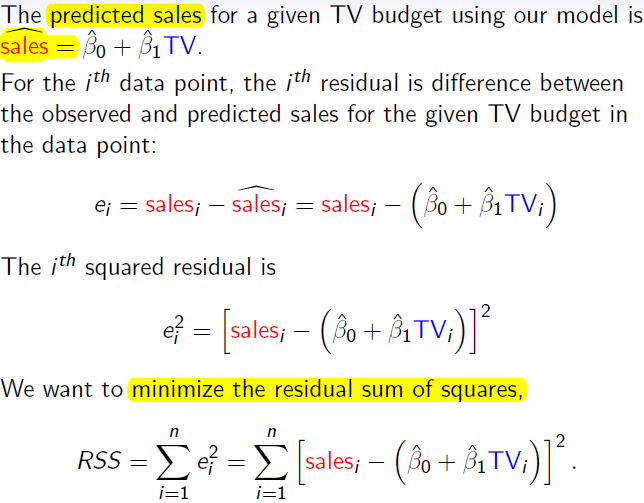
until later when we perform inference.

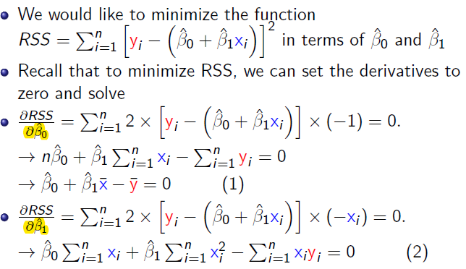
We want the estimated values, **β0** and **β1 (có dấu mũ)**.

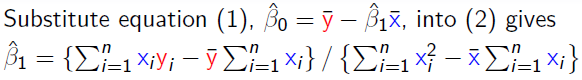
Minimise the function <-> Least sum of squares of distances

*We can’t get the distances because positive and negative distances will cancel out.* Intuitively, we want the line to be as close to the data points as possible. This "closeness" can be measured in terms of the vertical distance between each point to the line(represented by the length of the purple lines).

To express the total magnitude of the deviations, we sum up the squared residuals for all the data points. The resulting sum is the Residual Sum of Squares, abbreviated as RSS.

**





If we think of **ε** as a random variable, then there is inherent

uncertainty in our least squares estimates **β** 0 and **β** 1.

One way to quantify this uncertainty is by computing the

standard errors SE(**β**0) and SE(**β**1).

**Confidence intervals:**

The standard errors SE(**β**0) and SE(**β** 1) can be used to

compute confidence intervals.

A 95% confidence interval is defined as a range of values such

that, were this procedure to compute the range be repeated

on numerous samples, the fraction of calculated confidence

intervals (which would differ for each sample) that encompass

the true parameter value would tend toward 95%.

For linear regression, the 95% confidence interval for **β**0 is

Approximately [β0 – 2\*SE(β0), β0 + 2\*SE(β0)], *có mũ,* similar for **β**1

**Hypothesis testing:** testing the null hypothesis of H0 : There is no relationship between x and y

If p-value< α, then we reject H0 and conclude that there is a linear relationship between x and y at the α significance level. Typically α is 0.05 or 0.1

**Model diagnostics**

If the linear relationship does not seem to apply, it is often

useful to do any of the following:

(a) Transform the outcome variable, e.g. by taking the logarithm

(b) Transform the input variable(s)

(c) Add extra input variables or terms to the regression model

sales = **β** 0 + **β** 1TV + **β** 2TV2 + e, multiple linear regression

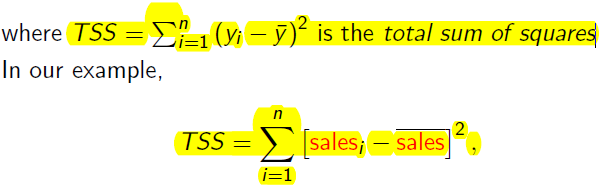
**Assessing the accuracy of the model**

The quality of a linear regression fit can be assessed using the residual standard error (RSE).

Larger RSE indicates poorer model fit. RSE = sqrt[RSS/(n-2)]

The R2 statistic is another measure of the fit of the model. Since R2 takes on a value between 0 and 1, it is independent of the scale of the outcome, unlike RSS. To calculate R2, we use the formula

R2 = (TSS-RSS)/TSS

\*wrt. the average value of y

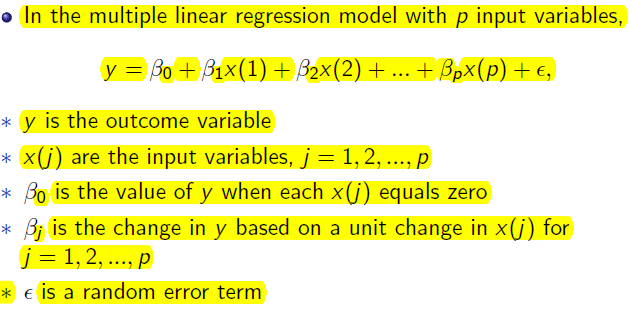
TSS measures the total variance in the response Y, and can be thought of as the amount of variability inherent in the response before the regression is performed. In contrast, RSS measures the amount of variability that is left unexplained after performing the regression. TSS - RSS measures the amount of variability in the response that is explained (or removed) by performing the regression. Hence, R2 measures the proportion of variability in outcome Y that can be explained using the predictor variable X with the linear model. Larger R2 indicates better model fit.

**Checking the normality assumption**

We have assumed that the error terms in the simple linear model sales = **β** 0 + **β** 1TV + + e

are normally distributed with a mean of zero and constant variance. To check for constant variance across all outcome (sales in our example) values along the regression line, use a simple plot of the residuals against the fitted outcome values.

**Multiple Linear Regression**



**K-MEANS CLUSTERING**

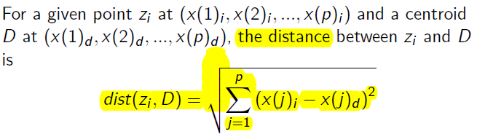
We seek to partition them into distinct groups so that the observations within each group are quite

similar to each other, while observations in different groups are quite different from each other.

The **center** is determined as the arithmetic average (mean) of each cluster's n-dimensional vector of attributes.

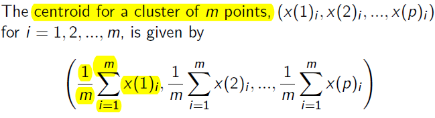
**1st step:** Choose the value of k and the k initial guesses/ initial starting points for the centroids.

**2nd step:** Compute the distance from each data point (xi ; yi ) to each centroid. For (x1, x2) and (y1, y2), …



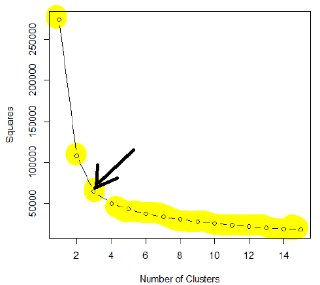
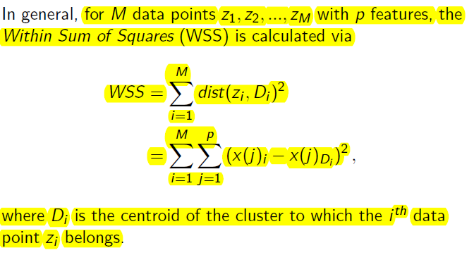
Since the **distance** from the point to the … centroid is the **shortest,** we classify this point as belonging to the … centroid.

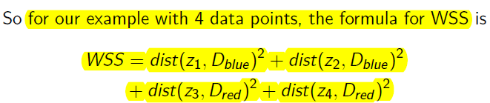
**3rd step:** compute the new centroids in each cluster (the mean of each cluster)



Repeat steps one and two until the algorithm converges to an answer (when the centroids are more or less stable). *May use data points as centroid*

**Within Sum of Squares, WSS**

****



*Lower WSS -> better clustering*

The process of identifying the appropriate value of k is referred to as finding the “elbow" of the WSS curve.

Although k-means is considered an unsupervised method, there are still several decisions that the practitioner must make:

(i) Which features should be included in the analysis?

(ii) What unit of measure (for example, miles or kilometers)

should be used for each feature?

(iii) Do the features need to be rescaled so that one feature does

not have a disproportionate effect on the results?

(iv) What other considerations might apply?

Whenever possible and based on the data, it is best to reduce the number of attributes to the extent possible. Too many attributes can minimize the impact of the most

important variables. The use of several similar attributes can place too much

importance on one type of attribute. For example, if five attributes related to personal wealth are

included in a clustering analysis, the wealth attributes dominate the analysis and possibly mask the importance of other attributes, such as age.

One useful approach is to identify any highly correlated attributes and use only one or two of

the correlated attributes in the clustering analysis.

**Unit of measure**

The algorithm will identify different clusters depending on the choice of the units of measure. For example, age in years and height in meters. *Height max ~2, whether age difference contributes more to the distance formula.*

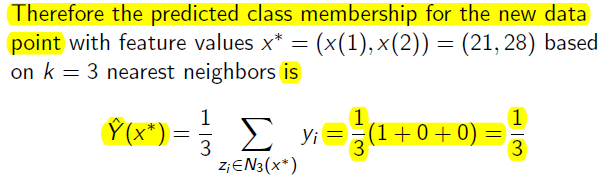
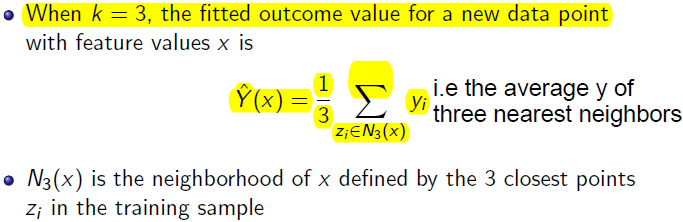
*Choosing the starting cluster also affects the result.*

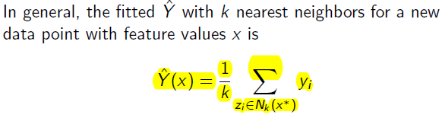
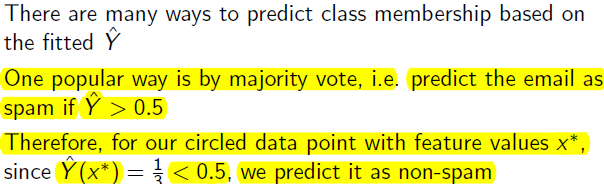
**K-NEAREST NEIGHBOR CLASSIFICATION**

In many cases, the outcome y is a categorical variable or class membership, for example S/U, 1/0).

An object is classified by **a majority vote** of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). To determine the k nearest neighbors, we will use the Euclidean distance in the feature space (x).

For example, we **set k = 3**, so we need to find **the three nearest data points** in the feature space. […] These three data points are the closest to our predicting point.





**Larger k**, smoother boundary between labels -> Model Complexity reduces

In general, the prediction error for a model can be decomposed into

error = bias^2+ variance + irreducible error

For larger k, we are also taking data points further away from the circle to compute our fitted value. This may lead to greater bias in our fitted value Y^.

Thus **when k increases** (i.e. more data points), the **variance decreases**, but **bias increases**

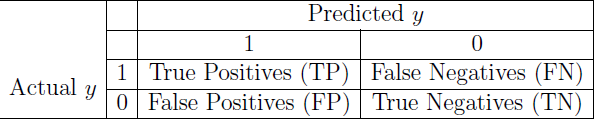
This is known as the bias-variance tradeoff.

**EVALUATE PERFORMANCE OF CLASSIFIERS**

not C

In a two-class classification, a pre-set threshold may be used to separate positives from negatives (e.g. we used the majority rule, Y^ < 0.5, in the k-nearest neighbour example).

**The confusion matrix**

****

TP and TN are the correct guesses.

A good classifier should have large TP and TN and small (ideally zero) FP and FN.

*Formula: Result + Predicted Answer*

**Accuracy =**

The accuracy (or the overall success rate). A good model should have a high accuracy score, but having a high accuracy score alone does not guarantee the model is well established.

**Error rate== 1- accuracy**

**True Positive Rate (TPR)=** shows the proportion of positive instances the classifier correctly identified:

**False Positive Rate (FPR)** shows what percent of negatives the classifier marked as positive; also called the false alarm rate or the type I error rate. [dư thừa]

**False Negative Rate (FNR)** shows what percent of positives the classifer marked as negatives, also called miss rate, type II error rate. [thiếu sót]

**Precision=** the percentage of instances marked positive that really are positive:

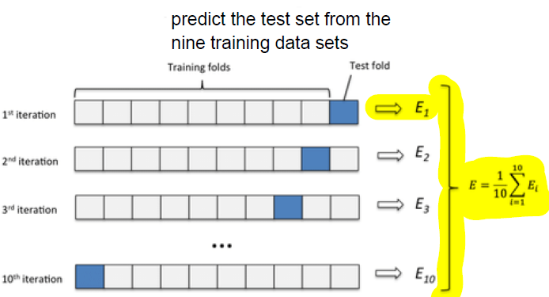
A well-performed model should have a high TPR that is ideally 1 and a low FPR and FNR that are ideally 0. Note that in general, the model that is more preferable may depend on the business situation.

* [small plz] Consider the example of e-mail spam littering. Some people (such as busy executives) only want important e-mail in their inbox and are tolerant of having some less important e-mail end up in their spam folder as long as no spam is in their inbox. In this case, a higher false positive rate (FPR) or type I errorcan be tolerated.
* Other people may not want any important or less important e-mail to be speci\_ed as spam and are willing to have some spam in their inboxes as long as no important e-mail makes it into the spam folder. In this case, a higher false negative rate (FNR) or type II error can be tolerated.
* The cost of having a person, who has the disease, to be instead diagnosed as disease-free is extremely high, since the disease may be highly contagious. Therefore, the false negative rate (FNR) or type II error needs to be low. A higher false positive rate (FPR) or type I error can be tolerated.
* Third example involves security screening at the airport. The cost of a false negative in this scenario is extremely high (not detecting a bomb being brought onto a plane couldresult in hundreds of deaths) whilst the cost of a false positive is relatively low (a reasonably simple further inspection) Therefore, a higher false positive rate (FPR) or type I error can be tolerated, in order to keep the false negative rate (FNR) or type II error low.

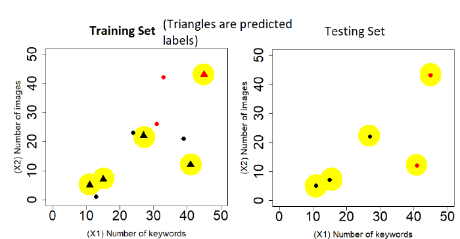
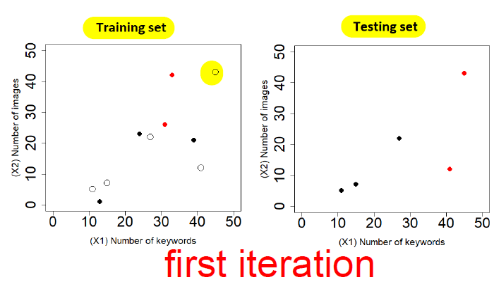
**N-Fold Cross-Validation**

* The entire dataset is randomly split into N datasets of approximately equal size.
* N-1 of these datasets are treated as the training dataset, while the remaining one is the test dataset. A measure of the model error is obtained.
* This process is repeated across the various combinations of N datasets taken N - 1 at a time.
* The observed N model errors are averaged across the N folds.

predict the test set from the nine training data sets

****

Example: Suppose our dataset consists of 10 data points. For 2-fold cross validation, we randomly split the whole dataset of 10 points into two datasets of 5.



* Accuracy= 4/5

2nd iteration: we use the second dataset as the training set and the first dataset as the testing set.

Accuracy=3/5

Thus the accuracy of the algorithm is (4/5 + 3/5)/2 = 0.7