WEEK -04: PARAMETRIC METHODS

Machine Learning Model evaluation metrics

The various ways to check the performance of our machine learning or deep learning model and why to use one in place of the other. We will discuss terms like:

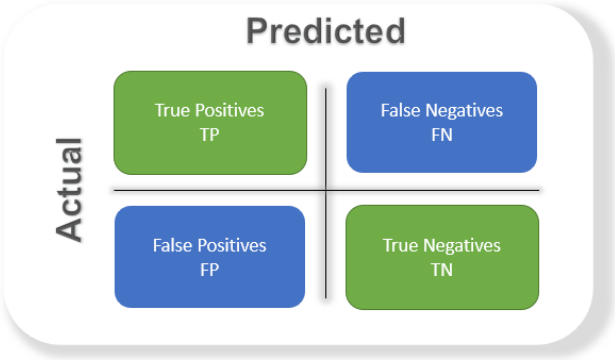
- Confusion matrix
- Accuracy
- Precision
- Recall
- Specificity
- F1 score
- Precision-Recall or PR curve
- **ROC** (**R**eceiver **O**perating **C**haracteristics) curve
- PR vs ROC curve.

For simplicity, we will mostly discuss things in terms of a binary classification problem where let's say we'll have to find if an image is of a cat or a dog. Or a patient is having cancer (positive) or is found healthy (negative). Some common terms to be clear with are:

True positives (TP): Predicted positive and are actually positive.
False positives (FP): Predicted positive and are actually negative.
True negatives (TN): Predicted negative and are actually negative.
False negatives (FN): Predicted negative and are actually positive.

Confusion matrix

It's just a representation of the above parameters in a matrix format. Better visualization is always good :)



Accuracy

The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Take for example a cancer detection model. The chances of actually having cancer are very low. Let's say out of 100, 90 of the patients don't have cancer and the remaining 10 actually have it. We don't want to miss on a patient who is having cancer but goes undetected (false negative). Detecting everyone as not having cancer gives an accuracy of 90% straight. The model did nothing here but just gave cancer free for all the 100 predictions.

We surely need better alternatives.

Precision

Percentage of positive instances out of the ***total predicted positive*** instances. Here denominator is the model prediction done as positive from the whole given dataset. Take it as to find out '*how much the model is right when it says it is right'.*

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate

Percentage of positive instances out of the ***total actual positive*** instances. Therefore denominator (*TP + FN)* here is the *actual* number of positive instances present in the dataset. Take it as to find out '*how much extra right ones, the model missed when it showed the right ones'.*

$$\frac{TP}{TP + FN}$$

Specificity

Percentage of negative instances out of the ***total actual negative*** instances. Therefore denominator (*TN + FP)* here is the *actual* number of negative instances present in the dataset. It is similar to recall but the shift is on the negative instances. *Like finding out how many healthy patients were not having cancer and were told they don't have cancer*. Kind of a measure to see how separate the classes are.

$$\frac{TN}{TN + FP}$$

F1 score

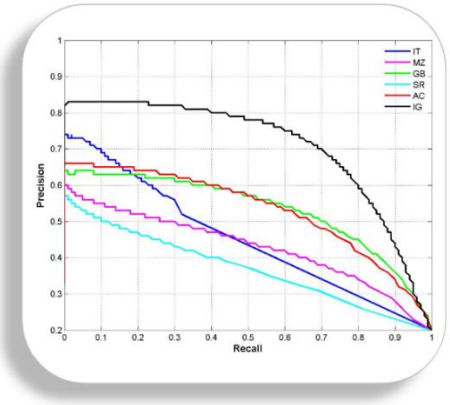
It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better. See that due to the product in the numerator if one goes low, the final F1 score goes down significantly. So a model does well in F1 score if the positive predicted are actually positives (precision) and doesn't miss out on positives and predicts them negative (recall).

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

One drawback is that both precision and recall are given equal importance due to which according to our application we may need one higher than the other and F1 score may not be the exact metric for it. Therefore either weighted-F1 score or seeing the PR or ROC curve can help.

PR curve

It is the curve between precision and recall for various threshold values. In the figure below we have 6 predictors showing their respective precision-recall curve for various threshold values. The top right part of the graph is the ideal space where we get high precision and recall. Based on our application we can choose the predictor and the threshold value. PR AUC is just the area under the curve. The higher its numerical value the better.

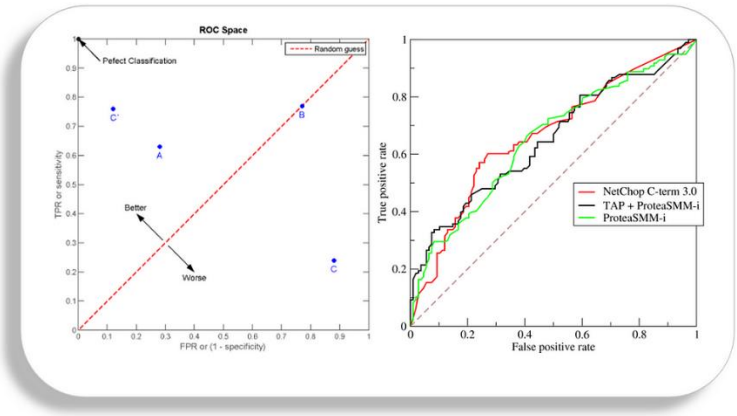


ROC curve

ROC stands for receiver operating characteristic and the graph is plotted against TPR and FPR for various threshold values. As TPR increases FPR also increases. As you can see in the first figure, we have four categories and we want the threshold value that leads us closer to the top left corner. Comparing different predictors (here 3) on a given dataset also becomes easy as you can see in figure 2, one can choose the threshold according to the application at hand. ROC AUC is just the area under the curve, the higher its numerical value the better.

$$True\ Positive\ Rate\ (TPR) = RECALL = \frac{TP}{TP+FN}$$

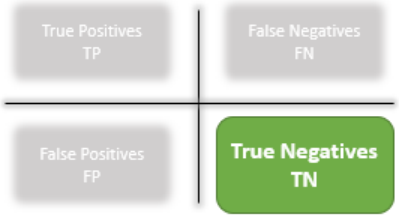
$$False\ Positive\ Rate\ (FPR) = 1 - Specificity = \frac{FP}{TN+FP}$$



PR vs ROC curve

Both the metrics are widely used to judge a models performance.

Which one to use PR or ROC?



The answer lies in TRUE NEGATIVES.

Due to the absence of TN in the precision-recall equation, they are useful in imbalanced classes. In the case of class imbalance when there is a majority of the negative class. The metric doesn't take much into consideration the high number of TRUE NEGATIVES of the negative class which is in majority, giving better resistance to the imbalance. This is important when the detection of the positive class is very important.

Like to detect cancer patients, which has a high class imbalance because very few have it out of all the diagnosed. We certainly don't want to miss on a person having cancer and going undetected (recall) and be sure the detected one is having it (precision).

Due to the consideration of TN or the negative class in the ROC equation, it is useful when both the classes are important to us. Like the detection of cats and dog. The importance of true negatives makes sure that both the classes are given importance, like the output of a ML/DL model in determining the image is of a cat or a dog.

Classification

Here the goal is to learn a mapping from inputs x to outputs y, where $y \in \{1,...,C\}$, with C being the number of classes. If C = 2, this is called binary classification (in which case we often assume $y \in \{0, 1\}$); if C > 2, this is called multiclass classification. If the class labels are not mutually exclusive (e.g., somebody may be classified as tall and strong), we call it multi-label classification, but this is best viewed as predicting multiple related binary class labels (a so-called multiple output model). When we use the term "classification", we will mean multiclass classification with a single output, unless we state otherwise.

One way to formalize the problem is as function approximation. We assume y = f(x) for some unknown function f, and the goal of learning is to estimate the function f given a labeled training set, and then to make predictions using $\hat{y} = \hat{f}(\mathbf{x})$. (We use the hat symbol to denote an estimate.) Our main goal is to make predictions on novel inputs, meaning ones that we have not seen before (this is called generalization), since predicting the response on the training set is easy.

Simple Linear Regression

Regression analysis may broadly be defined as the analysis of relationships among variables. This relationship is given as an equation that helps to predict the dependent variable Y through one or more independent variables. In regression analysis, the variable whose values vary with the variations in the values of the other variable(s) is called the **dependent variable** or **response variable**. The other variables which are independent in nature and influence the response variable are called **independent variables**, **predictor variables** or **regressor variables**.

Example: Suppose a statistician employed by a cold drink bottler is analysing the product delivery and service operation for vending machines. He would like to find how the delivery time taken by the delivery man to load and service a machine is related to the volume of delivery cases. The statistician visits 50 randomly chosen retailer shops having vending machines and observes the delivery time (in minutes) and the volume of delivery cases for each shop. He plots those 50 observations on a graph, which shows that an approximate linear relationship exists between the delivery time and delivery volume. If Y represents the delivery time and X, the delivery volume, the equation of a straight line relating these two variables may be given as

$$\mathbf{Y = a + bX \ldots (1)}$$

where a is the intercept and b, the slope.

In such cases, we draw a straight line in the form of equation (1) so that the data points generally fall near the straight line. Now, suppose the points do not fall exactly on the straight line. Then we should modify equation (1) to minimise the difference between the observed value of Y and that given by the straight line (a + b X). This is known as **error**.

The error e, which is the difference between the observed value and the predicted value of the variable of interest Y, may be conveniently assumed as a statistical error. This error term accounts for the variability in Y that cannot be explained by the linear relationship between X and Y. It may arise due to the effects of other factors. Thus, a more plausible model for the variable of interest (Y) may be given as

$$\mathbf{Y = a + b\,X + e \ldots (2)}$$

where the intercept a and the slope b are unknown constants and e is a random error component. Equation (2) is called a **linear regression model**.

Fitting of regression line

Let the given data of n pairs of observations on X and Y be as follows:

X: X_1 X_2 X_3 . … … Xi ….. … Xn

Y: Y_1 Y_2 Y_3 . … … Yi …..… Yn

where Y is the dependent variable and X, the independent variable.

Suppose, we wish to fit the following simple regression equation to the data: Y = a + bX where a is the intercept and b is the slope of the equation.

For fitting equation to the data on (X, Y), we follow the steps given below:
Step 1: We draw a scatter diagram by plotting the (X, Y) points given in data.
Step 2: We construct a table as given below and take the sum of the values of X_i, Y_i, X_iY_i, and X_i^2.
We write the values of $\sum X$, $\sum Y$, $\sum XY$ and $\sum X^2$ in the last row.
Step 3: We express of aˆ given in equation (1) as follows:

$$\hat{a} = \bar{Y} - b\bar{X} = \frac{1}{n}[\sum Y - b\sum X]$$

$$\hat{b} = \frac{n\sum XY - \sum X\sum Y}{n\sum X^2 - (\sum X)^2}$$

Where

substitute above values in the regression equation and get

$$\hat{Y} = \hat{a} + \hat{b}X$$

Multiple linear regression
Multiple linear regression is a method we can use to quantify the relationship between two or more predictor variables and a response variable.
The Regression Line: With one independent variable, we may write the regression equation as:

$$Y = a + bX + e$$

Where Y is an observed score on the dependent variable, *a* is the intercept, *b* is the slope, X is the observed score on the independent variable, and *e* is an error or residual.
We can extend this to any number of independent variables:

$$Y = a + b_1X_1 + b_2X_2 + \ldots + b_kX_k + e \quad (3.1)$$

Note that we have *k* independent variables and a slope for each. We still have one error and one intercept. Again we want to choose the estimates of *a* and *b* so as to minimize the sum of squared errors of prediction. The prediction equation is:

$$Y' = a + b_1X_1 + b_2X_2 + \ldots + b_kX_k \quad (3.2)$$

Finding the values of *b* (the slopes) is tricky for k>2 independent variables, and you really need matrix algebra to see the computations. It's simpler for k=2 IVs, which we will discuss here. But the basic ideas are the same no matter how many independent variables you have. If you understand the meaning of the slopes with two independent variables, you will likely be good no matter how many you have.
For the one variable case, the calculation of *b* and *a* was:

$$b = \frac{\sum xy}{\sum x^2}$$

$$a = \bar{Y} - b\bar{X}$$

For the two variable case:

$$b_1 = \frac{(\sum x_2^2)(\sum x_1y) - (\sum x_1x_2)(\sum x_2y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1x_2)^2}$$

and

$$b_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

At this point, you should notice that all the terms from the one variable case appear in the two variable case. In the two variable case, the other X variable also appears in the equation. For example, X_2 appears in the equation for b_1. Note that terms corresponding to the variance of both X variables occur in the slopes. Also note that a term corresponding to the covariance of X1 and X2 (sum of deviation cross-products) also appears in the formula for the slope.
The equation for *a* with two independent variables is:

$$a = \bar{Y} - b_1 \bar{X}_1 - b_2 \bar{X}_2$$

This equation is a straight-forward generalization of the case for one independent variable.

Regression metrics

Residual

The difference between the fitted value $\hat{Y}_i$ and Yi is known as the residual and is denoted by

$$r_i = Y_i - \hat{Y}_i, \qquad i = 1, 2, \ldots, n$$

The role of the residuals and its analysis is very important in regression modelling.

Mean Squared Error (MSE)

Mean squared error (MSE) measures the amount of error in statistical models. It assesses the average squared difference between the observed and predicted values. When a model has no error, the MSE equals zero. As model error increases, its value increases. The mean squared error is also known as the ***mean squared deviation (MSD)***.

For example, in regression, the mean squared error represents the average squared residual/error.

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n}$$

Where:
yi is the ith observed value.
ŷi is the corresponding predicted value.
n = the number of observations.

Squaring the error gives higher weight to the outliers, which results in a smooth gradient for small errors. Optimization algorithms benefit from this penalization for large errors as it is helpful in finding the optimum values for parameters. MSE will never be negative since the errors are squared. The value of the error ranges from zero to infinity. MSE increases exponentially with an increase in error. A good model will have an MSE value closer to zero.

Root Mean Square Error (RMSE)

Root Mean Squared Error (RMSE) is a popular metric used in machine learning and statistics to measure the accuracy of a predictive model. It quantifies the differences between predicted values and actual values, squaring the errors, taking the mean, and then finding the square root. RMSE provides a clear understanding of the model's performance, with lower values indicating better predictive accuracy.

RMSE is computed by taking the square root of MSE. RMSE is also called the Root Mean Square Deviation. It measures the average magnitude of the errors and is concerned with the deviations from the actual value. RMSE value with zero indicates that the model has a perfect fit. The lower the RMSE, the better the model and its predictions. A higher RMSE indicates that there is a large deviation from the residual to the ground truth. RMSE can be used with

different features as it helps in figuring out if the feature is improving the model's prediction or not.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Ref: https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics

Stochastic gradient descent (SGD)

Ref: https://machinelearningmastery.com/linear-regression-tutorial-using-gradient-descent-for-machine-learning/

Questions:

1. Consider the Table Contains the Average Annual Gold Rate from 1965 – 2022. Gold prices fluctuated throughout the year 2020 because of the COVID-19 epidemic. With gold functioning as a safe haven for investors, demand for the precious metal grew, and its price followed suit. During the epidemic, the stock market weakened, but it began to recover by the end of 2020 when the price of gold fell slightly.

It's crucial to remember that gold prices fluctuate during the year, and the figure below represents the average price for that year.

With the exception of a few lows shared across a few years, The table shows that the gold price trend has always been upward, supporting the claim that gold is a secure investment over extended periods of time.

Write a python program to find the fitted simple linear regression equation for the given data. Compare the coefficients obtained from sklearn model with your program. Compute the error, MSE and RMSE. Predict the gold price with the year 2025 for 1 gram.

This Table Contains the Average Annual Gold Rate from 1965 - 2022			
Year	Price (24 karat per 10 grams)	Year	Price (24 karat per 10 grams)
2022	₹ 52,950	1993	₹ 4,140
2021	₹ 50,045	1992	₹ 4,334
2020	₹ 48,651	1991	₹ 3,466
2019	₹ 35,220	1990	₹ 3,200
2018	₹ 31,438	1989	₹ 3,140
2017	₹ 29,667	1988	₹ 3,130
2016	₹ 28,623	1987	₹ 2,570
2015	₹ 26,343	1986	₹ 2,140
2014	₹ 28,006	1985	₹ 2,130
2013	₹ 29,600	1984	₹ 1,970
2012	₹ 31,050	1983	₹ 1,800
2011	₹ 26,400	1982	₹ 1,645
2010	₹ 18,500	1981	₹ 1,800
2009	₹ 14,500	1980	₹ 1,330
2008	₹ 12,500	1979	₹ 937
2007	₹ 10,800	1978	₹ 685
2006	₹ 8,400	1977	₹ 486
2005	₹ 7,000	1976	₹ 432
2004	₹ 5,850	1975	₹ 540
2003	₹ 5,600	1974	₹ 506
2002	₹ 4,990	1973	₹ 279
2001	₹ 4,300	1972	₹ 202
2000	₹ 4,400	1971	₹ 193
1999	₹ 4,234	1970	₹ 184
1998	₹ 4,045	1969	₹ 176
1997	₹ 4,725	1968	₹ 162
1996	₹ 5,160	1967	₹ 103
1995	₹ 4,680	1966	₹ 84
1994	₹ 4,598	1965	₹ 72

2. *Consider the Question no 1 gold price with following year-wise silver price. Write a python program to find the fitted multiple linear regression equation for the given data. Compare the coefficients obtained from sklearn model with your program. Compute the error, MSE and RMSE. Predict the gold and silver price with the year 2024 for 1 gram.*

Year	Silver Rates in Rs./Kg.	Year	Silver Rates in Rs./Kg.
1981	Rs.2715	2002	Rs.7875
1982	Rs.2720	2003	Rs.7695
1983	Rs.3105	2004	Rs.11770
1984	Rs.3570	2005	Rs.10675
1985	Rs.3955	2006	Rs.17405
1986	Rs.4015	2007	Rs.19520
1987	Rs.4794	2008	Rs.23625
1988	Rs.6066	2009	Rs.22165
1989	Rs.6755	2010	Rs.27255
1990	Rs.6463	2011	Rs.56900
1991	Rs.6646	2012	Rs.56290
1992	Rs.8040	2013	Rs.54030
1993	Rs.5489	2014	Rs.43070
1994	Rs.7124	2015	Rs.37825
1995	Rs.6335	2016	Rs.36990
1996	Rs.7346	2017	Rs.37825
1997	Rs.7345	2018	Rs.41400
1998	Rs.8560	2019	Rs.40600
1999	Rs.7615	2020	Rs.63435
2000	Rs.7900	2021	Rs.62572
2001	Rs.7215	2022	Rs.55100

Additional Questions

Write a python program for SGD by considering the year wise gold and silver price data. Compare the coefficients obtained from sklearn model with your program. Compute the error, MSE and RMSE. ***Predict the gold price with the year 2025 for 1 gram and gold and silver price with the year 2024 for 1 gram.***