ASSIGNMENT - 2

- Machine Learning -

Submitted by,
Shana Parween
S3 MCA
20MCA336.

Topic:-
Evaluating Model Performance : Precision & Recall, Sensitivity and specificity, Precision and Recall, F-measure, Cross Validation, K-fold Cross Validation, Bootstrap Sampling.

⇒ Evaluating Model Performance:-

- Process of evaluating ML algorithm.
- Algorithms have varying strengths and weaknesses.
- classifiers are evaluated, which means it reflects the type of data.
- There are 3 main types of data that are used to evaluate a classifier:
  - * Actual class value
  - * Predicted class values
  - * Estimated probability of the prediction.
- Goal : maintain two vectors [Actual and predicted class values] - must have same no. of values.

⇒ Confusion Matrix:-

- It is a table that categorizes predictions acco. to whether they match the actual value in the data.
- One of the table's dimensions indicates possible categories of predicted values while the other dimensions indicate same for actual values.

Actual class \ Predicted class	A	B
A	O	X
B	X	O

2 classes

Actual class \ Predicted class	A	B	C
A	O	X	X
B	X	O	X
C	X	X	O

3 classes.

- When the predicted value is same as actual value, this is a correct classification

- Correct predictions fall on the diagonal in the confusion matrix (denoted by O).
- The off-diagonal matrix cells (denoted by X) indicate cases where the predicted value differs from actual value. These are incorrect predictions.
- Performance measures for classification models are based on the counts of predictions falling on and off the diagonal in these tables.
- The most common performance measures consider the model's ability to discern one class versus all others.
- The class of interest is known as +ve class, while all others are known as -ve class.
- The relationship b/w +ve and -ve class predictions can be depicted as a 2x2 confusion matrix that tabulates whether predictions fall into one of 4 categories :-
  * True Positive (TP) - correctly classified as class of interest.
  * True Negative (TN) - correctly classified as not the class of interest
  * False Positive (FP) - Incorrectly classified as class of Interest
  * False Negative (FN) - Incorrectly classified as not the class of interest.
- eg:- Spam Classifier :-

	no	yes
no	TN	FP
yes	FN	TP

Actually spam

Predicted to be spam

⇒ Using Confusion Matrix to Measure Performance:-

- With 2x2 confusion matrix, we can formulate one def^n of prediction accuracy (success rate) as:-

$$\boxed{\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN}}$$

- The error rate, or proportion of incorrectly classified examples, is specified as:

$$\boxed{\text{error rate} = \frac{FP+FN}{TP+TN+FP+FN} = 1 - \text{accuracy}}$$

⇒ Sensitivity & Specificity:-

- Sensitivity of a model (also called the true positive rate), measures the proportion of +ve examples that were correctly classified.
- $\boxed{\text{Sensitivity} = \frac{TP}{TP+FN}}$
- Specificity of a model (also called true negative rate), measures the proportion of -ve examples that were correctly classified.
- $\boxed{\text{Specificity} = \frac{TN}{TN+FP}}$
- Given a confusion matrix for SMS classifier. Assuming that spam is the +ve class, we can confirm that the numbers in the confusionMatrix() output are correct. For eg, calculation for sensitivity is:

```
  sens <- 154/(154+29)
  sens
o/p→ 0.8415301
```

Similarly, for specificity we can calculate:

```
  spec <- 1202/(1202+5)
  spec
o/p → 0.9958575
```

- Sensitivity and specificity range from 0 to 1, with values close to 1 being more desirable.

⇒ Precision & Recall :-

- Closely related to sensitivity and specificity are two othee performance measures, related to compromise made in classif^n : precision and recall.
- Precision (also known as +ve predictive value) is defined as the proportion of +ve egs that are truly +ve.
- A precise model will only predict the +ve class in cases very likely to be +ve.

$$\boxed{\text{Precision} = \frac{TP}{TP+FP}}$$

- Recall is a measure of how complete the results are.
- This is same as sensitivity, only the interpretation differs.

$$\boxed{\text{Recall} = \frac{TP}{TP+FN}}$$

- A model with high recall captures a large portion of the +ve examples, means that it has wide breadth.
- eg:- An SMS spam filter has high recall, means that if majority of spam msgs are identified correctly.

⇒ F-Measure :-

- A measure of model performance that combines precision and recall into a single number is known as F-measure [also called F1-Score or F-Score].
- The F-measure combines precision and recall using the harmonic mean.
- The harmonic mean is used rather than the more common arithmetic mean since both precision & recall are expressed as proportions b/w 0 & 1.

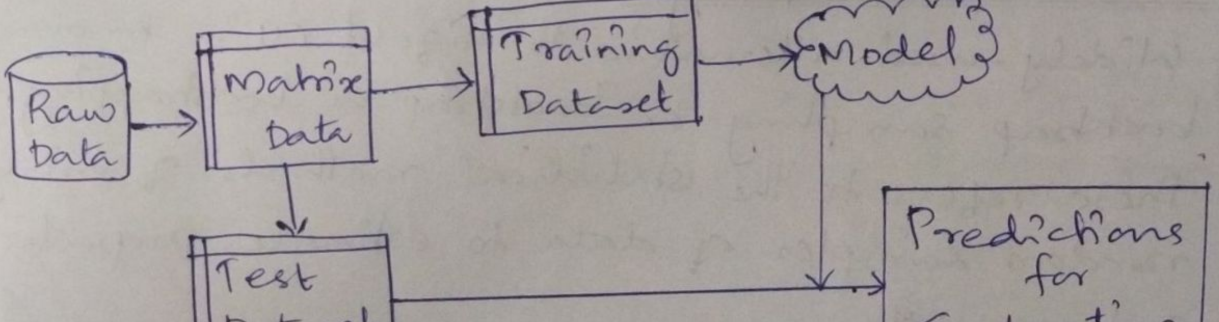
- Formula:-

$$\boxed{\begin{aligned} \text{F-measure} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{recall} + \text{precision}} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned}}$$

- Since the F-measure reduces the model performance to a single number, it provides a convenient way to compare several models side-by-side.
- Its possible to calculate F-scores using different weights for precision and recall, but choosing the weights can be tricky at best and arbitrary at worst.

⇒ The Holdout Method :-

- It is the procedure of partitioning data into training and test datasets.
- The training dataset is used to generate the model, which is then applied to the test dataset to generate predictions for evaluation.
- Typically, about 1/3 of data is held out for testing and 2/3 used for training.
- To ensure that training and test data do not have systematic differences, examples are randomly divided into 2 groups.



⇒ Cross-Validation :-

- The repealed hold-out is the basis of a technique known as k-fold cross-validation (k-fold CV), which has become the industry standard for estimating model performance.
- K-fold CV randomly divides data into 'k' completely separate random partitions called folds.
- Although k can be set to any number, the most common convention is to use 10-fold CV.
- For each of the 10 folds [each comprising 10% of the total data], a machine learning model is built on the remaining 90% of data.
- The fold's matching 10% sample is then used for model evaluation.
- After the process of training and evaluating model has occured for 10 times [with 10 different training/ testing combns], avg performance across all folds is reported.
- Datasets for cross validation can be created using 'createFolds ()' funcn.
- Similar to stratified random holdout sampling, this funcn will attempt to maintain same class balance in each of the folds as in the original dataset.

⇒ Bootstrap Sampling :-

- Widely-used alternative to k-fold CV is known as bootstrap sampling or bootstrap or bootstrapping.
- These refers to the statistical methods of using random samples of data to estimate properties of larger set.

- When this principle is applied to machine learning model performance, it implies the creation of several randomly selected training and test datasets, which are then used to estimate performance statistics.
- The results from various random datasets are then averaged to obtain final estimate of future performance.
- Difference:-

Cross-validation divides the data into separate partitions, in which each eg can appear only once. The bootstrap allows egs to be selected multiple times through a process of sampling with replacement.

- Using sampling with replacement, the probability that any given instance is included in training dataset is 63.2%. Consequently, pb of any instance being in test dataset is 36.8%.
- In other words, training data represents only 63.2% of available egs, some of which are repeated. In contrast with 10-fold CV, which uses 90% of egs for training, bootstrap sample is less representative of the full dataset.
- Final error rate is:

$$\text{Error} = 0.632 \times \text{error}_{\text{test}} + 0.368 \times \text{error}_{\text{train}}$$

- One advantage of bootstrap over cross-validation is that it tends to work better with very small datasets.