MLA- Session-3

Evaluation Metrics

- Evaluation metrics in machine learning are crucial for assessing how well a model performs on a given task. Depending on the problem type different metrics are used to measure the effectiveness and reliability of the model's predictions.

I. Classification Metrics

Accuracy

Precision

Recall / Sensitivity / True Positive Rate

f1 Score

II Regression Metrics

Mean Absolute Error

Mean Squared Error

Root Mean Squared Error

R-Squared

III clustering Metrics

Silhouette Score

Davies - Bouldin Index

Adjusted Rand Index

IV Ranking and Recommendation Metrics

- Mean Average Precision
- Normalized Discounted Cumulative gain (nDCG)
- Hit Rate

V Time Series Metrics

- Mean Absolute Percentage Error
- Symmetric Mean Absolute Percentage Error

VI NLP - Evaluation Metrics

- Bilingual Evaluation Understudy Score (BLEU)
- Recall - Oriented Understudy for Gisting Evaluation
- Perplexity

VII Image Segmentation Metrics

- Intersection over Union IoU
- Dice Coefficient

VIII Statistical Metrics

Correlation

Classification Metrics

Accuracy - Measure the proportion of correctly classified instances out of the total instances

$$\text{Accuracy} = \frac{\#\text{ Correctly classified instances}}{\#\text{ Total instances}}$$

Confusion Matrix

Predicted \ Actual	+	−
+	TP	FP
−	FN	TN

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

When the classes are balanced accuracy suits best in calculating the performance of the model.

Precision - How many selected items are relevant

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

When the false positives are costly this metric suits the best in calculating the performance of the model

Recall (Sensitivity or True Positive Rate)

- How many relevant items are selected

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

when the false- negatives are costly this metric suits the best in calculating the performance of the model.

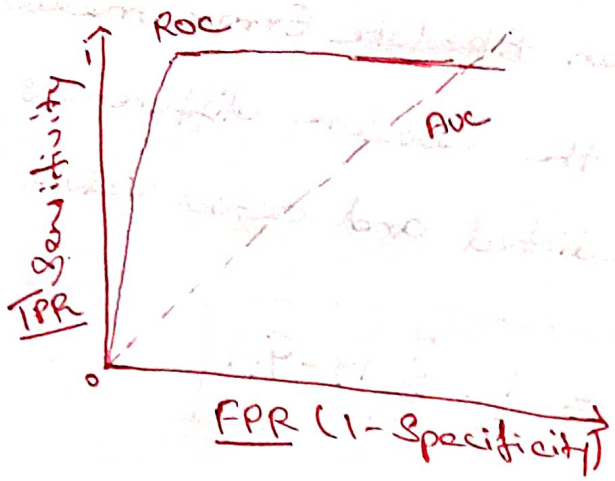
f1-Score: The harmonic mean of precision and recall. It balances the two when there is uneven class distribution.

$$f1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

when the class is imbalance, precision, recall, f1-score metric to be used for calculating/evaluating the performance of the system

AUC-ROC - Likelihood that the model will rank a random positive instance higher than a random negative instance.

Graphical representation of the performance of a binary classification model



When we need to evaluate model discrimination, can able to measure the performance at various thresholds.

Log Loss (Cross-Entropy Loss)

find accuracy of probabilistic classification models. It measures the difference betn. the actual class labels and the predicted probabilities.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(\hat{p}_i) + (1-y_i)\cdot\log(1-\hat{p}_i))$$

N - no. of samples.

y_i actual label (0 or 1) for the i^{th} sample

$\hat{p}_i$ predicted probability of the sample belonging to the class 1

Regression Metrics - To predict continous outputs

MAE - Mean Absolute Error measures the average of the absolute differences between predicted and actual values.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y - \hat{y}_i|$$

MSE - Mean Squared Error measures the average of the squared differences between predicted and Actual values.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

RMSE - Root Mean Square error

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

R-Squared - measures the proportion of variance in the dependent variable that is predictable from the independent variable

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$$R^2 = 1 - \frac{SS_{Res}}{}$$

Clustering Metrics: To

Silhouette Score - measures how similar an object is to its own cluster compared to other clusters

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

a(i) be the average distance between (i) and all other points in the cluster

b(i) be the average distance between i and all points in the nearest cluster.

Ranking and Recommendation Metrics:

P@k - Precision @k measures the proportion of relevant items in the top k results returned by the system

$$P@k = \frac{\text{No. of relevent items in top } k}{k}$$

R@k - Recall @k measures the proportion of relevent items that are found in the top k systems.

$$R@k = \frac{\#\text{ Relevant items at top } k}{\#\text{ of relevant items total}}$$

MAP – Mean Average precision scores across all users / queries.

– Average precision computes the precision after each relevant item is retrieved, average across all relevant items.

$$AP = \frac{1}{\#\text{relevant items}} \sum_{k=1}^{n} (P(k) \times rel(k))$$

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP_q$$

$P(k)$ precision @ k

$rel(k)$ – binary indicator for the relevance at rank k

Discounted Cumulative Gain

Measures the relevance of ranked items, with a logarithmic reduction in importance for lower-ranked items.

$$DCG = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}$$

rel_i – relevance of the item at rank i

Normalized DCG (nDCG)

$$\rightarrow nDCG = \frac{DCG}{IDCG}$$

Hit Rate: Measures whether atleast one relevant item is present in top k system

$$\text{Hit Rate} = \frac{1}{\text{Total Users.}} \sum_{i=1}^{N} 1(\text{atleast one relevant item in top k})$$

$$= \frac{\#\ \text{Relevant Items Recommended}}{\text{Total Recommended Items}}$$

Mean Reciprocal Rank

→ Measures the rank of the first relevant item in the ranked list.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank of the first relevant item.}}$$

Gradient Descent:

- Optimization algorithm used to minimize a function by iteratively moving towards the steepest descent, defined by the negative of the gradient

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$

θ → Parameters (weights)

α - Learning rate / Step size

$\nabla J(\theta)$ - Gradient of the cost function

↳ $J(\theta)$ - objective fn. - minimize

↳ $\nabla(J(\theta))$ - Vector of partial derivatives of the function

Iterations. The process is repeated until the model parameters converge to optimal values | the set of iterations is reached

① · Initialize Parameters.

② · Compute Gradient

③ Update the parameters.

$$\theta = \theta - \alpha \nabla J(\theta)$$

④ Repeat on 2 & 3 until convergence

Classification

Type of Supervised learning to predict a discrete label for a given input data point

- Assign one or more class labels to examples in a dataset based on their features.

→ Supervised learning.

→ Categorical output

→ Classification types

- binary classification
- Multiclass "
- Multilabel "

Spam Detection, Fraud Detection, Image Recognition, Sentiment Analysis, Credit Scor

Errors and Noise

Error - Difference between the predicted values and the actual values.

Noise - The random / unpredictable variation in the data that does not represent the underlying pattern / true signal.

Measurement errors.
Random fluctuations.
Outliers.
Missing / inaccurate data

Error directly impacts the model's performance as it measures how far off the predictions are from the actual values

* Errors can be minimized by improving the model, using better features, tuning hyper-parameters or collecting more data

Noise - Cannot be fully eliminated because it's inherent to the data.

* The best strategy for handling noise is to make the model more robust through regularization, cross-validation and outlier-detection

Parametric vs non-Parametric models.

Parametric model - Assumes a specific form for underlying function / distribution of data. The model summarizes the data with a fixed number of parameters which it tries to estimate during training.

Features:

- Number of parameters fixed

  The no. of parameters does not increase with the size of the dataset. For example in linear regression a fixed number of coefficients to estimate, irrespective of the amount of data

- → Strong assumptions about the data.

  These models assume a particular form for the function (linear, quadratic, Gaussian)

- → Simpler and faster

  Simple in computation and faster to train

- → Less flexible

  Since they make assumptions about the data parametric models might fail to capture

complex patterns.

Example:

Linear Regression → Assume the relationship between the independent feature (input var) & the dependent feature (output/target) is linear.

Logistic Regression → A binary classification model that assumes linear relationship between the input features and the log-odds of the target class.

Naive Bayes: Assumes that the features are continously independent of the target variable and the follow a specific distribution (Gaussian for continuous data)

Artificial Neural Networks: There are predefined layers and nodes, so the number of parameters is fixed and doesn't grow with the data size

Advantages:

- Simple to understand & Implement
- Less data required
- Faster training

Non-Parametric Models:

Doesn't assume any fixed form / distribution for the underlying data. Instead, it can adapt the complexity of the data and increase its capacity as more data becomes available

Features:

→ Flexible & data-driven - These models make fewer assumptions about the underlying data distribution. They are more flexible and can capture complex patterns.

→ Variable number of parameters.

The no. of parameters increases as the size of the dataset grows.

→ More data required:

Requires large dataset to perform well and avoid overfitting.

→ Slower to train and predict

Computationally expensive, both during training and when making predictions because they store and use more

information from the data

Examples:

KNN - kNearest Neighbors. - find the k-closest training examples for each test example to make predictions.

Decision Trees:

model splits the data based on the feature values, building a tree structure

The complexity of the tree depends on the data.

Support Vector Machines. - when using non-linear kernels, SVM become non-parametric as the number of support vectors grows with dataset size.

Gaussian Processes:

→ non-parametric Bayesian approach that defines a distribution over functions whose complexity grows with the data

Adv:

Flexible & Powerful, Fewer assumptions.

Good performance with large dataset.

Linear Algebra for Machine Learning.

- Mathematical foundation for representing and manipulating data, building models and optimizing algorithms.

Representation of Data

* Scalar - As numeric values can be represented.

* Vectors - Features of a data point are often represented as vectors.

ex: dataset of house prices
(each house represented in the form of 'Sq.feet, no. of rooms...')
Each feature is the element of the vector.

* Matrixes : A dataset with multiple data points or examples is often represented as matrix where rows represent individual data points & Columns represent features.

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_1 & x_2 & \dots & x_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

feature matrix label vector.

Tensor: A generalization of vectors and matrices to more than two-dimensions

Linear Models:

Linear Regression:

$y = mx + b$.

Logistic Regression - Applies a linear model to the data before applying sigmoid function to generate probabilities.

Dimensionality Reduction

- Matrix Decomposition.

Eigen Values and Eigen Vectors.

$Av = \lambda v$

$v \rightarrow$ Eigen vector

λ - Eigen Value for the matrix A

- Singular Value Decomposition (SVD)

Decomposes into three matrices

$U \Sigma V^*$

$A = U \Sigma V^A$

Transformations and Projections:

Linear Transformation - Scaling, rotation, translation on the vectors.

Projection - projecting data points from high dimensional space onto lower-dimensional subspace

Applied in PCA and feature extraction

Optimization and Gradient Descent

Gradient Descent are used to minimize loss functions and improve model performance

Gradients - tell us how the output of a model changes w.r.t its input (weights)
- Calculated using operations of algebra matrix derivatives, dot products

Neural Networks:

Feedforward Computation

input data are represented as vectors/ tensors multiplied by weighted matrices.

$$y = Wx + b$$

Backpropagation - gradient of loss function - matrix & vector derivatives.

Distance & Similarity measures:

Euclidean distance

- distance betn. two points.

Cosine Similarity. Measures the angle between two vectors

Kernel Methods: dot product betn. data points in high-dimensional space.

Eigenvalues & Eigenvectors → direction of Principal comp.

↳ Importance of each component

Spectral clustering → eigenvectors of graph's Laplacian matrix.

Stochastic Gradient Descent: efficiently optimize the models by updating weights iteratively based on a single data / small batch of points.

Regularization Techniques (L1, L2)

Norm of vector is a measure of its size/length

Euclidean Norm (L2)

L1 - Sum of absolute value of its coeff.

L2 - Sum of squared values " " "

Most effective Nodes to reduce computational cost