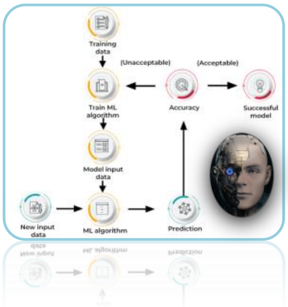
Machine Learning



Dr. Hrishikesh Bhaumik | hrishikesh.bhaumik@rcciit.org.in

ML

Quantification. Threshold fixing. ROC Graphics. ROC formulation.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

ML

Quantification

Quantification in machine learning refers to the process of **assigning numerical values** to certain **attributes, features, or aspects of data** in order to make them more amenable for **analysis** or **modeling**. It involves **converting qualitative** information into **quantitative data**, enabling algorithms to process and make decisions based on these values. **Quantification** is often used when dealing with **categorical** or **ordinal data** that do not have a natural numerical representation.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

ML

Quantification

There are different ways quantification can be applied in machine learning:

1. **Encoding Categorical Variables**: Categorical variables, which represent distinct **categories** or **labels**, need to be converted into **numerical values** for machine learning algorithms to work with them. Common techniques include:

- **Label Encoding**: Assigning a **unique integer** to each category. However, this may inadvertently create a hierarchy or order that doesn't exist in the data.
- **One-Hot Encoding**: Creating **binary columns** for each category, representing the presence or absence of that category. This avoids the hierarchical issue but can lead to increased dimensionality.
- **Ordinal Encoding**: When dealing with **ordinal variables** (categories with an inherent order), numerical values that reflect their order. For instance, a "low," "medium," and "high" rating can be encoded as 1, 2, and 3 respectively.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

Quantification

ML

3. **Feature Scaling**: In some cases, features might have different scales, which can affect the performance of algorithms. Scaling techniques like **standardization** (z-score normalization) or **min-max** scaling bring all features to a similar scale, helping algorithms converge faster and perform better.

4. **Quantization of Continuous Data**: Continuous numerical data can be **discretized** or **quantized** into bins. This can **simplify the representation** of data, **reduce noise**, and sometimes make the **data more interpretable**.

5. **Sentiment Analysis and Text Analysis**: In natural language processing (NLP), sentiment analysis involves quantifying the **emotional tone** of a piece of text. This might involve assigning a numerical value to represent the sentiment, such as a **positive or negative score**.

Dr. Hrishikesh Bhaumik hrishikesh.bhaumik@rcciit.org.in

Quantification

ML

6. **Feature Engineering**: Quantification can play a role in creating new features. For instance, you might calculate **statistical measures** (mean, median, standard deviation) from raw data and use these as features.

7. **Probability and Confidence Scores**: Many machine learning models output **probability scores** or **confidence scores** for their predictions. These scores quantify the model's level of certainty about its predictions.

Predicted Values \ Actual Values	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Dr. Hrishikesh Bhaumik hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

Threshold Fixing

ML

Threshold fixing is a **crucial aspect of classification problems** in machine learning, particularly when dealing with models that provide **probabilistic outputs** or **confidence scores**. Classification models often produce probabilities or scores that indicate the **likelihood of a data point** belonging to a certain class. However, to make a final decision about the predicted class, a **threshold** needs to be established that determines when a probability or score is considered high enough to classify a data point as belonging to a particular class.

In **binary classification problems** (where there are two possible classes), threshold fixing involves deciding whether a data point should be classified as the **positive class** or the **negative class** based on the model's output. The threshold is the value above which the predicted probability or score indicates a **positive prediction**, and below which it indicates a **negative prediction**.

Dr. Hrishikesh Bhaumik hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

Threshold Fixing

ML

Here are some key points to consider when fixing thresholds in machine learning:

1. **Default Threshold**: Some algorithms, like **logistic regression** or **support vector machines**, provide **probabilities** or **scores** as outputs. A common default threshold is 0.5, which means that if the predicted probability is greater than or equal to 0.5, the **positive class** is predicted; otherwise, the **negative class** is predicted.

2. **Precision and Recall Trade-off**: Adjusting the threshold can impact the **precision** and **recall** of your model. Lowering the threshold might increase recall (the ability to identify positive cases) but could decrease precision (the accuracy of positive predictions). Raising the threshold can have the opposite effect.

Dr. Hrishikesh Bhaumik hrishikesh.bhaumik@rcciit.org.in

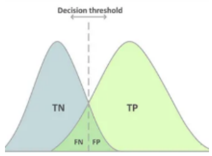
continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

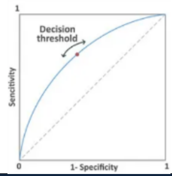
Threshold Fixing

ML

3. **Receiver Operating Characteristic (ROC) Curve**: The ROC curve plots the **true positive rate** against the **false positive rate** at various **threshold values**. It helps visualize the **trade-off** between **sensitivity** (recall) and **specificity** (1 - false positive rate) for different threshold levels.

4. **Area Under the Curve (AUC):** The AUC of the ROC curve provides an overall measure of the model's ability to discriminate between classes. A higher AUC indicates a better-performing model, but the choice of threshold still depends on your specific use case and priorities.





Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

Threshold Fixing

ML

5. **Domain Knowledge**: Consider **domain-specific factors** that might influence the **choice of threshold**. For instance, in medical diagnosis, the threshold might be set to favor higher precision to avoid false positives.

6. **Business Impact**: Evaluate the consequences of **false positives** and **false negatives** in the context of your application. For example, in fraud detection, false positives might inconvenience users, while false negatives could lead to significant financial losses.

Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

ROC Graphics

ML

ROC (Receiver Operating Characteristic) curves are graphical tools commonly used in machine learning to assess the **performance of binary classification models**. They help visualize the trade-off between the **true positive rate** (sensitivity) and the **false positive rate** (1 - specificity) as the discrimination threshold for classifying positive and negative samples is varied.

Interpreting and Creating ROC curves:

1. **True Positive Rate** (TPR) / **Sensitivity** / **Recall**: This is the ratio of correctly **predicted positive instances** to the **total actual positive instances**. It measures how well the model identifies the positive class.
TPR = TP / (TP + FN)

2. **False Positive Rate (FPR)**: This is the ratio of incorrectly **predicted positive instances** to the **total actual negative instances**. It measures how often the model makes a false positive prediction when the actual result is negative.
FPR = FP / (FP + TN)

Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

ROC Graphics

ML

3. **Creating an ROC Curve**:

- Obtaining the predicted probabilities or scores from the classification model for each instance in the test dataset.
- Vary the classification threshold from 0 to 1. For each threshold, classify instances as positive or negative based on whether their predicted probability is above or below the threshold.
- Calculate the TPR and FPR at each threshold value.
- Plot these TPR-FPR pairs on a graph, with TPR on the y-axis and FPR on the x-axis. Each point on the curve corresponds to a different threshold.

Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

ROC Graphics

4. **Interpreting the ROC Curve**:

- The ROC curve typically starts at the point (0, 0) and ends at the point (1, 1). The diagonal line represents the performance of a random classifier.
- A classifier's ROC curve above the diagonal indicates better-than-random performance. The closer the curve is to the upper-left corner (TPR = 1, FPR = 0), the better the model's performance.
- The area under the ROC curve (AUC-ROC) summarizes the model's performance across all thresholds. A higher AUC indicates a better-performing model. An AUC of 0.5 indicates random performance.

ML

ROC Graphics

5. **AUC-ROC Interpretation:**

AUC values range from 0 to 1, where:

0.5 represents random performance (no discrimination ability).

0.5 < AUC < 1 indicates better-than-random performance, with a higher AUC indicating better discrimination.

AUC = 1 represents a perfect classifier that can completely separate the classes.

6. **Choosing Models**:

When comparing multiple models, the one with a higher AUC-ROC tends to perform better in discriminating between classes. However, factors like domain knowledge and the specific trade-offs need to be considered.

Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

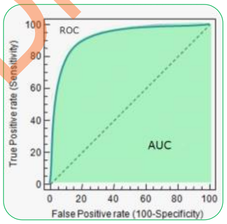
continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

ML

ROC Graphics

7. **Imbalanced Data**:

ROC curves can be informative even in cases of class imbalance, however, it might not provide a complete picture of model performance. Precision-Recall curves might be more suitable for highly imbalanced datasets.



Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

ML

Python, libraries like scikit-learn provide functions to calculate ROC curves :

Code:

```
import numpy as np
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

# Generate synthetic data for the example
X, y = make_classification(n_samples=1000, n_features=20, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a logistic regression classifier
clf = LogisticRegression()
clf.fit(X_train, y_train)

# Predict probabilities for the test set
y_pred_prob = clf.predict_proba(X_test)[:, 1]

# Compute the ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
```

Dr. Hrishikesh Bhaumik □ hrishikesh.bhaumik@rcciit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.

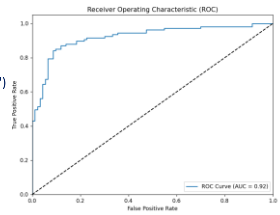
Python, libraries like scikit-learn provide functions to calculate ROC curves :

ML

Code:

```
# Compute the AUC (Area Under the Curve)
roc_auc = roc_auc_score(y_test, y_pred_prob)

# Plot the ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC)')
plt.legend(loc='lower right')
plt.show()
```



Dr. Hrishikesh Bhaumik hrishikesh.bhaumik@rccit.org.in

continue: Quantification. Threshold fixing. ROC Graphics. ROC formulation.