Text Classification and Responsible Classification

Applied Text Mining

Dr. Maryam Movahedifar

4-6 December 2024

University of Bremen, Germany
movahedm@uni-bremen.de

Universität
Bremen

DATA SCIENCE
CENTER

Outline

Introduction to Classification

Algorithms for Classification

Evaluation Metrics

Responsible Classification

Conclusion

Introduction to Classification

Types of Learning: Supervised vs. Unsupervised

- Supervised Learning:
  - The model is trained on labeled data, meaning each input has a corresponding output (label).
  - Example: Classifying emails as "Spam" or "Not Spam."
- Unsupervised Learning:
  - The model is trained on unlabeled data, meaning no predefined outputs are provided.
  - Example: Clustering news articles based on topics without knowing the categories.

Supervised Learning

- Definition: Learning from labeled data where each input has a known output (label).
- Label: The output or correct answer the model tries to predict.
- Examples:
  - **Text Classification:**
    - Input: "This book is amazing!"
    - Label: "Positive" sentiment
  - **Spam Detection:**
    - Input: "You have won a prize!"
    - Label: "Spam"
  - **Document Classification:**
    - Input: A news article about politics
    - Label: "Politics"

Unsupervised Learning

- Definition: Learning from unlabeled data, where the model discovers patterns without predefined outputs.
- No Labels: There are no predefined categories or answers provided.
- Examples:
  - **Clustering:**
    - Input: A collection of customer reviews
    - Output: Grouping them into clusters like "Positive" or "Negative" (discovered automatically).
  - **Topic Modeling:**
    - Input: A set of news articles
    - Output: Discovering topics such as "Sports," "Politics," etc., without prior labels.

Features, Prediction, Parameters, and Hyperparameters

In supervised learning, the key elements are the **Features** (input data), the **Prediction** (model output), the **Parameters** (model parameters that are trained), and **Hyperparameters** (user-defined settings). Understanding these elements is critical to building effective models.

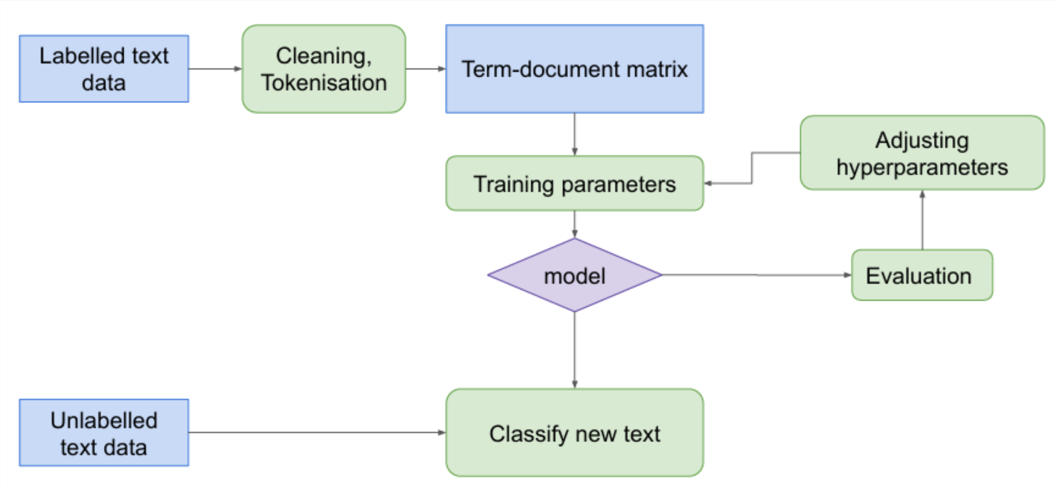
- Features: Information used to separate data into different classes.
- Prediction: The output of the model, compared with the correct labels (ground truth).
- Parameters: These are the values that the model learns during training (fitting parameters to data).
- Hyperparameters: These are the settings of the model that you define before training the model.

Types of Classification

Classification is a fundamental machine learning task, where the goal is to categorize data into predefined classes. There are two main types of classification:

- Binary Classification: Involves two categories, for example, true/false, spam/ham.
- Multiclass Classification: Involves multiple categories or classes, such as classifying articles into "sports," "technology," or "business."

Classification Workflow



Algorithms for Classification

Introduction to Classification Methods

In this section, we will explore various machine learning methods used for classification tasks. These methods are designed to help identify patterns in data and assign categories to new, unseen examples.

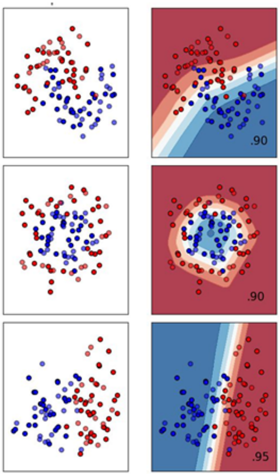
The following slides will go into detail about each of these methods and their applications in text classification.

- Logistic Regression
- Support Vector Machine (SVM)
- K-nearest Neighbours (KNN)
- Naive Bayes
- Decision Tree
- Ensemble Classifiers

Logistic Regression

Logistic Regression is used for binary classification tasks, predicting the probability of a given input belonging to one of two classes.

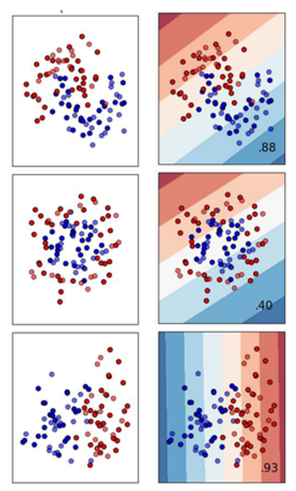
- **Assigns weights to input features** to calculate probabilities for classification.
- **Output is normalized** to a probability distribution (values between 0 and 1).
- **Pros:**
  - Simple and fast to train and provides probabilistic outputs
- **Cons:**
  - Assumes linear decision boundary and not suitable for non-linear data



Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification tasks. It works by finding a hyperplane that best separates data points of different classes.

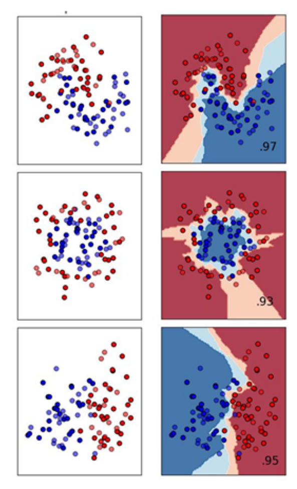
- **Finds a hyperplane** that separates the data into classes with maximum margin.
- Pros: Less sensitive to noisy data, making it effective for high-dimensional spaces.
- Cons: Requires linear separation, and performance may degrade if data is not linearly separable.



K-Nearest Neighbour (KNN)

K-Nearest Neighbour (KNN) is a simple, powerful classification algorithm that predicts the label of a text based on its ”nearest” neighbors in the training data.

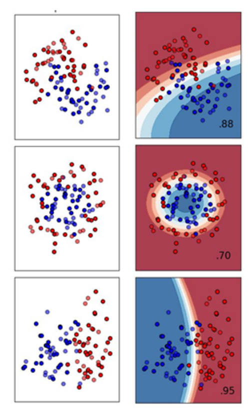
- **Classifies based on proximity** to the nearest neighbors in the training dataset.
- **Proximity measured** using the term-document matrix.
- **Takes the average** of the k nearest neighbors for classification.
- Pros: No assumptions about linearity or independence of features.
- Cons: For large datasets, quick to train but slow to classify.



Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' Theorem that assumes independence between features (words). It is simple and efficient for text classification.

- **Assumes independence:** Features (words) are assumed to be independent.
- **Estimates probabilities:** Calculates probability distributions for each word given the label.
- **Classify based on likelihood:** Uses probabilities to predict the most likely label for text.

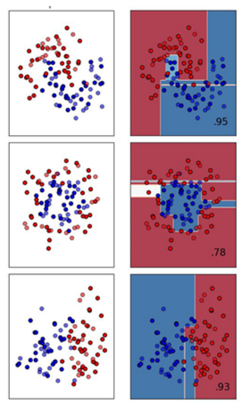


- **Pros:** Efficient for large datasets and works well with imperfect assumptions.
- **Cons:** Assumes feature independence and may struggle with correlated features.

Decision Tree

Decision trees are a type of model that splits data based on features, making decisions by choosing the most informative feature at each branch to classify data.

- **Generate a decision tree:** Choose the most informative feature at each branch to separate data.
- **Hyperparameter:** Maximum depth of the tree; deeper trees capture more detail.

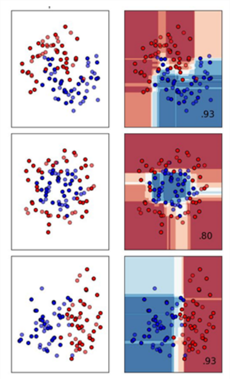


- **Pros:** Can capture complex relationships and is interpretable.
- **Cons:** Prone to overfitting, especially with deep trees, and sensitive to noisy data.

Ensemble Classifiers

Ensemble classifiers combine the predictions of multiple models to improve overall performance and reduce the volatility of single classifiers.

- **Random Forest Classifier:** Combines multiple decision trees and averages predictions to improve stability.
- **Voting Classifier:** Uses multiple classifiers to ”vote” on the result, with potential for classifiers of different types.



- **Pros:** Increased accuracy, reduces overfitting compared to a single model.
- **Cons:** Computationally expensive, requires more resources and time.

Evaluation Metrics

Accuracy and Precision

- Accuracy: Percentage of correct labels.
- Precision: Of the predicted positives, how many are correct?
- Recall: Of the actual positives, how many were correctly predicted?

$$F1\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Metrics: Multiclass Classification

- Evaluation in multiclass classification is less straightforward.
- We calculate Precision, Recall, and F1 for each class individually.
- But what if we want a single score for the entire model?

- Macro F1: Average of F1 scores for each class.
- Micro F1: Uses total number of true/false positives/negatives across all classes.

What Does Low Performance Mean?

Some potential explanations for low model performance:

- Noisy data or missing features.
- Not enough data to properly train the model.
- Underfitting or overfitting.

Responsible Classification

Responsible Classification

- Responsible Classification focuses on making fair and unbiased decisions.
- Ensures that machine learning models are transparent, ethical, and operate without harmful consequences.
- Involves addressing mentioned issues.

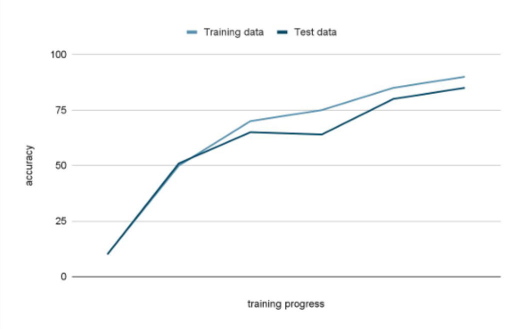
Noisy Data / Missing Features

- Garbage In, Garbage Out: A classifier learns only from the input provided; incorrect or incomplete data leads to poor model performance.
- Training Labels with Mistakes: Incorrect or noisy labels in the training data can mislead the model.
- Training Input Errors: Mistakes or inconsistencies in the features of the training data can affect the model's ability to learn correctly.
- Missing Essential Information: Sometimes, the features required to achieve perfect accuracy may simply not be present in the dataset.

Not Enough Data

Not Enough Data: The quality of the data might be fine, but our classifier has not seen enough examples during training.

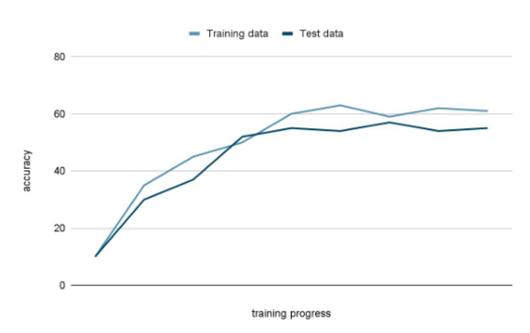
- The model is still improving as it learns more from the data.
- Training on more data can help the model generalize better and perform more accurately.



Underfitting

Underfitting: A classifier is underfitting when it is not capturing enough complexity of the data.

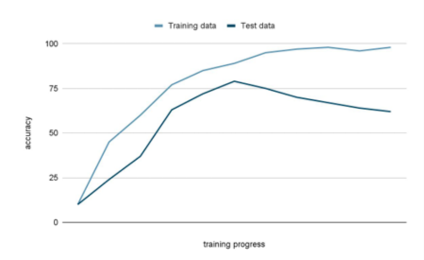
- The model performs poorly because it fails to capture important patterns.
- Underfitting does not improve with more data.
- A more complex model can capture the data's complexity and improve performance.



Overfitting

Overfitting: A classifier is overfitting when it matches the training data too closely, rather than capturing general trends.

- **High training performance:** Excellent accuracy on training data.
- **Poor generalization:** Struggles with new, unseen data.
- **Causes:**
  - Model complexity outweighs the available training data.
  - Model complexity exceeds the complexity of the actual problem.



Train/Validation/Test Split

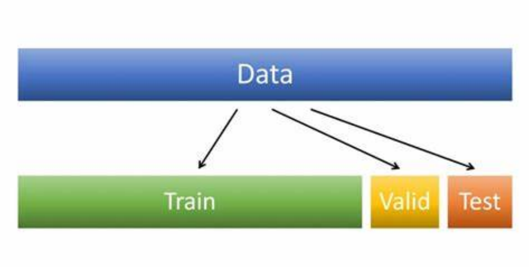
To build a reliable machine learning model, it's essential to split the data into three distinct sets to avoid overfitting and ensure robust evaluation.

- Train Set:
  - Used to optimize the model by adjusting its parameters.
  - The model learns patterns from this dataset.
- Validation Set:
  - Used to tune hyperparameters (e.g., learning rate, model depth).
  - Helps assess how well the model generalizes during training.
- Test Set:
  - Used for the final evaluation of the model's performance.
  - Provides an unbiased estimate of how the model performs on new data.

Visualizing the Data Split

Visual Overview:

- The dataset is divided into three parts: Train, Validation, and Test.
- Each part serves a unique role in building and evaluating the model.
- Ensures a balanced evaluation to avoid biased results.



Conclusion

Summary

- Text classification involves supervised learning to predict labels.
- Evaluating models requires understanding accuracy, precision, recall, and F1.
- Avoid overfitting and underfitting by careful model training.

Programming

Practical 2