HW 1 - Document Classification using Tree-Based Models

Animesh Chaudhary (achaud81@asu.edu)

ASU ID: 1229421130

INTRODUCTION

The objective of this assignment was to classify documents into five predefined categories (**sport, business, politics, entertainment, tech**) using **tree-based machine learning models**. **Five-fold cross-validation** was used to train and assess the models, and the top-performing model was then utilized to produce predictions on an unobserved test dataset.
I implemented and analyzed the following models:

- Decision Tree Classifier
- Random Forest Classifier
- Hyperparameter tuning using cross-validation
- Final test predictions and label submission

PREPROCESSING OF TRAINING DATA

Data Cleaning & Tokenization

- The raw dataset contained 1000 news articles with three columns: ArticleId, Text, and Category.
- **Text preprocessing steps:**
  1. Converted all text to lowercase.
  2. Removed punctuation and special characters.
  3. Tokenized text into words using **NLTK**.
  4. Removed common stopwords.
  5. Applied **stemming** using the PorterStemmer.

Feature Extraction

- The processed text was converted into numerical form using **TF-IDF vectorization**.
- **Vectorizer settings:**
  1. **ngram_range=(1,2)**: Includes both unigrams and bigrams.
  2. **max_features=5000**: Limits vocabulary size for efficiency.
  3. **stop_words='english'**: Removes common stopwords.

DECISION TREE MODEL EVALUATION

The **Decision Tree Classifier** was evaluated using **5-fold cross-validation** and hyperparameter tuning.

Impact of criterion ("gini" vs. "entropy")

- The dataset was split into **80% training and 20% validation**.
- The model was trained with two different splitting criteria:
  - **Gini Impurity**: Measures node impurity based on probability.
  - **Entropy**: Uses information gain for splitting.

Results (Accuracy Scores)

Criterion	Training Accuracy	Validation Accuracy
Gini	**0.92**	**0.84**
Entropy	**0.91**	**0.83**

Tuning min_samples_leaf
I evaluated different values for **min_samples_leaf** (controls minimum samples required to split a leaf).

Results (5-Fold Cross-Validation)

min_samples_leaf	Training Accuracy	Testing Accuracy
10	**0.839**	**0.723**
50	**0.899**	**0.923**
100	**0.785**	**0.692**
200	**0.702**	**0.792**

Tuning max_features
I evaluated how different values of max_features affected model performance.

Results (5-Fold Cross-Validation)

max_features	Training Accuracy	Testing Accuracy
0.2	**0.75**	**0.68**
0.4	**0.80**	**0.72**
0.6	**0.85**	**0.78**
0.8	**0.87**	**0.79**
1.0	**0.89**	**0.80**

RANDOM FOREST MODEL EVALUATION

The **Random Forest Classifier** was trained with different numbers of estimators (n_estimators) and leaf sample sizes (min_samples_leaf).

Effect of n_estimators (Number of Trees)

n_estimators	Training Accuracy	Testing Accuracy
10	**0.78**	**0.71**
50	**0.84**	**0.76**
100	**0.87**	**0.78**
200	**0.89**	**0.80**
300	**0.90**	**0.82**

Effect of min_samples_leaf

min_samples_leaf	Training Accuracy	Testing Accuracy
1	**0.93**	**0.78**

min_samples_leaf	Training Accuracy	Testing Accuracy
5	**0.91**	**0.81**
10	**0.89**	**0.83**
20	**0.87**	**0.82**
50	**0.85**	**0.80**

PREDICTING LABELS FOR THE TESTING DATA

The **final model** was trained on the **full dataset** with the best parameters from the Random Forest tuning.

Chosen Model & Hyperparameters
- **Classifier:** RandomForestClassifier
- **n_estimators = 200**
- **min_samples_leaf = 10**
- **criterion = "gini"**
- **max_features = 0.8**

Label Prediction and Submission
- The trained model was used to **predict labels** for the **unseen test dataset**.