WIDEBOT

Semantic Search in Articles using NLP

OVERVIEW

This project aims to classify English articles using text preprocessing, TF-IDF vectorization for feature extraction, KNN model, machine learning models for classification.

Data Loading and Preprocessing

Objective: **Load and preprocess text data for feature extraction and model training.**

1. **Dataset: Reuters-21578 dataset from the NLTK library.**

   1.1 Preprocessing Steps:

   - **Lowercasing: Convert text to lowercase for consistency.**
   - **Removing Punctuation: Strip out punctuation to focus on meaningful words.**
   - **Tokenization: Split the text into individual words (tokens).**
   - **Stop Words Removal: Remove common stop words using NLTK's stopwords list.**
   - **Stemming: Reduce words to their root forms using the Porter Stemmer.**

1.2Feature Extraction: TF-IDF Vectorization

Objective: Transform text data into numerical features using TF-IDF vectorization.

1. **TF-IDF Vectorizer: Convert the text data into TF-IDF features.**
   - ○ **max_features: Set to 1000 to limit the number of features to the top 1000 terms.**
2. **Vectorization:**
   - ○ **Fit the vectorizer on the training data and transform both the validation and test data.**
   - ○ **Extract keywords and their TF-IDF scores.**

1.3Classification Models

Objective: Train and evaluate classification models on the extracted features.

1. **k-Nearest Neighbors (KNN):**
   - ○ **Distance Metric: Cosine similarity.**
   - ○ **Neighbors: Set to 10.**
2. **Random Forest:**
   - ○ **Estimators: Set to 180.**

Model Evaluation

Objective: Evaluate model performance using accuracy and classification metrics.

1. **Accuracy: Measure the overall correctness of the model.**
2. **Classification Report: Provides precision, recall, and F1-score for each class.**

Results and Discussion

1. **Test Accuracy:**
    - **Random Forest**: 84%
    - **KNN**: 82%
2. **Classification Report: Detailed performance metrics for each class.**

Observations:

- **Random Forest performed slightly better than KNN.**
- **Some categories have low recall and precision, indicating difficulty in classifying those specific categories.**

Conclusion

The project demonstrates the effectiveness of TF-IDF vectorization for text classification using machine learning models. While the Random Forest model outperformed KNN.

Additional Requirements:

Tools Used:

- **NLTK for text preprocessing and tokenization**
- **Scikit-learn for TF-IDF vectorization, train-test split, encoding, K-Nearest Neighbors, Random Forest, and evaluation metrics**
- **NumPy for numerical operations**

External Resources:

- **Reuters dataset from NLTK**

Captioned Tables

Dataset Split Sizes

Set	Number of Articles
Training	5000
Validation	1500
Testing	3500

Top 10 Keywords by TF-IDF Score

Rank	Keyword	TF-IDF Score
1	'said'	High Value
2	'year'	High Value
3	'market'	High Value
...	...	...
10	'company'	High Value

Model Accuracy Comparison

Model	Accuracy
KNN	0.82
Random Forest	0.84

Reflection Questions:
The biggest challenge was in using new models and new techniques .

I have learned a lot of things in NLP and how to tune the models in the right way.