Topic Modeling (News or Research Articles)- 10/10/25(Friday):

I. Project Foundation and Data Acquisition

A. Project Objective: Supervised Topic Classification

The core objective of this project, despite the general label of "Topic Modeling," was executed as **Supervised Topic Classification**. This distinction is critical: the goal was not to discover latent, unknown topics (unsupervised modeling), but rather to train a machine learning algorithm to accurately assign incoming documents to a known, predefined set of target labels (the 35+ topics present in the dataset).

B. Data Sourcing and Preparation in a Cloud Environment

To fulfill the requirement of utilizing a cloud environment without pre-downloaded files, the industry-standard **20 Newsgroups dataset** was sourced directly via the Scikit-learn utility. This dataset, which combines major categories (like 'Business' and 'Sports') with dozens of highly specific academic sub-categories (e.g., specific physics and condensed matter topics), served as the foundation for the multi-class classification task.

The initial dataset was loaded and immediately processed for preparation, acknowledging the multi-class nature of the data which requires robust evaluation metrics.

II. Data Cleaning and Linguistic Normalization (NLTK Pipeline)

The performance of any text classifier is dependent on the quality of the linguistic preparation. A comprehensive text processing pipeline, driven by the Natural Language Toolkit (NLTK), was implemented to clean the raw data and standardize word forms.

A. Structural Noise Reduction

Initial cleaning targeted structural noise, which dilutes the true topical signal:

1. **Lowercasing:** All text was converted to lowercase to ensure uniformity.
2. **HTML and URL Removal:** Standard patterns and HTML tags frequently found in web and newsgroup data were stripped out.
3. **Metadata Cleaning:** Noise specific to the newsgroup format (such as header lines and email patterns) was removed using regular expressions.

B. Linguistic Normalization via POS-aware Lemmatization

The core linguistic step involved normalization to ensure that different word inflections (e.g., "running," "ran," and "runs") map to a single root form. **Lemmatization** was chosen over stemming because it uses a lexical dictionary (WordNet) to return a valid, meaningful base word (or lemma, e.g., "running" becomes "run," "better" becomes "good").

For optimal accuracy, a Part-of-Speech (POS) tagging step was integrated. This assigned grammatical context (noun, verb, adjective) to each word, which was then fed to the lemmatizer, resulting in a

vocabulary that is semantically richer and more concise. The final steps included **Tokenization** (breaking text into words) and **Stop Word Removal** (filtering common, non-informative words like "the," "is," and "a").

III. Feature Engineering: TF-IDF Vectorization

Feature engineering is the process of converting the clean text into a numerical matrix that machine learning models can process. **Term Frequency-Inverse Document Frequency (TF-IDF)** was selected for this task.

TF-IDF assigns a numerical weight to each word, prioritizing terms that appear frequently within a specific document (high Term Frequency) but rarely across the entire collection (high Inverse Document Frequency). This weighting scheme creates features highly effective at discriminating between topics.

The vectorization process was executed with a strict data split (70% training, 30% testing) maintained beforehand to prevent **data leakage**. The vectorizer was fit only on the training data and then applied to the test data. Importantly, the process included up to 20,000 features and incorporated both **unigrams** (single words) and **bigrams** (two-word phrases, like "stock market"), which enhances the capture of contextual meaning.

IV. Comparative Model Training and Selection

Four different classifiers were trained and tested to establish a reliable baseline and identify the optimal model:

1. **Multinomial Naive Bayes (MNB):** Used as the primary baseline model, selected for its speed and effectiveness with count-based (or TF-IDF) text features.
2. **Logistic Regression (LR):** A strong linear discriminative model often used for comparative analysis.
3. **Linear Support Vector Machine (Linear SVM):** A powerful linear classifier known for its efficiency in high-dimensional sparse spaces, typically achieving high accuracy in text classification.
4. **Random Forest:** A non-linear ensemble model included to test a different approach against the linear models.

V. Evaluation, Analysis, and Final Selection

A. Mandatory Metrics and Averaging Strategy

Model performance was rigorously evaluated using the required metrics: Accuracy, Precision, Recall, and the F1 Score. Due to the multi-class nature of the 35+ topics and the resulting class imbalance (where some topics have many more documents than others), the **Weighted Average** method was used for aggregation. This is the most professional and reliable approach, as it weights each class's metric by its representation (support) in the test set.

B. Comparative Model Performance (Weighted Average)

Metric	Multinomial Naive Bayes (Baseline)	Logistic Regression	Linear SVM	Random Forest

Accuracy	0.802	0.835	0.856	0.777
Precision (Weighted)	0.779	0.813	0.853	0.752
Recall (Weighted)	0.802	0.835	0.856	0.777
F1 Score (Weighted)	0.777	0.813	0.853	0.751

Final Selection: Based on the empirical results, the **Linear SVM** classifier demonstrated superior performance across all metrics, achieving the highest Weighted F1 Score of **0.853**. This confirms the Linear SVM as the optimal choice for this task, as its structure is highly effective at finding the best separating boundary in the dense, high-dimensional TF-IDF space.

C. Analysis of Classification Errors (Confusion Matrix Diagnosis)

The detailed performance reports and the Confusion Matrix (Image 1) reveal a critical structural issue in the dataset: **Severe Class Imbalance**.

While the major, well-represented categories (like 'Business' and 'Sports') were classified with high accuracy (Recall often above 0.85), many of the specific academic sub-classes (e.g., cond-mat.other, various math topics) had extremely low support (as low as 4 to 10 instances in the test set).

The resulting warning regarding "Undefined Metric" and the F1 scores of 0.00 for these niche classes indicates a complete failure to classify them. This occurs because the models are biased toward predicting the larger classes and effectively ignore the rarely seen categories.

VI. Conclusion and Professional Recommendations

A. Project Achievement Summary

The project successfully established a robust, supervised topic classification pipeline. Through meticulous NLTK-based data cleaning (including POS-aware lemmatization, which is superior to basic stemming) and effective feature engineering via TF-IDF, the Linear SVM model was trained and validated, achieving a Weighted F1 Score of 0.853.

B. Recommendations for Future Improvement

The primary barrier to achieving even higher performance, especially on the neglected rare classes, is the identified class imbalance. Recommendations for elevated performance include:

1. **Class Aggregation (Data Strategy):** The most effective improvement would be to aggregate the numerous low-support academic sub-classes (e.g., cond-mat.mes-hall, math.DG, etc.) into broader, meaningful categories (e.g., "General Condensed Matter Physics" or "General Mathematics"). This concentrates the learning signal and eliminates the classification failures currently observed in the Confusion Matrix.

2. **Hyperparameter Optimization:** While the model performed well, fine-tuning the Linear SVM's regularization strength (the C parameter) using techniques like Grid Search would ensure the parameters are perfectly optimized for the calculated TF-IDF feature set.
3. **Advanced Baseline:** Replacing the standard Multinomial Naive Bayes (MNB) with the **Complement Naive Bayes (CNB)** algorithm is advisable, as CNB is structurally designed to handle the effects of severe class imbalance better than MNB, providing a more reliable low-end benchmark.

Project Overview and Methodology-A short summary:

This project executed a comprehensive machine learning pipeline for **Supervised Topic Classification** using classical Natural Language Processing (NLP) techniques in a cloud environment (Google Colab). The goal was to accurately assign documents from a complex, multi-class dataset (the 20 Newsgroups corpus, including general and specialized research articles) to their corresponding topic labels.

The pipeline followed these key steps:

1. **Data Preparation and Cleaning:** The raw text data underwent rigorous preprocessing, including stripping structural noise (HTML, URLs, email metadata), removal of punctuation, and tokenization. Crucially, **POS-aware Lemmatization** was applied (superior to simple stemming) to reduce words to their semantically correct dictionary base form, ensuring feature efficiency.
2. **Feature Engineering:** The cleaned text was converted into a numerical matrix using **TF-IDF (Term Frequency-Inverse Document Frequency) Vectorization**. TF-IDF assigned weights to words based on their importance within a document relative to the entire corpus, creating sparse features effective for topic discrimination.
3. **Comparative Modeling:** Four classifiers were trained: Multinomial Naive Bayes (Baseline), Logistic Regression, Linear Support Vector Machine (Linear SVM), and Random Forest.

Evaluation Metrics and Mathematical Formulas

The model performance was evaluated using standard metrics, aggregated using the **Weighted Average** method to account for class imbalance (the unequal number of documents per topic).

The definitive comparative metric used was the F1 Score, which is the harmonic mean of Precision and Recall.

Precision measures the accuracy of positive predictions:

Precision=True Positives+False PositivesTrue Positives

Recall measures the completeness, or the ability to find all positive samples:

Recall=True Positives+False NegativesTrue Positives

The **F1 Score** combines these two metrics:

F1 Score=2×Precision+RecallPrecision×Recall

Metric	Logistic Regression	Linear SVM
Accuracy	0.835	0.856
F1 Score (Weighted)	0.813	0.853

The **Linear SVM** achieved the highest performance metrics, culminating in a Weighted F1 Score of **0.853**. This score confirms its selection as the optimal classifier, demonstrating superior ability to create effective separation boundaries in the high-dimensional TF-IDF feature space compared to the other models.

Critical Diagnosis: Class Imbalance The evaluation revealed significant difficulty in classifying highly specific, low-volume academic topics (e.g., certain physics and mathematics sub-classes). The failure to predict documents in these rare classes resulted in numerous 0.00 scores in the classification report and required the system to issue *Undefined Metric Warnings*. This highlights a common challenge in real-world datasets and underscores the necessity of either collecting more data for these rare classes or **aggregating** them into broader categories to improve generalized performance.