Project Title	Topic Modeling on The Indian Express News Article
Skills take away From This Project	Natural Language Processing (NLP), Text Preprocessing, Topic Modeling, Feature Engineering for Text, TF-IDF Vectorization, CountVectorizer, Data Visualization, supervised Learning, Exploratory Data Analysis (EDA), Project Documentation, Python (NLTK, SpaCy, Gensim, Scikit-learn, Matplotlib, Pandas)
Domain	Media and Journalism Analytics

Background:

In the fast-paced digital world, news platforms generate massive volumes of content across various domains every day. ***The Indian Express*****,** a prominent Indian news source, covers a wide range of topics such as Business, Technology, Sports, Education, and Entertainment. Understanding the distribution and nature of these news topics can provide valuable insights into national trends and media focus.

With the growing amount of textual data, manually organizing and analyzing such content becomes impractical. This project leverages **Natural Language Processing (NLP)** techniques to build a supervised topic classification model that can automatically classify news articles into predefined categories based on their content.

By using labeled data scraped from *The Indian Express*, this project aims to:

- Train machine learning models that can accurately classify news articles,
- Understand how linguistic features vary across different news categories,
- Enable faster and automated topic-based tagging of large news datasets.

This classification task demonstrates the application of NLP in the field of **Media and Journalism Analytics**, helping streamline content categorization and aiding in editorial decision-making and media monitoring.

Problem Statement:

With the exponential growth of digital news content, it becomes increasingly difficult to manually classify and organize articles by topic or category. *The Indian Express*, a leading Indian news outlet, publishes thousands of articles across various domains such as Business, Technology, Sports, Education, and Entertainment.

While these articles are categorized by domain, the classification is currently based on editorial tagging or web structure. Automating this classification process using machine learning can significantly reduce manual effort, ensure consistency, and enable real-time content analysis.

The primary objective of this project is to build a **supervised machine learning model** that can accurately classify a given news article into one of the five predefined categories using its textual features like headline, description, and full content.

This solution can:

- Improve searchability and recommendation systems on news platforms,
- Support trend analysis across categories,
- Assist in organizing and filtering content more efficiently.

Approach:

To classify news articles into predefined categories, this project follows a structured supervised machine learning pipeline that combines Natural Language Processing (NLP) techniques with classification algorithms.

1. Data Understanding & Exploration

- Load and inspect the dataset containing headlines, descriptions, content, URLs, and category labels.
- Perform Exploratory Data Analysis (EDA) to understand:
  - Distribution of articles across categories
  - Average word counts per category
  - Common terms used in each news category

2. Text Preprocessing

- Combine the text fields (Headline, Description, and Content) for a unified input.
- Apply standard preprocessing techniques:
  - Convert text to lowercase
  - Remove punctuation, numbers, and special characters
  - Tokenize the text
  - Remove stop words
  - Apply lemmatization to normalize the text

3. Feature Extraction

- Convert preprocessed text into numerical features using:

    - TF-IDF Vectorizer (with word-level or n-gram configuration)
    - Optionally experiment with CountVectorizer
- Evaluate both word-level and character-level n-grams to improve representation

4. Model Building

- Train and evaluate various supervised learning models including:
    - Logistic Regression (as a baseline)
    - Multinomial Naive Bayes
    - Random Forest Classifier
    - XGBoost Classifier
    - Optionally explore deep learning models such as LSTM or Transformer-based models like BERT

5. Model Evaluation

- Evaluate model performance using:
    - Accuracy
    - Precision, Recall, F1-Score
    - Confusion Matrix
    - Cross-validation to assess generalization

6. Hyperparameter Tuning

- Use GridSearchCV or RandomizedSearchCV to fine-tune parameters such as n_estimators, max_depth, and learning_rate

7. Final Output and Inference

- Generate final predictions on new/unseen data
- Visualize performance across categories using confusion matrix and performance metrics
- Save the trained model using joblib or pickle for deployment or integration with applications

.Project Evaluation Metrics

- **Accuracy**: Measures the overall correctness of the model by calculating the proportion of correctly predicted instances.
- **Precision**: Indicates how many of the articles predicted for a particular category are actually correct, helping minimize false positives.
- **Recall**: Measures how many actual articles in a category were correctly predicted, helping reduce false negatives.

- **F1-Score**: Provides a balance between precision and recall, especially useful when the dataset is imbalanced.
- **Confusion Matrix**: Displays a matrix of actual vs. predicted labels to analyze classification errors across categories.
- **Cross-Validation Score**: Ensures the model's generalizability by evaluating performance across multiple data splits.
- **Classification Report**: Summarizes precision, recall, F1-score, and support for each category, giving detailed insights into model performance.

Results:

By completing this project, the following outcomes were achieved:

- Successfully built a supervised machine learning model capable of classifying news articles from *The Indian Express* into five categories: Business, Technology, Sports, Education, and Entertainment.
- Demonstrated the effective use of NLP techniques such as text preprocessing, TF-IDF vectorization, and topic classification.
- Understood the importance of combining multiple text fields (headline, description, content) to improve classification accuracy.
- Gained hands-on experience in evaluating models using metrics like accuracy, precision, recall, F1-score, and confusion matrix.
- Learned how to apply cross-validation to ensure model robustness and generalization.
- Identified patterns and linguistic features that differentiate one news category from another.
- Strengthened understanding of how machine learning can be applied in real-world domains like media and journalism.

This project reinforces practical knowledge of text classification workflows and prepares learners for building scalable NLP solutions in industry settings.

Technical Tags:

Natural Language Processing, Text Classification, Supervised Learning, TF-IDF Vectorization, Text Preprocessing, Topic Classification, Scikit-learn, Pandas, NumPy, Matplotlib, NLTK, SpaCy, XGBoost, Logistic Regression, Multinomial Naive Bayes, Random Forest, Cross-Validation, Confusion Matrix

Data Set:

Data Set Link: Indian Express

About the Dataset:

This dataset provides a comprehensive collection of **10,000 news articles** sourced from *The Indian Express*, one of India's most reputable news publications. It spans five major domains: **Business, Technology, Sports, Education, and Entertainment**, with **2,000 articles per category**, ensuring a balanced distribution.

The articles were web-scraped and curated to focus exclusively on **Indian news and developments**, offering a valuable resource for Natural Language Processing (NLP) and Machine Learning (ML) applications.

Dataset Summary:

- **Total Rows**: 10,000
- **Total Columns**: 5
- **Language**: English
- **Focus Area**: Indian national events and topics

Column Descriptions:

- **Headlines**: The main title or headline of the news article.
- **Description**: A brief summary or snippet introducing the news article.
- **Content**: The full text body of the article, containing detailed information.
- **URL**: The link to the original source of the article on *The Indian Express* website.
- **Category**: The labeled domain of the article (e.g., Business, Education, Entertainment, Sports, Technology).

Credits:

The dataset content belongs to *The Indian Express* and is intended for academic and research purposes only.

Project Deliverables:

The following deliverables have been produced as part of this project:

1. Source Code and Notebooks

- Python scripts and Jupyter Notebooks for:
  - Data loading and preprocessing
  - Text cleaning and normalization
  - Feature extraction using TF-IDF
  - Model training and evaluation
  - Cross-validation and hyperparameter tuning

2. Trained Models

- Saved versions of trained machine learning models including:
  - Logistic Regression
  - Multinomial Naive Bayes
  - Random Forest
  - XGBoost
- Models saved in .joblib or .pkl format for reuse or deployment

3. Predictions and Outputs

- Final prediction files in CSV format with predicted categories
- Confusion matrix images and performance plots for visual analysis
- Classification reports summarizing precision, recall, and F1-score

4. Documentation

- Detailed project report covering:
  - Problem statement
  - Dataset description
  - Approach and methodology
  - Evaluation metrics
  - Results and insights
  - Challenges faced and learnings
- Clear explanations of key NLP concepts used in the project

Project Guidelines

- Text preprocessing steps included lowercasing, punctuation and number removal, tokenization, stopword removal, and lemmatization to clean and normalize the news content.
- Feature extraction must be performed using TF-IDF Vectorizer, with experiments on word-level and n-gram representations to enhance context capture.
- Multiple machine learning models were developed and evaluated, including Logistic Regression, Multinomial Naive Bayes, Random Forest, and XGBoost.
- Model performance must be measured using accuracy, precision, recall, F1-score, and confusion matrix, with cross-validation applied to ensure generalization.
- A series of experiments must be conducted by varying algorithms, vectorization techniques, and hyperparameters, with performance results logged for comparison.
- Final predictions must be saved in a CSV file format with appropriate column naming for category inference.
- The entire codebase must be developed using Python and Jupyter Notebooks, maintaining modularity and clear in-line documentation.
- Theoretical concepts such as TF-IDF vs. CountVectorizer, importance of lemmatization and n-grams, and classification metrics must be addressed and explained in the documentation.

Approval Workflow:

Created By:	Verified By:	Approved By:
Santhosh N		