Project Title	**Multi-Model Article Classification and Cloud Deployment Using Hugging Face, Streamlit, AWS EC2, and RDS**
Skills take away From This Project	**Text Classification, Machine Learning, Deep Learning, Transformers, Hugging Face Models, LLM, Streamlit or Gradio, AWS**
Domain	**AIOPS,MLOPS, NLP**

Problem Statement:

The objective is to develop and deploy a complete article classification system that leverages Machine Learning, Deep Learning, and fine-tuned Transformer-based models using Hugging Face libraries. The system must classify news articles into predefined categories such as World, Business, Sports, and Technology. Students are required to build a secure, scalable, and user-friendly web application using Streamlit, and deploy the entire solution on AWS infrastructure—including EC2 for hosting, S3 for model storage (if applicable), and RDS for logging user credentials and login activity. The application should allow users to input article content, choose a preferred model, view predictions with confidence scores, and store login information in a secure cloud database.

Objective:

To develop and deploy a robust article classification system that utilizes Machine Learning, Deep Learning, and Hugging Face Transformer models, making it accessible through a Streamlit-based web application hosted on AWS. The system should accurately classify input article content into predefined categories and securely track user session details, including login credentials and timestamps, using an AWS RDS database.

Business Use Cases:

1. Automated Content Tagging for News Portals and Blogs:
    a. Automatically classify news articles into relevant categories to enhance content organization, improve search functionality, and boost user engagement.
2. Smart Content Routing for E-Learning Platforms:
    a. Enable personalized course recommendations or content curation by classifying articles into educational domains (e.g., Technology, Business, World Affairs).
3. Real-Time News Categorization for Media Dashboards:
    a. Facilitate trend analysis and topic-based reporting by grouping and visualizing incoming articles by category using dynamic dashboards.
4. Academic Submission Classification:
    a. Automatically detect the subject area of student-submitted essays or reports to assist in academic categorization and plagiarism detection workflows.

.**Approach:**

1. Data Preparation

- Dataset: AG News Topic Classification Dataset
- Splitting:
    - ✓ Use both train.csv and test.csv.
    - ✓ Limit to 50% subset if needed **(ensure all 4 classes are included).**

2. Data Cleaning & Preprocessing

- **Manual Preprocessing Required:**
    - ✓ **Clean the text by removing:**
        - **HTML tags**
        - **URLs**
        - **Emojis**
        - **Punctuation**

- Extra whitespace
- ✓ Convert all text to lowercase.
- ✓ Perform tokenization and remove stopwords.

- Vectorization (as applicable to chosen model):
  - ✓ For traditional ML: TF-IDF or CountVectorizer
  - ✓ For DL: Tokenizer with sequence padding
  - ✓ For Transformers: Pretrained tokenizer (e.g., from Hugging Face)

3. Exploratory Data Analysis (EDA)

- Visualize class distribution using bar charts.
- Generate word clouds for each category.
- Calculate and plot:
  - ✓ Average word count per article
  - ✓ Description/title length distribution
- Create a heatmap showing frequent words per class.

4. Model Building

You must build three types of models (sample choices shown):

[1] Machine Learning Model: (e.g., Logistic Regression, Naive Bayes)
  i. Feature input: Vectorized text (TF-IDF)

[2] Deep Learning Model :(e.g., LSTM, GRU)
  i. Preprocessed & padded sequences
  ii. Include: Dropout, tuning, and embedding layers

[3] Pretrained Transformer Model: (e.g., fine-tuned BERT, DistilBERT)
  i. Use Hugging Face transformers for fine-tuning
  ii. Use Trainer API or manual training loop

Note : Students are encouraged to explore and select appropriate architectures.

5. Model Evaluation & Justification

- **Evaluate each model using:**
  - ✓ **Accuracy**
  - ✓ **Precision**
  - ✓ **Recall**
  - ✓ **F1-score**
  - ✓ **Confusion Matrix**
- **Create a comparison table with your results and analysis:**

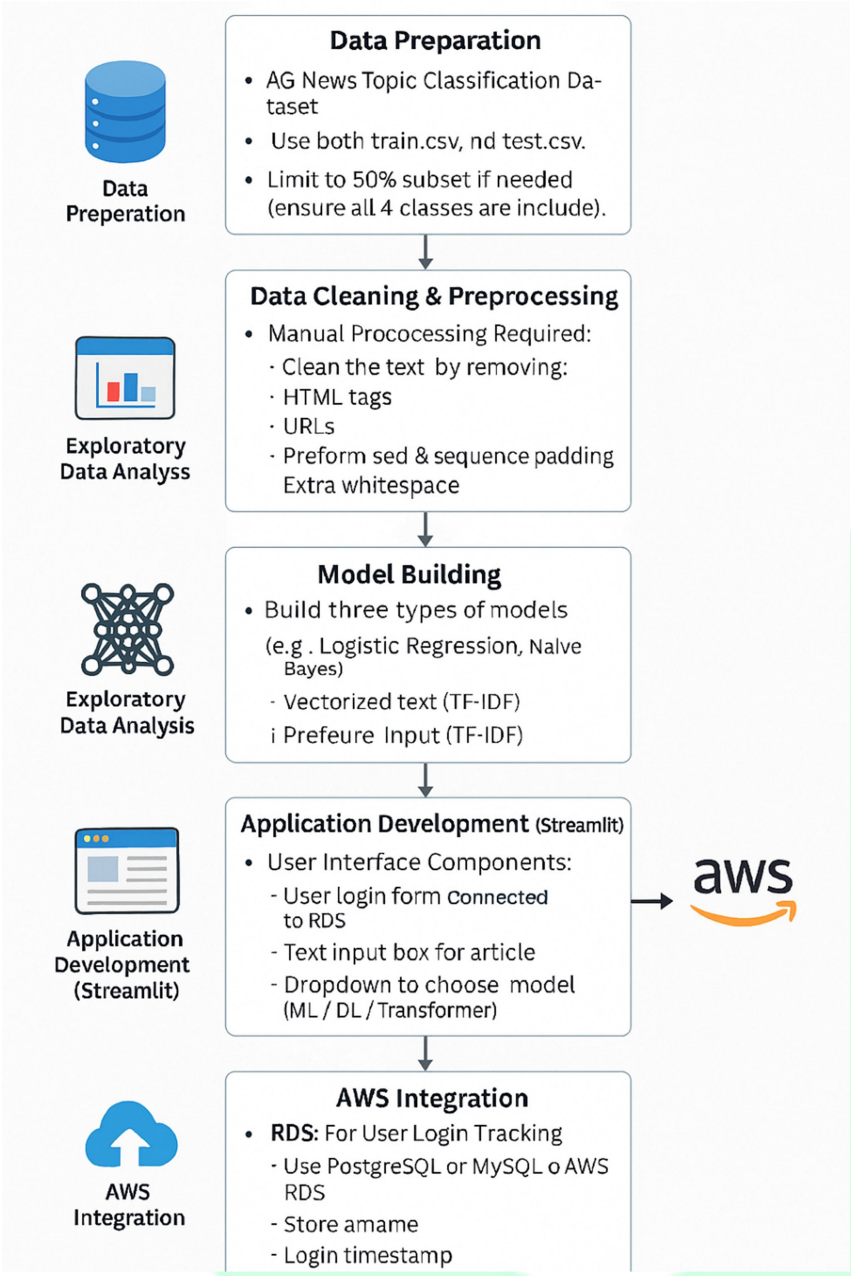
Model Type	Accuracy	F1-Score	Pros	Cons
Name of ML model				
Name of DL model				
Name of Pre-trained model				

6. Application Development (Streamlit)

- **User Interface Components:**
  - ✓ **User login form (connected to RDS)**
  - ✓ **Text input box for article**
  - ✓ **Dropdown to choose model (ML / DL / Transformer)**
  - ✓ **Display predicted category and confidence**

7. AWS Integration

- **RDS: For User Login Tracking**
  - ✓ **Use PostgreSQL or MySQL on AWS RDS**
  - ✓ **Store:**
    - **Username**
    - **Login timestamp**
    - **Selected model**
  - ✓ **Integration: Use SQLAlchemy or psycopg2 to connect with RDS from Streamlit (Learner choice)**
- **S3 :**
  - ✓ **Store model artifacts (.pkl, .pt, or .bin) in AWS S3**
  - ✓ **Load dynamically in Streamlit app**
- **EC2 (or Hugging Face Space) : Learner choice**
  - ✓ **Deploy Streamlit app on AWS EC2 (preferred for full control)**
  - ✓ **Optional: Compare deployment on Hugging Face Spaces**



Results:

- Fully functional web application for article classification.
- Hosted using AWS or Hugging Face Spaces with public access.
- Clean UI and logging capability.

Project Evaluation metrics:

- **Functionality:** Model should accurately classify articles into correct categories.
- **Performance:** Application should return predictions quickly and reliably.
- **Scalability:** Hugging Face Spaces should support multiple users concurrently.
- **Security:** If login or user tracking is implemented, it must be secure.
- **Usability:** Interface should be easy to use and visually clear.
- **Documentation:** Clear setup, usage, and model description included in the repository.

Technical Tags:

Text Classification, Hugging Face, Transformers, Streamlit, Gradio, Article Categorization, NLP, LLM, Python,AWS

Data Set:

https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset?resource=download

NOTE: If the dataset size exceeds your system's memory or computational capacity, you are permitted to use **50% of the dataset** for training and testing purposes. Ensure that the subset is representative and includes samples from all classes.

Data Set Explanation:

- The AG's news topic classification dataset is constructed by choosing 4 largest classes from the original corpus. Each class contains 30,000 training samples and 1,900 testing samples. The total number of training samples is 120,000 and testing 7,600.
- The file classes.txt contains a list of classes corresponding to each label.
- The files train.csv and test.csv contain all the training samples as comma-sparated values. There are 3 columns in them, corresponding to class index (1 to 4), title

and description. The title and description are escaped using double quotes ("), and any internal double quote is escaped by 2 double quotes (""). New lines are escaped by a backslash followed with an "n" character, that is "\n".

Project Deliverables:

- Source Code: Complete source code for the Streamlit application.
- Documentation: Detailed documentation including setup instructions, usage guide, and explanation of the architecture.
- Deployment Scripts: Scripts used for setting up the environment and deploying the application on Hugging Face services.
- Project Report: A report summarizing the project, approach taken, and results achieved.(optional)

Project Guidelines:

- Follow PEP 8 coding standards for Python
- Use Git for version control and push regularly to Hugging Face repo
- Ensure model loads efficiently and UI is intuitive
- Monitor Hugging Face resource limits (for CPU Spaces)
- If using databases, document schema and usage.

Timeline:

14 days from the date of document issuance.

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday-Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

For DE/BADM project/class topic doubt slot clarification session:

Booking link : https://forms.gle/NtkQ4UV9cBV7Ac3C8

Session timing:

For DE: 04:00 pm to 5:00 pm every saturday
For BADM 05:00 to 07:00 pm every saturday

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open only on Saturday (after 2 PM) and Sunday on Every Week

Timing:

For BADM and DE
Monday-Saturday (11:30AM to 1:00PM)

For DS and AIML
Monday-Saturday (05:30PM to 07:00PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA