

## AI Assignment - Task 1: Ticket Classifier & Entity Extractor

### Problem Statement

Creating a machine learning pipeline is important.

- Categories customer support tickets by the problem they have and how quickly they must be addressed
  - Extracts key entities (product, date, complaint keywords)
- Allows you to try predictions interactively with a built-in Gradio interface.

### Modules & Workflow

1. **Data Cleaning:** Lowercasing, punctuation removal, stopwords, lemmatization (NLTK)
2. **Feature Engineering:**
  - TF-IDF (text vectorization)

How long the tickets are

  - Sentiment score (TextBlob)
3. **Modeling:**
  - Category: Random Forest Classifier

The urgency level of this case is known as Random Forest Classifier.
4. **Entity Extraction:**
  - Product: matching tasks are set by rules.
  - Dates: ``dateparser``
  - Using specific complaints you prepare ahead of time
5. **Integration:**
  - ``process_ticket()`` returns prediction + extracted entities
6. **Gradio App:**
  - People can use single tickets or buy batches in the web application

### Model Evaluation

#### Issue Type Classification

- Accuracy is 100%
- F1-score: 1.00
- Confusion Matrix: [Displayed in notebook]

#### Urgency Level Classification

- Accuracy is 31%
- F1-score: 0.31
  - Confusion Matrix: [Displayed in notebook]

## 💡 Design Choices

We used traditional machine learning specifically for the purpose of clearer understanding.

- I put the basic entity rules in place, rather than using spaCy NER, to fit the needs of the assignment.
- A Gradio batch input option is now included as an extra for students

## 📁 How to Run

1. In Google Colab, press New Notebook to start working
2. Upload the Excel dataset ("dataset.xls")
3. Go ahead and run every cell in the code.
4. As soon as you finish, the Gradio interface will show up along with a public link.
5. Glue the stickers and watch to see what happens.

## 🎥 Demo Video

[ 📺 AI Assignment - Task 1.mkv ]

## ⚠️ Limitations

The extraction of products is handled based on rules and their case.

All models were built using basic ML requirements, as deep learning is not present.

## 🎥 Gradio App Link

<https://b3b02ba742c6838030.gradio.live/>

The screenshot shows a Gradio web interface titled "Batch Ticket Classifier". At the top, there is a terminal-like area with text: "colab notebook detected. To show errors in colab notebook, set debug=True in launch()", "\* Running on public URL: [https://b3b02ba742c6838030.gradio.live](\"https://b3b02ba742c6838030.gradio.live\")", and "This share link expires in 1 week. For free permanent hosting and GPU upgrades, run 'gradio deploy' from the terminal in the working directory to deploy to Hugging Face Spaces ([https://huggingface.co/spaces/](\"https://huggingface.co/spaces/\"))".

The main interface has a dark theme. On the left, there is a text input area labeled "ticket\_texts" with a placeholder "Paste multiple tickets...". Below this input area are two buttons: "Clear" and "Submit". On the right, there is an "output" box showing "{}". Below the output box is a "Flag" button.