Model Evaluation Report: Turkish Joke Generator using YTU GPT-2 Model

🎯 Project Goal

The aim of this project is to develop a Large Language Model (LLM) capable of generating Turkish jokes (fıkralar) based on different categories such as *Nasreddin Hoca*, *Temel*, *wife-husband jokes*, *student-teacher*, and *daily life humor*. As part of this goal, the publicly available **`ytu-ce-cosmos/turkish-gpt2`** model from Hugging Face was tested for its ability to generate relevant, grammatically correct, and humorous Turkish content.

🔍 Test Methodology

The evaluation consisted of:

- Supplying the model with **prompt sentences** based on typical joke starters.
- Measuring the quality of the output in terms of:
  - **Coherency and Meaning**
  - **Grammatical Structure**
  - **Humor/Entertainment Value**
  - **Category Relevance**
- Observing **variability and stability** in multiple generations.

Example Prompts Used:

- `"Temel bir gün camiye gitmiş..."`
- `"Nasreddin Hoca bir gün pazara gitmiş..."`
- `"Öğrenci öğretmenine sormuş..."`
- `"Adam bakkala gidip bir kilo domates istemiş..."`

❌ Observed Issues

Criteria	Observations
Semantic Coherence	Many outputs were **incoherent**, incomplete, or abruptly ended.
Grammatical Accuracy	Several outputs had **grammatical mistakes** or nonsensical phrasing.
Humor	Very few responses reflected **genuine humor** or classic joke structure.
Category Relevance	Outputs did **not align** with intended joke types (e.g., Nasreddin Hoca).
Control over Style/Theme	The model **lacked controllability** for targeted joke categories.

📉 Conclusion

The **`turkish-gpt2` model in its current form is not sufficient** for the needs of this project. While it can produce grammatically plausible Turkish sentences in some cases, it **fails to capture the humor**, structure, and category-specific tone that Turkish jokes typically follow.

🔧 Proposed Solution: Fine-Tuning

In order to improve the model's performance for Turkish joke generation, **fine-tuning on a domain-specific dataset** is required.

🔨 *What to Fine-Tune With:*

- A curated dataset of **categorized Turkish jokes**, ideally:

- ~5,000–10,000 high-quality entries
- Labeled by type: `nasreddin`, `temel`, `married_couple`, `student_teacher`, etc.
- Include full jokes with punchlines.

🔁 *How Fine-Tuning Will Help:*

Issue	Solution via Fine-Tuning
Lack of Humor	Model learns real joke patterns and punchlines.
No Category Awareness	Model learns to generate by category label or prefix.
Incoherent Output	Fine-tuning stabilizes generation and improves fluency.
Grammatical Errors	Exposure to correct sentence structures reduces mistakes.

🚀 Next Steps

1. **Data Collection & Cleaning**:
   a. Scrape or manually collect Turkish jokes.
   b. Normalize text, remove duplicates, categorize jokes.
2. **Prepare Fine-Tuning Dataset**:
   a. Format in Hugging Face `text` or `JSONL` format.
   b. Optionally use a prefix format like:

```
text
KopyalaDüzenle
<category>: Nasreddin
<joke>: Nasreddin Hoca bir gün göle yoğurt çalmaya gitmiş...
```

3. **Fine-Tune the Model**:

a. Use Hugging Face Transformers & Trainer API.
b. Recommended training on Google Colab or local GPU.

4. **Evaluate & Deploy Improved Model**:
    a. Re-test using the same prompts.
    b. Measure humor, fluency, and category alignment.
    c. Optionally deploy with a web or chatbot interface.

🗂 Files & Resources

- Model tested: `ytu-ce-cosmos/turkish-gpt2`
- Code: See `notebooks/model_evaluation.ipynb`
- Dataset: (To be created by team)
- Output Samples: See `outputs/test_outputs.txt`

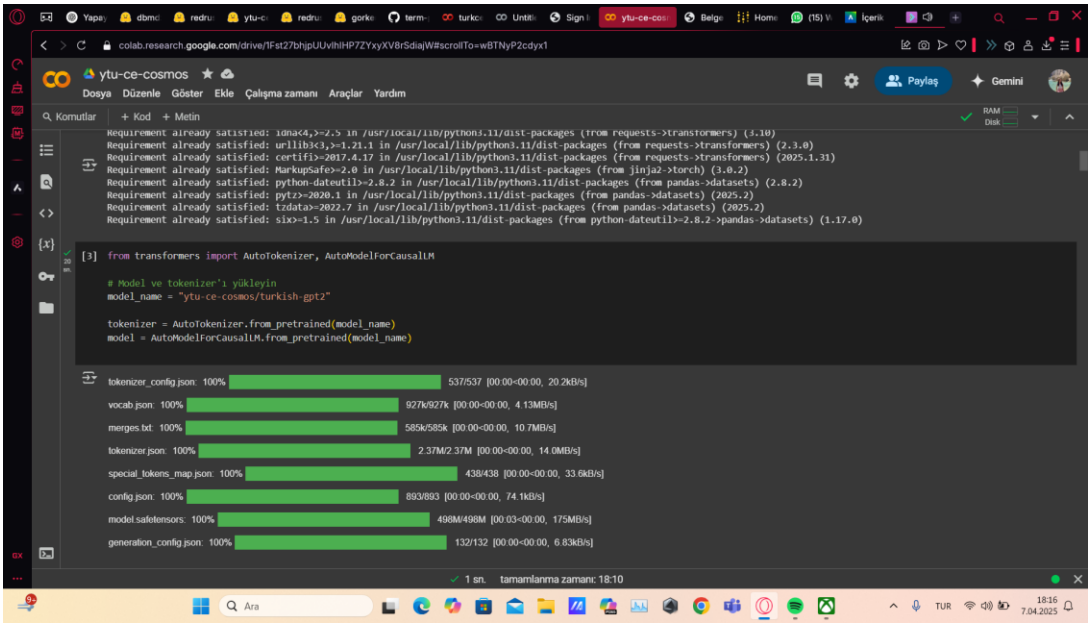
📌 Final Verdict

🔴 **The base GPT-2 Turkish model is not sufficient** for generating meaningful and funny Turkish jokes out of the box.

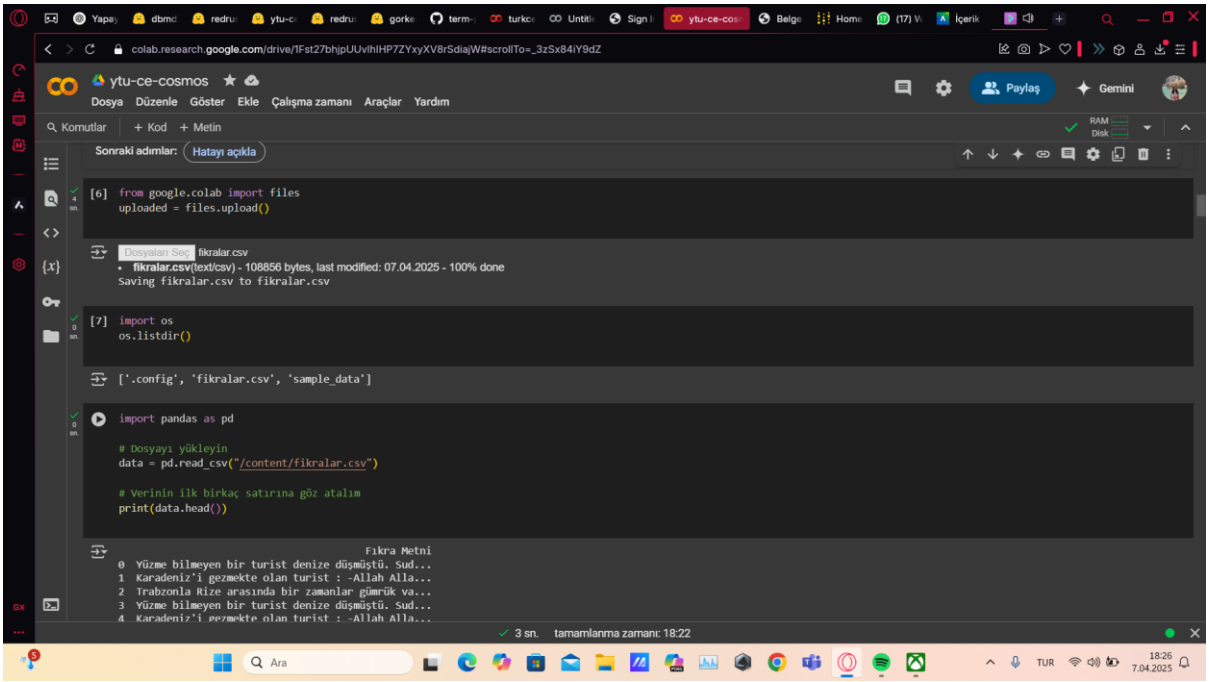
🟢 **Fine-tuning with a curated dataset is essential** to meet the project requirements for category-based joke generation and improve humor quality.

STEPS OF THE TESTİNG :

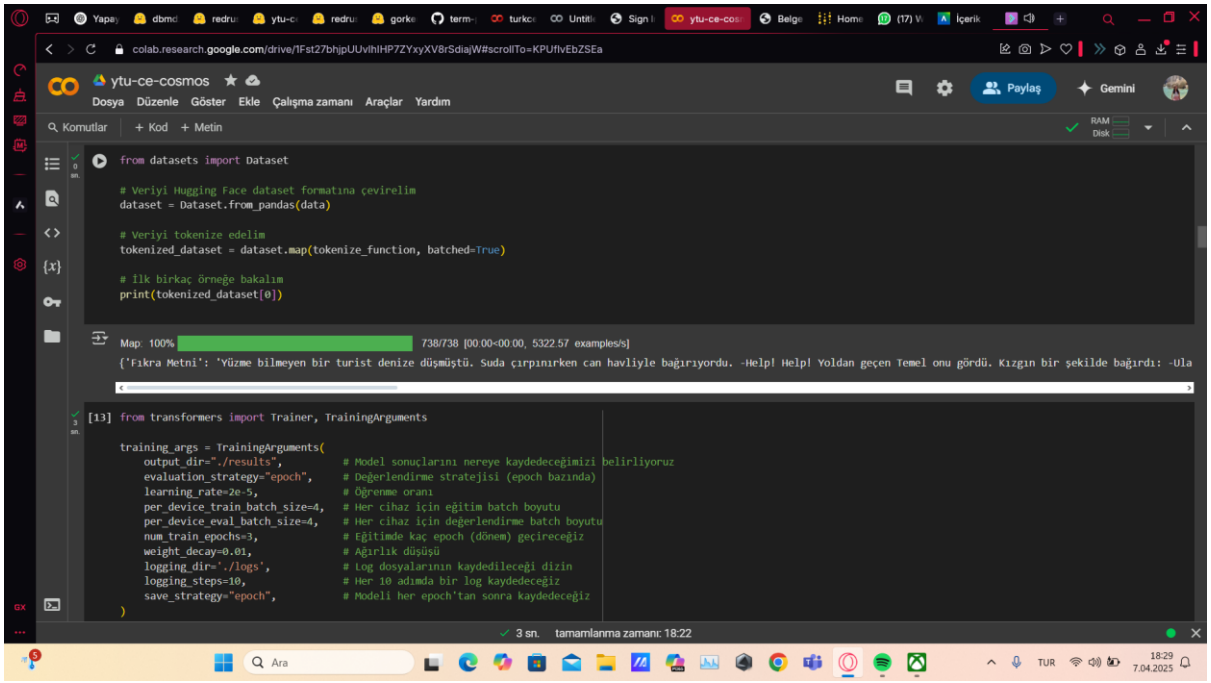
1. Here’s an explanation of the code for loading the **YTU-CE-COSMOS** model:



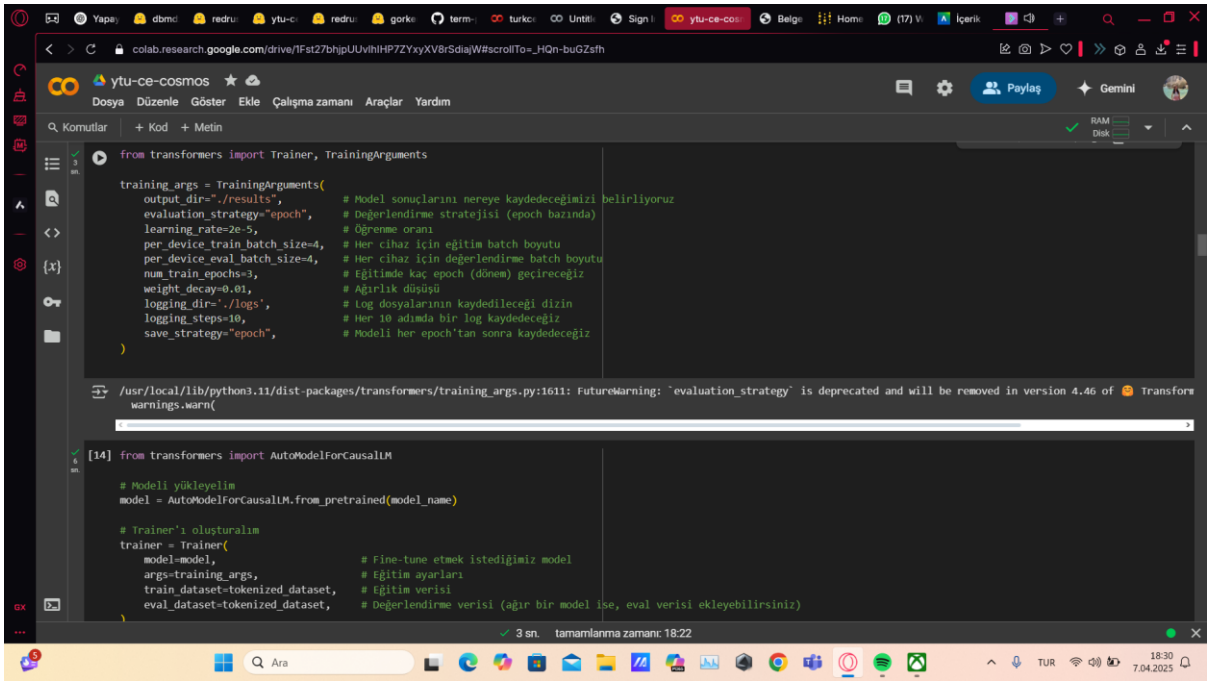
2. This code snippet reads a CSV file using the **pandas** library and displays the first few rows of the data.



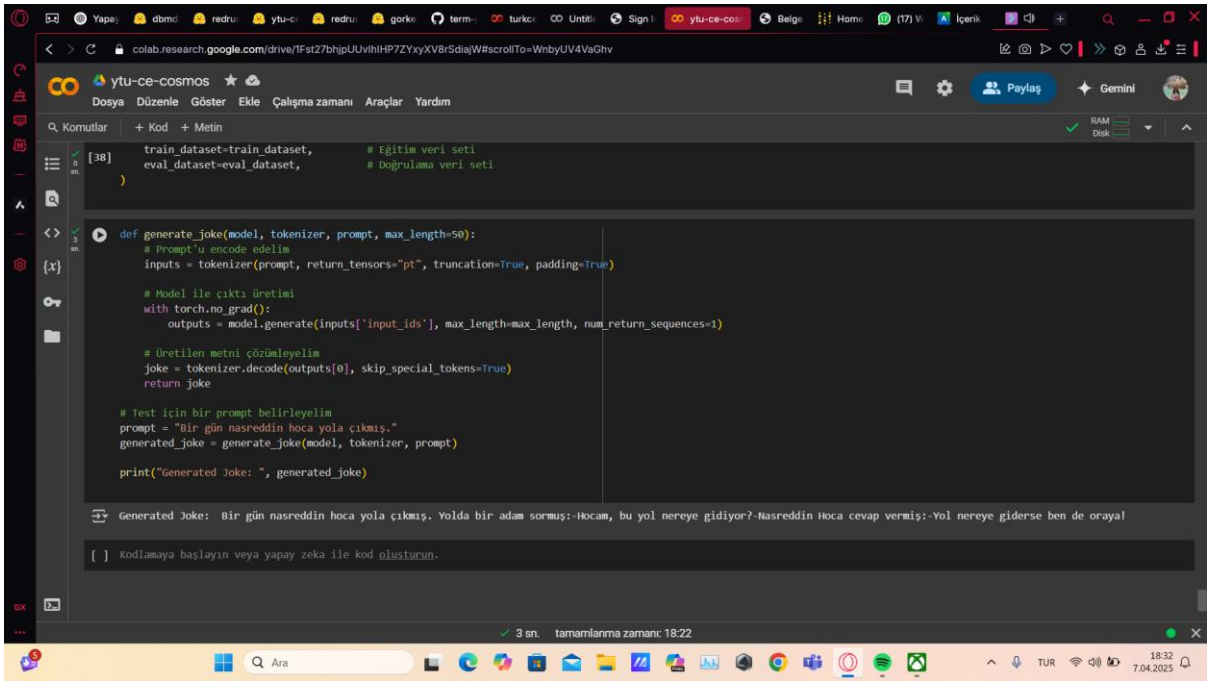
3. This code snippet converts your pandas DataFrame into a Hugging Face dataset format and tokenizes it.



4. This block of code is setting up the **training arguments** for your model training.



5. This code defines a function to generate jokes using the trained model



2)BERTürk

BERTürk (a BERT-based Turkish model) is a powerful model for Turkish language processing tasks. BERT, which stands for "Bidirectional Encoder Representations from Transformers," is particularly strong in tasks such as text classification, context understanding, and question-answering systems. However, it may have limitations in creative text generation tasks (e.g., writing jokes).

Advantages of BERTürk

1. **Turkish Language-Specific Training**:
    a. **BERTürk** is pre-trained specifically for the Turkish language, meaning it is trained with the linguistic structures, grammar, and vocabulary of Turkish. This allows the model to accurately process and understand Turkish texts.
2. **Context Understanding**:
    a. BERT is a bidirectional model, meaning it understands the context from both directions (left and right) of the text. This

enables the model to provide more meaningful outputs and better understanding of language nuances.

3. **Text Understanding and Classification**:
    a. **BERTürk** excels in tasks such as text classification, sentiment analysis, and named entity recognition. If your task involves categorizing or classifying jokes or similar tasks, this model would perform very well.
4. **Transfer Learning**:
    a. BERT-based models are great for **transfer learning**, meaning you can fine-tune the pre-trained model on your specific task, such as classifying or understanding jokes. This results in high performance with relatively less data.

Disadvantages of BERTürk

1. **Limited Performance in Creative Text Generation**:
    a. **BERT** models generally do not perform as well in "generative" tasks, such as generating new, creative content. BERT is designed for tasks like text understanding and classification, not text generation.
    b. For tasks like **joke writing**, models such as **GPT-2** are far more suitable because they are optimized for text generation. BERT, on the other hand, excels in tasks that require understanding and processing existing text, rather than creating new content from scratch.
2. **Limited Creativity**:
    a. While **BERTürk** excels in understanding the structure and meaning of the language, it may struggle with more creative tasks like writing jokes, which require an understanding of humor, creativity, and often unexpected connections. **GPT-2** is better suited for such tasks because of its autoregressive architecture designed for text generation.

3. **Model Structure**:
    a. BERT is typically more suited for tasks like text classification, named entity recognition, and question-answering. It is not as effective in generating creative, coherent, and humorous text, which is the case with **GPT-2** or **GPT-3**.
4. **Memory and Computational Resources**:
    a. BERT-based models are large and require significant computational resources. For very large datasets or when working with high-performance requirements, BERT may not be as efficient as other models, such as **GPT** models, in certain contexts.

When BERTürk is Ideal

- **Text Classification** (e.g., categorizing jokes).
- **Sentiment Analysis** (e.g., analyzing whether a joke is funny or not).
- **Question-Answering** (e.g., answering questions based on jokes).
- **Named Entity Recognition** (e.g., identifying characters in jokes).

Conclusion

BERTürk is a powerful model for text understanding and classification tasks. If your goal is to **categorize jokes** or **analyze the structure of jokes**, then BERTürk is a very suitable choice. However, for tasks that involve **joke writing** or creative content generation, models like **GPT-2** or **GPT-3** would perform better.

Summary Recommendations:

- **For Joke Writing**: Models like **GPT-2 Turkish** are recommended because they are designed for generative tasks.
- **For Joke Classification**: **BERTürk** would be a great choice for categorizing or classifying jokes.

1. BERTürk vs YTU-CE-Cosmos (GPT-2)

BERTürk:

- **Model Type:** BERT-based (Transformer encoder model).
- **Training Purpose:** BERT and its derivatives are typically trained using "masked language modeling" (MLM), meaning they are more suitable for tasks like text understanding, classification, and prediction rather than text generation.
- **Application Area:** BERT models are primarily used for **text classification**, **question answering**, and **named entity recognition (NER)** tasks.
- **Advantages:**
    - Excellent performance on tasks involving **text understanding** and **classification**.
    - Bidirectional transformer structure allows the model to consider both the past and future context of the text, resulting in meaningful contextual understanding.
- **Disadvantages:**
    - BERT-based models are **limited for text generation** tasks (such as writing jokes or creative text). BERT is designed to understand and classify text, not generate it, which limits its use for creative writing tasks.

YTU-CE-Cosmos (GPT-2):

- **Model Type:** GPT-2-based (Transformer decoder model).

- **Training Purpose:** GPT-2 is trained using "causal language modeling," which makes it highly suitable for **text generation** tasks. The model takes a starting prompt and generates coherent and contextually relevant text following it.
- **Application Area:** GPT-2 excels in tasks like **story generation**, **chatbots**, **text completion**, and **creative writing** (such as generating jokes).
- **Advantages:**
    - Highly effective for **text generation**, **creative writing**, **joke writing**, and similar tasks.
    - Takes a simple input prompt and generates meaningful, creative, and coherent text.
    - Flexible and adaptable for fine-tuning on various topics and producing content in different styles.
- **Disadvantages:**
    - Requires more computational resources and training data due to the large number of parameters.
    - Fine-tuning might require more experimentation and tuning to achieve optimal results.

Why GPT-2 (YTU-CE-Cosmos) is More Suitable for Your Project?

1. **Creative Text Generation:** Your project is focused on **joke writing**, which is a creative text generation task. **GPT-2** is designed for this purpose and has strong capabilities in generating creative and contextually coherent texts. On the other hand, **BERT** models are better at understanding and classifying text, making them less suitable for generating creative content like jokes.
2. **Training Objective:** GPT-2 is directly trained for **text generation**, meaning it is highly optimized to take a prompt and

generate meaningful, creative text following it. **BERT**, however, is designed to understand text, not to generate it, making it less effective for tasks like writing jokes.

3. **Flexibility: GPT-2** is highly flexible and capable of generating text across a wide range of topics. Given the diverse nature of joke styles and types, fine-tuning GPT-2 on your joke dataset will allow the model to produce high-quality jokes in different styles and formats. BERT models, being designed for classification tasks, don't provide the same level of flexibility for generating diverse types of creative content.
4. **Large-scale Training Data:** GPT-2 has been trained on vast datasets, which enables it to generate meaningful text across many domains. This is especially beneficial for your project, as it involves generating jokes in various formats and contexts, where GPT-2's broad training helps in producing natural, contextually relevant, and humorous content.
5. **Humor and Joke Generation:** GPT-2's capabilities in **natural language generation (NLG)** make it an ideal choice for generating jokes. The model can learn various comedic structures and incorporate humor into the text, making it perfect for tasks like joke generation, where creativity and context play a significant role.

Conclusion:

Given that your project focuses on **creative text generation** and **joke writing**, **GPT-2** (YTU-CE-Cosmos) is a more suitable model compared to **BERTürk**. GPT-2's strong capabilities in **text generation**, **flexibility**, and the ability to produce **creative content** make it an ideal choice for your needs, while BERT is better suited for text understanding and classification tasks, which are less relevant for your specific use case.

Seyda ARAT -200706036