

<b>Name-Akshaya Lakshmi Narasimhan</b>	<b>SRN- PES2UG23CS046</b>	<b>SECTION-A</b>
--	-------------------------------	------------------

## **GenAI**

### **Cell 1: Installation of Required Libraries**

The necessary libraries are installed to enable the task of text summarization.

It is essential, as the entire workflow depends on these tools being available.

Libraries such as Transformers and PyTorch provide the foundation for working with advanced language models.

## **Cell 2: Suppression of Warning Messages**

This cell focuses on reducing unnecessary distractions by suppressing warning messages.

While warnings are important during development, excessive messages can clutter meaningful output.

By filtering them out, the notebook allows the reader to focus on results rather than noise.

This decision reflects an intention to maintain clarity and direction.

It is a small but deliberate step toward a cleaner workflow.

## **Cell 3: Importing the Summarization Pipeline**

This cell introduces the summarization pipeline from the Transformers library.

Rather than building a model from scratch, this approach relies on a well-tested abstraction. It simplifies complexity while still allowing powerful functionality.

The pipeline serves as a bridge between raw text and meaningful summaries.

This step signals a shift from preparation to execution.

## **Cell 4: Selection of Computational Device**

We have to determine to use a cpu or a gpu

In this case I have executed it using a gpu for training efficiently

## **Cell 5: Loading the Pretrained Summarization Model**

This cell loads a pretrained DistilBART model designed specifically for summarization.

The model has already learned language patterns from large datasets.

Instead of building the model from scratch ,this approach builds upon existing knowledge.

From this moment, the system is ready to generate insight from text.

## **Cell 6: Initial Model Testing**

Here we test the model using a short paragraph. The purpose is not depth, but confirmation that the system functions correctly.

It ensures that the pipeline produces meaningful output before handling larger inputs.

This step reflects caution and responsibility in experimentation.

Only after this verification does the notebook move forward.

## **Cell 7: Defining a Technology News Article**

The text provides sufficient complexity for the summarization model to demonstrate its capabilities.

It mirrors real-world scenarios where information is dense and extensive.

By using realistic content, the experiment gains relevance.

This marks the transition from testing to application.

## **Cell 8: Generating a Standard Summary**

This cell generates a summary using balanced length constraints.

The goal is to capture key ideas without overwhelming the reader.

Word limits help maintain structure and readability.

The output demonstrates how effectively the model condenses information.

It represents the default, expected behavior of the system.

## **Cell 9: Generating a Short Summary**

This cell explores the effect of stricter length constraints.

By limiting the number of words, the model is forced to prioritize only the most important ideas. The result is concise, direct, and minimal.

This approach suits scenarios where attention is limited.

It highlights the flexibility of the summarization process.

## **Cell 10: Generating a Detailed Summary**

This cell allows the model more freedom by increasing summary length.

The output retains more context and nuance from the original text.

Such summaries are useful when deeper understanding is required.

This comparison shows that there is no single correct summary.

The level of detail depends on purpose.

## **Cell 11: Defining a Science News Article**

We introduce a science-focused article to test topic diversity.

The subject matter differs significantly from technology.

By changing domains, the robustness of the model can be observed.

It challenges the system to adapt rather than repeat patterns.

This step strengthens the evaluation.

## **Cell 12: Summarizing the Science Article**

This cell applies the summarization model to the science article.  
It follows the same process used for the technology text.  
Consistency in output indicates reliable model behavior.  
The summary reflects the model's ability to generalize.  
This reinforces confidence in its applicability.

## **Cell 13: Defining a Summary Comparison Function**

We introduce a function to generate summaries of varying lengths.  
Encapsulating this logic improves clarity and reusability.  
It allows structured comparison without repetitive code.  
This step reflects thoughtful design rather than improvisation.  
The function encourages systematic exploration.

## **Cell 14: Applying the Comparison Function**

We apply the comparison function to an article.  
Multiple summaries are generated and observed together.

The differences reveal how length influences information retention.

This comparison encourages analytical thinking.  
It supports informed decision-making in real applications.

## **Cell 15: Placeholder for User-Provided Content**

We create space for user-defined input.  
It invites exploration beyond predefined examples.  
The notebook becomes adaptable rather than fixed.  
This openness reflects real-world usage.  
It allows the user to step outside the script.



## **Cell 16: Summarizing User-Provided Article**

This cell processes the custom article entered by the user.

The summary demonstrates practical application of the system.

It moves from demonstration to personal use.

This step shows that the model is not limited to examples.

It responds to individual intent.

## **Cell 17: Batch Summarization of Articles**

This cell processes multiple articles together.

Batch processing improves efficiency and scalability.

It reflects real-world scenarios like news aggregation.

Handling multiple inputs shows system maturity.

The model moves beyond isolated tasks.

## **Cell 18: Performance Measurement**

This cell measures the time taken for summarization tasks.

Performance evaluation is essential for practical deployment.

It provides insight into speed and feasibility.

Efficiency matters when systems scale.

This step closes the workflow with reflection.