

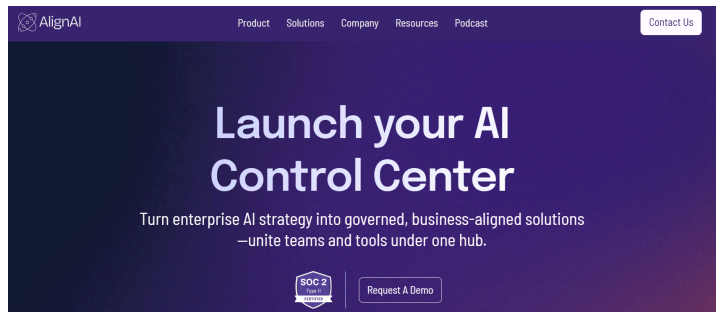
# **Sentiment Analysis Workshop**

## **Customer Feedback Classification**

**Using GenAI**

# About Me

- **Current Role:** Learning Experience Manager at [AlignAI](#)



- **Experience:** 15+ years in analytics
- **Passions:** Artificial Intelligence (AI), Governance, and Instructional Design
- **Memberships & Volunteer Work:** WIA and MORPC
- **Education:** Doctoral student at Franklin University



**Kristy Wedel**

Learning Experience Manager |  
Transforming Data Governance & An...



# Workshop Goals

- Explore AI for sentiment analysis
- Hands-on approach using **Cursor**
- Build, train, and evaluate a sentiment analysis model
- Gain practical skills to adapt and refine models

# Requirements

**Follow Sentiment Analysis Workshop Quick Start.docx Instructions**

## **Install Conda**

[Miniconda](#) (lightweight)

OR

[Anaconda](#) (comprehensive)

## **Install Cursor**

[Download Cursor](#)

# Files



<https://bit.ly/3PIK4RU>

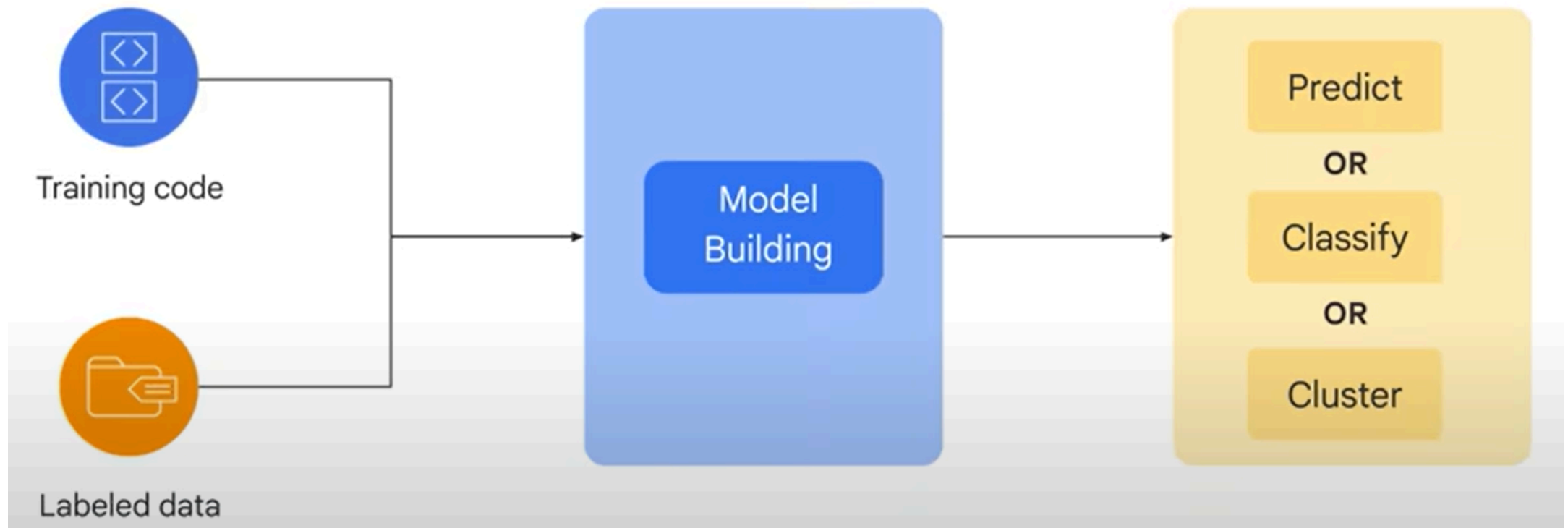
# Introduction to Sentiment Analysis

- NLP technique to classify text into:
  - Positive, Negative, Neutral
- Applications:
  - Customer feedback analysis
  - Social media monitoring
  - Prioritizing support tickets



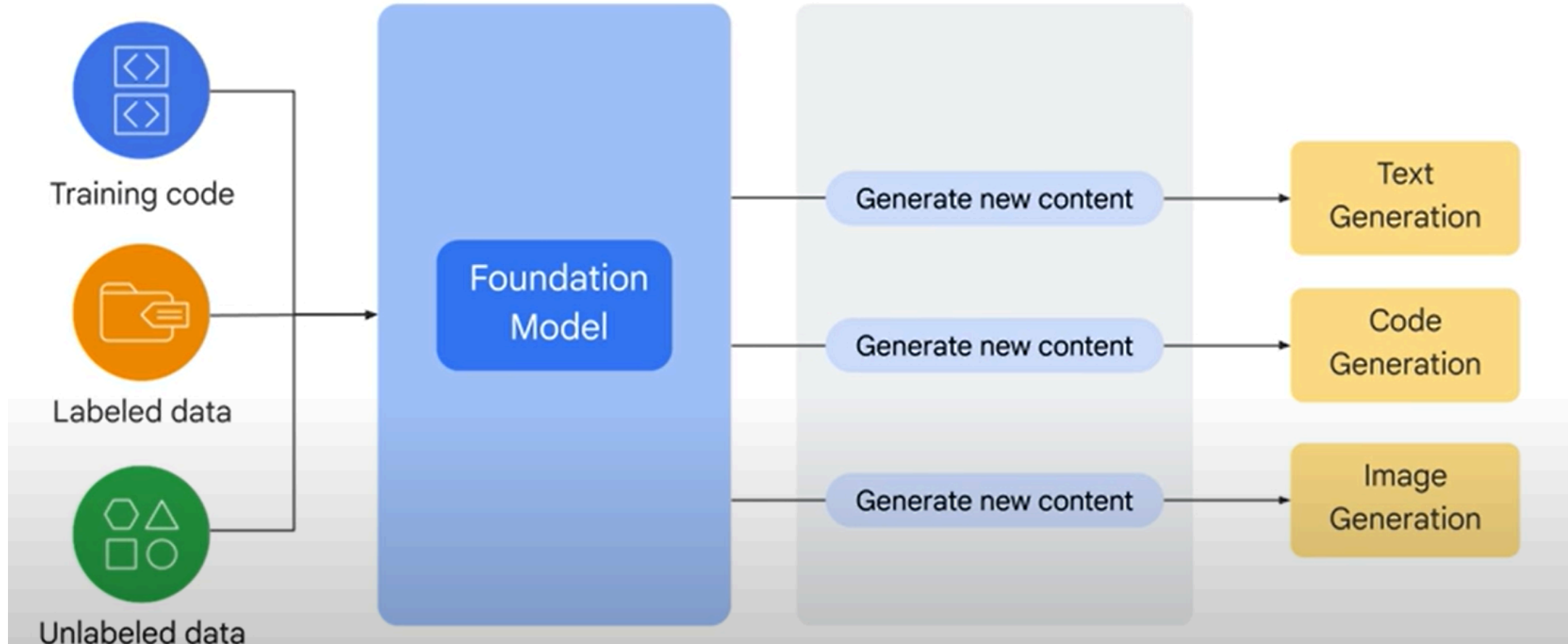
# Traditional ML

Classical Supervised & Unsupervised Learning



# Gen AI

## Gen AI Supervised, Semi-Supervised & Unsupervised Learning



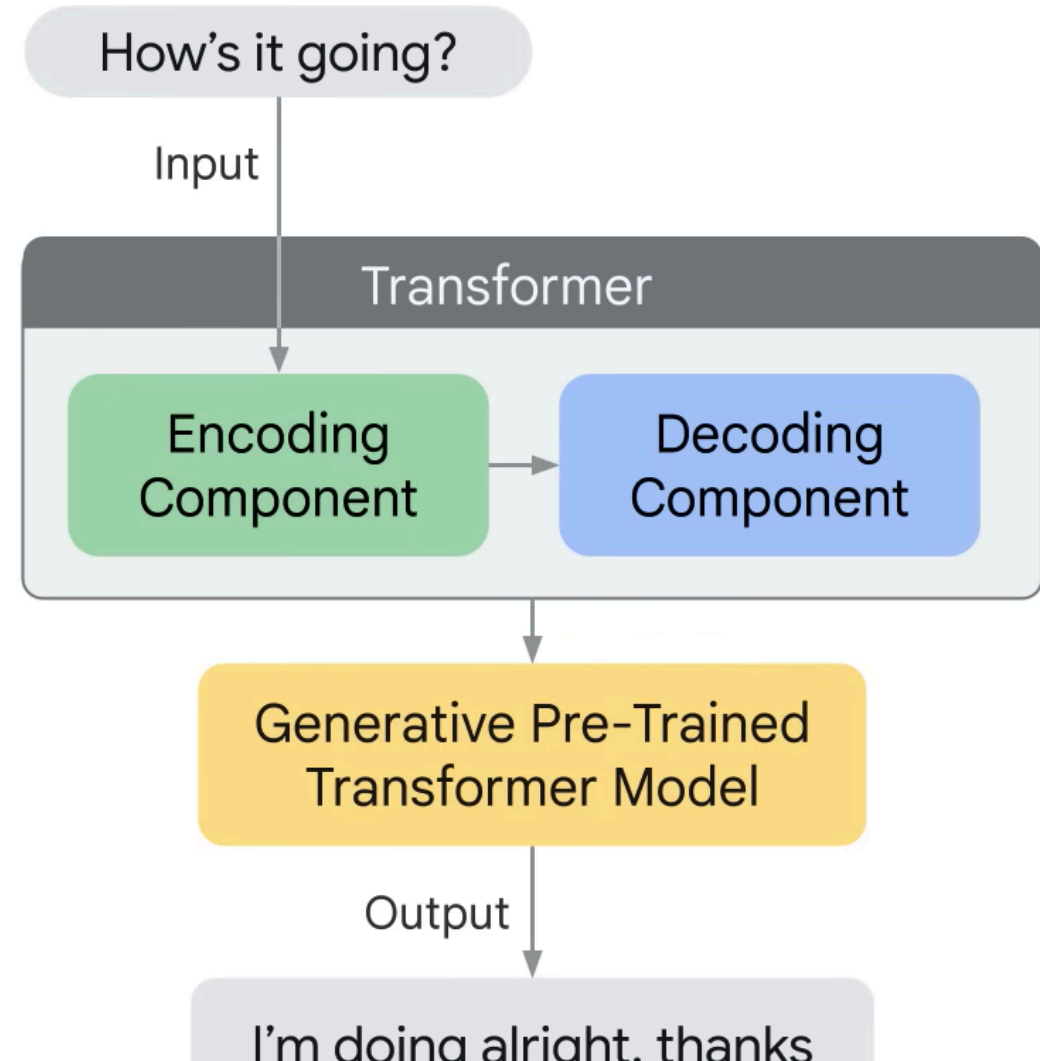


# How Gen AI Works

## How it Works

### Pre-Training:

- Large amount of Data
- Billions of parameters
- Unsupervised learning

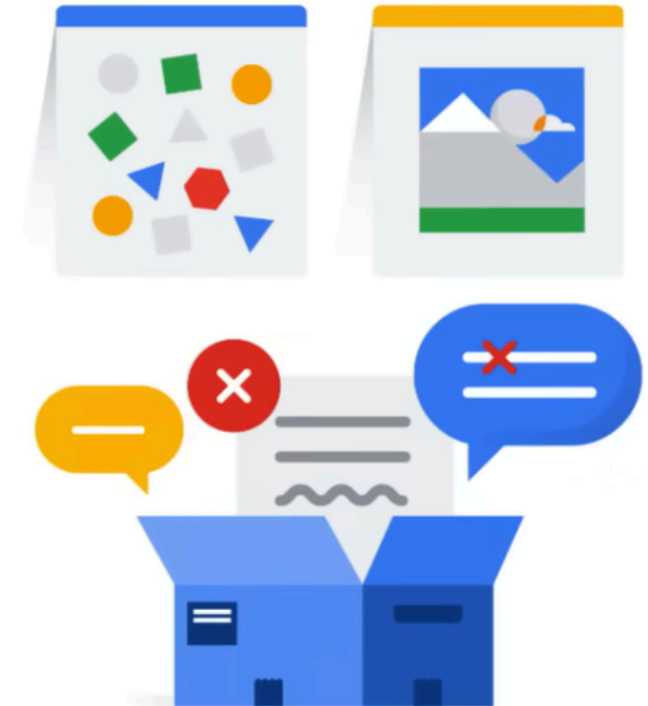


# Hallucinations

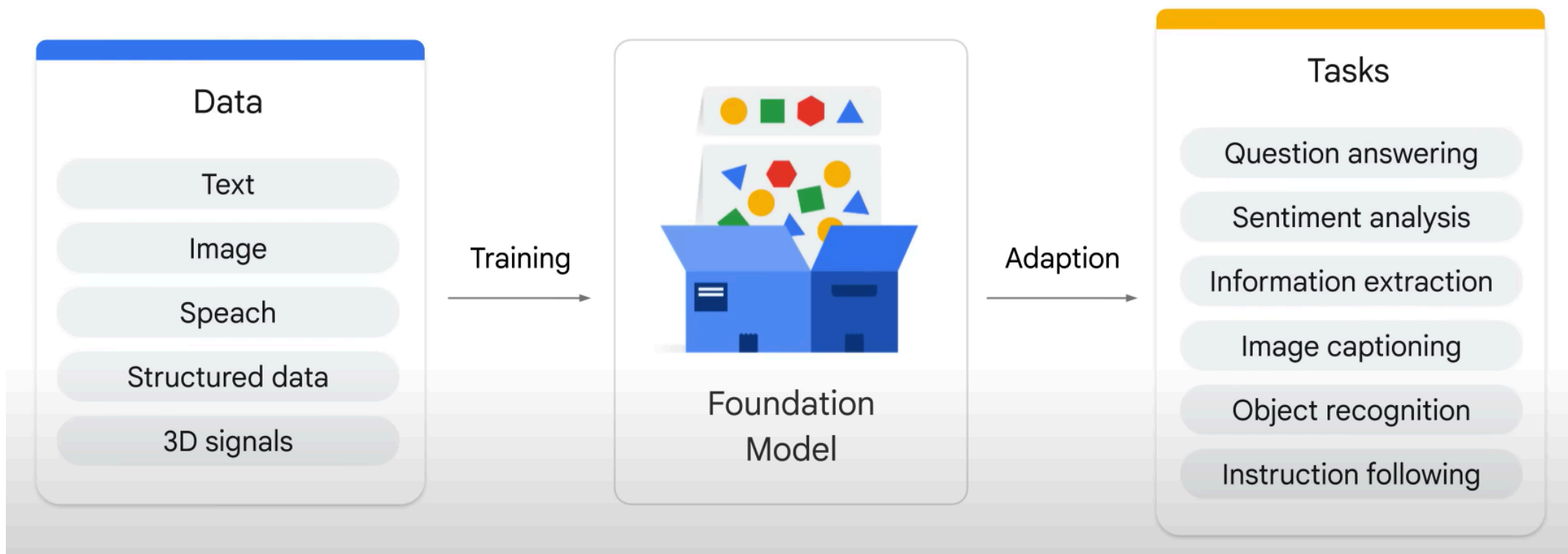
## Hallucinations

### Challenges

- ❗ The model is not trained on enough data
- ❗ The model is trained on noisy or dirty data
- ❗ The model is not given enough context
- ❗ The model is not given enough constraints



# Customization

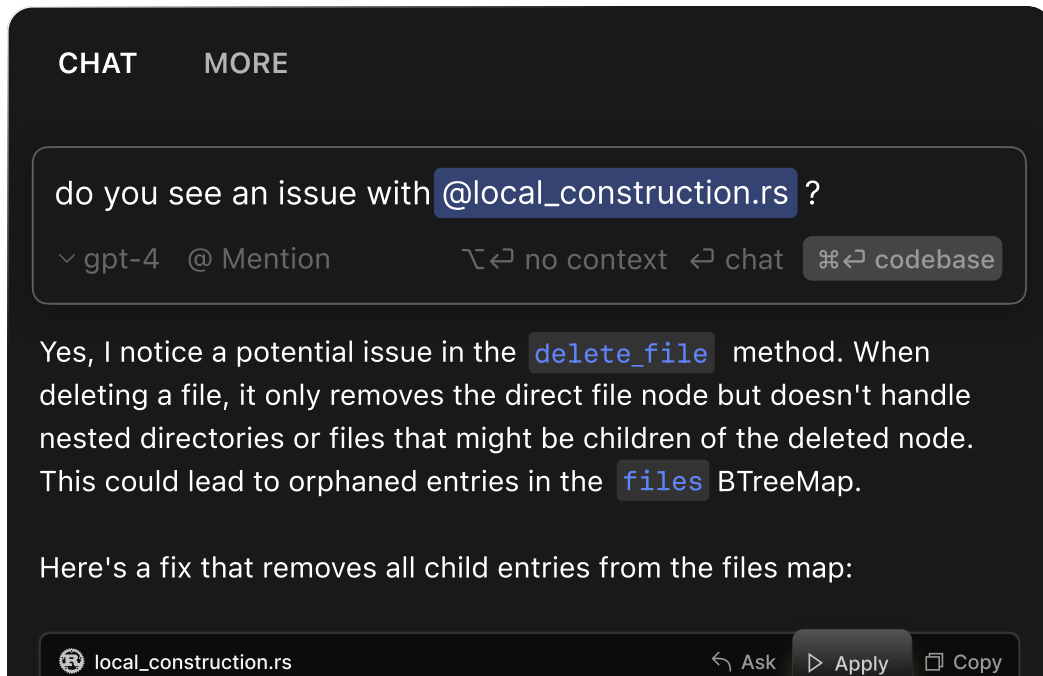


# What is Cursor?

- AI-powered code editor designed to boost developer productivity.
- Combines a clean interface with advanced AI capabilities.

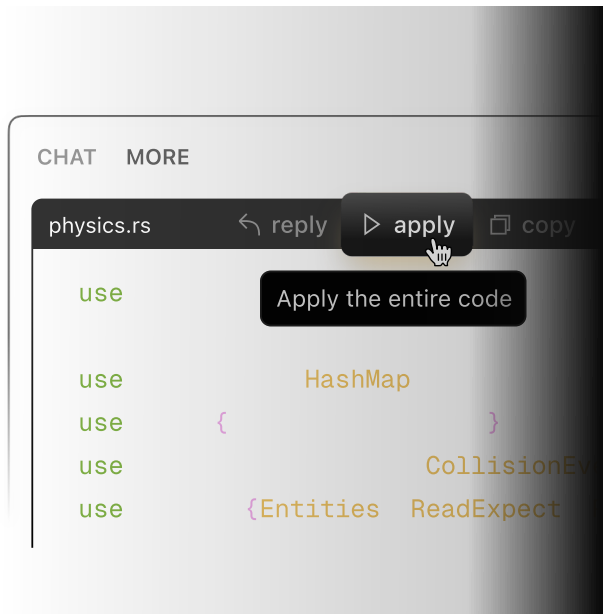
# Cursor: Chat

- Chat - Chat lets you talk with an AI that sees your codebase. The chat can always see your current file and cursor, so you can ask it things like: "Is there a bug here?". You can add particular blocks of code to the context with Ctrl+Shift+L or "@" You can chat with your entire codebase with Ctrl+Enter.



# Cursor: Instant Apply

- Instant Apply - Apply the code suggestions from chat back into your codebase by clicking the play button on top of any chat codeblock.



# Cursor Benefits

- Saves time by reducing repetitive tasks.
- Enhances code quality with AI recommendations.
- Simplifies onboarding for new developers.
- Improves collaboration with integrated tools.

# Workflow Steps

**Step 1:** Data Cleaning and Preprocessing

**Step 2:** Model Training and Evaluation

**Step 3:** Deployment and Prediction

**Step 4:** Real-Time Sentiment Predictions



# Data Overview

- **Dataset Details:**

- 1,000 customer feedback entries.
- Columns: Review\_ID , Review\_Text , Sentiment .
- Labels: Positive, Neutral, Negative.

# Google Colab

<https://tinyurl.com/3jccy5yc>

# Workflow Steps

**Step 1:** Data Cleaning and Preprocessing

**Step 2:** Model Training and Evaluation

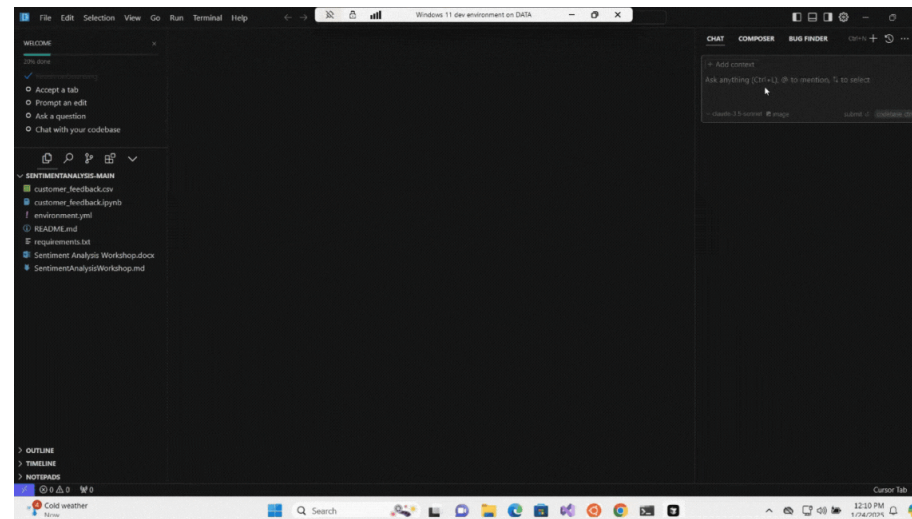
**Step 3:** Deployment and Prediction

# Load Data

## Prompt Example:

Write Python code to:

- Create a python jupyter notebook to load customer\_feedback.csv



# Data Preprocessing

- Data preprocessing is a critical step in data analysis and machine learning.
- It ensures the raw data is clean, structured, and usable for analysis.

## Key Steps

- **Data Cleaning:** Handle missing values, remove duplicates, and correct inconsistencies.
- **Data Transformation:** Normalize, encode variables, and create new features.
- **Data Integration:** Merge data from multiple sources, resolve schema mismatches.
- **Data Reduction:** Reduce dataset size while preserving important information.

# Cleaning for Sentiment Analysis

- **Text Cleaning:**
  - Remove special characters and stop words and convert to lowercase.
- **Tokenization:**
  - Split `Review_Text` into words for analysis.
- **Lemmatization:**
  - Reduce words to their base forms (e.g., "running" → "run").
- **Handling Missing Data:**
  - Drop or impute missing entries.
- **Deduplication:**
  - Remove duplicate reviews to avoid bias.

# Data Cleaning

## Prompt Example:

Write Python code to replace the existing data:

- Remove duplicate rows.
- Drop rows where "Review\_Text" or "Sentiment" columns have missing values.
- Convert all text in "Review\_Text" to lowercase.
- Remove punctuation from "Review\_Text."
- Remove common stop words like "the," "is," and "and."
- Apply lemmatization to "Review\_Text."

Then, save the dataset as a new csv and visualize the distribution by sentiment with a simple chart.

The screenshot shows a Jupyter Notebook titled "SentimentAnalysis.ipynb" in a Windows 11 environment. The notebook contains the following Python code:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Read the CSV file
df = pd.read_csv('customer_feedback.csv')

# Display the first few rows of the dataset
print("First 5 rows of the dataset:")
display(df.head())

# Get basic information about the dataset
print("Dataset Information:")
display(df.info())

# Display basic statistics of numerical columns
print("Basic statistics:")
display(df.describe())

# Check for missing values
print("Missing values:")
display(df.isnull().sum())
```

The output of the code shows the first 5 rows of the dataset:

Review_ID	Review_Text	Sentiment
1	Excellent service and fast delivery. Highly recommended.	Positive
2	Terrible experience, the product broke after a week.	Negative
3	Neutral feeling about this product, it's just average.	Neutral
4	This is the best purchase I've made this year!	Positive
5	The product is okay, but nothing special.	Neutral

On the right side of the notebook, there is a section titled "The notebook will" with a list of tasks:

1. Import commonly used data analysis libraries
2. Load your CSV file into a pandas DataFrame
3. Show the first 5 rows of data
4. Display information about the dataset (columns, data types, etc.)
5. Show basic statistics for numerical columns
6. Check for any missing values in the dataset

Below the list, there is a section titled "To use this notebook:" with instructions:

1. Make sure your "customer\_feedback.csv" file is in the same directory as the notebook.
2. Run each cell sequentially to see the output.
3. Adapt the code based on your specific needs and the structure of your CSV file.

At the bottom, there is a prompt: "Would you like me to add any specific analysis or visualization for your customer feedback data?"

# Workflow Steps

**Step 1:** Data Cleaning and Preprocessing

**Step 2:** Model Training and Evaluation

**Step 3:** Deployment and Prediction



# Generative AI Key Terms

- **Tokens:** The smallest units of text, such as words or subwords, used in AI processing.
- **Encoding:** The process of converting text into numerical representations for machine learning models.
- **Transformer:** A neural network architecture that uses self-attention to process sequences of data.
- **Loss:** A metric that measures the difference between predicted and actual outputs, guiding model training.
- **Epochs:** Complete passes through the entire training dataset during model training.

# Overview of Pretrained Models

## `distilbert-base-uncased-finetuned-sst-2-english`

- **Base Model:** DistilBERT, a lighter, faster version of BERT.
- **Fine-Tuned Task:** Sentiment analysis using the SST-2 dataset.
- **Performance:** Delivers accurate binary sentiment predictions (positive/negative).
- **Use Cases:** Ideal for analyzing customer feedback, social media, and reviews.

# Why Use Pretrained Models?

- Minimize training time and computational resources.
- Achieve state-of-the-art results with minimal fine-tuning.
- Versatile across various NLP tasks.

# Other Pretrained Models

## 1. **BERT:**

- Bidirectional context understanding for tasks like Q&A and classification.
- Popular variants: BERT-Base, BERT-Large.

## 2. **GPT:**

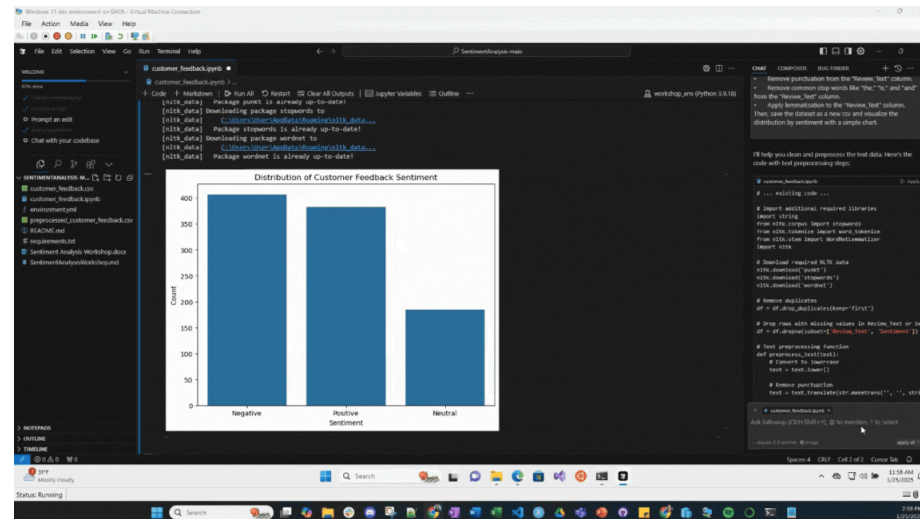
- Autoregressive model for text generation and conversational AI.

## 3. **DistilBERT:**

- Lightweight, fast, and efficient for edge devices.

## Prompt Example:

- Use Hugging Face Transformers and tensorflow to train a sentiment classification model:
- Model: distilbert-base-uncased-finetuned-sst-2-english.
  - Group Positive and Neutral sentiments together.
  - Provide metrics to evaluate: Accuracy, Precision, Recall, F1-score.



# **Overfitting vs. Underfitting in Machine Learning Models**

# What is Overfitting?

- **Definition:** When a model performs well on training data but poorly on unseen data.
- **Indicators:**
  - High training accuracy, low validation/test accuracy.
  - Large gap between training and validation loss.
- **Causes:**
  - Model memorizes training data instead of generalizing patterns.
  - Excessive model complexity (e.g., too many layers or parameters).

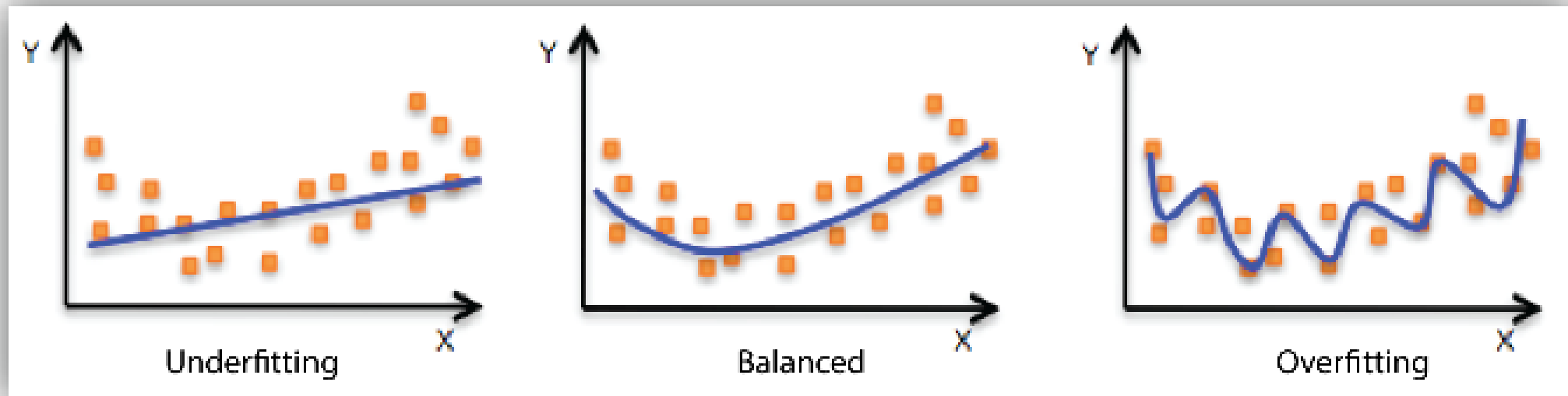
# What is Underfitting?

- **Definition:** When a model performs poorly on both training and unseen data.
- **Indicators:**
  - Low accuracy on training data.
  - Training and validation loss remain high.
- **Causes:**
  - Model fails to capture underlying patterns in the data.
  - Model is too simple (e.g., insufficient layers or parameters).



# Key Comparison

Overfitting	Underfitting
Memorizes training data	Fails to learn patterns
High training accuracy, low test accuracy	Low accuracy overall
Caused by excessive complexity	Caused by insufficient complexity



# Strategies to Address

## 1. For Overfitting:

- Data augmentation, dropout, reduce model complexity, K-fold cross-validation.

## 2. For Underfitting:

- Use a more complex model, increase training time, improve feature engineering, or collect more data.

# Workflow Steps

**Step 1:** Data Cleaning and Preprocessing

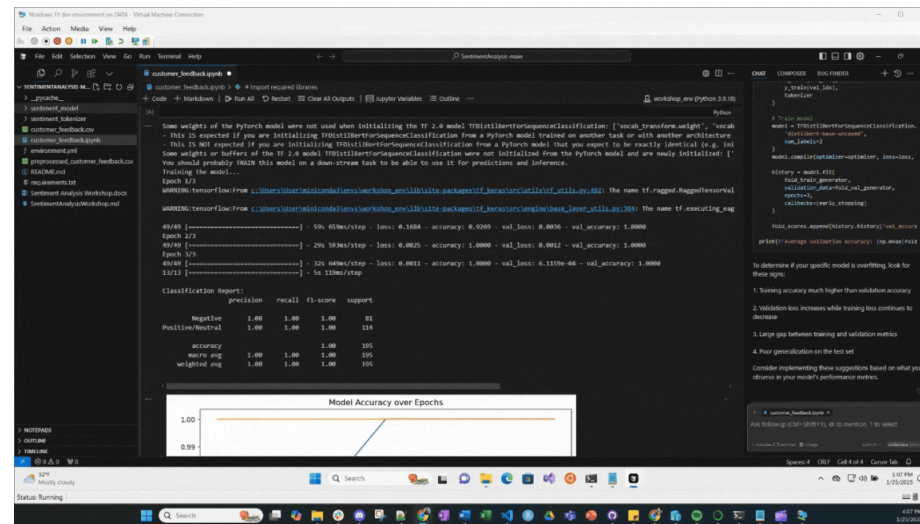
**Step 2:** Model Training and Evaluation

**Step 3:** Deployment and Prediction

## Prompt Example:

Deploy a FastAPI application for sentiment analysis with:

1. TensorFlow integration.
2. Environment variables to suppress TensorFlow warnings.
3. Endpoint: `"/predict"` to classify sentiment.
4. A test script.



# Conclusion

- **Key Steps:**

- Data preparation.
- Model training and evaluation.
- FastAPI deployment.

- **Takeaways:**

- High-quality data is critical.
- Deployment enables real-time applications.
- Simplifying sentiment categories impacts granularity.