

Build a Sentiment Classifier with Vibe Coding

Revisit the NLP pipeline, then compare a traditional classifier with an LLM. Work in pairs in Google Colab and use an LLM as a coding assistant.

Time 40 minutes	Format Pairs or groups of 3	Tools Google Colab + an LLM	Output Notebook + short reflection
--------------------	--------------------------------	--------------------------------	---------------------------------------

Scenario: A restaurant receives hundreds of online reviews. Your task is to build a small AI tool that predicts whether a review expresses positive (1) or negative (0) sentiment.

1 Remember the pipeline (5 minutes)



Explain to your partner what each stage does. Which stage turns words into numbers?

2 Get your dataset (5 minutes)

Dataset: UCI Machine Learning Repository - Sentiment Labelled Sentences (Kotzias et al., 2015). The full dataset has 3,000 labelled sentences; use just the Yelp file (1,000 restaurant-review sentences, 500 per class). 0 = negative, 1 = positive.

Download: archive.ics.uci.edu/dataset/331/sentiment+labelled+sentences

Steps: Click Download -> unzip -> find yelp_labelled.txt -> upload it to your Colab notebook using the Files pane. The file has a review and a 0/1 label, separated by a tab.

3 Build a traditional NLP classifier (15 minutes)

[] Read the file into a table with columns review and label; show the first 5 rows and check how many labels are 0 and 1.

[] Preprocess: lowercase and clean text; demonstrate tokenisation on one example. Keep negation words such as “not” and “never”.

[] Split labelled data into 80% training and 20% testing using a fixed random seed and a stratified split.

[] Extract features: turn reviews into numbers using TF-IDF, learned from training text only.

[] Train and predict: fit a simple Logistic Regression classifier, then predict labels for the test reviews.

[] Evaluate: print precision, recall, F1-score and a confusion matrix for the held-out test set.

Use the LLM to help write and debug code, but be able to identify the code responsible for each pipeline stage.

Test, Explain, Compare

The aim is to understand the pipeline, not merely to obtain working code.

Suggested prompt for your coding assistant

“Create Google Colab Python code for sentiment classification using my uploaded yelp_labelled.txt file. Show the five NLP pipeline stages clearly. Load tab-separated data, clean and tokenise text while keeping negations, make an 80/20 stratified train/test split, use TF-IDF plus Logistic Regression, and report precision, recall, F1 and a confusion matrix. Fit TF-IDF on training data only. Include comments explaining each stage and predict three new review sentences. Use beginner-friendly code.”

4 Demonstrate understanding (7 minutes)

- [] Show the lecturer the lines that perform preprocessing, feature extraction and model training.
- [] In one sentence, explain why TF-IDF is needed before training this classifier.
- [] Change one component: replace TF-IDF with Bag of Words (CountVectorizer). Re-run and record whether the test-set F1-score increases, decreases or stays similar.

5 Now use the LLM directly (5 minutes)

Choose five reviews from your held-out test set. Without showing their labels, ask your LLM to classify each review as Positive or Negative using the following prompt:

“Classify the sentiment of this restaurant review as Positive or Negative. Reply with only one label. Review: [paste review here]”

Compare the two methods on the same five reviews. Do not use the true labels in either model’s prompt. Check both predictions against the hidden dataset labels.

Review #	True label	TF-IDF classifier	Direct LLM
1	_____	_____	_____
2	_____	_____	_____
3	_____	_____	_____
4	_____	_____	_____
5	_____	_____	_____

6 Exit ticket and transition to the LLM lecture (3 minutes)

Discuss: (1) Which system was easier to build? (2) Which correctly classified more of your five test reviews? (3) Why might the two systems make different mistakes? (4) What is the difference between an LLM writing your Python code and an LLM doing sentiment classification itself?

Before you finish: Submit/share your Colab notebook or link, the test-set F1-score for both feature methods, your five-row comparison, and a 2-3 sentence reflection.

Ground rule: You may use an LLM to write code, but you must understand and explain it. Do not upload private or personal reviews.

Instructor note: Five reviews are too few for a reliable comparison. Source: Kotzias (2015), Sentiment Labelled Sentences, UCI ML Repository, DOI: 10.24432/C57604 (CC BY 4.0). Based on the module’s NLP lecture.