EEG-based Prediction of Student Understanding in an Online Lecture Setting

Zuzanna Bąk

Abstract

This project explores whether simple machine learning models can predict if a student understood an online lecture from EEG recordings collected during the class. I use the public "EEG data / Distance learning environment" dataset from Kaggle, which contains 68,831 EEG samples from 8 students who watched 11 short instructional videos. Each sample includes power-spectrum features computed from 14 EEG channels and a binary label indicating whether the student reported understanding the content.

I first describe the dataset and show that the labels are strongly structured by video: for most lectures all samples are labelled either "understood" or "did not understand". I then focus on 70 power-spectrum features and train a feed-forward neural network and an XGBoost classifier under different class imbalance strategies (original distribution, class weights, SMOTE). The models reach test accuracies between 0.72 and 0.76 and F1-scores above 0.83 for the majority class, but confusion matrices and ROC-AUC values reveal that they rarely identify the minority class. I conclude that with the current label structure these models mainly exploit global patterns (video identity and class imbalance) rather than fine-grained differences in EEG signals.

1 Introduction and Motivation

After the COVID-19 lockdown many schools and universities switched to online or hybrid teaching. Online videos are easy to scale, but it is hard for instructors to know if students pay attention and understand the material. EEG-based systems are sometimes proposed as a way to measure engagement or understanding in real time.

The goal of this project is very concrete: given EEG-derived features from a short interval during an online lecture, can a classifier predict whether the student later reported understanding the material (label 1) or not (label 0)? I treat this as a binary supervised learning problem and focus on making each step of the analysis reproducible and easy to follow.

2 Data and Exploratory Analysis

2.1 Dataset overview

The dataset consists of three CSV files:

- **EEG_data.csv**: main table with one row per EEG sample.
- **Subject_details.csv**: metadata for each student.
- **Video_details.csv**: metadata for each lecture video.

In **EEG_data.csv** there are exactly 68,831 rows and 87 columns. The rows are indexed from 0 to 68,830. All 68,831 entries are non-missing for each column, so there are no missing values in this table.

The 87 columns can be grouped as follows:

- **Identifiers**: `video_id` (integer 0–10) and `subject_id` (integer 0–7).
- **Raw EEG channels**: 14 columns such as `EEG.AF3`, `EEG.F7`, `EEG.F3`, ...
- **Power features**: 70 columns corresponding to 5 frequency bands for each of the 14 sensors (e.g., `POW.AF3.theta`, `POW.AF3.alpha`, ...).
- **Target label**: a binary column (named `Understanding` in the original dataset) with values 0 or 1.

The label distribution in the full dataset is:

$$\text{count}(y = 0) = 14{,}461, \quad \text{count}(y = 1) = 54{,}370.$$

This means that 21.0% of the samples are labelled "did not understand" and 79.0% are labelled "understood".

Subject_details.csv contains 8 rows and 6 columns, one row per student (subject IDs 0–7). There are 7 male students and 1 female student. Ages range from 11 to 24 years, with a mean age of 18.0 years and a standard deviation of 4.64 years. The file also stores education level, field of interest, and ethnicity, but I do not use these variables in the models.

Video_details.csv contains information about 11 lecture videos (video IDs 0–10). For each video it stores a title, URL, instructor name, and a short description of the topic.

2.2 Label distribution and video-level effects

Before fitting any models I examined how the understanding label behaves across the dataset.

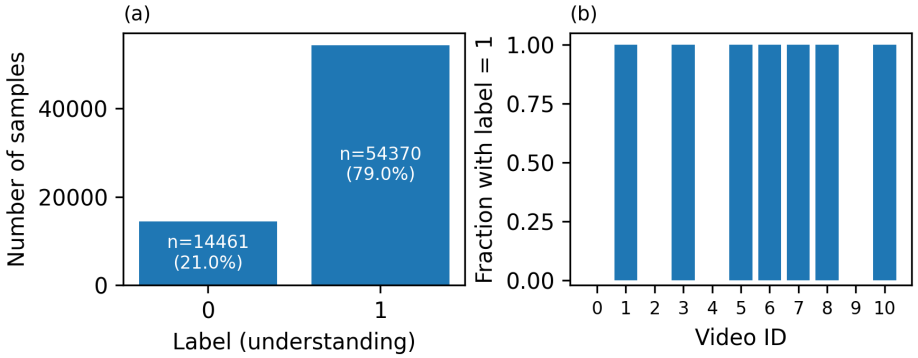

Figure 1: Class distribution in the dataset. (a) Overall number of samples with label 0 ("did not understand") and 1 ("understood"). (b) Fraction of samples with label 1 for each video. For most videos this fraction is exactly 0 or 1, so all samples in that video share the same label.

Figure 1a shows the overall label distribution. In total there are 68,831 samples, with 14,461 labelled as 0 (21.0%) and 54,370 labelled as 1 (79.0%). From a modeling perspective this means

that a naive classifier that always predicts $y = 1$ would already achieve 79% accuracy, so accuracy alone is not a sufficient metric.

To understand how labels behave across videos, I grouped the data by video and computed the mean of the understanding label for each video:

$$\text{mean-understanding}(v) = \frac{1}{n_v} \sum_{i:\, \text{video}_i = v} y_i,$$

where n_v is the number of samples for video v. The right panel in Figure 1b plots this mean for each video.

This reveals a key issue:

- For most videos the mean understanding is exactly 0.0 or exactly 1.0. In other words, for a given video either all samples are labelled "did not understand" or all samples are labelled "understood".
- Only the linear algebra topic, which appears in two videos, shows mixed labels.

This means that the label is almost constant within each video. If a model has access to video identity, it can predict the label almost perfectly without using the EEG features at all. Even when video identity is not used as an explicit feature, such a strong video-level pattern makes it hard to decide whether the model is learning genuine EEG patterns or simply memorizing which videos are "easy" or "hard".

I also checked understanding rates by subject. Some students almost always reported understanding, while others almost never did. However, the strongest pattern in the data is the video-level effect described above.

As an additional sanity check I applied t-SNE to the standardized feature matrix and projected all samples into two dimensions. When the points were colored by `video_id`, the projection formed clear clusters corresponding to individual lectures, while coloring by the binary understanding label produced no clean separation. Together with Figure 1, this supports the view that the dataset is structured primarily by lecture identity rather than by patterns in the EEG signal.

2.3 Preprocessing and features

I started by treating all 84 signal-related columns (14 raw EEG channels and 70 power-spectrum features) as potential inputs. After standardizing these features, I trained simple classifiers and visualized the space with t-SNE as described above. Under a naive row-wise train–test split the best feed-forward network using all 84 features reached a test accuracy of 0.58, but the confusion matrix showed that most of this performance came from predicting the majority class. When I removed `video_id` and enforced a subject-wise split, the same architecture achieved only 0.37 accuracy and the model was mostly guessing. This confirmed that using all raw EEG channels did not make the problem easier in a realistic setting.

Based on these observations I simplified the representation and focused on the power-spectrum (POW) features only. Concretely:

- I selected all 70 columns whose names contained the substring `"POW"` and used them as features X.
- I used the binary understanding column as the target y.

- I did not include `video_id` or `subject_id` as features. This was a deliberate choice, because using them would make it too easy for the model to exploit the video-level label pattern instead of learning from EEG.
- I standardised all 70 features to have zero mean and unit variance on the training set and applied the same scaling to the test set.

The resulting feature matrix therefore has 68,831 rows and 70 columns before splitting and is used in all final experiments reported below.

3 Modeling and Evaluation

3.1 Train–test split

I used a single train–test split created with `train_test_split` from `scikit-learn`. The split was:

- 58,110 samples (85% of the data) in the training set.
- 10,721 samples (15% of the data) in the test set.

The split was stratified by the binary label so that both sets keep the same class proportions:

$$\begin{aligned} \text{train: } \text{count}(y=0) = 12{,}072 \ (20.8\%), \quad \text{count}(y=1) = 46{,}038 \ (79.2\%), \\ \text{test: } \text{count}(y=0) = 2{,}389 \ (22.3\%), \quad \text{count}(y=1) = 8{,}332 \ (77.7\%). \end{aligned}$$

All reported metrics in the next section are computed on the 10,721-sample test set.

3.2 Models

I experimented with several modeling approaches before settling on two final tabular classifiers. In the first stage I treated all 84 signal-related columns (14 raw EEG channels and 70 power-spectrum features) as inputs and trained a small convolutional neural network (CNN) on 2D "heatmaps" derived from the EEG. In this setting the CNN reached at most about 55–58% test accuracy, and the confusion matrices showed that most of the performance came from predicting the majority class. Together with the t-SNE projection described in the EDA section, this suggested that more complex architectures do not automatically extract a useful representation from the raw channels.

For the final analysis I therefore used the 70 scaled power features described in the preprocessing section and focused on two standard models for tabular data:

Feed-forward neural network (FNN). I used `MLPClassifier` from `scikit-learn` as a fully-connected neural network. The model takes the 70-dimensional feature vector as input and outputs the probability of $y = 1$. I used the default `'relu'` hidden activation and `'adam'` optimizer, with a fixed random seed for reproducibility.

XGBoost classifier. As a tree-based comparison point I trained an `XGBClassifier`. Gradient-boosted trees can handle non-linear interactions between features and often work well on tabular data. I set the learning rate and number of trees to modest values to keep training fast and reduce the risk of overfitting.

Class-imbalance handling. Because the data are imbalanced (21% vs. 79%), I also tried several imbalance-handling strategies:

- using the FNN with and without `class_weight='balanced'`,
- adjusting `scale_pos_weight` in XGBoost,
- simple oversampling strategies such as SMOTE in the training set.

These variants slightly changed the metrics but did not solve the core issue that the models rarely detect the minority class. In the next section I report and interpret one configuration for each imbalance-handling strategy, using the exact metrics obtained in the Jupyter notebooks.

4 Results

4.1 FNN performance

Table 1 reports the performance of the feed-forward neural network under different imbalance-handling strategies. For each variant I show accuracy, F1-score for label 1, and ROC-AUC.

Table 1: FNN performance under different imbalance-handling strategies on the test set (10,721 samples).

Variant	Accuracy	F1-score	ROC-AUC
Original (no rebalancing)	0.734	0.836	0.704
Class weights	0.732	0.838	0.677
SMOTE	0.718	0.832	0.627

The original configuration reaches the highest ROC-AUC and slightly better F1-score than the imbalance-handling variants. However, all three settings struggle to recover the minority class (label 0).

To illustrate the error pattern, Table 2 shows the confusion matrix for the original FNN on the test set.

Table 2: Confusion matrix for the FNN model on the test set. True labels are in rows, predicted labels in columns.

	Pred 0	Pred 1
True 0	617	1,772
True 1	1,444	6,888

The FNN correctly classifies many samples with label 1, but it misclassifies a large fraction of the minority class. The model therefore prefers to predict "understood" (label 1) even when the true label is 0.

4.2 XGBoost performance

Table 3 reports the performance of the XGBoost classifier under the same imbalance-handling strategies.

Table 3: XGBoost performance under different imbalance-handling strategies on the test set (10,721 samples).

Variant	Accuracy	F1-score	ROC-AUC
Original (no rebalancing)	0.763	0.865	0.460
Class weights	0.577	0.710	0.445
SMOTE	0.609	0.733	0.489

The original XGBoost configuration achieves the highest accuracy and F1-score, but the ROC-AUC remains low. The confusion matrix in Table 4 helps explain this behaviour.

Table 4: Confusion matrix for the XGBoost model on the test set. True labels are in rows, predicted labels in columns.

	Pred 0	Pred 1
True 0	152	2,237
True 1	860	7,472

Although XGBoost reaches high accuracy, it predicts label 1 for almost all samples and rarely identifies label 0 correctly. The classifier therefore fails to learn EEG patterns that reliably distinguish the two states; instead, it mostly exploits the global class imbalance and the dataset structure described in the EDA section.

4.3 Comparison between FNN and XGBoost

Comparing the original (no rebalancing) configurations of both models, FNN and XGBoost reach test accuracies between 0.73 and 0.76 and F1-scores between 0.84 and 0.87 for label 1. However, XGBoost achieves this by almost always predicting the majority class, which is reflected in its low ROC-AUC (0.46) and the confusion matrix where very few samples with label 0 are detected. The FNN has slightly lower accuracy but a higher ROC-AUC (0.70), which suggests that it makes somewhat better use of the power features. In both cases the models mainly learn to predict "understood" and struggle to identify the minority class.

5 Discussion and Conclusion

This project shows that it is possible to train simple machine learning models on the 70 power-spectrum features and obtain test accuracies between 0.72 and 0.76 and F1-scores above 0.83 for the majority class. However, several aspects of the data and setup limit what we can conclude:

- The labels are almost constant within each video. For most lectures all samples are labelled either "did not understand" or "understood". This means that video-level difficulty is mixed with student understanding, and models can in principle rely on video identity instead of EEG.
- The train–test split is stratified at the row level, not at the subject or video level. The same student and video can appear in both sets, which makes the test performance optimistic compared to a true "new student / new video" scenario.

- The class imbalance (21% vs. 79%) encourages models to predict "understood" most of the time. The confusion matrices for both FNN and XGBoost confirm that the models rarely detect the minority class, even when class weights or SMOTE are used.

A more realistic evaluation would require a subject-wise or video-wise split (for example, training on a subset of students and videos and testing on the rest) and possibly collecting data where each video contains a mix of understanding and non-understanding labels. Within these limitations, this report documents the data characteristics, preprocessing steps, and model behaviour in a way that should allow another researcher to reproduce the analysis from the raw CSV files and the accompanying Jupyter notebooks. The results suggest that, in this dataset, EEG-based prediction of understanding is dominated by global structure (which lecture was watched and how the labels were assigned) rather than by subtle patterns in the EEG signal itself.

References

[1] Kaggle. *EEG Data / Distance Learning Environment*. Dataset available at: `https://www.kaggle.com/datasets/madyanomar/eeg-data-distance-learning-environment/data` (accessed: Nov 7, 2025).

[2] T. Zander and C. Kothe. Towards Passive Brain–Computer Interfaces: Understanding Implicit User States. *Journal of Neural Engineering*, 8(2), 2011.