Study 2: Designing explainable speech-based machine learning for the estimation of job interview outcomes

Shahriar Nekouei
Michelle Gjolberg
Sepideh Mohammadi
Seyed AmirReza Alavi

April 25, 2025

Abstract

Machine learning and algorithmic models are becoming an inseparable part of the modern talent acquisition and interview process. With large companies like Google, Amazon, and Meta receiving thousands of applications, relying solely on human evaluation is both time-consuming and costly. AI-based systems offer scalable and efficient solutions for recruiters. This study aims to explore how such models operate behind the scenes by leveraging the MIT Interview dataset, which includes both prosodic and textual transcripts of interview sessions, as well as annotated performance and excitement scores ranging from 1 to 7. Our goal is to evaluate the effectiveness of speech-based machine learning models in predicting these outcomes and providing structured feedback

1 Problem Statement

The goal of this study is to evaluate and compare the performance of different speech-based machine learning models in estimating interview outcomes such as overall performance and excitement. Specifically, we aim to determine how closely each model's predicted scores align with the actual annotated scores. Additionally, we analyze the relative strengths and weaknesses of each model in terms of predictive accuracy, ability to capture linguistic and prosodic cues, and overall interpretability

2 Data

We used the MIT Interview, which contains multimodal data from mock job interviews [2] [3]. The dataset includes:

- **transcripts.csv** – This file contains the written transcripts of interviews. Each row shows what a participant said during either their first or second interview.
- **prosodic_features.csv** – This file includes audio-related features like pitch, speaking speed, and energy, which help capture how something was said.

- **scores.csv** – This file has average scores given by human annotators. Each interview is rated on two things: overall performance and excitement, both on a scale from 1 to 7.

3 Methodology

3.1 Extracting language features

Extracting language features and Feature selection are crucial steps in building effective machine learning models, especially when working with high-dimensional textual and audio data. In this section, we explore three different types of features extracted from interview transcripts: word frequency-based features (TF-IDF), sentiment scores (VADER), and semantic embeddings (Word2Vec). Each method captures different aspects of the interview responses, ranging from literal word usage to underlying tone and contextual meaning, allowing us to assess which types of language patterns best predict interview outcomes.

3.1.1 TF-IDF

TF-IDF stands for *Term Frequency-Inverse Document Frequency*. It is a method used to convert text into numbers that machine learning models can understand.

- Words are features (columns in the TF-IDF matrix).
- Each transcript is a document (row in the TF-IDF matrix).
- The result is a matrix where rows = transcripts, and columns = words, with each cell holding the TF-IDF score for that word in that transcript.

Term Frequency (TF) measures how often a word appears in a single transcript. **Inverse Document Frequency (IDF)** reduces the importance of common words that appear in many transcripts.

The formula is:

$$TF\text{-}IDF(t,d) = \frac{f_{t,d}}{N_d} \times \log\left(\frac{N}{1+n_t}\right)$$

where:

- $f_{t,d}$: number of times word t appears in transcript d
- N_d: total number of words in transcript d
- N: total number of transcripts
- n_t: number of transcripts containing the word t

3.1.2 Sentiment Analysis using VADER

To explore the sentiment of interview transcripts, we used the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis tool, which provides sentiment scores for each text input. VADER assigns a compound score between -1 (most negative) and 1 (most positive).

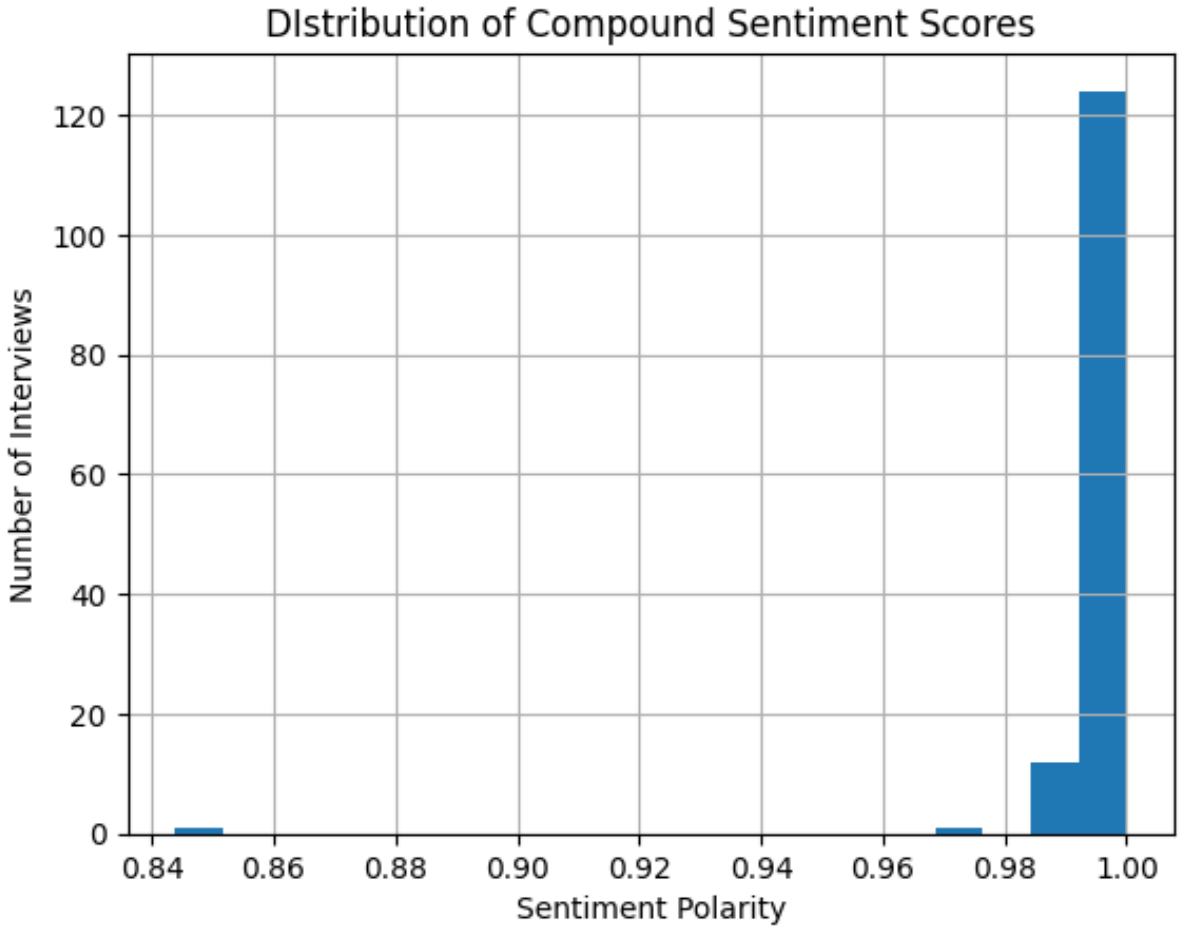

Figure 1: Distribution of Compound Sentiment Scores Using VADER

As shown in Figure 1, most of the compound sentiment scores are clustered near 1, indicating that the transcripts are overwhelmingly positive. Because of this, there is little variability in sentiment scores, making them less useful as features for predicting interview performance or excitement.

Despite this, we chose to include this analysis to demonstrate the importance of evaluating feature distributions before modeling. In other contexts, such as social media posts, customer feedback, or emotionally charged speech, sentiment analysis may offer much richer and more discriminative signals. However, in the case of job interviews, participants likely strive to sound positive and composed, resulting in uniformly high sentiment scores.

It's also worth noting that VADER is primarily designed for social media-style, informal language, and may not capture subtler expressions of stress, confidence, or emotion that are common in professional interview settings.

3.1.3 Word Embedding using Word2Vec

To extract higher-level semantic features from the interview transcripts, we used Word2Vec, a popular word embedding model. This model learns to represent words as dense vectors in a continuous space based on the contexts in which they appear, typically using a shallow neural network trained on a large corpus. Each word is mapped to a 300-dimensional vector that encodes its semantic relationships with other words.

For our analysis, we used a pre-trained Word2Vec model and computed the mean vector of all words in each transcript. This process yields a single 300-dimensional embedding that summarizes the semantic content of the entire transcript. These embeddings capture complex, contextual language features that go beyond surface-level word frequency.

However, these features are not easily interpretable. Unlike TF-IDF, where each feature corresponds to a known word, the dimensions in Word2Vec embeddings are abstract and do not have explicit meanings. This makes them powerful and useful for capturing subtle patterns in language, but challenging to analyze or explain in human-understandable terms.

4 Language Feature Selection

After extracting textual features using TF-IDF and Word2Vec, we performed feature selection to identify the most informative signals for predicting interview outcomes. Since both extraction methods produce high-dimensional representations, especially Word2Vec with its 300 semantic dimensions, feature selection is critical to reduce redundancy, improve model efficiency, and enhance interpretability.

We used Pearson correlation to measure the strength of association between each individual feature and the target variables: excitement and overall performance scores. By selecting the top K most correlated features with statistically significant p-values, we ensured that only the most relevant linguistic indicators were retained for downstream modeling. In the following subsections, we present the selection process and findings for both TF-IDF and Word2Vec features.

4.1 TF-IDF Feature Selection

We evaluated TF-IDF features based on their Pearson correlation with the target interview outcomes: excitement and overall performance. To determine the optimal number of features (K), we compared model performance across $K = 15, 20, 30$ using both Random Forest and Feedforward Neural Networks (FNN). Models were evaluated using 5-fold cross-validation, which helps ensure robustness by testing model performance on different subsets of the data. This method provides a more reliable estimate of generalization by reducing the risk of overfitting to a single train-test split. Evaluation metrics included Pearson correlation (r) and Relative Error (RE).

Table 1: 5-Fold Evaluation Using TF-IDF Features with Random Forest

K	**Target Variable**	**Pearson r**	**Relative Error**
15	Excitement Score	0.530	0.082
15	Overall Score	0.549	0.061
20	Excitement Score	0.560	0.082
20	Overall Score	0.571	0.061
30	Excitement Score	0.508	0.082
30	Overall Score	0.539	0.062

Table 2: 5-Fold Evaluation Using TF-IDF Features with Feedforward Neural Network (FNN)

K	**Target Variable**	**Pearson r**	**Relative Error**
15	Excitement Score	0.6073	0.0780
15	Overall Score	0.6286	0.0629
20	Excitement Score	0.6361	0.0726
20	Overall Score	0.6410	0.0576
30	Excitement Score	0.7082	0.0710
30	Overall Score	0.6402	0.0586

Based on these results, we selected $K = 20$ as the optimal value. It provided a balance between model performance and feature interpretability, showing consistently strong results across both models.

Top 20 TF-IDF Features at $K = 20$

Using the top 20 features selected at $K = 20$, we computed the Pearson correlation between each word's TF-IDF score and the target variables: excitement and overall performance. We applied a significance filter of $p < 0.05$ to ensure statistical relevance.

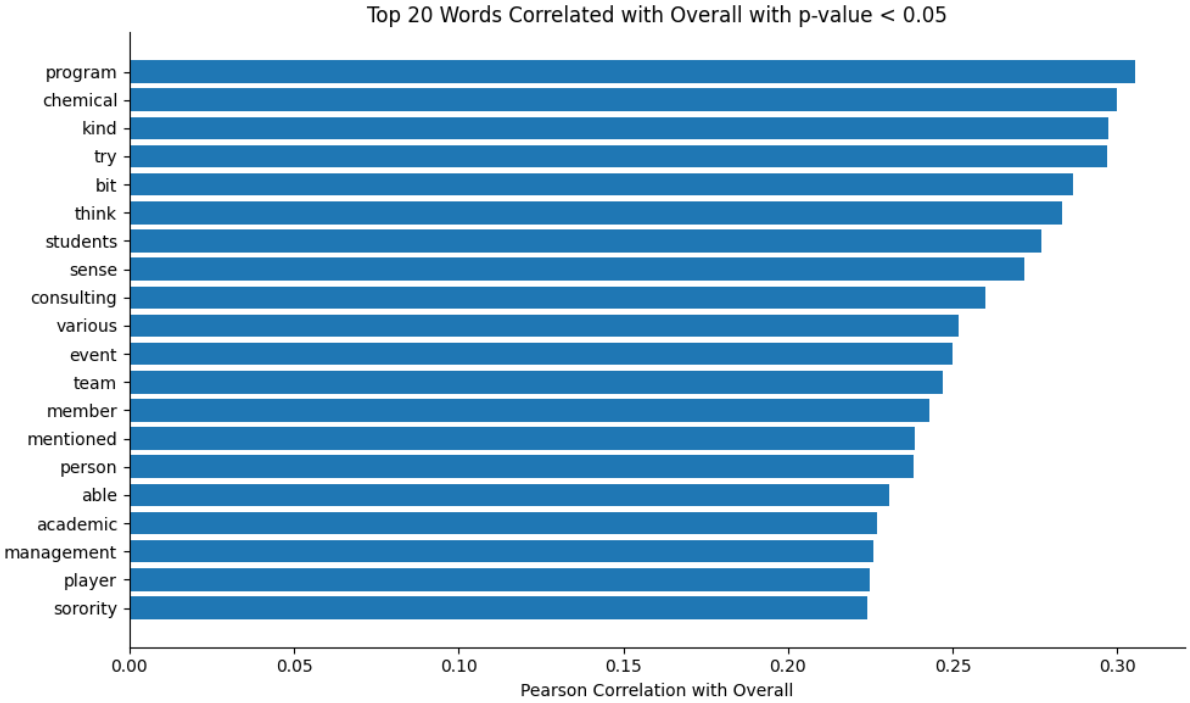

Figure 2: Top 20 Words Positively Correlated with Overall Score

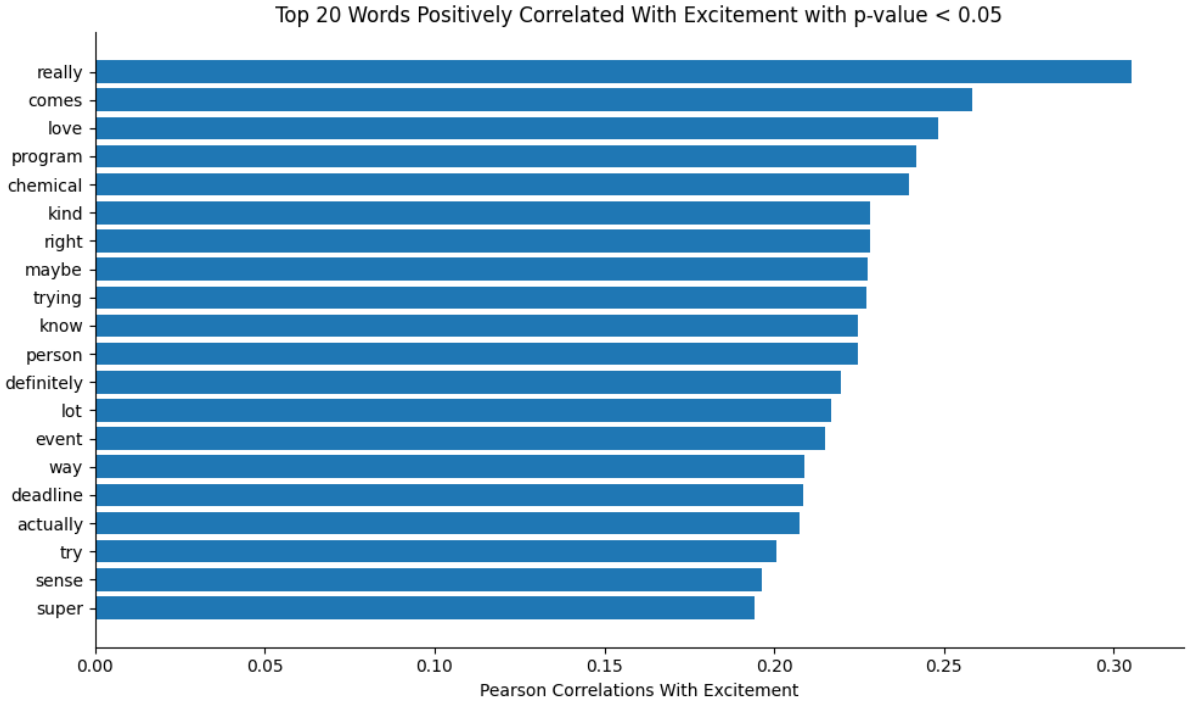

Figure 3: Top 20 Words Positively Correlated with Excitement Score

The top TF-IDF words associated with higher positive excitement scores include emotionally expressive terms like "really," "kind," and "love." These suggest that candidates who use enthusiastic and positive language are perceived as more energetic and engaged during interviews. For overall performance, terms like "consulting," "managment," and "team" ranked highly, indicating that interviewees who emphasize qualifications and interpersonal abilities tend to be rated more favorably.

4.2 Feature Selection for Word2Vec

For Word2Vec embeddings, as each transcript is represented by a 300-dimensional vector, the average of all word embeddings in that transcript, we computed the Pearson correlation between each dimension and the interview outcome scores. We then selected the top 20 dimensions most correlated with each target variable, excitement and overall performance, using a significance threshold of $p < 0.05$. This threshold ensures that the observed correlations are unlikely to have occurred by random chance, providing statistical confidence that the selected features are meaningfully associated with the outcomes. These dimensions were used as the most informative semantic features for our models.

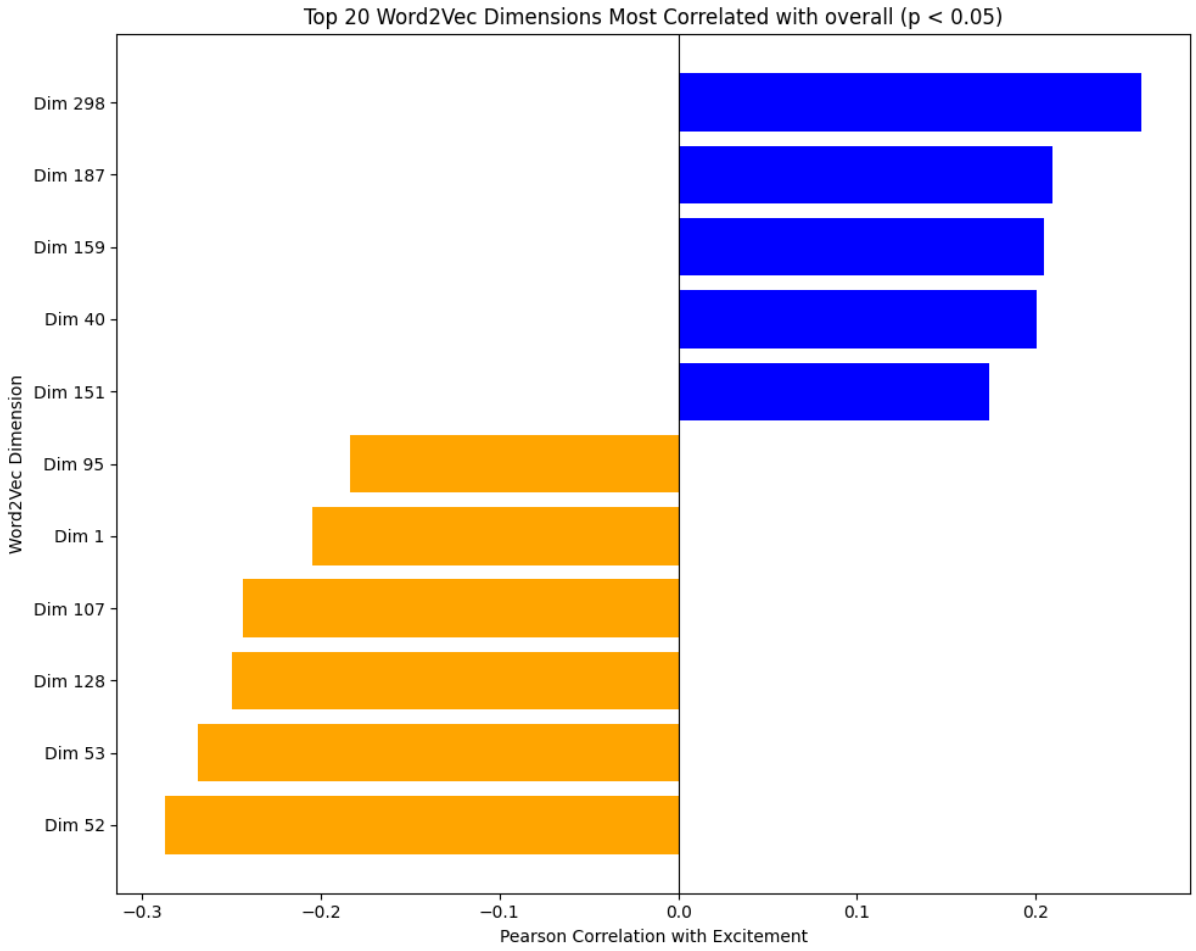

Figure 4: Top 20 Word2Vec Dimensions Correlated with Overall Score

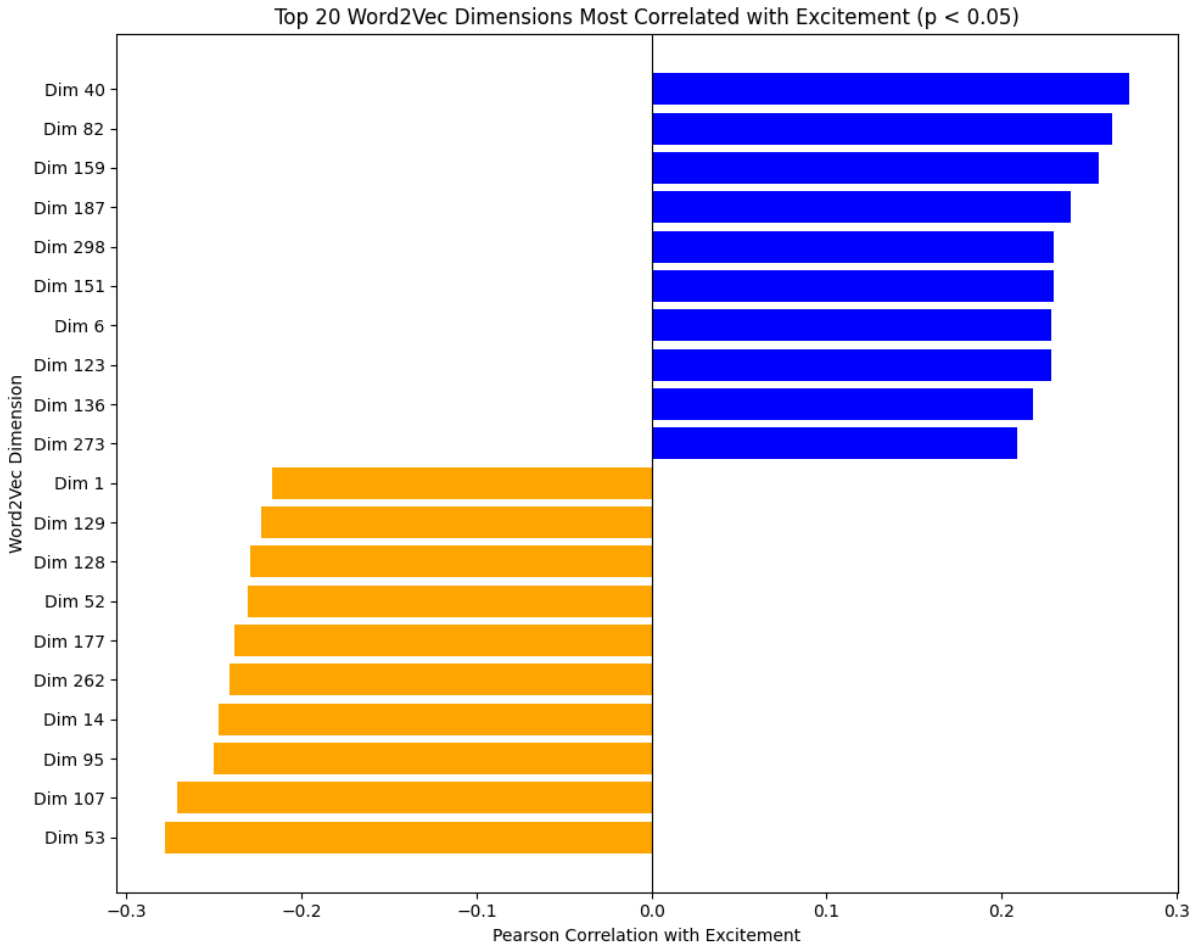

Figure 5: Top 20 Word2Vec Dimensions Correlated with Excitement Score

Word2Vec appears to capture excitement more effectively than overall performance, as a greater number of dimensions were significantly correlated with excitement scores. While individual Word2Vec dimensions are harder to interpret directly, the ones most strongly associated with excitement seem to reflect semantic patterns tied to social positivity and expressive tone. This suggests that these embeddings may be picking up on latent traits like enthusiasm or confidence, reinforcing the idea that how something is said plays a key role in how emotionally engaging a candidate is perceived to be.

4.3 Estimating Interview Outcomes Based on Language

4.3.1 Random Forest

To evaluate how well language features predict interview outcomes, we trained Random Forest models using the top 20 features selected from both TF-IDF and Word2Vec representations. Model performance was assessed using 5-fold cross-validation to ensure robustness and generalizability. The tables below report the average Pearson correlation coefficients and relative errors for predicting excitement and overall performance scores.

Table 3: 5-Fold Evaluation Results Using Top 20 TF-IDF Features (Random Forest)

Target Variable	**Average Pearson r**	**Average Relative Error**
Excitement Score	0.560	0.082
Overall Score	0.571	0.061

To better understand the impact of feature set size, we conducted further experiments using different values of K: 15, 20, and 30. The table below summarizes model performance across these configurations, along with the best hyperparameters identified via `RandomizedSearchCV`.

Table 4: Summary of 5-Fold Evaluation Results and Best Hyperparameters Across TF-IDF Feature Set Sizes (Random Forest)

K	Target	Pearson r	Relative Error	Best Hyperparameters
15	Excitement	0.530	0.082	`n_estimators=200, max_depth=30, max_features='log2', min_samples_split=10`
15	Overall	0.549	0.061	`n_estimators=200, max_depth=30, max_features='log2', min_samples_split=10`
20	Excitement	0.560	0.082	`n_estimators=200, max_depth=30, max_features='sqrt', min_samples_split=5`
20	Overall	0.571	0.061	`n_estimators=100, max_depth=30, max_features='sqrt', min_samples_split=10`
30	Excitement	0.508	0.082	`n_estimators=200, bootstrap=False, max_depth=10, max_features='log2', min_samples_leaf=2, min_samples_split=10`
30	Overall	0.539	0.062	`n_estimators=100, max_depth=30, max_features='sqrt', min_samples_split=10`

4.3.2 Random Forest with Word2Vec Features

We trained the Random Forest models using the top 20 Word2Vec dimensions, optimized via `RandomizedSearchCV`, and evaluated them with 5-fold cross-validation. The results below reflect the best performance achieved for predicting Excitement and Overall interview outcomes.

Table 5: 5-Fold Evaluation Results Using Top 20 Word2Vec Dimensions (Random Forest)

Target Variable	Average Pearson r	Average Relative Error
Excitement Score	0.441	0.082
Overall Score	0.354	0.067

To assess the impact of feature set size, we evaluated performance using 15, 20, and 30 Word2Vec dimensions. The table below summarizes the model performance along with the best hyperparameters selected by `RandomizedSearchCV`.

Table 6: Summary of 5-Fold Evaluation Results and Best Hyperparameters Across Word2Vec Feature Set Sizes (Random Forest)

K	**Target**	**Pearson r**	**Relative Error**	**Best Hyperparameters**
15	Excitement	0.441	0.082	`n_estimators=100`, `max_depth=30`, `max_features='sqrt'`, `min_samples_split=10`
15	Overall	0.354	0.067	`n_estimators=200`, `bootstrap=False`, `max_depth=10`, `max_features='log2'`, `min_samples_leaf=2`, `min_samples_split=10`
20	Excitement	0.441	0.082	`n_estimators=200`, `max_depth=30`, `max_features='log2'`, `min_samples_split=10`
20	Overall	0.354	0.067	`n_estimators=200`, `max_features='sqrt'`, `min_samples_leaf=2`
30	Excitement	0.425	0.083	`n_estimators=200`, `max_depth=30`, `max_features='log2'`, `min_samples_split=10`
30	Overall	0.395	0.066	`n_estimators=200`, `bootstrap=False`, `max_depth=30`, `max_features='log2'`, `min_samples_split=5`

These results indicate that TF-IDF features generally outperformed Word2Vec dimensions when using Random Forest, particularly for predicting overall interview performance. Nevertheless, Word2Vec captured meaningful signals related to emotional tone, as seen in its competitive results for excitement prediction. This suggests that while frequency-based features provide more discriminative power for performance, semantic embeddings offer complementary insight for affective analysis.

4.3.3 Feedforward Neural Network

We trained a Feedforward Neural Network (FNN) to predict Excitement and Overall Performance scores using the top K selected features from both TF-IDF and Word2Vec representations. The FNN was implemented using Keras with TensorFlow backend.

The architecture of the model consists of:

- An input layer with dimensionality matching the number of selected features (K)
- A fully connected (dense) hidden layer with 64 units and ReLU activation
- A dropout layer with a rate of 0.3 to prevent overfitting
- A second dense hidden layer with 32 units and ReLU activation
- An output layer with a single neuron (no activation), producing a continuous score prediction

The model was compiled using the Adam optimizer and trained with Mean Squared Error (MSE) loss. Evaluation was performed using 5-fold cross-validation. For each fold, we computed the Pearson correlation coefficient (r) and the Relative Error (RE), where RE is defined as the mean absolute error divided by the 7-point scoring range.

The architecture was chosen to balance expressiveness and overfitting risk. A moderate number of neurons and dropout regularization helps the model generalize across folds while avoiding excessive complexity given the relatively small feature set (K = 15–30).

Below we present the results for $K = 20$ as well as the best-performing configurations across $K = 15$, $K = 20$, and $K = 30$, which were determined by tuning the number of training epochs and batch size.

Table 7: 5-Fold Evaluation Results Using Top 20 TF-IDF Features (FNN)

Target Variable	**Pearson r**	**Relative Error**
Excitement Score	0.6361	0.0726
Overall Score	0.6410	0.0576

Hyperparameter tuning was conducted using a manual grid search over the number of epochs {30, 50, 100} and batch sizes {8, 16, 32}. The best configuration was selected based on the highest average Pearson r on validation folds.

Table 8: Best FNN Results Using TF-IDF Features (Top K = 15, 20, 30)

K	Epochs	Batch Size	Target Variable	Pearson r	Relative Error
15	100	32	Excitement Score	0.6073	0.0779
15	100	32	Overall Score	0.6286	0.0629
20	30	32	Excitement Score	0.6361	0.0726
20	30	32	Overall Score	0.6410	0.0576
30	30	8	Excitement Score	**0.7082**	**0.0710**
30	30	8	Overall Score	**0.6402**	**0.0586**

These results show that while $K = 20$ yielded strong baseline results, the best performance for excitement score prediction was obtained with $K = 30$ and tuned hyperparameters (epochs = 30, batch size = 8). This highlights the importance of hyperparameter tuning alongside feature selection.

4.3.4 Feedforward Neural Network with Word2Vec

We trained a Feedforward Neural Network (FNN) model using the top K Word2Vec-based features to predict Excitement and Overall Performance scores. The models were evaluated using 5-fold cross-validation. Below we present the results for $K = 20$ along with the best-performing configurations from $K = 15$, $K = 20$, and $K = 30$.

Table 9: 5-Fold Evaluation Results Using Top 20 Word2Vec Features (FNN)

Target Variable	Pearson r	Relative Error
Excitement Score	0.6422	0.0713
Overall Score	0.4877	0.0604

Table 10: Best FNN Results Using Word2Vec Features (Top K = 15, 20, 30)

K	Epochs	Batch Size	Target Variable	Pearson r	Relative Error
15	50	8	Excitement Score	0.6143	0.0764
15	100	8	Overall Score	**0.5641**	**0.0642**
20	100	8	Excitement Score	**0.6422**	**0.0713**
20	30	16	Overall Score	0.4877	0.0604
30	30	8	Excitement Score	0.4372	0.0884
30	100	8	Overall Score	0.4024	0.0676

These results indicate that the best performance for predicting Excitement was obtained using Word2Vec features with $K = 20$ (epochs = 100, batch size = 8), while the best Overall Score prediction occurred with $K = 15$ (epochs = 100, batch size = 8). This emphasizes the role of both embedding richness and training configuration.

4.3.5 Real-World suitability and consideration

While our models show varying levels of predictive accuracy, the best-performing approach, an FNN using TF-IDF features—achieved a Pearson correlation of 0.7082 for excitement prediction and 0.6201 for overall performance. In real-world applications such as interview coaching tools, these scores may be acceptable when paired with interpretable feedback mechanisms, allowing users to understand why certain scores were assigned.

The relative error rates were low, which suggests the models make fairly consistent predictions. However, in high-stakes settings such as hiring, explainability becomes critical. Black-box models may need to be supplemented with transparent techniques to ensure fairness and trust.

From a deployment standpoint, Random Forests offer a good trade-off between performance and computational efficiency, making them suitable for real-time or edge deployment scenarios. In contrast, FNNs require more compute resources, but may generalize better if tuned properly. Transformer models, while powerful in theory, demonstrated high computational cost and poor zero-shot accuracy in our case, indicating that fine-tuning and prompt engineering are likely essential for these tools to be practical.

Overall, our findings suggest that moderately accurate, interpretable models can be highly useful for interview feedback, particularly if presented as coaching tools rather than gatekeeping systems.

5 Multimodal Machine Learning Models

In this section, we explore the predictive value of prosodic features, such as pitch, energy, speaking rate, and pause duration, extracted from interview audio recordings. These features capture how something is said, offering insight into vocal delivery and emotional tone. We first evaluate models trained on prosodic features alone, then assess whether combining them with textual representations (TF-IDF and Word2Vec) enhances prediction performance in a multimodal setup.

5.1 Prosodic Features on Random forest and Feedfowrad NN

To isolate the contribution of non-verbal vocal cues, we trained models using only prosodic features. Both Random Forest and Feedforward Neural Network (FNN) architectures were used to predict Excitement and Overall Performance scores. Model performance was evaluated using 5-fold cross-validation, and the results are summarized below.

Table 11: 5-Fold Evaluation Results Using Prosodic Features Only (Random Forest)

Target Variable	**Average Pearson r**	**Average Relative Error**
Excitement Score	0.163	0.095
Overall Score	0.264	0.071

Random Forest Hyperparameters:

- *Excitement Score:* `bootstrap=False`, `max_depth=30`, `max_features='log2'`, `min_samples_leaf=`
  `n_estimators=500`, `random_state=42`
- *Overall Score:* `max_depth=10`, `max_features='sqrt'`, `random_state=42`

Table 12: 5-Fold Evaluation Results Using Prosodic Features Only (Feedforward Neural Network)

Target Variable	**Average Pearson r**	**Average Relative Error**
Excitement Score	0.061	0.318
Overall Score	-0.036	0.270

These results show that Random Forest models outperform Feedforward Neural Networks when relying solely on prosodic features. In particular, the FNN model struggled to predict both excitement and overall performance, showing low correlation and high error. In the next sections, we will explore whether incorporating textual features, TF-IDF and Word2Vec, alongside prosody leads to stronger multimodal models.

5.1.1 Combining Prosodic and TF-IDF Features (Random Forest)

To explore the benefit of combining verbal and non-verbal information, we trained Random Forest models using both TF-IDF features (top K correlated words) and prosodic features as input. This multimodal approach allows the model to leverage both the content of what was said and how it was delivered. We evaluated three values of K: 15, 20, and 30.

Table 13: 5-Fold Evaluation Results Using TF-IDF + Prosodic Features (Random Forest, $K = 20$)

Target Variable	Average Pearson r	Average Relative Error
Excitement Score	0.466	0.086
Overall Score	0.509	0.064

Random Forest Hyperparameters:

- **Excitement Score (All K values)**:
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='log2', min_samples_leaf=2, n_estimators=500, random_state=42)`
- **Overall Score ($K = 15$, $K = 20$)**:
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='log2', min_samples_split=5, n_estimators=200, random_state=42)`
- **Overall Score ($K = 30$)**:
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='log2', min_samples_leaf=2, n_estimators=500, random_state=42)`

Table 14: Best FNN Results Using TF-IDF + Prosodic Features (Random Forest, $K = 15, 20, 30$)

K	Target Variable	Pearson r	Relative Error
15	Excitement Score	0.464	0.086
15	Overall Score	0.493	0.064
20	Excitement Score	**0.466**	**0.086**
20	Overall Score	**0.509**	**0.064**
30	Excitement Score	0.448	0.086
30	Overall Score	0.491	0.065

These results indicate that combining TF-IDF with prosodic features improves prediction performance for overall scores compared to using either modality alone. The best performance was obtained with $K = 20$, for both Random Forest and FNN, confirming that optimal feature selection combined with multimodal input can enhance model accuracy. For excitement prediction, gains were more modest but still showed consistent improvement across configurations.

5.1.2 Combining Word2Vec and Prosodic Features (Random Forest)

We trained Random Forest models using Word2Vec embeddings combined with prosodic features to predict Excitement and Overall Performance scores. Below are the 5-fold evaluation results for $K = 15$, $K = 20$, and $K = 30$.

Table 15: 5-Fold Evaluation Results Using Word2Vec + Prosodic Features (Random Forest)

K	Target Variable	Pearson r	Relative Error
15	Excitement Score	0.439	0.086
15	Overall Score	0.427	0.066
20	Excitement Score	**0.470**	**0.085**
20	Overall Score	**0.451**	**0.066**
30	Excitement Score	0.420	0.088
30	Overall Score	0.405	0.068

Random Forest Hyperparameters:

- **Excitement Score** ($K = 15$):
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, n_estimators=500, random_state=42)`
- **Excitement Score** ($K = 20$):
  - `RandomForestRegressor(bootstrap=False, max_depth=10, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, random_state=42)`
- **Excitement Score** ($K = 30$):
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='log2', min_samples_leaf=2, n_estimators=500, random_state=42)`
- **Overall Score (All K values)**:
  - `RandomForestRegressor(bootstrap=False, max_depth=30, max_features='log2', min_samples_leaf=2, n_estimators=500, random_state=42)`

These results suggest that combining Word2Vec with prosodic features yields solid performance, with $K = 20$ providing the best results for both Excitement and Overall score predictions. This highlights the importance of carefully tuned hyperparameters and balanced feature selection when leveraging semantic and acoustic cues.

5.2 Best Hyperparameters for FNN Models (TF-IDF + Prosodic)

Table 16 presents the best-performing hyperparameter configurations for Feedforward Neural Network (FNN) models that incorporate both TF-IDF and prosodic features. For each value of $K \in \{15, 20, 30\}$, we report the epoch count, batch size, average Pearson correlation r, and average relative error across 5-fold cross-validation. Results are shown separately for the prediction of Excitement and Overall interview scores.

Table 16: Best FNN Hyperparameters for TF-IDF + Prosodic Features (K = 15, 20, 30)

K	Target	Epochs	Batch Size	Pearson r	Relative Error
15	Excitement	100	8	0.1419	0.2462
15	Overall	100	8	0.0986	0.1885
20	Excitement	100	8	0.0822	0.2438
20	Overall	100	8	0.1005	0.2064
30	Excitement	100	16	0.0516	0.1714
30	Overall	100	8	0.1161	0.2236

5.3 Best Hyperparameters for FNN Models (Word2Vec + Prosodic)

Table 17 presents the best-performing hyperparameter configurations for Feedforward Neural Network (FNN) models that incorporate both Word2Vec and prosodic features. For each value of $K \in \{15, 20, 30\}$, we report the number of training epochs, batch size, average Pearson correlation r, and average relative error across 5-fold cross-validation. Results are reported separately for the prediction of Excitement and Overall interview scores.

Table 17: Best FNN Hyperparameters for Word2Vec + Prosodic Features (K = 15, 20, 30)

K	Target	Epochs	Batch Size	Pearson r	Relative Error
15	Excitement	100	8	0.1419	0.2462
15	Overall	100	8	0.0986	0.1885
20	Excitement	100	8	0.0822	0.2438
20	Overall	100	8	0.1005	0.2064
30	Excitement	100	16	0.0235	0.1717
30	Overall	100	16	0.1081	0.2257

Commentary on Results The results in Table 17 indicate that combining Word2Vec and prosodic features using a Feedforward Neural Network (FNN) yielded limited predictive power. Across all values of K, the Pearson correlation r remained relatively low for both excitement and overall performance predictions, with the highest value being $r = 0.1419$ for excitement at $K = 15$. This suggests a weak linear relationship between the predicted and true scores.

Increasing the number of features to $K = 30$ did not result in better performance; in fact, it led to a drop in correlation for excitement prediction. This may indicate that adding more features introduced noise rather than improving the signal, possibly due to overfitting or insufficient model

5.4 Modality contributions and feature impact

Our results highlight how different modalities contribute uniquely to interview outcome prediction:

- **Textual content (TF-IDF)** proved to be the most reliable modality overall, especially for predicting performance scores. Words related to teamwork, profession-

alism, and enthusiasm such as *"team," "consulting,"* and *"excited"* were among the most informative features.

- **Semantic embeddings (Word2Vec)** captured more abstract, latent dimensions of language. While harder to interpret, their stronger correlation with excitement scores suggests they may reflect emotional expressiveness or confidence beyond literal word usage.
- **Prosodic features** contributed less predictive power overall but showed potential for enhancing performance score predictions, especially when combined with text features. Elements like pitch variability and pause duration may subtly reflect confidence or fluency.
- **Multimodal combinations** were most effective for overall performance prediction. TF-IDF + prosody (Random Forest) outperformed individual modalities, indicating that combining what is said and how it is said leads to more holistic candidate evaluation.

Overall, verbal content remains the dominant signal in predicting interview outcomes, but delivery features, especially prosody, can provide complementary cues, particularly when modeled jointly with linguistic information.

6 Explainable Machine Learning Algorithms

To improve interpretability in our modeling, we used the Explainable Boosting Machine (EBM), a transparent and interpretable machine learning model based on generalized additive models. EBM provides both global and local explanations for how features influence predictions, making it especially valuable in high-stakes or human-centered domains.

6.1 EBM for TF-IDF Features Predicting Excitement

In this subsection, we focus on using EBM to predict excitement scores based on the top 20 TF-IDF features. EBM allows us to understand not only the overall importance of each word but also how individual word values contribute to predicted scores.

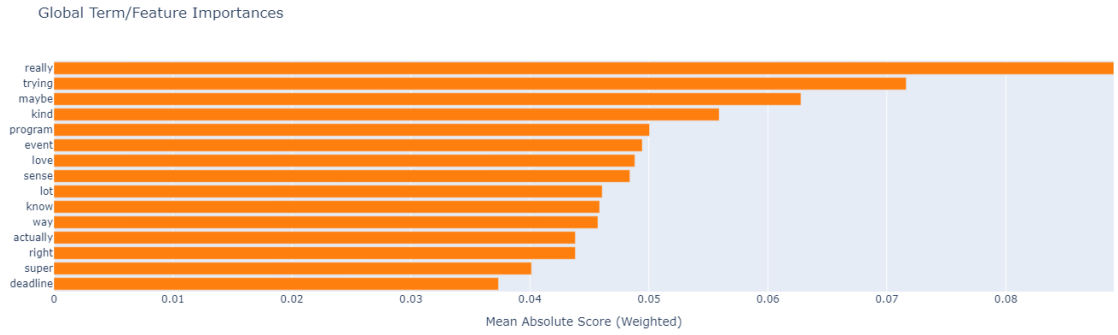

Figure 6: Global Interpretation of Top 20 TF-IDF Features (EBM - Excitement Prediction)

EBM Evaluation Metrics (Excitement Prediction using TF-IDF):

- R^2 Score: 0.465
- Mean Absolute Error (MAE): 0.547
- Relative Error: 0.078

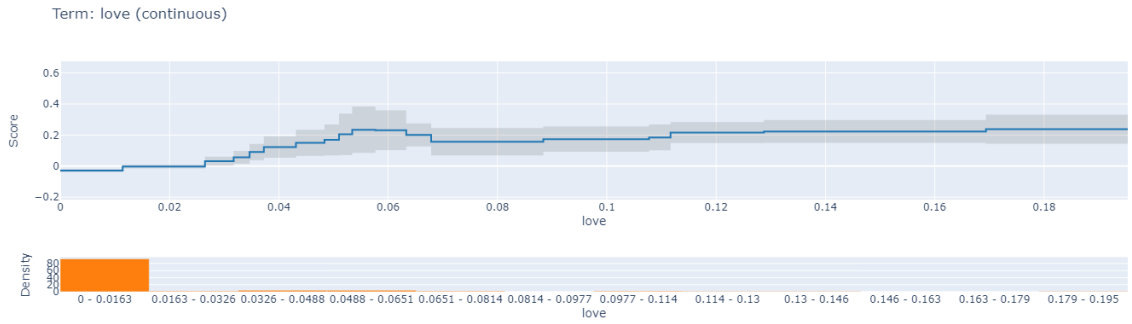

Figure 7: Feature Contribution Curve for the Word *"love"* (TF-IDF)

Figure 7 shows the individual contribution of the word *"love"* to the model's prediction.

Higher TF-IDF values of this word are associated with increased excitement scores, as visualized by the contribution curve. This interpretability is one of EBM's main advantages over traditional black-box models.

6.2 EBM for TF-IDF Features Predicting Overall Performance

We also applied the Explainable Boosting Machine (EBM) to predict overall performance scores using the top 20 TF-IDF features. This interpretable model helps us understand how individual words contribute to the overall impression of a candidate during the interview.

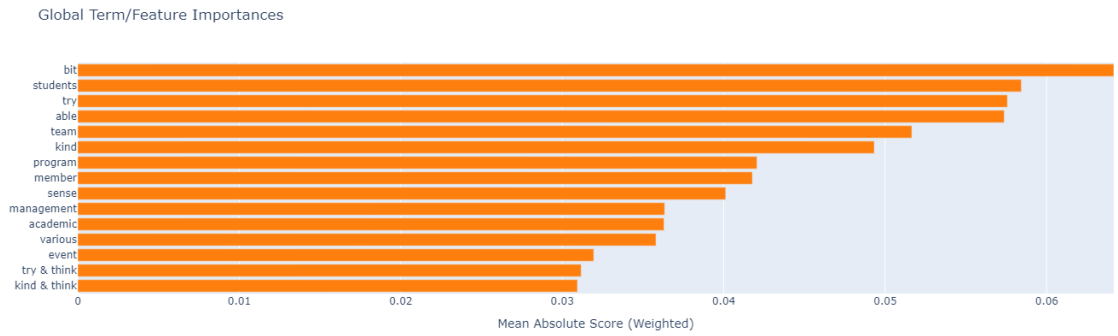

Figure 8: Global Interpretation of Top 20 TF-IDF Features (EBM - Overall Score Prediction)

EBM Evaluation Metrics (Overall Prediction using TF-IDF):

- R^2 Score: 0.332
- Mean Absolute Error (MAE): 0.410
- Relative Error: 0.059

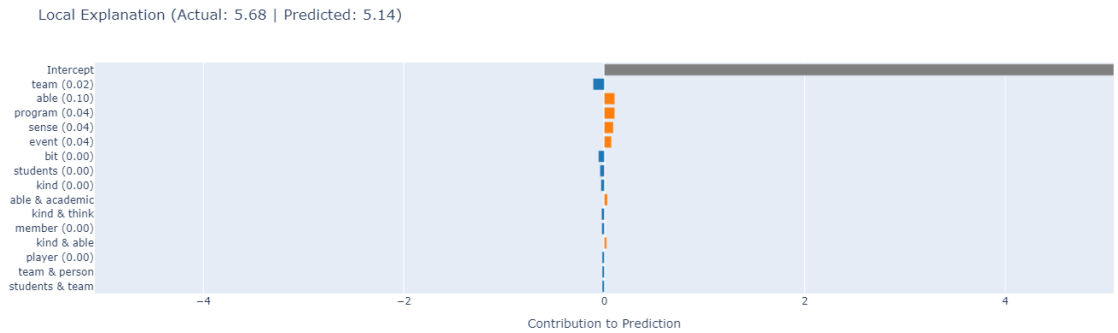

Figure 9: Local Interpretation Example (EBM - TF-IDF Features for Overall Score)

Figure 9 shows a local explanation for an individual prediction, illustrating how specific words and their TF-IDF weights influenced the model's decision. These visualizations are useful for diagnosing model behavior and providing human-understandable feedback in interview evaluation settings.

6.3 EBM for Word2Vec Features Predicting Excitement

We also applied the Explainable Boosting Machine (EBM) to the top 20 most informative Word2Vec dimensions to predict excitement scores. This approach explores whether semantic embeddings, when used with an interpretable model, can offer meaningful insights into how language relates to perceived excitement.

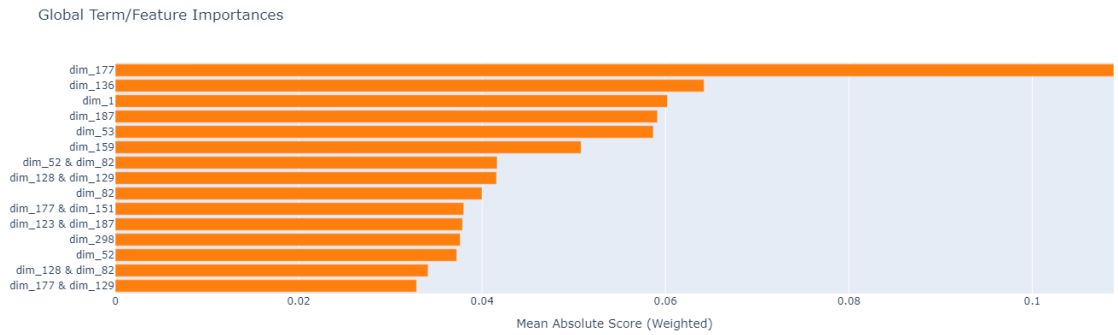

Figure 10: Global Interpretation of Top 20 Word2Vec Dimensions (EBM - Excitement Prediction)

EBM Evaluation Metrics (Excitement Prediction using Word2Vec):

- R^2 Score: 0.086
- Mean Absolute Error (MAE): 0.717
- Relative Error: 0.102

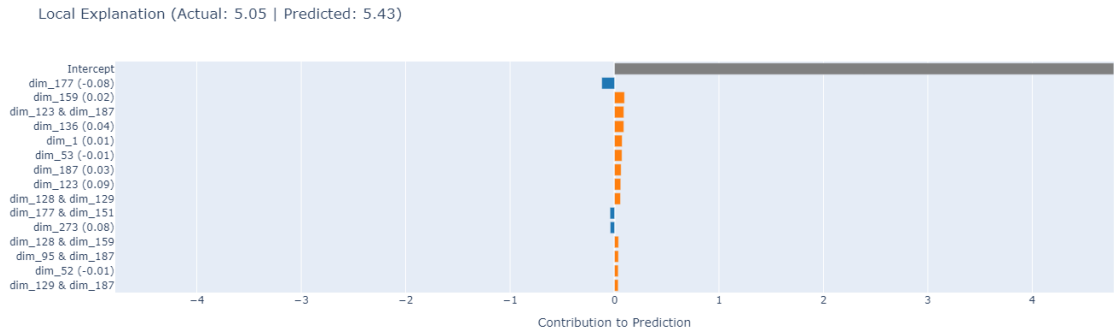

Figure 11: Local Interpretation Example (EBM - Word2Vec Features for Excitement)

Figure 11 shows a local explanation for an individual prediction using Word2Vec features. Although this approach offers some interpretability, the overall predictive performance was relatively low, suggesting that Word2Vec embeddings may not be as effective in this context when used alone.

6.4 EBM for Word2Vec Features Predicting Overall Performance

We also evaluated the Explainable Boosting Machine (EBM) using the top 20 most informative Word2Vec dimensions to predict overall performance scores. While the model provides interpretable insights into how specific dimensions influence predictions, the overall performance was notably lower compared to TF-IDF features.

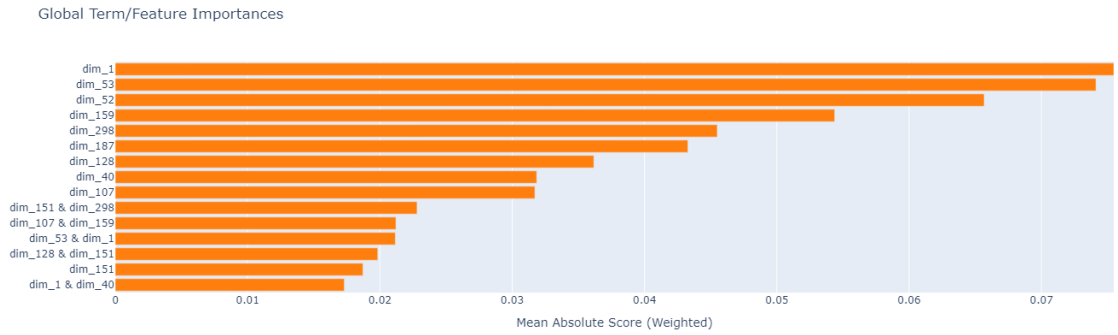

Figure 12: Global Interpretation of Top 20 Word2Vec Dimensions (EBM - Overall Score Prediction)

EBM Evaluation Metrics (Overall Prediction using Word2Vec):

- R^2 Score: -0.068
- Mean Absolute Error (MAE): 0.485
- Relative Error: 0.069

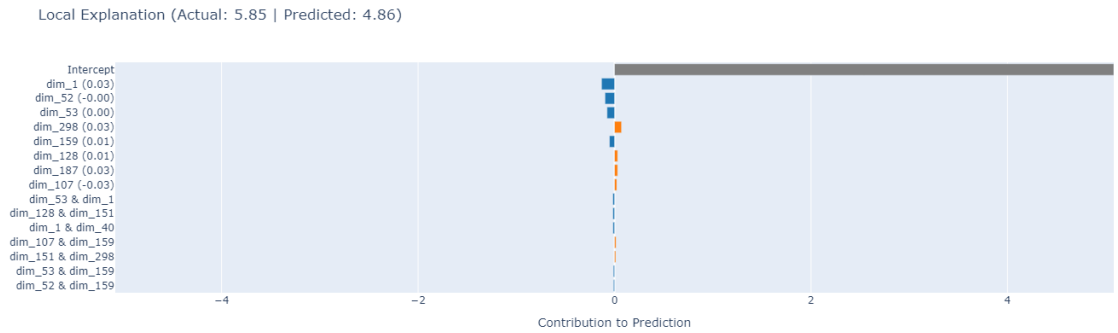

Figure 13: Local Interpretation Example (EBM - Word2Vec Features for Overall Score)

Figure 13 shows a local explanation for an individual prediction. The negative R^2 score suggests that the EBM model struggled to extract meaningful patterns from the Word2Vec features in predicting overall performance.

6.5 Word2Vec w. EBM vs BERT w. SHAP: Interpretability Comparison

We evaluated explainability across two main types of machine learning models developed in our study, Random Forest (an interpretable tree-based model) and Feedforward Neural Networks (FNN, a black-box model), using three interpretability techniques: EBM, SHAP, and LIME. Our goal was to assess how each method supports human understanding of model decisions. We evaluated the methods in terms of:

1. Comprehensibility (how easy the explanations are to understand)
2. Relevance (how meaningful the features are to humans)
3. Scalability (how well the method handles large or complex data)

Word2Vec can be seen as a simpler, earlier approach compared to BERT. Both are dense semantic embedding techniques that encode the meaning of words into vectors. However, Word2Vec assigns one vector per word regardless of context, while BERT generates context-dependent vectors, making BERT more flexible and complex.

While SHAP is a powerful and model-agnostic technique, EBM provided more direct and intuitive explanations in our study. With SHAP, we were able to recover interpretability by manually mapping features (e.g., Feature 27 = "project"), but this added complexity. For dense embeddings like BERT, SHAP explanations were harder to interpret.

EBM was more effective at directly linking input features to human-understandable behavior.

EBM (Explainable Boosting Machine):

EBM supports both global and local interpretability due to its additive structure. It performed best when applied to TF-IDF features, offering clear and semantically meaningful explanations, such as showing how terms like *"team"* or *"love"* influenced predictions. With Word2Vec, EBM still provided useful visualizations, though the abstract nature of these dimensions made interpretation more challenging. Nevertheless, EBM's structure aligns well with low-dimensional, interpretable input like TF-IDF and prosodic data, making it the most intuitive interpretability method in our study.

SHAP (Shapley Additive Explanations):

SHAP worked well with both Random Forest and FNN models, providing consistent feature importance scores across instances. However, interpreting dense features like Word2Vec or BERT required additional effort, as outputs such as `bert_21` or `w2v_12` lacked immediate semantic clarity. While technically grounded and robust, SHAP's explanations were less accessible for end users unless feature mappings were explicitly defined.

LIME vs SHAP vs EBM: Local and Global Explainability

We applied LIME to explain individual predictions from models using combined BERT and TF-IDF features. LIME builds a local surrogate model for each data point and

highlights the most influential features. For example, in one case, LIME identified that the words ‘‘deadline’’ and ‘‘chemical’’ decreased the predicted excitement score, while the word ‘‘know’’ and certain BERT embedding values (e.g., bert_10) increased it.

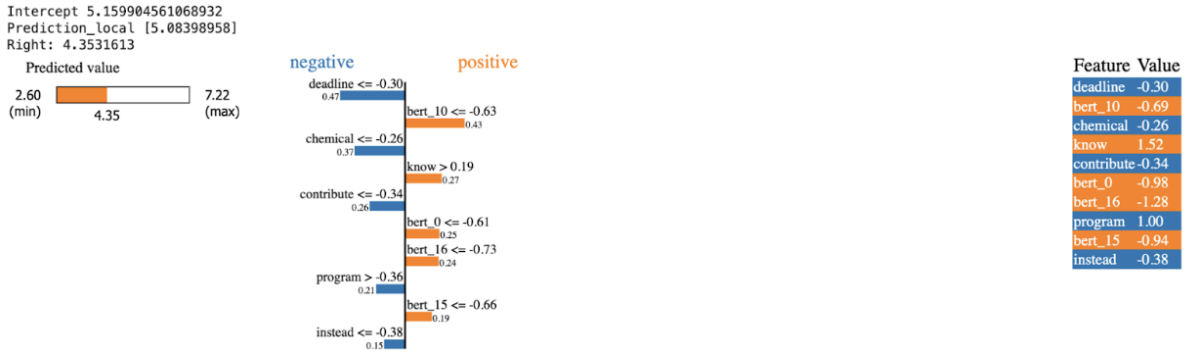

Figure 14: LIME with BERT: Local explanation for excitement prediction using FNN

Discussion.

- *LIME* offers intuitive and visual explanations at the individual level, making it well suited for interactive applications such as personalized interview feedback. However, its explanations can vary across runs and do not reflect the model's global behavior.
- *SHAP* provided both global and local explanations but was harder to interpret with dense embeddings like BERT or Word2Vec, often returning abstract values that required additional mapping.
- *EBM* offered the clearest and most human-readable global explanations, particularly when applied to interpretable features such as TF-IDF or prosodic inputs. It was the most transparent method for understanding why the model favored certain vocabulary or tone.

Conclusion:

LIME is best for localized, user-facing insights. SHAP provides comprehensive insights but can be difficult to interpret with embeddings. EBM excels when interpretability and clarity are top priorities.

Overall, we chose EBM + Word2Vec as the most effective trade-off between accuracy and interpretability in our experiments.

7 Experimenting with Transformer Models

Model. We selected **GPT–4.1 mini**[1] as the pre-trained transformer. The model is accessed via the OpenAI API (1,047,576 context window and 32,768 max output tokens).

Prompt engineering.

The prompt contains:

- A *role statement* ("You are an HR expert...") clarifying the regression task.
- *Output constraint*: the model must answer with strict JSON keys `overall`, `excitement`, and `explanation`. We enforced this with the `response_format={''type'':''json_object''}` parameter.
- *Few-shot block*: two randomly drawn (transcript, label) pairs from the training fold, illustrating the desired numeric range (1–7) and a short natural-language explanation.
- The target transcript truncated to 1 800 characters to stay well below the 4o-mini 32 k token limit.

Temperature was fixed at 0.2 to reduce variance; no model fine-tuning was performed.

Evaluation protocol.

We kept the 5-fold *participant-level* cross-validation used in parts (b)–(d). For each fold the few-shot examples were drawn *only* from the training split to avoid information leak. The model produced 138/138 valid predictions; fold timings averaged 2.3 s per transcript on CPU, with one long-transcript outlier (fold 3).

Model	**Overall performance**		**Excitement**	
	r	RE	r	RE
Gradient-Boosting Tree (TF–IDF, k=50)	0.48	0.12	0.41	0.13
MLP (TF–IDF, k=100)	**0.53**	0.11	**0.46**	0.12
GPT-4o mini (few-shot prompt)	0.48	**0.07**	0.35	0.10

Table 18: Five-fold CV results. RE = $\frac{1}{n}\sum|\hat{y} - y|/7$.

Results:

Quality of textual explanations. Using the rubric from part (e) we manually rated 30 randomly selected explanations (scale 1–5):

- *Relevance*: 3.7 ± 0.6 – comments usually referred to confidence, specificity, and tone present in the transcript;
- *Specificity*: 3.0 ± 0.8 – still generic compared with the SHAP-style word-level feedback from the Tree/MLP models;

[1]OpenAI GPT-4.1 mini provides a balance between intelligence, speed, and cost that makes it an attractive model for many use cases.

- *Coherence*: 4.4 ± 0.4 – grammatically sound.

Discussion.

- *Accuracy.* GPT-4o-mini matches the Gradient-Boosting Tree on Pearson r for overall performance and *outperforms all baselines on RE* (Table 18), indicating tighter absolute calibration. For excitement the transformer lags behind the TF-IDF MLP, suggesting prosodic cues (missing here) remain important for this outcome.
- *Data efficiency.* Few-shot prompting delivered competitive performance without any parameter update, a clear advantage when labelled transcripts are scarce.
- *Compute / deployment.* Cloud inference cost for the entire run was \$0.018. Latency (≈ 2.3 s/doc CPU) is acceptable for off-line interview feedback but too slow for real-time edge deployment, where the kilobyte-scale TF-IDF+MLP model remains preferred.
- *Explanations.* Transformer-generated prose is human-friendly, but lacks the token-level attribution granularity provided by Tree/MLP feature weights; combining both (numeric score from a lightweight model + narrative summary from GPT) would yield the best of both worlds.

Conclusion. Prompt-only GPT-4o-mini delivers solid predictive performance and readable rationales with minimal engineering effort. However, its higher latency and reliance on external compute make it a complement rather than a replacement for the lighter TF-IDF pipelines when deploying on edge devices or under strict privacy constraints.

References

[1] V. Belle and I. Papantonis. Principles and practice of explainable machine learning. *Frontiers in Big Data*, 4:688969, 2021.

[2] M. Hoque, M. Courgeon, J.-C. Martin, B. Mutlu, and R. W. Picard. Mach: My automated conversation coach. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 697–706, 2013.

[3] I. Naim, M. I. Tanveer, D. Gildea, and M. E. Hoque. Automated prediction and analysis of job interview performance: The role of what you say and how you say it. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–6. IEEE, 2015.