What Are the FinBERT Variables (`finbert\_0`, `finbert\_1`, ..., `finbert\_767`)?

~These are the 768-dimensional embedding vector output by the FinBERT model for each article.

Each article is converted into a numeric summary — a point in 768-dimensional space — where:

- Each of the `finbert\_i` values (for `i = 0 to 767`) captures a different latent feature or semantic signal in the article.

- They are not directly human-interpretable, but they contain rich information learned during pretraining on financial news.

---

They are not statistically independent.

- They are the output of a deep neural network, and the dimensions often have correlations between them.

- However, they each represent different learned patterns, such as:

- Tone (positive/negative)

- Financial uncertainty

- Company or market mentions

- Speculation, hedging language

- Action verbs, sentiment-bearing phrases

So while they aren’t independent, they are jointly useful in predictive modeling.

🔬 What Happens Inside FinBERT?

1. Tokenize the article

2. Embed each token into a vector

3. Pass through 12 Transformer layers

4. Extract the [CLS] token embedding → a 768-length vector that summarizes the entire article

That's what we're saving as `finbert\_0` to `finbert\_767`.

📈 What Are They Used For?

✅ 1. Features for Machine Learning Models

You can use them as predictor variables (features) in models like:

- Logistic regression → to classify market up/down reactions

- XGBoost or Random Forest → to estimate stock return magnitude

- LSTM/Transformer → if modeling sequences of news

✅ 2. Semantic Similarity

- Articles with similar embeddings are often \*\*topically or sentiment-wise similar\*\*

- You can use clustering (k-means) or t-SNE/PCA to explore this visually

✅ 3. Time Series Impact Modeling

- Use the embeddings over time (sorted by `Timestamp`) to study how \*\*news tone affects market performance

**📈 What Can We Do With These FinBERT Embeddings?**

Here’s a breakdown of **specific tasks** and how to use them:

**1. ✅ Stock Movement Prediction**

Task: Will the market or a specific stock go up or down after this article?

* Input: finbert\_0 to finbert\_767
* Output: Binary label (up/down)
* Model: Logistic regression, Random Forest, XGBoost, LSTM, Transformer

**2. ✅ Stock Return Regression**

Task: Predict how much the stock will go up/down (% change)

* Input: FinBERT embeddings
* Output: Continuous value (e.g., +1.5%)
* Model: Linear regression, Ridge/Lasso, XGBoost, etc.

**3. ✅ Market Volatility Forecasting**

Task: Will volatility rise after this news?

* Output: Volatility label or value (high/low or numeric)
* Useful in risk forecasting, portfolio strategy

**4. ✅ Clustering / Topic Discovery**

Task: Group similar articles together

* Use: k-means or t-SNE/PCA on FinBERT embeddings
* Benefit: Find themes (e.g., “tech optimism”, “trade war fear”)

**5. ✅ Semantic Search / Recommendation**

Task: “Find articles similar to this one”

* Use cosine similarity between FinBERT embeddings
* Can also build document recommender or timeline analysis

**6. ✅ Sentiment or Tone Analysis (Fine-tune or Supervised)**

Task: Train on your own labeled news (e.g., bullish/bearish, policy-favoring)

* Fine-tune a classifier using FinBERT + your labels

Q:

**🔄 Should We Average the 768 FinBERT Traits?**

**✅ You already are using an average — sort of.**

* In your code, you're using the **[CLS] token embedding**, which is a **single 768-dimensional vector**.
* This is **not an average of all token embeddings**, but a **summary vector** that BERT/FinBERT learns specifically to capture **the whole input's meaning**.

📌 **So: You should NOT manually average the 768 values**.

* They are already structured and ordered for maximum effectiveness.
* Manually averaging them loses information and makes them meaningless.

**🔍 Are These 768 Traits Independent?**

**❌ No, they are not independent.**

* These values come from a **deep neural network**, so they are **interdependent**.
* Think of them like **correlated brain activations**:
  + One might spike when the article talks about trade war threats.
  + Another might spike when optimism appears.
  + Some traits interact or **only light up in context**.

In practice:

* You can use **dimensionality reduction (PCA)** or **feature selection (LASSO)** if you want to decorrelate or reduce the size.
* But most ML models (e.g., tree-based or neural networks) work well with all 768 as-is.