1. How does Textblob calculate the Sentiment Polarity Score?

polarity = (Σ word\_scores) / n

Where:

* word\_scores = individual sentiment values (-1 to +1)
* n = number of sentiment-bearing words

The [**sentiment**](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.sentiment) property returns a namedtuple of the form Sentiment(polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0].

1. Subjectivity Score:

subjectivity = n\_sentiment\_words / n\_total\_words

The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

1. Intensifier Rule:

modified\_score = base\_score \* (1 + intensifier\_value)

Example: "very good"

* "good" base\_score = 0.7
* "very" intensifier\_value = 0.1
* modified\_score = 0.7 \* (1 + 0.1) = 0.77

1. Negation Rule:

negated\_score = -1 \* base\_score

Example: "not good"

* "good" base\_score = 0.7
* negated\_score = -1 \* 0.7 = -0.7

1. For compound phrases, scores are averaged:

phrase\_score = (Σ modified\_word\_scores) / n\_scored\_words

These calculations use Pattern's sentiment lexicon for the base word scores. The final sentiment is normalized to stay within [-1, 1] for polarity and [0, 1] for subjectivity.

Now that we understand the pieces:

polarity = (Σ(wi \* fi)) / (Σ|fi|)

where:

* wi = sentiment weight of word i (-1 to +1)
* fi = word modifying factor (intensifiers/negations)
* Σ|fi| = sum of absolute values of modifying factors

subjectivity = (Σ(si \* |wi|)) / (n \* max(|wi|))

where:

* si = subjectivity score of word i (0 to 1)
* |wi| = absolute value of word's sentiment weight
* n = number of scored words
* max(|wi|) = maximum absolute sentiment weight

NLTK's VADER (Valence Aware Dictionary and sEntiment Reasoner) calculates sentiment through these key steps:

1. Word-level scoring: Each word gets a sentiment intensity score from VADER's lexicon
2. Rule-based adjustments:
   * Punctuation (!!!) intensifies sentiment
   * Capitalization (GREAT vs great) increases intensity
   * Degree modifiers (very good vs good) affect scores
   * Negation handling ("not good")
   * But/contrast handling affects preceding scores
3. Normalization: Combines scores into:
   * compound: Normalized sum (-1 to +1)
   * pos/neg/neu: Proportions of text matching each sentiment

VALENCE – multi step process – not overly clear -

For BERT Models:

finBERT