\subsubsection{Bag of Words (BoW) Approach}

The Bag of Words (BoW) representation using CountVectorizer \footnote{\url{https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html}}. This process tokenizes the text data and constructs a vocabulary containing all unique words present in the corpus. The text data is then transformed into a matrix where each row represents a document, and each column corresponds to a word in the vocabulary. The value in each cell of the matrix indicates the number of times the word appears in the corresponding document.

The BoW representation generated doesn't take into account the order of words, and it merely focuses on their presence and frequency. This method captures the word distribution in the documents but might give equal importance to common and rare words. %1,481 words

\begin{figure}[ht]

\centering

\includegraphics[width=5in]{Figures/BoW Diagram.png}

\caption{Example of a Training Tweet in Bag of Words Representation}

\label{fig:gridSearch}

\end{figure}

\subsubsection{Term Frequency-Inverse Document Frequency (TF-IDF)}

The next step in the pipeline is to convert the BoW representation into a TF-IDF representation using TfidfTransformer \footnote{\url{https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfTransformer.html}}. The TfidfTransformer in the pipeline is responsible for transforming the BoW matrix, into a TF-IDF respresentation.

The rationale behind using TF-IDF with the BoW matrix is to emphasize the importance of words not just based on their frequency within a single document, but also in relation to their occurrence across the entire document set.

\begin{table}[ht]

\centering

\renewcommand{\arraystretch}{1.5} % Add extra vertical padding to rows

\begin{tabular}{|p{3cm}|p{10cm}|}

\hline

\textbf{Metric} & \textbf{Description} \\

\hline

Term Frequency (TF) & The raw frequency of a word in a document is normalized by dividing it by the total number of words in that document. This adjustment accounts for varying document lengths and provides a proportional measure of each word's contribution to the document. \\

\hline

Inverse Document Frequency (IDF) & This metric aims to quantify the significance of a word in the entire corpus. By calculating the logarithm of the total number of documents divided by the number of documents containing the word, common words that appear in many documents are assigned lower weights, while unique and context-specific words are assigned higher weights. \\

\hline

\end{tabular}

\caption{Description of Term Frequency and Inverse Document Frequency}

\label{table:tf\_idf}

\end{table}

The final TF-IDF representation is computed by multiplying the term frequency (TF) with the inverse document frequency (IDF) for each word in the BoW matrix.

The approach of combining Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) in a sentiment analysis pipeline is particularly useful for analyzing football-related tweets, as it provides a robust and effective method to represent the domain-specific text data for classification tasks. By initially using Bag of Words, the model captures the frequency distribution of words, which is essential in identifying football-related terms and expressions. Building upon this foundation, the pipeline applies TfidfTransformer to transform the raw term frequency matrix into a TF-IDF representation, which assigns higher weights to words that are more specific and relevant to football discussions, while downplaying the importance of common words that occur frequently across documents. This combination ensures that the model captures not only the word frequency information but also the significance of each football-related word within the corpus. As a result, the Naive Bayes classifier benefits from a richer feature set that better represents the underlying football-related text data, potentially leading to improved classification performance in sentiment analysis tasks. %1,847 words

\subsubsection{Multinomial Naive Bayes Classifier}

The Multinomial Naive Bayes classifier \footnote{\url{https://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html}} is used as the final step in the text classification pipeline to classify tweets as positive, negative or neutral. he Multinomial Naive Bayes algorithm is a variant of the Naive Bayes classifier, specifically designed to handle discrete data, such as word frequencies in text documents. It is based on the Bayes theorem, which calculates the probability of a given class (in this case, sentiment) given the observed features (word frequencies). It makes the assumption that the features (words) are conditionally independent given the class, which simplifies the computation and allows the model to scale well with large datasets.

The employment of the Multinomial Naive Bayes classifier, as opposed to the Bernoulli Naive Bayes classifier, is justified by the nature of the text data. This classifier is capable of capitalizing on the wealth of information furnished by word frequencies, potentially culminating in superior classification performance in comparison to a Bernoulli Naive Bayes classifier, which solely takes into account the presence or absence of words. Empirical evidence from numerous studies has demonstrated that the Multinomial Naive Bayes classifier outperforms its Bernoulli counterpart. Furthermore, prior research delineated within the literature review corroborates the utilization of the Multinomial Naive Bayes classifier in the given context \cite{nbmeth2, nbimp}.

In the training phase, the Multinomial Naive Bayes classifier comprehends the patterns and interconnections amongst the words in tweets and their corresponding sentiment labels (positive, negative, or neutral). Subsequently, when the classifier is provided with a new, untagged tweet, it ascertains the probabilities of the tweet being assigned to each sentiment label based on the word frequencies in the tweet. Consequently, the predicted sentiment of the tweet is assigned based on the sentiment label with the highest probability. The specific decision rules and thresholds for designating a sentiment label may differ depending on the specific execution of the classifier and the task at hand. %2,180 words