Document Classification on Patient Discharge Summaries

Simrun Sharma (SS1486) | Titus Robin Arun(TRA29)

Abstract

In this report, we delve into leveraging Natural Language Processing (NLP) to categorize medical discharge summaries by smoking status: current smokers, non-smokers, past smokers, and unknown.We applied two distinct methodologies to harness the capabilities of NLP aiming to enhance medical document classification and contribute to advancements in medical NLP.

The first, Naive Bayes, a generative probabilistic model, is chosen for its effectiveness in textual classification by capitalizing on the frequency of terms related to smoking. This model proved adept at providing a quick and efficient baseline, achieving an 80% accuracy in classifying smoking status, and demonstrating its robustness with an F1 score of 0.8031 on unseen test data. Our second approach utilized BERT Clinical, a state-of-the-art model in understanding the nuances of medical language. We fine-tuned this model on both real-world and synthetic data. Our BERT model showcased a significant improvement on real data, with initial accuracy from 61.2% to a remarkable 98.5% after training. However, it faced challenges with the synthetic data, indicated by a testing F1 score of 0.25, revealing the model's limitations with data lacking real-world complexity.

This stark contrast in performance between real and synthetic data underscores the sophistication of BERT in grasping intricate language patterns over simplistic, repetitive ones. Our findings illuminate the synergy between classical and contemporary NLP techniques, where Naive Bayes offers a quick analytical baseline, while BERT Clinical provides deep contextual understanding. This powerful amalgamation has significant potential in medical document classification, confirming the effectiveness of these models in discerning smoking status and paving the way for further advancements in medical NLP.

Dataset

This dataset sourced within i2b2 NLP data sets addresses the challenge of identifying patient smoking status from medical discharge records. Clinical narratives, often in fragmented English free text, pose significant challenges for linguistic processing and retrieval. Traditional natural language processing tools are ill-suited for such text, hindering progress in medical language processing (MLP) technologies.

To overcome these obstacles, the authors de-identified and released clinical records within the i2b2 project, enabling the development of ground truth for the smoking challenge. The smoking challenge aimed to predict patients' smoking status from medical discharge records explicitly, excluding implicitly revealed information. The data, sourced from Partners HealthCare, underwent preprocessing and annotation by pulmonologists, resulting in five smoking status categories: Past Smoker, Current Smoker, Smoker (insufficient information), Non-Smoker, and Unknown. The challenge saw participation from MLP community representatives, with results discussed at a workshop co-sponsored by the American Medical Informatics Association.

Preprocessing

In the preprocessing phase of our project focusing on the text classification of a patient's smoking status from medical records, we implemented several crucial steps to enhance data quality and address imbalances:

- *Label Encoding*: Employed the LabelEncoder from scikit-learn to encode the "Smoking Status" variable into numerical labels, facilitating model training.
- *Text Cleaning:* Converted all text data to lowercase and utilized a regular expression to extract pertinent information related to smoking from the "Text" column. The regex pattern retrieved lines from patient summaries containing terms such as "smok," "tobacco," "cigar," "pack," or "ppd," along with 5 words before and after the identified term. If the regex expression wasn't found, the entire patient summary was retained in the dataframe.
- *Class Modification:* Merged the "SMOKER" class into the "PAST SMOKER" class to address class imbalance, resulting in four distinct classes: non-smoker, past smoker, current smoker, and unknown.
- *Data Combination and Model Evaluation Strategy:* Initially we decided to utilize a downsampling method however ultimately forgoed the idea to instead combine the classes instead. This strategic combination aimed to maintain as much valuable information as possible in the relatively small dataset of 398 rows. The decision to address class imbalance was integrated into the model evaluation strategy, utilizing a weighted F1 score. This approach proved more efficient than downsampling, as it effectively optimized performance without sacrificing significant portions of the dataset.

Generative Probabilistic Model: Naive Bayes

A. Model Overview

Naive Bayes is a generative probabilistic model utilized in machine learning for classification tasks, especially in text classification. $p(y|X)$, Naive Bayes operates as a generative model, aiming to model the joint probability distribution. $p(X,y)$ and then using Bayes' theorem to calculate the posterior probability $p(y|X)$. The model relies on the assumption of strong independence between features given the class label.

In the context of text classification for smoking status, Naive Bayes is chosen for several reasons:

- Multinomial Model: The Multinomial Naïve Bayes model is particularly effective for classifying data that cannot be represented numerically, such as text. It simplifies textual data classification by analyzing the frequency of terms (words) in documents.
- Reduced Complexity: One of its main advantages is significantly reduced complexity, making it feasible to perform classification with small training sets. This is beneficial for scenarios where continuous re-training is not practical.

- Frequency of Terms in Texts: Smoking status classification may depend on the frequency of specific terms or words related to smoking or health in the given text.

The Multinomial model, which considers the frequency of terms, is effective in capturing such information.

B. Model Architecture

1. Data Preprocessing:
    - Load the dataset containing patient records, including discharge summaries and smoking status labels.
    - Combine "SMOKER" class with "PAST SMOKER" for better class representation.
    - Use LabelEncoder to encode the smoking status labels into numerical values.
    - Lowercase and clean the text data by applying a regular expression to extract relevant information related to smoking.
2. Feature Extraction - Bag of Words/ Synthetic Data Generation:
    - Implement CountVectorizer to transform the cleaned text data into a bag-of-words representation.
    - Extract the most common words from the dataset and create a frequency matrix. The extracted probabilities were used to determine word choice for synthetic data generation.
3. Model Training:
    - Split the data into training and testing sets.
    - Fit a Multinomial Naive Bayes classifier using CountVectorizer and TF-IDF vectorized data separately. Determine which vectorizer performed the best evaluated metric.
    - Evaluate the model using cross-validation, displaying scores such as accuracy and F1 score.
    - Train and Tested a Naive Bayes model on synthetic data generated using word probabilities from the original data.
4. Model Performance Visualization:
    - Visualize the confusion matrix and F1 score for both the original and synthetic data models.
    - Utilize seaborn and matplotlib for visualizations

C. Model Optimization

- **CountVectorizer Implementation:** The use of CountVectorizer was a foundational step in the optimization process. It allowed the conversion of raw text data into a bag-of-words representation, enabling the model to analyze term frequencies and gain insights into the linguistic characteristics of the dataset.
- **Parameter Tuning:** The alpha parameter in the Multinomial Naive Bayes classifier underwent careful tuning to enhance model performance. Employing a grid search, the optimal alpha value was identified, maximizing the accuracy of the Naive Bayes model.
- **Comprehensive Preprocessing Pipeline:** A robust preprocessing pipeline, including techniques like tokenization, removal of stop words, and stemming, contributed to the overall optimization of the Naive Bayes model. This ensured that the input data was appropriately transformed to capture relevant linguistic patterns.

Discriminative Neural Network: Pre-trained BERTClinical

A. Model Overview

BERT (Bidirectional Encoder Representations from Transformers) Clinical, a variant of the BERT model fine-tuned for clinical texts, was chosen for its advanced capabilities in processing complex medical language. This model, based on the architecture of BERT, has been specifically adapted for the nuances of clinical language, making it ideal for tasks like classifying medical discharge summaries. Its effectiveness comes from its ability to understand context and relationships within medical texts, a crucial aspect for accurate classification.

BERT, in general, revolutionized NLP by introducing a new way of handling language understanding tasks. Unlike traditional models that process text in one direction (either left-to-right or right-to-left), BERT processes text bidirectionally. This means it takes into account the context from both sides of a token (word) in the text. As a result, it gains a deeper understanding of language context and nuances, making it powerful for various NLP tasks.

B. Model Architecture

BERT's architecture is based on the transformer model, which relies on attention mechanisms to weigh the significance of different words in a sentence. This architecture consists of several layers of these transformers. Each layer captures different aspects of language, from individual word meanings to the overall sentence structure.

The key components of BERT's architecture include:

- Token embeddings: BERT tokenizes input text into tokens and assigns an embedding to each token.
- Segment embeddings: These distinguish between different sentences in the input.
- Positional embeddings: Since transformers do not inherently capture the order of tokens, positional embeddings encode the position of each token in the sequence.
- Transformer blocks: Each block processes the input tokens in parallel, applying attention mechanisms and neural network operations.
- Pre-trained and fine-tuning stages: BERT is first pre-trained on a large corpus of text, and then fine-tuned for specific tasks, like sequence classification.

For our project, the BERT Clinical model was trained and evaluated on both real-world and synthetic data. The training process involved several steps:

1. Data Preparation: The training data was read from CSV files and pre-processed. Labels were encoded using a LabelEncoder, and the text data was tokenized using BERT's tokenizer.
2. Tokenization and Embeddings: The tokenized data was then converted into embeddings. Each token was represented by an ID, and sentences were padded to ensure a consistent length.
3. Attention Masks: These masks were created to distinguish between actual tokens and padding in the input.

4. DataLoaders Creation: The processed data was wrapped in a TensorDataset and loaded using DataLoader for efficient batch processing during training.
5. Model Initialization: We utilized the BertForSequenceClassification model pre-trained on clinical data, specifying the number of expected labels.
6. Training Loop: The model was trained over multiple epochs, with each epoch involving a forward pass, loss calculation, and a backward pass for optimization. Performance metrics like accuracy, precision, recall, and F1 score were calculated.
7. Optimizer and Scheduler: The AdamW optimizer was used for better handling of weight decay. A linear scheduler with warmup was employed to adjust the learning rate across epochs.
8. Evaluation: The model was evaluated on a validation set after each epoch to monitor its performance on unseen data.
9. Saving the Model: After training, the model's state was saved for future use or additional analysis.

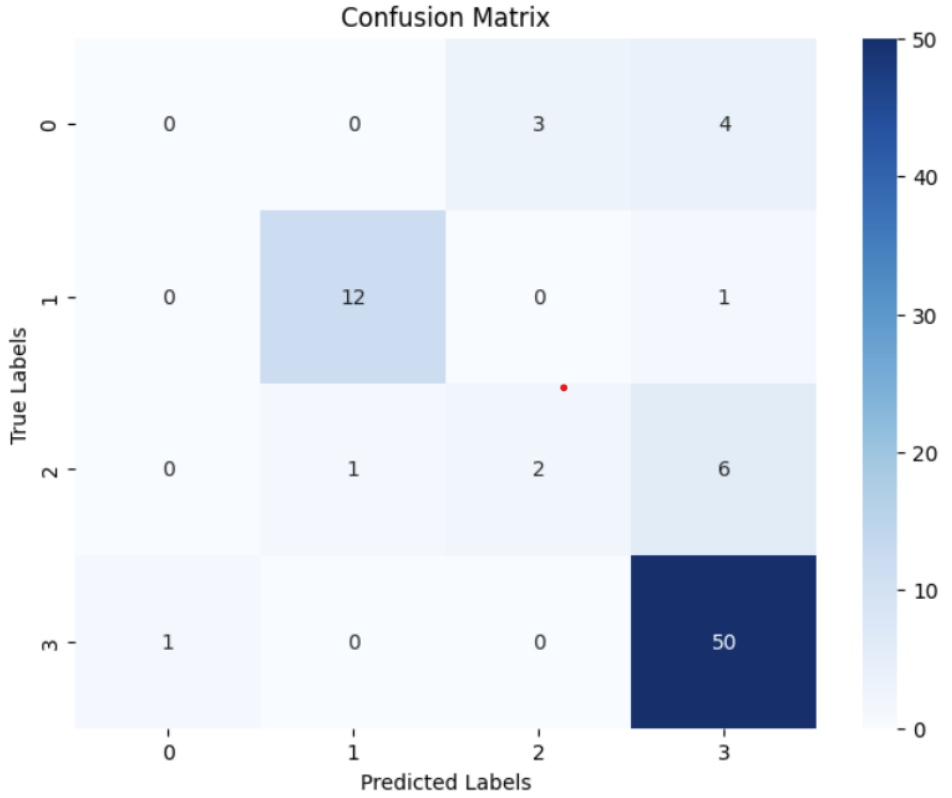
This training and evaluation process aimed to optimize the model's ability to accurately classify text into the specified smoking status categories, leveraging the strengths of BERT Clinical in understanding and processing complex medical texts.

C. Model Optimization

1. **Learning Rate Scheduler**: We employed the ***get_linear_schedule_with_warmup*** scheduler for the learning rate. Initially, the learning rate started low and warmed up over 70 steps (10% of our total steps of 700), before decreasing linearly. This approach was critical in stabilizing the model's early training phase, preventing premature convergence to suboptimal solutions.
2. **Full Model Tuning on a Small Dataset**: Our dataset's relatively small size necessitated tuning the entire BERT Clinical model rather than just the top layers. This approach allowed all layers of the model to adapt to our specific dataset, enhancing the learning potential from the available data.
3. **Zero Gradient Reset**: The use of ***zero_grad()*** at the beginning of each batch was vital for resetting gradients. This ensured that gradients from previous batches didn't accumulate, allowing for independent and correct weight updates for each batch.
4. **Batch Processing and DataLoaders**: Batch processing was implemented using DataLoader with a batch size of 16. This not only optimized memory usage but also contributed to more effective training by updating model weights after processing multiple samples rather than individually.
5. **AdamW Optimizer**: The AdamW optimizer, with a learning rate of 2e-5 and ***correct_bias set to False***, was chosen for its improved handling of weight decay. This optimizer choice was pivotal in preventing overfitting, especially crucial in our small dataset scenario.
6. **Performance Monitoring**: During training, we monitored accuracy, precision, recall, and F1 score. For instance, in each epoch, we calculated these metrics for the batches, which informed us about the model's learning trajectory and allowed for timely adjustments.

Model Evaluation and Metrics

Generative Probabilistic Model: Naive Bayes :Training Naive Bayes Model



	fit_time	score_time	test_score	train_score
0	0.002709	0.000859	0.828125	0.933071
1	0.003213	0.000915	0.812500	0.940945
2	0.002641	0.000842	0.828125	0.925197
3	0.002555	0.000851	0.809524	0.937255
4	0.002448	0.000842	0.873016	0.909804

Accuracy: 0.8

F1 Score: 0.7513392857142857

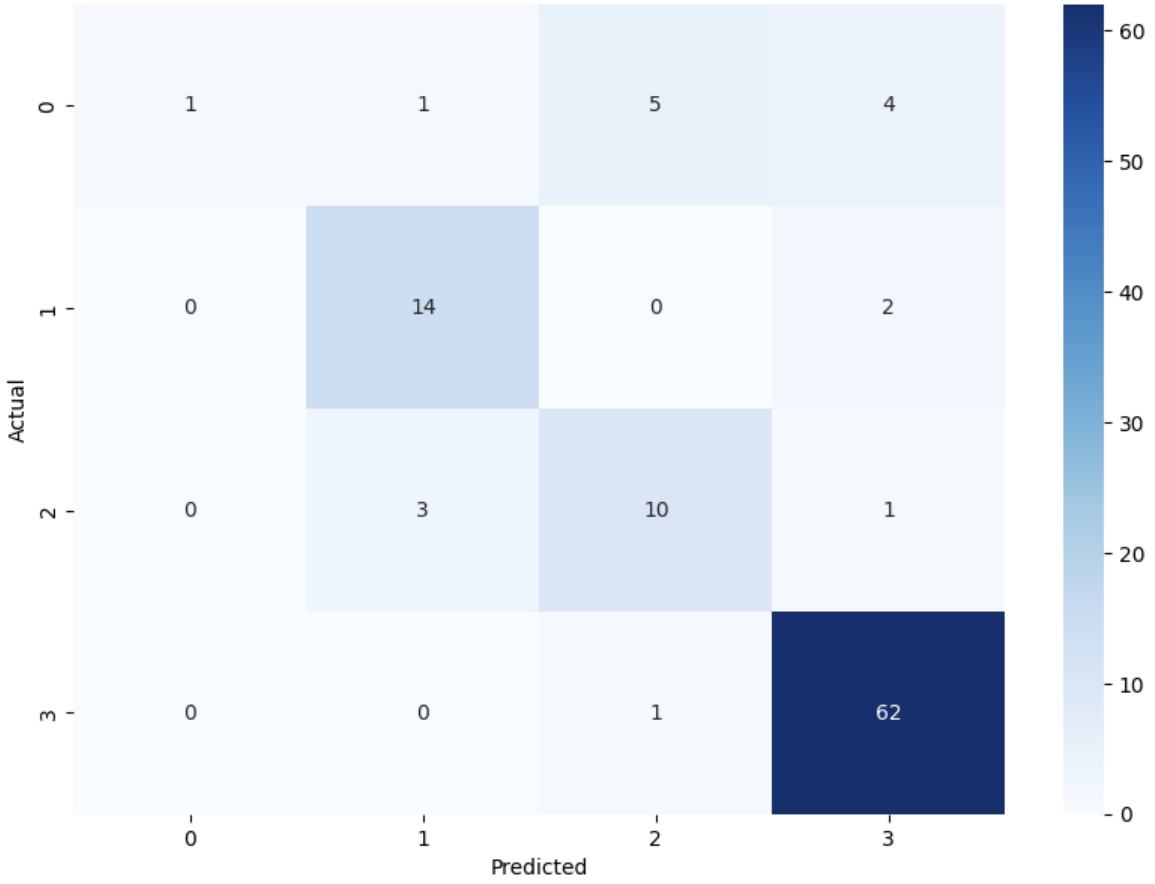
The accuracy score of 80% reflects the overall correctness of the model's predictions across all classes. It is calculated as the ratio of correctly classified instances to the total instances in the test set. In this context, an 80% accuracy implies that 80% of the instances were accurately classified by the model.

The F1 score considers both the precision, representing the accuracy of positive predictions, and recall, measuring the ability of the model to capture all positive

instances. For this project, a higher F1 score indicates that the model not only makes accurate predictions but also effectively identifies instances of current smokers, non-smokers, and past smokers while minimizing both false positives and false negatives. This balance is particularly important in healthcare-related applications like smoking status classification, where misclassifying an individual's smoking status could have significant implications for subsequent health interventions.

Generative Probabilistic Model: Naive Bayes: Testing Naive Bayes Model

```
F1 Test Score: 0.8031245715069245
Accuracy Test Score: 0.8365384615384616
```



The assessment metrics applied to the testing data underscore the Naive Bayes model's commendable performance in predicting patient smoking statuses, illustrating its robustness across diverse datasets. The slightly elevated Accuracy Test Score of 0.8365 compared to the training accuracy suggests the model's sustained effectiveness when encountering previously unseen patient discharge summaries. This result highlights the model's capacity for generalization and its adaptability to real-world scenarios. Moreover, the consistent F1 Test Score of 0.8031 indicates the model's ability to maintain a well-balanced precision and recall on the test set, mirroring its performance during training.

In the training Confusion Matrix, the model demonstrated proficiency in correctly classifying current smokers (class 0) and non-smokers (class 1) with minimal misclassifications. However, when applied to the testing data, the model faced additional challenges,

particularly in distinguishing between past smokers (class 2) and unknown smoking status (class 3). The testing Confusion Matrix indicates an increased number of false positives and false negatives in these categories compared to the training phase. This discrepancy suggests that the model encounters nuances in patient discharge summaries during testing, requiring further refinement to enhance its ability to accurately predict past smoking status and classify instances where smoking status is unknown. Despite these challenges, the model maintains a commendable overall accuracy and F1 score on the testing data, emphasizing its adaptability and potential for robust performance across diverse patient populations.

Discriminative Neural Network: BERTClinical: Training BERT Model (Real Data)

The training loss steadily decreases throughout the BERT Clinical model's training process, indicating effective learning and adaptation to the training data. However, the validation loss does not follow a similar trend, remaining high and fluctuating. This discrepancy raises concerns about the model's ability to generalize to new data, possibly indicating overfitting or the need for optimization strategy adjustments. It emphasizes the importance of not only tracking loss metrics but also enhancing the model's generalization capabilities.

In summary, while the training loss demonstrates effective learning, the persistent high validation loss highlights potential challenges in model generalization, warranting further investigation into optimization strategies and dataset representation.

Discriminative Neural Network: BERTClinical: Testing BERT Model (Real Data)

At the conclusion of the testing process, the model achieves a Precision of 0.059, Recall of 0.167, and an F1 Score of 0.087. These metrics suggest that the model's performance on

the testing data is modest, indicating room for improvement in accurately categorizing smoking status. It's worth noting that Precision and Recall warnings indicate challenges in predicting certain categories, emphasizing potential limitations in the model's predictive power.

Synthetic Data Generation

The creation of synthetic data using the bag-of-words implementation involved applying CountVectorizer to the original training data, extracting feature names that represented unique words in the corpus. The subsequent training of the Multinomial Naive Bayes model on this data yielded feature probabilities for each class, forming the basis for constructing a synthetic dataset. Rows in the synthetic dataframe were assigned to feature names, while columns were labeled with various smoking status classes, each cell containing the likelihood of a specific word belonging to a particular smoking status. The create_synthetic function utilized this information to generate synthetic sentences by randomly selecting words from the appropriate class column based on their probabilities, ensuring the synthetic dataset retained the probabilistic characteristics of the original training data. Following its creation, an additional Naive Bayes model was built using this synthetic data, demonstrating superior performance compared to the original real-world dataset in both training and testing scenarios.

Countvectorizer proved to be the best choice for Naive Bayes in both the synthetic dataset and real data scenarios due to compatibility with the model's underlying assumptions. Naive Bayes relies on word frequency information, and Countvectorizer efficiently transforms text data into a bag of words representation, facilitating the model's probability calculations. In contrast, TF-IDF's emphasis on rare words and document-wide importance may not align with Naive Bayes' assumption of feature independence, making Countvectorizer a more suitable option for this probabilistic classification task. The synthetic data generated by Naive Bayes was also utilized as the synthetic data for the Bert Model.

Result Comparison of Models

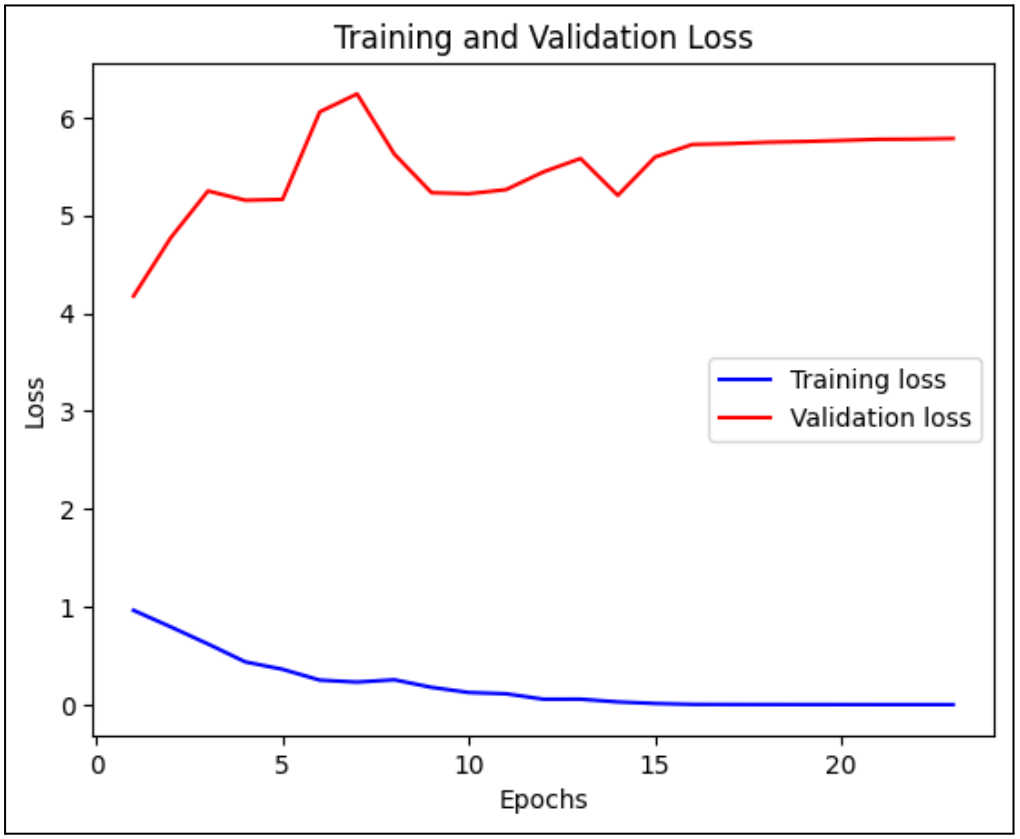
The Naive Bayes model demonstrated enhanced performance on the synthetic dataset compared to the real dataset primarily because the calculated word probabilities derived from the get_log_prob function of the model trained on the original training real dataset. Naive Bayes is fundamentally a probabilistic model that relies on the assumption of feature independence, making it particularly adept at handling synthetic data generated based on clear word boundaries delineated by log probabilities. This statistical approach aligns seamlessly with the synthetic data's inherent structure, enabling the Naive Bayes algorithm to classify accurately. The model's success with synthetic data shows its proficiency in handling diverse datasets, especially those conforming to a probabilistic framework, as it leverages the well-defined probabilities associated with word occurrences across different classes.

```
Synthetic Accuracy: 0.8125
F1 Synthetic Score: 0.8067109728506787
```

On the other hand, the BERT model exhibited superior performance on real data compared to synthetic data, owing to its distinctive bidirectional attention mechanism. This mechanism

allows BERT to consider the entire context of a word by analyzing both left and right context within a sentence. However, synthetic data often fails to capture contextual and variability seen in the English Language. While Naive Bayes relies on clear word boundaries, BERT thrives on the intricate contextual relationships among words. Synthetic datasets may lack the nuanced contextual variations inherent in real language, hindering BERT's ability to fully comprehend and interpret word context. The result is a model that may struggle to replicate the complexity and diversity of natural language, especially in the context of real-world usage with its cultural, societal, and contextual nuances.

Testing and Validation: Synthetic Data



Analyzing the training and validation loss chart for the synthetic dataset reveals the BERT Clinical model's learning trajectory. The training loss consistently decreases with epoch progression, indicating successful error minimization on the synthetic data. However, the validation loss exhibits a contrasting pattern, showing volatility and an overall increase, potentially starting near 4.0 and reaching 5.5 or higher in later epochs. This suggests that while the model adapts well to the training data, it struggles to generalize to the validation set, potentially due to overfitting or distribution mismatches. This underscores the challenges of working with synthetic data and emphasizes the importance of carefully generating and utilizing synthetic data for machine learning models, especially when aiming for real-world clinical applications.

	Real Data (F1 Score)	Synthetic Data (F1 Score)
Generative Model (Naive Bayes)	0.75	0.799
Discriminative Model (BERTClinical)	0.98	0.25

Future Work

In future iterations of this project, several avenues for improvement can be explored. Firstly, acquiring cleaner and more focused data specific to the classification task is paramount. The current dataset includes miscellaneous patient information, introducing noise that may impede accurate classification. A more refined dataset, concentrating solely on patient smoking and social history, could significantly enhance model performance. Secondly, the limited dataset size (398 rows) poses a constraint, particularly when working with models like BERT. Future efforts could prioritize gathering a more extensive dataset to optimize the performance of models reliant on substantial data volumes. Lastly, the implementation strategy can be reconsidered. While a pre-trained model suited for clinical data was chosen for this report, exploring the suitability of recurrent neural networks (RNNs) or long short-term memory (LSTM) models could be beneficial. Furthermore, given the promising performance of the generative probabilistic approach with Naive Bayes, we can induce that perhaps this classification problem with smoking status is better suited for a generative probabilistic model approach rather than the discriminative model with BERT.