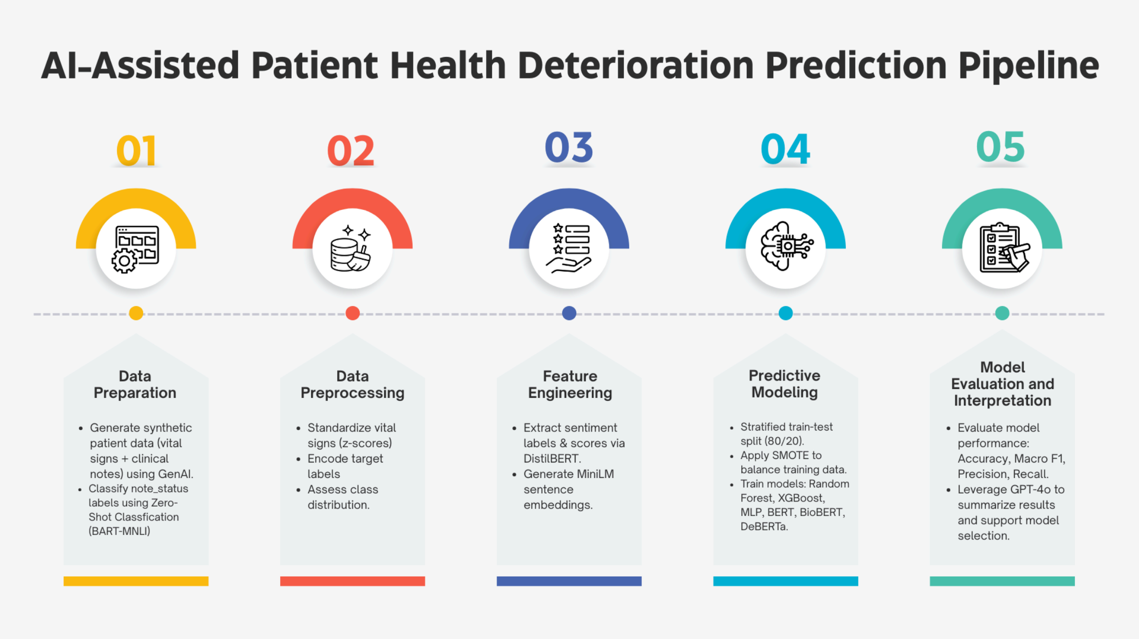
**Name:** Wong Yi Ting
**Student ID:** S2152880
**Course:** WQD7005 | Semester 2 | 2024/2025
**Title:** Comprehensive AI-Assisted Final Report
**Github Repo:** ***https://github.com/LorraineWong/WQD7005-Data-Mining-S2152880***
**Google Colab:** *Project_health_deterioration_model.ipynb*

## 1.0 Introduction

Predicting patient health deterioration is an important challenge in clinical decision-making as early identification of high-risk patients can greatly improve treatment outcomes and help allocate healthcare resources more effectively. However, traditional methods often do not fully utilize the valuable information found in unstructured clinical records and lack integration with advanced AI technologies. This project addresses this limitation by applying Large Language Models (LLMs), Small Language Models (SLMs), and Generative AI (GenAI) to create comprehensive features from vital signs and clinical text. A variety of predictive models, including traditional machine learning and Transformer-based models, are developed and evaluated to classify patient health conditions. In addition, LLMs are used to help explain model performance and improve transparency. Through this AI-assisted framework, the project aims to enhance predictive analytics for patient deterioration and support more informed and timely clinical decisions.

## 2.0 Methodology

This project adopts an AI-assisted pipeline to predict patient health deterioration based on structured vital signs and unstructured clinical notes. As shown in Figure 1, the process integrates GenAI, LLM, and SLM technologies across five key stages: Data Preparation, Data Preprocessing, Feature Engineering, Predictive Modeling, and Model Evaluation and Interpretation. Each step is described in detail below.



AI-Assisted Patient Health Deterioration Prediction Pipeline

01 Data Preparation

- Generate synthetic patient data (vital signs + clinical notes) using GenAI.
- Classify note_status labels using Zero-Shot Classfication (BART-MNLI)

02 Data Preprocessing

- Standardize vital signs (z-scores)
- Encode target labels
- Assess class distribution.

03 Feature Engineering

- Extract sentiment labels & scores via DistilBERT.
- Generate MiniLM sentence embeddings.

04 Predictive Modeling

- Stratified train-test split (80/20).
- Apply SMOTE to balance training data.
- Train models: Random Forest, XGBoost, MLP, BERT, BioBERT, DeBERTa.

05 Model Evaluation and Interpretation

- Evaluate model performance: Accuracy, Macro F1, Precision, Recall.
- Leverage GPT-4o to summarize results and support model selection.

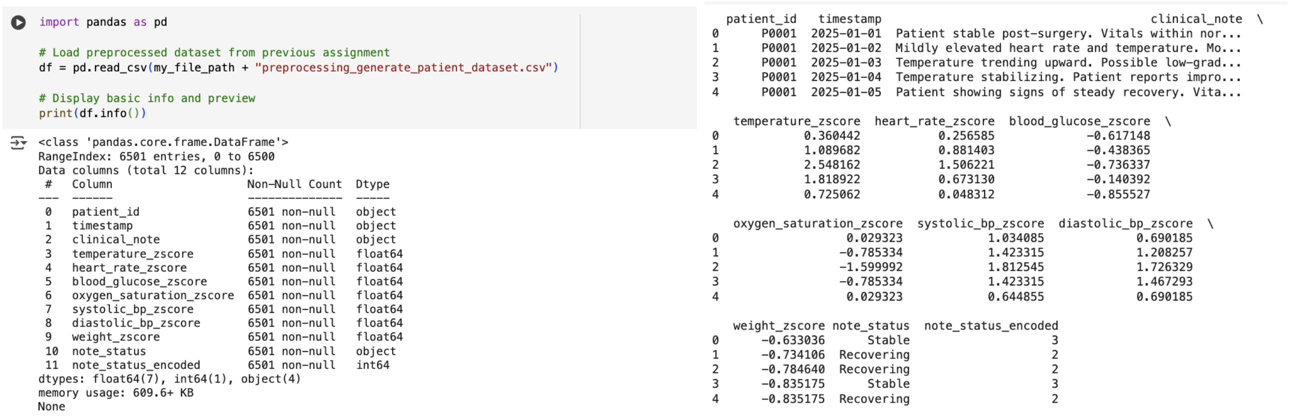
Figure 1. AI-Assisted Patient Health Deterioration Prediction Pipeline

## 2.1 Dataset Preparation

The synthetic patient dataset used in this project was generated in a previous assignment. The dataset contains 6,501 daily records of 500 patients and each record contains vital signs, clinical notes, and corresponding health status labels. The health status labels (note_status) are created using zero-shot classification (BART-MNLI) based on the clinical notes. In this project, the preprocessed dataset was loaded from the prepared CSV file for further analysis.

## 2.2 Data Preprocessing

The preprocessed patient dataset was loaded and validated for completeness and consistency (Figure 2). Vital sign features were standardized using z-score normalization to align feature scales, and clinical note statuses were encoded into numerical labels for multi-class classification.A class distribution analysis was performed to assess label balance (Figure 3), which revealed moderate imbalance across classes. These preprocessing steps ensured data readiness for subsequent feature engineering and modeling.



```python
import pandas as pd

# Load preprocessed dataset from previous assignment
df = pd.read_csv(my_file_path + "preprocessing_generate_patient_dataset.csv")

# Display basic info and preview
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6501 entries, 0 to 6500
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   patient_id                6501 non-null   object
 1   timestamp                 6501 non-null   object
 2   clinical_note             6501 non-null   object
 3   temperature_zscore        6501 non-null   float64
 4   heart_rate_zscore         6501 non-null   float64
 5   blood_glucose_zscore      6501 non-null   float64
 6   oxygen_saturation_zscore  6501 non-null   float64
 7   systolic_bp_zscore        6501 non-null   float64
 8   diastolic_bp_zscore       6501 non-null   float64
 9   weight_zscore             6501 non-null   float64
 10  note_status               6501 non-null   object
 11  note_status_encoded       6501 non-null   int64
dtypes: float64(7), int64(1), object(4)
memory usage: 609.6+ KB
None
```

```
  patient_id   timestamp                                      clinical_note  \
0      P0001  2025-01-01  Patient stable post-surgery. Vitals within nor...
1      P0001  2025-01-02  Mildly elevated heart rate and temperature. Mo...
2      P0001  2025-01-03  Temperature trending upward. Possible low-grad...
3      P0001  2025-01-04  Temperature stabilizing. Patient reports impro...
4      P0001  2025-01-05  Patient showing signs of steady recovery. Vita...

   temperature_zscore  heart_rate_zscore  blood_glucose_zscore  \
0            0.360442           0.256585             -0.617148
1            1.089682           0.881403             -0.438365
2            2.548162           1.506221             -0.736337
3            1.818922           0.673130             -0.140392
4            0.725062           0.048312             -0.855527

   oxygen_saturation_zscore  systolic_bp_zscore  diastolic_bp_zscore  \
0                  0.029323            1.034085             0.690185
1                 -0.785334            1.423315             1.208257
2                 -1.599992            1.812545             1.726329
3                 -0.785334            1.423315             1.467293
4                  0.029323            0.644855             0.690185

   weight_zscore note_status  note_status_encoded
0      -0.633036      Stable                    3
1      -0.734106  Recovering                    2
2      -0.784640  Recovering                    2
3      -0.835175      Stable                    3
4      -0.835175  Recovering                    2
```

Figure 2. Dataset Structure and Sample Preview

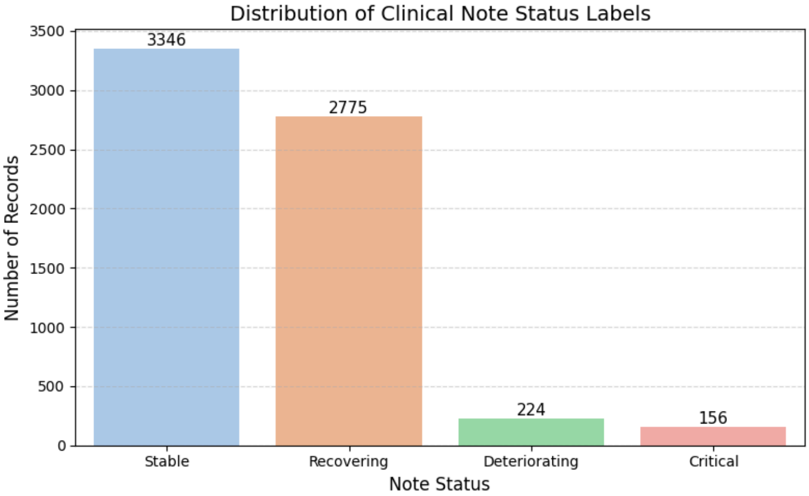
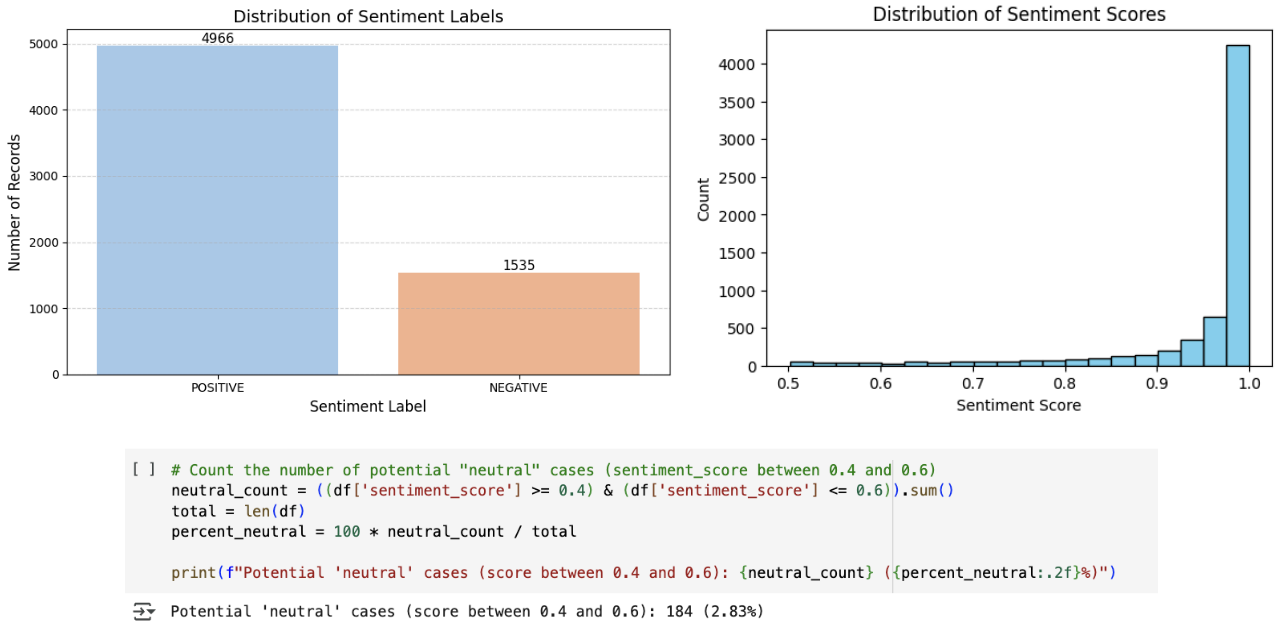


Figure 3. Distribution of Clinical Note Status Labels

## 2.3 Feature Engineering

To enhance model performance, this project incorporated advanced NLP-based feature engineering on clinical notes. First, sentiment analysis was performed using the DistilBERT model to generate sentiment labels and sentiment scores for each record. The sentiment scores were analyzed to assess feature suitability, with only 2.83% of records falling within an ambiguous neutral range (Figure 4). Sentiment labels were encoded as binary features for modeling.

In addition, MiniLM sentence embeddings were generated using the all-MiniLM-L6-v2 model, resulting in 384-dimensional semantic vectors that capture the contextual meaning of clinical text. These embeddings were integrated into the feature matrix (Figure 5), which combined normalized vital signs, encoded sentiment features, and MiniLM embeddings, yielding a comprehensive 393-feature input space for model training.

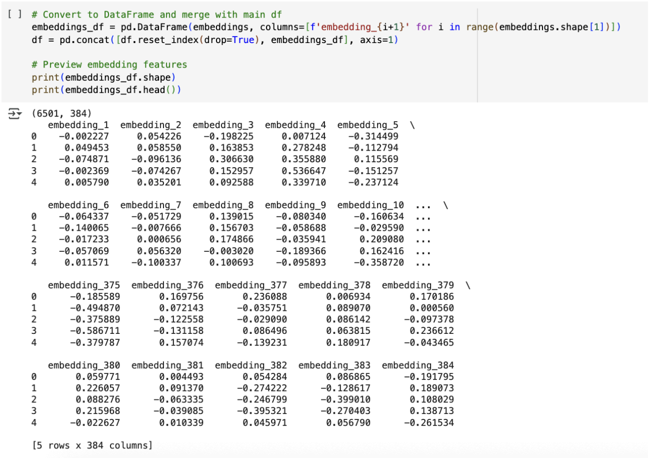


```
[ ] # Count the number of potential "neutral" cases (sentiment_score between 0.4 and 0.6)
    neutral_count = ((df['sentiment_score'] >= 0.4) & (df['sentiment_score'] <= 0.6)).sum()
    total = len(df)
    percent_neutral = 100 * neutral_count / total

    print(f"Potential 'neutral' cases (score between 0.4 and 0.6): {neutral_count} ({percent_neutral:.2f}%)")
```

```
Potential 'neutral' cases (score between 0.4 and 0.6): 184 (2.83%)
```

Figure 4. Distribution of Sentiment Labels, Sentiment Scores and Analyzed Feature Suitability



```
[ ] # Convert to DataFrame and merge with main df
    embeddings_df = pd.DataFrame(embeddings, columns=[f'embedding_{i+1}' for i in range(embeddings.shape[1])])
    df = pd.concat([df.reset_index(drop=True), embeddings_df], axis=1)

    # Preview embedding features
    print(embeddings_df.shape)
    print(embeddings_df.head())
```

```
(6501, 384)
   embedding_1  embedding_2  embedding_3  embedding_4  embedding_5  \
0    -0.002227     0.054226    -0.198225     0.007124    -0.314499
1     0.049453     0.058550     0.163853     0.278248    -0.112794
2    -0.074871    -0.096136     0.306630     0.355880     0.115569
3    -0.002369    -0.074267     0.152957     0.536647    -0.151257
4     0.005790     0.035201     0.092588     0.339710    -0.237124

   embedding_6  embedding_7  embedding_8  embedding_9  embedding_10  ...  \
0    -0.064337    -0.051729     0.139015    -0.080340     -0.160634  ...
1    -0.140065    -0.007666     0.156703    -0.058688     -0.029590  ...
2    -0.017233     0.000656     0.174866    -0.035941      0.209080  ...
3    -0.057069     0.056320    -0.003020    -0.189366      0.162416  ...
4     0.011571    -0.100337     0.100693    -0.095893     -0.358720  ...

   embedding_375  embedding_376  embedding_377  embedding_378  embedding_379  \
0      -0.185589       0.169756       0.236088       0.006934       0.170186
1      -0.494870       0.072143      -0.035751       0.089070       0.000560
2      -0.375889      -0.122558      -0.029090       0.086142      -0.097378
3      -0.586711      -0.131158       0.086496       0.063815       0.236612
4      -0.379787       0.157074      -0.139231       0.180917      -0.043465

   embedding_380  embedding_381  embedding_382  embedding_383  embedding_384
0       0.059771       0.004493       0.054284       0.086865      -0.191795
1       0.226057       0.091370      -0.274222      -0.128617       0.189073
2       0.088276      -0.063335      -0.246799      -0.399010       0.108029
3       0.215968      -0.039085      -0.395321      -0.270403       0.138713
4      -0.022627       0.010339       0.045971       0.056790      -0.261534

[5 rows x 384 columns]
```

Figure 5. Preview of MiniLM Sentence Embedding Features

## 2.4 Predictive Modeling

To support model development, the dataset was split into stratified training and test sets with an 80/20 ratio to preserve class distribution. SMOTE was applied to the training set to address class imbalance, resulting in a balanced dataset for model training.

A diverse set of predictive models was implemented to explore different modeling strategies. Traditional machine learning models included Random Forest, XGBoost, and MLP Neural Network. These models utilized a feature set combining z-score normalized vital signs, sentiment label and score derived from clinical notes, and MiniLM-based sentence embeddings.

In contrast, Transformer-based models (BERT-base, BioBERT, and DeBERTa) were fine-tuned directly on the raw clinical notes. This approach enabled the models to capture rich contextual information from unstructured text, supporting more nuanced health status prediction.

Each model was trained on the balanced training set and prepared for evaluation on the untouched test set. Key hyperparameters for each model were selected based on empirical testing and standard recommendations. For

Transformer-based models, fine-tuning was performed with a learning rate of 2e-5, batch size of 8, and 5 training epochs.

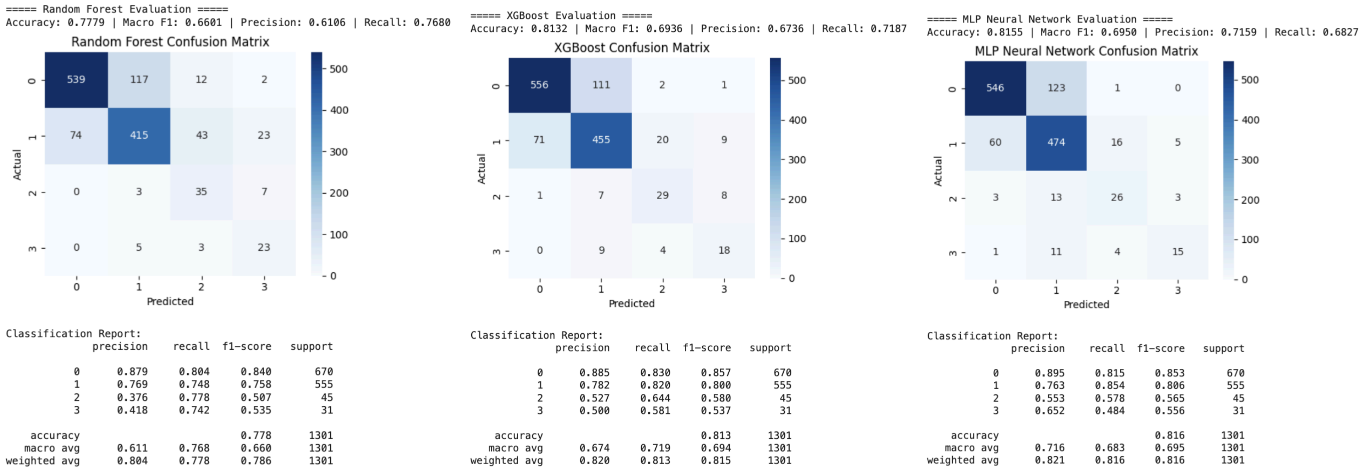
## 2.5 Model Evaluation and Interpretation

Model performance was assessed on the untouched test set using multiple metrics: Accuracy, Macro F1-score, Precision, and Recall, providing a comprehensive view of classification quality across imbalanced classes.

Among traditional models (Figure 6), XGBoost and MLP Neural Network outperformed Random Forest. Specifically, MLP achieved a balanced performance with an Accuracy of 81.55% and Macro F1-score of 0.6950 after SMOTE balancing, while XGBoost reached an Accuracy of 81.32% and F1-score of 0.6936. These results indicate that enriching structured features with sentiment and MiniLM embeddings improved the capacity of traditional models.

In contrast, Transformer-based models (Figure 7) demonstrated superior performance. DeBERTa achieved the best overall results, with an Accuracy of 87.93%, Macro F1-score of 0.7732, Precision of 0.7904, and Recall of 0.7604. BioBERT and BERT-base also performed strongly, benefiting from end-to-end fine-tuning on clinical texts and leveraging contextualized representations.

Table 1 summarizes the performance comparison across all models. The integration of sentiment features and MiniLM embeddings significantly enhanced traditional models, while Transformer models further advanced predictive performance through contextual clinical text modeling.



```
===== Random Forest Evaluation =====
Accuracy: 0.7779 | Macro F1: 0.6601 | Precision: 0.6106 | Recall: 0.7680
```

```
Classification Report:
              precision    recall  f1-score   support

           0      0.879     0.804     0.840       670
           1      0.769     0.748     0.758       555
           2      0.376     0.778     0.507        45
           3      0.418     0.742     0.535        31

    accuracy                          0.778      1301
   macro avg      0.611     0.768     0.660      1301
weighted avg      0.804     0.778     0.786      1301
```

```
===== XGBoost Evaluation =====
Accuracy: 0.8132 | Macro F1: 0.6936 | Precision: 0.6736 | Recall: 0.7187
```

```
Classification Report:
              precision    recall  f1-score   support

           0      0.885     0.830     0.857       670
           1      0.782     0.820     0.800       555
           2      0.527     0.644     0.580        45
           3      0.500     0.581     0.537        31

    accuracy                          0.813      1301
   macro avg      0.674     0.719     0.694      1301
weighted avg      0.820     0.813     0.815      1301
```

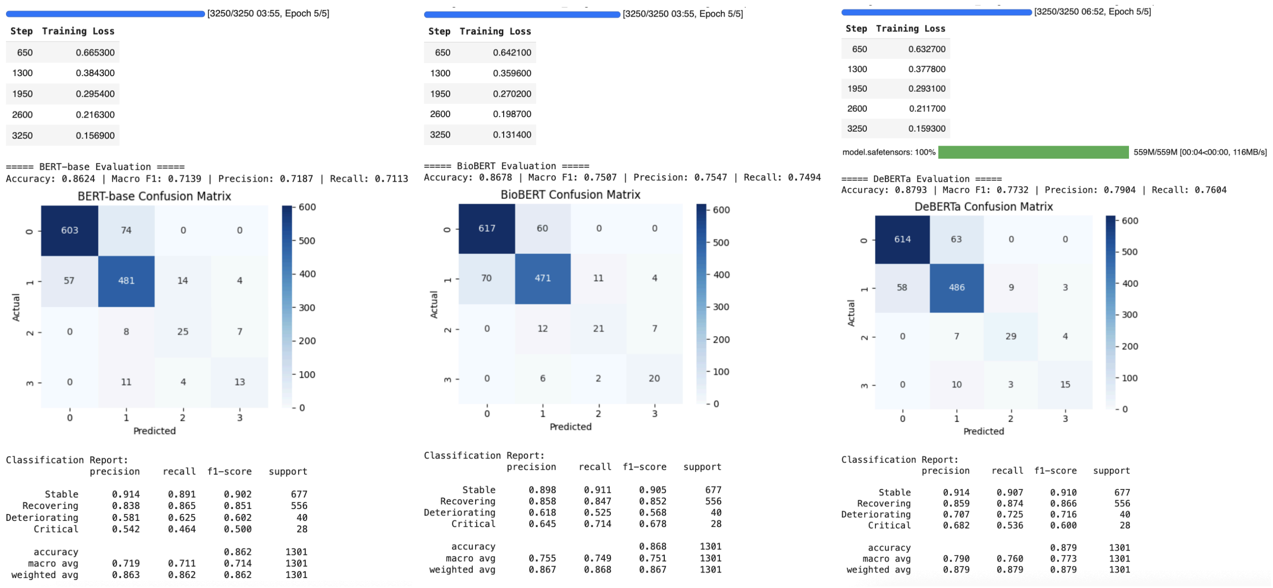
```
===== MLP Neural Network Evaluation =====
Accuracy: 0.8155 | Macro F1: 0.6950 | Precision: 0.7159 | Recall: 0.6827
```

```
Classification Report:
              precision    recall  f1-score   support

           0      0.895     0.815     0.853       670
           1      0.763     0.854     0.806       555
           2      0.553     0.578     0.565        45
           3      0.652     0.484     0.556        31

    accuracy                          0.816      1301
   macro avg      0.716     0.683     0.695      1301
weighted avg      0.821     0.816     0.816      1301
```

Figure 6. Confusion Matrices for Traditional Machine Learning Models (Random Forest, XGBoost, MLP Neural Network)



[3250/3250 03:55, Epoch 5/5]

| Step | Training Loss |
|---|---|
| 650 | 0.665300 |
| 1300 | 0.384300 |
| 1950 | 0.295400 |
| 2600 | 0.216300 |
| 3250 | 0.156900 |

```
===== BERT-base Evaluation =====
Accuracy: 0.8624 | Macro F1: 0.7139 | Precision: 0.7187 | Recall: 0.7113
```

BERT-base Confusion Matrix

```
Classification Report:
               precision    recall  f1-score   support

       Stable      0.914     0.891     0.902       677
   Recovering      0.838     0.865     0.851       556
Deteriorating      0.581     0.625     0.602        40
     Critical      0.542     0.464     0.500        28

     accuracy                          0.862      1301
    macro avg      0.719     0.711     0.714      1301
 weighted avg      0.863     0.862     0.862      1301
```

[3250/3250 03:55, Epoch 5/5]

| Step | Training Loss |
|---|---|
| 650 | 0.642100 |
| 1300 | 0.359600 |
| 1950 | 0.270200 |
| 2600 | 0.198700 |
| 3250 | 0.131400 |

```
===== BioBERT Evaluation =====
Accuracy: 0.8678 | Macro F1: 0.7507 | Precision: 0.7547 | Recall: 0.7494
```

BioBERT Confusion Matrix

```
Classification Report:
               precision    recall  f1-score   support

       Stable      0.898     0.911     0.905       677
   Recovering      0.858     0.847     0.852       556
Deteriorating      0.618     0.525     0.568        40
     Critical      0.645     0.714     0.678        28

     accuracy                          0.868      1301
    macro avg      0.755     0.749     0.751      1301
 weighted avg      0.867     0.868     0.867      1301
```

[3250/3250 06:52, Epoch 5/5]

| Step | Training Loss |
|---|---|
| 650 | 0.632700 |
| 1300 | 0.377800 |
| 1950 | 0.293100 |
| 2600 | 0.211700 |
| 3250 | 0.159300 |

model.safetensors: 100% 559M/559M [00:04<00:00, 116MB/s]

```
===== DeBERTa Evaluation =====
Accuracy: 0.8793 | Macro F1: 0.7732 | Precision: 0.7904 | Recall: 0.7604
```

DeBERTa Confusion Matrix

```
Classification Report:
               precision    recall  f1-score   support

       Stable      0.914     0.907     0.910       677
   Recovering      0.859     0.874     0.866       556
Deteriorating      0.707     0.725     0.716        40
     Critical      0.682     0.536     0.600        28

     accuracy                          0.879      1301
    macro avg      0.790     0.760     0.773      1301
 weighted avg      0.879     0.879     0.879      1301
```

Figure 7. Confusion Matrices for Transformer-based Models (BERT-base, BioBERT, DeBERTa)

| Model | F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Random Forest | 0.6601 | 0.7779 | 0.6106 | 0.7680 |
| XGBoost | 0.6936 | 0.8132 | 0.6736 | 0.7187 |
| MLP Neural Network | 0.6950 | 0.8155 | 0.7159 | 0.6827 |
| BERT-base | 0.7139 | 0.8624 | 0.7187 | 0.7113 |
| BioBERT | 0.7507 | 0.8678 | 0.7547 | 0.7494 |
| DeBERTa | 0.7732 | 0.8793 | 0.7904 | 0.7604 |

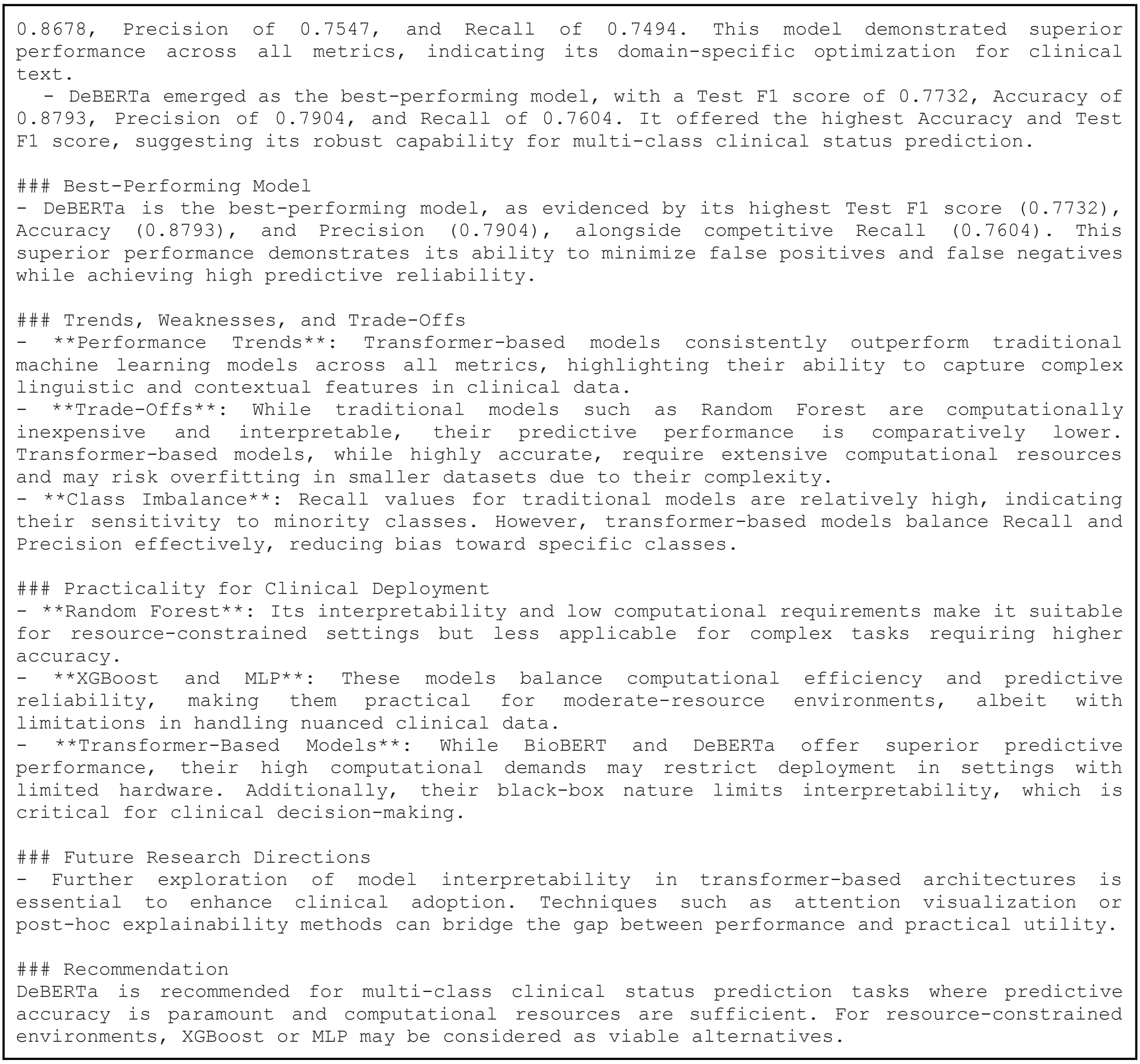
Table 1. Performance Comparison of All Models (Accuracy, F1-score, Precision, Recall)

To enhance interpretability, GPT-4o was used to assist in summarizing performance trends, cross-model comparisons, and practical implications based on the model evaluation results. The LLM-generated summary (Figure 8) highlights key insights and trade-offs between traditional and Transformer-based models, supporting evidence-based recommendations for real-world clinical applications.

```
### Comparative Analysis of Model Performance

- **Traditional Machine Learning Models**:
   - Random Forest achieved a Test F1 score of 0.6601, Accuracy of 0.7779, Precision of
0.6106, and Recall of 0.768. While demonstrating relatively strong Recall, its lower
Precision indicates a higher likelihood of false positives.
  - XGBoost outperformed Random Forest with a Test F1 score of 0.6936, Accuracy of 0.8132,
Precision of 0.6736, and Recall of 0.7187, achieving a more balanced performance across
metrics.
  - The Multi-Layer Perceptron (MLP) Neural Network showed marginal improvement over XGBoost
with a Test F1 score of 0.695, Accuracy of 0.8155, Precision of 0.7159, and Recall of 0.6827,
excelling particularly in Precision but reflecting slightly diminished Recall.

- **Transformer-Based Models**:
   - BERT-base demonstrated substantial performance enhancements over traditional models,
achieving a Test F1 score of 0.7139, Accuracy of 0.8624, Precision of 0.7187, and Recall of
0.7113. It balanced Precision and Recall effectively while improving overall classification
accuracy.
  - BioBERT further improved upon BERT-base, yielding a Test F1 score of 0.7507, Accuracy of
```



```
0.8678, Precision of 0.7547, and Recall of 0.7494. This model demonstrated superior
performance across all metrics, indicating its domain-specific optimization for clinical
text.
  - DeBERTa emerged as the best-performing model, with a Test F1 score of 0.7732, Accuracy of
0.8793, Precision of 0.7904, and Recall of 0.7604. It offered the highest Accuracy and Test
F1 score, suggesting its robust capability for multi-class clinical status prediction.

### Best-Performing Model
- DeBERTa is the best-performing model, as evidenced by its highest Test F1 score (0.7732),
Accuracy (0.8793), and Precision (0.7904), alongside competitive Recall (0.7604). This
superior performance demonstrates its ability to minimize false positives and false negatives
while achieving high predictive reliability.

### Trends, Weaknesses, and Trade-Offs
- **Performance Trends**: Transformer-based models consistently outperform traditional
machine learning models across all metrics, highlighting their ability to capture complex
linguistic and contextual features in clinical data.
- **Trade-Offs**: While traditional models such as Random Forest are computationally
inexpensive and interpretable, their predictive performance is comparatively lower.
Transformer-based models, while highly accurate, require extensive computational resources
and may risk overfitting in smaller datasets due to their complexity.
- **Class Imbalance**: Recall values for traditional models are relatively high, indicating
their sensitivity to minority classes. However, transformer-based models balance Recall and
Precision effectively, reducing bias toward specific classes.

### Practicality for Clinical Deployment
- **Random Forest**: Its interpretability and low computational requirements make it suitable
for resource-constrained settings but less applicable for complex tasks requiring higher
accuracy.
- **XGBoost and MLP**: These models balance computational efficiency and predictive
reliability, making them practical for moderate-resource environments, albeit with
limitations in handling nuanced clinical data.
- **Transformer-Based Models**: While BioBERT and DeBERTa offer superior predictive
performance, their high computational demands may restrict deployment in settings with
limited hardware. Additionally, their black-box nature limits interpretability, which is
critical for clinical decision-making.

### Future Research Directions
- Further exploration of model interpretability in transformer-based architectures is
essential to enhance clinical adoption. Techniques such as attention visualization or
post-hoc explainability methods can bridge the gap between performance and practical utility.

### Recommendation
DeBERTa is recommended for multi-class clinical status prediction tasks where predictive
accuracy is paramount and computational resources are sufficient. For resource-constrained
environments, XGBoost or MLP may be considered as viable alternatives.
```

Figure 8. AI-Assisted Comparative Summary of Model Performance Trends and Practical Implications

## 3. Discussion and Future Recommendations

The evaluation results demonstrate that Transformer-based models consistently outperformed traditional machine learning approaches in predicting patient health deterioration from clinical notes and vital signs. The top-performing model is DeBERTa and achieved the highest accuracy and F1-score, highlighting its ability to leverage contextual understanding of clinical text. BioBERT also performed strongly, benefiting from domain-specific pretraining. Among traditional models, XGBoost and MLP Neural Network showed competitive results with relatively lower computational demands, making them practical options for resource-constrained settings.

The integration of sentiment features and MiniLM embeddings significantly improved the performance of traditional models. These features provided additional semantic and emotional information, allowing models to better differentiate between health status categories. Figure 8 presents an AI-assisted comparative summary of model performance trends and practical implications generated using GPT-4o, further supporting these observations.

Although the results are promising, this study has several limitations. The synthetic dataset exhibits class imbalance, particularly for the Critical and Deteriorating categories. Although SMOTE was used to balance the training set, the test set remained imbalanced, which likely affected model performance on minority classes. The sentiment labels were generated using a pre-trained DistilBERT model; this approach introduces potential noise, as not all clinical notes clearly map to sentiment categories. Additionally, the MiniLM embeddings were generated from individual clinical notes and did not capture relationships across multiple notes for the same patient. The current modeling framework also treats each record as an independent sample, without considering the sequence of patient records over time. In real-world clinical practice, understanding how a patient's condition evolves across multiple days is often critical for accurate prediction.

Future research can address these limitations in several ways. First, incorporating temporal modeling techniques, such as Transformer architectures optimized for sequential data, could better capture patient health trajectories and disease progression patterns. Second, combining structured vital sign data with contextualized clinical text embeddings in a unified multi-modal architecture may further improve predictive performance and robustness. Validation on real-world clinical datasets and external cohorts is essential to assess model generalizability beyond synthetic data. Finally, further enhancing the use of LLM-assisted explanations to generate clinician-friendly summaries of model predictions would support more transparent and actionable clinical decision-making.

## 4. Conclusions

This project demonstrates that combining advanced NLP techniques with structured vital sign data enables effective prediction of patient health deterioration. Transformer-based models, particularly DeBERTa and BioBERT, achieved superior performance by leveraging contextual understanding of clinical text. Sentiment features and MiniLM embeddings further enhanced model performance across all approaches. While limitations remain, such as data imbalance and lack of temporal modeling, this work highlights the potential of AI-assisted pipelines to support more informed and transparent clinical decision-making. Future efforts will focus on validating these models on real-world clinical datasets and improving interpretability for clinical users.

## 5. AI Usage Disclosure

I utilised the AI writing tool ChatGPT (GPT-4o) to assist in refining the writing style and enhancing the clarity of the text. I also used GPT-4o to generate an interpretive summary of model performance trends and practical implications (Figure 8), based on the analysis and results I conducted. All AI outputs were carefully reviewed, edited, and supplemented with my own study analysis and understanding of the topic.