BENG0095 Group Assignment
Predicting Hospital Readmission Rates for Diabetic Patients

Group Name: GROUP D
Department of Biomechanical Engineering
University College London
London, WC1E 6BT

January 19, 2024

1 Introduction

The applications of machine learning are becoming increasingly popular in many disciplines, notably in the healthcare sector. Machine learning allows the performance of automated tasks previously attended to by physicians, and therefore aims for increased productivity from the hospital personnel. Machine learning methods enhance healthcare and medical research, as the number of electronic medical records is increasing. The most popular applications of machine learning in medicine are diagnosis and outcome prediction.

The objective of the classification task outlined in this essay is to automate the prediction of readmissions among diabetic patients discharged from hospitals. Our model aims to ascertain whether these patients are likely to be readmitted within 30 days, more than 30 days, or not readmitted at all. Our primary dataset includes 101,766 hospital visits spanning 130 different hospitals, and it offers insights into 50 unique features pertaining to these visits. The development and application of an accurate predictive model can greatly reduce the workload of physicians and improve the quality of decision-making.

2 Data Exploration

A key step in our project was data visualisation, which guided our approach to the classification problem. As previously mentioned, the initial training dataset contained 101,766 patient records, each offering 50 features related to their visit. A challenge arose due to the predominantly categorical nature of the data. For instance, in the case of max glucose level, the data categories were limited to above 200, above 300, or 'normal', without specific values. This restriction meant we could only conduct frequency-based analysis rather than average-based analysis. Additionally, this categorical data limited our ability to make robust estimations for significant features like weight, which was recorded in only about three percent of the dataset, yet is crucial for understanding readmission rates.

In the end, we chose to visualise our data by looking at the proportions of readmission rates depending on different categorical factors that we believed would have a significant impact - except for weight as there was little given data for it. We chose to use a proportionate view rather than absolutes to avoid the data being misleading through different population counts for factors such as race (race was made up largely of Caucasians, meaning that it was almost impossible to compare with other races using absolute values). Figure 1 demonstrates the key visualisations from the training data.

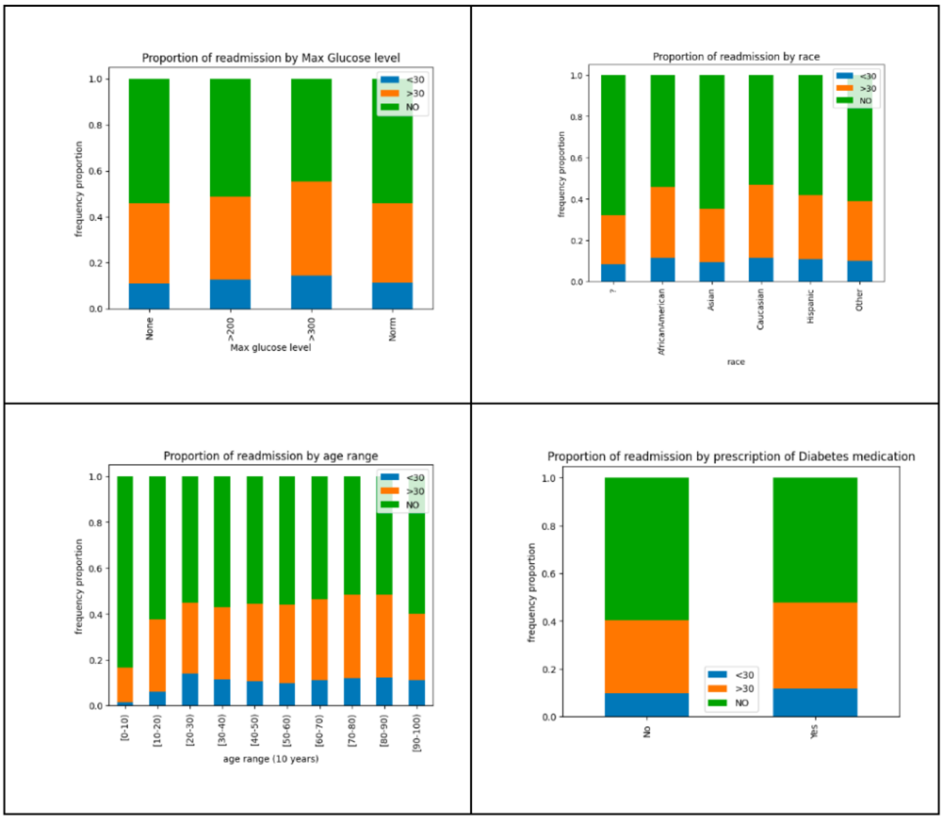


Figure 1: Key Visualisations for training data

2.1 Key Takeaways

- In the dataset, the *Race* feature does not seem to have a tangible impact on the frequency proportion of readmission or the time between each readmission. Contradicting with the standard; however, this is likely due to our dataset being skewed in terms of the amount of people of each race as mentioned prior.

- Within the *Age Range* feature, the categories between *10-100* show very similar frequency proportions, not having a very large impact on readmission rate except for *0-10*. This makes sense as children in the age range of *0-10* tend to be much healthier, and the common type of diabetes in this age range, type 1, is much less common in comparison to Type 2, and if there is Type 2, there tends to be a much shorter duration of disease.

- The prescription of medication features suggests that *"No"* has a lower frequency proportion of readmittance. This could be due to a variety of factors, from disease severity increasing over time and additional health conditions due to being immunocompromised. Finally, some potential patients may not be adhering to their medications.

- The *"max glucose level"* feature seems to show a relationship with readmittance frequency proportion at higher levels. This could be due to poor blood sugar control as this indicates a higher consumption of sugar or organ damage. The higher levels can also impact overall health leading to other complications due to being in a diabetic ketoacidosis state or hyperosmolar hyperglycemic state.

- Another finding was that the features *"diag_1"*, *"diag_2"*, and *"diag_3"*, which are the primary, secondary, and additional secondary diagnoses, are highly categorical. They respectively have 848, 923, and 954 distinct values. The potential issues of working with a highly categorical feature are that firstly, it highly expands the dimensionality of the dataset containing these features. Hence, we knew we would need methods that could deal with the high dimensionality of columns like these.

The initial visualization provided a deeper insight into the dataset, particularly highlighting any skewness present. Recognizing the predominantly categorical nature of the data, we opted for minimal feature selection in most of our approaches. This decision was informed by the nature of the tree-based and deep-learning methods selected for our model, which do not necessitate extensive feature selection for performance enhancement due to their robustness in dealing with irrelevant features. While it's true that eliminating unnecessary features can enhance computational efficiency and potentially quicken the learning rate, in this context, the impact on our model's overall effectiveness would likely be minimal. However, what we did want to consider was the imbalance in the classes as the dataset was heavily skewed towards the *"NO"* readmission with around 54 percent of the data, the *"over 30"* equal to 35 percent, and then very little for *"under 30"* with 11 percent.

3 Methodology

Our approach to this classification problem combines two methods: First, we apply manual feature engineering and a sampling strategy to feed into Random Forest (RF), XGBoost, and MLP classifiers. Second, we employ limited feature engineering for fine-tuning BERT. This strategy allows us to compare both traditional machine learning techniques and deep learning capabilities to solve the classification problem at hand.

3.1 Decision tree-based models

XGBoost works well with tabular data, and as explained in the Data Visualization section, our data is very categorical and has several dimensions (Hong et al., 2022). This is because it is robust to randomness, fast and stable. It also includes regularization in its objective function, which can reduce overfitting in a dataset with a large number of categorical features, such as ours. We also employ Random forests. Random Forests and XGBoost are both decision tree-based models, but Random Forests uses a bagging approach to reduce overfitting instead of employing gradient boosting with regularization (Hong et al., 2022). Just like XGBoost, it performs automatic feature selection due to the importance it attributes to relevant features, which is useful for datasets such as ours. According to literature, XGBoost often offers a stronger performance over Random Forest (Hong et al., 2022).

3.2 MLP classifier

Literature states that tree-based models are highly performant on tabular data and generally outperform deep learning methods (Grinsztajn et al., 2022). However, we still wanted to use some deep learning models given the highly dimensional nature of our data. Firstly, we used a Multi-Layer Perceptron (MLP) classifier. An MLP is a type of neural network that consists of multiple layers of nodes or neurons, each connected to others across layers. The 'multi-layer' aspect refers to the presence of one or more hidden layers situated between the input and output layers. Our MLP classifier was configured with two hidden layers, each layer composed of a set of neurons that process inputs from the previous layer. The use of the Rectified Linear Unit (ReLU) activation function in these hidden layers is a strategic choice: ReLU helps the model learn non-linear patterns efficiently and is known for its effectiveness in preventing the vanishing gradient problem. By using an MLP classifier alongside the tree-based methods on the same dataset, we sought to capitalize on the MLP's potential to more effectively capture non-linear relationships, a domain where it tends to yield good results.

3.3 BERT Fine-Tune

BERT stands for "Bidirectional Encoder Representations from Transformers". BERT is distinct for its deep learning approach that comprehends the context of words within their surrounding text. It achieves this by pre-training on a large amount of text data, employing techniques like masked language modeling and next-sentence prediction (Devlin et al., 2018). This pre-training enables BERT to effectively embed text, representing sentences and paragraphs in high-dimensional space.

The ability of BERT to represent text in such detail makes it suitable for text classification tasks. It can embed a text sequence, such as a group of sentences, and then classify it by adding a final layer. This layer, known as the sequence classification head, is randomly initialized and positioned on top of the encoder. Its primary function is to generate predictions across a predetermined number of classes. Integrating this classification layer with the pre-trained BERT model allows us to capitalize on its advanced text embedding capabilities. This method is commonly used for the classification of tweets by their sentiment (Gani and Chalaguine, 2022).

Fine-tuning involves taking a pre-trained model and further training it on a new dataset and task. This process builds upon the existing weights of the model, which were developed during its training on the initial dataset. The underlying idea is that the knowledge — represented by the model's weights — acquired from the original task can be beneficial for a new, related task. By continuing training from these pre-established weights, the model is refined, enhancing its proficiency in the new task without starting from scratch (Tajbakhsh et al., 2016). This approach leverages the foundational learning from the initial training, making it an efficient method for extending the model's capabilities to new domains or downstream tasks. In our case, fine-tuning will make the model better at embedding our patient data specifically.

However, fine-tuning all of the parameters of BERT's over 100 million parameters was unfeasible given the limited access to GPUs. Hence, a fine-tune via LoRA had to be done. LoRA stands for "low-rank adaptation" and is built on the concept that the weight matrices of layers within these large language models have low intrinsic rank, and hence can accurately be approximated by matrix decomposition. This suggests that during training, the updated weights can be calculated with these smaller decomposed weight matrices than the number of parameters in the original large weight matrix (Hu et al., 2021). In the context of fine-tuning BERT, consider its pre-trained weights as a matrix $W \in \mathbb{R}^{d \times h}$. Typically, during fine-tuning, the weights updated through backpropagation are added directly to this initial matrix, resulting in an updated weight matrix $W + W'$. However, with LoRA fine-tuning, the W' is approximated by matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times h}$, where r is significantly smaller than d and h. As can be seen, the number of trainable parameters decreases from $d \times h$ to $(d \times r) + (r \times h)$. r is a tunable parameter which determines the size and maximum rank of the decomposed matrices. Researchers who developed LoRA discovered that this method of matrix decomposition enables efficient fine-tuning thanks to the low intrinsic rank of the weight matrix W (Hu et al., 2021). By leveraging this property, LoRA maintains the model's performance while significantly reducing the computational overhead typically associated with fine-tuning large models like BERT, making it suitable for the task at hand given the lack of computing resources.

It must be noted that weight updates are specifically targeted at the attention layers within the BERT model. These layers are used to determine the relative importance of different parts of our text embeddings, directly influencing how the model classifies patient information. This focused updating of the attention layers allows the model to adapt more effectively to the nuances of our specific dataset.

4 Data Preprocessing

The decision trees and MLP classifier required a different dataset structurally than the BERT-fine tuning. There was an opportunity for more feature engineering for the first models compared to the fine-tune.

4.1 Decision trees and MLP classifier

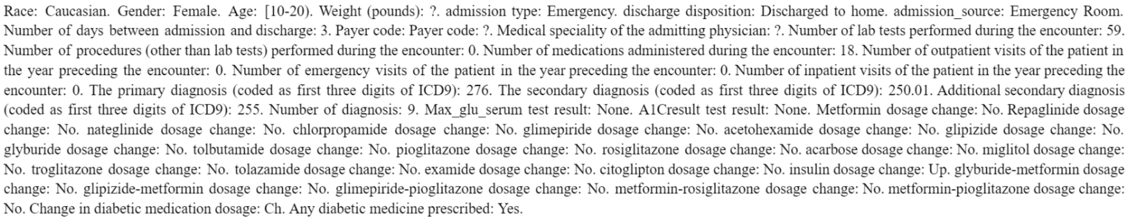
For these models, we tried three methods. In the first one, we applied very limited manual feature engineering where we one-hot encoded all columns with unique categories except for the continuous columns. This resulted in over 2000 columns per patient, used to establish a baseline understanding of our models' performance without further preprocessing.

In the second method, we employed a detailed approach to manual feature engineering. Our aim was to minimize the use of one-hot encoding for categorical variables due to the increased dimensionality it introduces. We transformed variables such as 'gender', 'age', and 'race' into numerical representations, reducing the complexity of our dataset. Another significant transformation was applied to the highly categorical columns `"diag_1"`, `"diag_2"`, and `"diag_3"`. Instead of one-hot encoding, which would have significantly increased the dimensionality, we converted these codes directly into numerical values, resulting in 215 features per patient. However, for some categorical columns, one-hot encoding was unavoidable, applied selectively to columns like `'admission_type_id'` and `'medical_specialty'` to balance between dimensionality reduction and retaining important categorical information.

The third method involved employing Synthetic Minority Over-sampling Technique (SMOTE) on this dataset with manual feature engineering. SMOTE generates synthetic samples for the minority class, balancing the class distribution. This technique is particularly effective in situations like ours, where class imbalance could potentially skew the model's learning process.

4.2 BERT Fine Tune

For the fine-tuning of BERT, we needed to structure the data such that the input encapsulated the entire patient context, and the output corresponded to an encoded classification. To achieve this, we concatenated all elements of a row across all columns, incorporating relevant information from the column names to provide context. This process resulted in a comprehensive, single-string representation for each patient, encapsulating all their data attributes. Figure 2 below demonstrates an example 'Patient context' for BERT fine-tune.



Race: Caucasian. Gender: Female. Age: [10-20). Weight (pounds): ?. admission type: Emergency. discharge disposition: Discharged to home. admission_source: Emergency Room. Number of days between admission and discharge: 3. Payer code: Payer code: ?. Medical speciality of the admitting physician: ?. Number of lab tests performed during the encounter: 59. Number of procedures (other than lab tests) performed during the encounter: 0. Number of medications administered during the encounter: 18. Number of outpatient visits of the patient in the year preceding the encounter: 0. Number of emergency visits of the patient in the year preceding the encounter: 0. Number of inpatient visits of the patient in the year preceding the encounter: 0. The primary diagnosis (coded as first three digits of ICD9): 276. The secondary diagnosis (coded as first three digits of ICD9): 250.01. Additional secondary diagnosis (coded as first three digits of ICD9): 255. Number of diagnosis: 9. Max_glu_serum test result: None. A1Cresult test result: None. Metformin dosage change: No. Repaglinide dosage change: No. nateglinide dosage change: No. chlorpropamide dosage change: No. glimepiride dosage change: No. acetohexamide dosage change: No. glipizide dosage change: No. glyburide dosage change: No. tolbutamide dosage change: No. pioglitazone dosage change: No. rosiglitazone dosage change: No. acarbose dosage change: No. miglitol dosage change: No. troglitazone dosage change: No. tolazamide dosage change: No. examide dosage change: No. citoglipton dosage change: No. insulin dosage change: Up. glyburide-metformin dosage change: No. glipizide-metformin dosage change: No. glimepiride-pioglitazone dosage change: No. metformin-rosiglitazone dosage change: No. metformin-pioglitazone dosage change: No. Change in diabetic medication dosage: Ch. Any diabetic medicine prescribed: Yes.

Figure 2: Example 'Patient context' for BERT fine-tune

The 'Patient Context' in Figure 2 showcases a comprehensive semantic profile, combining column names with corresponding data, which is ideal for BERT's fine-tuning. BERT's capability to embed and interpret this rich semantic information allows for effective classification.

5 Model Training

For all models, a 0.95 to 0.05 test-validation split was used to use as much data from the dataset during training. Additionally, ten-fold cross-validation was employed for one of the random forests models, but not for the other models due to a lack of computational resources and efficiency. For testing, the given test dataset was used, where the respective data transformations were appropriately applied. Figure 3 demonstrates the machine learning pipeline used for this project.

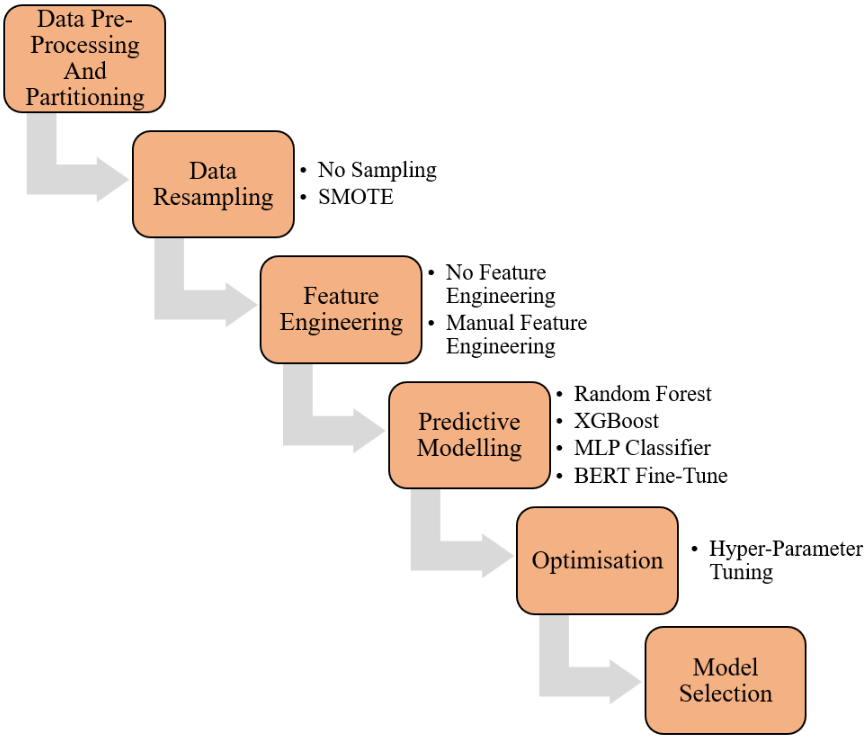


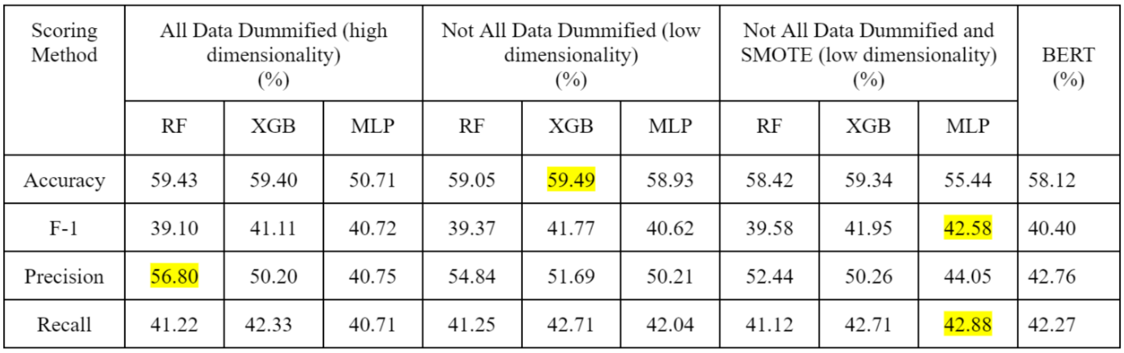
Figure 3: Machine Learning Pipeline for Patient Readmission Classification

Decision trees and MLP classifier were implemented using the Sci-kit Learn library on our datasets. For Random Forest and XGBoost models, default hyperparameters were used, while for the MLP Classifier, a specific architecture with two layers (100 neurons in the first layer and 50 neurons in the second layer), both using a ReLU activation function, was chosen.

The BERT fine-tuning took several hours to train due to the model's complexity. There were two hyper-parameters that we tuned and specifically selected which were r and α. As previously mentioned, the variable r represents the rank of the decomposed weight matrices in LoRA fine-tuning. The magnitude of r directly influences the size of these matrices: a larger r results in bigger matrices, leading to a greater number of weights requiring updates during training. Additionally, a higher value of r also implies more extensive changes to the model's weights. This can be beneficial, depending on the nature of the dataset, as it allows for the input data to have a greater influence on the weight adjustments during training. We set our r to 8, as it was the default recommended by the paper (Hu et al., 2021). α is a regularisation term used during training. Specifically, the ratio $\frac{\alpha}{r}$ is multiplied to the updated weight matrices. The larger this ratio is, suggests less regularisation, allowing for more substantial changes to the model's weights. We set our α equal to 16. We did some parameter tuning and found this combination was the best one. All other hyperparameters of the model are dealt with by Huggingface's "Parameter Efficient Fine Tuning (PEFT)" library that we used, which provides the optimal hyperparameters for this transfer learning task.

6 Results

For each model, the following scores are calculated: accuracy, precision, recall, and F1-Score. While accuracy is commonly used as the primary metric for scoring a model, it is less effective for unbalanced datasets. The F-1 score, on the other hand, calculates the mean of precision and recall, essentially ranking on false positives versus false negatives. In the context of predicting diabetes readmission, the F-1 score provides a balanced assessment of performance. Combining both F1 score and accuracy and prioritizing them in our ranking will ensure the model chosen from the ranking is the best for the given application (Yacouby, Axman and Alexa, n.d).



Scoring Method	All Data Dummified (high dimensionality) (%)			Not All Data Dummified (low dimensionality) (%)			Not All Data Dummified and SMOTE (low dimensionality) (%)			BERT (%)
	RF	XGB	MLP	RF	XGB	MLP	RF	XGB	MLP	
Accuracy	59.43	59.40	50.71	59.05	59.49	58.93	58.42	59.34	55.44	58.12
F-1	39.10	41.11	40.72	39.37	41.77	40.62	39.58	41.95	42.58	40.40
Precision	56.80	50.20	40.75	54.84	51.69	50.21	52.44	50.26	44.05	42.76
Recall	41.22	42.33	40.71	41.25	42.71	42.04	41.12	42.71	42.88	42.27

Figure 4: Results across all combinations of models, demonstrating the model's Accuracy, F-1, Precision, and Recall scores

The results presented in Figure 4 compare the performance of the machine learning models in patient readmission prediction across different data preprocessing techniques. The highlighted figures indicate the highest scores achieved by each model under different conditions. Overall, XGBoost performed the best across all data preprocessing techniques, with its highest accuracy score coming from the not all data dummified dataset. This model also has comparable metrics for F-1, precision, and recall. Overall, random forests arguably performed second best but seemed to benefit from the higher dimensionality of the all-data dummified dataset with limited feature engineering, as it performed best on this dataset, which can be seen in its high precision. It can be noted that implementing SMOTE seemed to marginally reduce scores for both random forests and XGBoost. However, the MLP classifier with SMOTE implemented excelled in F1 and Recall, indicating its proficiency in balancing precision and the correct identification of true positives. Despite this, it exhibited lower overall accuracy for this dataset. The MLP classifier didn't perform very well, especially for the all-data dummified dataset, demonstrating that it may not have been the right model for this task. The BERT fine-tune performed the worst out of all models for this task, achieving 58.12 percent accuracy.

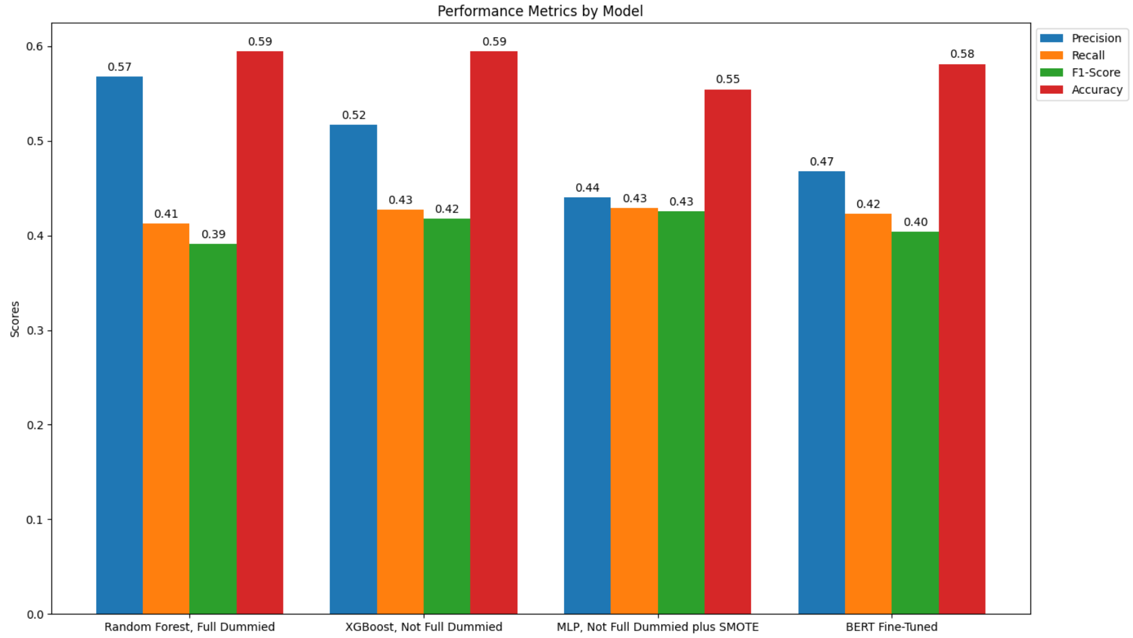


Figure 5: Peak Performance Metrics of Random Forest, XGBoost, MLP, and BERT models

Figure 5 depicts a bar chart comparing the top-performing models within each algorithm category. For Random Forest and XGBoost, we focused on models that excelled in Precision and Accuracy. The MLP classifier was selected based on superior F1-Score and Precision performance, while for BERT, we display the sole fine-tuned variant. For each distinct dataset type, a specific model architecture yielded the best results: Random Forest excelled with fully dummified data, XGBoost showed superior performance on data that was not fully dummified, and the Multi-Layer Perceptron (MLP) was most effective on partially dummified data with SMOTE sampling applied.

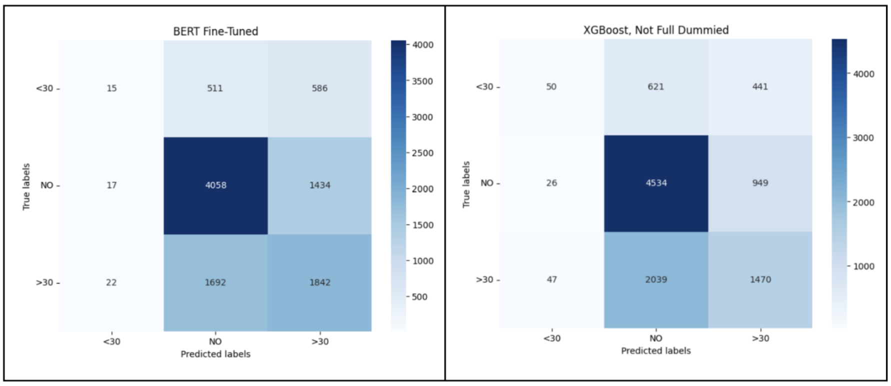


Figure 6: Confusion matrix heatmaps for BERT-Fine Tuned and XGBoost (not all dummified data)

To have a better understanding of how the model was making false predictions, we decided to create confusion matrix heatmaps for all our models. The two that we found interesting can be seen in Figure 5. Based on our findings in Figures 5 and 6, we made the visualization seen in Figure 6.

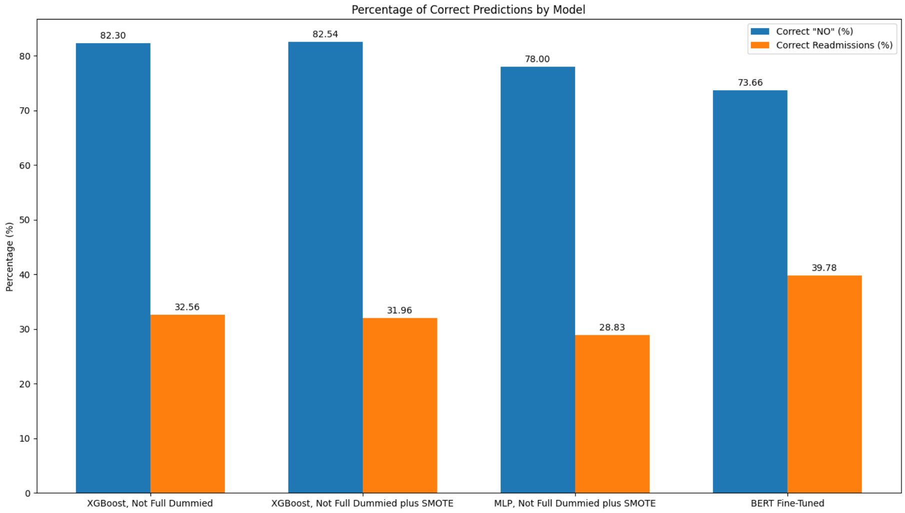


Figure 7: Percentage of Correctly predicting “No readmission” versus “Readmission”

Figure 7 presents a bar chart comparing the accuracy of various models in predicting patient readmissions. Our focus was on the general prediction of readmission, without distinguishing between readmissions occurring within or beyond 30 days. The chart categorizes the accuracy into two sections: correct predictions of non-readmissions (labeled "NO") and of readmissions. We compared four models: two leading XGBoost models, the MLP with the highest F-1 score, and BERT. The XGBoost model, which was not fully dummified with SMOTE, had an 82.54 percent accuracy rate for predicting non-readmissions and 32.96 percent for readmissions, with the other two non-BERT models showing similar performance. In contrast, the BERT model, specifically fine-tuned for this task, had a 73.66 percent accuracy for non-readmissions and a significantly higher 39.78 percent accuracy for readmissions. This finding is noteworthy as it demonstrates BERT’s superior ability to distinguish between no readmission and readmission cases, regardless of the 30-day timeframe. This aspect could be more crucial in clinical settings because predicting readmission is more important than distinguishing between whether they are readmitted within 30 days or in more than 30 days.

7 Conclusion

In conclusion, the analysis of different machine learning models, feature engineering methods, and sampling techniques for predicting patient readmissions, as presented through the results and figures, demonstrates implications for the classification task at hand. XGBoost emerged as the most effective model overall, particularly with the dataset where not all data was dummified. It displayed a balanced performance across accuracy, F1, precision, and recall metrics. The Random Forest model also showed promising results, especially in scenarios with high dimensionality data, indicating its suitability for complex datasets. However, the application of SMOTE seemed to slightly reduce the performance of both XGBoost and Random Forest models. The MLP classifier, while excelling in F1 and recall with SMOTE, demonstrated limitations in overall accuracy, especially with the all-data dummified dataset. This suggests that while MLP is proficient in balancing false positives and negatives, it may not be the most reliable for this specific application. In addition, the manual feature engineering marginally improved accuracy levels for Random Forests and XGBoost, but significantly improved the MLP classifier’s performance.

The BERT model, despite its lower overall accuracy, showed a unique strength in distinguishing between no readmission and readmission cases, without the need to differentiate based on the 30-day timeframe. This capability of the BERT model is particularly significant in clinical settings where the primary concern is often the prediction of readmission in general, rather than the specific timeframe within which it occurs. It is interesting to see that the semantic understanding this model could capture had a different impact on our results compared to the other classical machine learning techniques used. A future line of research would be to focus on fine-tuning the BERT model specifically for the binary classification of hospital readmission versus no readmission to see how the model performs. It would also be interesting to try fine-tuning without LoRA to see if improved performance is achieved by retraining all weights directly.

Bibliography

1. Chawla, N.V., Bowyer, K.W., Hall, L.O. and W. Philip Kegelmeyer (2002). *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research, [online] 16, pp. 321–357. doi:https://doi.org/10.1613/jair.953.

2. Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* [online] arXiv.org. Available at: https://arxiv.org/abs/1810.04805.

3. Gani, R. and Chalaguine, L. (2022). *Feature Engineering vs BERT on Twitter Data.* [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2210.16168.

4. Grinsztajn, L., Oyallon, E. and Varoquaux, G. (2022). *Why Do tree-based Models Still Outperform Deep Learning on Tabular data?* arXiv:2207.08815 [cs, stat]. [online] Available at: https://arxiv.org/abs/2207.08815.

5. Hong, W., Zhou, X., Jin, S., Lu, Y., Pan, J., Lin, Q., Yang, S., Xu, T., Zarrin Basharat, Zippi, M., Fiorino, S., , .., Stock, S., Grottesi, A., Chen, Q. and Pan, J. (2022). *A Comparison of XGBoost, Random Forest, and Nomograph for the Prediction of Disease Severity in Patients With COVID-19 Pneumonia: Implications of Cytokine and Immune Cell Profile.* Frontiers in Cellular and Infection Microbiology, 12. doi:https://doi.org/10.3389/fcimb.2022.819267.

6. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2021). *LoRA: Low-Rank Adaptation Of Large Language Models.* [online] Available at: https://arxiv.org/pdf/2106.09685.pdf.

7. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J. (2016). *Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?* IEEE Transactions on Medical Imaging, 35(5), pp. 1299–1312. doi: https://doi.org/10.1109/tmi.2016.2535302.

8. Yacouby, R., Axman, D. and Alexa, A. (n.d.). *Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models.* [online] Available at: https://aclanthology.org/2020.eval4nlp-1.9.pdf.