THE UNIVERSITY OF SYDNEY

COMP5310 Project Stage 2

Develop and evaluate a predictive model

Mental health in the pregnancy during the COVID-19

Contents

1 Setup

1.1 Topic and research question

The COVID-19 pandemic has profoundly affected global mental health, disproportionately impacting pregnant individuals. Amidst this crisis, our research investigates the psychological effects on expectant mothers, focusing on how socioeconomic status, pandemic-related health perceptions, and neonatal outcome concerns drive their anxiety and depression. Our pivotal question is: **How do socioeconomic factors, perceived pandemic risks, and neonatal outcome concerns make pregnant women depressed during the COVID-19 era?**

Addressing this question promises to unravel the complex mental health ramifications of the pandemic on expectant mothers, benefiting key stakeholders:

- **Academic Researchers:** Our insights enable researchers to explore the pandemic's psychological dimensions further, enriching academic discourse and sparking targeted investigations into maternal stressors during health crises.
- **Clinical Practitioners:** Therapists and healthcare professionals gain crucial data to refine intervention strategies, offering personalized support to mitigate pregnant women's mental health challenges.
- **Policymakers:** With a deeper comprehension of the psychological impact on pregnant women, policymakers can develop targeted supports, crafting policies that address the unique needs of expectant mothers in times of crisis.

Our findings aim to facilitate a multidisciplinary approach to understanding and addressing the psychological impacts of the pandemic, contributing to a foundation for future pregnant women and their family's health.

1.2 Dataset

Our dataset comprises a collection of 10,772 entries across 16 meticulously selected variables, aimed at painting a comprehensive picture of the psychological well-being of pregnant women amidst the COVID-19 pandemic.

Our primary dataset was sourced from the Kaggle platform [1] in January 2024, and discovered by our team two months later. The Canadian government delivers the original data in the form of surveys with English and French versions to all women 17 years and older, pregnant, residing in Canada, within 35 weeks of gestation, and proficient in English or French as part of the Pregnancy during the COVID-19 Pandemic (PdP) project[1]. After data cleaning, the dataset contains 10,749 rows and 21 columns, which qualifies as a relatively large dataset. Given the number of instances, the processing might be very time-consuming.

We cannot control the data collection process; therefore, we have no knowledge of potential biases introduced during the process. While cleaning the data, we imputed some missing values, which might have introduced biased data points. Additionally, we numerically encoded all categorical attributes, although some algorithms can handle categorical features directly. This encoding might also introduce biases.

1.3 Modelling agreements

Our study aims at the mental health of the woman during pregnancy and focuses on the Edinburgh Postnatal Depression Scale (EPDS) as a critical tool [2]. It helps to understand the level of intensity of different levels of depressive symptoms. **0-9:** Represents almost no depressive symptoms, with the label "Within Normal Limits". **10-12:** Represents slight depressive symptoms present, with the label "Mild". **13-15:** Represents moderate depressive symptom, with the label "Moderate". **16 and above:** Represents serious signs of depression need urgent professional help, with the label "Severe". Two measurements are below:

F1 Score : The F1 score represents the harmonic mean of precision and recall. It achieves its highest value at 1 and its lowest value at 0. [3]

Precision : Precision intuitively measures the classifier's capability to avoid misclassifying a negative sample as positive. [4]

[1] Dataset on the Kaggle platform: Mental health in the pregnancy during the COVID-19.

2 Evaluate a Predictive Model

2.1 Predictive model

2.1.1 Model description

XGBoost stands for eXtreme Gradient Boosting. It is an implementation of the gradient boosting decision tree algorithm. The XGBoost integrates regularization terms (L1 and L2), which help reduce overfitting and improve model generalization. It is highly efficient and easily transportable [5]. This method is special due to its high effectiveness in regression, classification, and ranking tasks [6].

Assumptions in the context of our dataset

Interactions of Features: The dataset includes numerical data and derived categorical classes from scores. Capturing interactions through trees is likely valid as these features can effectively reflect underlying patterns in a boosting framework.

Additive Predictors: Converting EPDS [2] and PROMIS Anxiety [7] scores into categorical classes could simplify the modeling process. This approach aligns well with the additive nature of boosting, allowing simple models to progressively capture intricate decision boundaries and enhance the effectiveness of the overall modelling strategy.

Independent Samples: Each person's result contributes a single record, fitting well with independent samples. The presence of a collection date does not impact the analysis since there is no repeated measure per individual.

Strengths of XGBoost for this research and dataset:

Scalability and Performance: It can quickly process huge datasets with a large number of features [5], which is indispensable for the computation of extensive clinical datasets.

Regularization: It can prevent overfitting, which is important when using derived categorical data with complex decision boundaries.

Limitations of XGBoost for this research and dataset:

Human-Understandable: While a few trees are understandable by humans, hundreds or thousands may not be easily interpreted. This can pose challenges if model interpretation is required.

Sensitive to Outliers: XGBoost is sensitive to outliers because subsequent classifiers try to compensate for the mistakes of previously built ones, leading to a significant impact on model training.

2.1.2 Model algorithm

XGBoost builds on traditional gradient boosting frameworks and introduces advanced concepts that optimize performance this is the pseudo code A.1 in the following two main underlying principles:

Gradient Boosting Framework

Gradient Descent Optimization: XGBoost uses gradient descent algorithms to minimize the loss when adding new models. Each tree is built to minimize a loss function. This is conceptually akin to stepping down a hill in a manner calculated to reach the bottom most efficiently.

Additive Training: Unlike methods that need to retrain existing models. When new data comes in XGBoost incrementally builds the model using an additive manner. It starts with a single model (usually a stump, comprising just a root node and two leaves), and successively adds new models that predict the residuals or errors of prior models combined.

$$\mathrm{Obj}(\theta) = L(\theta) + \Omega(\theta)$$

This is the Objective function where L is the loss function that measures the difference between the predicted and actual values, and Ω is the regularization term that penalizes the complexity of the model to avoid overfitting. This is the pseudo code for the objective function A.2.

Regularization

$$\Omega(\theta) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

T is the number of leaves in the tree,

w_j represents the weight of the j-th leaf,

γ is the parameter that controls the penalty for the number of leaves,

λ is the parameter that provides the L_2 regularization on the leaf weights.

A key feature of XGBoost over other implementations of gradient boosting is the inclusion of a regularization term in the loss function Ω. This helps to control over-fitting and ensuring the robustness of the model.

In this research task, we used a few most important and common hyperparameters in the fine tune phase [8].

- **max_depth:** Maximum depth of a tree. It controls the maximum depth of each tree in the boosting process.
- **learning_rate:** Learning rate controls the contribution of each tree to the final prediction. A lower learning rate requires more trees to be added to the ensemble but can improve generalization.
- **n_estimators:** Number of boosting rounds trees to be run. It determines the number of trees to fit, which is the number of boosting iterations.

There are still a lot of hyperparameters that could be used in the fine tun phase. The reason we do not use it in our model is mainly due to resource consumption and their impact on the quality of the model is not too obvious.

- **reg_alpha:** L1 regularization term on weights. It adds an L1 regularization term to the loss function, which can help prevent overfitting by penalizing large weights.
- **reg_lambda:** L2 regularization term on weights. It adds an L2 regularization term to the loss function, which can also help prevent overfitting by penalizing large weights.

2.1.3 Model development

Advanced data preprocessing

Due to XGBoost decision trees determine splits based on order statistics rather than feature scale. Also, it can handle high-dimensional data effectively due to its built-in regularization terms that penalize overly complex models. So the feature scaling and dimensional reduction are unnecessary for preprocessing.

For feature engineering, we have decided to use EPDS and PROMIS Anxiety as our target attributes. Since XGBoost is a gradient-boosted decision tree model, it is easier and more accurate to predict results within a smaller range.

The original EPDS and PROMIS Anxiety scores range from 1 to 30 and 1 to 100, respectively. It is challenging to train a model to predict results within these broad ranges by our dataset size. We have observed that both EPDS and PROMIS Anxiety scores can be categorized into four distinct stages. The method for dividing EPDS scores was previously discussed in section 1.3.1 on Predicted Attributes. The PROMIS is Patient Reported Outcomes Measurement Information System [9]. which could divided score as follows [10]

Below 55: Within Normal Limits ; **55-60:** Mild ; **60-70:** Moderate ; **Above 70:** Severe

The rationale behind the division of data into training

Divide data into training, validation and test sets could significantly help the model's training and evaluation. The training set is to be used to train the model or fit the model. The validation set is to help to fine tune or evaluate the model during the training stage. The test set is used to test the model performance which could treat it as unseen data in reality.

In this research, we have used the 'train_test_split' and 'GridSearchCV' in 'sklearn' package.

train_test_split: could split the dataset into training and test sets in a ratio of 0.8 : 0.2 with random and stratified splits by setting `test_size=0.2` and `stratify=y`.

GridSearchCV: could automatically complete the grid search and cross-validation steps. Cross-validation divides the dataset into subsets, allowing each part to serve as both training and validation data. For hyperparameters, we have used 'max_depth', 'learning_rate', and 'n_estimators' for all models, as they have the greatest influence on the XGBoost model. This is achieved by setting `param_grid=param_grid_list` and `cv=5`, where `param_grid_list` contains all the hyperparameter settings.

2.2 Model evaluation and optimization

2.2.1 Model evaluation

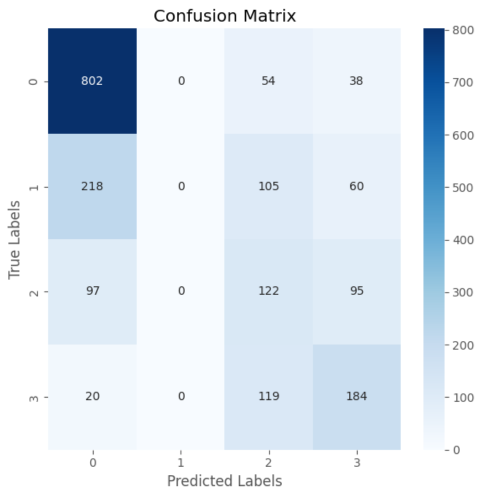
Model Performance

We have selected F1 score, Precision, and AUC-ROC curve as our success metrics. The mean F1 score is 0.41, mean Precision is 0.37, and mean AUC-ROC is 0.79. While the model may be performing better than random guessing, there is still room for improvement in terms of both precision and overall classification performance.

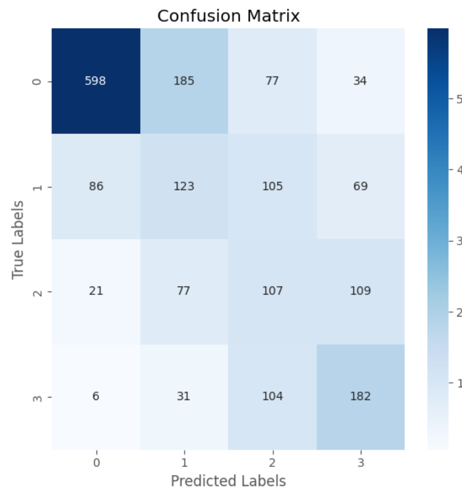
Upon evaluation, we observed that in the confusion matrix (Figure 1a), all entries in the column corresponding to class 1 are zero. This indicates the model's failure to correctly identify any true positives for this class, suggesting a potential class imbalance in the dataset.

Implications and Areas for Improvement

F1 Score, Precision and confusion matrix show that class 1 is zero, suggesting that the model either lacks sufficient information to make predictions for this class or the class is underrepresented. However, sample weights can address this issue.



(a) Before Optimization



(b) After Optimization

2.2.2 Model optimization

For advanced optimization, we integrated sample weights into our model and iteratively adjusted multiple hyperparameter values using grid search and cross-validation. This methodology empowered us to assign varying weights to individual samples during the training process.

By adopting this approach, we successfully tackled the issue of an empty class 1, as evidenced by the improvements observed in the second confusion matrix (Figure 1b). Additionally, our efforts resulted in noteworthy enhancements in performance metrics: the F1 score increased from 0.41 to 0.47, Precision from 0.37 to 0.47, and AUC-ROC to 0.77.

Although the model underwent optimization, the improvements in F1 score, precision, and AUC-ROC were marginal. This could be attributed to the model's imperfect alignment with the research or the inadequacy of the dataset for training XGBoost effectively.

3 Discussion

3.1 Classifiers

XGBoost is notable for its regularization feature, which is crucial for preventing overfitting in datasets with complex decision boundaries, such as ours. However, a significant limitation of XGBoost is its dependence on large volumes of training data. With only about 9000 records available in our research, XGBoost may not perform optimally. The final best result is F1 score is 0.47, Precision is 0.47.

3.2 Resampling

Resampling techniques, which help balance datasets without altering their distribution, have shown significant improvements in our models. Future investigations should explore the application of resampling across all models to identify the most suitable approach for our specific context.

3.3 Future Work

We have not yet considered 'Anxiety' as a potential target variable, despite its relevance to our current target 'Depression'. In future work, we could consider a multitask approach to leverage shared knowledge across these related tasks, potentially enhancing model learning. In addition, time efficiency, crucial in real-world applications, was not a focus in this study but should be considered in future evaluations to provide a more comprehensive comparison of the models.

4 Conclusion

Empirical analysis revealed the least important features to be 'Language', 'Delivery_Mode', and 'NICU_Stay', with 'Language' understandably having minimal impact. Conversely, the most critical features were identified as 'Delivery_Date', 'Threaten_Life', and 'Gestational_Age_At_Birth'. The significance of 'Delivery_Date' may be attributed to its overlap with critical periods such as the pandemic, increasing stress levels in expectant mothers. Similarly, perceived threats to life significantly contribute to stress, confirming our hypothesis on the impact of these factors on maternal depression. This analysis supports our research question, indicating the major role of specific factors in influencing maternal mental health.

References

[1] C. Lebel, L. Tomfohr-Madsen, G. Giesbrecht, *et al.*, "Prenatal mental health data and birth outcomes in the pregnancy during the covid-19 pandemic dataset," *Data in Brief*, vol. 49, p. 109 366, 2023, ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2023.109366. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923004857.

[2] J. L. Cox, J. M. Holden, and R. Sagovsky, "Detection of postnatal depression: Development of the 10-item edinburgh postnatal depression scale," *British Journal of Psychiatry*, vol. 150, no. 6, pp. 782–786, 1987. DOI: 10.1192/bjp.150.6.782.

[3] scikit-learn contributors. "Scikit-learn - f1 score documentation." (2024), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (visited on 05/07/2024).

[4] scikit-learn contributors. "Scikit-learn - precision_recall_fscore_support documentation." (2024), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support (visited on 05/07/2024).

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[6] NVIDIA, *Xgboost*, https://www.nvidia.com/en-us/glossary/xgboost/, Accessed: May 6, 2024.

[7] D. Cella, S. W. Choi, D. M. Condon, *et al.*, "Promis® adult health profiles: Efficient short-form measures of seven health domains," *Value in Health*, vol. 22, no. 5, pp. 537–544, 2019, ISSN: 1098-3015. DOI: https://doi.org/10.1016/j.jval.2019.02.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1098301519301184.

[8] T. Chen, C. Guestrin, and Contributors, *Xgboost parameters*, Accessed: May 6, 2024, XGBoost, 2022.

[9] PROMIS Health Organization, *Promis health*, https://www.promishealth.org/57461-2/, Accessed: May 6, 2024.

[10] HealthMeasures. "Promis anxiety scoring manual." (2023), [Online]. Available: https://www.healthmeasures.net/administrator/components/com_instruments/uploads/PROMIS%20Anxiety%20Scoring%20Manual_08Sept2023.pdf.

References

[1] C. Lebel, L. Tomfohr-Madsen, G. Giesbrecht, *et al.*, "Prenatal mental health data and birth outcomes in the pregnancy during the covid-19 pandemic dataset," *Data in Brief*, vol. 49, p. 109 366, 2023, ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2023.109366. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923004857.

[2] J. L. Cox, J. M. Holden, and R. Sagovsky, "Detection of postnatal depression: Development of the 10-item edinburgh postnatal depression scale," *British Journal of Psychiatry*, vol. 150, no. 6, pp. 782–786, 1987. DOI: 10.1192/bjp.150.6.782.

[3] scikit-learn contributors. "Scikit-learn - f1 score documentation." (2024), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (visited on 05/07/2024).

[4] scikit-learn contributors. "Scikit-learn - precision_recall_fscore_support documentation." (2024), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support (visited on 05/07/2024).

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[6] NVIDIA, *Xgboost*, https://www.nvidia.com/en-us/glossary/xgboost/, Accessed: May 6, 2024.

[7] D. Cella, S. W. Choi, D. M. Condon, *et al.*, "Promis® adult health profiles: Efficient short-form measures of seven health domains," *Value in Health*, vol. 22, no. 5, pp. 537–544, 2019, ISSN: 1098-3015. DOI: https://doi.org/10.1016/j.jval.2019.02.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1098301519301184.

[8] T. Chen, C. Guestrin, and Contributors, *Xgboost parameters*, Accessed: May 6, 2024, XGBoost, 2022.

[9] PROMIS Health Organization, *Promis health*, https://www.promishealth.org/57461-2/, Accessed: May 6, 2024.

[10] HealthMeasures. "Promis anxiety scoring manual." (2023), [Online]. Available: https://www.healthmeasures.net/administrator/components/com_instruments/uploads/PROMIS%20Anxiety%20Scoring%20Manual_08Sept2023.pdf.

A Appendix

Algorithm A.1 XGBoost Tree Boosting Algorithm

procedure XGBOOST
  Input: training set $\{(x_i, y_i)\}_{i=1}^{n}$, a differentiable loss function $\ell(y, \hat{y})$, number of iterations T
  Output: a series of trees $\{f_t\}_{t=1}^{T}$
  $\hat{y}_i^{(0)} \leftarrow 0$ ▷ Initialize predictions
  for $t \leftarrow 1$ to T **do** ▷ Loop over number of iterations
    $g_i \leftarrow \partial_{\hat{y}^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)})$ ▷ Compute gradient
    $h_i \leftarrow \partial^2_{\hat{y}^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)})$ ▷ Compute hessian
    $f_t \leftarrow \mathrm{tree}(X, g, h)$ ▷ Grow tree using grad. and hess.
    $\hat{y}_i^{(t)} \leftarrow \hat{y}_i^{(t-1)} + f_t(x_i)$ ▷ Update model predictions
  end for
end procedure
function TREE(X, g, h)
  Create an empty tree
  while not *max_ depth* **do**
  Find the best split based on $\sum_{i \in I_L} g_i^2 / (\sum_{i \in I_L} h_i + \lambda) + \sum_{i \in I_R} g_i^2 / (\sum_{i \in I_R} h_i + \lambda) - (\sum_{i \in I} g_i^2 / (\sum_{i \in I} h_i + \lambda))$
  Add split to tree
  end while
  return tree
end function

Algorithm A.2 XGBoost Objective Function

Require: Dataset $D = \{(x_i, y_i)\}$, number of trees K, learning rate η, regularization parameters γ, λ
  Initialize model $\phi(x) = 0$
  for $k = 1$ to K **do**
    Compute gradient statistics: $g_i = \frac{\partial l(y_i, \phi(x_i))}{\partial \phi(x_i)}$, $h_i = \frac{\partial^2 l(y_i, \phi(x_i))}{\partial \phi(x_i)^2}$
    Build tree f_k by minimizing:

$$\sum_{i=1}^{n} [g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$

    Update model: $\phi(x) \leftarrow \phi(x) + \eta f_k(x)$
  end for
  return $\phi(x)$